*УДК 621.39, 004.7*

*Зайцев Д.А., Шинкарчук Т.Н.*
*Zaitsev D.A., Shinkarchuk T.N.*

# МОДЕЛИРОВАНИЕ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ В СИСТЕМЕ NS

# MODELING TELECOMMUNICATION NETWORKS WITH SYSTEM NS

***Аннотация***. Выполнено построение модели фрагмента Европейской магистрали Интернет в среде моделирующей системы NS. Для оценки пропускной способности магистрали, времени доставки пакета и процента доставленных пакетов разработаны соответствующие скрипты ОС Unix, выполняющие анализ полной трассы имитационного моделирования. Подтверждена адекватность ранее представленных моделей, построенных в форме раскрашенных сетей Петри.

***Summary***. The development of model for a fragment of European Internet backbone in the environment of simulation system NS was implemented. For the estimation of backbone's throughput, packet's delivery time and delivered packets percentage the corresponding scripts of OS Unix, using the analysis of complete simulation trace, were written. The adequacy of earlier represented models in the form of colored Petri nets was confirmed.

Повышение сложности современных телекоммуникационных технологий (что косвенно подтверждается увеличением объёма их стандартных спецификаций) создаёт проблему в применении аналитических методов для оценки характеристик проектируемых систем и сетей, и обуславливает широкое применение имитационного моделирования. Указанную тенденцию иллюстрирует анализ трудов известного ежегодного симпозиума IEEE MASCOTS (Моделирование, анализ и имитация компьютерных и телекоммуникационных систем) [1], более двух трети докладов которого посвящено исследованию телекоммуникационных сетей именно методами имитационного моделирования.

В последнее время широкое развитие получили специализированные системы имитационного моделирования телекоммуникационных сетей, такие как NS, GTNeS, OPNET, NISTNET, DummyNet, ModelNet, Ohio Network Emulator, ENDE, Emulab, EMPOWER, NSE, Vint/NS, NETWARS. Центральное место среди них занимает система NS (Network Simulator) [2], лидирующая по количеству применений в реальных проектах. Система NS представляет новое поколение моделирующих систем, созданных как расширение объектно-ориентированного языка программирования. Применение языка программирования позволяет описать заголовки пакетов и особенности взаимодействия компонентов с максимальной степенью детализации в соответствии со стандартами, что повышает реалистичность моделей.

Альтернативным является подход, основанный на использовании языка раскрашенных сетей Петри для представления моделей [3], успешно применённый для исследования сетей Ethernet [4], TCP/IP и MPLS [5], Bluetooth [6]. Раскрашенные сети системы CPN Tools [7] представляют собой комбинацию графа сети Петри и языка функционального программирования CPN ML. Язык ML даёт преимущества языка программирования при описании заголовков пакетов и алгоритмов их обработки, а граф сети Петри позволяет наглядно представить сложное взаимодействие компонентов.

Однако при исследовании магистральных TCP/IP и MPLS сетей в CPN Tools [5] не были выполнены сравнения полученных результатов с измерениями характеристик реальных магистралей. Вопрос измерения реальных магистралей является достаточно сложным по организационным причинам, поэтому предлагается косвенная проверка путём сравнения с результатами, полученными с помощью других систем. В качестве такой эталонной системы для сравнения выбрана NS ввиду её широкого промышленного применения и подтверждённой адекватности построенных в её среде моделей реальным телекоммуникационным процессам [1].

Целью настоящей работы является построение моделей магистральных TCP/IP сетей в системе NS и сравнительная оценка полученных характеристик с результатами моделирования в CPN Tools [5].

**1. Обзор системы NS.** Система NS [2] разработана в университете Беркли в сотрудничестве с компанией Xerox. Она реализована на языке C++ и использует язык OTcl в качестве командного и конфигурационного интерфейса. Для описания моделей создан текстовый язык, являющийся расширением языка OTcl. Модель представляет собой файл с расширением .tcl, который затем подаётся на вход симулятора ns. Для наглядного представления процессов моделирования используется аниматор nam (Network Animator), кроме того, графическое представление характеристик моделей может быть получено в программе xgraph.

