

ОБЕСПЕЧЕНИЕ ПОТРЕБНЫХ НАГРУЗОК СЕТЕВЫХ ИНТЕРФЕЙСОВ УТИЛИТОЙ PING ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРОТОКОЛА ICMP

М.К. Бойченко

noc@bmstu.ru

И.П. Иванов

ivanov@bmstu.ru

А.Ю. Кондрагъев

В.А. Лохтуров

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Самоподобные свойства трафика в сетях с технологией Ethernet проявляются при высоких коэффициентах использования интерфейсов транзитных и оконечных узлов. Поэтому в экспериментальных исследованиях разработчикам, интеграторам и администраторам компьютерных сетей и их транспортных систем необходимо иметь инструмент регулирования нагрузки на сетевые интерфейсы, альтернативный дорогостоящим генераторам трафика. Предложена технология обеспечения требуемой нагрузки с использованием повсеместно распространенной утилиты *ping* программного обеспечения протокола ICMP или ее модернизированного варианта *nanoping*. Работоспособность метода подтверждается приводимыми результатами экспериментальных исследований

Ключевые слова

Компьютерная сеть, сетевой интерфейс, пропускная способность, трафик, нагрузка, Ethernet, самоподобие

Поступила в редакцию 29.12.2015

© МГТУ им. Н.Э. Баумана, 2016

Введение. Пульсация сетевого компьютерного трафика является общепризнанным фактом [1–3]. В сетях технологии Ethernet интенсивность потоков кадров меняется десятки и сотни раз, при этом нагрузка на интерфейсы оконечных и транзитных сетевых узлов может превышать 100 %. В [1] отмечено, что при высоких нагрузках (коэффициентах использования) самоподобный характер трафика проявляется в существенном различии необходимых объемов буферной памяти интерфейсов сетевых узлов, переполнение которых чревато потерей информации и резким снижением производительности компьютерной сети. Отсутствие достаточно простых математических моделей, адекватных реально протекающим в интерфейсах процессам обработки информации, является серьезным негативным фактором для обоснованного проектирования буферов сетевых интерфейсов разработчиками сетевого оборудования, а также для оптимального выбора моделей транзитных узлов транспортной системы будущей компьютерной сети сетевыми интеграторами. Положение усугубляется сокрытием фирмами–производителями результатов экспериментальных исследова-

ний функционирования конкретных коммутаторов и маршрутизаторов в таких условиях их работы, при которых нагрузка на интерфейсы близка или превышает 100 % [1]. Наиболее рациональное решение в данных обстоятельствах — это предоставление сетевым администраторам возможности самостоятельно экспериментально исследовать функционирование сетевых интерфейсов при их высоких нагрузках. Предлагаемые на рынке программные пакеты генерации потоков информации достаточно дороги и практически недоступны для администраторов компьютерных сетей малобюджетных организаций, поэтому весьма целесообразно распространение технологий и программных пакетов, базирующихся на уже существующем, свободно распространяемом, сетевом программном обеспечении.

Разработка инструментария для проведения исследований. В МГТУ им. Н.Э. Баумана за основу подобных разработок было принято программное обеспечение протокола Internet Control Message Protocol (ICMP). Модернизация утилиты *ping* этого программного обеспечения позволила провести ряд успешных экспериментальных исследований не только в сетях Ethernet, но и в сетях Fast Ethernet и Gigabit Ethernet [4, 5]. Далее приведены результаты применения этой же утилиты (а точнее ее аналога *nanoping*) для создания больших нагрузок на любом интерфейсе узла компьютерной сети с заданным DNS-именем или IP-адресом, обеспечиваемых одним из хостов сети.

