УДК 004.942

Моделирование динамики взаимодействия WEB-сервера и пула клиентов в компьютерной сети в условиях атаки «Отказ в обслуживании»

Д. С. Кулябов, Д. Е. Ланеев

Кафедра системы телекоммуникаций
Российский университет дружбы народов
ул. Миклухо-Маклая, 6, 117198, Москва, Россия

С помощью симулятора сетей NS2 моделируется и исследуется взаимодействие web-сервера и пула клиентов в сети РУДН «Донская». Моделируется DoS-атака Ping Flood в этой сети и анализируются её последствия. Произведена оценка возможностей внешнего Интернет канала локальной компьютерной сети РУДН «Донская».

Ключевые слова: симулятор, атака, имитационное моделирование, NS2, DoS, HTTP, Ping Flood.

Введение

С появлением Интернета и глобализацией сетевой инфраструктуры возникла угроза безопасности приватных данных, ценных корпоративных ресурсов и т.д. Все чаще и чаще злоумышленники пытаются взломать системы безопасности с целью получения доступа к важнейшим ресурсам и данным с помощью специализированных атак. Эти атаки, такие как рассматриваемые в работе атаки DoS (Denial of Service — атака типа «отказ в обслуживании») [1], становятся все более изощренными и простыми в исполнении.

Важность проблемы сетевой безопасности мало у кого вызывает сомнения. Многочисленные взломы, сетевые атаки, иные разнообразные преступления в сфере высоких технологий никому уже не в новинку. Однако не все знают, что помимо прямого ущерба при взломе системы, связанного с хищением данных или несанкционированным использованием ресурсов, владельцу взломанной системы угрожает и косвенный ущерб: если система была использована как стартовая площадка для атаки, его могут привлечь к уголовной ответственности за несоблюдение необходимых мер защиты от взломщиков. Методики взлома систем постоянно совершенствуются [2]. Обнаруживаются новые уязвимые места в программном обеспечении, забывается об устранении уже известных. Традиционные меры защиты — брандмауэры и сетевые системы обнаружения вторжения (Network Intrusion Detection System, NIDS) — не всегда способны предотвратить сетевую атаку. А результат успешно осуществленной атаки — взлом системы. Для надежной защиты необходимо иметь представление об инструментарии и методах взломщиков.

Разработка и поддержка сетевой защиты — это интеллектуальное соревнование между нападающим и защитником. Постоянное развитие методик нападения и защиты требует пристального внимания. Ценой совершенной ошибки будет потерянная информация [3]. Для разработки плана защиты необходимо иметь представление о сетевой активности защищаемой системы. Необходимо иметь представление и обладать возможностью контроля системных сервисов и пользовательских приложений. Детальное представление схемы сетевой активности существенно облегчит построение сетевой защиты [3].

Моделируемая в данной работе атака DoS является наиболее известной и распространенной формой хакерских атак. Кроме того, против атак такого типа труднее всего создать стопроцентную защиту. Такие атаки не требуют углубленных

знаний в программировании и среди хакеров считаются наиболее простыми в применении. Тем не менее, они приносят вред не меньший, а часто и больший, чем другие атаки. Таким образом они вынуждают сетевых администраторов и специалистов по компьютерной безопасности принимать меры по их нейтрализации и обнаружению. DoS-атаки несут угрозу глобальной межсетевой инфраструктуре. Пока алгоритм контроля переполнения в TCP стеке устойчив к различным сетевым условиям, он подразумевает наличие уязвимости к высокоскоростным атакам, которые обычно осуществляются множеством систем [4]. В этой работе рассматриваются также и низкоскоростные DoS атаки, которые, в отличие от высокоскоростных, хуже обнаруживаются роутерами и IDS (Intrusion Detection System).

Проектирование современной сетевой инфраструктуры, несомненно, требует предварительного компьютерного моделирования разрабатываемой системы. Известно, что модель аппроксимирует свойства и поведение исследуемой сети и, как следствие, позволяет решать задачи по оптимизации и ее управлению. Также на модели возможна апробация тех или иных решений, что несравнимо дешевле, чем на реальной системе, и исключает возможные ошибки в ней.

Одним из средств, которое позволяет осуществлять имитационное моделирование сетей связи и обладает целым рядом достоинств, среди которых высокая производительность, хорошая масштабируемость, визуализация результатов и гибкость, является Network Simulator 2 [5]. С помощью графического аниматора Nam (Network Animator – Аниматор сети), входящего в полную версию NS2 [6], возможна наглядная визуализация работы смоделированной системы, наблюдение движения пакетов по сети, их потери, состояния очередей с различными дисциплинами обслуживания [7].

В разделе 1 работы рассмотрены архитектура и возможности системы NS2, с помощью которой осуществляется моделирование. В разделе 2 работы описывается DoS-атака Ping Flood [8,9]. В разделе 3 моделируется сценарий взаимодействия web-сервера и пула пользователей на примере локальной компьютерной сети РУДН «Донская», с целью оценки возможностей внешнего Интернет канала локальной компьютерной сети. Также моделируется DoS-атака Ping Flood, происходящая в данной сети в определенный момент времени. Произведен анализ результатов моделирования.

1. NS2 — средство для имитационного моделирования

Под имитационной моделью понимается математическая модель, реализованная как программное обеспечение для компьютера и использующая специальные или стандартные языки программирования. Для построении такой модели могут применяться как статические модели, используемые для мониторинга состояния сети в заданные моменты времени, так и динамические модели — дискретные модели. Примером статической модели могут служить аналитические методы расчета из теории массового обслуживания. Примером же динамической модели могут служить процессы генерации заявок или пакетов, либо процессы их обработки. На сегодняшний день существует несколько программных средств для осуществления имитационного моделирования сетей связи: от библиотек функций для стандартных компиляторов до специализированных языков программирования. Они различаются языком программирования, используемым для реализации собственных пользовательских функций и протоколов.

NS2, как и его предшественники, является программным обеспечением с открытым исходным кодом. Отсюда вытекает целый ряд преимуществ. ПО такого типа распространяется бесплатно, отсутствуют ограничения на право использования, модификации и распространения. Также преимуществом данных систем является доступность обновлений и дополнений (новые библиотеки, протоколы и т.п.).

NS2 является мультиоперационным программным средством. Установка полной версии NS2 возможна под управлением следующих операционных систем: SunOS,

Solaris, Linux, FreeBSD, Windows 95/98/ME/NT/2000. Для установки полной версии NS2 необходимо наличие компилятора C++. Важным достоинством полной версии NS2 является возможность модификации его ядра и специализированная под конкретного пользователя настройка.

