



Коммутация пакетов: оценка и управление задержкой

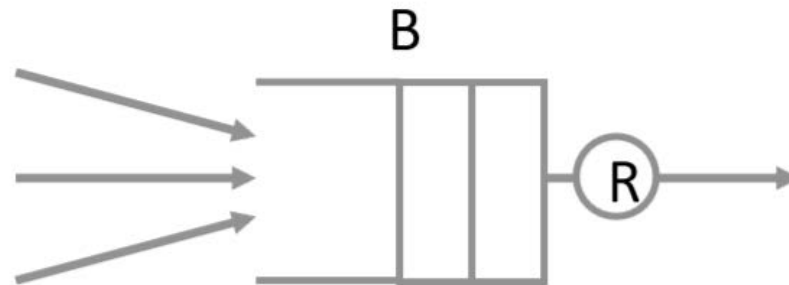
Введение в компьютерные сети

проф. Смелянский Р.Л.

Лаборатория Вычислительных комплексов

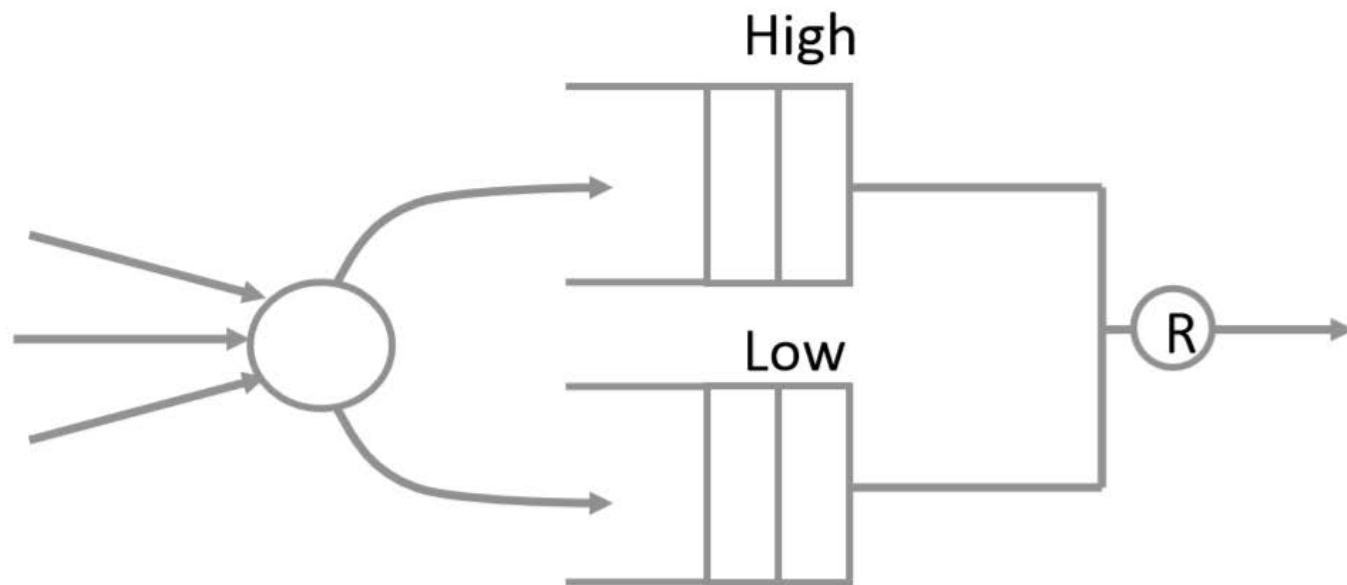
ф-т ВМК МГУ

Пакетный коммутатор: FIFO

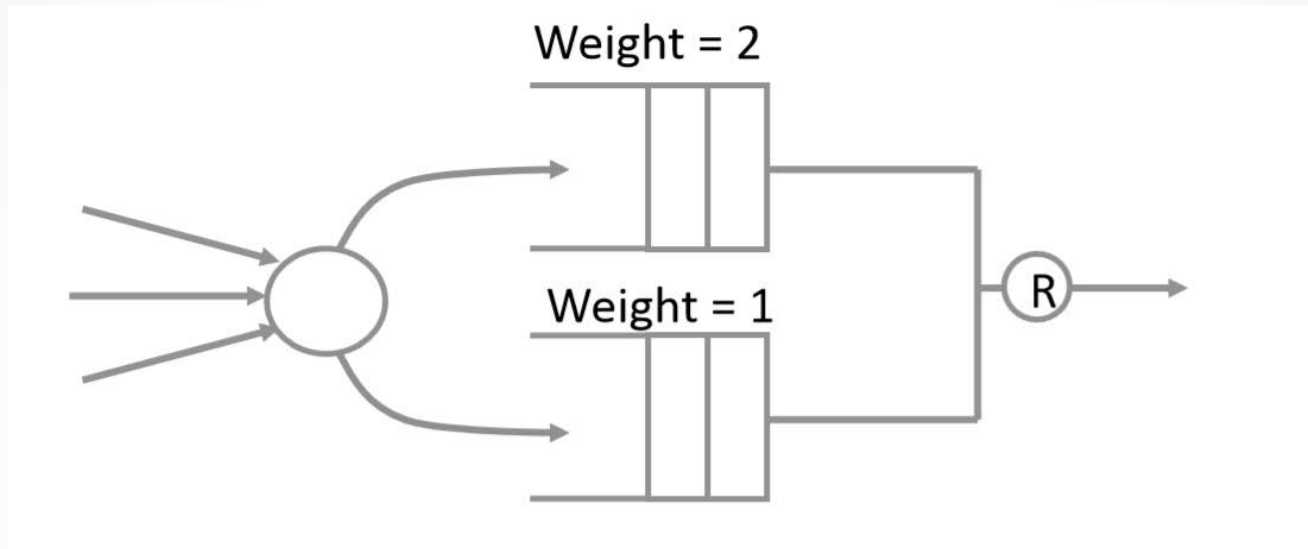


- Как регулировать качество сервиса (скорость)?