Идея заключается в организации генерации из одного хоста нескольких потоков эхо-запросов с кадрами протокола ICMP на исследуемый интерфейс. При этом увеличивается нагрузка не только на адресуемый интерфейс, но и на все промежуточные интерфейсы сетевых узлов, находящиеся на маршруте от хоста (источника эхо-запроса) до хоста (приемника), формирующего на сетевом уровне эталонной модели ISO/OSI эхо-ответ на пришедший эхо-запрос. Ограничение сетевым уровнем — несомненное достоинство предлагаемой технологии экспериментальных исследований, так как не затрагивается прикладной уровень и наносится минимально возможный ущерб производительности вычислительного процесса, запускаемого на хосте (приемнике) эхо-запроса. Особенностью утилиты *ping* протокола ICMP является ожидание эхо-ответа на каждый эхо-запрос с замером времени Round Trip Time (RTT), поэтому нагрузкой интерфейса с помощью одного потока эхо-запросов можно управлять лишь в весьма ограниченном диапазоне $0 \leq \rho \leq 0,5$. Это объясняется тем, что собственно нагружаемый интерфейс задействуется лишь в течение времени прохождения через него кадра эхо-запроса. При пропускной способности $R = 10$ Мбит/с время прохождения фрейма длиной L_f байт составляет $\tau_{10} = 800L_f$ нс. Такое же время затрачивается интерфейсом на пропускание через себя эхо-ответа. Следовательно, минимальное время для расчета RTT не может быть меньше двух значений τ_{10} и нагрузка не может превышать 50 %. В действительности в RTT включаются составляющие времени, которое тратится операционной системой на формирование эхо-запроса, передачу его из оперативной памяти компьютера-источника в буфер сетевого адаптера, на формирование операционной системой компьютера-приемника эхо-ответа и т. д. Именно

поэтому $\rho \leq 0,5$. Если к тому же формирование серий посылок эхо-запросов осуществляется не с опцией «-f», а с опцией «-i» (указывающей период следования эхо-запросов), то нагрузка еще меньше, так как часть периода компьютер-источник будет простаивать. Период следования эхо-запросов по умолчанию не должен быть меньше 1 мс, поскольку в утилите *ping* (*nanoping*) протокола ICMP ограничена интенсивность ответов на эхо-запросы количеством 1000 с^{-1} по запросам со всех хостов сети. Без принятия надлежащих мер в зондируемом хосте даже при пропускной способности его сетевого адаптера в 10 Мбит/с получить высокий коэффициент использования одним потоком ICMP-пакетов невозможно.

Методика проведения экспериментальных исследований. Реализация задачи одновременного запуска нескольких потоков на множестве узлов сети с общим для нескольких экземпляров набором параметров представлена на рис. 1.

```
# cat example.task
node1:node2eth1:1:|1|:-w 300 -f -q -s 64
node1:node2eth1:16:|1|8|16|:-w 300 -f -q -s 1472
node3:node4eth2:8:|2|8|:-n 100000 -f -q -s 64

# cat runtask.sh
ping_cmd=nanoping

while read flownum src dst count print flow ping arg
do
    while [ "${count}" -gt "0" ]
    do
        if echo ${print flow} | grep -q \|${count}\|
        then
            #     ech runtask2/${1}.${flownum}.${dst}.${count}.res
            ssh $src "(sleep 2; $ping_cmd $ping_arg $dst 2>&1
            >runtask2/${1}.${flownum}.${dst}.${count}.res) &"&
            else
            #     echo ${count}
            ssh $src "(sleep 2;
            $ping_cmd $ping_arg $dst >/dev/null 2>&1) &"& fi
            count=$((count - 1))
        done
    done < $fname

echo begin run
sleep $w; sleep 4
echo end run
```

Рис. 1. Скрипт запуска множества потоков эхо-запросов на узлах компьютерной сети

Скрипт написан на языке командного интерпретатора Unix. Для указания, на каких узлах сети и какое количество потоков эхо-запросов с определенным набором параметров запускается, используется файл описания задания (*example.task*). Каждая строка последовательно (через двоеточие) содержит:

– порядковый номер для дополнительной идентификации потоков эхо-запросов;

- имя узла сети, на котором запускаются потоки идентичных эхо-запросов;
- имя узла сети, на который отправляются эхо-запросы с указанием интерфейса этого узла в случае использования на адресуемом узле нескольких сетевых адаптеров;
- количество потоков эхо-запросов;
- номера потоков для последующего вывода в файл результатов статистической обработки этих потоков эхо-запросов;
- опции каждого эхо-запроса, идентичные для всех потоков рассматриваемой строки задания.