Существует и урезанная версия (компилированная) NS2, предназначенная для работы в некоторых ОС, в частности в ОС Windows. Ограничения заключаются в отсутствии возможности перекомпиляции ядра и добавления новых компонентов. В то же время эта версия обладает рядом достоинств: для её установки требуется всего 3 МБ свободного места на жестком диске компьютера, она проста в использовании и инсталляции.

В NS2 на уровне ядра реализованы почти все существующие на сегодняшний день протоколы: MPLS, IPv6, OSPF, а также множество дисциплин обслуживания очередей: RED, WFQ, CBQ, SFQ и т.д. В качестве системного языка используется C++, позволяющий обеспечить высокую производительность и возможность модификации ядра NS2 для поддержки пользовательских протоколов и функций. В качестве управляющего интерфейса используется язык OTcl. Определять и управлять моделью можно как через интерфейс OTcl, так и с использованием языка C++.

В пакет NS2 входит средство анимации результатов моделирования Nam [5], с помощью которого можно наблюдать работу моделируемой сети, её топологию, анимацию движения пакетов по каналам связи или их потери, вести визуальный мониторинг состояния очередей и проводить анализ данных [7]. Полная версия NS2 также содержит визуализатор Xgraph [10], с помощью которого можно изобразить графически полученные результаты моделирования, находящиеся в соответствующем файле, в который записываются данные в процессе симуляции.

В NS2 существует и математическая поддержка, позволяющая создавать различные виды трафика, как, например, трафика, подчиняющегося пуассоновскому закону, так и самоподобному и т.д. Гибкая архитектура NS2 даёт возможность её пользователям реализовывать собственные математические функции и законы на C++ [11].

В NS2 также реализована возможность моделирования возникновения ошибок (искажение или потеря информации) на канальном уровне, на уровне битов или пакетов. Есть возможность реализации собственной модели ошибок [11].

2. Описание моделируемой атаки Ping Flood

2.1. Атака «Отказ в обслуживании» (Denial of Service — DoS)

DoS-атаки выделяются среди других сетевых атак, так как в их задачи не входит получение доступа к определенным ресурсам или информации. Их целью является вывод из строя серверов, рабочих станций, брандмауэров, маршрутизаторов или, таким образом, всей атакуемой сети в целом. В то же время такие атаки могут использоваться как вспомогательные для вывода из строя брандмауэра или маршрутизатора, чтобы затем беспрепятственно проникнуть в атакуемую сеть и получить доступ к нужной информации.

Наиболее известные разновидности DoS: TCP SYN Flood, Ping of Death, Tribe Flood Network (TFN) и Tribe Flood Network 2000 (TFN2K), Trinco, Stacheldracht, Trinity, Ping Flood [1].

DoS-атаки выводят из строя объекты, перегружая их информацией, которую они должны обрабатывать. Например, при атаке на серверные приложения, Web-сервер или FTP-сервер, DoS-атаки занимают все возможные соединения, сервисы и держат их в занятом состоянии, что парализует работу сервера. DoS-атаки могут использовать интернет-протоколы, такие как TCP и ICMP. Большинство DoS-атак могут использовать ошибки в программном коде, в системах безопасности и общие слабости системной архитектуры. Но некоторые типы DoS-атак перегружают сеть пакетами или запросами, тем самым парализуя её работу. Такой тип атаки трудно предотвратить, так как для борьбы с ней необходимо перекрывать или фильтровать ненужный трафик. Но делать это имеет смысл лишь у провайдера, так как

фильтрация на входе в атакуемую сеть не даст результатов из-за того, что канал уже будет забит ненужными пакетами. В DoS-атаках может участвовать множество узлов, при этом увеличивается эффективность таких атак. Такие DoS-атаки называют распределенными DoS-атаками (DDoS — Distributed DoS) [4].

Угроза DoS-атак может снижаться следующими способами:

1. Использование функций анти-DoS. Соответствующая конфигурация на маршрутизаторах и брандмауэрах может снизить эффект от атак DoS, например ограничивая число полуоткрытых каналов.
2. Ограничение и фильтрация трафика (traffic rate limiting) [4]. Организация может попросить своего провайдера (ISP) ограничить объем второстепенного трафика, например ICMP-трафика, который используется для диагностических целей, в то же время атаки типа DoS часто его используют. Примером может служить моделируемая в данной работе атака Ping Flood.

2.2. Затопление ICMP-пакетами, атака PING FLOOD

Одной из разновидностей DoS-атак является Ping Flood [8,9]. Эта атака использует программу «ping», предназначенную для оценки качества линии, в режиме «агрессивного» тестирования. В этом режиме запросы посылаются с минимальными интервалами между ними. Данный режим используется для оценки работоспособности сети при максимальной нагрузке. Программа «ping» работает следующим образом: программа посылает ICMP-пакет [12] типа ECHO REQUEST, выставляя в нем время и его идентификатор; ядро машины-получателя отвечает на подобный запрос пакетом ICMP ECHO REPLY; программа также выдает время прохождения пакета. При обычном режиме работы «ping» не загружает сеть, но в агрессивном режиме может вызвать перегрузку небольшого канала, тем самым парализовать его работу. Атакуемая система тратит свои вычислительные ресурсы, отвечая на эти запросы. Результатом атаки может стать снижение производительности атакуемого компьютера. Атака Ping Food может быть и распределенной. В этом случае посылаются ICMP-пакеты типа echo request с исходным адресом атакуемой системы на broadcast-адреса крупных сетей. Таким образом, каждая из машин, получившая этот запрос, ответит на него, послав ICMP-пакет типа ECHO REPLY на адрес атакуемой машины. Тем самым произойдет перегрузка канала, на котором находится атакуемая система. Распознать такую атаку можно по резко возросшей нагрузке на сеть или по повышенному количеству ICMP-пакетов. Меры защиты могут заключаться в настройке маршрутизаторов, при которой они будут фильтровать ICMP-трафик или лимитировать его.

3. Построение модели и анализ результатов

Для оценки реальных возможностей внешнего Интернет канала сети РУДН «Донская» необходимо построить модель циркуляции web-трафика между локальной сетью РУДН «Донская» и web-сервером. Также нужно смоделировать DoS-атаку ping flood, как вполне вероятное событие в современной компьютерной сети и наиболее простой тип атаки, тем самым выяснить насколько канал чувствителен к атакам подобного вида.

