

УДК 621.391

**ОСОБЕННОСТИ ПРОВЕДЕНИЯ ЛАБОРАТОРНОГО ЭКСПЕРИМЕНТА  
ПО ИССЛЕДОВАНИЮ ПРОЦЕССОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ В  
ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЯХ**

ЛЕМЕШКО А.В., СЕМЕНЯКА М.В., АЛИ САЛЕМ АЛИ

Харьковский национальный университет радиоэлектроники

**MAIN FEATURES OF THE EXPERIMENTAL INVESTIGATION OF QUEUE  
MANAGEMENT PROCESSES IN TELECOMMUNICATION NETWORKS**

LEMESHKO A.V., SEMENYAKA M.V., ALI SFLEM ALI

Kharkov national university of radio electronics

***Аннотация.** В работе описаны основные особенности, методика проведения экспериментального исследования для проверки адекватности математических моделей управления очередями на участке реальной сети с помощью существующих механизмов управления очередями. Проведен сравнительный анализ полученных результатов математического моделирования и лабораторного эксперимента.*

***Annotation.** In the given article the main features and methodology aspects of experimental investigations for queue management models verification on the real network are presented. Comparative analysis of the obtained results of mathematical modeling and laboratory test is proposed.*

**ВВЕДЕНИЕ**

Ключевой задачей телекоммуникационных систем является обеспечение качества обслуживания запросов пользователей. Обеспечение качества обслуживания в свою очередь базируется на успешном решении задач управления трафиком, среди которых важное место занимают задачи управление очередями, так как именно они позволяют существенно улучшить такие показатели как средняя задержка, джиттер и уровень потерь пакетов без наращивания величины доступных сетевых ресурсов, прежде всего, пропускной способности каналов связи. В свою очередь, эффективность решения задач управления очередями зависит от адекватности моделей и методов, заложенных в их основу. Поэтому одним из наиболее важных этапов разработки моделей и методов управления трафиком в телекоммуникационной сети является их проверка в ходе экспериментального (лабораторного) исследования. При этом целью данного исследования выступает решение следующих частных задач:

- проверка адекватности разработанной модели балансировки нагрузки на буферный ресурс сетевого узла телекоммуникационной сети;
- анализ достоверности и эффективности полученных результатов решения оптимизационной задачи по управлению буферным ресурсом сетевого узла.

В ходе решения этих задач в зависимости от особенностей их постановки, предполагаемых результатов, требуемого уровня качества решения необходимо использовать как средства аналитического и имитационного моделирования, так и возможности натурального (лабораторного) эксперимента. Традиционно с помощью аналитического моделирования можно получить при различных исходных данных достаточно большой перечень результатов, которые удовлетворяют условиям разработанной модели, но требуют своего уточнения в ходе имитационных или натуральных испытаний. То есть средства аналитического моделирования должны органично дополняться возможностями натурального (лабораторного) эксперимента.

**ОБОСНОВАНИЕ ВЫБОРА ПАКЕТА НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ**

Среди множества существующих средств генерирования трафика: D-ITG (Distributed Internet Traffic Generator), Mtools, Rude/Crude, Mgen, Iperf, UDPgenerator пакет нагрузочного тестирования D-ITG показал наилучшие результаты по производительности [1, 2] при генерировании потоков UDP трафика. В сравнении с генераторами TCP-трафика: Iperf, Netperf, IP Traffic, IxChariot, D-ITG также показал один из наилучших результатов [3]. Результаты сравнения D-ITG с другими известными генераторами трафика представлены в табл. 1.

Таблица 1 – Сравнение различных пакетов нагрузочного тестирования

Название пакета	Iperf	Netperf	D-ITG	IP Traffic	IxChariot
Интерфейс	Командная строка	Командная строка	Командная строка, графический интерфейс	Графический интерфейс	Графический интерфейс
Поддержка различных платформ	Да	Да	Да	Только Windows	Да
Руководство пользователя	Да	Да	Да	Да	Да
Поддерживаемые протоколы	TCP, UDP	TCP, UDP, SCTP, DLPI	TCP, UDP, ICMP, VoIP	TCP, UDP, IGMP	TCP, UDP, RTP, SPX, VoIP
Управление распределением длины пакета	Нет	Нет	Да	Да	Нет
Распределение вероятности	Нет	Нет	Да	Да	Нет
Ведение файлов журналов	Да	Нет	Да	Да	Да
Internet Protocol	IPv6 и IPv4	IPv6 и IPv4	IPv6 и IPv4	IPv6 и IPv4	IPv6 и IPv4
Величины измерения характеристик потока	– джиттер; – количество потерянных пакетов; – пропускная способность.	– количество потерянных пакетов; – пропускная способность; – время отклика; – использование ядра.	– задержка в прямом направлении; – задержка в обратном направлении; – джиттер; – количество потерянных пакетов; – пропускная способность.	– задержка в прямом и обратном направлении; – джиттер; – количество потерянных пакетов; – пропускная способность.	– задержка в прямом направлении; – джиттер; – количество потерянных пакетов; – пропускная способность; – MOS; – MDI.
Свободное программное обеспечение	Да	Да	Да	Да	Нет

D-ITG – это программный пакет для генерирования трафика, позволяющий управлять распределением и длинны пакета (Packet Size) и времени между отправкой пакетов (Inter Departure Time), среди которых экспоненциальное, равномерное, распределение Коши, нормальное, распределение Парето, Пуассоновское распределение, гамма-распределение. D-ITG поддерживает протоколы IPv4 и IPv6, позволяя генерировать трафик на сетевом, транспортном уровнях и уровне приложений. В отличие от других генераторов трафика D-ITG является свободным программным обеспечением [2]. D-ITG можно использовать на оборудовании под управлением Linux, Windows, а также схожих с Linux платформах. Пакет позволяет генерировать множество потоков с различной функциональностью. D-ITG поддерживает протоколы TCP, UDP, ICMP, DCCP, SCTP, DNS, Telnet, VoIP (G.711, G.723, G.729, Voice Activity Detection, Compressed RTP). D-ITG способен обеспечить высокую скорость передачи данных между отправителем и получателем (до 612 Мбит/с).

С помощью D-ITG возможно производить измерения задержки передачи пакетов в направлении из начала в конец (one-way-delay, OWD), а также измерение времени передачи пакетов в направлении из начала в конец и обратно (round-trip-time, RTT), определить количество потерянных пакетов (packet loss), джиттер (jitter) и пропускную способность (throughput) тракта передачи. Пакет D-ITG способен хранить информацию об отправленном и принятом трафиках, как на стороне отправителя, так и на стороне получателя. К тому же, D-ITG позволяет передавать и хранить отчеты процесса генерирования трафика на удаленном сервере.