Для описания топологии сети NS предлагает две основных концепции: узел (node) и связь (link). Узлы и связи имеют множество атрибутов, позволяющих описывать реальное оборудование. Для создания узла используется команда:

```
$ns node
```

Созданные узлы нумеруются моделирующей системой последовательно с нуля, кроме того, им могут быть присвоены имена, например, имена T1 и T2:

```
set T1 [$ns node]
set T2 [$ns node]
```

Для задания атрибутов узлов служит команда

```
$ns node-config <attribute> <value>
```

Так атрибут – addressType задаёт тип адресации узлов и вместо стандартной flat может быть выбрана иерархическая структура адреса hierarchical с указанием количества уровней иерархии и количества бит для каждого уровня. Так, например, структура стандартного IP-адреса может быть описана как

```
$ns node-config –addressType hierarchical 4 8 8 8 8
```

Связи между узлами могут быть симплексными, дуплексными и имеют такие атрибуты, как пропускная способность, задержка, тип очереди и другие. Следующая команда описывает дуплексную связь между узлами $R1 и $R2 с пропускной способностью 1 Mbps, задержкой 10 ms и очередью с потерями при переполнении:

```
$ns duplex-link $R1 $R2 1Mb 10ms DropTail
```

Для описания трафика сети вводится концепция агентов (agent). Отдельными агентами описываются источники и стоки трафика. Агенты присоединяются к узлам. Направление трафика задаётся путем логического соединения источника со стоком:

```
set tcp [new Agent/TCP]          #создать новый TCP источник
set sink [new Agent/TCPSink]     #создать новый TCP сток
$ns attach-agent $T1 $tcp  #присоединить агент tcp к узлу T1
$ns attach-agent $T2 $sink #присоединить агент sink к узлу T2
$ns connect $tcp $sink           #соединить источник tcp со стоком sink
```

Конкретный трафик генерируется протоколами прикладного уровня, такими как Telnet, HTTP, FTP, например:

```
set ftp [new Application/FTP]    #создать генератор FTP трафика
$ftp attach-agent $tcp           #присоединить генератор ftp к агенту tcp
```

Возможно моделирование случайного трафика с различными законами распределения. Предоставляются атрибуты для задания размера пакетов и интенсивности трафика, например:

```
$ftp set packetSize_ 1000  #задать размер пакета в 1000 байт
$ftp set interval_ 0.02          #задать интервал между пакетами в 0.02 с
```

Работа симулятора во времени управляется командами запуска и завершения генерации трафика:

```
$ns at 1.25 "start $ftp"    #запустить генератор ftp в момент времени 1.25 с
$ns at 4.67 "stop $ftp"     #остановить генератор ftp в момент времени 4.67 с
```

Для исследования характеристик модели следует либо сохранить полную трассу процесса моделирования для последующего анализа, либо дополнить модель пользовательскими процедурами, выполняющими сбор и обработку требуемой статистической информации. Сохранение полной трассы в файле my-tracefile.out выполняется командой:

```
$ns trace-all [open my-tracefile.out w]
```

Простейшие модели телекоммуникационных сетей могут быть описаны указанными командами, однако, для построения сложных моделей требуется написание процедур на языке Tcl, например, для автоматической генерации сложной топологии и трафика, а также оценки характеристик моделей в процессе имитации.

В настоящее время система NS содержит библиотеки классов для моделирования сетей TCP/IP, MPLS, локальных, мобильных, спутниковых и радио сетей. Кроме того, создано множество классов для исследования маршрутизации, так для TCP/IP сетей возможно моделирование статической и динамической маршрутизации. При необходимости моделирования новых протоколов либо оборудования требуется разработать соответствующую библиотеку классов на языке C++ и интегрировать её в среду NS.

**2. Описание топологии магистральной сети.** В настоящей работе построена модель магистральной TCP/IP сети, исследованной в [5]. Структурная схема сети представлена на Рис. 1. Сеть состоит из 6 терминальных подсетей T1-T6, территориально расположенных в указанных странах, и магистральной сети, образованной семью маршрутизаторами R1-R7.