3.1. Построение модели web-сервера и пула клиентов

По данным статистики http-трафик ежедневно создают около 200 пользователей Интернет сети РУДН «Донская». В то время как общее число пользователей составляет 500. В данной модели нас интересует только среднее число пользователей Интернет, использующих канал в данный момент времени, что составляет приблизительно 50 пользователей. В среднем в день скачивается около 2 Гбайт данных.

Моделируемая компьютерная сеть состоит из 2 коммутаторов, 2 маршрутизаторов, одного сервера и 50 пользователей. Коммутаторы №1, №2 и маршрутизатор

№3 находятся в пользовательской сети. Маршрутизатор №4 (сеть РУДН «Крест») является промежуточным маршрутизатором между пользовательской сетью и web-сервером.

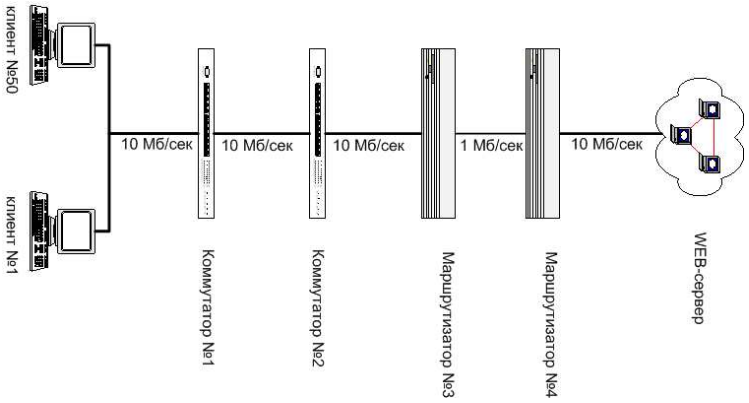

Рис. 1. Модель компьютерной сети

К коммутатору №1 подсоединены 50 пользователей дуплексным каналом в 10 Мбит/сек каждый с задержкой 1 мс. Коммутаторы №1, №2 и маршрутизатор №3 соединены каналом 10 Мбит/сек с задержкой 1 мс. Маршрутизатор №3 является конечным в сети РУДН «Донская». Он подключен к маршрутизатору №4 дуплексным каналом в 1 Мбит/сек с задержкой 10 мсек. В свою очередь маршрутизатор №4 соединен каналом 10 Мбит/сек с задержкой 10 мсек с узлом №5, который будем считать web-сервером или внешней сетью. Все запросы пользователей Интернет поступают к нему. Все остальные промежуточные маршрутизаторы на пути к провайдеру объединены в один маршрутизатор №5. Узким местом в сети будет канал в 1 Мбит/сек. между маршрутизатором №4 и №3, то есть между сетью РУДН «Донская» и сетью РУДН «Крест». Поэтому производить мониторинг очереди имеет смысл лишь на маршрутизаторе №4. Таким образом, между маршрутизатором №4 и №3 будет находиться очередь типа RED размером 1000 пакетов.

За основу пользовательской сети, то есть сети, пользователи которой посылают запросы на web-сервер, был взят реальный сегмент сети РУДН «Донская». Число пользователей этой сети было принято равным 50, близкое к реальному. Несмотря на то, что была взята только часть реальной пользовательской сети, количество пользователей совпадает с действительным, и модель будет описывать реальное поведение сети, так как узкое место будет находиться не в пользовательской сети, а на оконечном маршрутизаторе между провайдером и пользовательской сетью.

Для построения модели и симуляции будем использовать симулятор NS2. Построим модель, в которой клиенты открывают несколько web-страниц с сайтов, находящихся на одном web-сервере. Каждая страница содержит несколько объектов, каждый из которых доставляется по протоколу TCP. Итак, в рамках NS2 модель будет состоять из 4 промежуточных узлов, 1 узла — сервера и 50 узлов — клиентов:

```
set num_server 1
set num_client 50
```

Общее количество узлов, учитывая промежуточные, будет составлять 55:

```
set num_node [expr 4 + [expr $num_client + $num_server]]
```

Создадим num_node (55) узлов:

```
for {set i 0} {$i < $num_node} {incr i} {
  set n($i) [$ns node]}
```

Соединим узлы №1, №2, №3, №4 каналами с соответствующей моделируемой сетью пропускной способностью и задержкой. Также укажем тип очереди RED и размер очереди в 1000 пакетов на узле №4:

```
$ns duplex-link $n(1) $n(2) 10Mb 1ms DropTail
$ns duplex-link $n(2) $n(3) 10Mb 1ms DropTail
$ns duplex-link $n(3) $n(4) 1.0Mb 10ms RED
$ns queue-limit $n(4) $n(3) 1000
```

Соединим сервер и узел №4:

```
for {set i 0} {$i < $num_server} {incr i}
  {$ns duplex-link $n(4) $n([expr $i + 5]) 10Mb 10 DropTail }
```

Подсоединим 50 узлов (клиентов) к узлу №1 (коммутатор №1):

```
#conection from 1 to clients
for {set i 0} {$i < $num_client}{incri}
  {$ns duplex-link $n(1)$n([expr[expr $i+$num_server]+5]) 10Mb
    1ms DropTail}
```

Зададим пул серверов (в данном случае состоящий из одного сервера) и пул клиентов:

```
$ns set dst_ "";  #define list of web servers
for {set i 0} {$i<$num_server} {incr i} {
  $ns set dst_ "[$ns set dst_] [list [expr $i + 5]]"
  $ns set src_ "";  #define list of web clients
  for {set i 0} {$i < $num_client} {incr i} {
    $ns set src_ "[$ns set src_][list [expr[expr $i+$num_server]+
5]]"}
```

Зададим файл для записи событий симуляции:

```
#Open the Trace file
set tf [open out.tr w]
$ns trace-all $tf
```

Зададим количество сессий равным 250, в среднем 5 на одного клиента:

```
#Number of Sessions
set numSession 250
```

