УДК 004.942

Р. Е. ШАРЫКИН

АПРОБАЦИЯ МОДЕЛИ СТОХАСТИЧЕСКОЙ КОЛЛАБОРАЦИОННОЙ ЗАЩИТЫ ОТ ВИРУСОВ

Белорусский Государственный Университет

В статье рассматривается реализация на языке Java модели стохастической коллаборационной защиты от вирусов, разработанной в рамках модели Распределенных Объектно-Ориентированных Стохастических Гибридных Систем (РООСГС), и ее анализ. Целью работы является апробация модели стохастической коллаборационной защиты от вирусов в условиях, приближенных к реальным, на пути к внедрению ее использования в реальном окружении. Излагается методика трансляции спецификации системы на языке SHYMaude, предназначенном для спецификации и анализа РООСГС в рамках переписывающей логики, в соответствующую реализацию алгоритма на языке Java. Система разворачивается на виртуальных машинах, вирус и система оповещения моделируются стохастически. Выделяется ряд метрик, таких как время до насыщения вируса, доля зараженных узлов по достижению насыщения, максимальная скорость распространения вируса. Для получения оценок выбранных метрик используется метод Монте-Карло с вычислением доверительных интервалов. Анализ проводится на основе сигмовидного графика распространения вируса по времени в присутствии системы защиты. Реализуются два протокола передачи сообщений между узлами, TCP/IP и UDP. Исследуется влияние типа протокола и сопряженных с ним издержек на эффективность системы защиты. Для оценки потенциала уменьшения издержек, связанных с деталями протокола, проводится анализ исходной модели РООСГС, модифицированной для моделирования таких издержек. Исследуется влияние других параметров модели, необходимых для перехода к следующим шагам внедрения данной модели в практическое использование. Предлагается иерархический подход к обобщению системы, позволяющий сделать систему масштабируемой на большое количество узлов.

Ключевые слова: *совместная антивирусная защита; математическое моделирование; распределенные системы; стохастические системы; статистический анализ.*

Введение

Ввиду растущей важности и сложности распределенных систем защиты от вирусов, применение формальных методов на различных этапах разработки, от модели до ее реализации в качестве готового приложения представляется важной задачей. Можно выделить такие этапы разработки, как построение предварительной математической модели, апробация модели в условиях, приближенных к реальным, фактическая реализация системы на основе доработанной с учетом всех обнаруженных аспектов модели.

В [1] рассматривается подходы к построению предварительной математической модели и статистическому анализу сложных динамических распределенных стохастических систем с коммуникацией, а также применение данных подходов к построению и анализу модели коллаборационной стохастической системы защиты от вирусов.

За основу рассматриваемой системы защиты взята система, предложенная в [2]. Система в [2] имела недостаток, заключающийся в возможности нахождения «успешной» атаки в том смысле, что был найдена методика построения атаки [3], следуя которой вирус имел возможность заразить все узлы системы. Данная методика базировалась на использовании формальной модели системы и системы автоматического доказательства теорем. Система автоматического доказательства теорем находила контрпример к формально выраженному утверждению, что не существует последовательности действий вируса приводящей к полному заражению системы. Данный контрпример служил «рецептом» заражения всей системы.

Для преодоления данной проблемы в [1] было предложено сделать систему защиты стохастической. Методика в [3] использовала тот факт, что группы оповещения о заражении фиксировались для системы защиты в начале ее работы. В [1] были предложены случайно

распределенные по всем узлам группы оповещения, которые периодически обновлялись. Статистический анализ системы показал статистическую эффективность такого подхода. Также, данный подход исключает возможность атаки, описанной в [3].

В качестве формальной основы модели используется модель Распределенных Объектно-Ориентированных Систем (РООСГС) [4], основанная на вероятностном расширении [5] переписывающей логики [6]. В рамках модели РООСГС учитываются основные аспекты рассматриваемой системы защиты. Для спецификации и анализа полученной системы используется подход, описанный в [7]. Данный подход позволяет использовать формальные методы в трех аспектах: для задания модели используется язык спецификации SHYMaude [7], разработанный для спецификации моделей РООСГС, для задания метрик системы используется язык QuaTEx, разработанный для формальной спецификации количественных темпоральных свойств системы [8], для статистического анализа применяется инструмент MultiVeStA [9], использующий метод Монте-Карло для построения доверительных интервалов значений метрик.

В данной статье рассматривается следующая фаза разработки стохастической коллаборационной защиты от вирусов. Предварительно исследованная в [1] система транслируется в приложение Java, которое разворачивается на виртуальных машинах, имеющих общую сеть. Приложение имеет встроенную систему статистического анализа, основанную на методе Монте-Карло. Строится график доли зараженных узлов с течением времени и рассчитываются метрики, аналогичные предложенным в [1].

