УДК 621.39, 004.7

Д.А. ЗАЙЦЕВ, д-р техн. наук, Т.Р. ШМЕЛЁВА, канд. техн. наук,
В. РЕТЧИТЗЕГГЕР, д-р философии, Б. ПРОЛЛ, д-р философии

ОЦЕНКА ВЛИЯНИЯ ЗЛОНАМЕРЕННОГО ТРАФИКА НА ФУНКЦИОНИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ РЕШЕТОК

Облачные вычисления и вычисления на решетках [1, 2] представляют собой альтернативу концепции централизованного суперкомпьютера и обладают рядом преимуществ, среди которых следует отметить: высокую надежность, так как сбой или выход из строя узла незначительно влияет на работоспособность всей решетки; возможность использования простаивающих вычислительных ресурсов всего мира; доступность информации, приложений и процессоров.

Широкое распространение вычислений на решетках и облаках привносит вместе со значительными преимуществами и некоторые недостатки, обусловленные уязвимостью быстрорастущей структуры. Так, в [3, 4] была обнаружена возможность блокирования решетки посредством злонамеренного трафика. В указанных работах были применены классические сети Петри, поэтому, такие числовые характеристики как вероятность блокирования, процент падения производительности решетки и его влияние на качество обслуживания не были исследованы.

Моделирование систем [5], в частности сетей [6], раскрашенными сетями Петри (CPN) позволяет выполнить оценку производительности и качества обслуживания. Раскрашенные сети Петри содержат средства для модульной композиции модели, описания временных характеристик и статистической обработки результатов имитационного моделирования. Раскрашенные сети Петри моделирующей системы CPN Tools были успешно применены для исследования таких современных сетевых технологий как Ethernet, MPLS (Multiprotocol Label Switching), PBB (Provider Backbone Bridge), что отражено в библиотеке реальных примеров, размещенных на сайте системы (http://cpntools.org).

Настоящая работа представляет собой дальнейшее развитие моделей решеток в форме сетей Петри [3] и преобразование их в класс раскрашенных сетей Петри с целью числовой оценки последствий [7], которые может нанести злонамеренный трафик, представляющий собой угрозу работоспособности решетки. В качестве перспективного направления исследований планируются контрмеры для противодействия этой угрозе.

Композиция моделей решеток

Модель решетки собирается как квадратная (прямоугольная) матрица коммуникационных узлов (DCE), снабженная терминальным оборудованием (DTE), присоединенным к границам. Каждая DCE имеет четыре порта, расположенные на сторонах квадрата единичного размера и работает в полнодуплексном режиме, обеспечивая два канала для независимой передачи и приема пакетов. DCE осуществляет коммутацию пакетов между портами, основанную на принципе «запомнить и отправить» (store-and-forward). DTE производит и потребляет пакеты, для моделирования либо рабочей нагрузки решетки, либо специального (злонамеренного) трафика. Типовая модель DCE названа *n* (сокращение «node» – узел). Типовые модели DTE названы *l, r, u, b* (сокращения имен границ «left» – левый, «right» – правый, «upper» – верхний и «bottom» – нижний соответственно); они имеют незначительные отличия, как это будет описано далее.

Мы используем систему адресации узлов, в рамках прямоугольной решетки размера $k1 \times k2$ посредством двух целых чисел (*i, j*) в соответствии с рис. 1, *а*, где первое число обозначает строку, а второе – столбец. DCE пронумерованы от *1* до *k1* в вертикальном направлении, и от *1* до *k2* в горизонтальном направлении, в то время как DTE имеют один из индексов равный *0* или *k1+1 (k2+1*).

Описание адреса узла имеет следующую форму: *colset an = product INT * INT;*

Модель решетки собирается из моделей узлов; имя узла содержит суффикс, равный его адресу; каждый узел снабжается позицией с суффиксом «a», содержащим его адрес. Тип узла определяется тегом подстановки перехода, записанным в маленьком прямоугольнике: *n, l, r, u, b*.