Для каждой сессии зададим распределение времени между отправкой запросов к страницам и объектам экспоненциальное со средними значениями соответственно 5 секунд и 10 мс. [13]. Каждая страница содержит тридцать объектов с размером по распределению Парето с параметром 1.1 [13, 14]:

```
set launchTime 0 for {set i 0} {$i < $numSession} {incr i}
{
   set interPage [new RandomVariable/Exponential]
   $interPage set avg_ 5

   set pageSize [new RandomVariable/Constant]
   $pageSize set val_ 30
   set interObj [new RandomVariable/Exponential]
   $interObj set avg_ 0.01
   set objSize [new RandomVariable/ParetoII]

   $objSize set shape_ 1.1
   $pool create-session $i $numPage [expr $launchTime + 0.1] \
           $interPage $pageSize $interObj $objSize
   set launchTime [expr $launchTime + [$interSession value]]
}
```

Будем вести мониторинг среднего и текущего размера очереди между узлом №4 и №3 и записывать значения в файл:

```
#Tracing a queue
set redq [[$ns link $n(2) $n(0)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_ $redq
attach $tchan_
```

Выделим данные о среднем размере очереди и её текущем значении и запишем в файл temp.queue. Запустим xgraph для визуализации данных, находящихся в этом файле:

```
set awkCode {
  {
   if ($1 == "Q" && NF>2) {
     print $2, $3 >> "temp.q";
     set end $2

   }
   else if ($1 == "a" && NF>2)
   print $2, $3 >> "temp.a";
  }
  }
  set f [open temp.queue w]
  puts $f "TitleText: red"
  puts $f "Device: Postscript"

  if { [info exists tchan_] } {
  close $tchan_
  }
  exec rm -f temp.q temp.a
  exec touch temp.a temp.q

  exec awk $awkCode all.q

  puts $f \"queue
  exec cat temp.q >@ $f

  puts $f \n\"ave_queue
  exec cat temp.a >@ $f
  close $f

  exec ../xgraph-12.1/xgraph -bb -tk -x time -y queue \
    temp.queue &
```

Симуляция будет длиться 400 секунд:

```
$ns at 400 "finish"
```

На этом построение модели циркуляции HTTP-трафика закончено и можно приступить к построению модели DoS-атаки.

3.2. Моделирование атаки PING FLOOD

В построенной выше модели компьютерной сети моделируется DoS-атака ping flood. Атака исходит с узла №5, то есть из внешней сети. Её цель — перегрузка канала в 1 Мбит/сек. ICMP-пакетами и, таким образом, парализация его работы. На узле №5 и №3 находится PING-агент:

```
set p1 [new Agent/Ping]
$ns attach-agent $n(5)
$p1 set p0 [new Agent/Ping]
$ns attach-agent $n(3) $p0
#Connect the two agents $ns
connect $p0 $p1
```

Ping-агент на узле №5 будет генерировать ICMP-пакеты размером 64 байта и посылать их на узел №3. Агент PING на узле №3 в ответ на пришедший ICMP-пакет типа ECHO REQUEST ответит на подобный запрос пакетом ICMP ECHO REPLY. Пакеты будут генерироваться с интервалом 5×10^{-4} секунд. Генерация будет длиться 10 секунд, начиная с момента времени 150 секунд до момента времени 160 секунд. Всего будет послано 20000 ICMP-пакетов. Таким образом будет генерироваться ICMP-трафик в 1 Мбит/сек.:

```
set kk 0.0005
set j 150.0
for {set i 1} {$i < 20000} { incr i} {
  set j [expr $j+$kk]
  $ns at $j "$p0 send" }
```

Запустим симулятор NS2, чтобы начать симуляцию с ранее указанной продолжительностью:

```
puts "ns started"
$ns run
```

Построенная выше модель DoS-атаки относится к высокоскоростным, то есть происходит перегрузка канала огромным количеством пакетов в течение определенного промежутка времени. Очень часто встречаются низкоскоростные DoS-атаки, которые в отличие от высокоскоростных хуже обнаруживаются роутерами и IDS. Низкоскоростные DOS-атаки или периодические импульсные атаки представляют собой потоки трафика в виде импульсов. Они состоят из коротких со специально подобранной длительностью пиков, повторяющихся с определенной, специально выбранной частотой.

Теперь рассмотрим подробнее, как работает такой тип DoS-атаки в сочетании с механизмом тайм-аутов TCP [15]. Рассмотрим один поток TCP-трафика. Если общий трафик (DOS атаки и обычный) в течение пика достаточен, чтобы произошли потери пакетов, то этот поток прекратится по тайм-ауту и будет произведена попытка отправить новый пакет по прошествии времени RTO (Retransmission Timeout) [4].

Механизм тайм-аутов, с одной стороны, обеспечивает устойчивый алгоритм управления перегрузкой канала, с другой, предоставляет возможность проведения низкоскоростной атаки, которая использует уязвимость динамики изменения таймеров повторной передачи. В частности, атакующий может вызвать циклическое прекращение по тайм-ауту потока трафика, отправляя мощные короткие импульсы трафика, имеющие длительность, сопоставимую с RTT (Round-Trip Time) шкалой и периодичность, сопоставимую с RTO. Если периодичность посылки DoS-трафика совпадает с RTO нормального трафика, то обычный трафик будет постоянно получать тайм-аут, вследствие чего потери будут приближаться к 100%, а пропускная способность приблизится к нулю. Кроме того, если период DoS посылок примерно равен RTO, то будет наблюдаться существенное (но не полное) снижение полосы пропускания. Несмотря на снижение пропускной способности атакуемого, средняя скорость атаки будет довольно низкой, что затрудняет обнаружение подобной атаки [4].

Смоделируем и такой тип атаки. Для того чтобы смоделировать низкоскоростную DoS-атаку, необходимо смоделировать периодические (импульсные) посылки ICMP-пакетов с определенной длительностью, сопоставимой с RTT шкалой и периодичностью более медленной шкалы RTO. Для этого будем генерировать ICMP пакеты с интервалом 5×10^{-4} секунд в течение 4.5 секунд в цикле через 1 секунду:

```
set kk 0.0005
set j 150.0 for {set k 1} {$k < 11} {incr k} {
  for {set i 1} {$i < 9000} { incr i} {
    $ns at $j "$p0 send"
    set j [expr $j+$kk] }
  set j [expr $j+1] }
```

Таким образом смоделировано 11 импульсов и сгенерирован ICMP-трафик в 0.8 Мбит/сек. длительностью 55 секунд.

3.3. Анализ результатов моделирования

С помощью визуализатора Xgraph изобразим графически полученные результаты моделирования, находящиеся в соответствующем файле.

Таким образом, на рис. 2 представлен график, показывающий средний размер очереди на узле №4 при количестве пользователей Интернет равным 50. Средний размер очереди стремится к 1000 пакетам, максимальный размер очереди также составляет 1000 пакетов, то есть происходят потери пакетов. Рисунок 3 показывает среднее количество теряемых на узле №4 TCP-пакетов.

