

Военно-космическая академия имени А.Ф.Можайского
Факультет специальных информационных технологий
Кафедра систем сбора и обработки информации



КУРСОВАЯ РАБОТА

На тему «Исследование распределения времени задержки пакетов трафика в компьютерной сети при оценке качества (QoS)»

Курсантов 63 курса
группы №691/2
Белоцкого Н.Г.
Левицкого А.В.

Научный руководитель
п-к Гнидко К.О.
(Профессор 65 кафедры)

ОГЛАВЛЕНИЕ

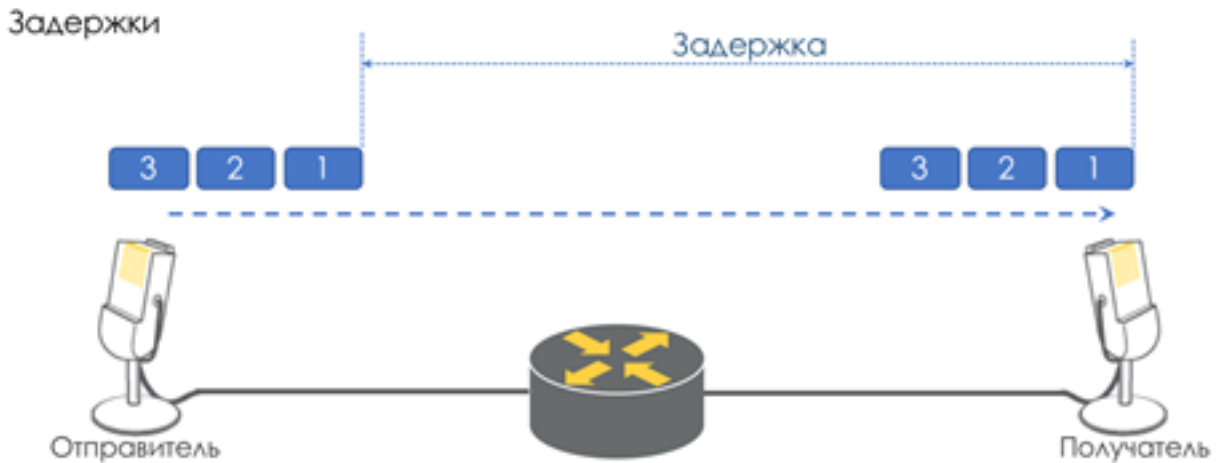
1. ВВЕДЕНИЕ	3
2. ТЕОРЕТИЧЕСКОЕ ПОЛОЖЕНИЕ	5
2.1 Три модели обеспечения QoS	5
2.2 Предотвращение перегрузок (Congestion Avoidance)	5
2.3 Управление перегрузками (Congestion Management)	5
3. ХОД ИССЛЕДОВАНИЯ	7
3.1 Получение исходных данных	7
3.2 Представление выборки	10
3.3 Статистический анализ выборки	10
4. ЗАКЛЮЧЕНИЕ И ВЫВОДЫ	14
5. ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА	15

1. ВВЕДЕНИЕ

Quality of Service (с англ. «качество обслуживания») — это набор технологий, которые запускаят высокоприоритетные приложения и трафик при лимитированной пропускной способности. Это означает, что более важный трафик будет обработан быстрее, а задержки по сети будут минимальны.

Измерения, касающиеся QoS, включают:

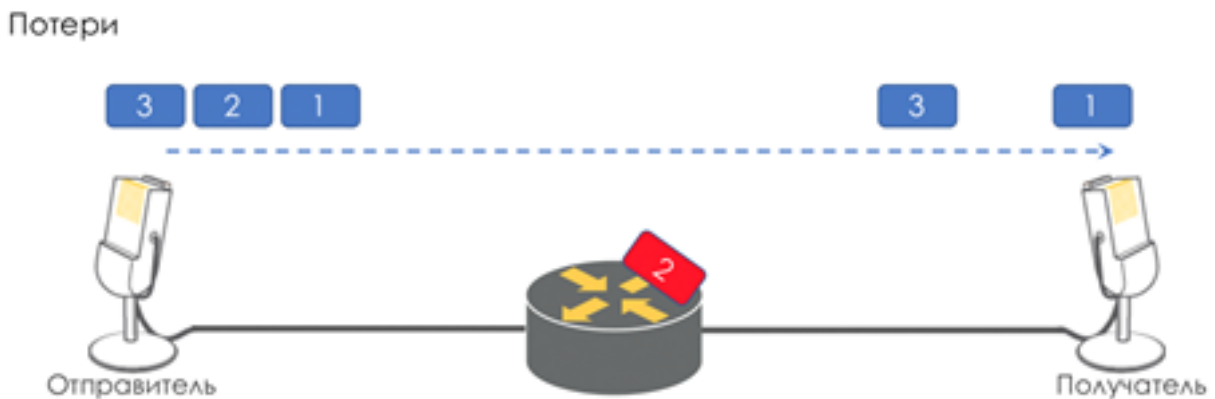
1. Задержка – время, за которое данные доходят от отправителя до получателя.