Пакет передается между парой DTE и имеет следующее описание:

*colset pkt = record da:an * sa:an * co:STRING * ts:INT timed;*

где *da* – адрес назначения, *sa* – адрес отправителя, строка *co* представляет собой некоторое содержимое пакета, поле *ts* добавлено, для того чтобы запомнить временной штамп, когда пакет был отправлен в решетку, для последующего вычисления времени доставки пакета.

Для вычисления размеров буфера, используются элементарные фишки формы

colset cc = unit with c;

Каждый канал описан парой позиций: первая типа *pkt*, для размещения пакета и вторая типа *cc*, для указания размера буфера. Размеры буфера измеряются в количестве пакетов; буферы портов имеют размер, равный единице. Каждый порт состоит из двух каналов: входной с суффиксом «i», для получения пакетов и выходной с суффиксом «o», для передачи пакетов. Таким образом, порт представлен четверкой указанных контактных позиций для соединения с соседним узлом. Нумерация портов выполнена по часовой стрелке, начиная с верхнего порта, номер которого равен единице.

Для того чтобы собрать модель решетки, соединение узлов осуществляется через слияние соответственных контактных позиций соседних устройств, как показано на рис. 1, *а*. В горизонтальном направлении порт 2 левого узла совмещается с портом 4 правого узла; в вертикальном направлении порт 3 верхнего узла совмещается с портом 1 нижнего узла. Для того чтобы избежать двойственности в именах портов, рассматриваются только имена левого (номер 4) и верхнего (номер 1) портов текущего узла. Правый (номер 2) и нижний (номер 3) имена портов не появляются в модели; вместо них используются имена портов левого (номер 4) и верхнего (номер 1) соседних узлов. Поэтому, суффикс «i/o» соответствует входному/выходному каналу либо левого (номер 4), либо верхнего (номер 1) портов одного из соединенных узлов.