Рисунок 4 показывает поведение очереди на узле №4 во время высокоскоростной DoS-атаки. С момента времени 150 секунд до момента времени 160 секунд

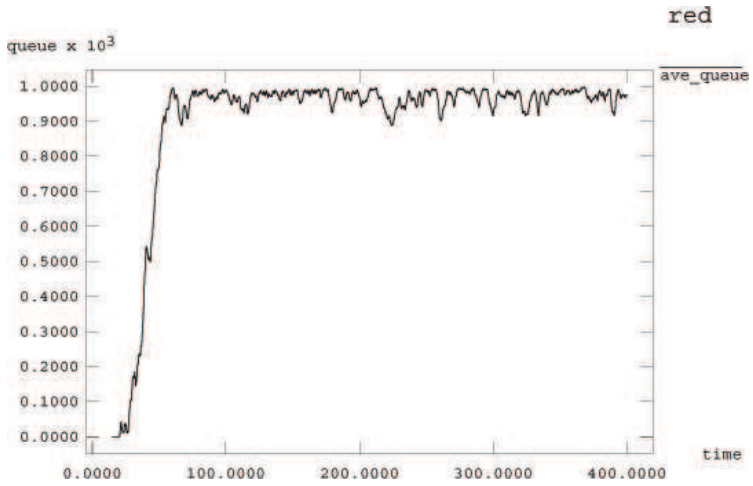

Рис. 2. Средний размер очереди на узле №4 при количестве пользователей Интернет 50

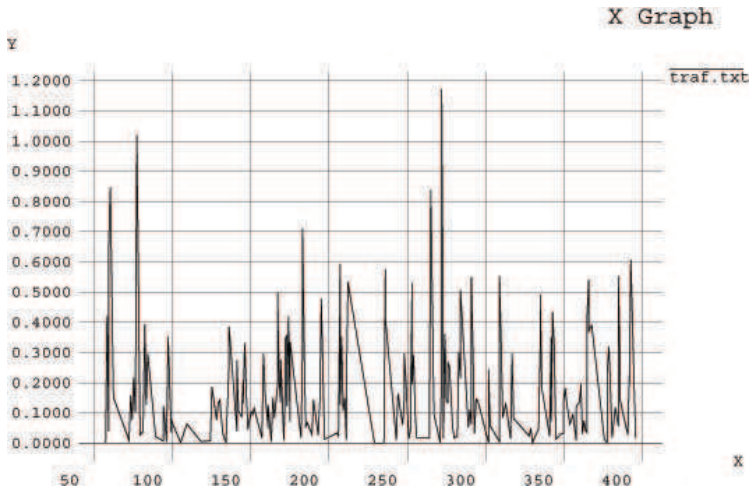

Рис. 3. Среднее количество теряемых на узле №4 TCP-пакетов

очередь заполнена до своего лимита. Происходят большие потери TCP-пакетов, отображенные на рис. 5.

С помощью ниже приведенного кода из файла с результатами моделирования out.tr можно рассчитать TCP трафик в Мбит/сек, а также количество потерянных пакетов:

```
cat out.tr | grep " 4 3 tcp " | grep ^r | ./column 5 1 |
 awk '{dif = $2 - old2; if(dif==0) dif = 1; if (dif < 0.5)
  {old1=old1+$1} if(dif >=  0.5) {printf("%d\t%f\n", $2,
  ($1 + old1) / dif*8/1000000); old1 = $1; old2 = $2}}' >
   tcp.txt

cat out.tr | grep " 4 3 tcp " | grep ^d | ./column 5 1 |
awk '{dif = $2 - old2; if(dif==0) dif = 1; if (dif < 0.5)
{old1=old1+$1} if(dif >=  0.5) {printf("%d\t%f\n", $2, ($1 +
old1) / dif*8/1000000); old1 = $1; old2 = $2}}' > drop.txt
```

На рис. 6 представлен график TCP-трафика в Мбит/сек, входящего в пользовательскую сеть через узел №4. Видно, что, начиная с момента времени 150 секунд, происходит резкое сокращение TCP-трафика. Таким образом высокоскоростная DoS-атака снижает полезную пропускную способность канала.

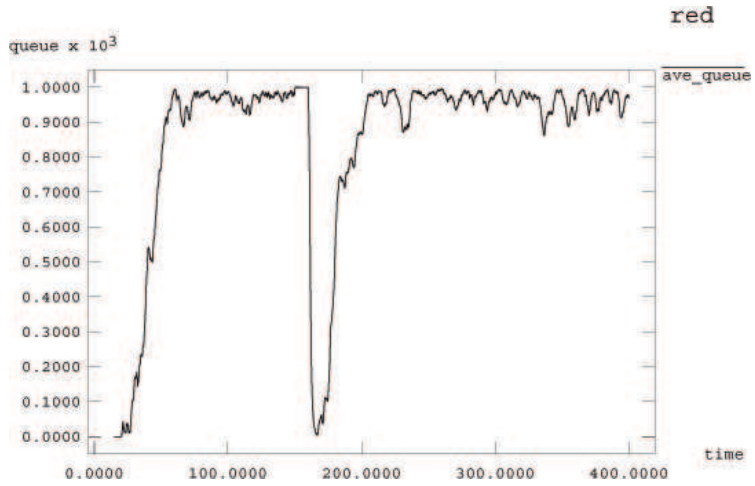

Рис. 4. Средний размер очереди на узле №4 во время высокоскоростной DoS-атаки.