Число и характер поступающего в сеть трафика задавались с помощью пакета D-ITG, исходя из его функциональных возможностей, а именно способности генерировать множество потоков с различными параметрами путем задания значения поля DSCP, высокой производительности и точности полученных результатов. D-ITG позволяет явно указывать значения поля TOS (Type of Service) – байта в заголовке IP-пакета, содержащего набор критериев, определяющих тип обслуживания. Поле TOS имеет 6 бит поля Differentiated Services Code Point (DSCP) и 2-битовые поля Explicit Congestion Notification. DSCP – поле в IP-пакете, позволяющее назначить сетевому трафику различные уровни обслуживания. Для достижения этого каждый пакет в сети помечался кодом DSCP и соответствующим ему уровнем обслуживания. Биты 0–2 определяли приоритет (precedence) IP-пакета, бит 3 – требование к задержке (delay) передачи IP-пакета (0 – по умолчанию, 1 – низкая задержка), 4 – требование к пропускной способности (throughput) маршрута, по которому должен отправляться пакет (0 – низкая, 1 – высокая пропускная способность) 5 – требование к надежности (reliability) передачи (0 – по умолчанию, 1 – высокая надежность).

С помощью D-ITG задавалась продолжительность эксперимента и начальная задержка. Связь между отправителем и получателем осуществлялась с помощью выделенного канала сигнализации, который определял правила проведения эксперимента (Traffic Specification Protocol). К тому же, стороной отправителя можно было удаленно управлять с помощью ITGapi. На стороне отправителя D-ITG запускался в режиме приложения, т.е. генерированием трафика можно было управлять и удаленно.

### ОБЩАЯ СХЕМА ЛАБОРАТОРНОГО ЭКСПЕРИМЕНТА

Общая схема лабораторного эксперимента представлена на рис. 1. В эксперименте взаимодействовали три модуля. 1-й и 3-й модули представляли собой конечные станции абонентов. Стороной отправителем трафика являлся модуль 1, получателем трафика выступал модуль 3. Взаимодействие конечных станций осуществлялось через модуль 2, который являл собой участок сети, представленный маршрутизаторами компании Cisco серии C2801. В качестве конечных станций абонентов использовались стандартные персональные компьютеры, которые подключались к маршрутизаторам непосредственно или через коммутаторы посредством технологии FastEthernet.

Модуль 1 выполнял генерирование трафика. С помощью пакета нагрузочного тестирования D-ITG был сгенерирован многопоточковый трафик с различными параметрами интенсивности, приоритета обслуживания пакетов, длины пакета и распределения времени отправки пакетов для каждого потока. В модуле 2 путем статической настройки интерфейсов маршрутизатора, политик обслуживания трафика, классов трафика апробировались результаты математического моделирования распределения трафика.

В качестве механизма управления очередями на маршрутизаторе был выбран алгоритм CBWFQ, благодаря возможности гибкого управления очередями для каждого трафика, поступающего на интерфейс маршрутизатора [4]. Модуль 3 выступал анализатором трафика. Благодаря встроенным возможностям пакета D-ITG были получены показатели, по которым оценивалось качество обслуживания каждого трафика: средняя задержка (RFC 1242), джиттер (RFC 1889), потери пакетов (RFC 1242).

### НАСТРОЙКА КОНЕЧНОЙ СТАНЦИИ, ГЕНЕРИРУЮЩЕЙ ТРАФИК

С помощью пакета нагрузочного тестирования D-ITG, установленного на конечных станциях, задавалось количество трафиков и их параметры. Для примера, на рис. 2 показаны настройки для генерирования 4 потоков трафика (Flow1, Flow2, Flow3, Flow4) по протоколу UDP. Каждому трафику было задано уникальное имя для его идентификации на приемной стороне. Трафики Flow1 и Flow2 имели статические настройки параметров интенсивности, распределения времени отправки и длины пакетов. Трафики Flow3 и Flow4 имели характеристики соответствующие игровому трафику (Flow3) и голосовому трафику (Flow4).