Модель узла DCE

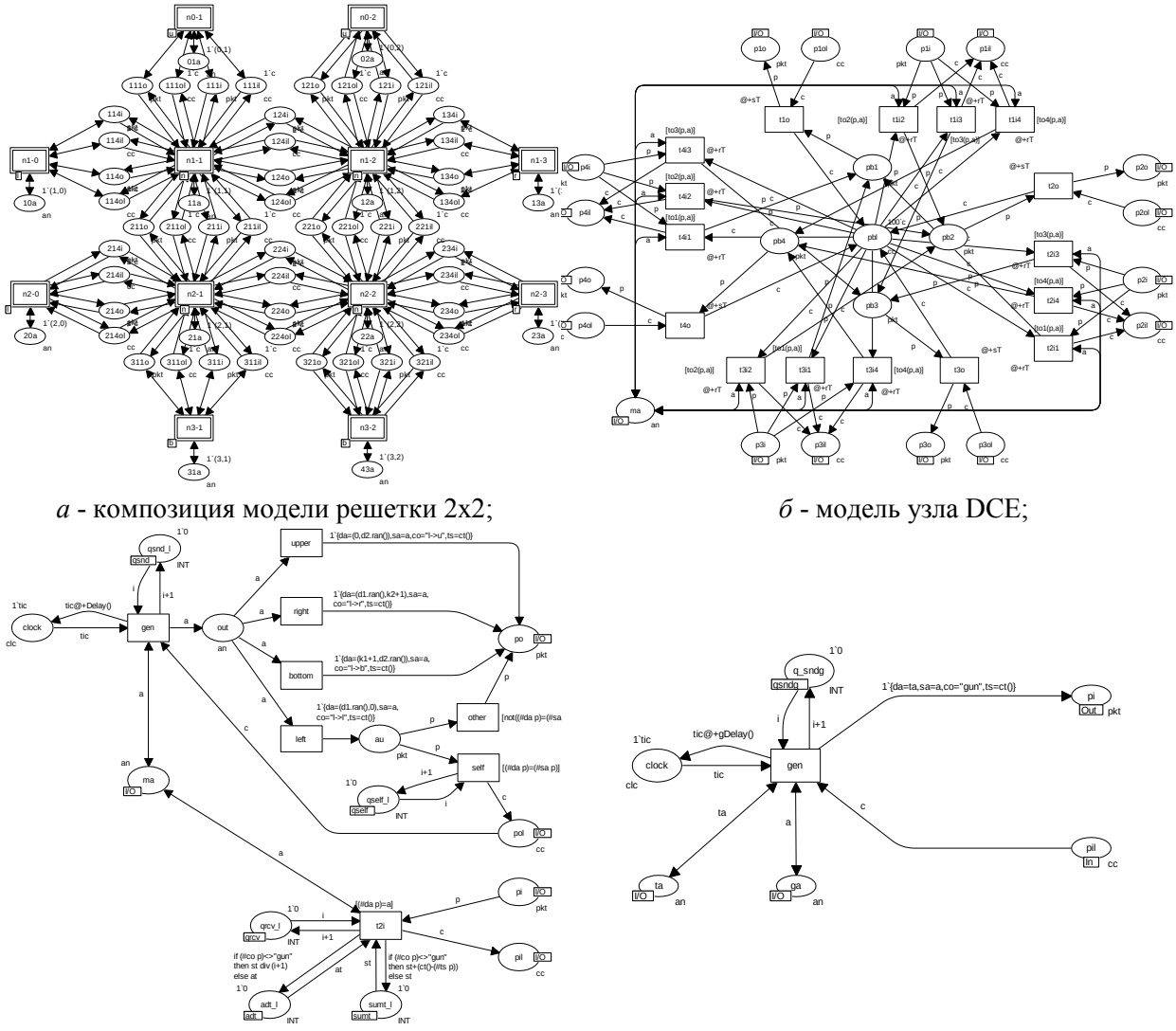
Модель узла DCE представлена на рис. 1, *б*. Она состоит из *4x4=16* описанных выше контактных позиций портов, расположенных на сторонах квадрата: *p1o, p1ol, p1i, p1il, p2o, p2ol, p2i, p2il, p3i, p3il, p3o, p3ol, p4i, p4il, p4o, p4ol*. Порядок позиций на рис. 1, *б* обеспечивает соединение каналов противоположных типов в композиции решетки: входной канал соединяется с выходным каналом соседнего узла и наоборот. Порт 2 соединяется с портом 4 правого соседнего узла, и порт 3 соединяется с портом 1 нижнего соседнего узла. Контактная позиция *ma* содержит адрес текущего узла DCE.

Внутренний буфер узла представлен пятью следующими позициями: позиции *pb1, pb2, pb3, pb4* содержат пакеты, перенаправленные в соответствующий порт, в то время как позиция *pbl* представляет собой доступный размер буфера. Кроме того, позиции *pb1, pb2, pb3, pb4* являются дополнительными по отношению к позиции *pbl*, что означает следующее: запоминание пакета в одной из позиций *pb1, pb2, pb3, pb4* изымает фишку из *pbl,* и извлечение пакета из одной из позиций *pb1, pb2, pb3, pb4* размещает фишку в *pbl*.

Выходной канал порта моделируется одиночным переходом – для четырех портов *t1o, t2o, t3o, t4o* соответственно. Например, для порта 1 переход *t1o* извлекает пакет из позиции *pb1* и помещает пакет в позицию *p1o*; кроме того *t1o* извлекает фишку из *p1ol* и помещает

фишку в *pbl*, моделируя изменения размера буферов. Условие запуска переходов *t1o* включает присутствие пакета в соответствующей секции внутреннего буфера *pb1* и доступность выходного буфера порта назначения – присутствие фишки в позиции *p1ol*.

Входной канал порта моделируется посредством трех переходов – по переходу для каждого возможного направления передачи, исключая порт прибытия пакета; таким образом, моделируется решение о перенаправление пакета. Например, для порта 1, переход *t1i2* моделирует перенаправление в порт 2, *t1i3* – в порт 3, *t1i4* – в порт 4. Конфигурация инцидентных дуг перехода отличается только именем секции внутреннего буфера. Например, переход *t1i2* извлекает пакет из *p1i* и помещает пакет в *pb2*; кроме того, *t1i2* извлекает фишку из *pbl* и помещает фишку в *p1il*, моделируя изменения в длинах буферов. Условия запуска перехода *t1i2* включает присутствие пакета во входном буфере *p1*i порта 1 и доступность внутреннего буфера – присутствие фишки в позиции *pbl*.