Файл описания задания обрабатывается на центральном узле сети скриптом *runtask.sh* путем указания его в качестве аргумента. Сначала из файла описания выбирается информация об аргументах утилиты *ping (nanoping)*. Затем в цикле перебираются имена хостов, на которых с помощью механизма удаленного выполнения команд запускается нужное количество экземпляров процессов формирования и отправки в сеть эхо-запросов. В качестве механизма удаленного выполнения команд можно использовать протокол Remote Shell (RSH) или современную безопасную реализацию Secure Shell (SSH) с применением аутентификации по публичным ключам, для чего на всех указываемых в задании хостах (источниках) должны функционировать соответствующие процессы протоколов. В приведенном примере запускается один поток с узла сети *node1* на интерфейс *eth1* узла *node2* с опциями «*-w 300*» (время выполнения процесса 300 с); «*-i 0,002*» (период следования каждого эхо-запроса 2 мс); «*-q*» (подавление вывода результатов каждого эхо-запроса на монитор); «*-s 18*» (размер формируемого эхо-запроса протокола ICMP принимается равным 18 байтам, что приводит к формированию минимально возможного Ethernet-кадра в 46 байтов). Одновременно на этом же узле сети организуется формирование шестнадцати потоков эхо-запросов с одинаковыми опциями: «*-w 300*» (время выполнения процесса 300 с); «*-f*» (организация потоковой отправки эхо-запросов в каждом потоке с минимально возможным межкадровым интервалом между ними); «*-q*» (подавление вывода результатов каждого эхо-запроса на монитор); «*-s 1472*» (размер каждого эхо-запроса протокола ICMP принимается равным 1472 байтам, что обеспечивает формирование Ethernet-кадра с максимально возможной полезной нагрузкой в 1500 байтов). Все шестнадцать формируемых потоков эхо-запросов направляются на тот же, что и ранее, интерфейс *eth1* того же узла *node2*. В файл результатов выводятся статистические данные по первому, восьмому и шестнадцатому потокам. В третьей строке файла-задания на узле сети *node3* — организация восьми процессов генерации потоков эхо-запросов с опциями: «*-n 100 000*» (количество посылок эхо-запросов в каждом потоке 1 000 00); «*-f*» (организация потоковой отсылки); «*-q*» (подавление вывода на монитор); «*-s 64*» (размер ICMP кадра эхо-запроса 64 байта). Все потоки направляются на интерфейс *eth2* узла *node4*. Результаты статистической обработки сохраняются для второго и восьмого потоков.

При формировании операционной системой компьютер-источник нескольких потоков эхо-запросов нагрузка на его сетевой интерфейс (и все последующие