Пакетный коммутатор: строгие приоритеты



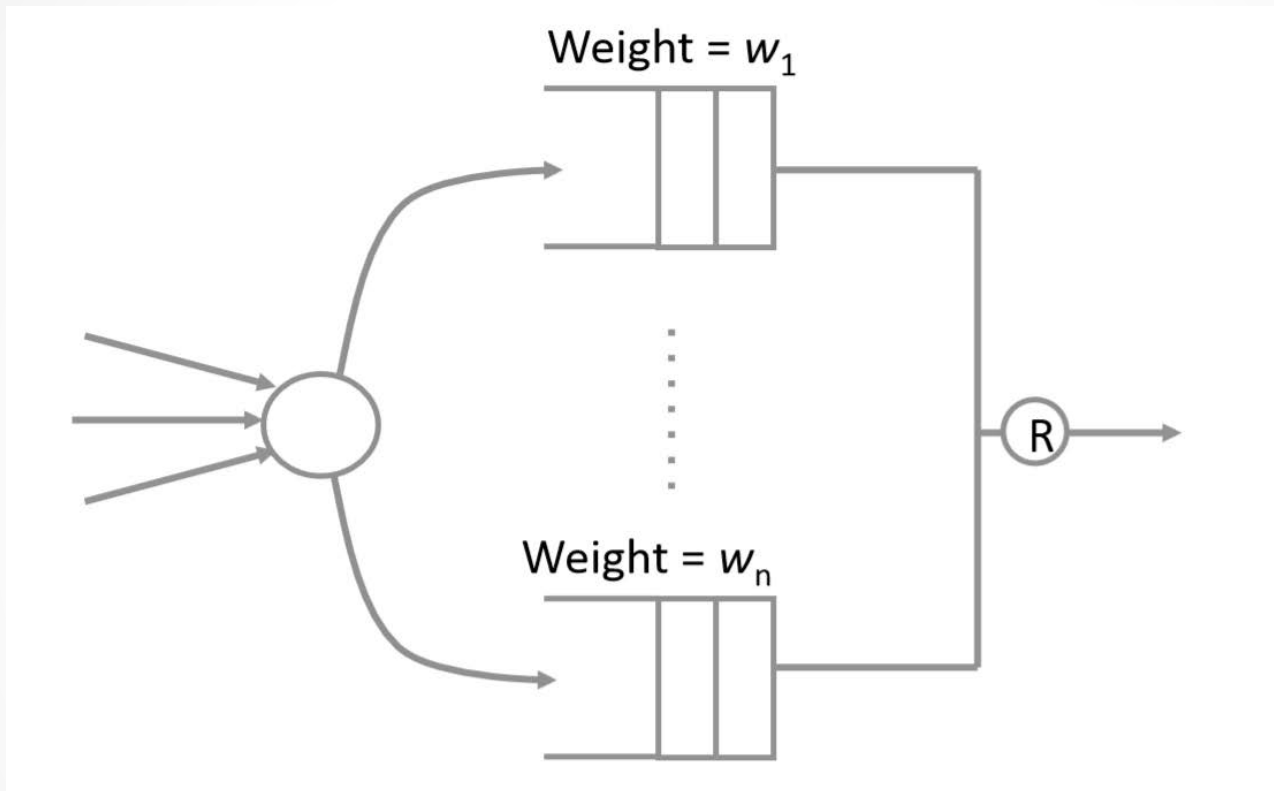
Пакетный коммутатор: приоритеты с весами



