



Коммутация пакетов: задержки и буферизация

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

Кафедра АСВК ф-т ВМК МГУ

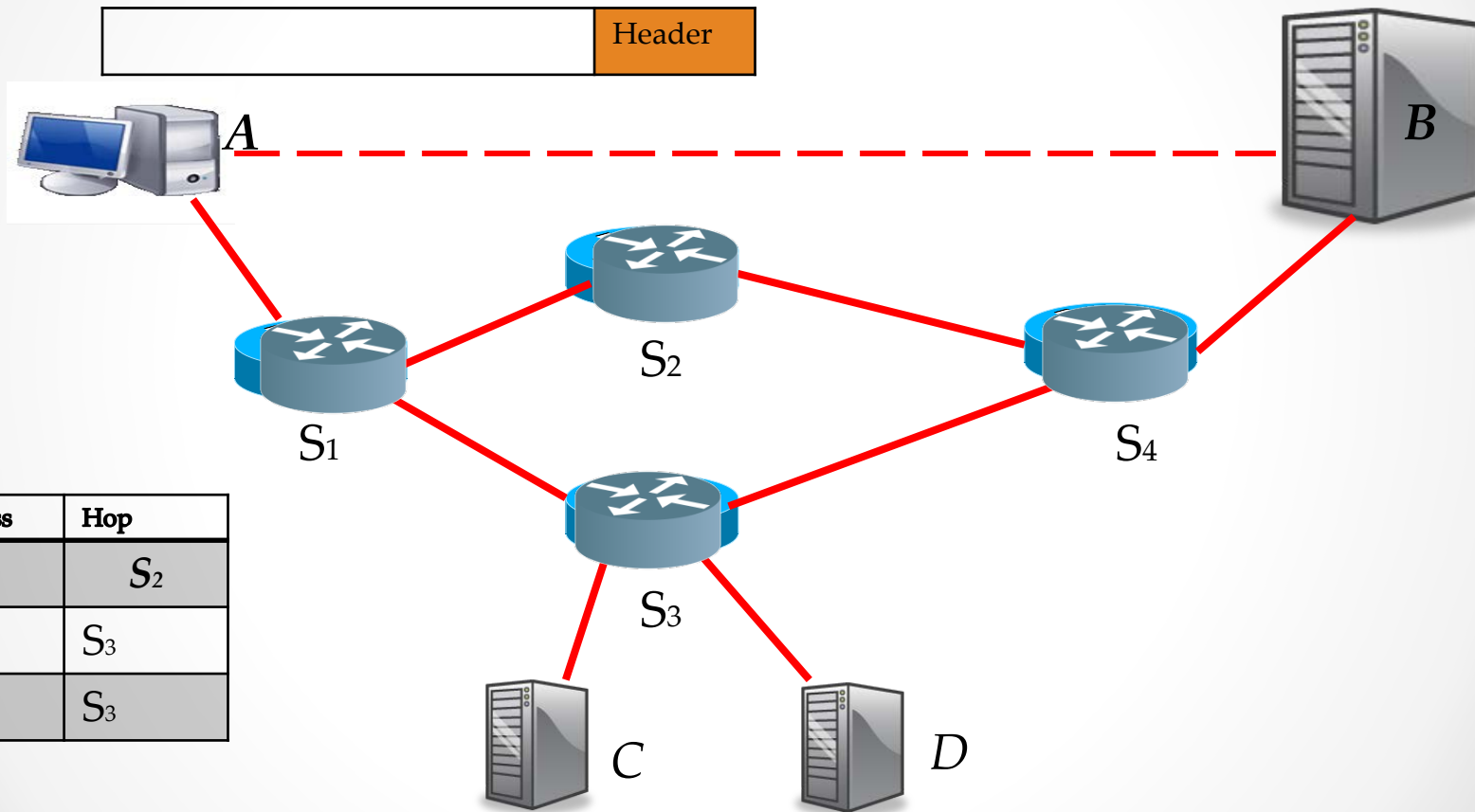


Содержание

- **Принцип коммутации пакетов**
- **Коммутация пакетов: виды задержек**
 - Задержка распространения
 - Задержка пакетизации
 - Сквозная задержка (end-to-end delay)
 - Задержка в очереди (Queuing delay)



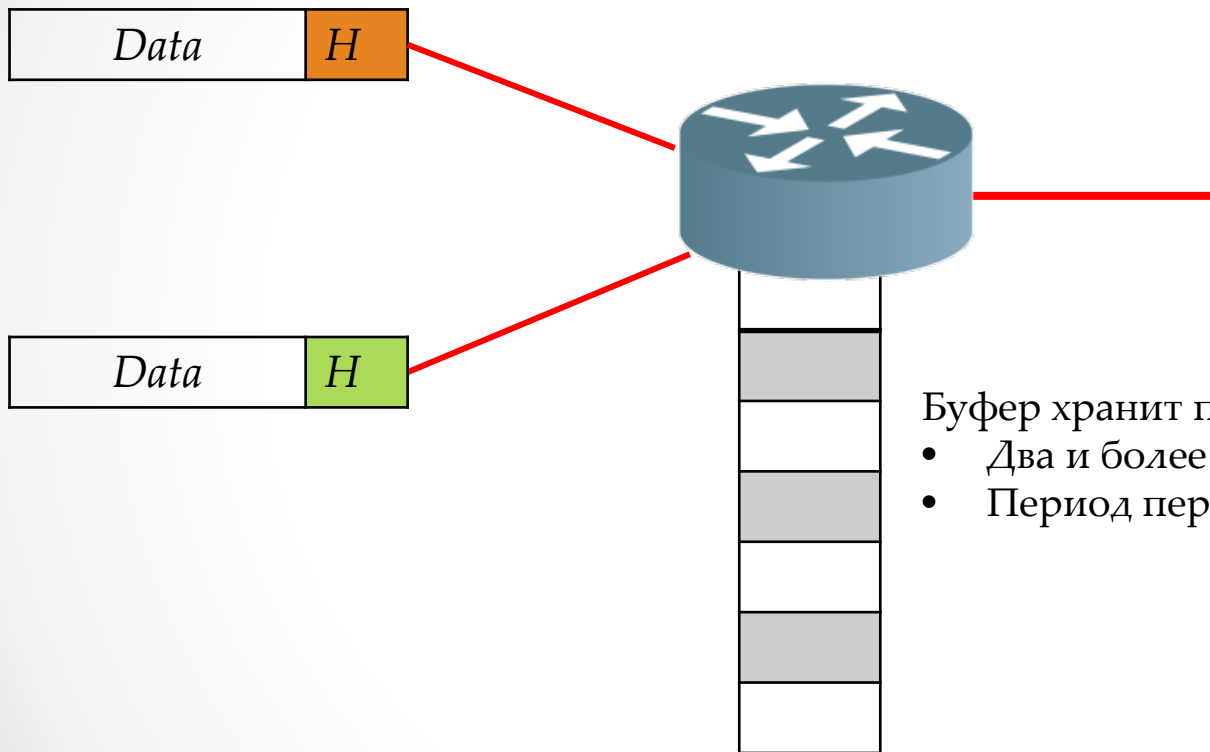
Коммутация пакетов



Address	Hop
<i>B</i>	<i>S₂</i>
<i>C</i>	<i>S₃</i>
<i>D</i>	<i>S₃</i>



Коммутация пакетов: буферизация



Буфер хранит пакеты:

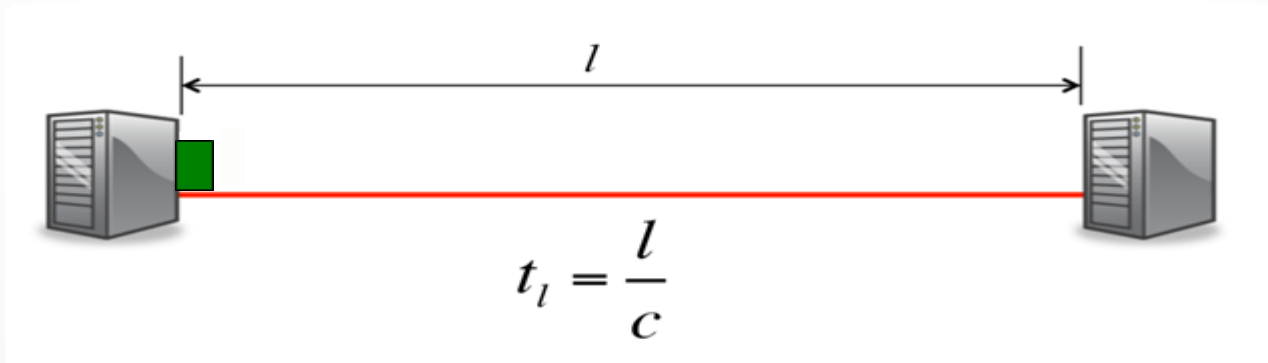
- Два и более пакеты пришли одновременно
- Период перегрузки



Коммутация пакетов: задержка распространения

Задержка распространения

t_l : время распространения одного бита по каналу длины l
со скоростью c

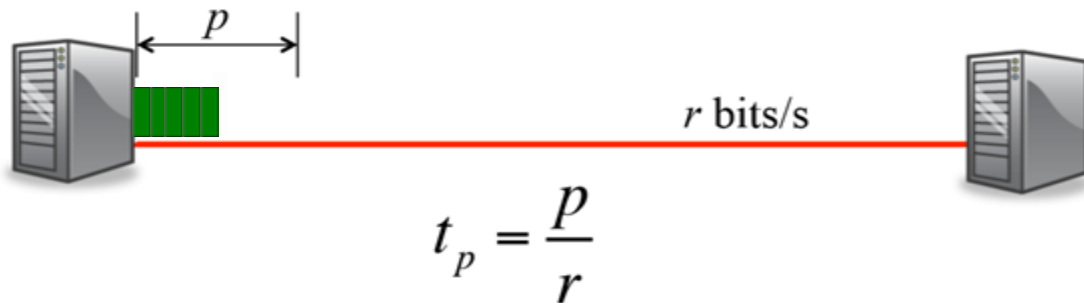


Пример: при $c = 2 \times 10^8$ м/с, распространение бита на 1 000 км занимает 5 μ с



Задержка пакетизации

Задержка пакетизации t_p : время, за которое биты пакета с первого до последнего переданы в канал

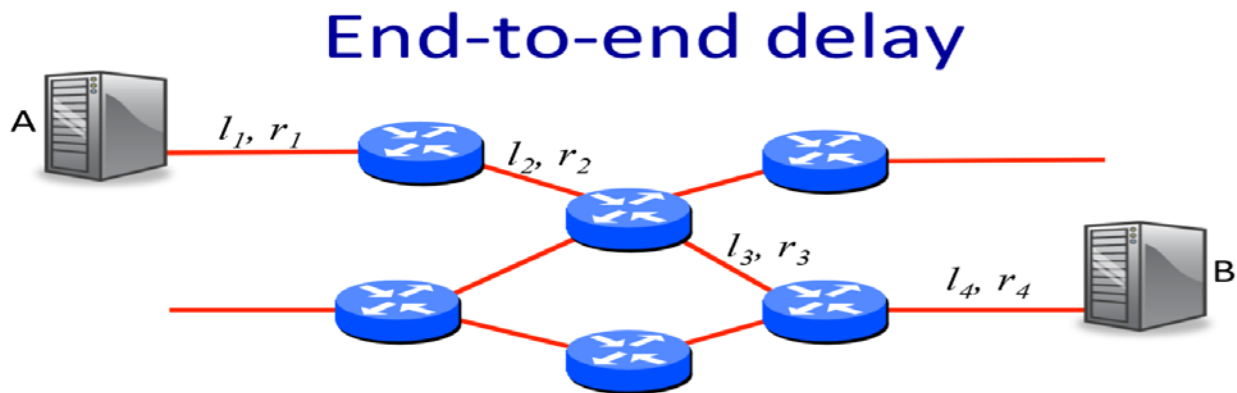


Пример 1: $p = 64$ Б пакет будет передан в канал с пропускной способностью $r = 100$ Мб/с за $5.12 \mu\text{s}$

Пример 2: 1Кб пакет будет передан в 1 Кб/с канал за 1.024 с.



Сквозная задержка (e2e delay)



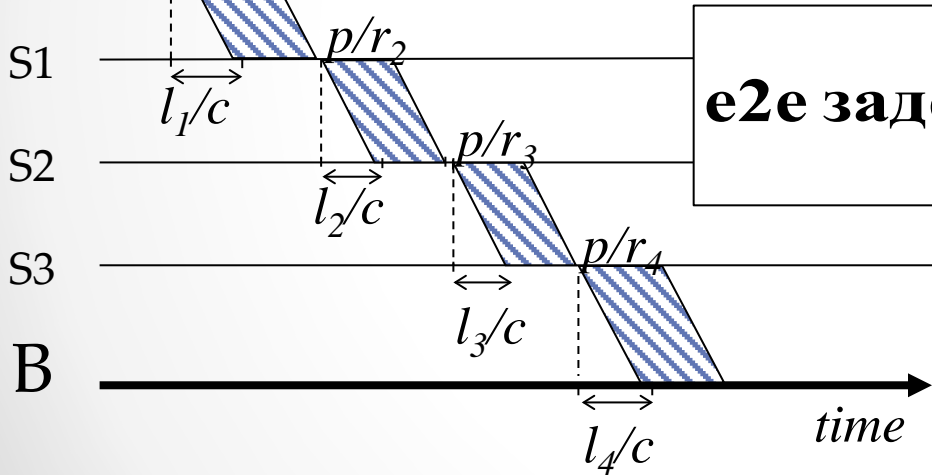
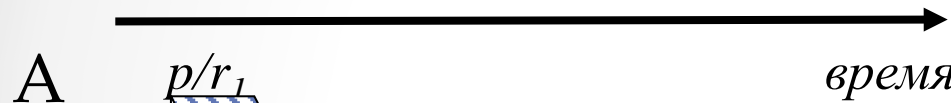
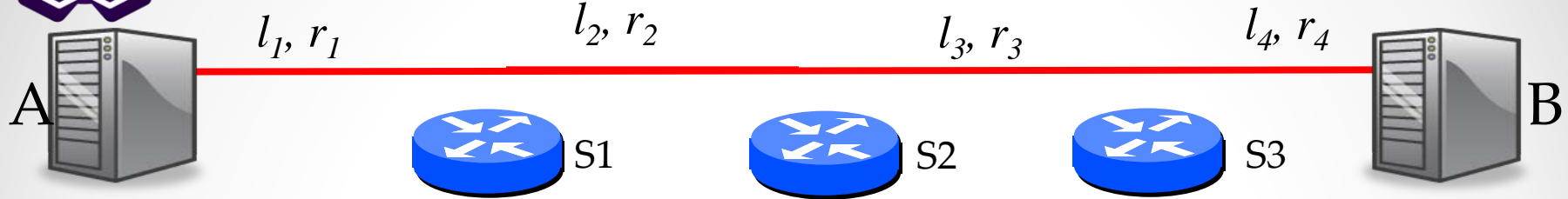
Пример: За какое время пакет длины p дойдет от A до B (от момента когда послан первый бит до момента когда придет последний бит). Предполагаем что коммутаторы принимают и сразу передают пакет

$$t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} \right)$$



Сквозная задержка (e2e delay)