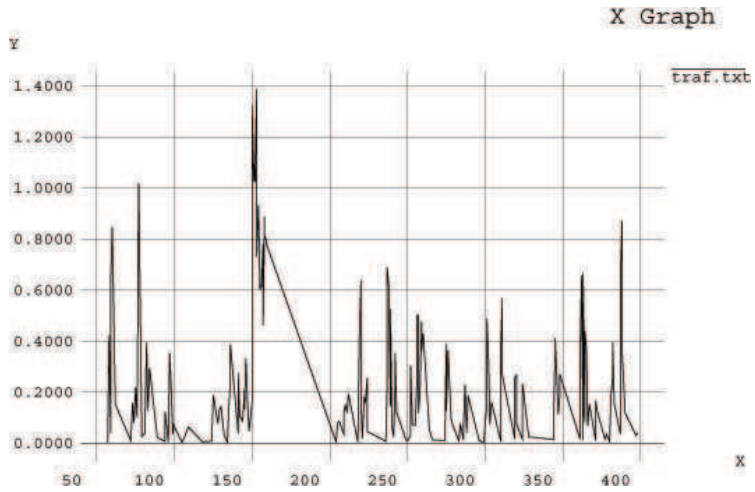

Рис. 5. Потери на узле №4 при высокоскоростной DoS-атаке

Когда канал начинает забиваться ICMP-пакетами, возникает большое количество потерь, TCP-стек многих потоков начинает работать по второй, большей временной шкале с отметками тайм-аутов повторной передачи пакетов (RTO). Чтобы избежать перегрузки канала, поток трафика уменьшается до одного пакета, и по прошествии времени RTO пакет пересылается заново. При последующих потерях, время RTO удваивается с каждым следующим тайм-аутом. Эта задержка видна на графике размера очереди. В момент времени 160 секунд, после прекращения атаки размер очереди вначале почти нулевой в течение 1-2 секунд из-за резкого уменьшения TCP-трафика. Через 1-2 секунды после окончания атаки происходит медленный старт TCP-потоков, и очередь снова заполняется.

На следующем рис. 7 представлен график размера очереди при низкоскоростной DoS-атаке ping flood. Через интервал времени 1 секунда происходят импульсные атаки длительностью 4.5 секунд. Видно, что после прерывания многих TCP-потоков по тайм-ауту будет произведена попытка отправить повторные пакеты по прошествии времени RTO. Именно в этот момент начинается очередной импульс Dos-атаки (на графике момент времени 154.5 секунд). Таким образом снова происходит потеря повторных TCP-пакетов (рис. 8), и цикл повторяется.

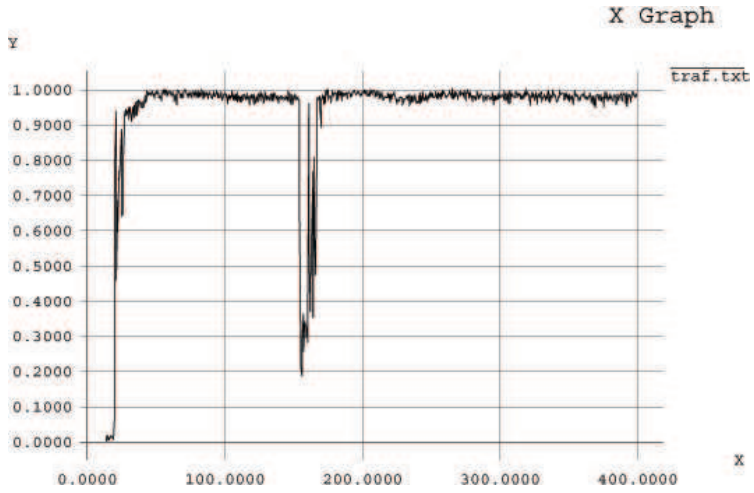

Рис. 6. TCP-трафик в Мбит/сек. из узла №4 в узел №3 при высокоскоростной DoS-атаке.

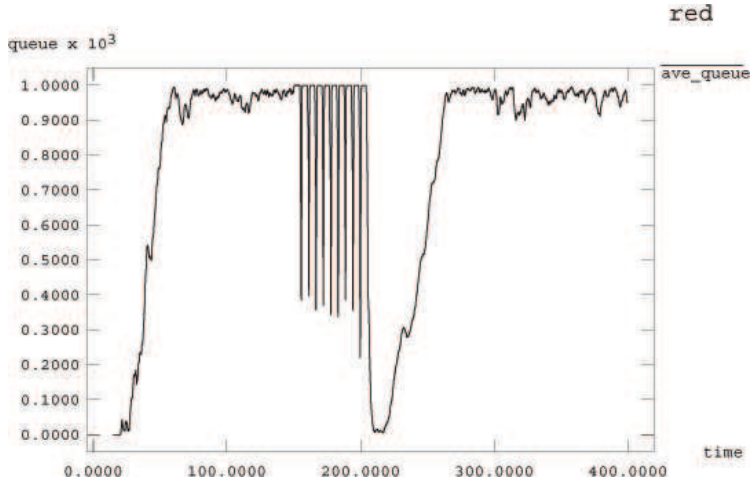

Рис. 7. Средний размер очереди при низкоскоростной DoS-атаке

На рис. 9 представлен график TCP-трафика в Мбит/сек из узла №5 в узел №4, а на рис. 10 — из узла №4 к узлу №3.

На рис. 9 мы видим, что в то время как пропускная способность канала, идущего от узла №5 до узла №4, составляет 10 Мбит/сек, канал используется в среднем лишь на 1–1.5 Мбит/сек. Это действует механизм регулирования TCP-трафика. В момент атаки TCP-трафик снижается до 0.1–0.2 Мбит/сек.

Рисунок 11 показывает размер очереди на узле №4 при нормальной работе сети и увеличенном количестве пользователей до 200. При таком количестве пользователей средний размер очереди колеблется в районе своего лимита. Подсчитаем трафик в Мбит/сек от отдельно взятого пользователя, то есть трафик, идущий от сервера к этому пользователю по его запросу при количестве пользователей 50 и 200:

```
cat out.tr | grep " 4 20 tcp " | grep ^r | ./column 5 1 |
awk '{dif = $2 - old2; if(dif==0) dif = 1; if (dif < 0.5)
{old1=old1+$1} if(dif >=  0.5) {printf("%d\t%f\n", $2, ($1 +
 old1)/ dif*8/1000000); old1 = $1; old2 = $2}}' > user.txt
```

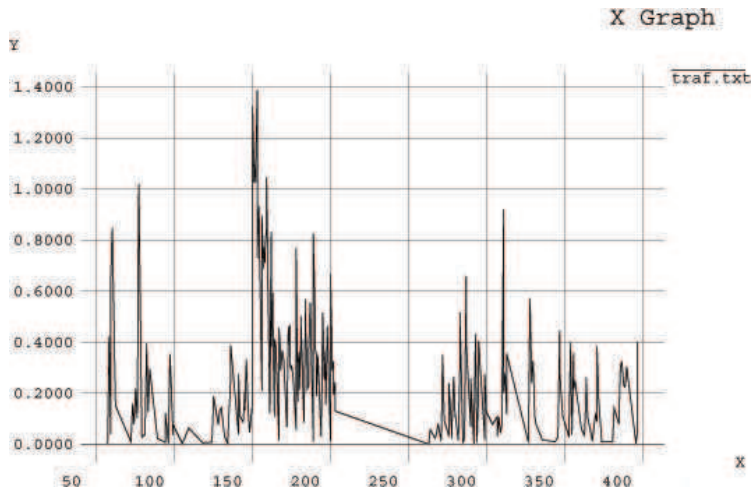

Рис. 8. Потери на узле №4 при низкоскоростной DoS-атаке.