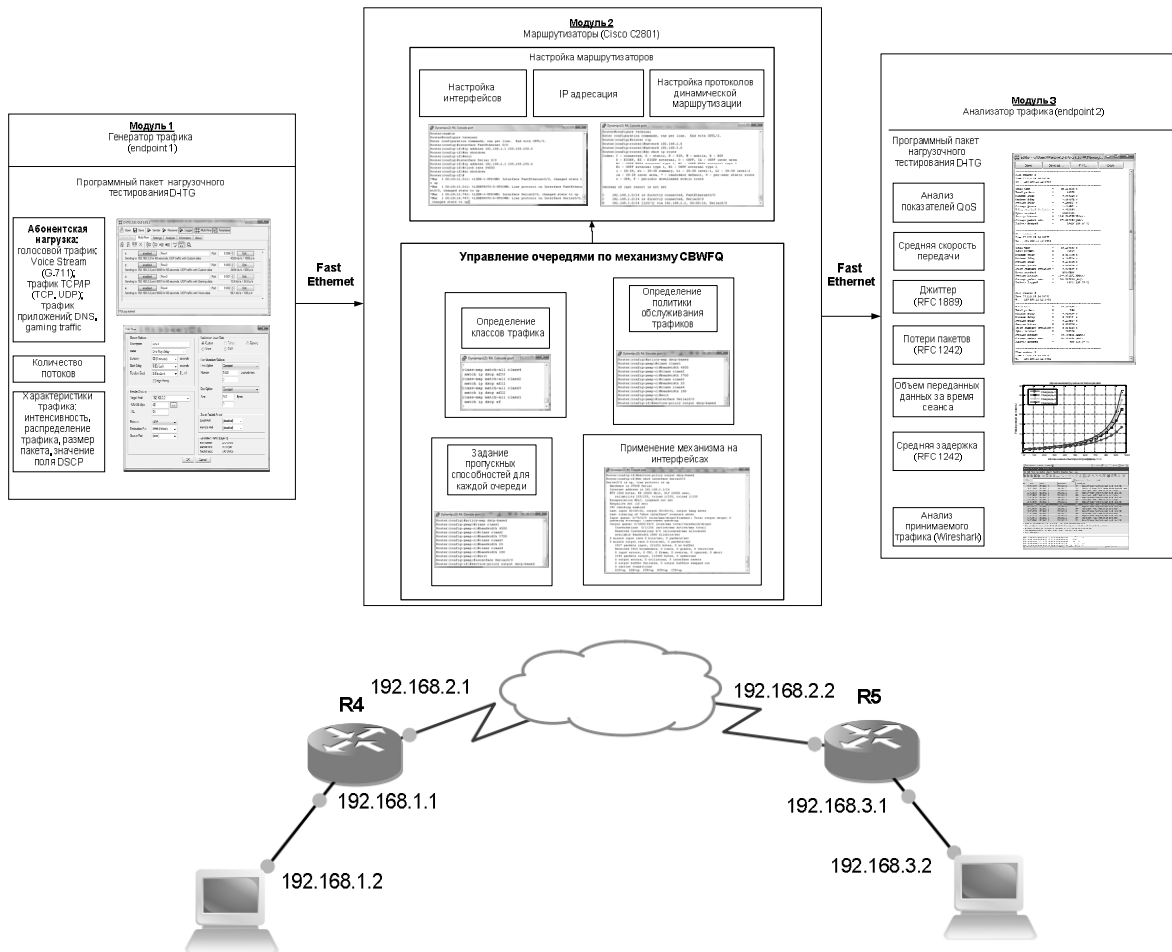


Рисунок 1 – Общая схема лабораторного эксперимента

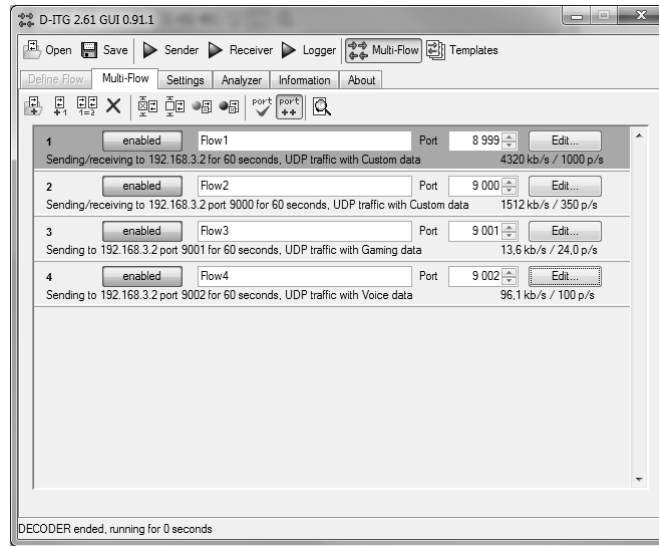
Скорости передачи для Flow1 и Flow2, например, были указаны 4320 Кбит/с и 1512 Кбит/с статически, для Flow3 и Flow4 скорость передачи была выбрана автоматически 13,6 Кбит/с и 96,1 Кбит/с, что соответствует параметрам игрового и голосового трафика. Пример настройки потока Flow1 показан на рис. 2б, 2в. Для измерения задержки был выбран режим измерения времени передачи пакета из конца в конец и обратно. Проблема, возникающая при определении задержки в одном направлении, заключалась в синхронизации времени на конечных станциях, что трудно обеспечить без дополнительных программных средств. Продолжительность эксперимента была выбрана равной 60 с для каждого трафика. Начальная задержка равнялась нулю. На рис. 2 IP-адрес получателя был 192.168.3.2, значение поля кода дифференцированной услуги (DSCP) для Flow1, Flow2, Flow3, Flow4 составляло EF, AF11, AF22, AF33 соответственно (рис. 2в).

Проблема, возникающая при маркировке пакетов, заключалась в том, что трафик, сгенерированный на конечных станциях под управлением операционной системы Windows посредством пакета D-ITG имел значение поля DSCP по умолчанию (Best Effort), независимо от того какое значение было указано в настройках генератора. Поэтому для решения данной проблемы использовались конечные станции под управление операционной системы Ubuntu Linux.

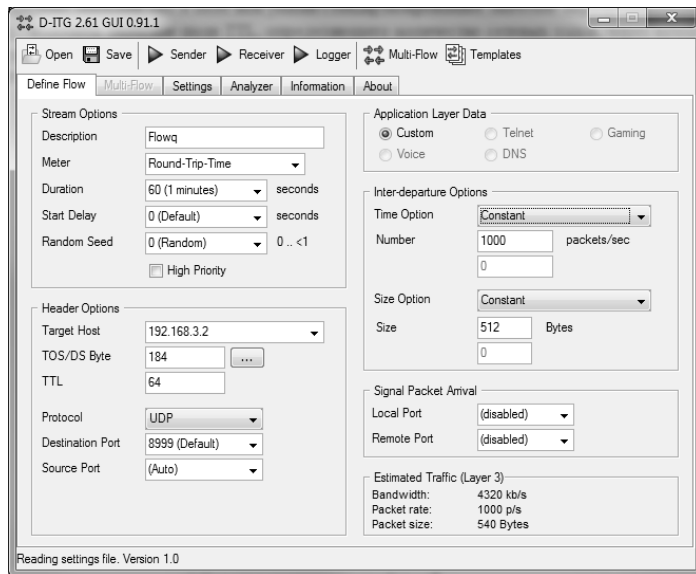
### ВЫБОР И НАСТРОЙКА АЛГОРИТМА ОБСЛУЖИВАНИЯ ОЧЕРЕДЕЙ НА МАРШРУТИЗАТОРАХ CISCO

Для эмуляции соединения между маршрутизаторами использовались последовательные интерфейсы стандарта V.35. При настройке пропускной способности исходящего канала связи на маршрутизаторах, с целью обеспечения требуемой физической скорости передачи пакетов, устанавливалась необходимая частота синхронизации (clock rate) (рис. 3а). В качестве механизма управления очередями был выбран механизм взвешенного справедливого обслуживания на основе классов тра-

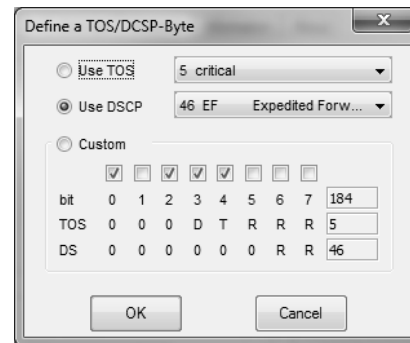
фика (CBWFQ) ввиду его широких возможностей по конфигурированию очередей. Механизм Class-based weighted fair queuing (CBWFQ) является расширением стандартного механизма справедливого обслуживания WFQ и обеспечивает поддержку созданных пользователем классов трафика [4]. В CBWFQ возможно определить классы трафика на основе соответствия каким-либо критериям, в том числе и по протоколу передачи трафика, с помощью списков контроля доступом (ACLs), а также на основе входящих интерфейсов. Пакетам, которые удовлетворяют критерию отбора, присваивается значение заданного класса трафика. Очередь в маршрутизаторе резервируется под каждый класс, и трафик, принадлежащий определенному классу, направляется в очередь для этого класса.