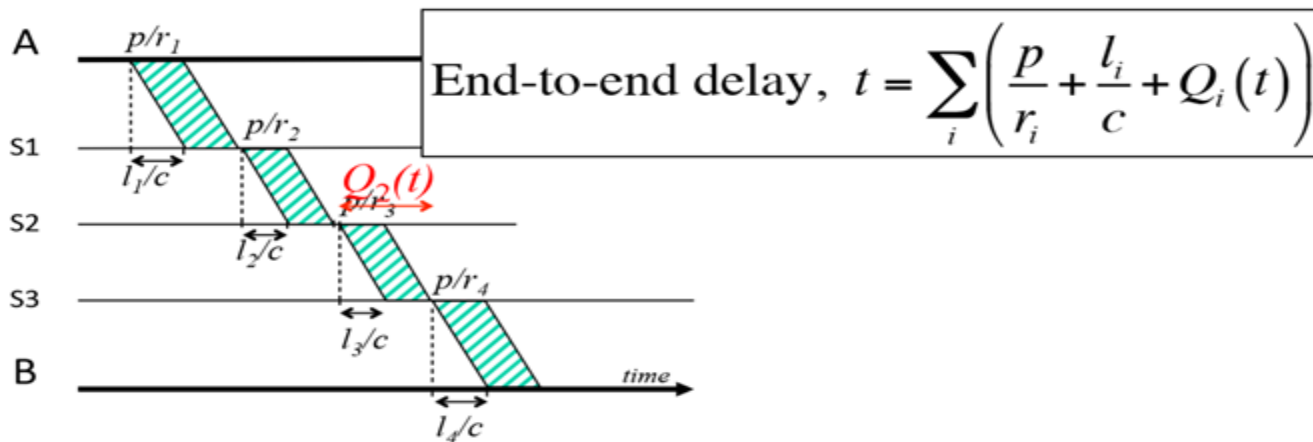
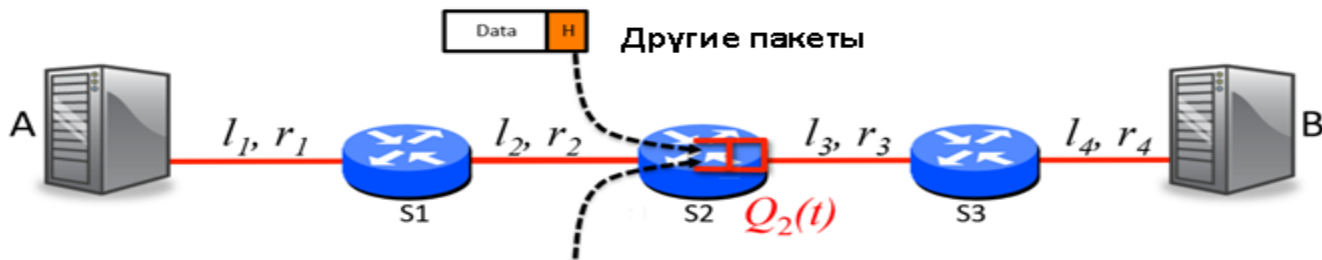
ОДИН ПОТОК



e2e задержка, $t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} \right)$

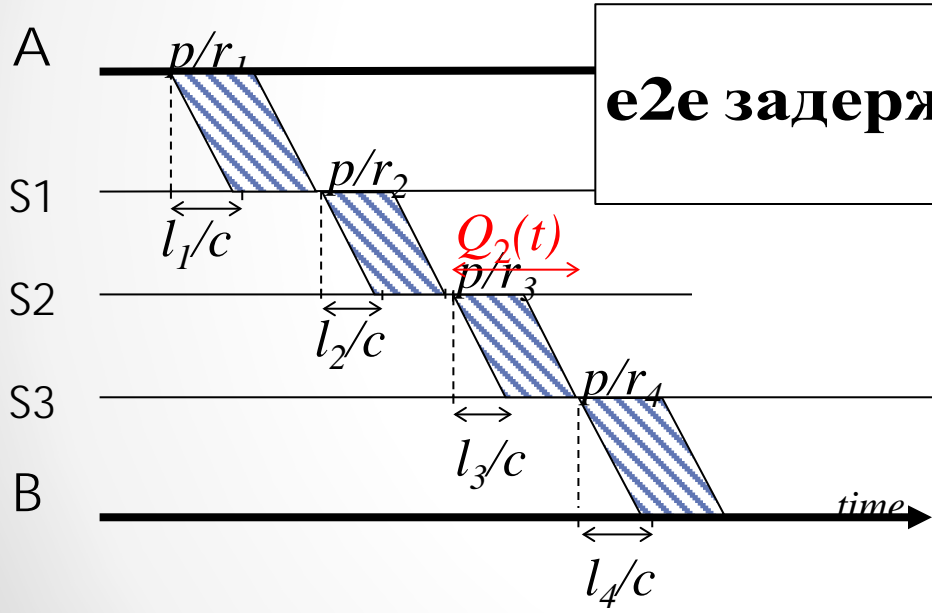
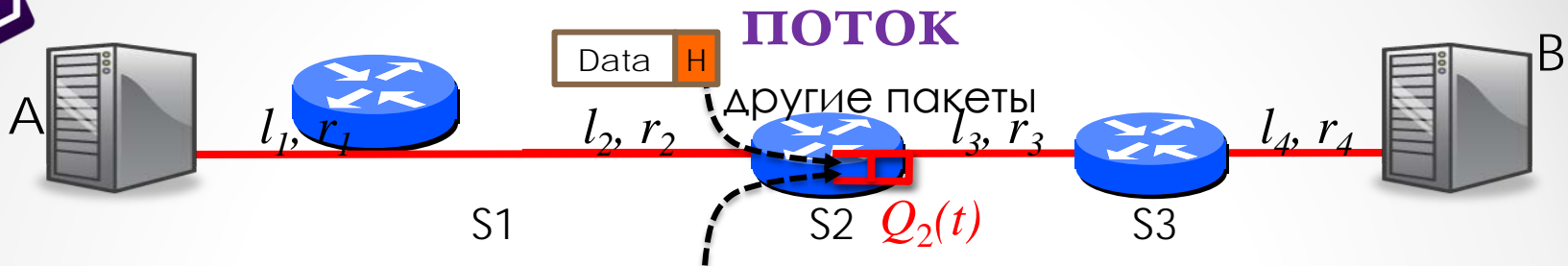


Сквозная задержка (e2e delay)



$$\text{End-to-end delay, } t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$$

Сквозная задержка (e2e delay) не один



e2e задержка, $t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$



Влияние буферизации на e2e задержку

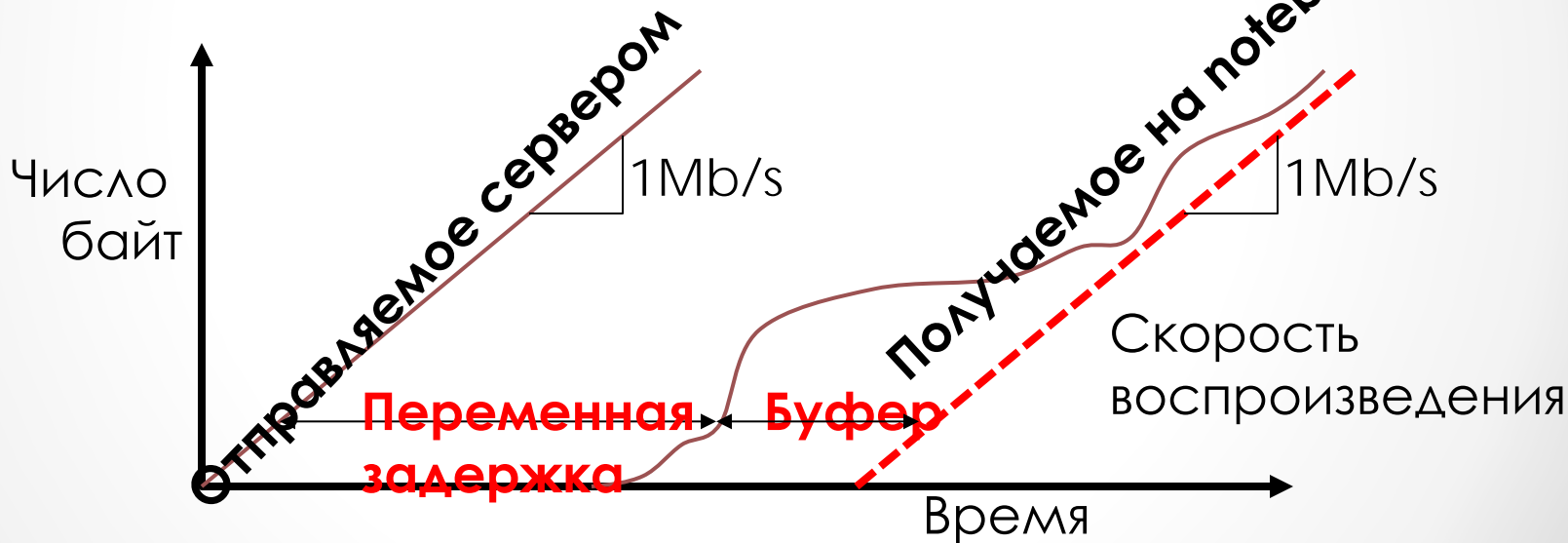
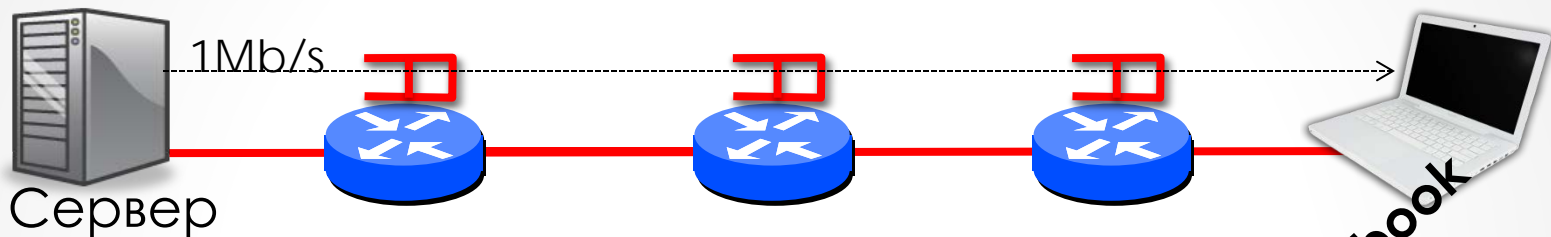
Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

Лаборатория Вычислительных комплексов
ф-т ВМК МГУ

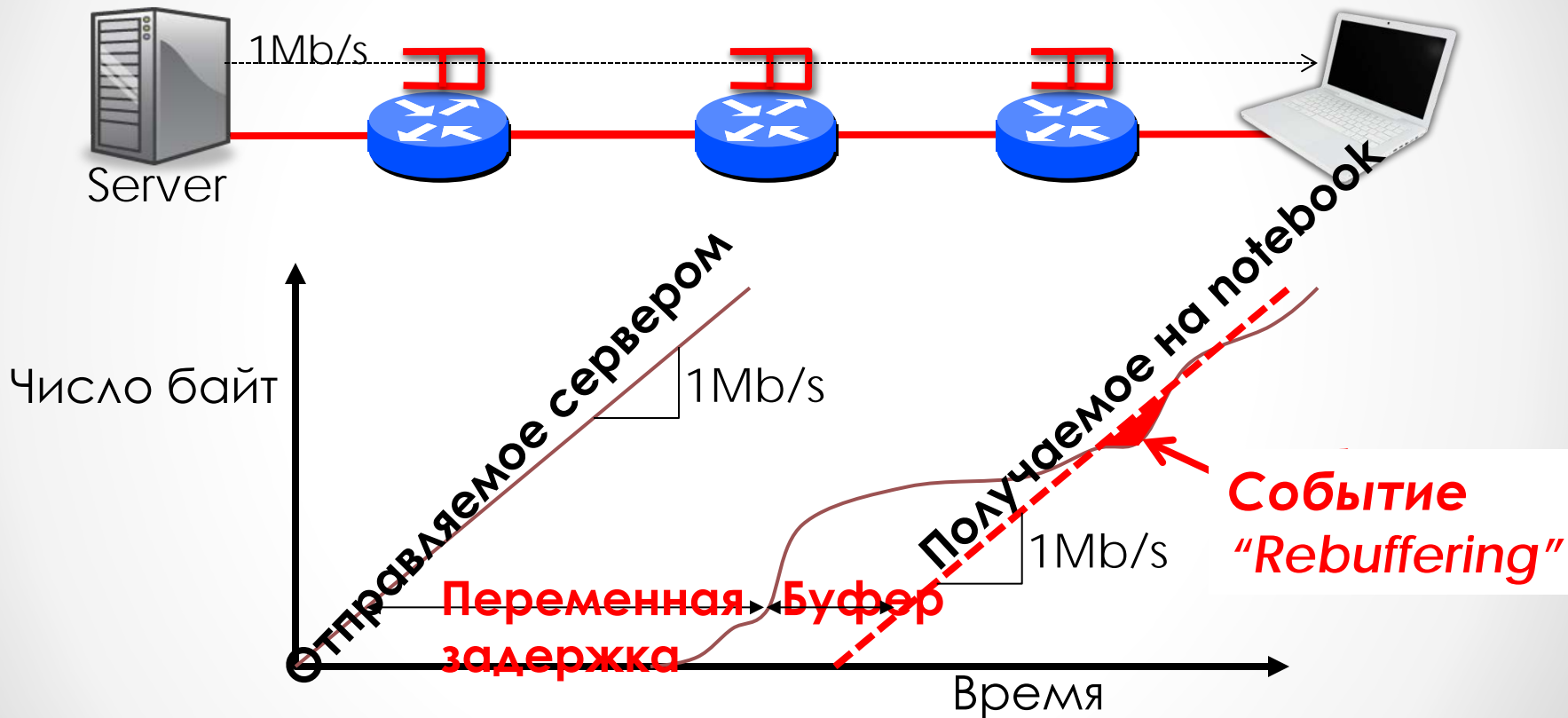


Влияние буферизации на e2e задержку





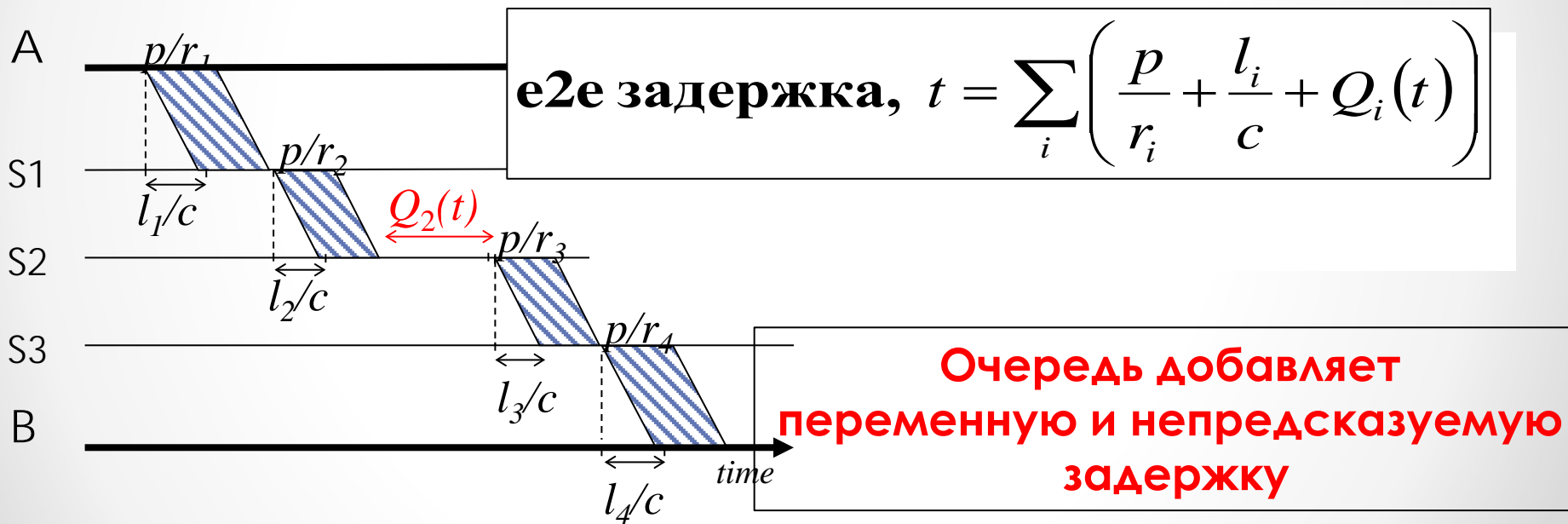
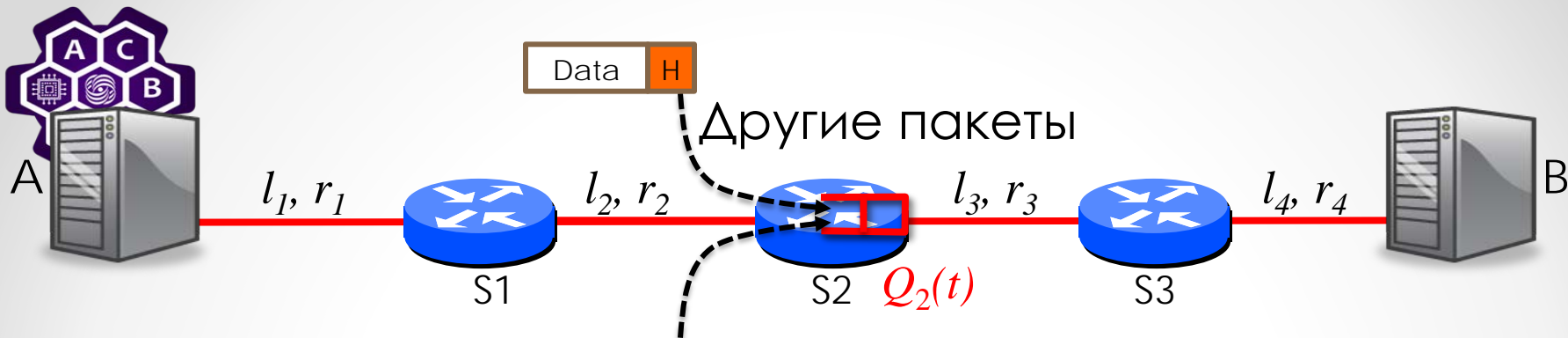
Влияние буферизации на e2e задержку