Целью данного исследования является апробация модели в условиях, приближенных к реальным. В процессе реализации системы выясняется, что есть две возможности реализации механизма передачи сообщений – с помощью протоколов TCP/IP и UDP. Протокол TCP/IP гарантирует доставку сообщений и порядок их получения, в то же время он имеет заметно большие временные задержки. Протокол UDP просто отсылает сообщения и имеет минимальные задержки. Были исследованы оба варианта протокола и оценен вклад от использования протокола UDP. Для оценки максимально возможной выгоды от использования более быстрых протоколов был произведен статистический анализ исходной модели РООСГ, на основе которой осуществлялась реализация, максимально приведенной в соответствие с практической реализацией.

Также было оценено влияние размера групп оповещения и общего количества узлов и рассмотрен вопрос масштабируемости системы.

Реализация коллаборационной стохастической системы защиты от вирусов

В качестве основы для коллаборационной стохастической системы защиты от вирусов используется коллаборационная система защиты от вирусов, предложенная в [2]. Для предотвращения атаки, описанной в [3], вводятся группы оповещения, покрывающие узлы сети случайным образом.

Ввиду того, что предлагаемая модель уже является стохастической, для повышения описательной способности модели, вводятся вероятности в модель вируса, модель обнаружения вируса и алгоритм защиты. Рассматривается вирус, случайно выбирающий узел для заражения через экспоненциально распределенные промежутки времени. Система обнаружения моделируется посредством ложноположительных, когда узел не заражен, но система обнаружения сигнализирует о наличии вируса, и ложноотрицательных, когда узел заражен, но вирус не определяется, вероятностей. Доставка сообщений в системе защиты занимает экспоненциально распределенный промежуток времени.

Алгоритм системы защиты представлен на рис. 1. На рисунке используются следующие обозначения: овал – начало работы, прямоугольники содержат действия, ромбы – условные ветвления, круг обозначает узел с определенным идентификатором id, стрелки без сообщений обозначают простой переход, стрелки с сообщением вида < id <- command > обозначают, что стрелка активируется при получении объектом данного сообщения. Система начинает работу с выполнения действий, исходящих из овала «Старт» и далее реагирует на получаемые сообщения. Запланированные сообщения обозначаются [t, msg], где

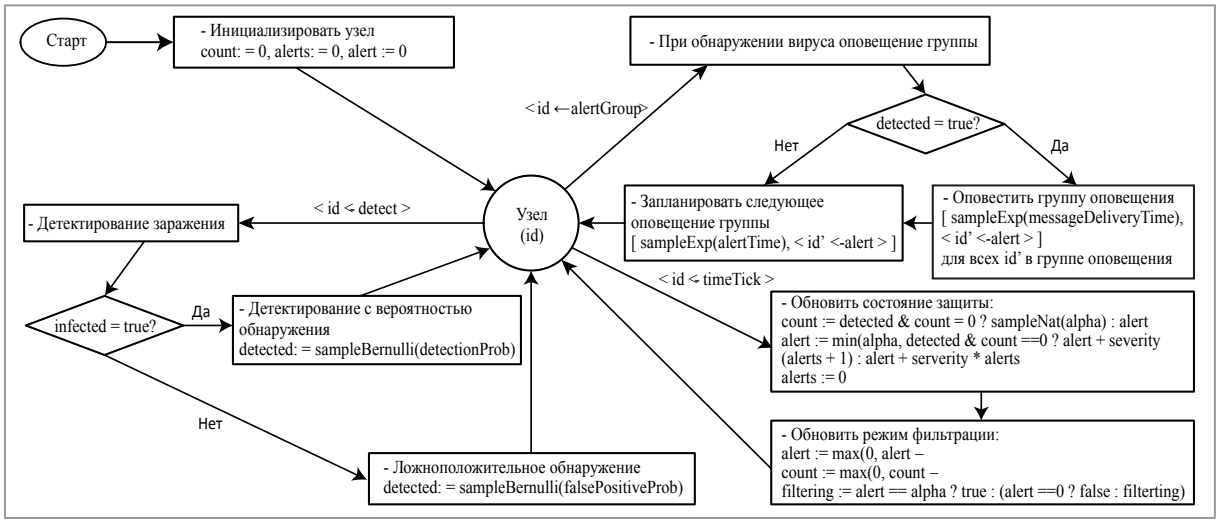

Рис. 1 Коллаборационная стохастическая система защиты от вирусов

t – запланированное время активации сообщения msg. Детальное описание системы и ее динамики может быть найдено в [1].