на маршруте) возрастает. При этом линейная зависимость нагрузки от количества посылаемых потоков эхо-запросов наблюдается до определенного предела. Само время формирования эхо-запросов линейно зависит от количества потоков. Если суммарное время передачи сетевым адаптером компьютера (источника) этого набора эхо-запросов меньше времени формирования, то нагрузка в соответствующем сегменте транспортной системы линейно растет. Однако, как правило, сетевой адаптер в рассматриваемом процессе передачи информации в сеть является «узким местом» с точки зрения производительности. Достаточно длинные кадры, передаваемые сетевым адаптером побитно в кабельный сегмент, к моменту формирования операционной системой следующего эхо-запроса из набора экземпляров утилиты *ping* (*nanoping*) еще присутствуют в аппаратном буфере адаптера, что приводит к образованию очереди. Пребывание в очереди увеличивает значение RTT для каждого эхо-запроса, снижая при этом нагрузку от каждого потока кадров и суммарную нагрузку в соответствующем сегменте транспортной системы. Образование очередей в буферной памяти интерфейсов при росте нагрузки процесс закономерный для систем массового обслуживания [6–8], поэтому очевидна постановка вопроса об их переполнении в определенных условиях. Однако к потере кадров в системе компьютер–источник эта ситуация не приводит, так как сигнал о возможности переполнения буферной памяти сетевого адаптера отслеживается операционной системой, которая «тормозит» процессы генерации очередного эхо-запроса и его передачи в адаптер (например, организацией петлевого прохождения непомяющегося в буферную память кадра с фиксацией этого факта количеством *pipe* в выходных результатах утилиты *ping*). Своеобразная обратная связь сетевого интерфейса компьютера–источника с его операционной системой обеспечивает защиту процесса генерации эхо-запросов от возможных потерь информации «внутри» компьютера из-за переполнения буферной памяти.

Проведение экспериментальных исследований. При планировании экспериментов по обеспечению высокой нагрузки интерфейсов была использована экспериментальная сеть, разработанная в МГТУ им. Н.Э. Баумана для проведения целого ряда исследований. Сам эксперимент заключался в посыле ряда потоков эхо-запросов с одного узла сети на другой. Для исключения влияния промежуточных транзитных узлов компьютер–источник эхо-запросов соединялся с компьютером–приемником кроссовым кабелем. Для уменьшения количества генерируемых потоков пропускные способности интерфейсов обоих компьютеров принудительно устанавливались в 10 Мбит/с. Генерировались серии из одного, двух, четырех, восьми и шестнадцати потоков, в каждом устанавливались опции: «-w 300», «-f», «-q», «-s 1472». Исследовались результаты первого и последнего в группе потоков эхо-запросов. Для потоковой посылки эхо-запросов нагрузка в каждом потоке может быть вычислена по формуле:

$$\rho = \frac{\tau_{10}}{RTT},$$

где $\tau_{10} = 800L_f$ нс — время задействия интерфейса передаваемым кадром в окне размером L_f байтов; RTT — математическое ожидание (*avg*) времени оборота эхо-запроса, подсчитываемое утилитой *nanoping*.

Результаты экспериментов приведены на рис. 2.

=== 1 поток ===

```
99374 packets transmitted, 99373 received, 0% packet loss,
  time 299998ms
rtt min/avg/max/mdev = 2637615/2967576/3822843/410802 ns,
ipg/ewma
3018915.000/2718750.000 ns
```

=== 2 потока ===

```
108310 packets transmitted, 108309 received,
0% packet loss,
time 299999ms
rtt min/avg/max/mdev = 2640310/2718669/3193736/14116 ns,
ipg/ewma
2769848.000/2711170.000 ns
```

```
108309 packets transmitted, 108308 received, 0% packet
loss,
time 299997ms
rtt min/avg/max/mdev = 2613631/2718738/3132456/13467 ns,
ipg/ewma
2769854.000/2714933.000 ns
```

=== 4 потока ===

```
60956 packets transmitted, 60955 received, 0% packet loss,
time 299999ms
rtt min/avg/max/mdev = 3629529/4871212/5081284/20531 ns,
ipg/ewma
4921657.000/4362347.000 ns
```

```
60958 packets transmitted, 60957 received, 0% packet loss,
time 299998ms
rtt min/avg/max/mdev = 2710501/4871038/5307341/30187 ns,
ipg/ewma
4921479.000/4873465.000 ns
```

=== 8 потоков ===

```
30489 packets transmitted, 30488 received, 0% packet loss,
time 299999ms
rtt min/avg/max/mdev = 2707196/9793776/10023159/138144 ns,
ipg/ewma
9839924.000/4085698.000 ns
```