$$R_1 = R * 2/(2+1)$$

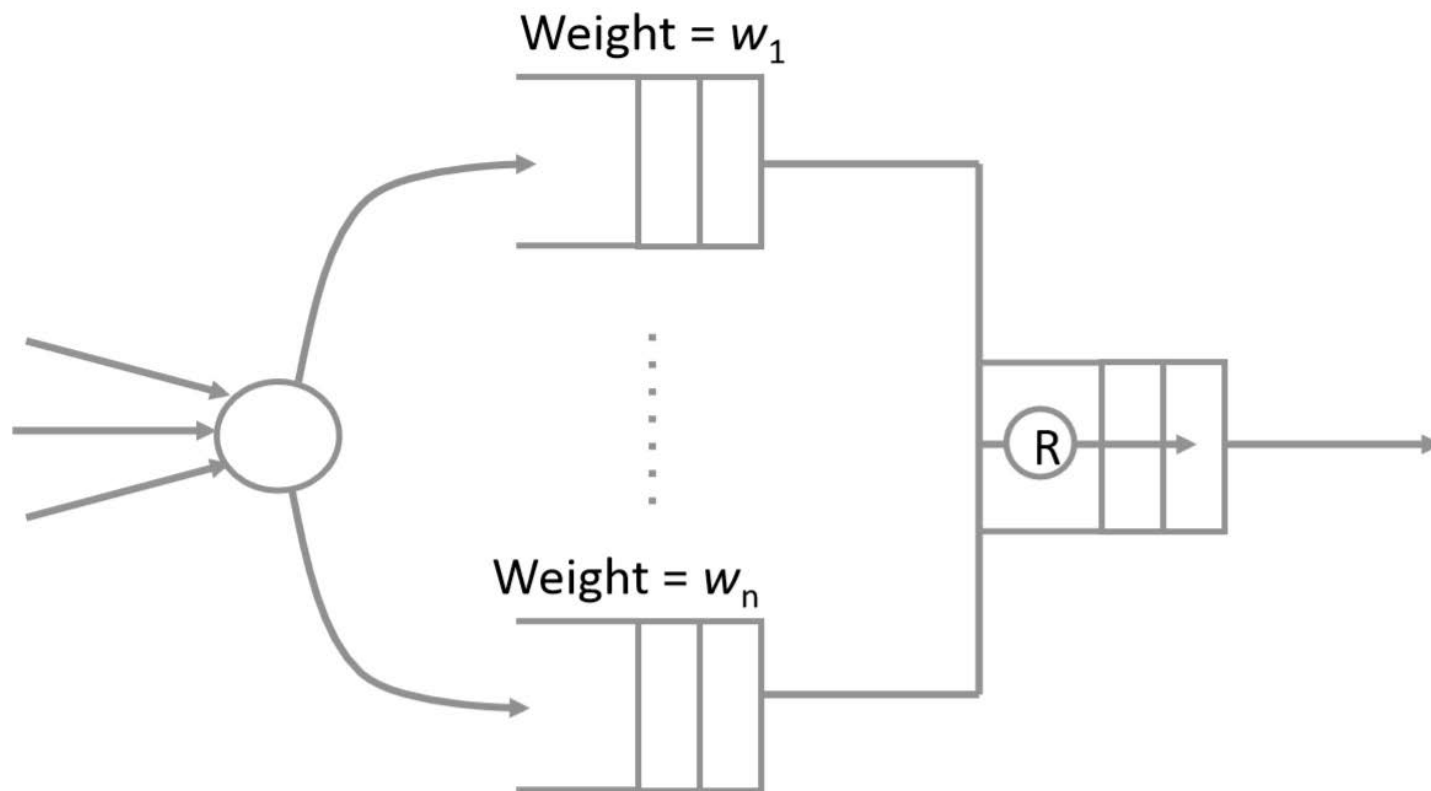
$$R_2 = R * 1/(2+1)$$

Пакетный коммутатор: приоритеты с весами

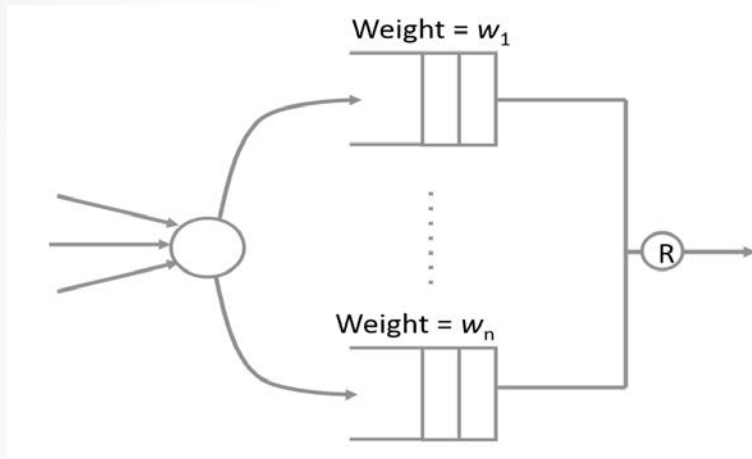


$\left(\frac{w_i}{\sum_{i=1}^n w_i} * R \right)$ - скорость обслуживания i -го потока;

Пакетный коммутатор: пакетное vs побитное обслуживание



Пакетный коммутатор: оценка задержки



Weighted Fair Queuing

$$FR_1 = SR_1 + L * \left(\frac{\sum_{i=1}^n w_i}{w_1 * R} \right) - \text{время окончания обслуживания}$$

L/w_1 - число раундов

$$\left(\frac{w_1}{\sum_{i=1}^n w_i} * R \right) - \text{скорость обслуживания;}$$
$$w_1 / \left(\frac{w_1}{\sum_{i=1}^n w_i} * R \right) - \text{время обслуживания на раунде}$$

$$SR = \max \{FR-1, \text{now}\}$$

$$FR = SR + L/w_i \text{ где } L \text{ длина пакета}$$

Выводы

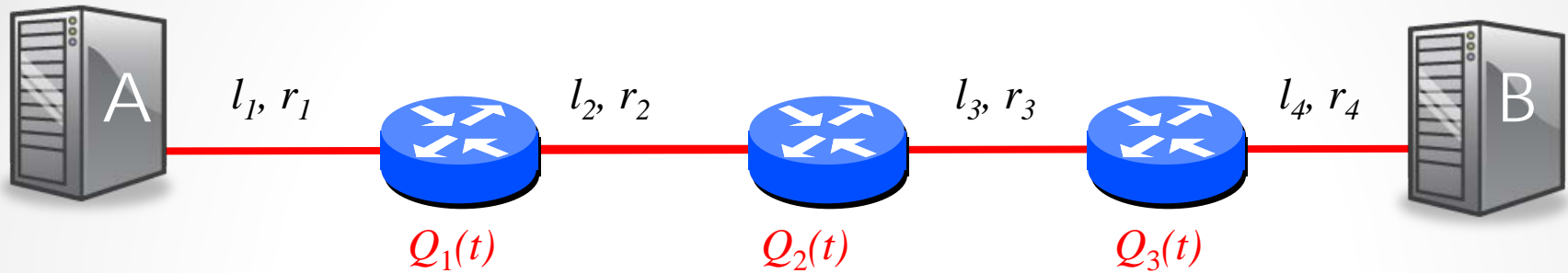
- FIFO очередь - нет приоритетов, не гарантирована скорость
- Строгие приоритеты: высокоприоритетный трафик «не видит» низкоприоритетного трафика в сети. Полезно, когда высокоприоритетного трафика ограниченное количество
- Waited Fair Queuing (WFQ) позволяет каждому потоку обеспечить гарантированный сервис, планируя их в порядке bit-by-bit finishing time