Система защиты производит детекцию наличия вируса через экспоненциально распределенные промежутки времени. Результат детекции зависит от наличия реального заражения и корректируется ложноположительными (вирус отсутствует, но система сигнализирует наличие) и ложноотрицательными (вирус присутствует, но система его не детектирует) вероятностями. При положительном результате детекции вируса, система рассылает сообщения об обнаружении заражения группе оповещения. При получении такого оповещения, узел увеличивает внутренний уровень тревожности. При превышении данным уровнем предустановленного критического значения, система переходит в режим фильтрации. В состоянии включенной системы фильтрации предполагается, что вирус не может заражать другие узлы, а также узел не может быть заражен при попытке его заражения извне. При отсутствии получаемых оповещений, система равномерно уменьшает уровень тревожности с течением времени. При достижении нулевого уровня тревожности, система выходит из режима фильтрации в обычный режим.

Трансляция спецификации SHYMaude в приложение на языке Java

Спецификация на языке SHYMaude [7] может быть легко транслирована в приложение на языке Java. Использование фреймворка Spring еще более упрощает процедуру, предоставляя библиотеки, необходимые для реализации асинхронных объектов и средства коммуникации для них.

Более детально, трансляция производится следующим образом. Каждый класс SHYMaude представляется классом Java. Дискретные и непрерывные переменные класса SHYMaude определяются полями класса Java. Их тип выбирается либо double для непрерывных переменных, либо long, int, boolean, enum в зависимости от контекста для дискретных переменных. Стохастические дифференциальные уравнения (при их наличии), задающие динамику непрерывных переменных, реализуются с помощью разностных схем. Переписывающие правила реализуются в виде соответствующих блоков кода на Java, обновляющих поля, соответствующие переменным SHYMaude в соответствии с правилами перезаписи.

Сообщения могут быть разделены на два класса: сообщения, посылаемые объектом класса самому себе с целью запланировать некоторые действия через промежуток времени и сообщения, посылаемые другим объектам для передачи информации и/или команд на выполнение определенных действий. Сообщения первого типа либо, в простейшем случае, реализуются с помощью подходящего метода используемого фреймворка Java, позволяющего задержать выполнение потока на заданный период времени (в случае одного управляющего сообщения на класс объекта), либо с помощью

некоторого организованного списка запланированных сообщений, каждое из которых вызывает выполнение своего блока кода. Сообщения второго типа отсылаются и принимаются либо с помощью подходящего фреймворка обмена сообщениями внутри узла в случае, если отсылающий и принимающий объект находятся на одном узле, либо с помощью библиотек отсылки-получения сообщений в сети с использованием подходящего сетевого протокола.

На каждом узле выполняется приложение, которое запускает потоки для каждого объекта, определяемого спецификацией. Внутри каждого потока производится ожидание входящих сообщений, как запланированных потоком самому себе, так и от других объектов данной программы и программ, запущенных на других узлах. При получении сообщения, выполняются соответствующие действия, планируются и/или отсылаются новые сообщения.

Трансляция спецификации в язык Java производится естественно в виду того, что и SHYMaude, и Java основываются на объектно-ориентированной модели. Также, наиболее распространенные фреймворки для языка Java предоставляют широкий спектр возможностей по работе с отсылкой-получением сообщений как внутри приложения, так и между приложениями, расположенными на разных узлах сети. Полученная реализация позволяет провести апробацию модели без больших затрат на реализацию системы на более низкоуровневых языках, таких как C/C++. Также, как будет показано далее, на этом этапе возможно статистическое исследование полученной реализации для дальнейшего выявления аспектов, критических для разрабатываемой системы.

Статистический анализ реализации системы защиты

Была проведена апробация системы защиты в условиях, приближенным к реальным. Спецификация была транслирована в приложение Defense [10], реализованное на языке Java с использованием фреймворка Spring. Система узлов была представлена в виде виртуальных машин реализованных на VirtualBox с операционной системой Windows 10, объединенных с помощью внутренней сети VirtualBox.

Каждый узел выполняет приложение Defense, которое реализует модель вируса, модель обнаружения вирусов и систему защиты, описанных ранее. Для проведения статистического анализа приложение Defense имеет web-составляющую, доступную по url, которая предоставляет панель управления приложением. Внешний вид панели управления представлена на рис. 2.

На панели отображается параметр распределения Стьюдента, определяющий вероятность того, что истинное среднее находится в вычисленном доверительном интервале, номер текущего прогона системы, целевое количество прогонов, устанавливаемый перед запуском анализа, текущее время, затраченное на анализ (время симуляции).

Далее, в таблице рассчитываются три метрики, являющиеся средним значением следующих величин: доля зараженных узлов в конце каждого прогона, максимальная скорость распространения вируса на прогоне, время до насыщения вируса. Данные метрики более подробно были рассмотрены в [1].

Под таблицей отображается усредненный график распространения вируса: по оси x представлено время внутри прогона, по оси y – усредненная по прогонам доля зараженных узлов на данный момент времени.