Рис. 2 (начало). Листинг результатов обеспечения нагрузки сетевых адаптеров компьютеров эхо-запросами размером 1472 байта

```

30484 packets transmitted, 30483 received, 0% packet loss,
time 299998ms
rtt min/avg/max/mdev = 3454346/9795301/10012359/82103 ns,
ipg/ewma
9841501.000/8710583.000 ns

== 16 потоков ==

14921 packets transmitted, 14917 received, 0% packet loss,
time 299994ms
rtt min/avg/max/mdev = 3517285/79103820/1983447196/-
6076538437164650 ns,
pipe 5, ipg/ewma 20106872.000/77235818.000 ns

14939 packets transmitted, 14935 received, 0% packet loss,
time 299998ms
rtt min/avg/max/mdev = 12339861/79210775/1983237024/
-5989747665710630
ns, pipe 5, ipg/ewma 20082932.000/76818779.000 ns

```

Рис. 2 (окончание). Листинг результатов обеспечения нагрузки сетевых адаптеров компьютеров эхо-запросами размером 1472 байта

Во всех случаях время задействия сетевых интерфейсов компьютера-источника и компьютера-приемника составляет 1 230 400 нс, так как ICMP эхо-запросы и эхо-ответы передаются во фреймах размером 1538 байтов. Для одного потока нагрузка интерфейсов составляет 41,5 %, что и следовало ожидать. Каждым из двух потоков интерфейсы загружаются на 46,5 %, что дает суммарную нагрузку 93 %. В обоих потоках в сумме передано 216 619 ICMP-пакетов за 300 с, т. е. интенсивность нагрузки ICMP-протокола составила 722 с^{-1} , не превышая 1000 с^{-1} . Для четырех потоков «саморегулирование» команды *netopring* ожиданием эхо-ответов приводит к нагрузке 25,3 % по каждому потоку, суммарная нагрузка интерфейсов незначительно превышает 100 %. Общее число ICMP-пакетов в четырех потоках составляет 24 400 за 300 с, что соответствует интенсивности 814 с^{-1} . Восемь потоков загружают интерфейс на 12,6 %, что также незначительно превышает 100 % в сумме и дает ту же суммарную интенсивность обеспечения эхо-ответов. Однако дальнейшее удвоение количества формируемых эхо-запросов приводит к существенному увеличению RTT, сообщениям *pipe* в каждом потоке и переполнению разрядной сетки при подсчете среднеквадратического отклонения (*mdev*) RTT, что свидетельствует об ошибочности результатов экспериментов.

Аналогичные эксперименты были проведены при тех же аппаратно-программных параметрах экспериментальной сети для размеров ICMP-пакетов в 512 байтов. Для этой величины эхо-запросов $\tau_{10} = 462\,400$ нс. Результаты экспериментов показаны на рис. 3.