Коммутация пакетов: управление задержкой

Введение в компьютерные сети
проф. Смелянский Р.Л.
Лаборатория Вычислительных комплексов
ф-т ВМК МГУ

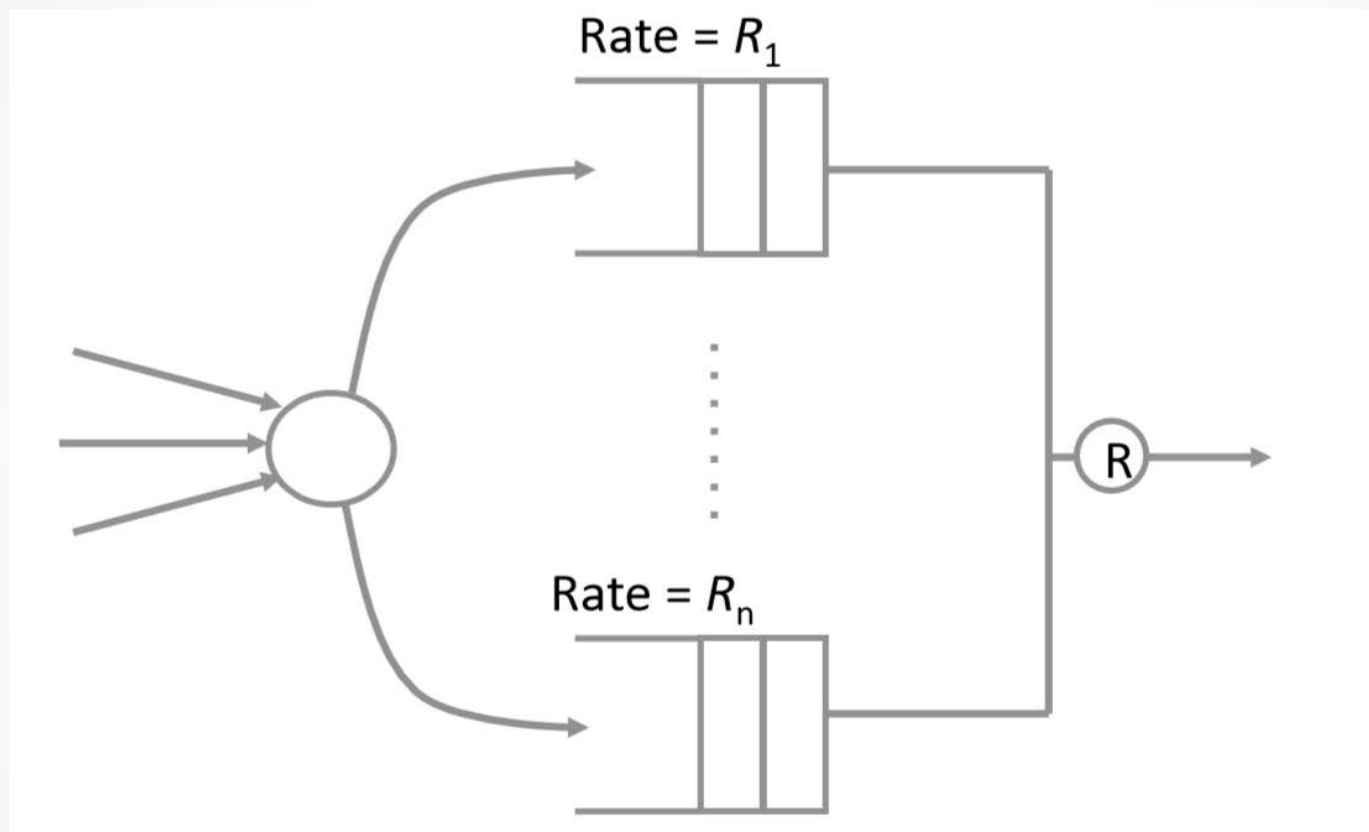
Факторы влияющие на задержку



$$\text{е2е задержка, } \tau = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$$

Если мы знаем верхние границы $Q_1(t), Q_2(t)$ and $Q_3(t)$, то мы знаем верхнюю границу для е2е задержки.