С помощью панели управления можно запустить как одиночный прогон системы, так и запустить статистический анализ. Статистический анализ проводится методом Монте-Карло. Строиться усредненный график доли зараженных узлов по времени эволюции для визуального анализа. Также рассчитываются три параметра: доля зараженных узлов при достижении насыщения вирусом, максимальная скорость распространения вируса и время до насыщения вируса. Момент насыщения вируса определяется как момент времени, до которого в течение предзаданного интервала не происходит дальнейшего роста количества зараженных узлов.

Первая версия системы в качестве протокола передачи сообщений использовала TCP/IP. В данном протоколе устанавливается соединение, затем передается сообщение. Метод требует подтверждения от принимающего узла о том, что сообщение было получено корректно. Данные особенности метода приводят

к тому, что передача сообщений занимает некоторое, пусть и небольшое, время. В первой версии системы данный метод использовался как для попытки заражения, так и для рассылки оповещений и назначения групп оповещения. Результаты статистического анализа представлены на рис. 2.

Во второй версии системы для передачи оповещений и назначения групп оповещения был использован протокол UDP. Особенностью протокола UDP является то, что он использует простую модель передачи данных, без «рукопожатий» и подтверждений, что делает его значительно более быстрым. Целью исследования данного варианта системы было оценить на практике влияние выбора протокола на эффективность. Для попыток заражения по-прежнему использовался протокол TCP/IP, как наиболее широко распространенный протокол, используемый для взаимодействия в сетях. Результаты статистического анализа представлены на рис. 3.

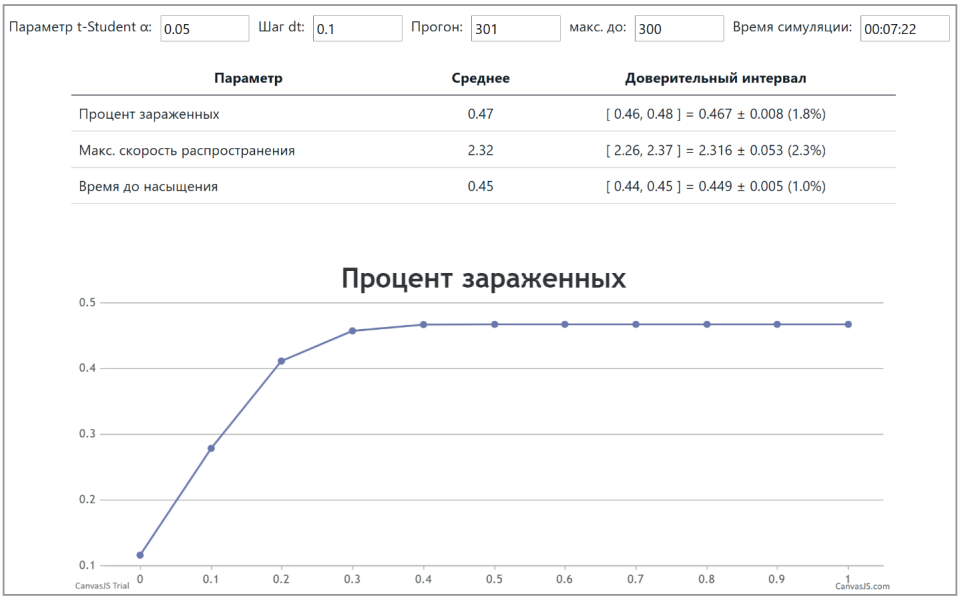

Рис. 2 Результаты статистического анализа реализации на Java стохастической коллаборационной защиты от вирусов с использованием только протокола TCP/IP