а) пример задание количества трафиков



б) пример задания параметров трафиков



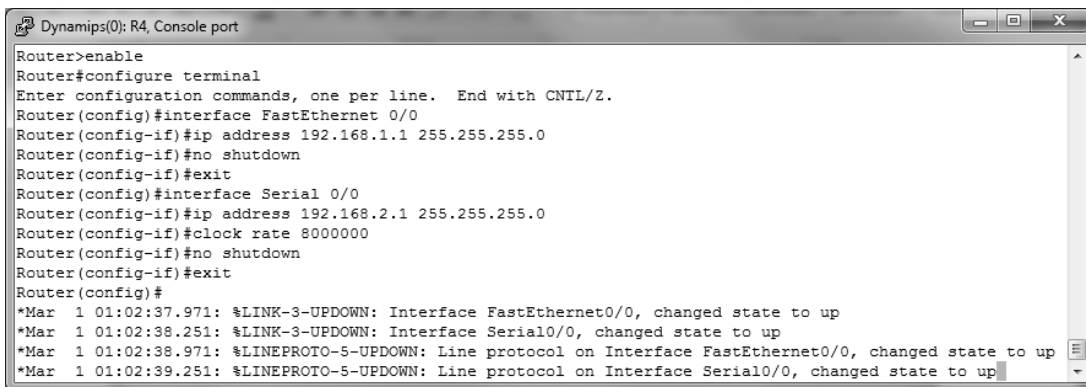
в) пример задания значения поля DSCP

Рисунок 2 – Примеры задания параметров сетевого трафика в пакете D-ITG

Проблема, возникшая при маркировке пакетов, заключалась в том, что трафик, сгенерированный на конечных станциях под управлением операционной системы Windows посредством пакета D-ITG имел значение поля DSCP по умолчанию (Best Effort), независимо от того какое значение было указано в настройках генератора. Поэтому для решения данной проблемы использовались конечные станции под управлением операционной системы Ubuntu Linux.

## ВЫБОР И НАСТРОЙКА АЛГОРИТМА ОБСЛУЖИВАНИЯ ОЧЕРЕДЕЙ НА МАРШРУТИЗАТОРАХ CISCO

Для эмулирования соединения между маршрутизаторами использовались последовательные интерфейсы стандарта V.35. При настройке пропускной способности исходящего канала связи на маршрутизаторах, с целью обеспечения требуемой физической скорости передачи пакетов, устанавливалась необходимая частота синхронизации (clock rate) (рис. 3а). В качестве механизма управления очередями был выбран механизм взвешенного справедливого обслуживания на основе классов трафика (CBWFQ) ввиду его широких возможностей по конфигурированию очередей. Механизм Class-based weighted fair queuing (CBWFQ) является расширением стандартного механизма справедливого обслуживания WFQ и обеспечивает поддержку созданных пользователем классов трафика [4]. В CBWFQ возможно определить классы трафика на основе соответствия каким-либо критериям, в том числе и по протоколу передачи трафика, с помощью списков контроля доступом (ACLs), а также на основе входящих интерфейсов. Пакетам, которые удовлетворяют критерию отбора, присваивается значение заданного класса трафика. Очередь в маршрутизаторе резервируется под каждый класс, и трафик, принадлежащий определенному классу, направляется в очередь для этого класса.

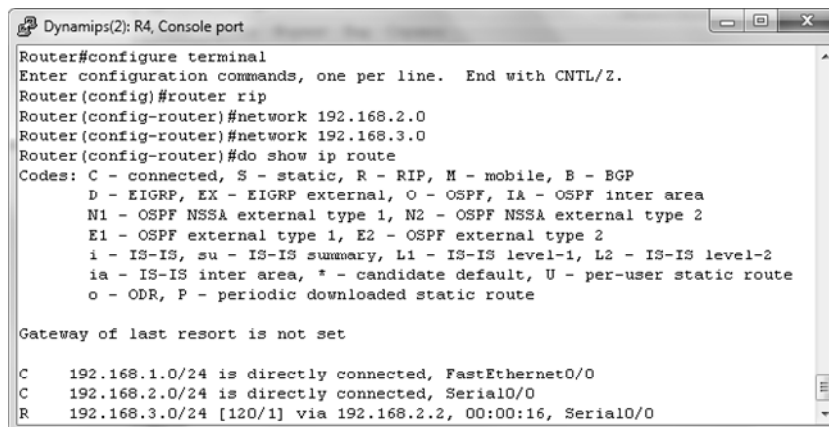


```

Dynamips(0): R4, Console port
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet 0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Serial 0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#clock rate 8000000
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
*Mar 1 01:02:37.971: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 01:02:38.251: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar 1 01:02:38.971: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar 1 01:02:39.251: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up

```

а) пример настройки адресации на интерфейсах и задания пропускной способности каналов связи



```

Dynamips(2): R4, Console port
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial0/0
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:16, Serial0/0

```

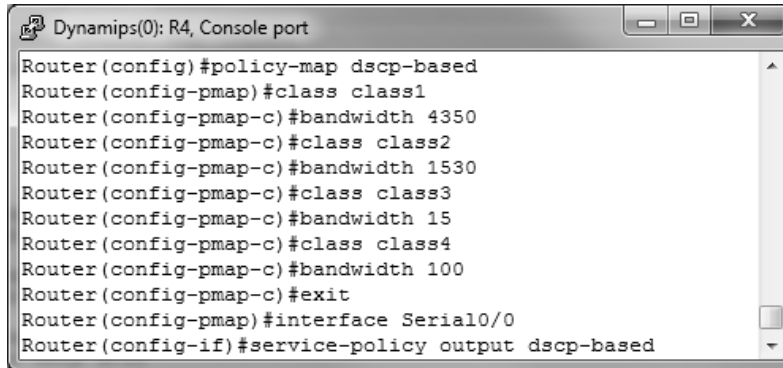
б) пример настройки протокола динамической маршрутизации

**Рисунок 3** – Пример настройки адресации и маршрутизации на маршрутизаторе сети

После задания класса, исходя из критериев отбора в него трафика, можно изменить его характеристики (назначить классу пропускную способность, вес, максимальный размер очереди). Пропускная способность, назначенная классу, гарантирует доставку пакетов заданного класса в случае возникновения перегрузки канала. Передача пакетов, принадлежащих определенному классу, зависит от выделенной для класса пропускной способности и максимального размера очереди. После достижения очередью максимального размера, новые пришедшие пакеты вызывают отбрасывание пакетов в очереди в зависимости от выбранной политики отбрасывания.