а - композиция модели решетки 2x2; *б* - модель узла DCE;

в - модель узла DTE; *г* - модель пушки трафика.

Рис. 1. Модель решетки

Случайная коммутация пакетов является простейшим маршрутным решением, но она не обеспечивает корректную доставку пакетов в соответствии с адресом назначения. Внутри модели решение о коммутации пакетов осуществляется посредством четырех предикатов переадресации *to1(p, a), to2(p, a), to3(p, a), to4(p, a)*. Они используются как функции запуска переходов, перенаправляющих пакет в соответствующий порт. Несколько предикатов могут быть истинными для заданного пакета; в этом случае выбор направления выполняется слу-

чайно между соответствующими переходами. Выбор может быть детерминированным, например, когда выбирается направление с большей разницей адресов, но случайный выбор работает лучше.

Предикаты переадресации используют следующие вспомогательные предикаты, где константы *k1* и *k2* определяют размер прямоугольной решетки в вертикальном и горизонтальном направлениях соответственно.

Разрешенное направление передачи:

fun v1(p:pkt,a:an)=((#1(#da p))<(#1 a)); fun v3(p:pkt,a:an)=((#1(#da p))>(#1 a));
fun v4(p:pkt,a:an)=((#2(#da p))<(#2 a)); fun v2(p:pkt,a:an)=((#2(#da p))>(#2 a));

Принадлежность адреса назначения к соответствующей границе:

fun db4(p:pkt)=((#2(#da p))=0); fun db2(p:pkt)=((#2(#da p))=(k2+1));
fun db1(p:pkt)=((#1(#da p))=0); fun db3(p:pkt)=((#1(#da p))=(k1+1));

Принадлежность текущего DCE к соответствующей границе:

fun cb4(a:an)=((#2 a)=1); fun cb2(a:an)=((#2 a)=k2);
fun cb1(a:an)=((#1 a)=1); fun cb3(a:an)=((#1 a)=k1);

Назначение пакета к соответствующему соседнему узлу:

fun nb4(p:pkt,a:an)=((#2 a)=(#2 (#da p)+1)) andalso ((#1 a)=(#1 (#da p)));
fun nb2(p:pkt,a:an)=((#2 a)=(#2 (#da p)-1)) andalso ((#1 a)=(#1 (#da p)));
fun nb1(p:pkt,a:an)=((#1 a)=(#1 (#da p)+1)) andalso ((#2 a)=(#2 (#da p)));
fun nb3(p:pkt,a:an)=((#1 a)=(#1 (#da p)-1)) andalso ((#2 a)=(#2 (#da p)));

Предикаты перенаправления:

fun to1(p:pkt,a:an)=v1(p,a) andalso
((not (db1(p) andalso cb1(a))) orelse (db1(p) andalso cb1(a) andalso nb1(p,a)));
fun to3(p:pkt,a:an)=v3(p,a) andalso
((not (db3(p) andalso cb3(a))) orelse (db3(p) andalso cb3(a) andalso nb3(p,a)));
fun to4(p:pkt,a:an)=v4(p,a) andalso
((not (db4(p) andalso cb4(a))) orelse (db4(p) andalso cb4(a) andalso nb4(p,a)));
fun to2(p:pkt,a:an)=v2(p,a) andalso
((((not (db2(p) andalso cb2(a))) orelse (db2(p) andalso cb2(a) andalso nb2(p, a)));

Описанные выше предикаты представляют собой результат анализа поведения модели и отражают определенный баланс простоты с использованием только локальной информации и обеспечения достаточно хорошей производительности решетки. Они сами по себе могут быть изучены наряду с другими факторами, которые влияют на производительность и качество обслуживания решетки.