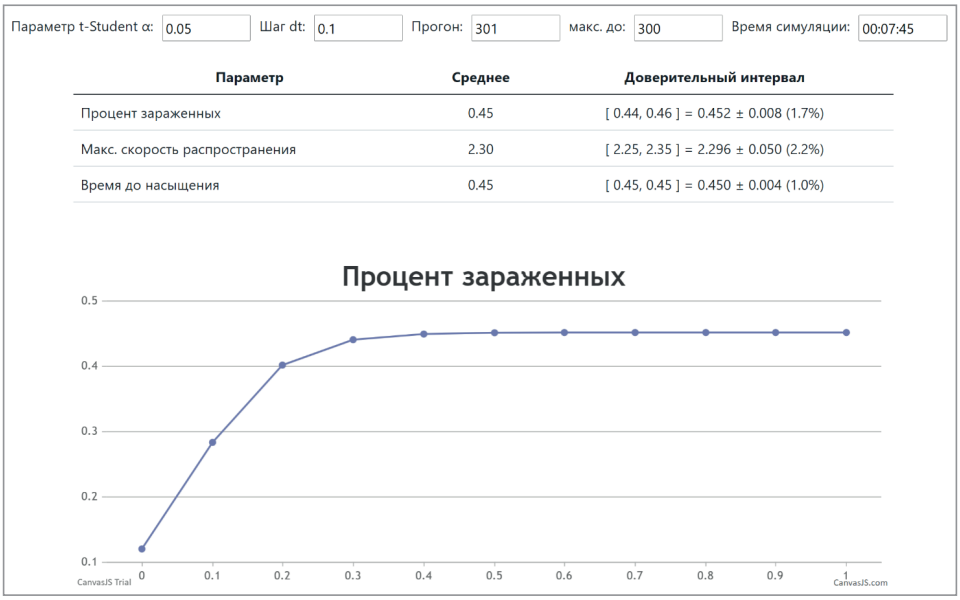

Рис. 3 Результаты статистического анализа реализации на Java стохастической коллаборационной защиты от вирусов с использованием протокола UDP для оповещений и назначения групп оповещения

Как видно из рис. 2 и 3, использование более быстрого протокола дает эффект за счет более быстрого информирования групп оповещения.

Статистический анализ модели с учетом задержек протокола TCP/IP

В [1] проводилось исследование математической модели данной системы защиты. Для получения более детальной оценки вклада задержек, связанных с использованием протокола TCP/IP, в спецификацию модели из [1], максимально приведенную в соответствие с реализацией, была введена задержка, экспоненциально распределенная, с математическим ожиданием, соответствующим значению, измеренному на прогонах реализации Java. Далее, был проведен статистический анализ системы с задержкой, связанной с использованием протокола TCP/IP, и системы с минимальной задержкой в 1мс, так как оценить точно задержку, связанную с использованием протокола UDP, представляется затруднительным в виду отсутствия подтверждений о получении сообщений в данном протоколе. Результат анализа двух вариаций модели представлен на рис. 4.

Как видно из графика, протокол с минимальной задержкой оповещений (условно помеченный как протокол UDP*), показывает несколько лучший результат, чем протокол, полностью основанный на TCP/IP.

Также для уточнения влияния размера групп были проведены дополнительные исследования математической модели. Для случая 10 узлов был проведен анализ доли зараженных узлов в конце симуляции для групп размером от 1 (лишь один узел оповещается) до 10 (оповещаются все узлы). Результаты исследования представлены на рис. 5.