Пакетный коммутатор: гарантированная задержка





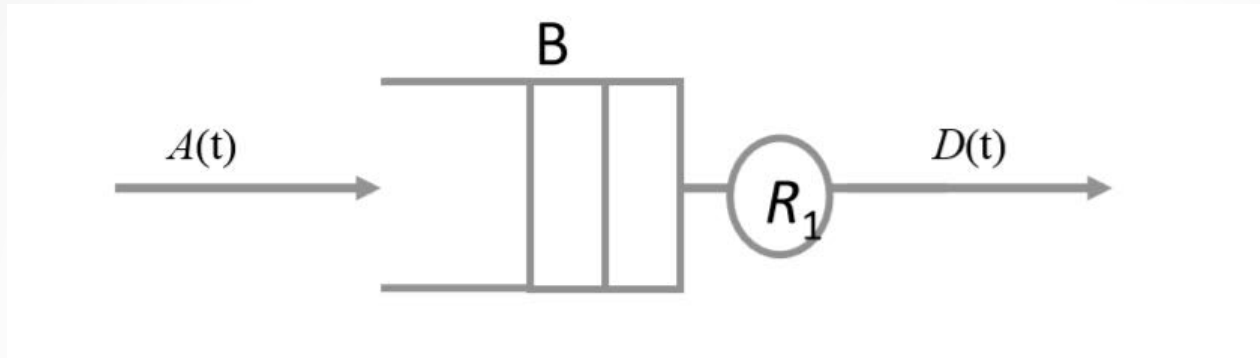
Пакетный коммутатор: управление задержкой пакета

Мы знаем уже как управлять

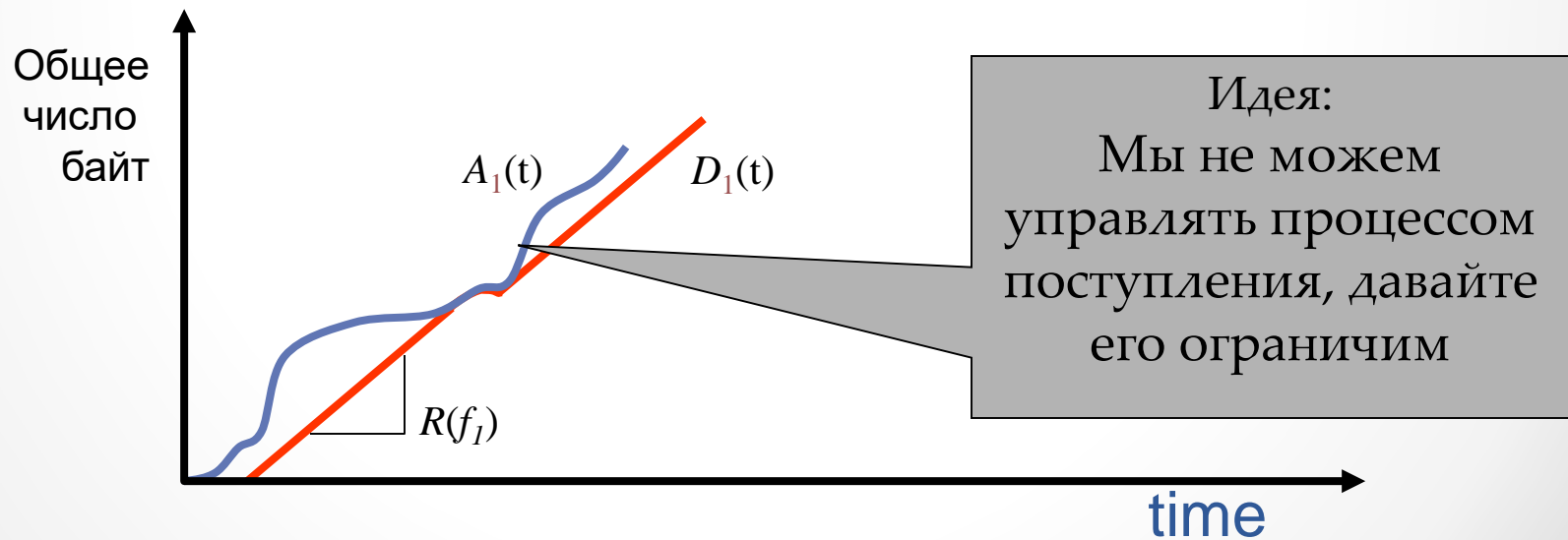
1. Скоростью обслуживания очереди (WFQ)
2. Длиной каждой очереди

Как можно было бы гарантировать,
что ни один пакет не будет сброшен ?