2. Джиттер «дрожание» - разница задержек при передаче данных.

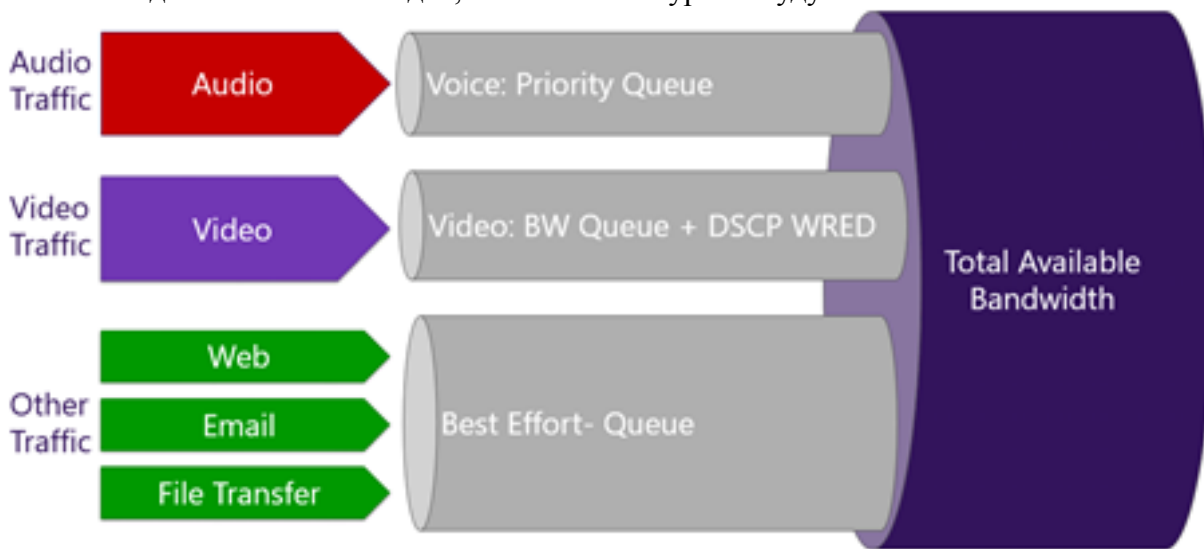


3. Потери – количество пакетов, потерянных во время передачи данных по сети.



4. Полоса пропускания – количество пакетов, проходящие по сети за определенный период времени

Благодаря технологии QoS можно научить маршрутизатор разделять пропускную способность и тогда ни потоковое видео, ни звонок в Skype не будут заикаться.



Как работают механизмы QoS для упорядочивания пакетов и выделения полосы пропускания:

- управление очередями;
- управление полосой пропускания.

Чтобы обеспечить приоритетное обслуживание определенного типа трафика, используют технологию QoS (Quality of Service). QoS обеспечивает выделенную полосу пропускания, контролирует задержку и джиттер, а также ограничивает потери данных.

2. ТЕОРЕТИЧЕСКОЕ ПОЛОЖЕНИЕ

2.1 Три модели обеспечения QoS

1. **Best Effort** — никакой гарантии качества. Все равны.

Самый простой подход к реализации QoS, с которого начинались IP-сети и который практикуется и по сей день — иногда потому что его достаточно, но чаще из-за того, что никто и не думал о QoS.

2. **IntServ** — гарантия качества для каждого потока. Резервирование ресурсов от источника до получателя.

Название отражает стремление в одной сети одновременно предоставлять услуги для реал-тайм и не-реал-тайм типов трафика, предоставив, при этом первым приоритетное право использования ресурсов через резервирование полосы. Возможность переиспользования полосы, на которой все и зарабатывают, и благодаря чему IP выстрелил, при этом сохранялась.