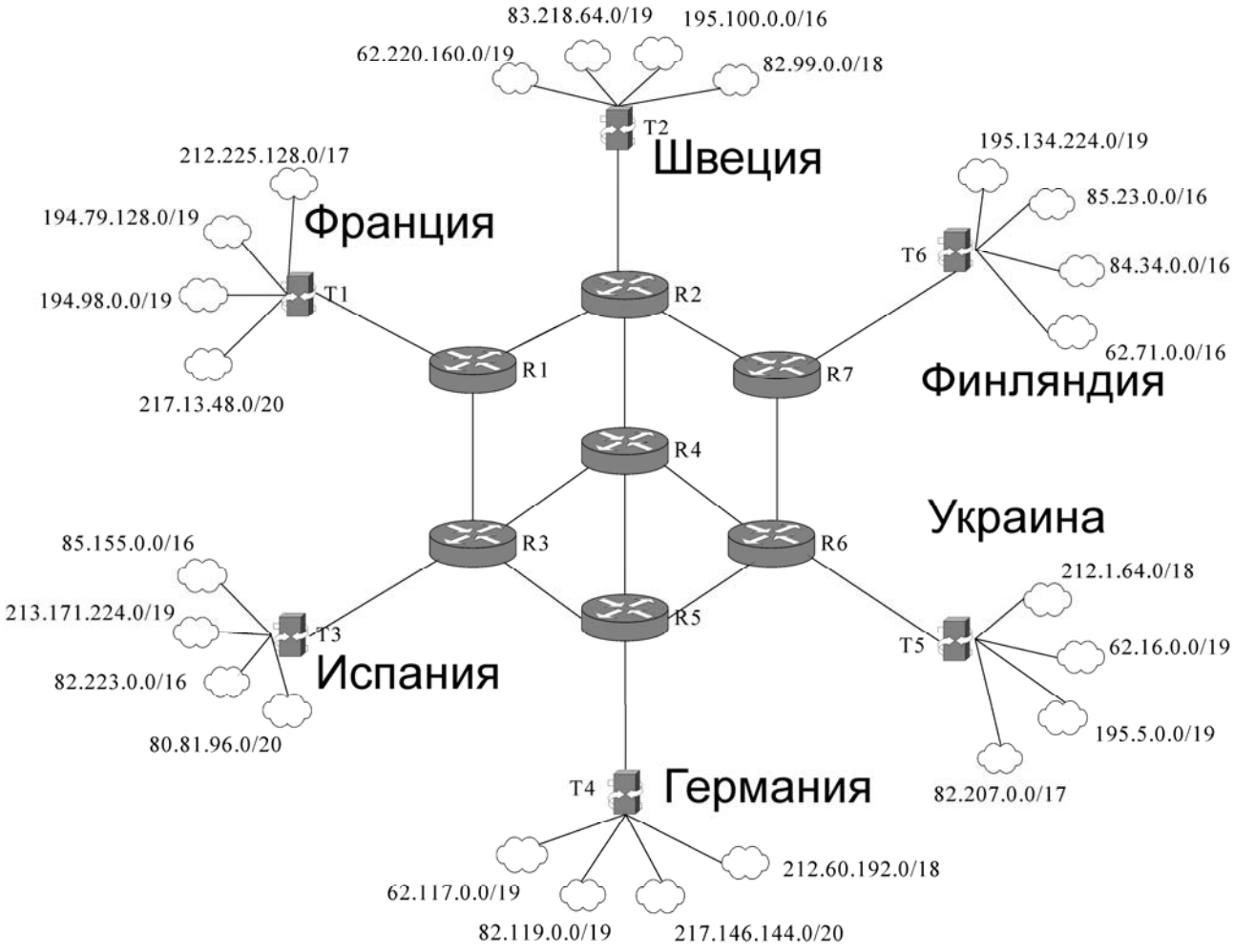



Рисунок 1 – Структурная схема магистральной сети

При построении модели структура IP-адреса не рассматривалась, поэтому использована простая схема адресов plain. Кроме того, применена, как и в работе [5], статическая маршрутизация. Топология сети описана последовательностью tcl команд, представленной на рис. 2.

```
#Создать узлы

set R1 [$ns node]
set R2 [$ns node]
set R3 [$ns node]
set R4 [$ns node]
set R5 [$ns node]
set R6 [$ns node]
set R7 [$ns node]

set T1 [$ns node]
set T2 [$ns node]
set T3 [$ns node]
set T4 [$ns node]
set T5 [$ns node]
set T6 [$ns node]
```

```
#Создать связи между узлами

$ns duplex-link $R1 $R2 1Mb 10ms DropTail
$ns duplex-link $R2 $R7 1Mb 10ms DropTail
$ns duplex-link $R7 $R6 1Mb 10ms DropTail
$ns duplex-link $R6 $R5 1Mb 10ms DropTail
$ns duplex-link $R5 $R3 1Mb 10ms DropTail
$ns duplex-link $R3 $R1 1Mb 10ms DropTail
$ns duplex-link $R4 $R2 1Mb 10ms DropTail
$ns duplex-link $R4 $R3 1Mb 10ms DropTail
$ns duplex-link $R4 $R5 1Mb 10ms DropTail
$ns duplex-link $R4 $R6 1Mb 10ms DropTail

$ns duplex-link $T1 $R1 1Mb 10ms DropTail
$ns duplex-link $T2 $R2 1Mb 10ms DropTail
$ns duplex-link $T6 $R7 1Mb 10ms DropTail
$ns duplex-link $T5 $R6 1Mb 10ms DropTail
$ns duplex-link $T4 $R5 1Mb 10ms DropTail
$ns duplex-link $T3 $R3 1Mb 10ms DropTail
```

Рисунок 2 – Описание топологии сети

Заметим, что в системе NS временные характеристики задаются для связей; стандартное множество атрибутов узла не содержит временную задержку. Для этих целей необходимо создание специальных процедур-агентов узла. Поэтому в настоящей модели времена обработки пакетов маршрутизаторами вынесены на инцидентные связи. Заметим, что время передачи пакета вычисляется системой NS по следующей формуле:

$$s / b + d,$$

где s – размер пакета в битах, b – скорость связи в бит/с, d – задержка связи. Указанная задержка связи d и используется для представления времени обработки пакета маршрутизатором.