Влияние буферизации на e2e задержку

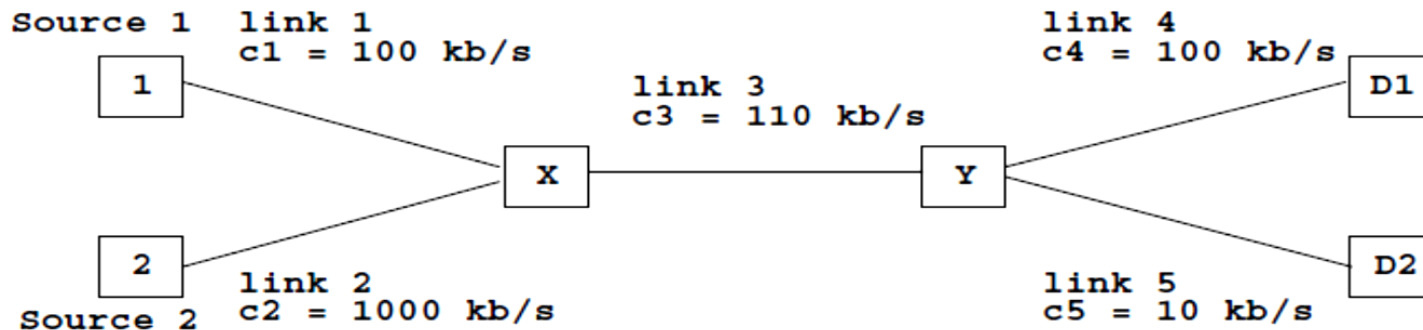
- При коммутации пакетов e2e задержка - величина изменчивая. Буферизация позволяет сгладить эти изменения.
- Можно делать буфер у получателя сразу большим, но тогда будет большая задержка на старте до его заполнения.
- Приложение должно оценивать задержку, устанавливать размер буфера воспроизведения, и изменять этот размер при изменении задержки.





Пример

Источники ничего не знают о распределении пропускных способностей каналов



$$\lambda_1 = 100\text{kb/s}$$

$$\lambda_2 = 1000\text{kb/s}$$

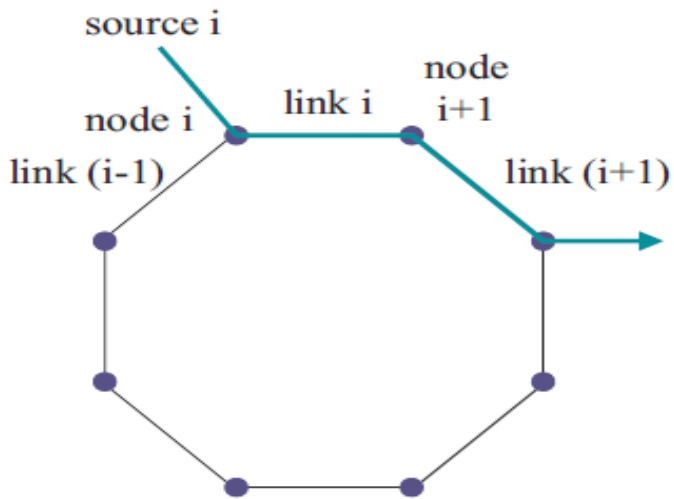
$$\lambda'_1 = \lambda'_2 = 10\text{kb/s} \text{ Общая пропускная способность} = 20\text{kb/s}$$

Вывод: отправитель должен знать, что у него впереди на пути



Пример

$$\begin{cases} \lambda'_i = \min \left(\lambda_i, \frac{c_i}{\lambda_i + \lambda'_{i-1}} \lambda_i \right) \\ \lambda''_i = \min \left(\lambda'_i, \frac{c_{i+1}}{\lambda'_i + \lambda_{i+1}} \lambda'_i \right) \end{cases}$$



$$\lambda' = \frac{c\lambda}{\lambda + \lambda'} \quad (*)$$

$$\lambda'' = \frac{c\lambda'}{\lambda + \lambda'}$$

$$\lambda' = \frac{\lambda}{2} \left(-1 + \sqrt{1 + 4\frac{c}{\lambda}} \right) \quad \lambda'' = c - \frac{\lambda}{2} \left(\sqrt{1 + 4\frac{c}{\lambda}} - 1 \right) \quad (**)$$

$$\sqrt{1 + u} = 1 + \frac{1}{2}u - \frac{1}{8}u^2 + o(u^2) \quad \text{при} \quad u \rightarrow 0$$

$$\lambda'' = \frac{c^2}{\lambda} + o\left(\frac{1}{\lambda}\right) \quad \text{При} \quad \lambda \rightarrow \infty, \lambda'' \rightarrow 0$$



Заключение

- Сквозная задержка состоит из трех компонентов:
 1. Задержка распространения
 2. Задержка пакетизации
 3. Задержка в буфере коммутатора
- Буферизация позволяет сгладить влияние задержки в очередях
- Важно уметь правильно оценить размер буфера.
- Без обратной связи не знаем как распределяются ресурсы на маршруте. В сети может наступить коллапс!
- Отправителю необходима обратная связь с сетью.



Коммутация пакетов: очереди и их свойства

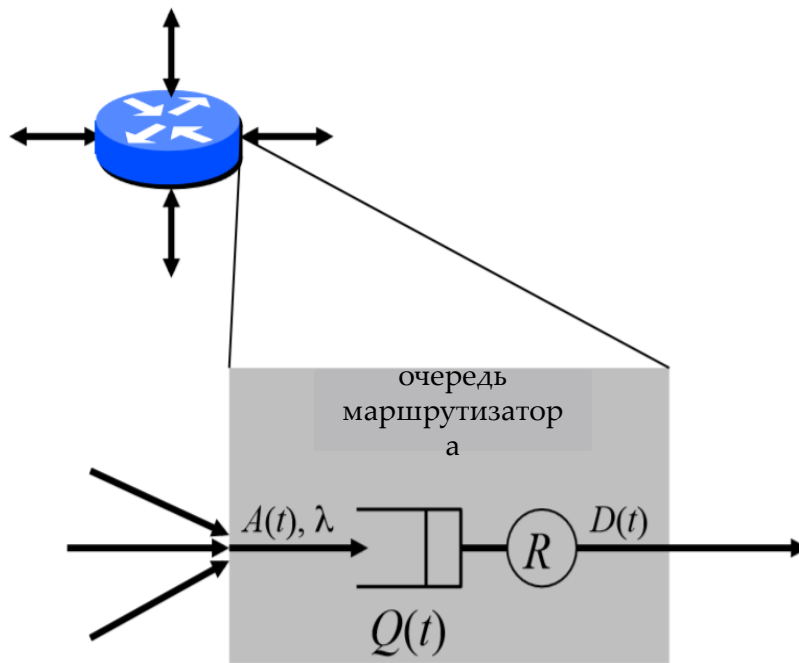
Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

Кафедра АСВК ф-т ВМК МГУ



Простая модель очереди -маршрутизатора

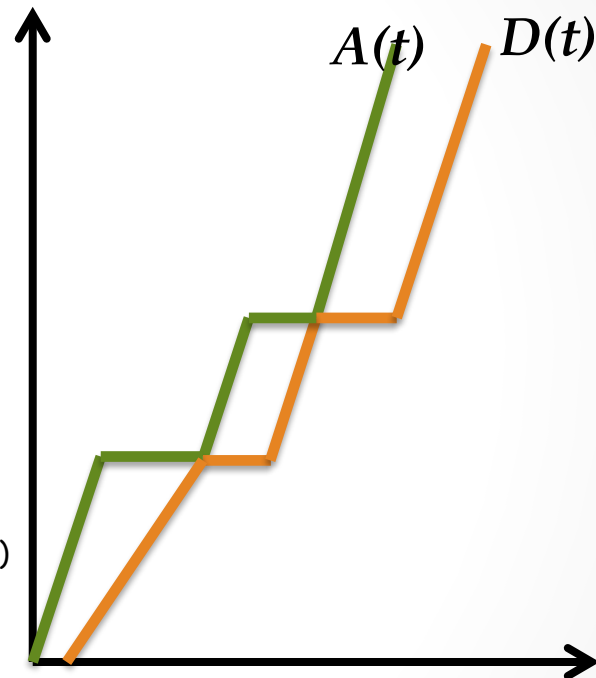
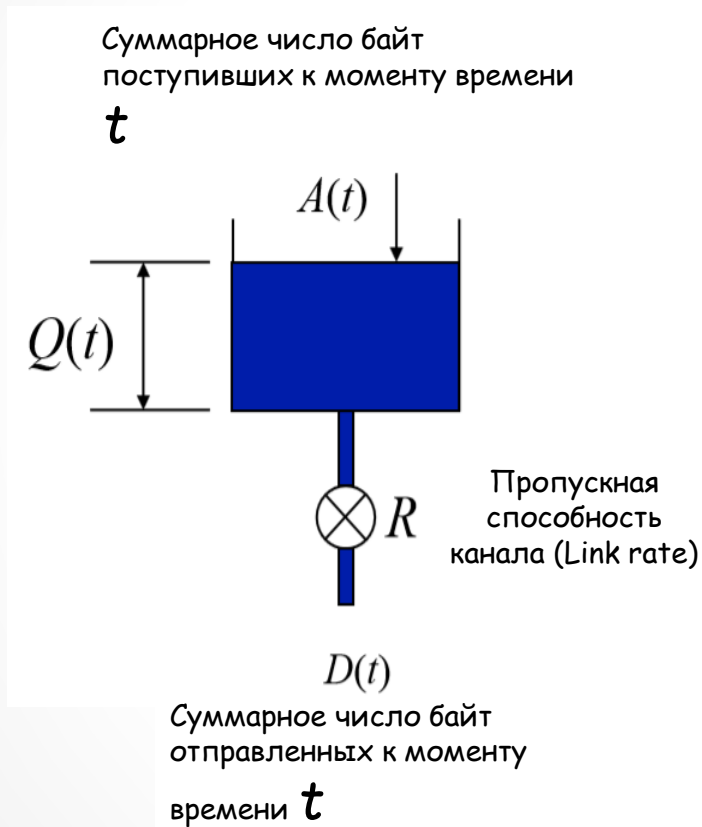


Свойства $A(t), D(t)$:

- $A(t), D(t)$ не возрастающие
- $A(t) \geq D(t)$



Простая модель очереди

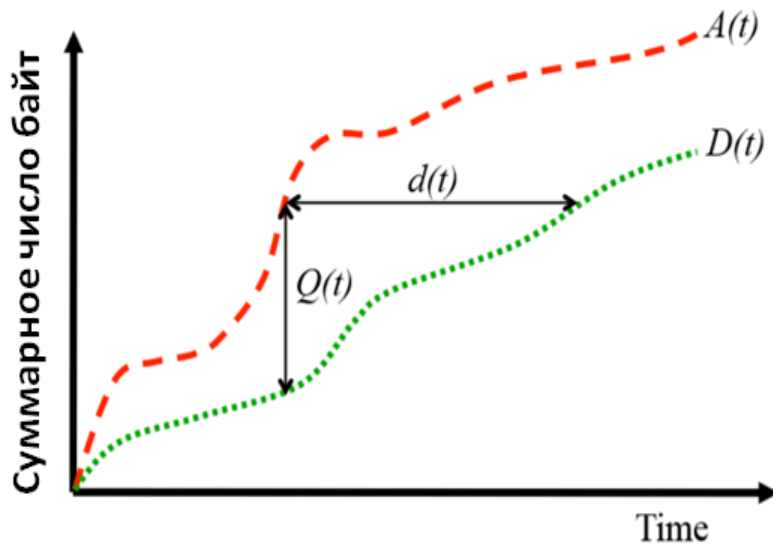


Свойства $A(t), D(t)$:

- $A(t), D(t)$ неубывающие
- $A(t) \geq D(t)$



Простая модель очереди

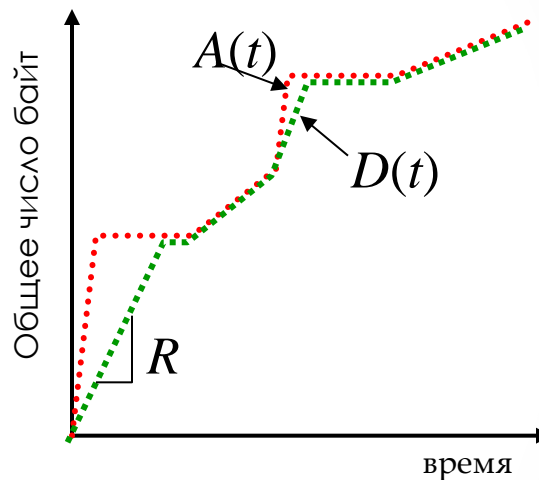
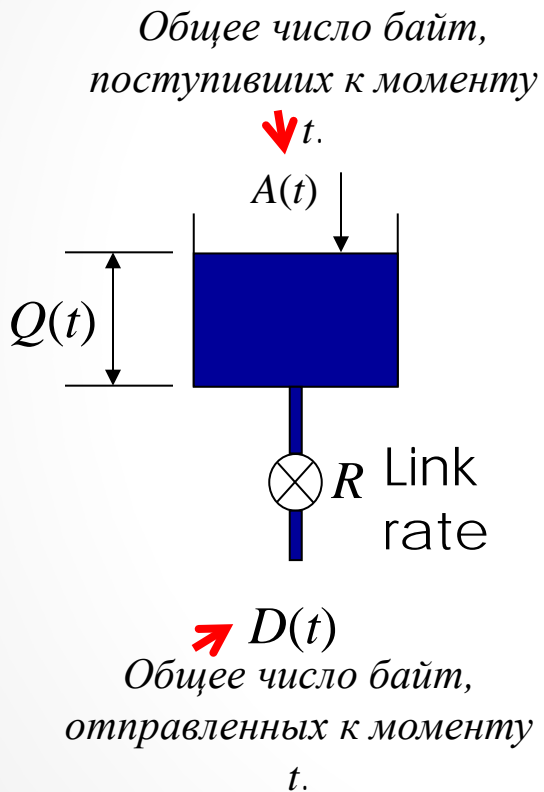


Длина очереди: $Q(t) = A(t) - D(t)$.