3. **DiffServ** — нет никакого резервирования. Каждый узел сам определяет, как обеспечить нужное качество.

Оно отражает, что мы дифференцируем сервисы, предоставляемые различным приложениям, а точнее их трафику, иными словами разделяем/дифференцируем эти типы трафика.

2.2 Предотвращение перегрузок (Congestion Avoidance)

В жизни любого маршрутизатора наступает момент, когда очередь переполняется. Куда положить пакет? Если положить его решительно некуда, то это означает, что буфер кончился и не будет возможности найти для него место.

В данном случае есть два пути: отбросить либо этот пакет, либо те, что уже заняли очередь. В первом случае если отбрасываем те пакеты, что уже в очереди, то безвозвратно теряем их. В другом случае теряем отправленный пакет. Эти два подхода называются **Tail Drop** и **Head Drop**.

Tail Drop — наиболее простой механизм управления очередью — отбрасываем все вновь пришедшие пакеты, не помещающиеся в буфер.

Head Drop отбрасывает пакеты, которые стоят в очереди уже очень долго. Их уже лучше выбросить, чем пытаться спасти, потому что они, скорее всего, бесполезны. Зато у более актуальных пакетов, пришедших в конец очереди, будет больше шансов прийти вовремя. Плюс к этому Head Drop позволит не загружать сеть ненужными пакетами. Естественным образом самыми старыми пакетами оказываются те, что в самой голове очереди, откуда и название подхода.

2.3 Управление перегрузками (Congestion Management)

Когда всё плохо, приоритет обработки следует отдать более важному трафику. Важность каждого пакета определяется на этапе классификации. Необязательно все буферы должны быть забиты, чтобы приложения начали испытывать проблемы. Самый простой пример — голосовые пакетики, которые толпятся за большими пачками крупных пакетов приложения, скачивающего файл.

Это увеличит задержку, испортит джиттер и, возможно, вызовет отбрасывания. То есть мы имеем проблемы с обеспечением качественных услуг при фактическом отсутствии перегрузок.

Эту проблему призван решить механизм управления перегрузками (Congestion Management). Трафик разных приложений разделяется по очередям, как мы уже видели выше. Вот только в результате всё снова должно слиться в один интерфейс. Сериализация всё равно происходит последовательно.

Каким же образом разным очередям удаётся предоставлять различный уровень сервисов? По-разному изымать пакеты из разных очередей. Занимается этим диспетчер:

1. **FIFO** - только одна очередь, все в BE, C - несправедливость.

Простейший случай, по сути отсутствие QoS, - весь трафик обрабатывается одинаково - в одной очереди. Пакеты уходят из очереди ровно в том порядке, в котором они туда попали, отсюда и название: **первым вошёл - первым и вышел**.

2. **PQ** - дорогу олигархам, холопы уступают.

Второй по сложности механизм и попытка разделить сервис по классам - **приоритетная очередь**. Трафик теперь раскладывается в несколько очередей согласно своему классу - приоритету (например, хотя не обязательно, те же BE, AF1-4, EF, CS6-7). Диспетчер перебирает одну очередь за другой.

3. **FQ** - все равны.

Следующий механизм - **механизмы честных очередей**. Честность заключается в том, что диспетчер оперирует числом **не пакетов, а числом битов**, которые можно передать из каждой очереди.

4. **DWRR** - все равны, но некоторые ровнее.

Каждая очередь имеет отдельную **кредитную линию** в битах. При проходе из очереди выпускается столько пакетов, на сколько хватает кредита. Из суммы кредита вычитается размер того пакета, что в голове очереди. Если разность больше нуля, этот пакет изымается, и проверяется следующий. Так до тех пор, пока разность не окажется меньше нуля.