Кроме того, для наглядного графического отображения процессов передачи пакетов в Nam задано графическое отображение очередей на линиях связи магистральной сети, представленное на рис. 3.

```
#Включить мониторинг очередей маршрутизаторов
$ns duplex-link-op $R1 $R2 queuePos 0.5
$ns duplex-link-op $R2 $R7 queuePos 0.5
$ns duplex-link-op $R7 $R6 queuePos 0.5
$ns duplex-link-op $R6 $R5 queuePos 0.5
$ns duplex-link-op $R5 $R3 queuePos 0.5
$ns duplex-link-op $R3 $R1 queuePos 0.5
$ns duplex-link-op $R4 $R2 queuePos 0.5
$ns duplex-link-op $R4 $R3 queuePos 0.5
$ns duplex-link-op $R4 $R5 queuePos 0.5
$ns duplex-link-op $R4 $R6 queuePos 0.5
```

Рисунок 3 – Мониторинг очередей магистральной сети

**3. Описание трафика.** Трафик сети описан в соответствии с [5]. Каждая терминальная сеть генерирует потоки к каждой другой терминальной сети. Поскольку взаимодействие клиент-сервер [4] в [5] не рассматривалось, для генерации потокового трафика выбран агент UPD. На рис. 4 представ-

лен фрагмент агентов трафика и их соединений для терминальной сети T2; для других терминальных сетей фрагменты имеют аналогичную форму. Классы трафика (class) выбраны для отображения различными цветами пакетов различных терминальных сетей в nam.

```
set udp2_1 [new Agent/UDP]
$udp2_1 set class_ 2
$ns attach-agent $T2 $udp2_1

set udp2_3 [new Agent/UDP]
$udp2_3 set class_ 2
$ns attach-agent $T2 $udp2_3

set udp2_4 [new Agent/UDP]
$udp2_4 set class_ 2
$ns attach-agent $T2 $udp2_4

set udp2_5 [new Agent/UDP]
$udp2_5 set class_ 2
$ns attach-agent $T2 $udp2_5

set udp2_6 [new Agent/UDP]
$udp2_6 set class_ 2
$ns attach-agent $T2 $udp2_6
```

```
set null2_1 [new Agent/Null]
$ns attach-agent $T2 $null2_1

set null2_3 [new Agent/Null]
$ns attach-agent $T2 $null2_3

set null2_4 [new Agent/Null]
$ns attach-agent $T2 $null2_4

set null2_5 [new Agent/Null]
$ns attach-agent $T2 $null2_5

set null2_6 [new Agent/Null]
$ns attach-agent $T2 $null2_6

$ns connect $udp2_1 $null1_2
$ns connect $udp2_3 $null3_2
$ns connect $udp2_4 $null4_2
$ns connect $udp2_5 $null5_2
$ns connect $udp2_6 $null6_2
```

Рисунок 4 – Агенты доставки трафика (для терминальной сети T2)

Для генерации трафика использовано случайное экспоненциально распределённое время и фиксированная длина пакетов, равная 500 байтов. Пример генератора для пары узлов T2 и T4 изображён на рис. 5.

```
set exp2_4 [new Application/Traffic/Exponential]
$exp2_4 set packetSize_ 500
$exp2_4 set burst_time_ 400ms
$exp2_4 set idle_time_  800ms
$exp2_4 set rate_           100K
$exp2_4 attach-agent $udp2_4
```

Рисунок 5 – Генератор трафика (для пары узлов T2 и T4)

Генератор exp2_4 включается на случайный экспоненциально распределённый интервал времени со средним burst_time_ и выключается на случайный экспоненциально распределённый интервал времени со средним idle_time_; во включенном состоянии он генерирует пакеты с постоянной битовой скоростью rate_. При исследовании модели использован последовательный запуск и остановка генераторов трафика. На рис. 6 приведен пример для терминальной подсети T2.

```
$ns at 0.40 "$cbr2_1 start"
$ns at 0.44 "$cbr2_3 start"
$ns at 0.48 "$cbr2_4 start"
$ns at 0.52 "$cbr2_5 start"
$ns at 0.56 "$cbr2_6 start"
```

```
$ns at 4.20 "$cbr2_1 stop"
$ns at 4.24 "$cbr2_3 stop"
$ns at 4.28 "$cbr2_4 stop"
$ns at 4.32 "$cbr2_5 stop"
$ns at 4.36 "$cbr2_6 stop"
```

Рисунок 6 – Запуск и остановка генераторов трафика (для терминальной сети T2)

Следует отметить, что использование процедур языка tcl с операторами цикла для выбора пар терминальных сетей позволяет значительно сократить размер модели.

