

# 1. Постановка задачи, обзор существующей литературе по теме

## 1.1. Постановка задачи

### 1.1.1. Актуальность задачи

Передача данных в компьютерных на сегодняшний день является основополагающим способом коммуникации не только людей, но и филиалов предприятий, государственных и военных структур. Важной задачей становится обеспечить должное качество и повысить вероятность передачи данных или в случае протоколов с гарантированной доставкой уменьшить количество попыток передать данные. Для этого выгодно обеспечивать на каждом промежуточном пункте передачи данных наибольшую вероятность того, что передача пройдет успешно. Однако в силу физической ограниченности канала связи, временных затрат на обработку пакетов и отсутствия бесконечно больших буферов в промежуточных узлах возникают проблемы, приводящие к потере данных. Такие узлы на пути трафика называют "узким местом" (bottleneck). Они сильно влияют на работу сети, из-за чего нужно уметь оперативно решать проблемы на подобных узлах для повышения эффективности всей сети.

### 1.1.2. Определение проблемы

Устранить проблемы на медленных участках сети можно с помощью установки нового и более мощного оборудования, однако это дорогостоящее решение и не всегда возможное. Другим вариантом является возможность делить трафик на классы и устанавливать требования к сети для каждого класса в отдельности. С помощью такой методики появляется возможность понизить вероятность потери существенных пакетов, которые могут являться причиной высокой нагрузки сети, и обеспечить должное обслуживание каждого типа трафика в соответствии с требованиями.

Однако для поддержания такой возможности системы ядра IP-сети должны обладать возможностью дифференцирования и обслуживания различных типов сетевого трафика в зависимости от предъявляемых ими требований, однако Негарантированная доставка данных не предполагает прове-

дения какого-либо различия между потоками информации в сети, что препятствует передаче трафика, требующего выделения заданного минимума сетевых ресурсов и гарантии предоставления определенных услуг. Для решения описанной выше проблемы и было введено такое понятие, как качество обслуживания (quality of service – QoS) в сетях IP. QoS предоставляет набор требований, предоставляемых к ресурсам сети при транспортировке потока данных. [1] Для учёта этих требований в узлах сети используются алгоритмы обслуживания пакетов, называемые дисциплины обслуживания очередей (queuing disciplines, qdisc, ДО). Конфигурация qdisc влияет на то, как выбираются пакеты из очередей пакетов (packet queues), какие пакеты отбрасываются, как обрабатывается тот или иной тип трафика и так далее.[2]

Из вышесказанного следует, что дисциплины обслуживания очередей играют большую роль в эффективной работе узла. Значит, чем более гибкая ДО используется, чем больше нюансов трафика учитывает, тем точнее и эффективнее будут обрабатываться пакеты, что повысит производительность узла и, следовательно, всей системы.

### ***1.1.3. Цель работы***

Цель данной работы заключается в реализации пакетного планировщика, который будет предоставлять гибкие возможности для настройки управления трафиком и его классификации. Платформой для реализации пакетного планировщика был выбрано Linux, так как на его основе существует большое количество дистрибутивов, которые используются в качестве сетевого ПО (Red Hat Enterprise Linux, Debian, Ubuntu Server, OpenWRT).

## **2. Сравнительный анализ алгоритмов qdisc**

### **2.1. Дисциплины обслуживания очередей**

**Дисциплина обслуживания очередей (queuing discipline, qdisc)** – это алгоритм выбора и обработки пакета из очереди пакетов, где под очередью подразумевается буфер пакетов.

ДО бывают следующих видов.

**Приоритетные** – ДО, которые назначают для разных видов трафика

**Бесприоритетные** – ДО, которые не приоритезируют трафик. К таким ДО относится, к примеру, fifo.

**Классовые**

**Бесклассовые**

## 2.2. Алгоритмы, реализованные в ядре Linux

### 2.3. Бесклассовые алгоритмы (Classless)

- **CODEL** (controlled-delay active queue management)
- **FIFO** (first in first out)
- **FQ** (fair queue)
- **RED** (random early detection)
- **SFB** (stochastic fair blue)

### 2.4. Классовые алгоритмы (Classful)

- **CBQ** (class based queueing) – обработка пакетов в очередях
- **HFSC** (heirarchy fair service curve's control) –
- **HTB** (heirarchy token bucket) –

## 3. Формальное описание выбранного алгоритма и его реализация

## 4. Заключение

## 5. Список использованной литературы

1. Шринивас Вегешна, Качество обслуживания в сетях IP, 2003
2. Иван Песин, Повесть о Linux и управлении трафиком,  
<http://computerlib.narod.ru/html/traffic.htm>