Задержка в очереди $d(t)$ – время, которое байт, поступивший в момент t , пробыл в очереди, при условии что дисциплина очереди FIFO.



Простая модель очереди

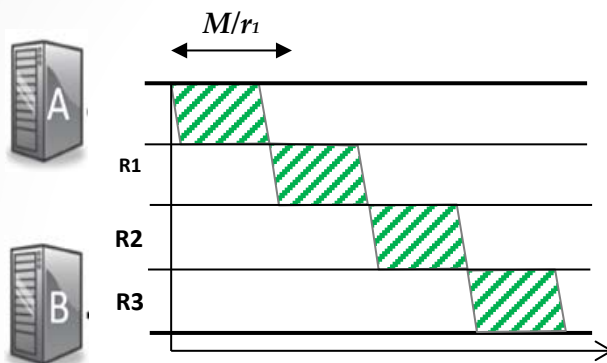


Свойства $A(t)$, $D(t)$:

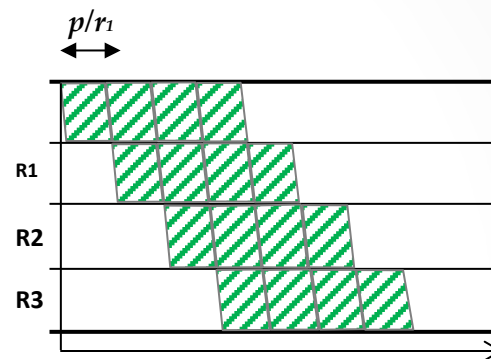
- $A(t)$, $D(t)$ неубывающие
- $A(t) \geq D(t)$



Коммутация пакетов: конвейеризация



$$\text{End-to-end delay, } t = \sum_i \left(\frac{M}{r_i} + \frac{l_i}{c} \right)$$

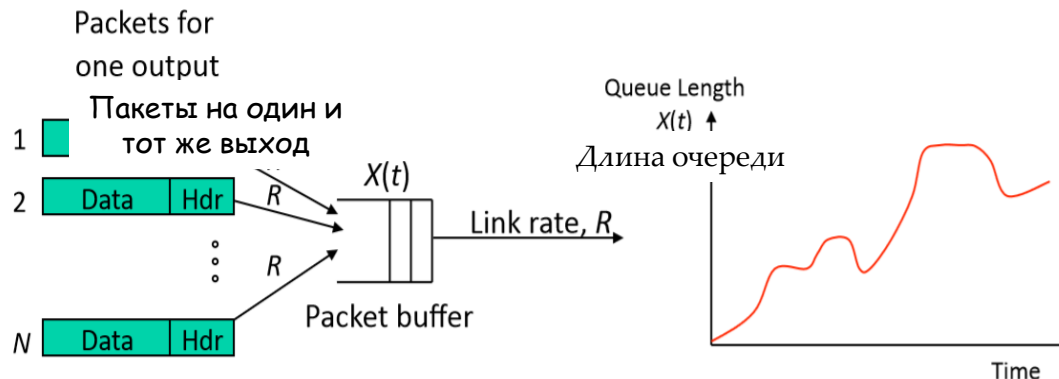


$$\text{End-to-end delay, } t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} \right) + \left(\frac{M}{p} - 1 \right) \frac{p}{r_{\min}}$$

Разбиение сообщения на пакеты позволяет передавать их параллельно по всем линиям, сокращая e2e задержку.



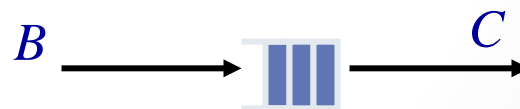
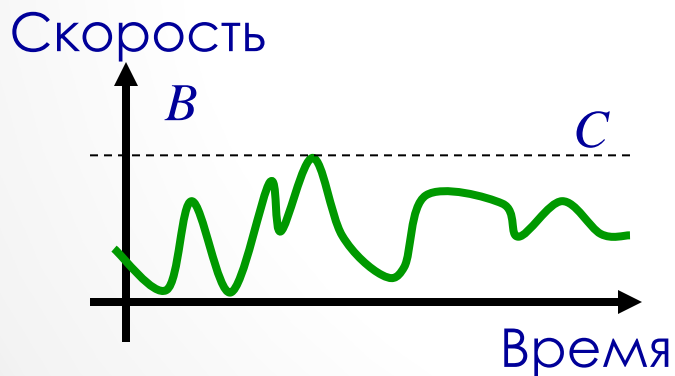
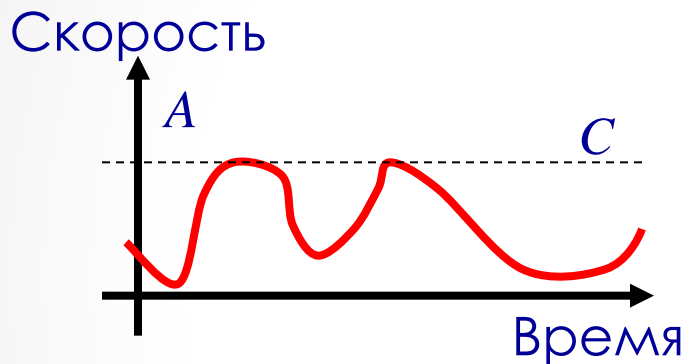
Коммутация пакетов: статистическое мультиплексирование



- Поскольку буфер сглаживает временные всплески (короткие периоды, когда скорость превышает R), то на выходе канал может работать с меньшей скоростью, чем $N \times R$
- Поскольку размер буфера ограничен B , то возникнут потери.

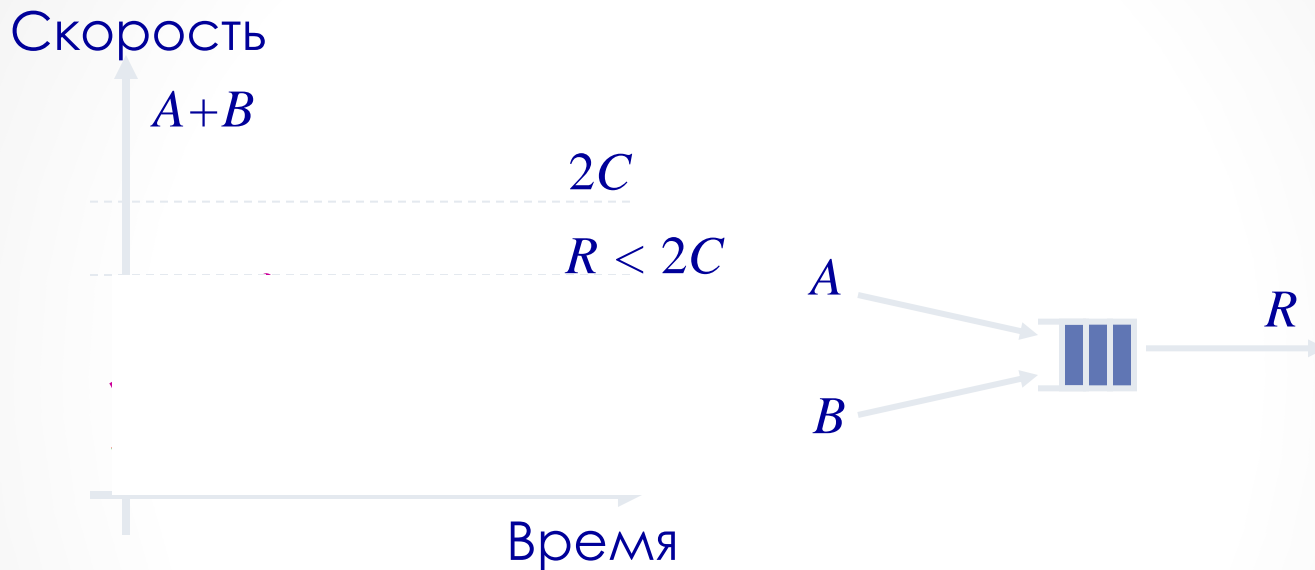


Коммутация пакетов: статистическое мультиплексирование





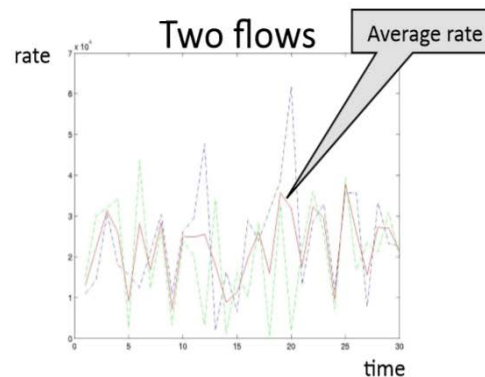
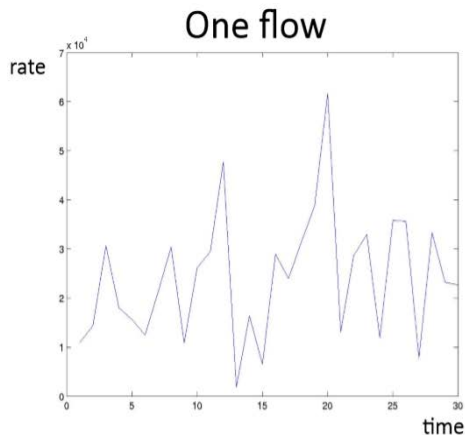
Коммутация пакетов: статистическое мультиплексирование



Выигрыш от статистического мультиплексирования = $2C/R$



Коммутация пакетов: статистическое мультиплексирование



Чем больше потоков, тем глаже, ровнее функция средней скорости



Коммутация пакетов: простая модель с очередями

- Для понимания динамики пакетов в сети вполне подходит простая детерминированная модель.
- Разбиение сообщения на пакеты позволяет нам параллельно передавать несколько пакетов, сокращая ее задержку.
- Статистическое мультиплексирование позволяет передавать N потоков через один и тот же канал не увеличивая его в N раз!



Свойства очередей

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

Кафедра АСВК ф-т ВМК МГУ



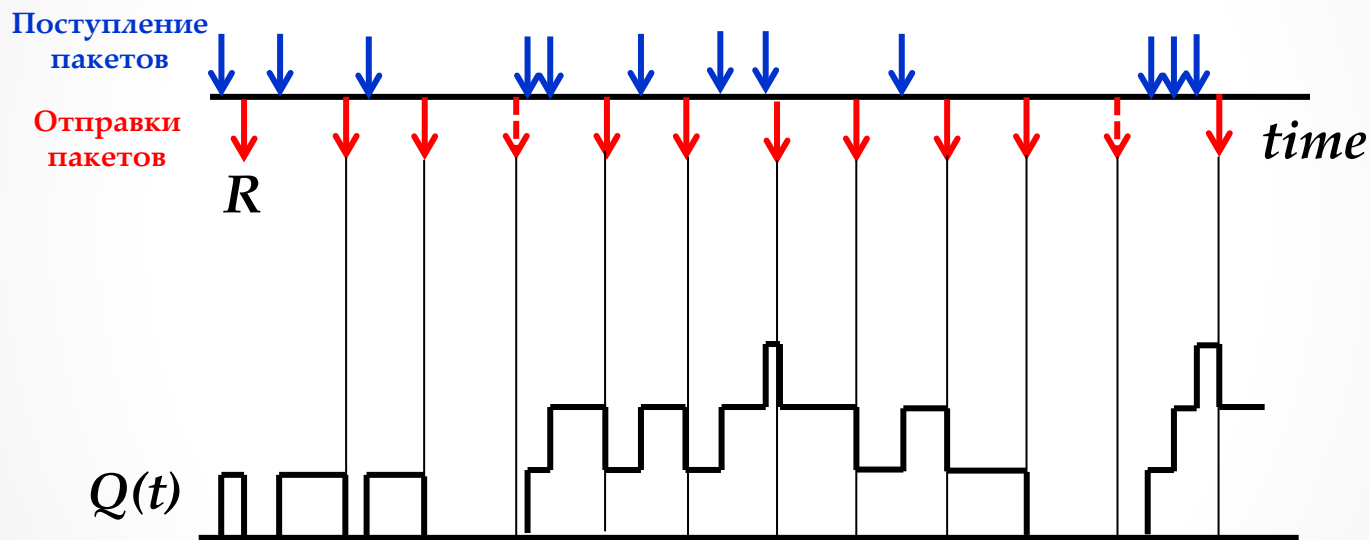


Очереди со случайным процессом поступления

- *Обычно процесс поступления пакетов сложен и трудно предсказуем Поэтому его часто моделируют случайным процессом.*
- *Изучает очереди с такими процессами Теория Очередей (Queuing Theory - теория массового обслуживания).*
- *Здесь мы рассмотрим некоторые свойства таких очередей.*



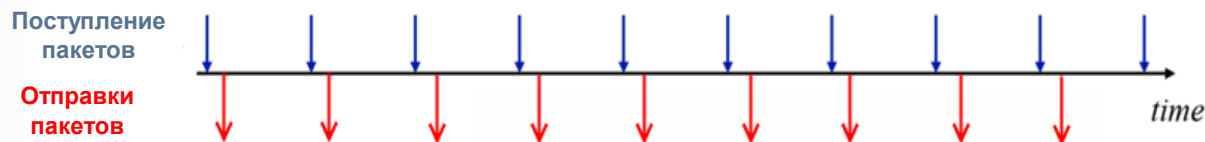
Эволюция очереди со временем





Свойство 1 : равномерность потока сохраняет задержку постоянной

Периодические одиночные поступления





Свойство 2 очередей: неравномерность увеличивает задержку

Периодические множественные поступления

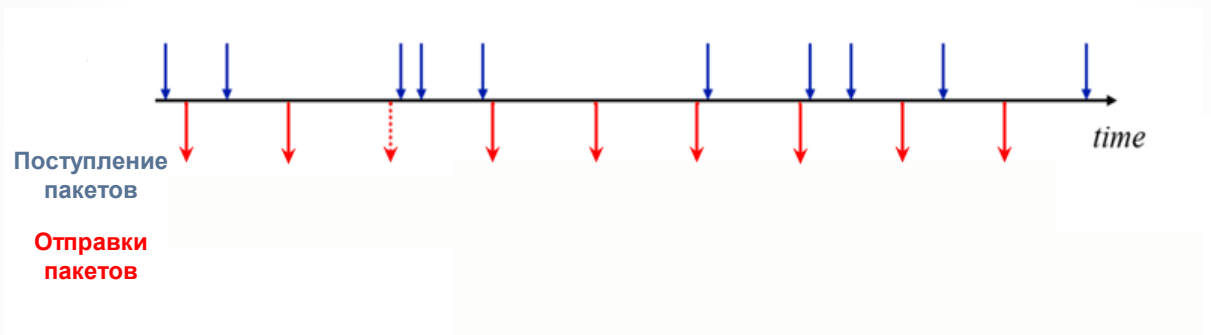


Регулярная неравномерность сохраняет задержку в среднем, но задержка в очереди для части пакетов возрастает, носит периодический, предсказуемый характер



Свойство 3 очередей: случайность увеличивает задержку

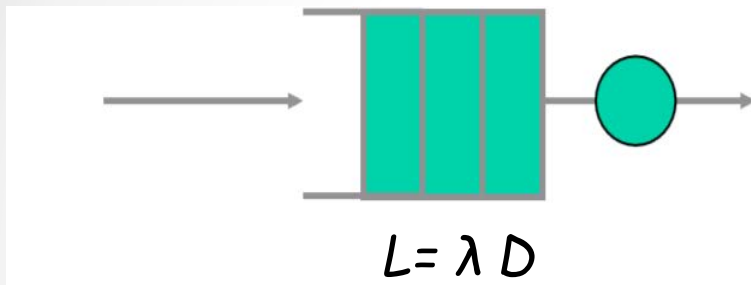
Случайное поступления



При случайном поступлении пакетов
случайно меняется и средняя длина очереди



Свойство 4: Формула Литтла



$N(T)$ - число поступивших заявок

$A(T)$ - время нахождения всех заявок в системе

$\lambda(T) = N(T)/T$ - ср. число заявок

$D(T) = A(T)/N(T)$ - ср. время заявки в системе

$L(T) = A(T)/T$ - ср. число заявок на

обслуживании в системе

L - среднее число заявок в системе (в очереди + в обслуживании)