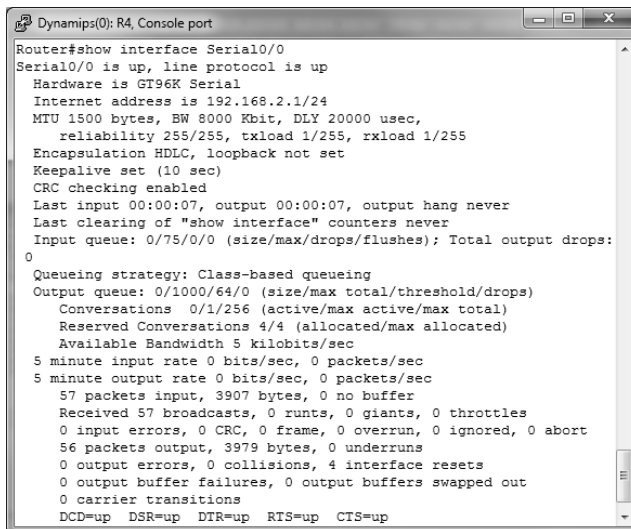
Для примера (рис. 3) в качестве интерфейсов, на которых применялся механизм управления очередями CBWFQ, использовались последовательные интерфейсы стандарта V.35. На рис. 3а пока-

зан пример настройки интерфейса с адресом 192.168.1.1, частотой синхронизации 8000 Кбит/с. На рис. 3а, 3б показан пример настройки динамической маршрутизации на стороне отправителя трафика. На рис. 4а, 4б показан пример задания четырех классов трафика class1, class2, class3, class4, в которых критерием для классификации трафика выступало поле DSCP в заголовке IP-пакета. К классам class1, class2, class3, class4 относился трафик со значением поля DSCP равным EF, AF11, AF22, AF33 соответственно. Для классов class1, class2, class3, class4 была выделена пропускная способность физического интерфейса равная 4500 Кбит/с, 3700 Кбит/с, 20 Кбит/с, 150 Кбит/с соответственно. Результаты применения настроек на интерфейсе показаны на рис. 3б.

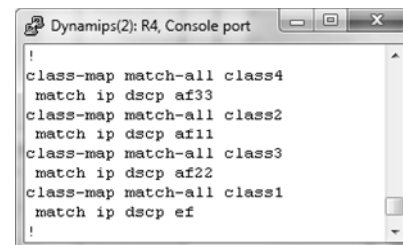


```
Dynamips(0): R4, Console port
Router(config)#policy-map dscp-based
Router(config-pmap)#class class1
Router(config-pmap-c)#bandwidth 4350
Router(config-pmap-c)#class class2
Router(config-pmap-c)#bandwidth 1530
Router(config-pmap-c)#class class3
Router(config-pmap-c)#bandwidth 15
Router(config-pmap-c)#class class4
Router(config-pmap-c)#bandwidth 100
Router(config-pmap-c)#exit
Router(config-pmap)#interface Serial0/0
Router(config-if)#service-policy output dscp-based
```

а) пример задания классов трафика, политики управления очередями, пропускных способностей для каждой очереди и применение политики на интерфейсе



```
Dynamips(0): R4, Console port
Router#show interface Serial0/0
Serial0/0 is up, line protocol is up
Hardware is GT96K Serial
Internet address is 192.168.2.1/24
MTU 1500 bytes, BW 8000 Kbit, DLY 20000 usec,
  reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
CRC checking enabled
Last input 00:00:07, output 00:00:07, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
Queueing strategy: Class-based queueing
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/1/256 (active/max active/max total)
Reserved Conversations 4/4 (allocated/max allocated)
Available Bandwidth 5 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
57 packets input, 3907 bytes, 0 no buffer
Received 57 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
56 packets output, 3979 bytes, 0 underruns
0 output errors, 0 collisions, 4 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```