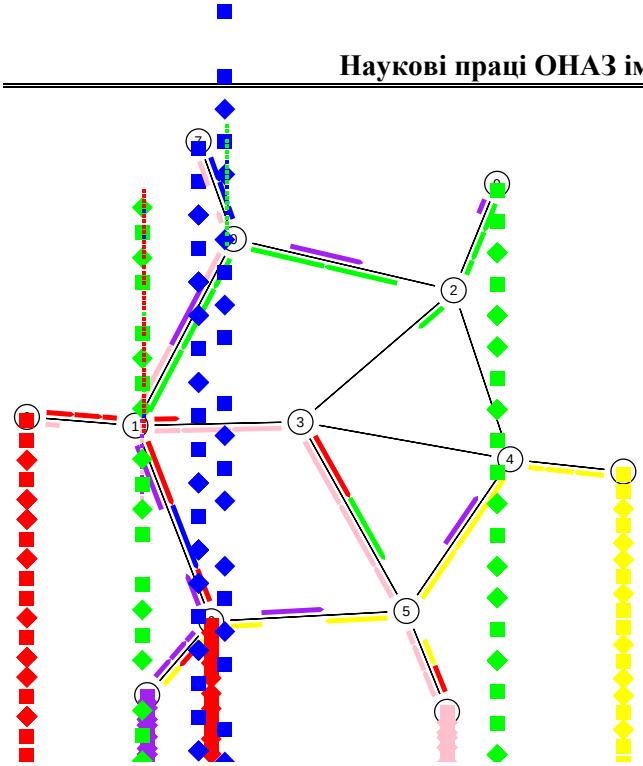



Рисунок 7 – Пример анимации динамики сети

**4. Оценка характеристик модели.** Поведение построенной модели наблюдалось в аниматоре Nam, графически отображающем динамику передачи пакетов и состояние очередей. Пример образа экрана nam приведен на рис. 7. Заметим, что nam использует номера узлов в соответствии с их порядком появления в описаниях. Анимация отображает графически потоки передаваемых пакетов вдоль связей, а также возникающие очереди (последовательность квадратов и ромбов соответствующего цвета), что позволяет оценить узкие места в сети визуально.

Для оценки числовых характеристик пропускной способности и качества обслуживания возможно два подхода:

– сохранение и последующий анализ полной трассы симулятора;

– создание специальных tcl процедур для вычисления и отображения характеристик.

В настоящей работе применён первый из указанных подходов. Запись файла трассы симулятора имеет вид:

```
+ 0.671379 7 0 exp 500 ------- 1 7.3 11.5 3 12
```

где в первой колонке указан знак операции: «+» – постановка в очередь, «-» – выборка из очереди, «r» – получение пакета, «d» – потеря пакета; во второй колонке указано время в секундах, третья и четвёртая колонки отображают начальный и конечный узлы связи; далее следует тип пакета, его длина, флаги протокола и идентификатор потока; поля отправителя и получателя заголовка пакета; номер последовательности; уникальный идентификатор пакета. Используя уникальный идентификатор пакета можно легко выделить трассу пакета из файла, например, трассу 12 пакета с помощью команды Unix:

```
grep " 12$" my-tracefile.out
```

Полученный результат представлен на рис. 8. Пакет 12 отправлен узлом 7 (T1) узлу 11 (T5). На трассе прослеживаются события проследования очередей и получения пакета промежуточными узлами 0 (R1), 1 (R2), 3 (R4), 5 (R6). Время доставки пакета может быть вычислено как разность времен получения конечным узлом и отправления начальным: 0,741379 – 0,671379.

```
+ 0.671379 7 0 exp 500 ------- 1 7.3 11.5 3 12
- 0.671379 7 0 exp 500 ------- 1 7.3 11.5 3 12
r 0.685379 7 0 exp 500 ------- 1 7.3 11.5 3 12
+ 0.685379 0 1 exp 500 ------- 1 7.3 11.5 3 12
- 0.685379 0 1 exp 500 ------- 1 7.3 11.5 3 12
r 0.699379 0 1 exp 500 ------- 1 7.3 11.5 3 12
+ 0.699379 1 3 exp 500 ------- 1 7.3 11.5 3 12
- 0.699379 1 3 exp 500 ------- 1 7.3 11.5 3 12
r 0.713379 1 3 exp 500 ------- 1 7.3 11.5 3 12
+ 0.713379 3 5 exp 500 ------- 1 7.3 11.5 3 12
- 0.713379 3 5 exp 500 ------- 1 7.3 11.5 3 12
r 0.727379 3 5 exp 500 ------- 1 7.3 11.5 3 12
+ 0.727379 5 11 exp 500 ------- 1 7.3 11.5 3 12
- 0.727379 5 11 exp 500 ------- 1 7.3 11.5 3 12
r 0.741379 5 11 exp 500 ------- 1 7.3 11.5 3 12
```