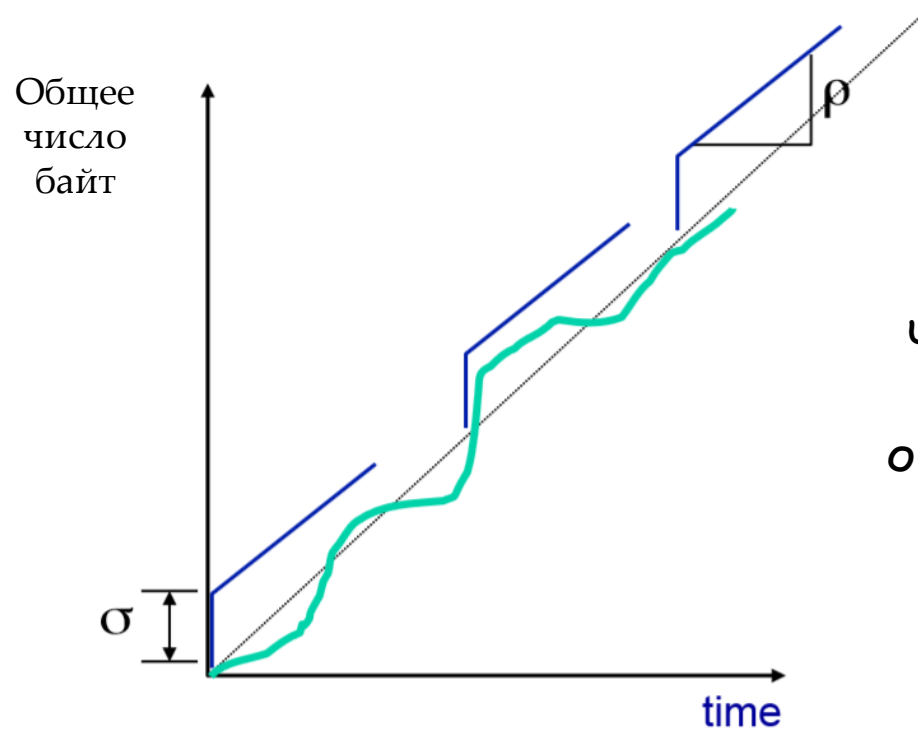
Пакетный коммутатор: zoom одной очереди



Пакетный коммутатор: zoom одной очереди

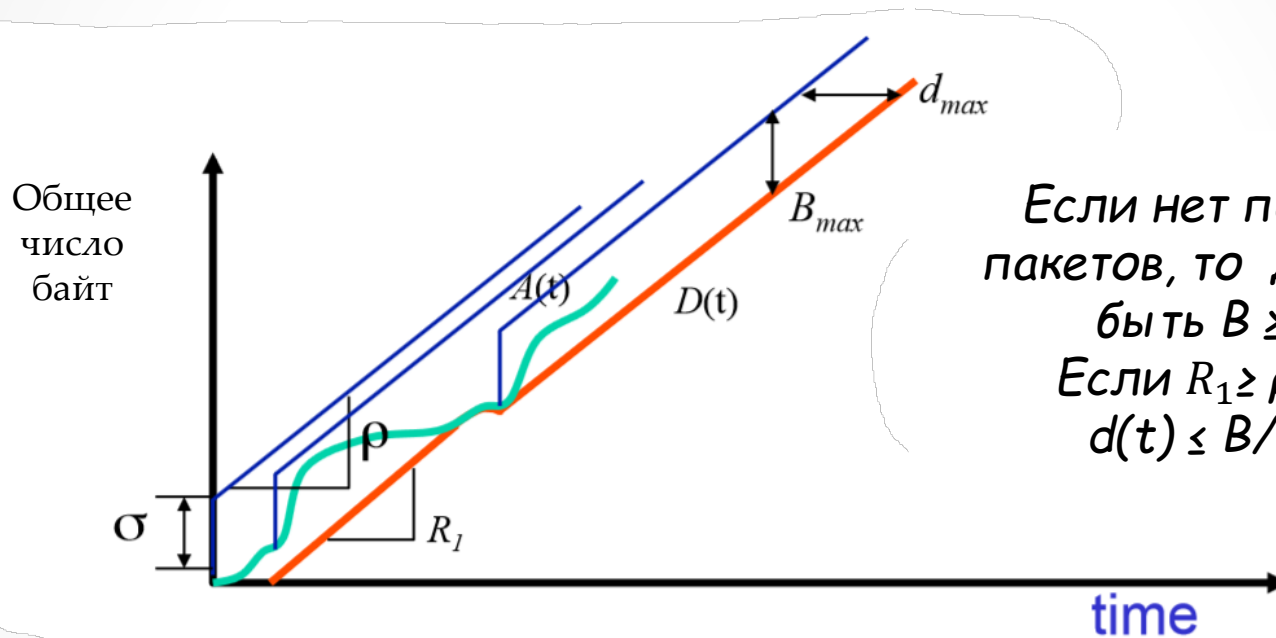


Ограничение трафика



Число бит, которые могут поступить за период t , ограничено величиной $\sigma + \rho t$
В нашем примере
 $\sigma = B, \rho = R_1$