Алгоритм коммутации базируется на разрешенном направлении передачи информации, определенном разницей текущего адреса и адреса назначения по горизонтальной и вертикальной оси координат; разрешенное направление задано предикатами *v*. Это основное правило с исключениями для границ коммуникационной решетки. Когда DTE назначения и текущий DCE расположены на той же самой границе ($db \wedge cb$), пакет не должен быть доставлен к ошибочному DTE. Применение основного правила разрешает передачу в двух разрешенных направлениях, но DTE не обеспечивает перенаправление пакетов; таким образом, перенаправление к DTE отличному от DTE назначения должно быть запрещено. Специальный случай ($db \wedge cb \wedge nb$) представляет направление к соседнему DTE назначения, который разрешен. Соответствующая булева функция может быть минимизирована как $\overline{(db \wedge cb)} \vee db \wedge cb \wedge nb = \overline{db} \vee \overline{cb} \vee nb$.

Временные характеристики модели заданы двумя параметрами *sT* и *rT*, которые представляют собой временную задержку отправки и получения пакета соответственно. Напомним, что в соответствии с концепцией времени в CPN Tools, переход срабатывает мгновенно, но соответствующая временная задержка применяется к его фишкам, которые проводят время в выходных позициях, оставаясь в недоступном состоянии.

Модель узла DTE

Простейшие функции узла DTE – производить и потреблять пакеты; вычислительные аспекты работы DTE также могут быть промоделированы, но в настоящей работе они не рассматриваются. Модель DTE узла левой границы представлена на рис. 1,*в*; модель подразделяется на каналы передачи и приема.

Передающий канал моделируется переходом *t2i,* который поглощает полученные пакеты и подсчитывает их общее количество для всей модели в позиции *qrcv_l* объединенного множества позиций *qrcv*. Для более точных оценок отправленные и полученные пакеты могут быть подсчитаны отдельно для каждой пары из $(2\cdot k1+2\cdot k2)^2-(2\cdot k1+2\cdot k2)$ взаимодействующих устройств. Кроме того, переход *t2i* выполняет вычисление такой характеристики качества обслуживания как среднее время доставки пакету в позиции *adt_l* объединенного множества позиций *adt*; с этой целью общая сумма индивидуальных времен доставки всех пакетов аккумулируется в позиции *sumt_l* объединенного множества позиций *sumt*. Для вычисления времени доставки пакета использовано выражение *ct()-(#ts p)* в надписи дуги, которая соединяют переход *t2i* с позицией *sumt_l*. Функция *ct()* возвращает текущее модельное время; ее описание имеет следующий вид

fun ct()=IntInf.toInt(time());

Отметим, что условный оператор *if* с условием *(#co p)<>"gun"* обеспечивает расчет времени доставки только для пакетов рабочей нагрузки решетки.

Передающий канал состоит из таймера представленного переходом *gen*, периодичность запуска которого контролирует позиция *clock*; ее единственная фишка типа *tic* задерживается случайной функцией *Delay(),* которая определяет период запуска. Общее количество отправленных пакетов для всей модели подсчитывает позиция *qsnd_l* объединенного множества позиций *qsnd*. В результате фишка, которая совпадает с адресом текущего узла *a,* попадает в позицию *out* и представляет собой заготовку для сборки пакета.

Один из альтернативных переходов *left, upper, right, bottom*, который выбирается случайно, срабатывает и генерирует пакет, направленный к соответствующей границе; его адрес назначения избирается случайно в диапазоне адресов границ через стандартную функцию *ran()*; вспомогательные типы *d1* и *d2* имеют следующее описание

colset d1=int with 1..k1;

colset d2=int with 1..k2;

Единственная разница четырех типов узлов границ *l, u, r, b* состоит в фильтрации своего собственного адреса назначения, эту операцию выполняет один из соответствующих переходов *left, upper, right, bottom*. На рис. 1в представлена модель левого узла, поэтому фильтрация выполнена после перехода *left,* который размещает сгенерированный пакет в промежуточную позицию *au*. Переход *self* извлекает и удаляет пакеты, которые направлены к текущему устройству и подсчитывает их общее количество для всей модели в позиции *qself_l* объединенного множества позиций *qself,* в то время как переход *other* выводит другие пакеты в буфер выходного канала устройства. Когда пакет размещается в выходной порт, временной штамп сохраняется в поле *ts* посредством выражения *ts=ct().*