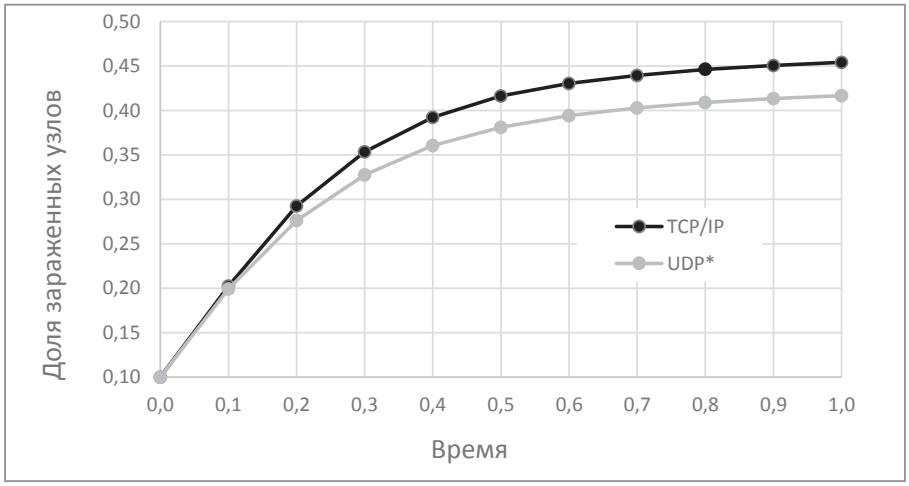

Рис. 4 Доля зараженных узлов по времени симуляции, полученный при анализе модели

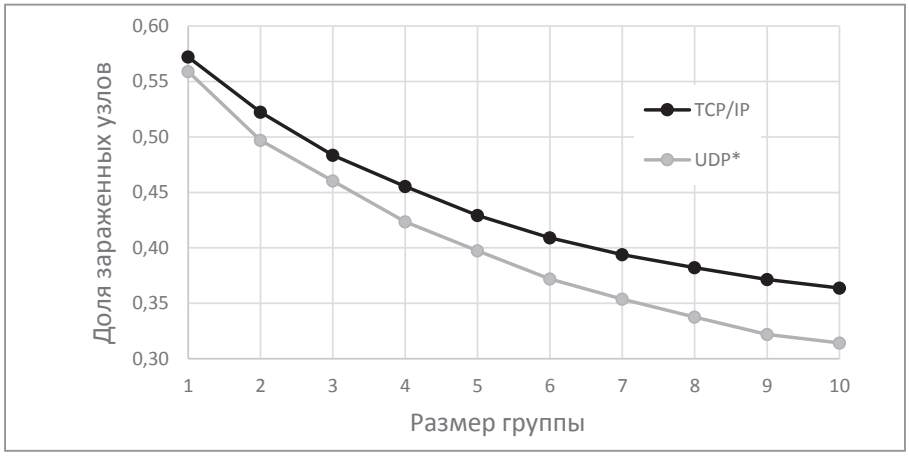

Рис. 5 Доля зараженных узлов в конце симуляции от размера групп, полученный при анализе модели

Как видно из графика, наблюдается линейный рост эффективности при увеличении размера группы. Реально выбираемый размер группы определяется максимально допустимой нагрузкой на сеть, вследствие увеличения количества сообщений, пересылаемых системой защиты при росте размера группы. Количество таких сообщений пропорционально количеству узлов, умноженному на размер группы.

Также, представляет интерес изучение эффективности системы в ситуации, когда размер группы является фиксированным фактором общего количества узлов, который назовем коллабарационным фактором k_G. Тогда размер группы $G = k_G * N$, где N – общее количество узлов. Мы исследовали эффективность для оригинального $k_G = 0.4$, то есть $G = 0.4 * N$, использованный в [2]. При вариации общего количества узлов от 5 до 50 с шагом 5 был получен график зависимости доли зараженных узлов, представленный на рис. 6.

Как видно из графика, увеличение общего количества узлов повышает эффективность модели. Это объясняется тем, что, после полноценной реакции, система «изолирует» зараженные узлы в некотором объеме, после чего заражение прекращается.

Как было отмечено ранее, увеличение количества узлов может повышает количество сообщений, передаваемых системой защиты. Когда размер группы определяется фиксированным фактором, то общее количество передаваемых сообщений в единицу времени на узел может быть оценено сверху фактором пропорциональным m_{num} k_G * N. Верхний предел линейно растет по количеству узлов. В системе с несколькими тысячами узлами, в некоторых сценариях заражения, может наблюдаться нагрузка на сеть, образованная оповещениями.

Данная нагрузка может быть устранена путем применения иерархического подхода к построению системы. Для преодоления роста количества узлов предлагается объединение узлов в кластеры. Внутри каждого кластера реализуется указанная система защиты. Один из узлов объявляется ведущим и отвечает за кластер. Возможны разные подходы к определению, что считать зараженным кластером. Можно положить, что кластер считается зараженным, если хотя бы один узел в кластере заражен. Коммуникация происходит между ведущими узлами, которые действуют от имени кластеров. Далее, рассмотрим кластеры более высоких уровней, состоящие из кластеров более низких уровней. Выберем ведущие узлы на кластерах более высоких уровней и реализуем аналогичную систему на каждом уровне.

Верхнюю оценку общего количества сообщений в системе в единицу времени на один узел можно подсчитать следующим образом. Пусть N_i – количество кластеров уровня i (N_0 обозначает количество узлов на нулевом уровне), k_G – групповой коэффициент, n – количество уровней. Тогда количество сообщений в системе в единицу времени на один узел будет пропорционально. Положим, что мы следуем схеме, в которой количество кластеров на всех уровнях одинаково и равно или меньше

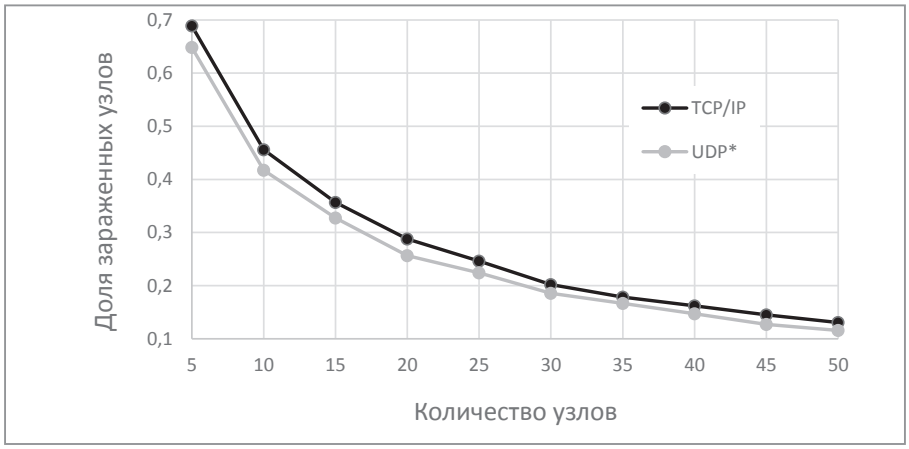

Рис. 6 Доля зараженных узлов в конце симуляции от общего количества узлов при фиксированном факторе размера группы, полученный при анализе модели