Рисунок 8 – Пример трассы пакета

В настоящей работе предложен простой способ оценки характеристик модели на основе анализа файла трассы с помощью командных файлов ОС Unix, не требующий использования специальных классов системы NS. На рис. 9 и 10 представлены командные файлы (скрипты) ОС Unix для вычисления средней пропускной способности и времени доставки пакета соответственно.

```
t0=`cut -f2 -d " " my-tracefile.out|sort -n|head -1`  #начальное время
tl=`cut -f2 -d " " my-tracefile.out|sort -n|tail -1`  #конечное время
D=`rexpr $tl - $t0`                                   #интервал времени
S=0
for i in 7 8 9 10 11 12; do  #цикл по терминальным сетям
  for p in `grep "^r [0123456789.]* [0123456789]* $i " my-tracefile.out|cut -f6
-d " "`; do
    S=`expr $S + $p`         #суммирование длин полученных пакетов
  done
done
T=`expr $S \* 8`
T=`rexpr $T / $D`
echo "Traffic: $T bps"
```

Рисунок 9 – Скрипт вычисления пропускной способности (трафика)

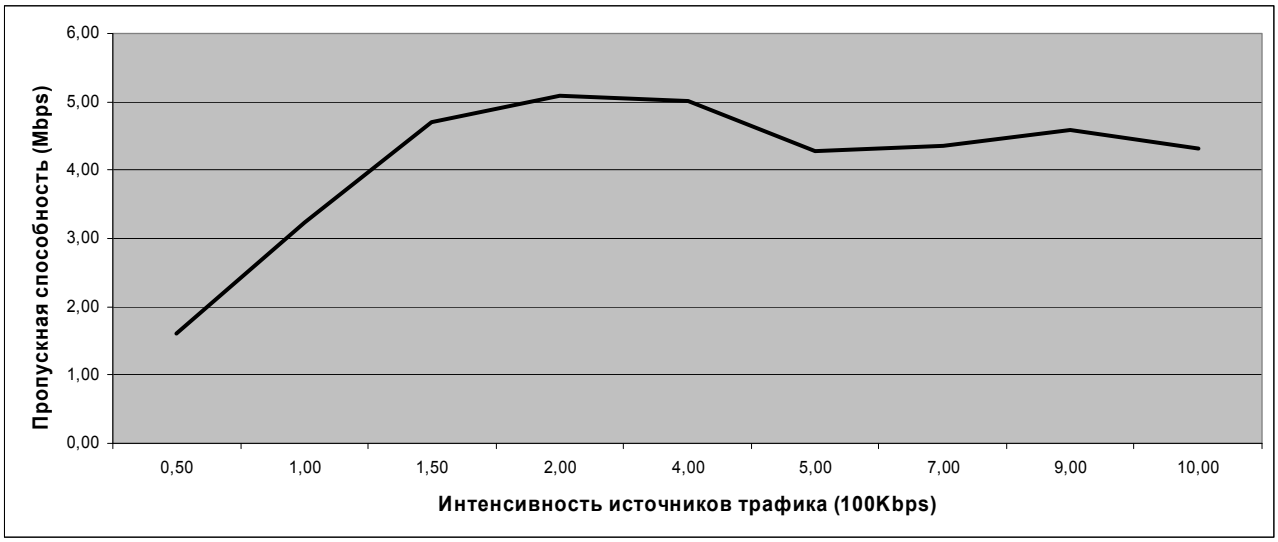
Переменная i принимает значения номеров узлов терминальных сетей; доставка пакета фиксируется по его получению узлом назначения; количество полученных пакетов накапливается в переменной S. Команда rexpr является модификацией expr для вычисления выражений в действительных числах.

```
m=`cut -f12 -d " " my-tracefile.out|sort -n|tail -1` #максимальный номер пакета
m1=`expr $m + 1`
i=0; j=0; s=0
while [ $i -le $m ]  #цикл по всем пакетам
do
  if [ `grep " $i$" my-tracefile.out|tail -1|cut -f1 -d " "` != "d" ]
  then                    #не потерянный пакет
    t1=`grep " $i$" my-tracefile.out|head -1|cut -f2 -d " "`
    t2=`grep " $i$" my-tracefile.out|tail -1|cut -f2 -d " "`
    j=`expr $j + 1`
    dt=`rexpr $t2 - $t1` #время доставки пакета
    s=`rexpr $s + $dt`
  fi
  i=`expr $i + 1`
done
adt=`rexpr $s / $j`
dl=`rexpr $j / $m1`
echo "Average delivery time: $adt s. Delivered percent: $dl"
```