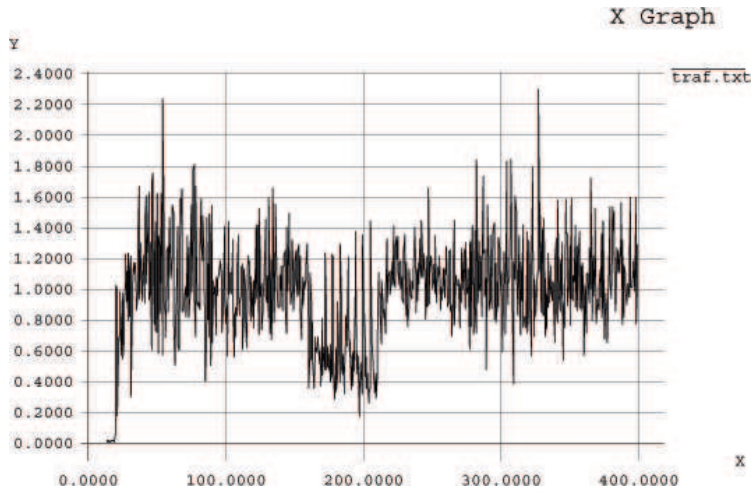

Рис. 9. TCP-трафик в Мбит/сек. из узла №5 в узел №4 при низкоскоростной DoS-атаке

Ниже, на рис. 12 и рис. 13 представлены графики TCP-трафика от одного случайно выбранного пользователя. Таким образом, средний размер трафика от одного пользователя при общем количестве пользователей 50 составляет примерно 20 кб/сек. А средний размер трафика от одного пользователя при общем количестве пользователей 200 составляет 5 кб/сек.

Можно сделать вывод что при количестве пользователей 50 в целом канал справляется с нагрузкой, так как средний трафик от одного пользователя составляет 20 кб/сек. При количестве пользователей 200 канал не справляется с нагрузкой, происходят большие потери пакетов, как следствие увеличиваются задержки и уменьшается трафик от одного пользователя.

Заключение

Сетевые атаки столь же разнообразны, как и системы, против которых они направлены. Некоторые атаки отличаются большой сложностью. Другие может осуществить обычный пользователь, даже не предполагающий, какие последствия

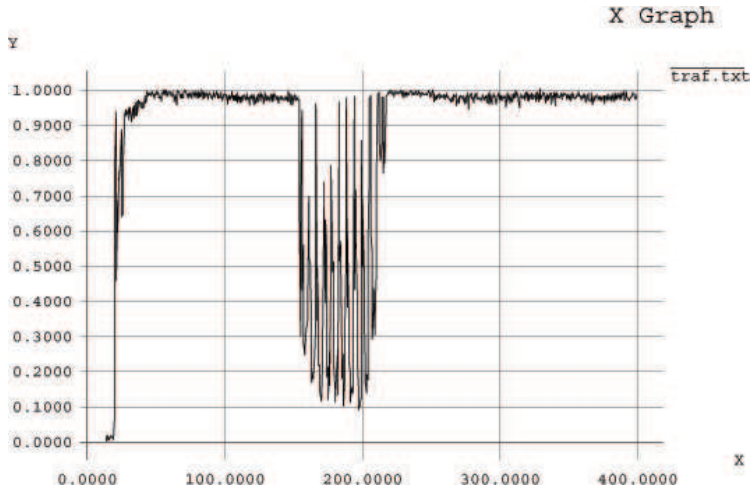

Рис. 10. TCP-трафик в Мбит/сек. от узла 4 к узлу №3 при низкоскоростной DoS-атаке