количества узлов на нулевом уровне N_i N_0. Тогда $m_{num} \leq k_G * n * N_0$, что значительно меньше аналогичной оценки для дизайна системы без использования иерархического подхода. Например, при $N = 2700$, $k_G = 0.4$ получаем размер группы в 1080 узлов и, как следствие, оценка будет равна 1080 для неирархической системы. Однако при использовании описанного иерархического подхода с тремя уровнями, то есть $n = 3$, $N_0 = 30$, получаем оценку $m_{num} \leq 0.4 * 3 * 30 = 36$, что означает значительно меньшую нагрузку на сеть.

Заключение

В данной работе была описана апробация реализации коллаборационной стохастической системы защиты от вирусов в условиях, приближенных к реальным.

Было изучено две вариации алгоритма, с использованием протоколов TCP/IP и UDP в части рассылки оповещений об обнаружении вирусов и назначении групп оповещений. Было установлено, что применение протокола UDP приводит к увеличению статистической эффективности системы защиты. Был исследован возможный потенциал увеличения эффективности за счет снижения задержек в доставке сообщений.

Было исследовано влияние размера группы оповещения и общего количества узлов, при сохранении в процентном соотношении размера групп. Было выяснено, что увеличение размера общего количество узлов приводит к заметному росту эффективности системы, однако для некоторых сценариев заражения, сопряжено с увеличением нагрузки на сеть ввиду одновременного увеличения количества оповещений в сети. Был предложен механизм иерархической организации системы защиты для предотвращения быстрого роста данной нагрузки.

Таким образом, предлагаемая система может использоваться для реальных задач защиты сетей, состоящих из произвольного количества узлов.

ЛИТЕРАТУРА

1. **Шарыкин, Р.Е.** Применение Формальных Методов при Проектировании Коллаборационной Системы Противовирусной Защиты / Р.Е. Шарыкин, А.Н. Курбацкий // Журнал Белорусского государственного университета. Математика. Информатика.– 2020.– № 1.– С. 59–69.
2. **Briesmeister, L.** Microscopic simulation of a group defense strategy / L. Briesmeister, P. Porras // Proceedings of Workshop of Principles of Advanced and Distributed Simulation, Monterey, California, US, 1–3 June, 2005 /; eds.: D. Nicol [et al].– Los Alamitos, California, US: IEEE Computer Society, 2005.– P. 254–261.
3. **Briesmeister, L.** Automatically deducing propagation sequences that circumvent a collaborative worm defense / L. Briesmeister, P. Porras // Proceedings of International Performance Computing and Communications Conference, Phoenix, Arizona, US, 10–12 April, 2006 /.– Los Alamitos, California, US: IEEE Computer Society.– P. 587–592.
4. **Шарыкин, Р.Е.** Модель распределенных объектно-ориентированных стохастических гибридных систем / Р.Е. Шарыкин, А.Н. Курбацкий // Журнал Белорусского государственного университета. Математика. Информатика.– 2019, № 2.– С. 52–61.
5. **Agha, G.A.** PMaude: Rewrite-based specification language for probabilistic object systems / G.A. Agha, J. Meseguer, K. Sen // Electronic Notes in Theoretical Computer Science.– 2006.– Vol. 153, iss. 2, № 2.– P. 213–239.
6. **Meseguer, J.** Conditional rewriting logic as a unified model of concurrency / J. Meseguer // Theoretical Computer Science.– 1992.– Vol. 96, iss. 1.– P. 73–155.
7. **Шарыкин, Р.Е.** Верификация Распределенных Объектно-Ориентированных Стохастических Гибридных Систем / Р.Е. Шарыкин, А.Н. Курбацкий // Вестник Гродненского Государственного Университета имени Янки Купалы. Серия 2. Математика. Физика. Информатика, вычислительная техника и управление.– 2019.– Том 9, № 3.– С. 123–132.
8. **Sen, K.** On statistical model checking of stochastic systems / K. Sen, M. Viswanathan, G. Agha // Lecture Notes in Computer Science.– 2005.– Vol. 3576.– P. 266–280.
9. **Sebastio, S.** MultiVeStA: Statistical model checking for discrete event simulators / S. Sebastio, A. Vandin // Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 December, 2013 / Brussels, Belgium: Institute for Computer Sciences; eds.: A. Horvath [et al].– 2013.– P. 310–315.
10. **Шарыкин, Р.Е.** Реализация в среде Java коллаборационной системы защиты от вирусных атак. [Электронный ресурс] // GitHub: [сайт].– Режим доступа: https://github.com/shymaude.– Дата доступа: 02.11.2021.