3. ХОД ИССЛЕДОВАНИЯ

3.1 Получение исходных данных

В данной работе технология QoS использовалась в домашних условиях

Порядок включения QoS на роутере:

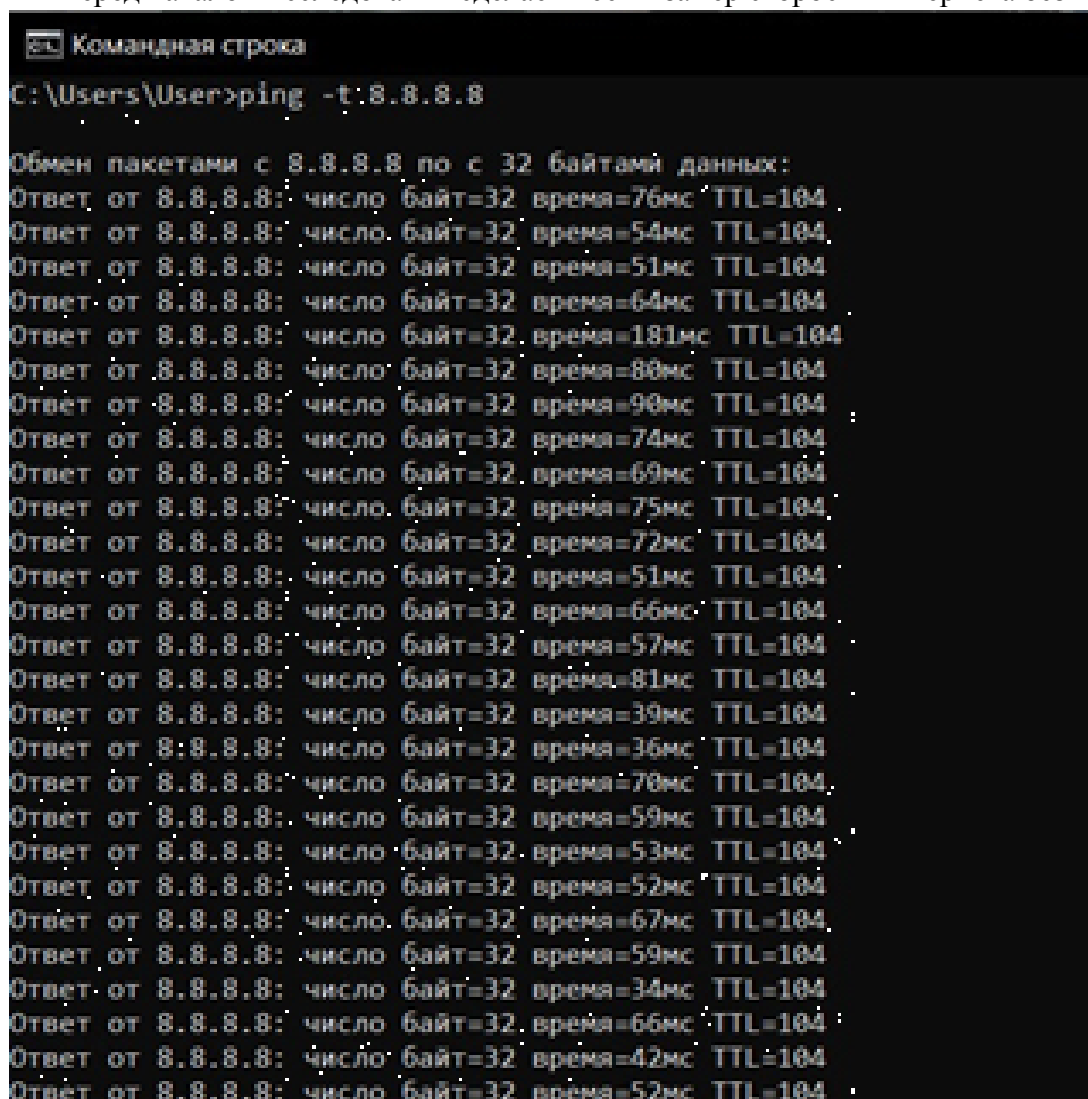
1. Открыть панель администрирования.
2. В адресную строку браузера ввести IP-адрес маршрутизатора.
3. Войти под персональным именем и паролем (указано в руководстве к маршрутизатору).
4. После входа выбрать вкладку «NAT», затем «QoS».
5. Нажать «Включить», оставив порт установленным на WAN, а Packet Scheduler и Queuing Discipline в состоянии по умолчанию.
6. Заполнить значения восходящую и нисходящую линии связи (вводные данные составляют 80-95% значения, полученного в ходе теста скорости).

В начале откроем командную строку и введем команду:

```
ping - 8.8.8.8
```

данная утилита помогает проверить связность сети, в нашем случае с DNS сервером Google.