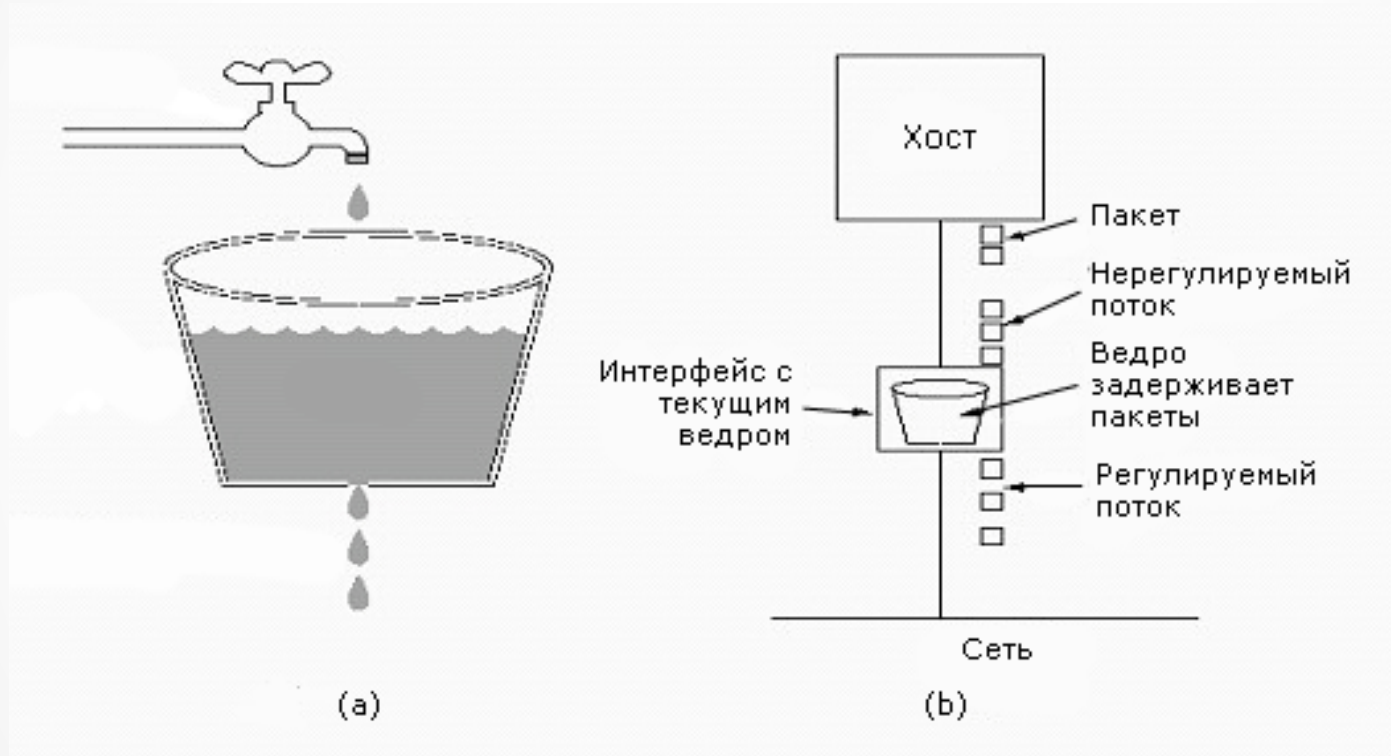
(σ, ρ) ограничение поступления и минимальная скорость обслуживания



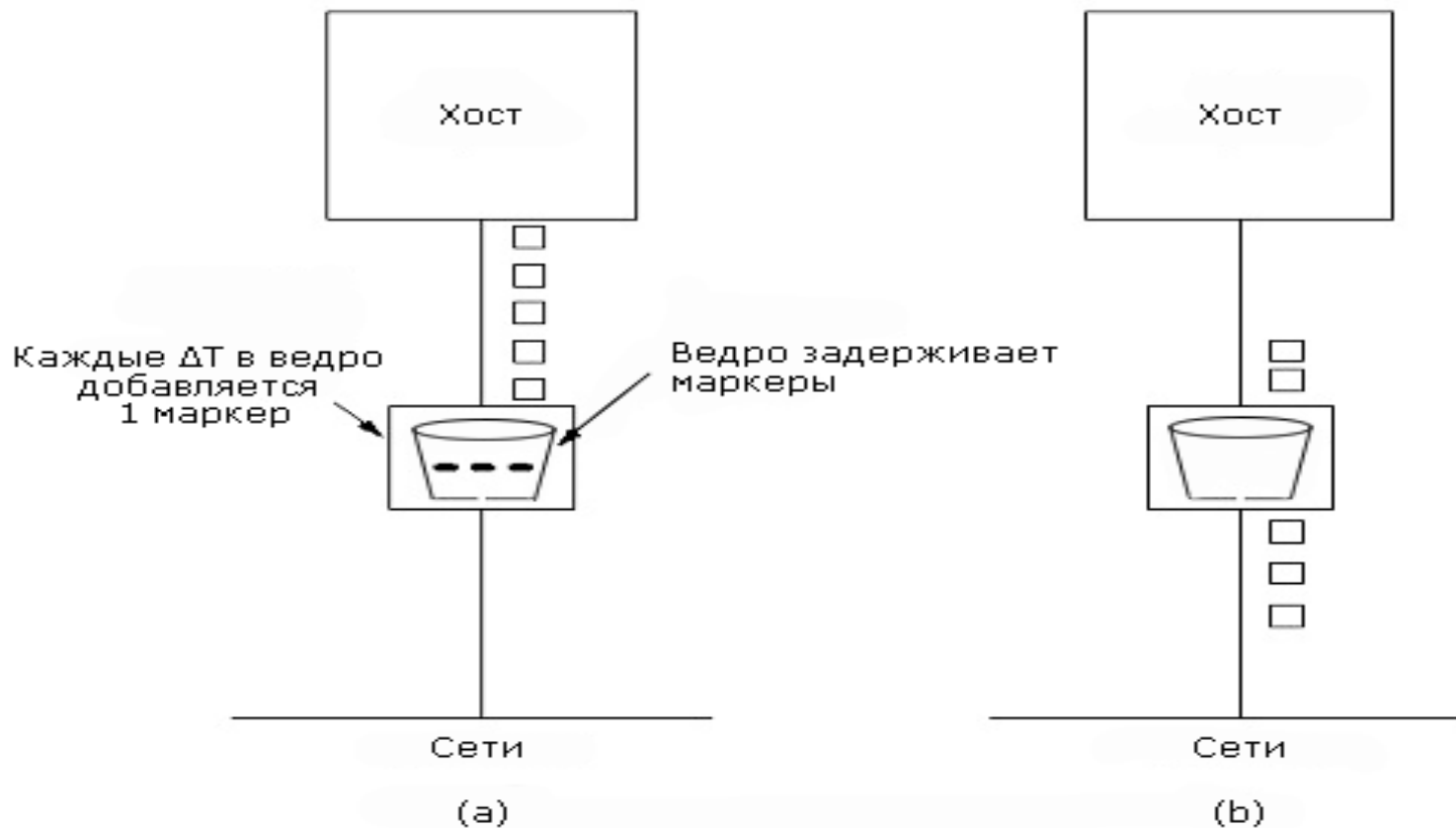
Если нет потери пакетов, то должно быть $B \geq \sigma$.
Если $R_1 \geq \rho$, то $d(t) \leq B / R_1$

Если поток ограничен текущим буфером и маршрутизатор использует дисциплину обслуживания WFQ, то можно гарантировать e^2e задержку.