```
Dynamips(2): R4, Console port
!
class-map match-all class4
 match ip dscp af33
class-map match-all class2
 match ip dscp af11
class-map match-all class3
 match ip dscp af22
class-map match-all class1
 match ip dscp ef
!
```

б) Пример проверки применения настроек на интерфейсе

**Рисунок 4** – Пример настроек алгоритма обслуживания очередей CBWFQ на маршрутизаторах Cisco

## АНАЛИЗ РЕЗУЛЬТАТОВ ЛАБОРАТОРНОГО ЭКСПЕРИМЕНТА

Анализ результатов исследования с точки зрения динамики основных показателей качества обслуживания осуществлялся с использованием как пакета нагрузочного тестирования D-ITG, так и с помощью средств Matlab. Для качественного анализа поступающего трафика (определения порядка прибытия пакетов на конечную станцию) использовалась программа-анализатор трафика Wireshark (рис. 5).

Пакет D-ITG (рис. 6) позволял оценить основные показатели качества обслуживания пакетов сгенерированного трафика (рис. 7): среднюю задержку пакетов (RFC 1242), количество отброшенных пакетов (RFC 1242), интенсивность пакетов, вариацию задержки (RFC 1889).

Известно, что в ходе управления очередями, как правило, решаются задачи, связанные с организацией очередей, определением порядка их наполнения пакетами того или иного трафика, а также закрепления за каждой очередью некоторой доли пропускной способности исходящего канала связи (физического интерфейса). Именно эти задачи формализуются в рамках известных моделей и методов управления очередями, сводя, как правило, технологическую проблему к решению той или иной

оптимизационной задачи. В качестве примера математической модели управления очередями была выбрана потоковая математическая модель динамического управления очередями.

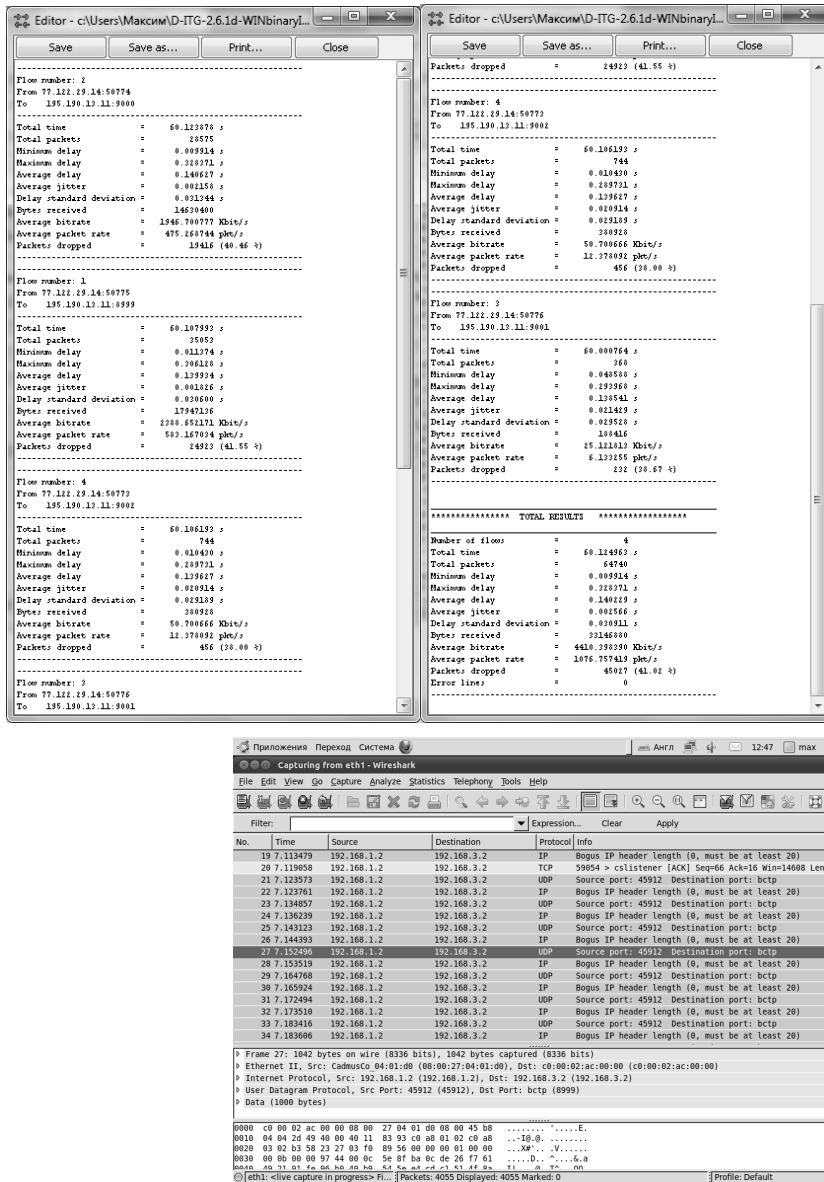


Рисунок 5 – Пример работы анализатора трафика Wireshark

Модель динамического управления очередями подробно описана в [5]. Сравнительный аналитический анализ моделей динамической балансировки очередей представлен в [6]. В рамках модели  $a_i$  ( $i = \overline{1, M}$ ) – интенсивность трафика  $i$ -го класса, поступающего на обслуживание маршрутизатором ТКС. Кроме того,  $b_j$  ( $j = \overline{1, N}$ ) – часть пропускной способности исходящего канала связи (КС), которая выделена для обслуживания пакетов из  $j$ -й очереди. При этом необходимо выполнить условие

$$\sum_{j=1}^N b_j = b, \quad (1)$$

где  $b$  – пропускная способность исходящего КС.

С целью предотвращения перегрузки маршрутизатора в целом необходимо обеспечить выполнение следующего условия:



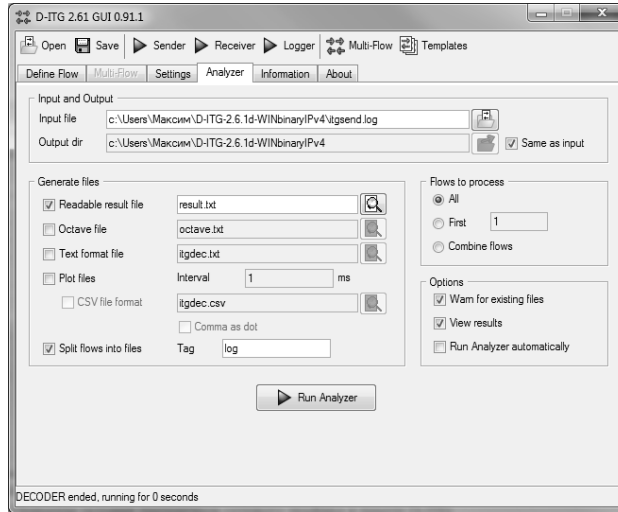


Рисунок 6 – Запуск встроенного анализатора трафика в пакете D-ITG  
ПОТОКОВАЯ МОДЕЛЬ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ

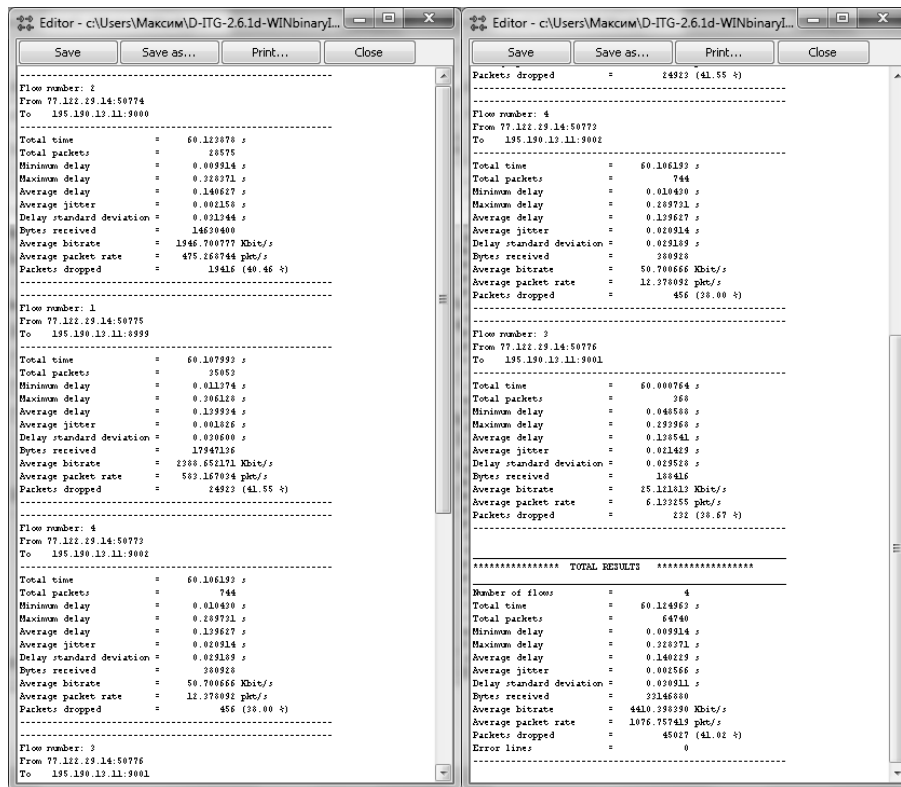


Рисунок 7 – Анализ результатов проведенного эксперимента по скоростным, временным показателям и показателям надежности

$$\sum_{i=1}^M a_i \leq b, \quad (2)$$

которое выполняется, прежде всего, за счет эффективной маршрутизации трафика в сети.

Придать динамический характер процессу управления очередями удалось за счет введения переменной  $x_{ij}$ , под которой подразумевалась часть  $i$ -го трафика, который будет направлен для об-

служивания в  $j$ -ю очередь. Согласно физическому смыслу переменной  $x_{ij}$  имеют место следующие дополнительные условия:

$$x_{ij} \geq 0 \quad (i = \overline{1, M}, j = \overline{1, N}), \quad (3)$$

$$\sum_{j=1}^N x_{ij} = a_i \quad (i = \overline{1, M}), \quad (4)$$

$$\sum_{i=1}^M x_{ij} \leq b_j \quad (j = \overline{1, N}). \quad (5)$$

Выполнение условия (4) гарантирует обслуживание всех пакетов трафика, поступающих на отдельный маршрутизатор. Условия (5) вводятся для предотвращения перегрузки очередей маршрутизатора по пропускной способности. В качестве искомым переменных выступал расширенный по содержанию и размерности вектор

$$\mathbf{r} \quad \mathbf{x} = \begin{bmatrix} x_{ij} \\ \dots \\ b_j \end{bmatrix} \quad (i = \overline{1, M}; j = \overline{1, N}), \quad (6)$$

т.е. задачи, связанные с распределением поступающих на маршрутизатор информационных потоков по очередям ( $x_{ij}$ ) и определением объема пропускной способности, выделенной для  $j$ -й очереди ( $b_j$ ), решались согласованно. Кроме того, необходимым является условие предотвращения перегрузки очереди по ее длине

$$\bar{n}_j \leq n_j^{\max} \quad (j = \overline{1, N}), \quad (7)$$

где  $\bar{n}_j$  и  $n_j^{\max}$  – средняя и максимально возможная длина  $j$ -й очереди.