λ - среднее число заявок, поступивших в секунду

D - среднее время пребывания заявки в системе (в очереди + в обслуживании, т.е. задержка)

Это свойство верно если ни одна заявка не теряется/сбрасывается



Пуассоновский процесс

- Процесс поступления является Пуассоновским если:

- Вероятность поступления k заявок за t секунд

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

- Интервалы между последовательными поступлениями независимы (нет регулярности)

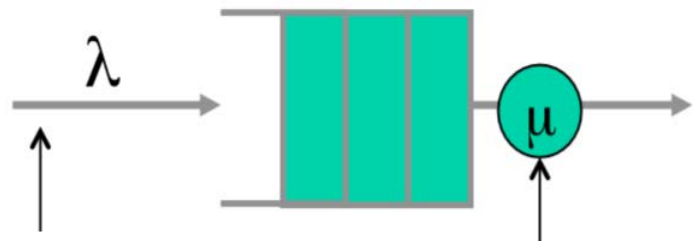
- Тогда число заявок поступивших за время t

$$E = \lambda t$$

- Пуассоновские процессы хорошо моделируют многие случайные процессы (телефонные звонки, распад частиц, шумы в электрических цепях и т.д.)
- Удобный математический аппарат



Простая модель очереди



Пуассон

Экспоненциальный

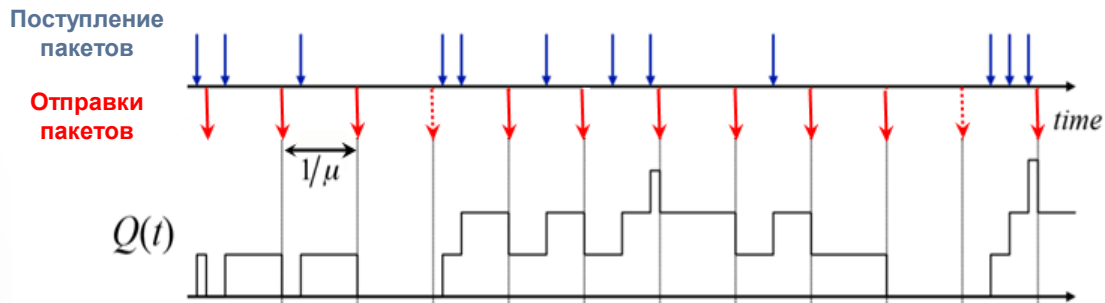
$$L = \frac{\lambda}{|\mu - \lambda|}$$

λ – ср. число заявок в сек.
 μ – ср. число обслуженных в сек.
предполагаем $\mu > \lambda$
 $D = 1/(\mu - \lambda)$

при $\lambda = \mu$ скорость роста очереди будет пропорциональна корню из t

$\lambda > \mu$ скорость роста очереди будет линейно пропорциональна t

Модель очереди в маршрутизаторе





Свойства очередей

- **Свойства очередей:**
 - Равномерность сохраняет задержку постоянной
 - Неравномерная регулярность увеличивает задержку, но сохраняет постоянной в среднем
 - Случайность делает переменной даже задержку в среднем
 - Формула Литтла: $L = \lambda d$
- *Поступление пакетов не Пуассоновский процесс, но такие процессы как поступление web запросов, новых потоков, могут быть описаны как Пуассоновские процессы*
- *M/M/1- простая модель очереди описывается через распределения входящего и исходящего потоков*



Коммутация пакетов: как работает пакетный коммутатор

Введение в компьютерные сети
чл.-корр. РАН Смелянский Р.Л.
Кафедра АСВК ф-т ВМК МГУ

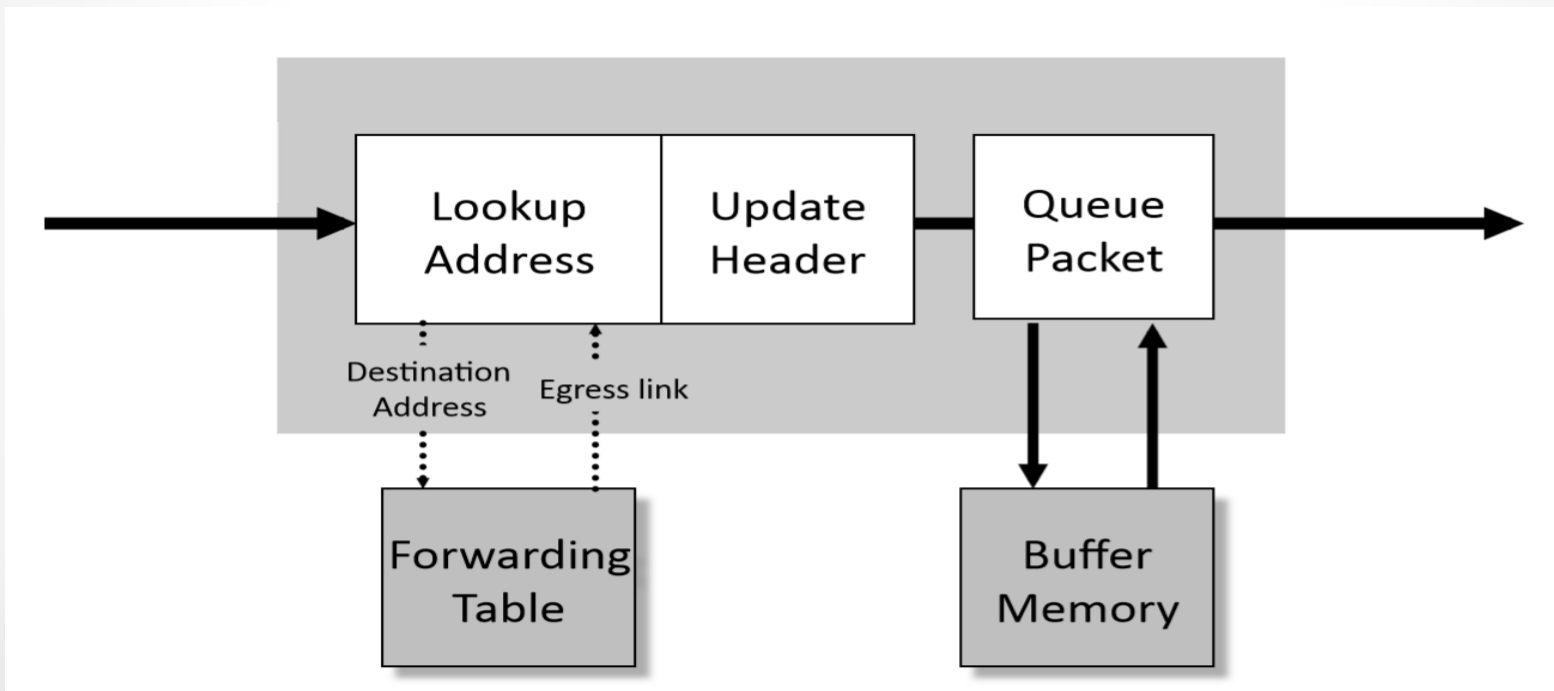


Содержание

- *Как устроен пакетный коммутатор*
- *Что делает пакетный коммутатор:*
 - *Ethernet switch*
 - *Internet router*
- *Как устроен поиск адресов в:*
 - *Ethernet switch*
 - *Internet router*
- *Виды буферизации в коммутаторах*

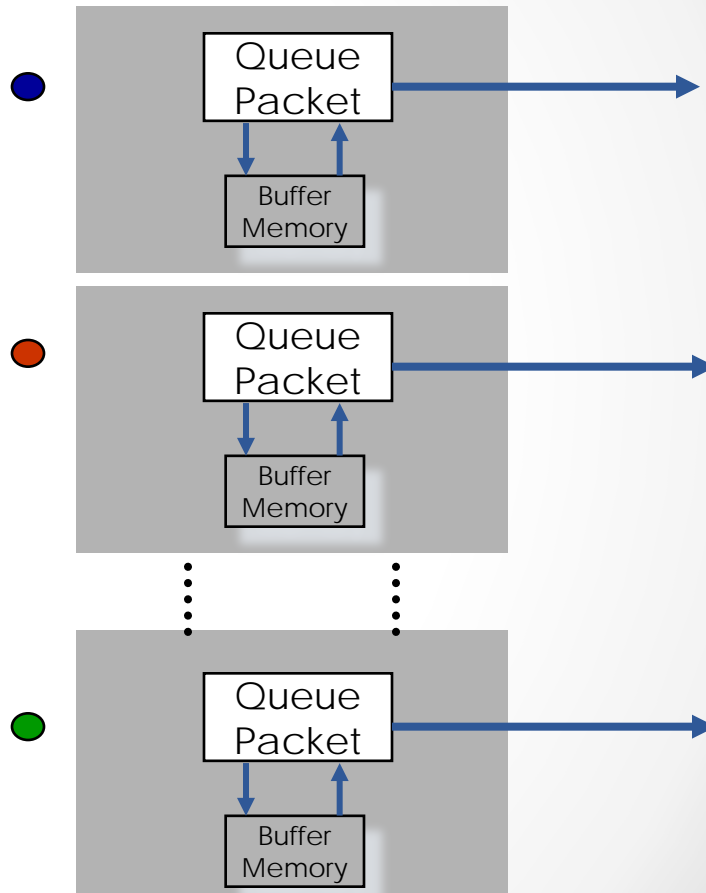
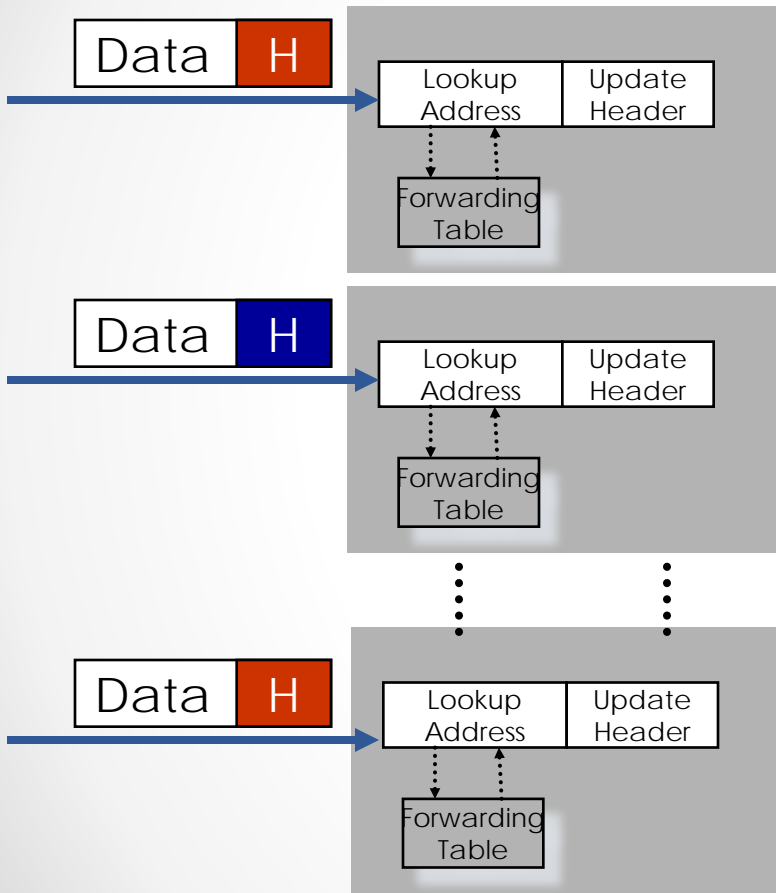


Как устроен пакетный коммутатор





Пакетный коммутатор





Канальный (Ethernet) коммутатор

1. Проверяет заголовок каждого прибывающего кадра
2. Если адрес DA есть в таблице коммутации, то кадр передают на надлежащий выходной порт
3. Если адрес DA нет в таблице, кадр рассылается по всем портам, кроме того на который пришел.
4. Когда придет ответ на разосланный пакет, то по его адресу отправителя и номеру порта, с которого он пришел, мы узнаем, куда надо направлять пакеты с такими адресами получателей.



Интернет маршрутизатор

1. Если Ethernet DA поступившего кадра есть Ethernet адрес маршрутизатора, то принять кадр, иначе сбросить его.
2. Просмотреть поля IP version и длина дейтаграммы
3. Сократить поле TTL, Если поле TTL = 0, сбросить пакет
4. Пересчитать контрольную сумму IP заголовка
5. Если IP DA есть в таблице маршрутизации, переслать на надлежащий выходной порт для следующего скачка (hop)
6. Если такого адреса нет, то ICMP сообщение
7. Найти Ethernet DA для следующего маршрутизатора
8. Построить новый Ethernet кадр и отправить его



Базовые операции коммутатора

1. Поиск адреса: как адрес ищется в таблице маршрутизации?
2. Коммутация: передача пакета на надлежащий выходной порт.



Поиск адреса: Ethernet

Таблица Ethernet адресов

Ethernet DA	Действие
0xA8B72340E678	Передать на порт 7
0xB3D22571053B	Передать на порт 3
.....

Адреса хранятся в хэш-таблице
Ищем в хэш-таблице точное совпадение



Поиск адреса: IP

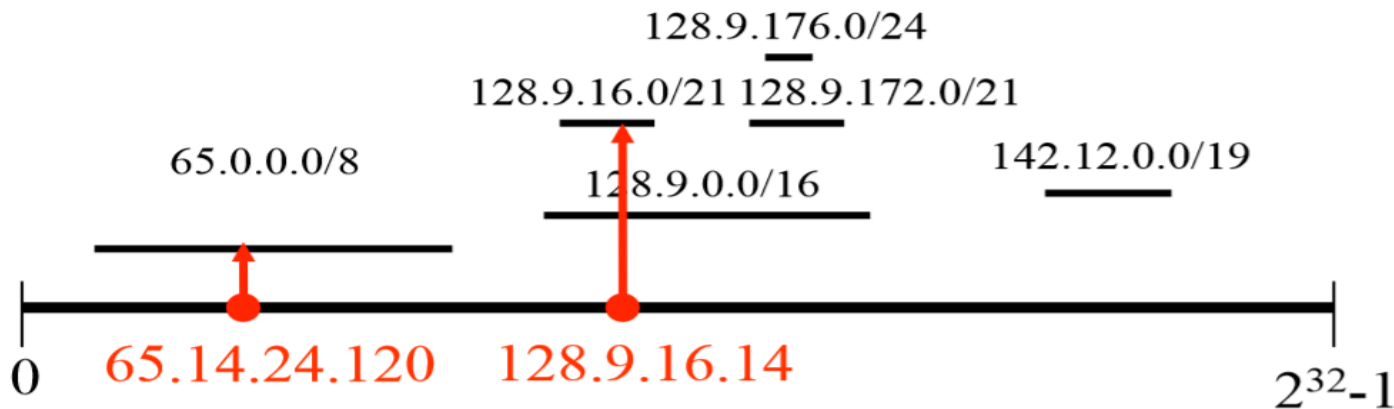
Таблица IP адресов в маршрутизаторе

IP DA	Действие
127.43.57.99	Передать на 56.99.32.16
123.66.44.x	Передать на 22.45.21.126
76.9.x.x	Передать на 56.99.32.16
.....