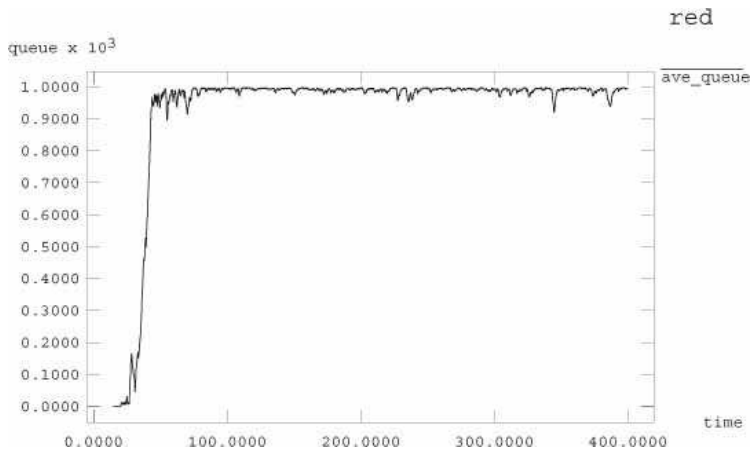

Рис. 11. Средний размер очереди на узле №4 при количестве пользователей 200

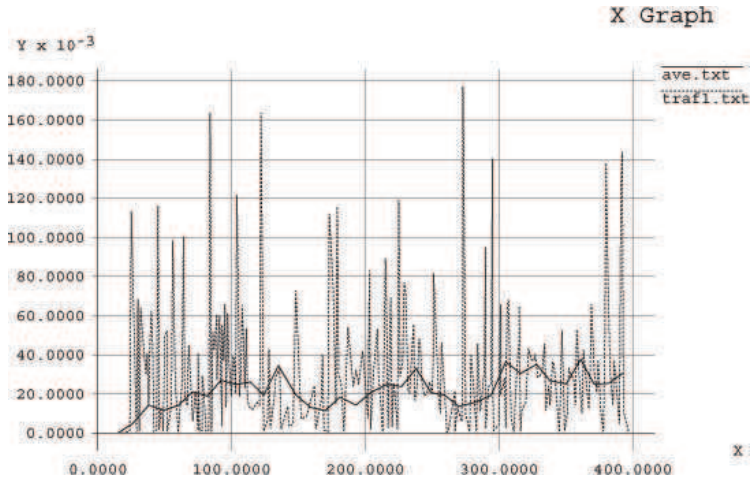

Рис. 12. Трафик от одного пользователя при общем количестве клиентов 50

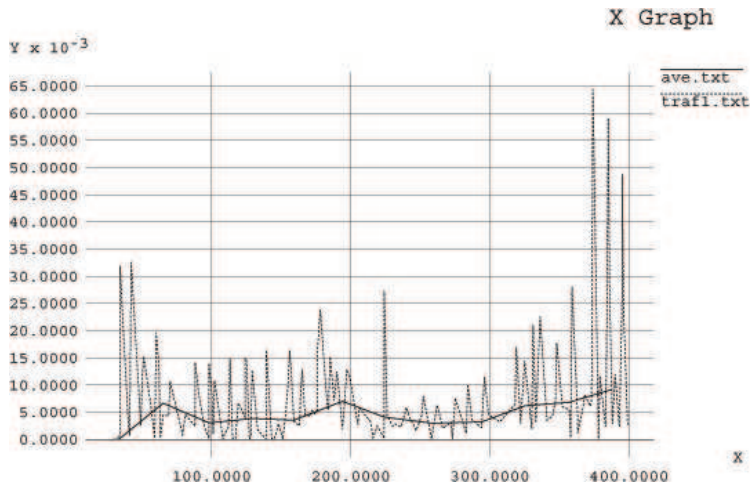

Рис. 13. Трафик от одного пользователя при общем количестве клиентов 200

может иметь его деятельность. Для оценки типов атак необходимо знать некоторые ограничения, изначально присущие протоколу TCP/IP.

В работе были рассмотрены наиболее распространенные атаки — DoS-атаки.

Была построена модель компьютерной сети, состоящей из пула пользователей Интернет, промежуточных маршрутизаторов и web-сервера. Таким образом, в работе была смоделирована циркуляция HTTP-трафика в РУДН.

Была смоделирована DoS-атака Ping Flood, как высокоскоростная, так и низкоскоростная, использующая уязвимости в протоколе TCP. Используя в своих целях механизм тайм-аутов TCP, низкоскоростная DoS-атака оказывает такое же воздействие, как и высокоскоростная, но за счет более низкой скорости ее труднее обнаружить средствами безопасности.

С помощью визуализатора Xgraph по результатам моделирования были получены графики, показывающие средний размер очереди на промежуточном маршрутизаторе между пользовательской сетью и web-сервером, размер TCP-трафика до и во время атаки, а также потери TCP-пакетов во время атаки.

Был проведен анализ результатов работы смоделированный сети до и во время атаки, а также с увеличенным количеством пользователей. Наблюдалось уменьшение полезной пропускной способности канала при проведении DoS-атаки Ping Flood как при высокоскоростной, так и при низкоскоростной. Таким образом, подтвердилась уязвимость смоделированной сети при проведении подобных атак.

Методы борьбы с такими атаками могут заключаться лишь в фильтрации ICMP-трафика ещё у провайдера, что бывает не всегда возможно.

Также при анализе результатов моделирования был сделан вывод, что если ежедневное количество пользователей Интернет сотавляет 50, канал между сетью РУДН «Донская» и поставщиком услуг Интернет условно будет справляться с нагрузкой, и максимально возможный трафик от одного пользователя составит примерно 20 кб/сек. При количестве пользователей большим 200 канал условно не будет справляться с нагрузкой, так как максимально возможный трафик от одного пользователя составит примерно 5 кб/сек., что является неприемлемым для работы в сети Интернет. Для устранения проблемы, очевидно, необходимо увеличить пропускную способность канала.

Литература

1. CERT® Coordination Center Denial of Service Attacks. — 2001. — `http://www.cert.org/tech_tips/denial_of_service.html`.
2. *Красоткин А.* Ловушка для злодея // Мир ПК. — № 4. — 2003.

3. *Красоткин А.* Щит и меч // Chip. — № 4. — 2003.
4. *Kuzmanovic A., Knightly E. W.* Low-Rate TCP-Targeted Denial of Service Attacks. — Houston, USA: Rice University, 2003. — http://www.acm.org/sigs/sigcomm/sigcomm2003/papers/p75-kuzmanovic.pdf.
5. *McCanne S., Floyd S.* NS Network Simulator. — http://www.isi.edu/nsnam/ns/.
6. *Buchheim T.* Nam: Network Animator. — http://www.isi.edu/nsnam/nam/.
7. *Greis M.* Tutorial for the UCB/LBNL/VTNT ns2. — http://www.isi.edu/nsnam/ns/tutorial/index.html.
8. Ping Flood. — 2005. — http://www.iss.net/security_center/advice/Underground/Exploitz/Floods/Ping_Flood/default.htm.
9. JTC 017 Ping Flood (ICMP Echo) Detection // Agilent N2X. Journal of Internet Test Methodologies. — 2004. — http://advanced.comms.agilent.com/n2x/docs/journal/JTC_017.html.
10. *McCanne S., Floyd S.* Xgraph. — http://www.isi.edu/nsnam/xgraph/.
11. *McCanne S., Floyd S.* The ns manual. — http://www.isi.edu/nsnam/ns/ns-documentation.html.
12. *Baccala B.* ICMP // An Internet Encyclopedia. — http://www.ysn.ru/docs/cie/Course/Section3/15.htm.
13. ETSI, Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.2.0). — 1998. — http://www.nt.tuwien.ac.at/rapid_prototyping/teaching/lectures/sdesign/material/tr101.112_320.pdf.
14. *Crovella M., Bestavros A.* — Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. — http://www.cs.bu.edu/faculty/crovella/paper-archive/self-sim/journal-version.pdf.
15. *Baccala B.* The TCP Protocol // An Internet Encyclopedia. — http://www.ysn.ru/docs/cie/Course/Section4/index.htm.

UDC 004.942

A Dynamics of Client Pool Interaction with web-server Simulation in Computer Network Under Denial of Service Attack Conditions

D. S. Kulyabov, D. E. Laneev

Telecommunication Systems Department
Peoples' Friendship University of Russia
Miklukho-Maklaya str., 6, 117198, Moscow, Russia

By means of a simulator of networks NS2 it is modelled and investigated interactions of a web-server and a client pool in network PFUR «Donskaya». DoS-attack Ping Flood in this network is modelled and its consequences are analyzed. The estimation of opportunities of the external Internet channel of local computer network PFUR «Donskaya» is made.