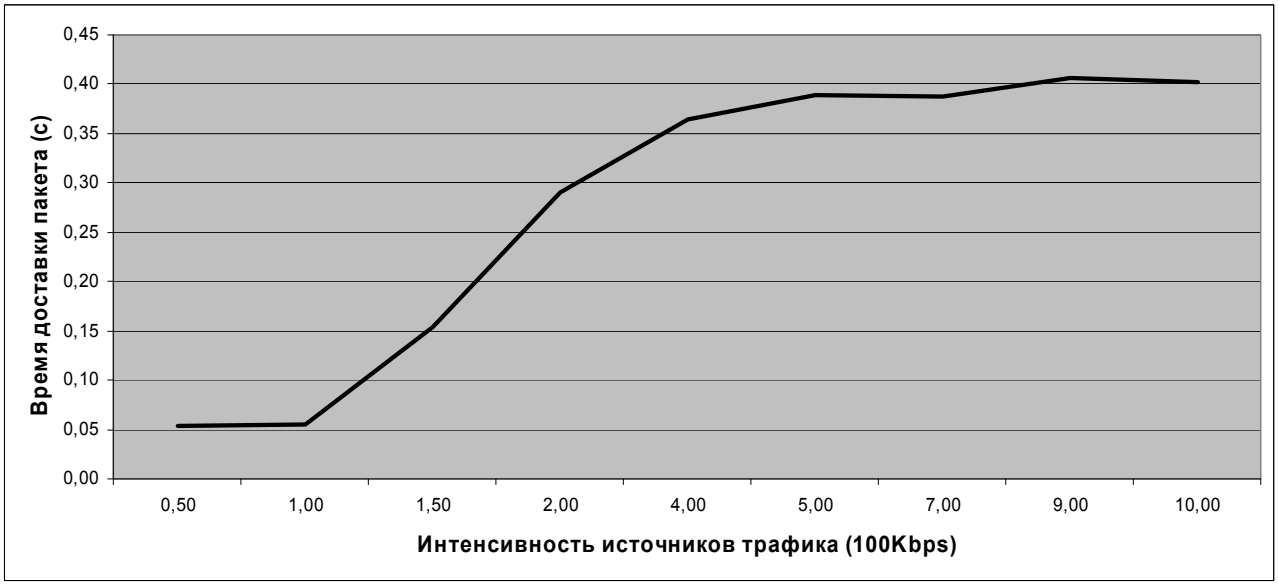
Рисунок 10 – Скрипт вычисления времени доставки

Создание специальных tcl процедур для накопления требуемой статистической информации позволяет выполнять вычисление характеристик «на лету» в процессе имитации, а также получать представление динамики характеристик во времени с помощью программы xgraph. Однако для разработки таких процедур необходимо углублённое изучение системы NS и библиотек языков tcl и C++.