**Ищут совпадение по самому длинному префиксу,
а не точное совпадение.**



Поиск совпадения по самому длинному префиксу

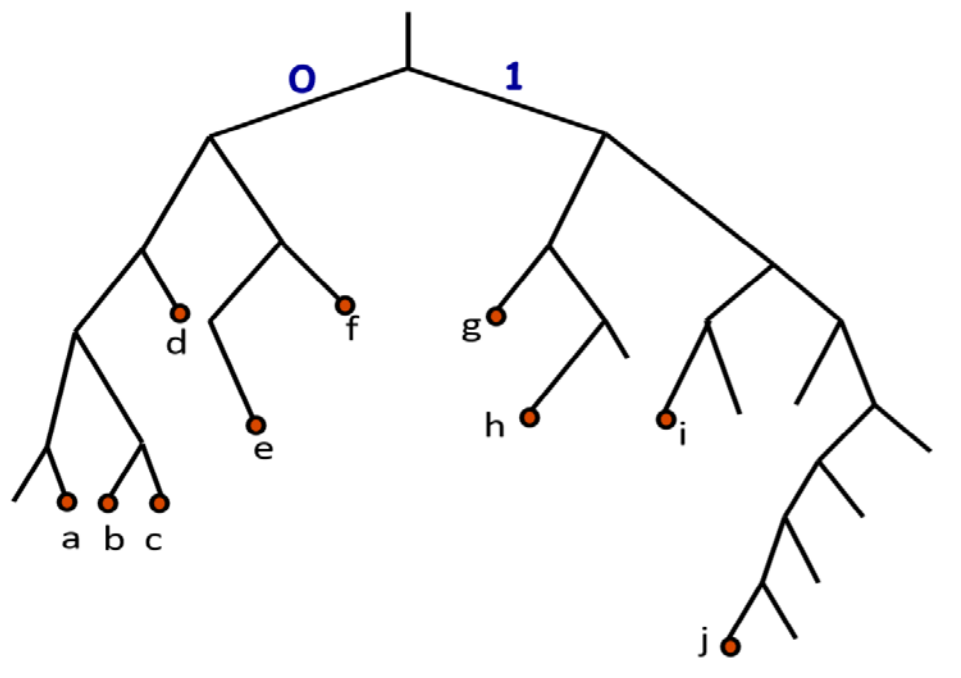


128.9.16.0	1000 0000	0000 1001	0001 0 000	
128.9.172.0	1000 0000	0000 1001	1011 1 100	
128.9.16.14	1000 0000	0000 1001	0001 0 000	



Поиск совпадения самого длинного префикса

Entry	Prefix
a	00001
b	00010
c	00011
d	001
e	0101
f	011
g	100
h	1010
i	1100
j	11110000





Поиск совпадения самого длинного префикса (ТСАМ - Ternary Content Addressable Memory)

Entry	Prefix
a	00001
b	00010
c	00011
d	001
e	0101
f	011
g	100
h	1010
i	1100
j	11110000

Двоичное
представление
+ Маска



Entry	Prefix
a	00001XXX 11111000
b	00010XXX 11111000
c	00011XXX 11111000
d	001XXXXX 11100000
e	0101XXXX 11110000
...	...
j	11110000 11111111



Обобщенный коммутатор

< Совпадение, Действие >

Совпадение	Действие
IP DA = X	Передать на порт 7
EthDA=Y & IP DA = Z	Сброс пакета

Обобщение поиска и коммутации в коммутаторах, маршрутизаторах и т.п.



Как работает коммутатор: заключение

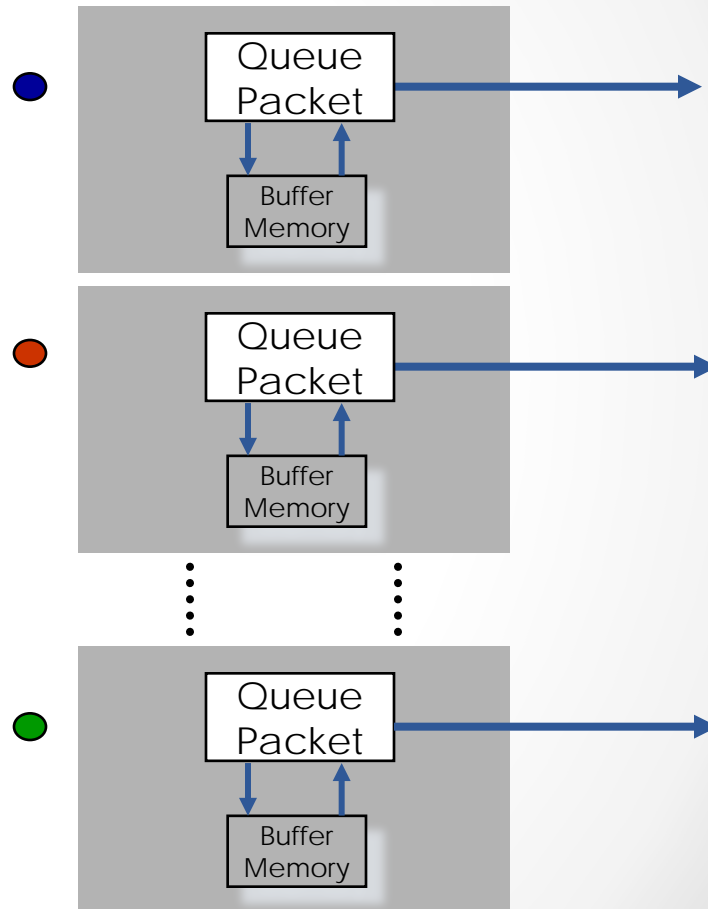
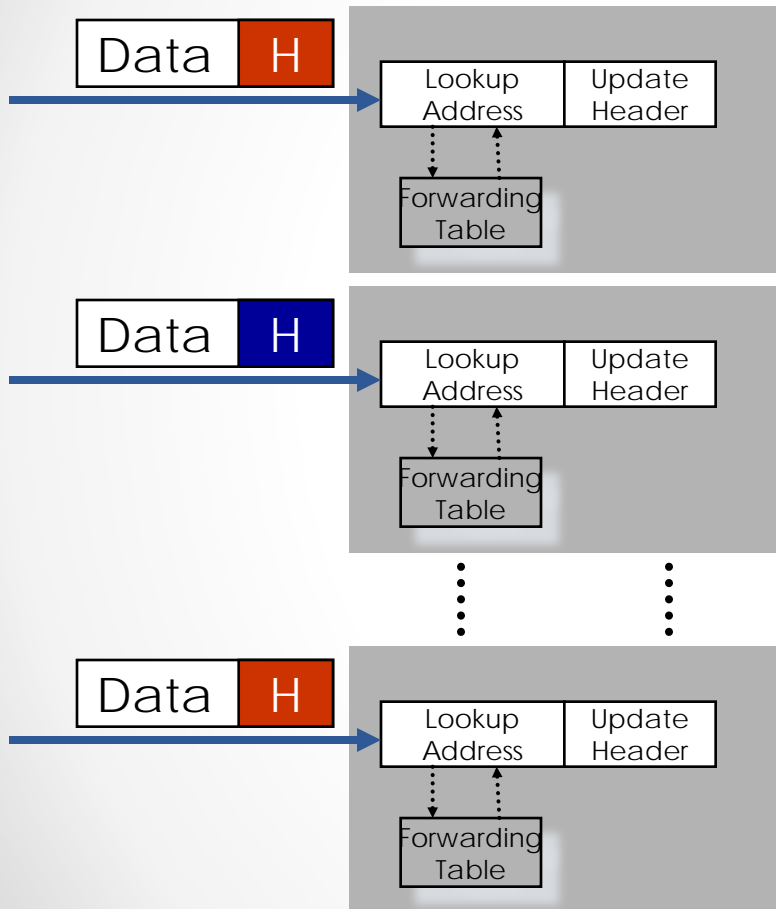
- **Пакетный коммутатор выполняет две базовые операции:**
 - Поиск адресов в таблице коммутации
 - Передача на надлежащий выходной порт
- По сути, Ethernet коммутаторы и маршрутизаторы выполняют одинаковые действия
- Поиск адреса в коммутаторе и маршрутизаторе происходит по-разному.