Что касается функций распределения случайных величин, то для случайного выбора альтернативных переходов и выполнения стандартной функции *ran()* CPN Tools использует равномерный закон распределения. Кроме того CPN Tools предлагает широкий диапазон известных законов распределения, для описания пользовательских случайных функций. Например, распределение Пуассона выбрано для функции *Delay(),* соответствующее описание имеет форму *fun Delay()=poisson(10.0);*

Моделирование рабочей нагрузки решетки

Моделирование рабочей нагрузки решетки, которую создают модели терминальных устройств *l, u, r, b,* позволяет отладить модель в целом и оценить влияние ее параметров на

производительность и качество обслуживания решетки, а также изучить поведение решетки в условиям пиковой нагрузки. Использована модель решетки размером 8x8.

Были изучены следующие базовые параметры модели, которые влияют на поведение решетки:

– размер внутреннего буфера *bs* узла DCE;

– интенсивность рабочей нагрузки – параметр распределения Пуассона *wl* модели DTE;

– производительность DCE – временные задержки *rT* и *sT* получения и передачи пакетов, соответственно.

При малых значениях размера внутреннего буфера *bs*, например равном 10 (пакетов), решетка попадает в тупик даже при трафике, которой равняется 10 % от пропускной способности решетки. Достаточно большой буфер, например с размером 10000 и более, не позволяют наблюдать тупики даже при пиковой нагрузке. В большинстве имитационных экспериментов выбран размер буфера равный 100, что позволило наблюдать как наличие, так и отсутствие тупиков на приемлемых интервалах времени.

Табл. 1 показывает, что решетка приходит в тупик даже при рабочей нагрузке, когда размер буфера равняется 100. В большинстве случаев тупик означает, что буферы некоторых коммуникационных устройств DCE заполнены также как буферы их портов. Разница количества отправленных и полученных пакетов приближается к общему размеру всех буферов решетки *k1*k2*bs+2*k1+2*k1+2*k2*, который равняется 6432 (пакетов) для рабочих параметров табл. 1.

Следовательно, пиковая нагрузка решетки (ее максимальная производительность) при выбранной интенсивности обслуживания *rT*=*sT*=5 примерно равна *gp*=3 пакетов за единицу модельного времени, которое достигается при интенсивности рабочей нагрузки *wl*=10.0. Аббревиатура MTU означает «единица модельного времени» (model time unit) определенное значение которой может быть выбрано в процессе масштабирования временных характеристик реальных решеток по методике, представленной в [6]. Заметим, что за порогом рабочей нагрузки *wl*=9.0 были выполнены более длительные вычислительные эксперименты, что отмечено в таблице, но тупик не был достигнут. При приближении к полному тупику, производительность решетки может падать, несмотря на высокую интенсивность трафика из-за частичных промежуточных тупиков, которые задерживают доставку пакетов. Для дальнейшего изучения поведения решеток была выбрана рабочая нагрузка в 30 %, которая достигается при *wl*=30.0.

Таблица 1

Интенсивность рабочей нагрузки (*wl*)	*Шаг (Step)*	*Время (Time)*	Количество отправленных пакетов *qsnd-qself*	Количество принятых пакетов *qrcv*	Производительность решетки *gp* (packets/MTU)	Среднее время доставки пакета (MTU)
90.0	10000000	1538449	530029	529997	0,34	78
30.0	10000000	512851	529941	529868	1.03	79
10.0	10000000	171028	530036	529743	3.1	97
9.0	10000000	153805	529765	529348	3.44	130
8.0	583651*	11326	33464	28434	2.51	236
5.0	141824*	1825	9458	6229	3.41	231

rT=*sT*=5, *bs*=100; * – *решетка приходит к полному тупику – отсутствуют активные переходы*

Моделирование злонамеренного трафика

Для моделирования злонамеренного трафика были построены модели пушек пакетов и их копии были присоединены к границам решетки. Были изучены следующие основные характеристики пушек, которые влияют на поведение модели: количество и расположение, их мишени и интенсивность работы.