Перед началом исследования сделаем тест – замер скорости интернета без включения QoS:



```
Командная строка
C:\Users\User>ping -t:8.8.8.8

Обмен пакетами с 8.8.8.8 по 32 байтами данных:
Ответ от 8.8.8.8: число байт=32 время=76мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=54мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=51мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=64мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=181мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=80мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=90мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=74мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=69мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=75мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=72мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=51мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=66мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=57мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=81мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=39мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=36мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=70мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=59мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=53мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=52мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=67мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=59мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=34мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=66мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=42мс TTL=104
Ответ от 8.8.8.8: число байт=32 время=52мс TTL=104
```

Рисунок 1.1 – отправка пакетов на DNS сервер Google без использования QoS


```
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=11мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=15мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Ответ от 8.8.8.8: число байт=32 время=6мс TTL=109
Статистика Ping для 8.8.8.8:
  Пакетов: отправлено = 50, получено = 50, потеряно = 0
  (0% потерь)
Приблизительное время приема-передачи в мс:
  Минимальное = 5мсек, Максимальное = 15 мсек, Среднее = 6 мсек
C:\Users\USER>
```

Рисунок 1.2 – отправка пакетов на DNS сервер Google с использованием QoS

Можем, заметить, посмотрев на рисунок 1.2, что скорость интернета возросла многократно, скорость скачков в сети стала практически не заметна.

3.2 Представление выборки

Выпишем время отправки пакетов для дальнейшего анализа:

1.	5	26.	6
2.	5	27.	6
3.	5	28.	6
4.	5	29.	6
5.	5	30.	6
6.	5	31.	6
7.	6	32.	6
8.	6	33.	6
9.	6	34.	6
10.	6	35.	6
11.	6	36.	6
12.	6	37.	6
13.	6	38.	6
14.	6	39.	6
15.	6	40.	7
16.	6	41.	7
17.	6	42.	7
18.	6	43.	8
19.	6	44.	9
20.	6	45.	9
21.	6	46.	9
22.	6	47.	10
23.	6	48.	11
24.	6	49.	14
25.	6	50.	15

3.3 Статистический анализ выборки

Разобьем все элементы на интервалы, найдем их середины, количество элементов в каждом интервале.

Вычислим:

Размах вариации R

$$R = x_{max} - x_{min}$$

Длину интервала h :

$h = \frac{R}{i}$, где h – количество интервалов, в нашем случае их шесть.

Частоты p_i^*

$$p_i^* = \frac{n_i}{n}$$

Накопленные частоты z_i

$$z_1 = p_1^*; z_2 = p_1^* + p_2^*; \dots; z_i = \sum_k^i p_k$$

Эмпирическую функцию распределения $F_n^*(x)$

$$F_n^*(x) = \sum_{i: \bar{a}_i < x} p_i^*$$

$$x_{max} = 181$$

$$x_{min} = 34$$

$n = 50$

Выборочное среднее \bar{x}

$$\tilde{x} = \frac{\sum x_i}{n}$$

Выборочная дисперсия S^2

$$s^2 = \sum_{i=1}^l x_i^2 p_i^2 - \bar{x}^2$$

Выборочное СКВО S

$$s = \sqrt{s^2}$$

Исправленная выборочная дисперсия \bar{s}^2

$$\bar{s}^2 = \frac{n}{n-1} * s^2$$

Исправленное выборочное СКВО \bar{s}

$$\bar{s} = \sqrt{\bar{s}^2}$$

Выборочная медиана μ

$$\mu = \begin{cases} x_k + 1 & n = 2k + 1 \\ \frac{x_k + x_{k+1}}{2} & n = 2k \end{cases}$$