REFERENCES

1. **Sharykin, R.** Application of Formal Methods in the Design of a Collaborative Virus Defense System / R. Sharykin // Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika.– 2020.– Vol. 1.– P. 59–69.

2. **Briesmeister, L.** Microscopic simulation of a group defense strategy / L. Briesmeister, P. Porras // Proceedings of Workshop of Principles of Advanced and Distributed Simulation, Monterey, California, US, 1–3 June, 2005 /; eds.: D. Nicol [et al].– Los Alamitos, California, US: IEEE Computer Society, 2005.– P. 254–261.
3. **Briesmeister, L.** Automatically deducing propagation sequences that circumvent a collaborative worm defense / L. Briesmeister, P. Porras // Proceedings of International Performance Computing and Communications Conference, Phoenix, Arizona, US, 10–12 April, 2006 /.– Los Alamitos, California, US: IEEE Computer Society.– P. 587–592.
4. **Sharykin, R.** A model of Distributed Object-Based Stochastic Hybrid Systems / R. Sharykin, A. Kourbatski // Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika.– 2019.– № 2.– P. 52–61.
5. **Agha, G.A.** PMaude: Rewrite-based specification language for probabilistic object systems / G.A. Agha, J. Meseguer, K. Sen // Electronic Notes in Theoretical Computer Science.– 2006.– Vol. 153, iss. 2, № 2.– P. 213–239.
6. **Meseguer, J.** Conditional rewriting logic as a unified model of concurrency / J. Meseguer // Theoretical Computer Science.– 1992.– Vol. 96, iss. 1.– P. 73–155.
7. **Sharykin R.** Verification of Distributed Object-Oriented Stochastic Hybrid Systems. Systems / R. Sharykin, A. Kourbatski // Vestnik Grodnenskogo Gosudarstvennogo Universiteta imeni Yanki Kupaly. Seriya 2. Matematika. Fizika. Informatika, vychislitel'naya tekhnika i upravlenie.– 2019.– Vol. 9, № 3.– P. 123–132.
8. **Sen, K.** On statistical model checking of stochastic systems / K. Sen, M. Viswanathan, G. Agha // Lecture Notes in Computer Science.– 2005.– Vol. 3576.– P. 266–280.
9. **Sebastio, S.** MultiVeStA: Statistical model checking for discrete event simulators / S. Sebastio, A. Vandin // Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 December, 2013 / Brussels, Belgium: Institute for Computer Sciences; eds.: A. Horvath [et al].– 2013.– P. 310–315.
10. **Sharykin, R.** Java implementation of the stochastic collaborative virus defense system [Electronic resource] // GitHub: [site].– Mode of access: https://github.com/shymaude.– Date of access: 02.11.2021.

Поступила	*После доработки*	*Принята к печати*
02.11.2021	*30.11.2021*	*01.12.2021*

R. SHARYKIN

APPROBATION OF THE STOCHASTIC GROUP VIRUS PROTECTION MODEL

The article discusses the implementation in Java of the stochastic collaborative virus defense model developed within the framework of the Distributed Object-Based Stochastic Hybrid Systems (DOBSHS) model and its analysis. The goal of the work is to test the model in conditions close to the real world on the way to introducing its use in the practical environment. We propose a method of translating a system specification in the SHYMaude language, intended for the specification and analysis of DOBSHS models in the rewriting logic framework, into the corresponding Java implementation. The resulting Java system is deployed on virtual machines, the virus and the group virus alert system are modeled stochastically. To analyze the system we use several metrics, such as the saturation time of the virus propagation, the proportion of infected nodes upon reaching saturation and the maximal virus propagation speed. We use Monte Carlo method with the computation of confidence intervals to obtain estimates of the selected metrics. We perform analysis on the basis of the sigmoid virus propagation graph over time in the presence of the defense system. We implemented two versions of the system using two protocols for transmitting messages between nodes, TCP/IP and UDP. We measured the influence of the protocol type and the associated costs on the defense system effectiveness. To assess the potential of cost reduction associated with the use of different message transmission protocols, we performed analysis of the original DOBSHS model modified to model message transmission delays. We measured the influence of other model parameters important for the next steps towards the practical use of the model. To address the system scalability, we propose a hierarchical approach to the system design to make possible its use with a large number of nodes.

Шарыкин Роман Евгеньевич, соискатель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета. Научные интересы: моделирование и анализ сложных систем, информационная безопасность, переписывающая логика.

Sharykin Raman Yauhenavich, aspirant of the Department of Software Engineering, Faculty of Applied Mathematics and Computer Science, Belarusian State University. Scientific interests: modeling and analysis of complex systems, information security, rewriting logic.

Email: sharykin@gmail.com