Важно отметить, что средняя длина очереди зависит нелинейно от соотношения интенсивности потока к величине пропускной способности КС, выделенной для обслуживания пакетов из данной очереди, т.е. от параметра

$$r_j = \frac{\sum_{i=1}^M x_{ij}}{b_j}$$

Моделируя, например, процесс обслуживания каналом связи пакетов системой массового обслуживания M/M/1/n<sup>max</sup> получаем:

$$\bar{n} = \frac{r^2 [1 - (n^{\max} + 1)r^{n^{\max}} + n^{\max} r^{n^{\max} + 1}]}{(1 - r^{n^{\max} + 2})(1 - r)} - r \quad (8)$$

С целью более полного соответствия получаемых решений принципам концепции Traffic Engineering Queues, касающихся обеспечения сбалансированной загрузки буферных ресурсов, условие (7) приведено к форме.

$$\frac{p_j}{v \cdot d_j} \cdot \bar{n}_j \leq b \cdot n_j^{\max} \quad (j = \overline{1, N}), \quad (9)$$

где  $p_j$  и  $d_j$  – соответственно величина приоритета и длина пакетов из  $j$ -й очереди;  $v$  – некоторый нормировочный коэффициент, который должен сглаживать различие в порядке значений приоритета ( $0 \div 7$ ) и длины пакета в байтах;  $b$  – верхний динамически управляемый порог длин очередей.

В качестве критерия оптимальности получаемых решений выступало условие

$$\min_{x, b} b, \quad (10)$$

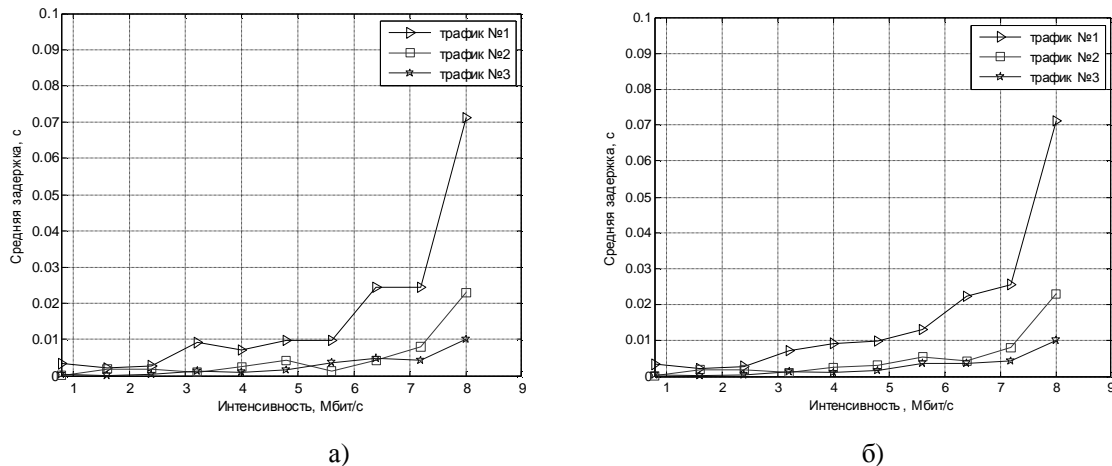
что соответствует минимизации верхнего порога длин очередей на маршрутизаторе ТКС, взвешенно-го относительно таких характеристик потока, как длина пакета и его приоритет.

Ввиду того, что модель управления очередями на узлах телекоммуникационной сети сведена к решению оптимизационной задачи, то на первом этапе исследования использовались возможности пакета аналитического моделирования Matlab, представленные программами Optimization Toolbox 3. Конечные решения оптимизационной задачи, представленные в виде порядка распределения трафиков по очередям маршрутизатора и пропускных способностей КС для каждой очереди, в дальнейшем использовались при настройке сетевого оборудования (маршрутизаторов Cisco с поддержкой различных политик управления очередями) с точки зрения конфигурирования механизмов управления очередями на его интерфейсах.

### РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

В ходе исследования модели были приняты следующие исходные данные: число потоков и очередей равно трем. Суммарная интенсивность входящего в сеть потока изменялась от 1 до 8 Мбит/с. Пропускная способность исходящего канала связи была равна 8 Мбит/с. Приоритеты трафика изменялись от 0 до 7. Максимальная длина каждой из очередей составляла соответственно 70, 25 и 10 пакетов. В результате через первую очередь передавался трафик с первым приоритетом, через вторую – трафик с третьим приоритетом и через третью – с седьмым приоритетом.

Установлено, что модель соответствуют требованиям концепции Traffic Engineering Queues, обеспечивая оптимальную балансировку очередей, взвешенную на основе приоритета (рис. 8а), т.к. удалось обеспечить более эффективное обслуживание более приоритетного трафика. Результаты математического моделирования процесса управления очередями и лабораторного эксперимента по показателю средней задержки представлены на рис. 8.



**Рисунок 8** – Зависимость средней задержки от интенсивности трафика на входе сети в результате математического моделирования (а) и в ходе лабораторного эксперимента (б)

Сходимость результатов математического моделирования к результатам лабораторного эксперимента определялась исходя из расхождения численных значений средней задержки пакетов (рис. 8)

$$S = \left| 1 - \frac{t_{cp\_м}}{t_{cp\_экс}} \right| \cdot 100\%,$$

где  $t_{cp\_м}$  – средняя задержка передачи пакетов в результате математического моделирования,  $t_{cp\_экс}$  – средняя задержка передачи пакетов в ходе лабораторного эксперимента.

Средняя задержка зависела напрямую от длины очереди на маршрутизаторе. В ходе исследований наблюдались низкие значения отклонения средней задержки пакетов ( $\leq 3\%$ ) в области низких ( $\leq 2,5$  Мбит/с) и высоких (от 6,5 до 8 Мбит/с) нагрузок на маршрутизатор. Это обусловлено тем, что при низкой нагрузке очереди практически не возникали, а задержка была обусловлена лишь небольшим временем перенаправления пакета из входящего интерфейса в исходящий канал связи. В области же высоких нагрузок очередь маршрутизатора (рис. 9) была загружена практически на 100%, что