Выборочная асимметрия A

$$A = \frac{1}{\bar{s}^3} \sum_{i=1}^l (\tilde{x}_i - \bar{x})^3 p_i^*$$

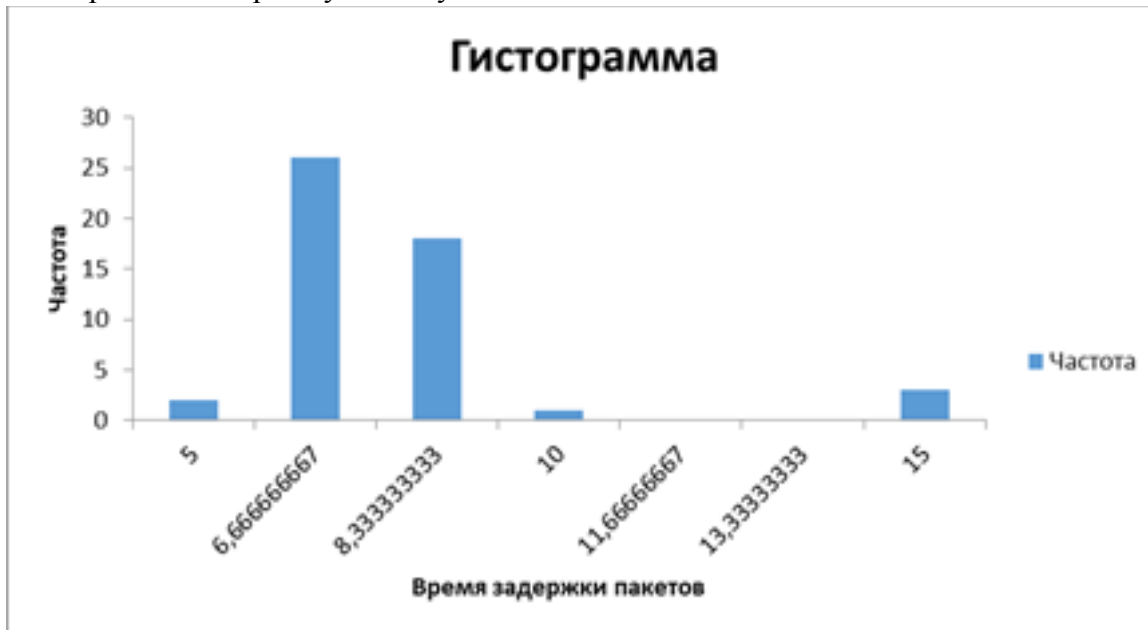
Выборочный эксцесс E

$$E = \frac{1}{\bar{s}^4} \sum_{i=1}^l (\tilde{x}_i - \bar{x})^4 p_i^* - 3$$

I_i	I_1	I_2	I_3	I_4	I_5	I_6	Σ
x_i	46,25	70,75	95,25	119,75	144,25	168,75	—
n_i	28	18	1	0	0	3	50
p_i	0,56	0,36	0,02	0	0	0,06	1
z	0,56	0,92	0,94	0,94	0,94	1	—
$F_n^*(x)$	0,022857143	0,014693878	0,000816327	0	0	0,002449	—
Выборочное среднее \tilde{x}							63,4
Выборочная дисперсия S^2							870,3625
Исправленная выборочная дисперсия \bar{s}^2							888,125
Исправленное выборочное СКВО \bar{s}							29,8014261
Выборочная медиана μ							54
Выборочная асимметрия A							2,5736866
Выборочный эксцесс E							9,45887796

<i>Карман</i>	<i>Частота</i>
5	2
6,6666667	26
8,3333333	18
10	1
11,6666667	0
13,3333333	0
15	3

Строим гистограмму по получившимся значениям:



На гистограмме видим распределение с изолированным пиком. На ряду с обычным распределением любого типа по одну сторону от распределения находится маленький пик. Причиной появления данного пика может служить ошибка выборки, но в нашем случае это невозможно, изменение времени задержки сети может быть связано с повышением нагрузки на сеть, но благодаря использования технологии QoS в нашей сети выровнялось время задержки.

4. ЗАКЛЮЧЕНИЕ И ВЫВОДЫ

На примеры статистических оценок и наглядного примера удостоверились в работе технологии QoS, в ее эффективном применении для стабильной работы сети, успешно проанализировали время задержек и узнали распределение по времени. Технология QoS помогает поддерживать производительность сети, гарантировать бесперебойную передачу трафика, а также регулярно оценивать состояние IT-инфраструктуры сети. QoS очень сильно повышает качество и снижает время задержки во время интернет трансляций и других мероприятий требующих большого расхода трафика.

5. ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. QoS-Enabled Networks: Tools and Foundations (Wiley Series on Communications Networking & Distributed Systems) 2nd Edition.
2. ТЕОРИЯ ВЕРОЯТНОСТИ И МАТЕМАТИЧЕСКАЯ СТАТИСТИКА Руководство по решению задач Н.В. Васильева.
3. Special Topic – QoS (HUAWEI TECHNOLOGIES CO., LTD.)
4. QoS от Huawei: Special Edition QoS(v6.0)
5. QoS в MPLS: MPLS and Quality of Service