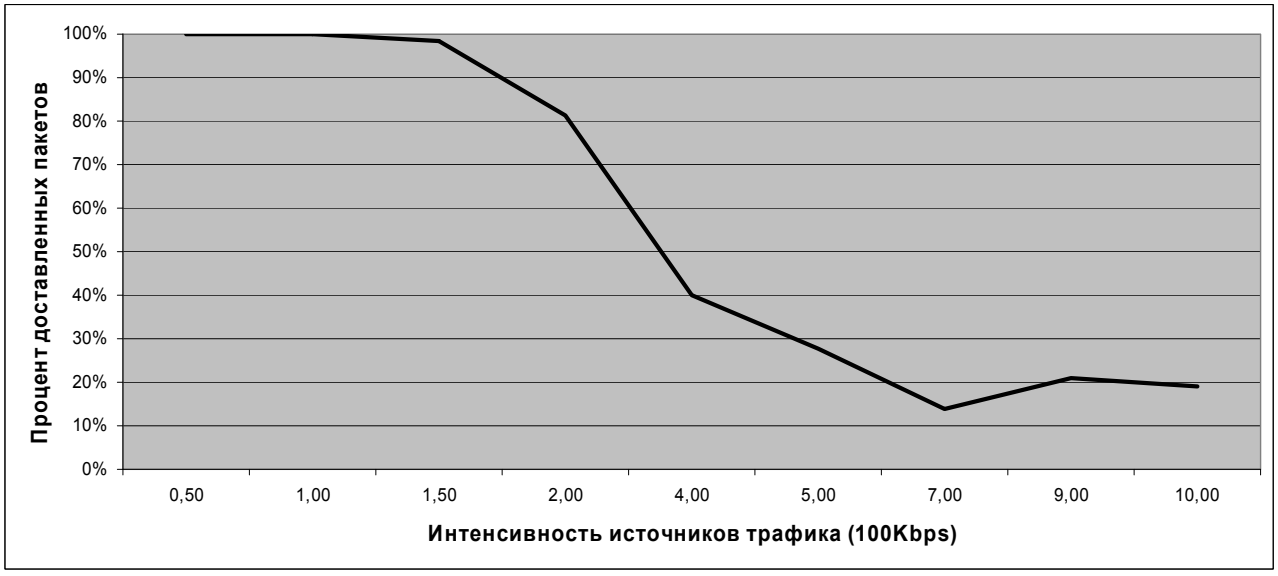
**5. Анализ результатов моделирования.** Выполнено имитационное моделирование процессов в указанном фрагменте Европейской магистрали Интернет (рис. 1) для различной интенсивности потоков, генерируемых терминальными сетями. Оценивались пропускная способность магистрали, среднее время доставки пакета и процент доставленных пакетов по трассе моделирования с помощью скриптов, представленных на Рис. 8–10. Полученные результаты представлены на рис. 11.




а) пропускная способность сети




б) среднее время доставки пакета




в) процент доставленных пакетов

Рисунок 11 – Результаты оценок характеристик сети

Поведение сети исследовалось при пиковых нагрузках. Интенсивность источников трафика до 150 Kbps соответствует нормальной нагрузке сети с долей потерянных пакетов не превышающей 2-3% и временем доставки пакета 0,05-0,15 с.; при этом пропускная способность всей сети примерно равняется суммарному сгенерированному трафику. Дальнейший рост интенсивности трафика приводит к снижению пропускной способности из-за роста очередей и потери пакетов и значительному увеличению времени доставки пакета (до 0,4 с).

Выполнен пересчёт временных задержек переходов для моделей, ранее построенных в форме раскрашенных сетей Петри [5] в системе CPN Tools, в пропускные способности и задержки каналов системы NS. Отличия полученных оценок пропускной способности и времени доставки пакета составляют не более четырёх процентов, что является косвенным подтверждением адекватности моделей, построенных в CPN Tools, реальным процессам в магистральных сетях.

Таким образом, в настоящей работе выполнено построение модели фрагмента Европейской магистрали Интернет в среде моделирующей системы NS. Для оценки пропускной способности магистрали, времени доставки пакета и процента доставленных пакетов разработаны соответствующие скрипты ОС Unix, выполняющие анализ полной трассы имитационного моделирования. Подтверждена адекватность ранее представленных моделей, построенных в форме раскрашенных сетей Петри.

***Литература***


1. *Proceedings* of 12th Annual Meeting of the IEEE / ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, October 5-7, 2004 (MASCOTS 2004). – Volendam (Netherlands). – 2004. – 614 p.
2. *The ns Manual* / Ed.: Kevin Fall, Kannan Varadhan. – The VINT Project, 2006. – 414 p.
3. *Jensen K.* Colored Petri Nets - Basic Concepts, Analysis Methods and Practical Use. – Springer-Verlag, 1997. – Vol. 1-3. – 673 p.
4. *Зайцев Д.А., Шмелёва Т.Р.* Моделирование коммутируемой локальной сети раскрашенными сетями Петри // Зв'язок. – 2004. – Т. 46 – № 2. –С. 56-60.
5. *Зайцев Д.А., Сакун А.Л.* Исследование эффективности технологии MPLS с помощью раскрашенных сетей Петри // Зв'язок. – 2006. – Т. 65. – №5. – С. 49-55.
6. *Зайцев Д.А., Березнюк М.В.* Исследование эффективности использования адресного пространства протокола Bluetooth // Радиоэлектроника. Информатика. Управление. – 2006. – №1. – С. 57-63.
7. *Albert K., Jensen K., Shapiro R.* Design/CPN: A Tool Package Supporting the Use of Colored Nets // Petri Net Newsletter. – April 1989. – P. 22-35.