Простая модель пакетной пушки представлена на рис. 1, *г*. Ее работа напоминает передающий тракт узла DTE (рис. 1, *в*), но два адреса, источника *a* и *ta* назначения (мишени) заданы маркировкой внешних позиций *ga* и *ta* соответственно. Периодичность запуска пушек определено случайной функцией *gDelay()* с Пуассоновским распределением и параметром *gl*. Число сгенерированных пакетов (выстрелов) вычисляется в позиции *q_sndg* объединенного множества позиций *qsndg*. Отметим, что потребление пакетов производимых пушкой выполняет обычное терминальной устройство DTE.

Модель решетки с рабочей нагрузкой была дополнена пушками пакетов и исследовано их влияние на поведение модели. Одиночная пушка пакетов и пара пушек пакетов были присоединены к разным граничным узлам с использованием разных мишеней. В качестве основных характеристик использованы: параметр интенсивности работы пушек *gl*, интервал времени, за который модель попадает в полный тупик, а также процент пакетов созданных пушками. Случайная схема присоединения одиночной пушки или нескольких пушек с произвольными мишенями несущественно влияет на поведение решетки.

Наиболее существенные результаты получены для следующих конфигураций:

– одиночная пушка, которая стреляет в диагональном направлении – «диагональ»;

– пара пушек с взаимными мишенями – «дуэль трафика»;

– пара пушек с общей мишенью – «фокус трафика»;

– пара пушек с перекрестными мишенями – «перекрестный огонь».

Дуэль трафика может быть обозначена как (4,0)<->(4,9) с индикацией узлов присоединения пушек и их мишеней; характеристики работы решетки описаны в табл. 2. Дуэль трафика приводит к полному тупику через дополнительную нагрузку менее чем 10%. Следующие конфигурации пушек и мишеней иллюстрируют: диагональ (4,0) ->(8,9); фокус (4,0) ->(9,4), (4,9) ->(9,4); перекрестный огонь (4,0) ->(9,8), (4,9) ->(9,1). Их характеристики сравниваются в табл. 3. Выстрелы по диагонали изучаются, потому что прямые выстрелы не блокируют решетку; например, единственная пушка (4,0) ->(4,9) не блокирует решетку даже при *gl*=1.0. Схема полного тупика изображена на рис. 2; на стрелках указаны количества пакетов в соответствующих секциях буферов.

Таблица 2

Интенсивность трафика пушки (*gl*)	Шаг (*Step)*	Время (*Time)*	Количество отправленных пакетов *qsnd-qself*	Количество полученных пакетов *qrcv*	Количество пакетов выстрелянных из пушек *gsnd*	Производительность решетки *gp* (packets/MTU)	Среднее время доставки пакета *adt* (MTU)
4.0	191081*	23711	12774	8057	1826	0,34	76
5.0	254671*	26673	15430	11574	2687	0,43	73
6.0	10000000	392083	405187	535697	130658	1,37	67
9.0	10000000	425434	439792	534381	94688	1,26	66

rT=*sT*=5, *bs*=100, *wl*=30.0; (4,0)<->(4,9); * – *решетка приходит к полному тупику – отсутствуют активные переходы.*

В табл. 3 сравнение выполнено для одинаковой интенсивности пушек *gl*=5.0; демонстрируется факт, что фокус трафика приводит решетку к тупику быстрее всего, а дуэль трафика можно более легко замаскировать, потому что она несущественно влияет на качество обслуживания, поэтому тупик происходит мгновенно.

Таким образом, результаты имитационного моделирования подтвердили гипотезу, выдвинутую в [3, 4] что решетка может быть заблокирована посредством злоумышленного трафика. В простейших случаях нужны одна или две пушки трафика, добавляющие нагрузку, которая не превышает 5 % от пиковой нагрузки решетки. Подробные описания композиции моделей и полученных результатов представлены в [8].