Пакетный коммутатор: виды буферизации

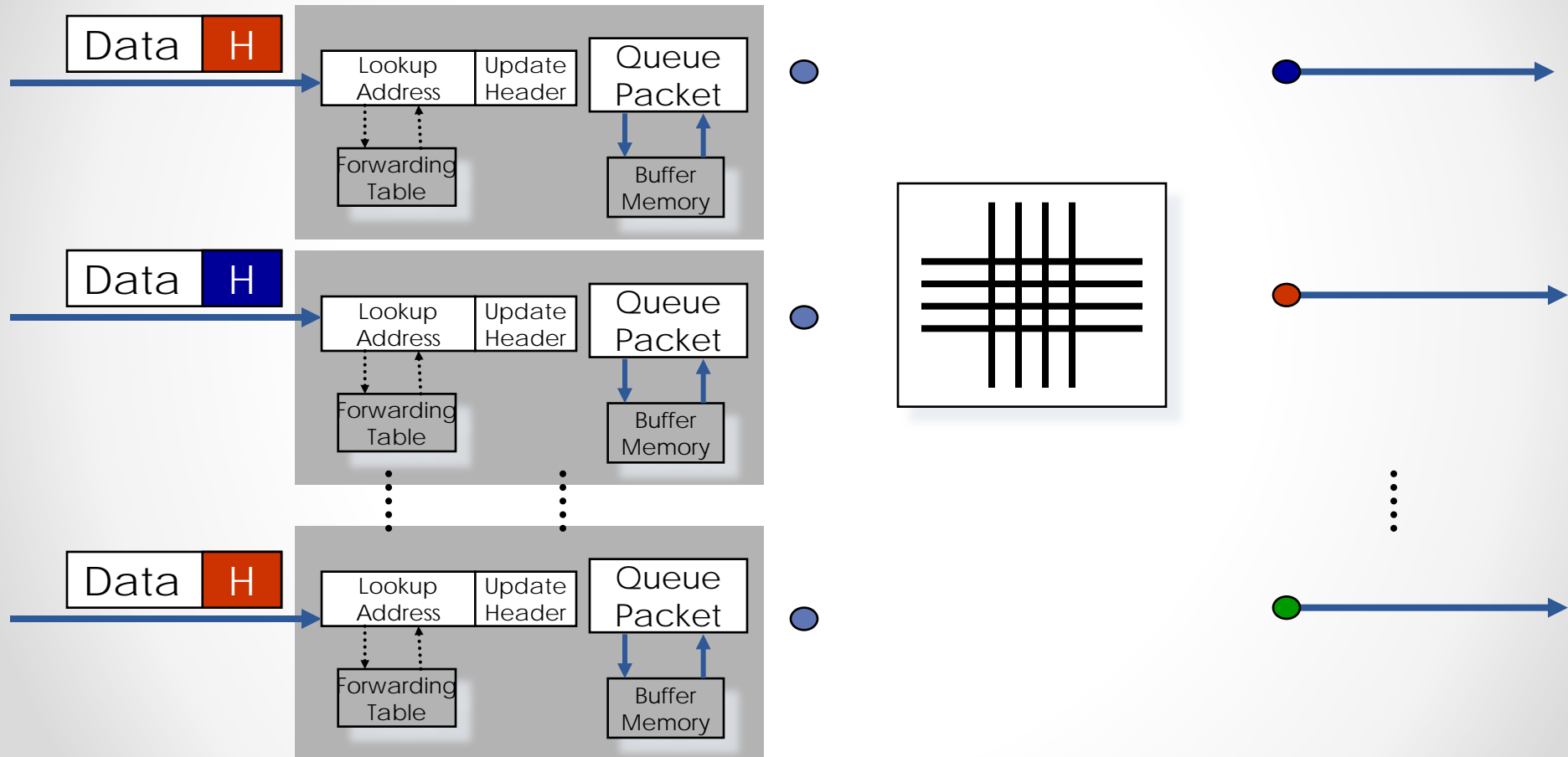


Буферизация на выходе



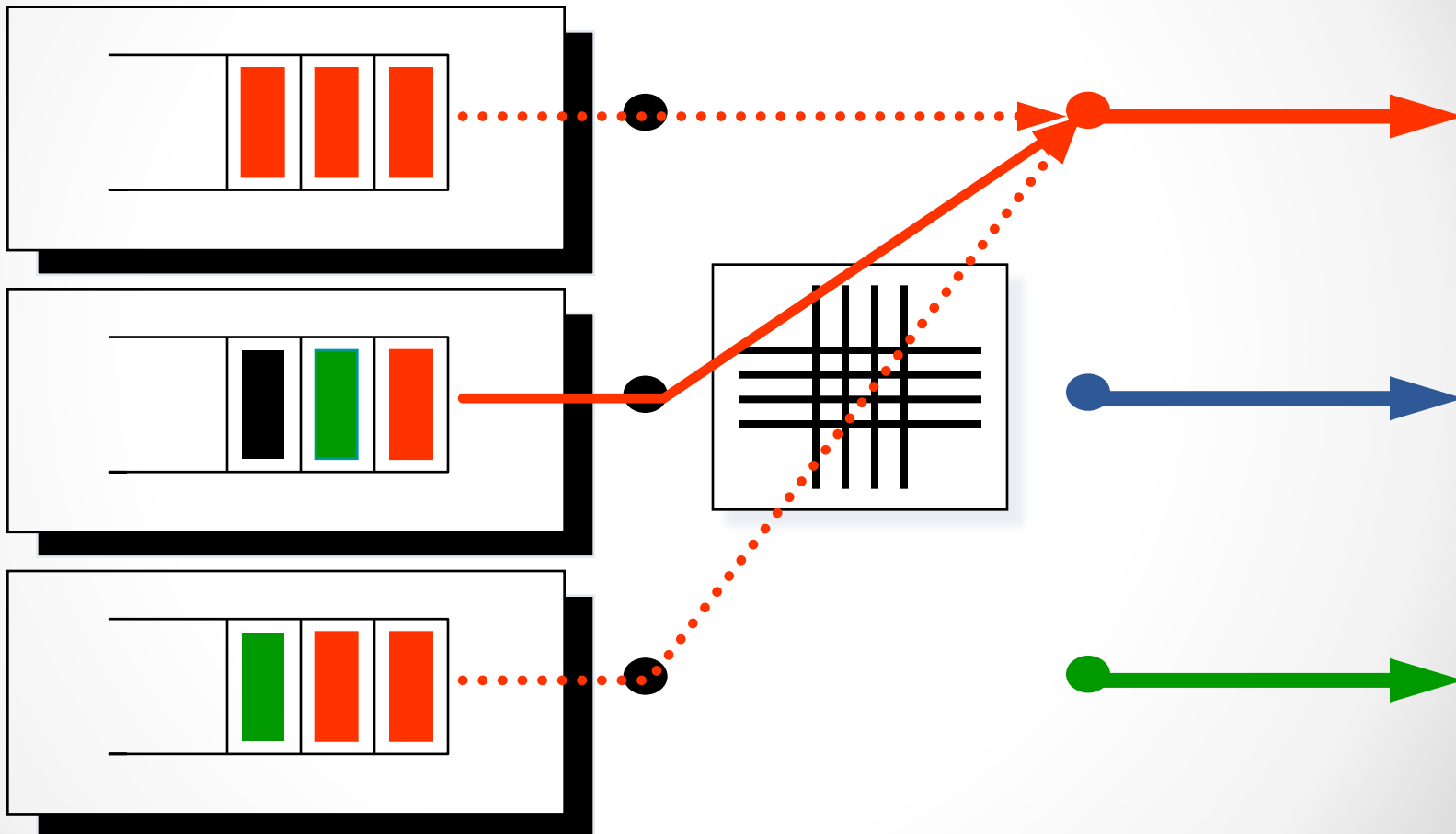


Буферизация на входе



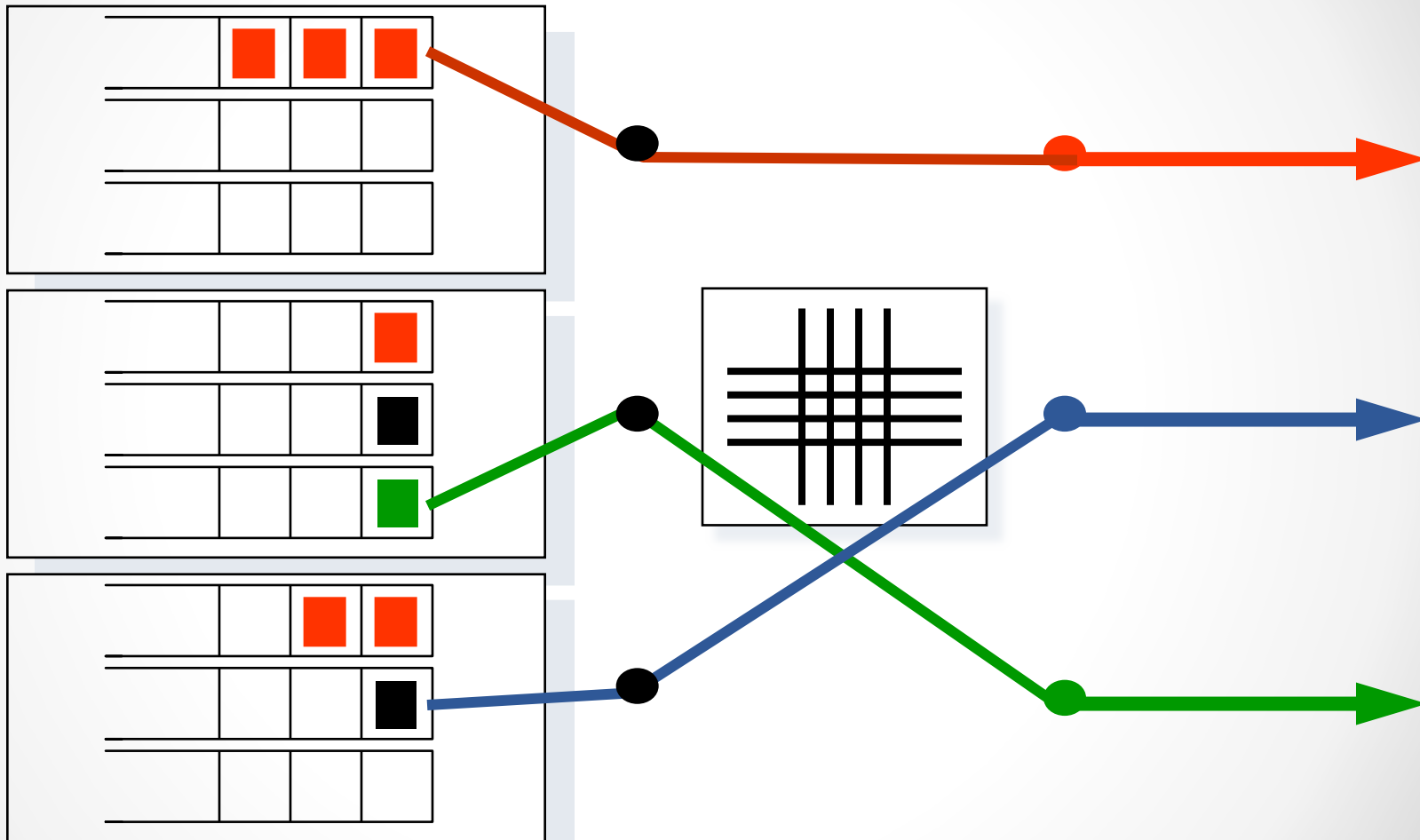


Блокировка на входе



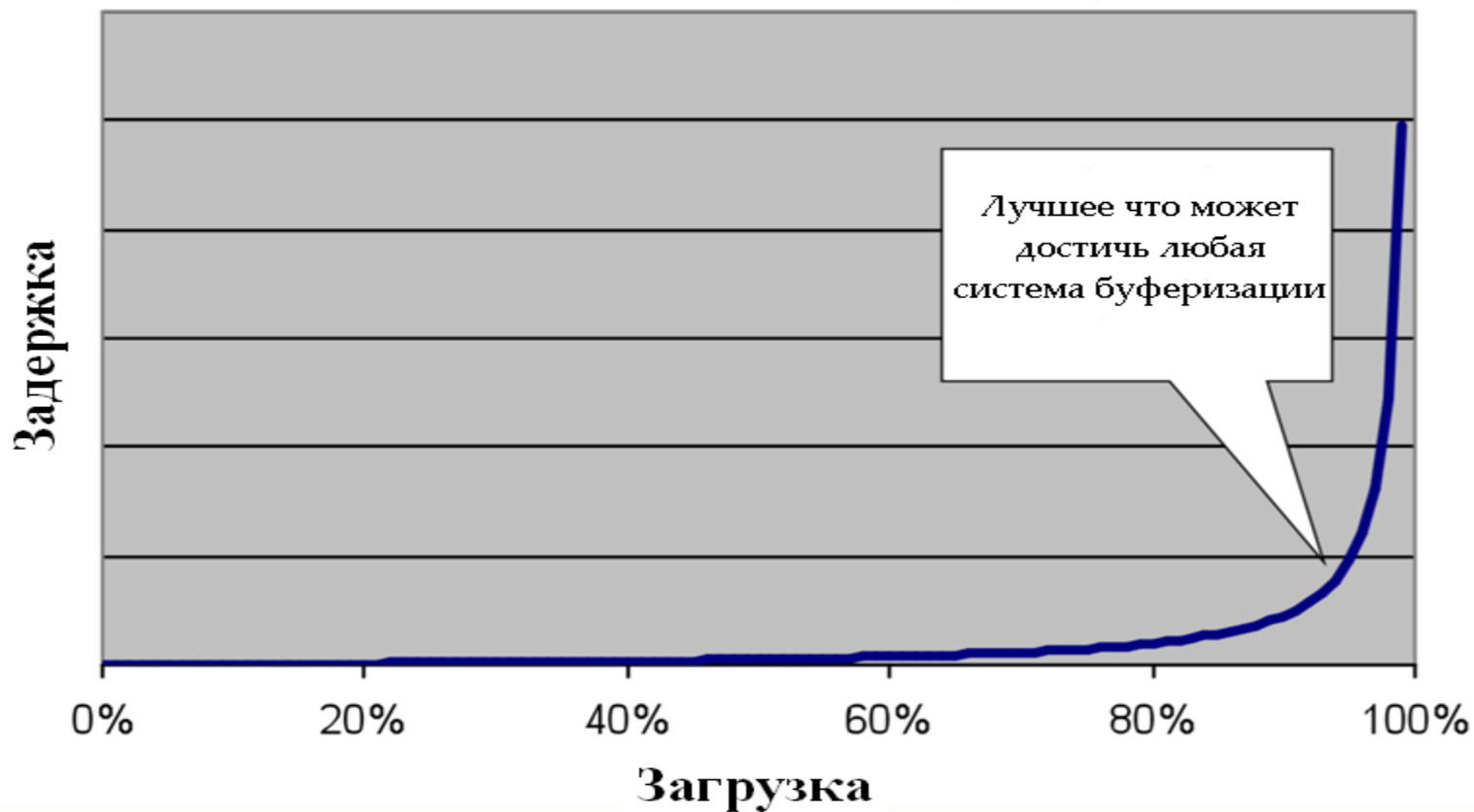


Виртуальные очереди на выход





Пакетный коммутатор с буферизацией на выходе (OQ)





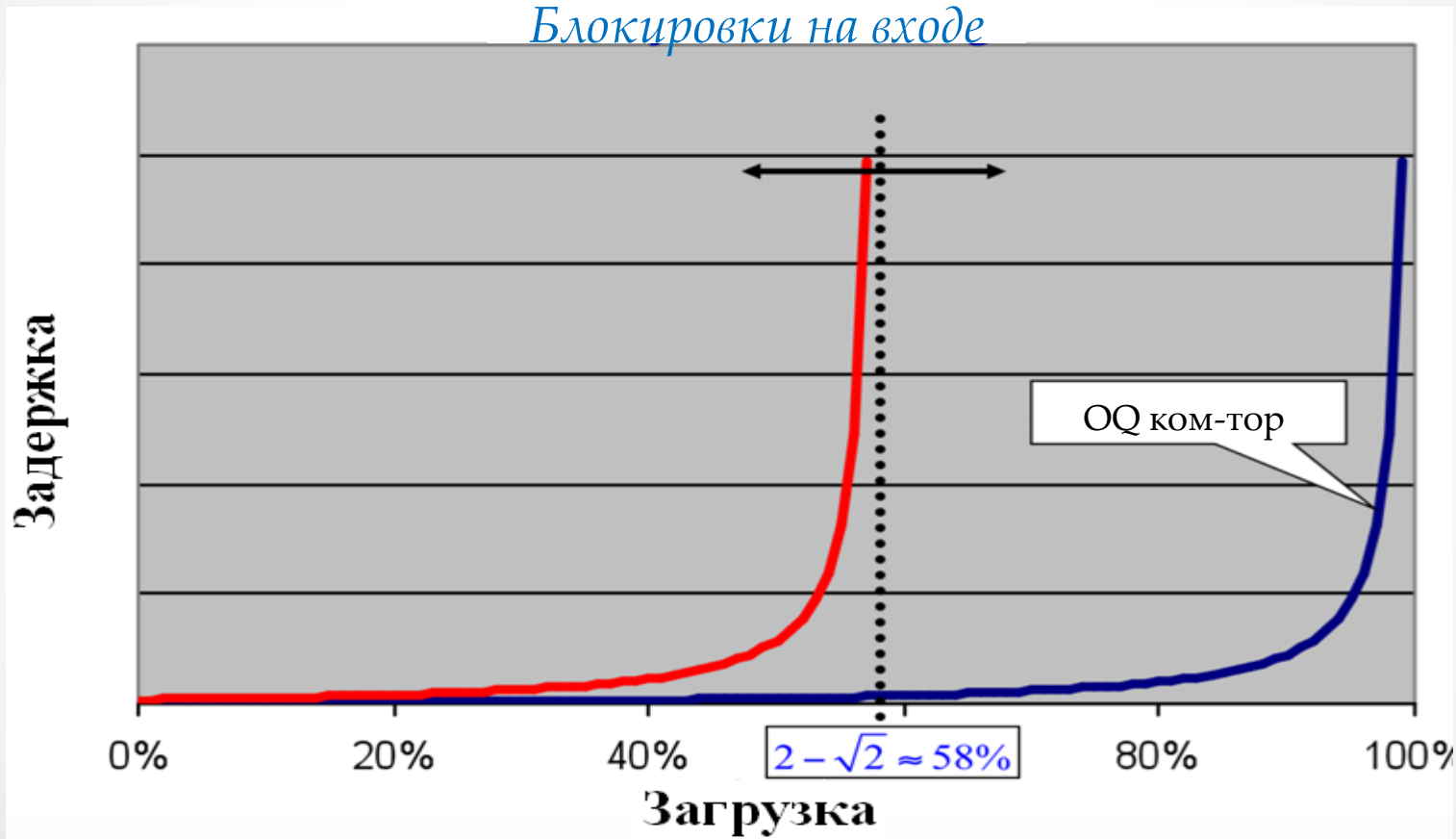
Свойства ОО коммутатора

1. Работают с минимальными потерями
2. Максимальная пропускная способность
3. Ожидаемая задержка минимальна
4. Очень большие требования по памяти для буферизации



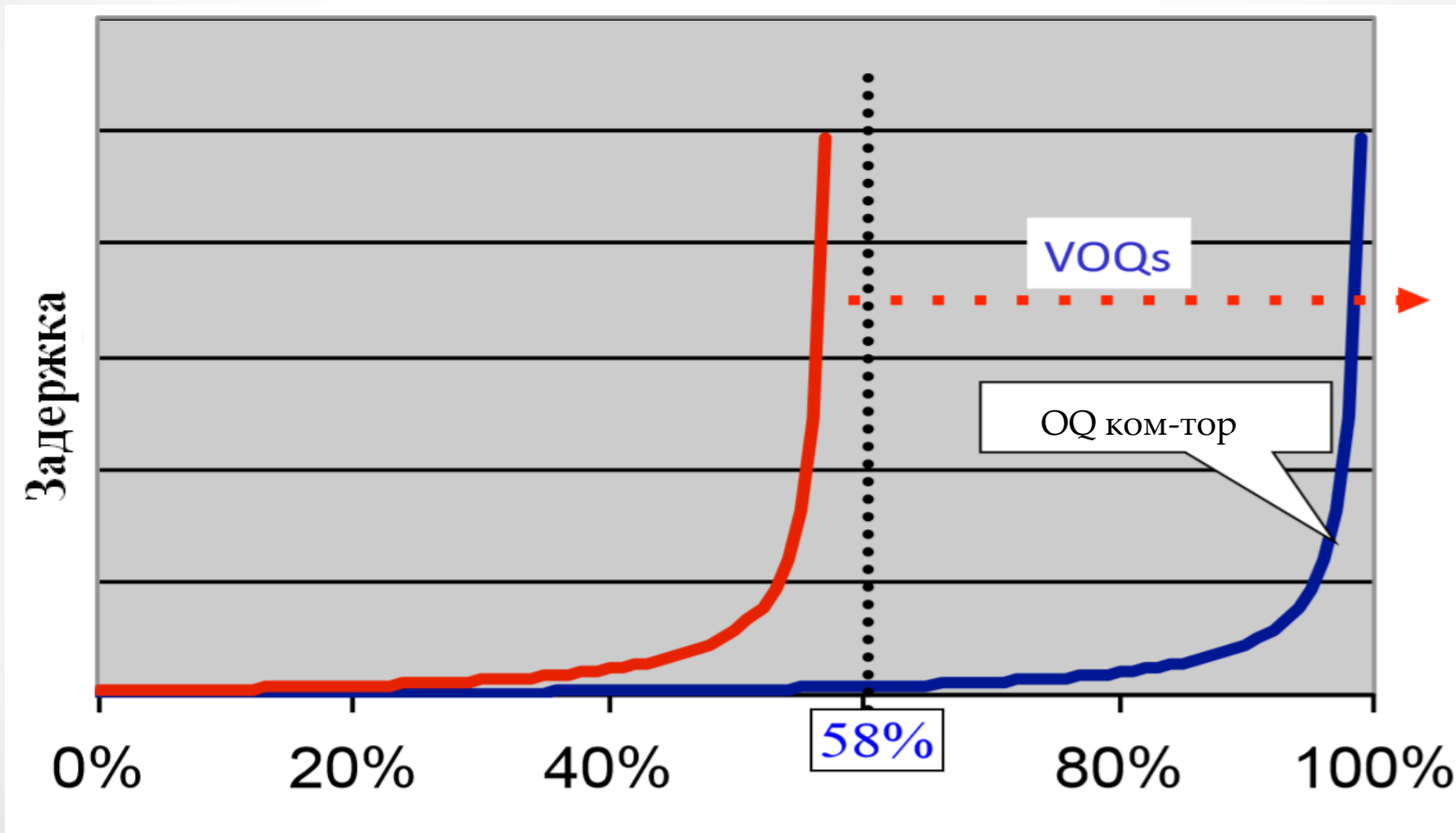
Пакетный коммутатор с буферизацией на входе

Блокировки на входе





Пакетный коммутатор с буферизацией на входе (с виртуальными очередями на выходе)





2013 Oktva



Пакетный коммутатор: заключение

- Пакетный коммутатор выполняет две базовые операции:
 - Поиск соответствия в таблице коммутации
 - Передачу на надлежащий выходной порт
- Буферизация на входе - низкая пропускная способность из-за блокировок на входах, не требовательная к скорости работы и емкости очереди
- Буферизация на выходе - максимальная пропускная способность, минимальная задержка пакета, но требует высокой скорости работы и емкости буфера.
- Высокоскоростные коммутаторы используют буферизацию на входе с очередями виртуальных выходов для увеличения пропускной способности.