вызвало стабилизацию величины средней задержки и минимальные расхождения в результатах лабораторного эксперимента и аналитического моделирования. В области же средних нагрузок (от 2,5 до 6,5 Мбит/с), благодаря постоянному мониторингу загруженности маршрутизатора, наблюдалось динамическое изменение длины очереди, что и обусловило основные расхождения результатов математического моделирования и лабораторного эксперимента ( $\leq 10\%$ ).

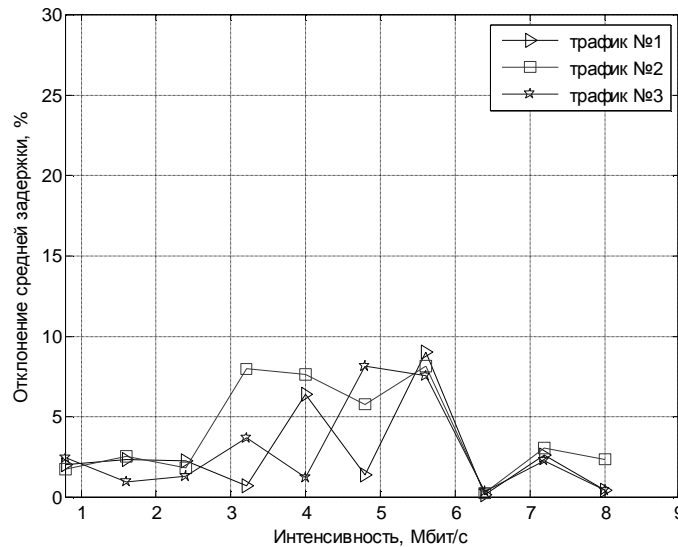


Рисунок 9 – Сравнение результатов математического моделирования и лабораторного эксперимента

### ВЫВОДЫ

Важным этапом в разработке и исследовании моделей и методов управления очередями является их экспериментальное исследование на реальном участке сети. Именно в рамках лабораторного эксперимента можно получить наиболее достоверные оценки эффективности полученных решений с обоснованием области применения этих результатов [7].

В статье предложена методика проведения эксперимента для исследования процессов управления очередями. Для генерирования трафика, передаваемого в сети, использовался пакет нагрузочного тестирования D-ITG. В качестве механизма управления очередями для проверки результатов математического моделирования был выбран механизм CBWFQ, благодаря его широким функциональным возможностям по управлению очередями, возможности в явном виде указывать значения пропускных способностей, закрепленных за каждой очередью, выбора политик обслуживания очередей, а также максимального размера каждой очереди.

В работе описан пример анализа адекватности и достоверности результатов, получаемых с помощью математической модели динамического управления очередями [5]. Сходимость результатов математического моделирования к результатам лабораторного эксперимента определялась исходя из отклонения показателя средней задержки пакетов. В результате проведенного сравнения наблюдались низкие значения отклонения средней задержки передачи пакетов ( $\leq 3\%$ ) в области низких и высоких нагрузок на маршрутизатор. Это обусловлено отсутствием динамического характера изменения длины очереди на маршрутизаторе, так как при низких нагрузках очереди на интерфейсе не возникают, а при высоких – буфер маршрутизатора почти переполнен. В области же средних нагрузок на физическом интерфейсе маршрутизатора в результате динамического изменения средней длины очереди наблюдаются основные расхождения в показателе средней задержки ( $\leq 10\%$ ) при математическом моделировании от результатов лабораторного эксперимента. Результаты исследований в целом подтвердили адекватность модели (1)-(10) и достоверность получаемых на ее основе результатов моделирования.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Avallone S., Guadagno S., Emma D., Pescape A., Ventre G. D-ITG Distributed Internet Traffic Generator // Proceedings of the First International Conference on the Quantitative Evaluation of Systems (QEST'04) – 2004. – Vol. 4. – P 58-59.
2. Avallone S., Emma D., Pescape A., Ventre G. Performance evaluation of an open distributed platform for realistic traffic generation // Performance evaluation. International journal – 2005. – P 359-392
3. Kolahi S. S., Narayan S., Nguyen Du. D. T., Sunarto Y. Performance Monitoring of Various Network Traffic Generators // UKSIM'11 Proceedings of the 2011 UKSim 13th International Conference on Computer Modelling and Simulation – 2011. – P 501-506.
4. Class-Based Weighted Fair Queuing [Электронный ресурс] / Cisco IOS Software Releases 12.0. – Режим доступа: [http://www.cisco.com/en/US/docs/ios/12\\_0t/12\\_0t5/feature/guide/cbwfq.html](http://www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html)
5. Али С. Али, Симоненко А.В. Потокковая модель динамической балансировки очередей в MPLS–сети с поддержкой Traffic Engineering Queues [Электронный ресурс] // Проблемы телекоммуникаций. – 2010. – № 1 (1). – С. 59 – 67. – Режим доступа к журн.: [http://pt.journal.kh.ua/2010/1/1/101\\_ali\\_balancing.pdf](http://pt.journal.kh.ua/2010/1/1/101_ali_balancing.pdf).
6. Лемешко А.В., Али С. Али, Семеняка М.В. Разработка и исследование потоковых моделей динамической балансировки очередей на маршрутизаторах мультисервисной телекоммуникационной сети // Системи обробки інформації. – Вип. 8(98). – 2011. - С. 218-223.
7. Евсеева О.Ю. Методика экспериментального исследования методов управления трафиком в сети [Электронный ресурс] // Проблемы телекоммуникаций. – 2011. – № 3 (5). – С. 89 – 104. – Режим доступа к журн.: [http://pt.journal.kh.ua/2011/3/1/113\\_yevseyeva\\_experiment.pdf](http://pt.journal.kh.ua/2011/3/1/113_yevseyeva_experiment.pdf).