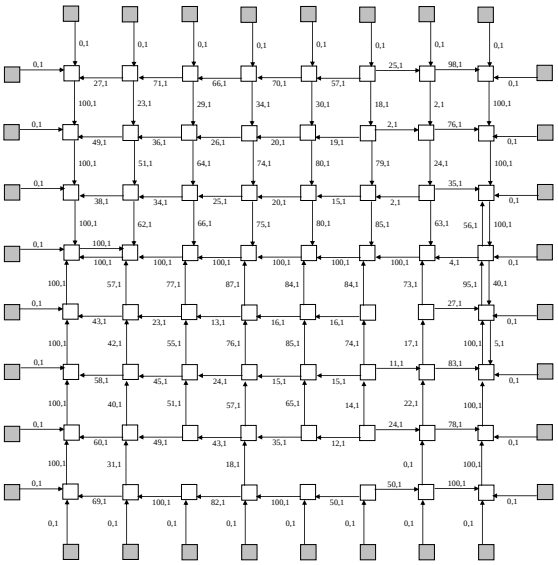


Рис. 2. Схема полного тупика, пример
(bs=100, *wl*=10.0, *gl*=5.0, *td=(4,0)<->(4,9), Step*=290529, *Time*=10941*).*

Таблица 3

Конфигурация пушек	Время полного тупика (MTU)	Процент выстрелянных пакетов	Производительность решетки *gp* (packets/MTU)	Среднее время доставки пакета *adt* (MTU)
Диагональ	43162	13%	0.67	131
Дуэль	26673	15%	0.43	73
Фокус	20199	21%	0.43	224
Перекресток	30510	25%	0.62	110

Таким образом, исследованы вопросы блокирования решеток посредством злонамеренного трафика в среде моделирующей системы CPN Tools. Показано, что даже при низкой рабочей нагрузке около 30 % решетка может быть блокирована полным тупиком узлов DCE с помощью дуэли трафика с дополнительной нагрузкой менее чем 5 %. Таким образом, подтверждена уязвимость структур решетки к атакам злонамеренного трафика.

Основное направление перспективных исследований – построение более реалистичных моделей современных архитектур узлов DCE, например с возможностью сквозной передачи пакетов без обязательной буферизации (cut-through). Реальные устройства, преодолевают тупики, как локальные, так и глобальные, используя две простые функции: потерю пакетов, приходящих в перегруженное устройство и очистку заполненных буферов по таймауту.

Конечная цель работы – создание контрмер для обнаружения и противодействия атакам различных видов трафика на решетки, которые позволят избежать падения производительности и качества обслуживания.

Список литературы: 1. *Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications* (4 Vol.), Information Resources Management Association (USA), IGI-Global, 2012, 2134 pp. 2. *Preve, N.P.* (Ed.). Grid Computing: Towards a Global Interconnected Infrastructure. Springer, 2011, 312 p. 3. *Shmeleva T.R., Zaitsev D.A., Zaitsev I.D.* Analysis of Square Communication Grids via Infinite Petri Nets // Transactions of Odessa National Academy of Telecommunication, no. 1 (2009) 27–35. 4. *Зайцев, Д.А., Шмелёва, Т.Р.* Верификация коммуникационных структур гиперкуба параметрическими сетями Петри // Кибернетика и системный анализ, №1, 2010, С. 119–128. 5. *Jensen, K., Kristensen, L.M.* Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer, 2009, 384p. 6. *Zaitsev, D.A., Shmeleva, T.R. Simulating Telecommunication Systems with CPN Tools: Students' book*. Odessa: ONAT, 2006, 60 p. 7. *Retschitzegger W., Baumgartner N., Gottesheim W., Mitsch S., Schwinger W.* Situation Prediction Nets – Playing the Token Game for Ontology-Driven Situation Awareness // Proceedings of 29th International Conference on Conceptual Modeling (ER), Vancouver, Nov. 2010. 8. *Аналіз ефективності обчислювальних граток розфарбованими сітями Петрі*. Звіт про НДР, номер держреєстрації 0113U002700 / Д.А. Зайцев, Т.Р. Шмельова // Одеса : МГУ, 2014.– 51 с.

Международный гуманитарный университет *Поступила в редколлегию 11.02.2014*