Пакетный коммутатор: текущий буфер (идея)



Текущий буфер с маркерами



Текущий буфер с маркерами

- *Обозначим*

S - длительность временного увеличения трафика на входе;

ρ - скорость поступления маркеров Б/с;

M - максимальная скорость входного трафика Б/с;

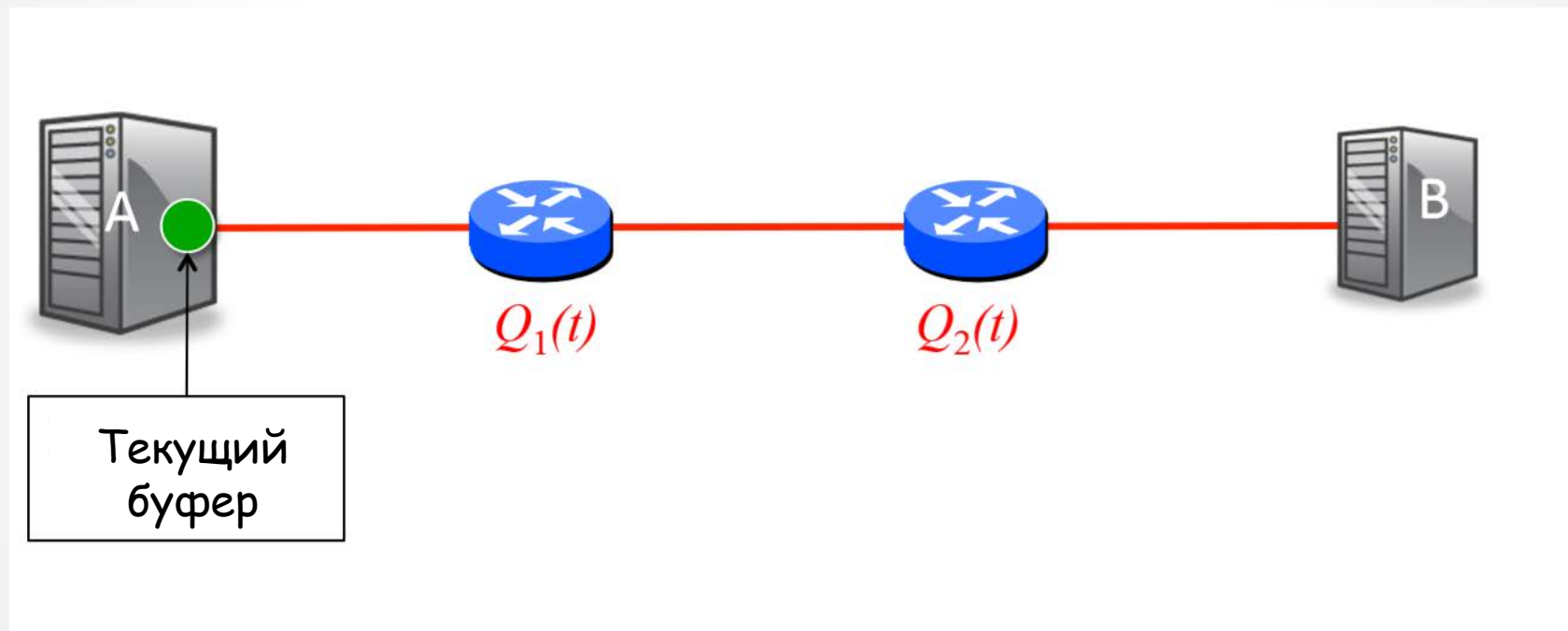
B - емкость буфера в байтах.

Тогда $B + \rho S = MS$.

Откуда получаем, что

$$S = B / (M - \rho).$$

Пакетный коммутатор: задержка



Пакетный коммутатор: пример

Для приведенного фрагмента сети приложению требуется скорость в 10 Мб/с и e2e задержка не более 5 мс при пакетах в 1 000 байт



задержка пакетизации и распространения получим 0.696 ms

$Q \leq 4.304$ ms так как по условию общая задержка не более 5 ms

разделим эту задержку равными долями между этими двумя маршрутизаторами (2.152 ms каждому)

$$B > 10 \text{ Mb/s} * 2.152 \text{ ms} = 21\,520 \text{ b} = 2690 \text{ B}$$

Заключение

- Если мы знаем длину очереди и дисциплину ее обслуживания, то мы можем ограничить величину задержки в ней.
- Выбрав длину очереди, и, используя WFQ, мы можем определить скорость обслуживания.
- Поэтому самое главное не допустить сброса пакетов. Для этого можно использовать метод «текущего ведра».
- Таким образом, мы можем ограничить величину e_2e задержки.

Несмотря на то, что технически это возможно, лишь некоторые сети могут управлять e_2e задержкой.

Причины:

- Слишком сложно и хлопотно
- В большинстве сетей комбинация прогнозирования и приоритетов дает вполне приемлемые результаты.