Ожидаемое снижение коэффициента использования одного потока составляет 41,5...40,5 %. В дальнейшем характер изменения нагрузки аналогичен предыдущему случаю. Однако даже для шестнадцати потоков отсутствие сообщений *pipe* свидетельствует о корректности результатов проведенных измерений. Лишь при тридцати двух потоках возникают ошибки в экспериментах.

```
=== 1 поток ===
251727 packets transmitted, 251726 received, 0% packet
loss,
time 299999ms
rtt min/avg/max/mdev = 1043114/1142399/1626036/25055 ns,
ipg/ewma
1191769.000/1114773.000 ns

=== 2 потока ===

266581 packets transmitted, 266580 received, 0% packet
loss,
time 299999ms
rtt min/avg/max/mdev = 1041566/1093777/1590607/17527 ns,
ipg/ewma
1125365.000/1105864.000 ns

266580 packets transmitted, 266580 received, 0% packet
loss,
time 299998ms
rtt min/avg/max/mdev = 1041838/1093564/1587781/17555 ns,
ipg/ewma
1125365.000/1106244.000 ns

=== 4 потока ===

162193 packets transmitted, 162192 received, 0% packet
loss,
time 299999ms
rtt min/avg/max/mdev = 1528779/1816461/2059721/14988 ns,
ipg/ewma
1849655.000/1821076.000 ns

162193 packets transmitted, 162192 received, 0% packet
loss,
time 299999ms
rtt min/avg/max/mdev = 1358022/1817088/2298117/15359 ns,
ipg/ewma
1849655.000/1818915.000 ns

=== 8 потоков ===

81123 packets transmitted, 81122 received, 0% packet loss,
time 299998ms
rtt min/avg/max/mdev = 1119813/3664469/3891395/53330 ns,
ipg/ewma
3698115.000/3663460.000 ns

root@node1:~/runtask2# cat
val_20151029_8.task.node2eth1.8.res
81118 packets transmitted, 81117 received, 0% packet loss,
time 299999ms
rtt min/avg/max/mdev = 1301306/3664728/3882767/46930 ns,
ipg/ewma
3698355.000/3408322.000 ns
```

Рис. 3 (начало). Листинг результатов обеспечения нагрузки сетевых адаптеров компьютеров эхо-запросами размером 512 байтов

=== 16 потоков ===

```
40556 packets transmitted, 40555 received, 0% packet loss,
time 299998ms
rtt min/avg/max/mdev = 3166133/7364006/7553485/61684 ns,
ipg/ewma
7397315.000/4509051.000 ns
```

```
40587 packets transmitted, 40586 received, 0% packet loss,
time 299997ms
rtt min/avg/max/mdev = 1284898/7358327/7594146/197304 ns,
ipg/ewma
7391639.000/6986774.000 ns
```

=== 32 потока ===

```
20003 packets transmitted, 19999 received, 0% packet loss,
time 299995ms
rtt min/avg/max/mdev = 3349339/56880329/1947144676/-
2896394069619398 ns,
pipe 5, ipg/ewma 14998268.000/56412077.000 ns
```

```
19803 packets transmitted, 19801 received, 0% packet loss,
time 299987ms
rtt min/avg/max/mdev = 9391957/57038482/1947707156/
-3680640238188100 ns,
pipe 5, ipg/ewma 15149358.000/48028049.000 ns
```

Рис. 3 (окончание). Листинг результатов обеспечения нагрузки сетевых адаптеров компьютеров эхо-запросами размером 512 байтов

Заключение. С помощью повсеместно используемой утилиты *ping* ICMP-протокола вполне возможно обеспечить требуемую нагрузку на интерфейсы узлов транспортной системы компьютерной сети, варьируя количеством запускаемых потоков эхо-запросов и их размером в необходимом для экспериментальных исследований диапазоне.

ЛИТЕРАТУРА

1. Столлингс В. Современные компьютерные сети. СПб.: Питер, 2003. 783 с.
2. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2012. 960 с.
3. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2011. 944 с.
4. Иванов И.П., Кондратьев А.Ю., Лохтуров В.А. Модернизация процесса измерений интервалов времени в операционных системах современных компьютеров // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2012. № 4. С. 44–59.
5. Бойченко М.К., Иванов И.П., Лохтуров В.А. Определение характеристик сетевого трафика компьютера // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2015. № 2. С. 133–140. DOI: 10.18698/0236-3933-2015-2-133-140
6. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 432 с.

7. Бойченко М.К., Иванов И.П., Кондратьев А.Ю. Доступность ресурсов транспортных подсистем корпоративных сетей // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2010. № 3. С. 103–118.

8. Бойченко М.К., Иванов И.П. Мониторинг ресурсов узлов корпоративной сети // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2010. № 2. С. 114–120.

Бойченко Максим Константинович — начальник информационного центра Управления информатизации – Вычислительный центр МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

Иванов Игорь Потапович — д-р техн. наук, доцент, заведующий кафедрой «Теоретическая информатика и компьютерные технологии», проректор по информатизации и модернизации МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

Кондратьев Александр Юрьевич — начальник отдела Управления информатизации — Вычислительный центр МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

Лохтуров Вячеслав Александрович — ведущий электроник лаборатории АИС Управления информатизации – Вычислительный центр МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

Просьба ссылаться на эту статью следующим образом:

Бойченко М.К., Иванов И.П., Кондратьев А.Ю., Лохтуров В.А. Обеспечение потребных нагрузок сетевых интерфейсов утилитой ping программного обеспечения протокола ICMP // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2016. № 4. С. 74–84.
DOI: 10.18698/0236-3933-2016-4-18

**REGULATING LOADS OF NETWORK INTERFACES USING PING
UTILITY OF ICMP PROTOCOL**

M.K. Boychenko

noc@bmstu.ru

I.P. Ivanov

ivanov@bmstu.ru

A.Yu. Kondrat'iev

V.A. Lokhturov

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

Self-similar properties of traffic in networks with Ethernet technology occur at high rates of network interface utilization of end and transit nodes. Therefore, research developers, integrators and administrators of computer networks and transport systems need to have a tool for regulating the network interface load other than costly traffic generators. In this article we offer a technique to ensure the desired load using the ubiquitous ping utility of software ICMP protocol, or its upgraded version — nanoping. The results of our experimental research prove the efficiency of this method

Keywords

Computer network, network interface, bandwidth, traffic, load, Ethernet, self-similarity

REFERENCES

- [1] Stallings W. High-speed networks and internets. Performance and quality of service. Prentice Hall PTR Upper Saddle River, N.J., USA, 2001.
- [2] Tanenbaum A., Wetherall D. Computer networks. 5th ed. Prentice Hall, Inc., Upper Saddle River, N.J., 2011.
- [3] Olifer V.G., Olifer N.A. Komp'yuternye seti. Printsipy, tekhnologii, protokoly [Computer networks. Principles, technologies, protocols]. St. Petersburg, Piter Publ., 2011. 944 p.
- [4] Ivanov I.P., Kondrat'ev A.Yu., Lokhturov V.A. Modernization of process of measuring time intervals in operating systems of contemporary computers. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2012, no. 4, pp. 44–59 (in Russ.).
- [5] Boychenko M.K., Ivanov I.P., Lokhturov V.A. Definition of characteristics for computer's network traffic. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2015, no. 2, pp. 133–140 (in Russ.). DOI: 10.18698/0236-3933-2015-2-133-140
- [6] Kleinrock L. Queueing systems: Volume 2: Computer applications. N.Y. Wiley, 1976.
- [7] Boychenko M.K., Ivanov I.P., Kondrat'ev A.Yu. Availability of resources of transport sub-systems of corporative networks. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2010, no. 3, pp. 103–118 (in Russ.).
- [8] Boychenko M.K., Ivanov I. P. Monitoring of resources of corporate network junctions. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2010, no. 2, pp. 114–120 (in Russ.).

Boychenko M.K. — Head of Information Control and Computing Center, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

Ivanov I.P. — Dr. Sci. (Eng.), Head of Theoretical Informatics and Computer Technology Department, Pro-rector for Information and Modernization, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

Kondrat'ev A.Yu. — Head of Computer Technology Department, Information Control, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

Lokhturov V.A. — Leading electronics engineer of the AIS laboratory of the Administration on Informatization – Computing Center, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Boychenko M.K., Ivanov I.P., Kondrat'ev A.Yu., Lokhturov V.A. Regulating Loads of Network Interfaces using Ping Utility of ICMP Protocol. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2016, no. 4, pp. 74–84. DOI: 10.18698/0236-3933-2016-4-74-84