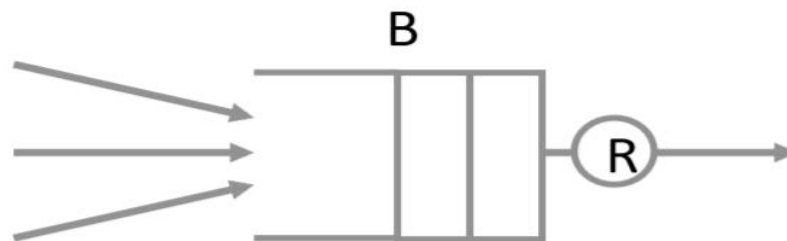
Коммутация пакетов: оценка и управление задержкой

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.
Кафедра АСВК ф-т ВМК МГУ



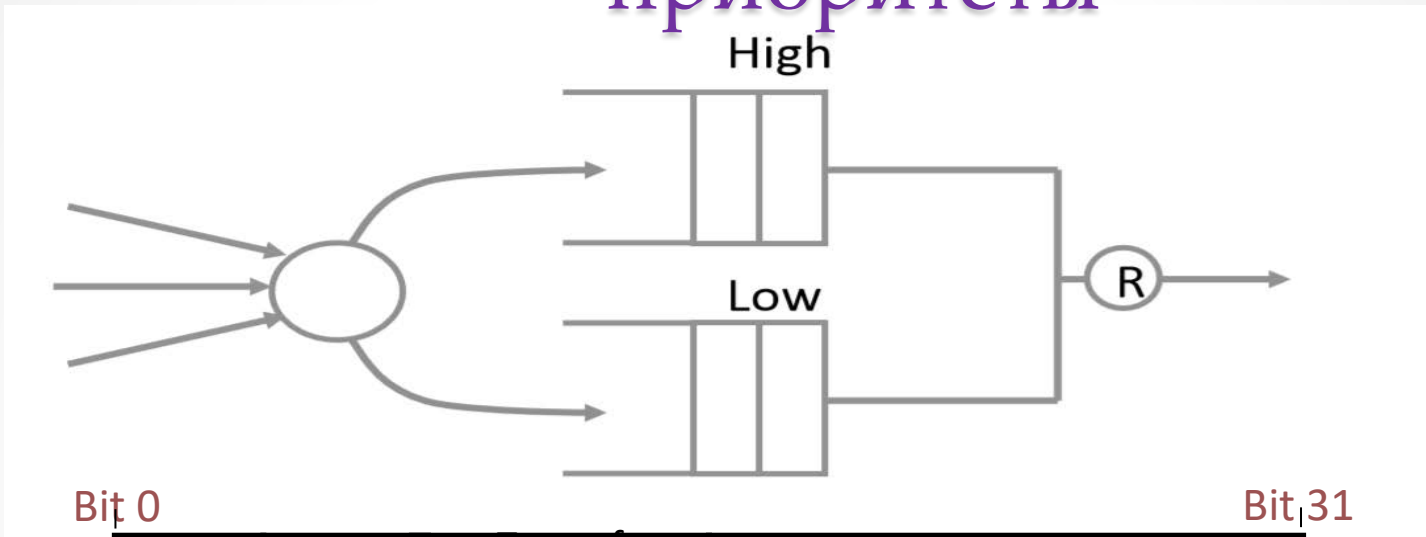
Пакетный коммутатор: FIFO



- Как регулировать качество сервиса (скорость)?



Пакетный коммутатор: строгие приоритеты



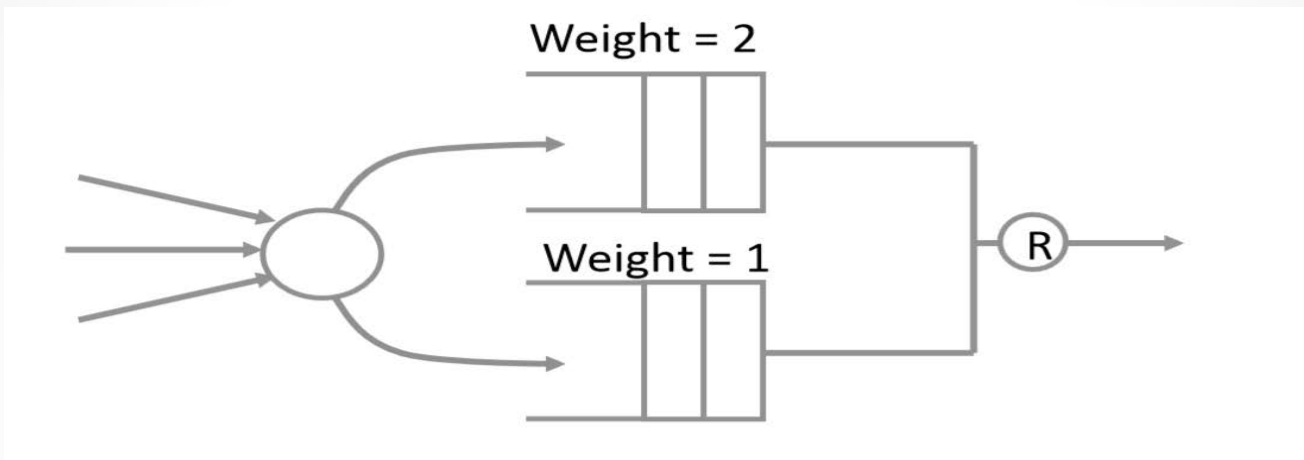
Bit 0

Bit 31

Version	Header Length	Type of Service	Total Packet Length	
Packet ID		Flags	Fragment Offset	
Time to Live "TTL"	Protocol ID		Checksum	
Source IP Address				
Destination IP Address				
(OPTIONS)				(PAD)



Пакетный коммутатор: приоритеты с весами

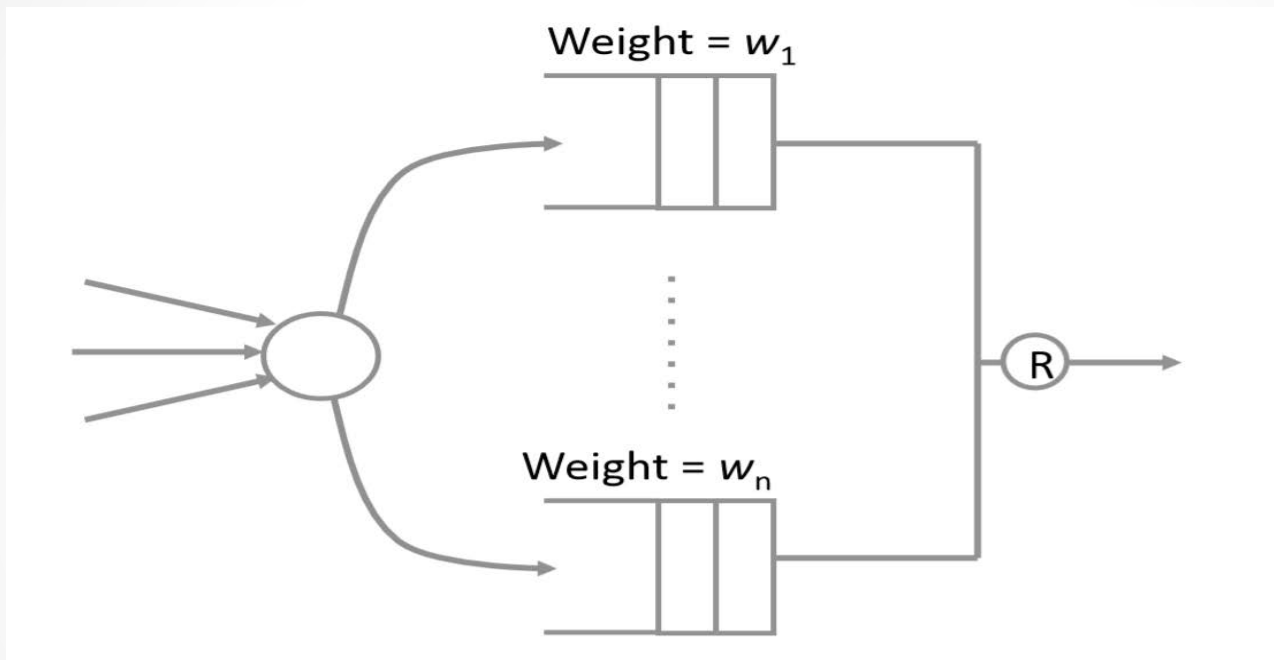


$$R_1 = R * 2/(2+1)$$

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



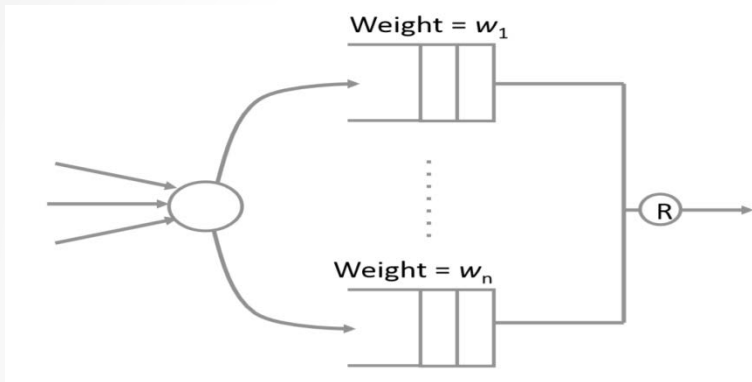
Пакетный коммутатор: приоритеты с весами



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



Пакетный коммутатор: оценка задержки



Weighted Fair Queuing

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

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

$SR = \max \{FR-1, now\}$; $FR = SR + L/w_i$ где L 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 в предположении
 - Дисциплина обслуживания очереди - FIFO
 - Нет потери пакетов



Коммутация пакетов: управление задержкой

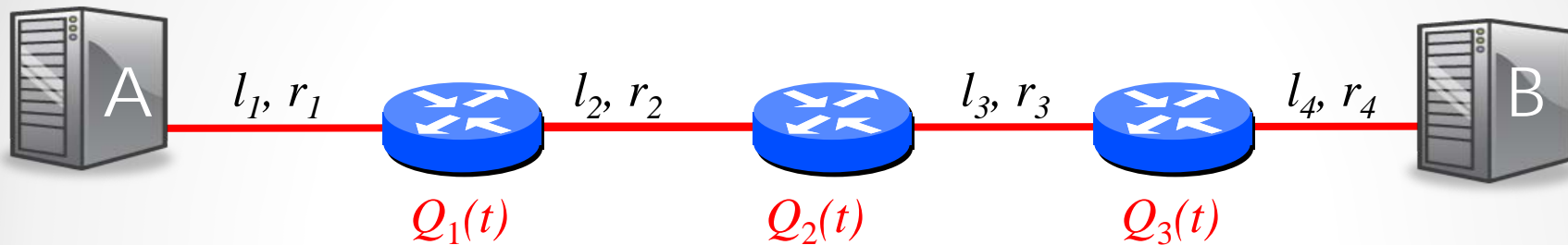
Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.Л.

Кафедра АСВК ф-т ВМК МГУ



Факторы влияющие на задержку

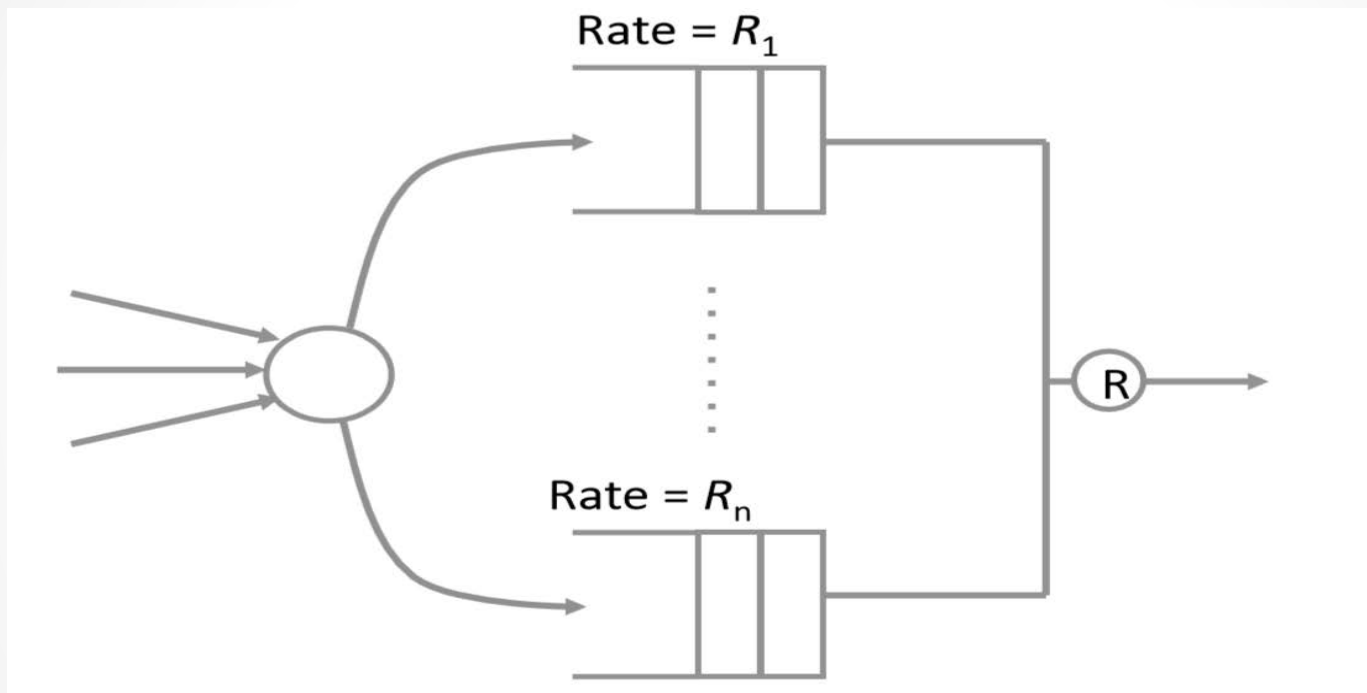


$$\text{е2е задержка, } \tau = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$$

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





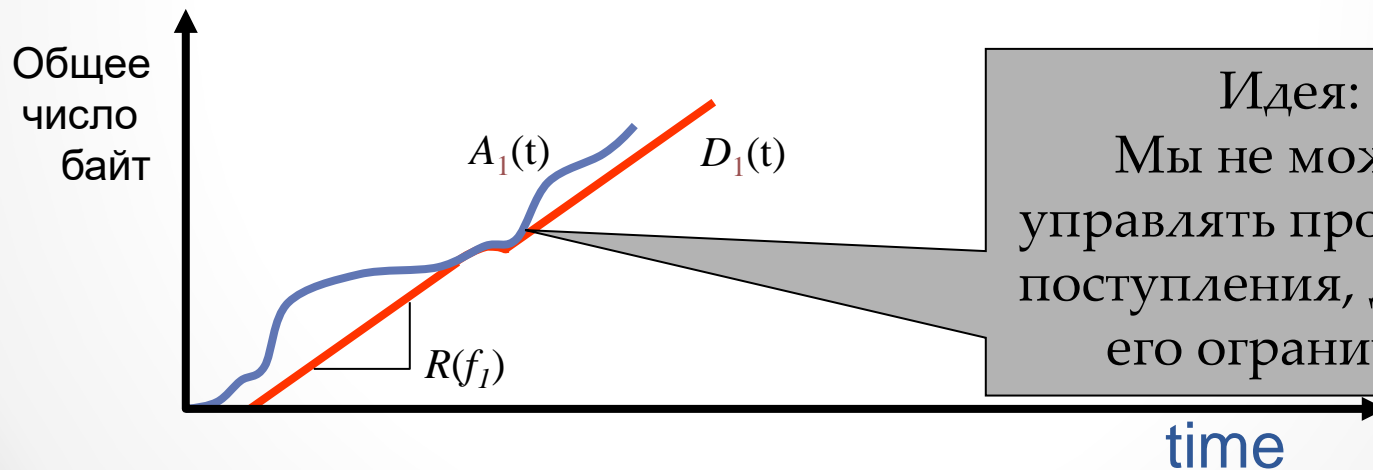
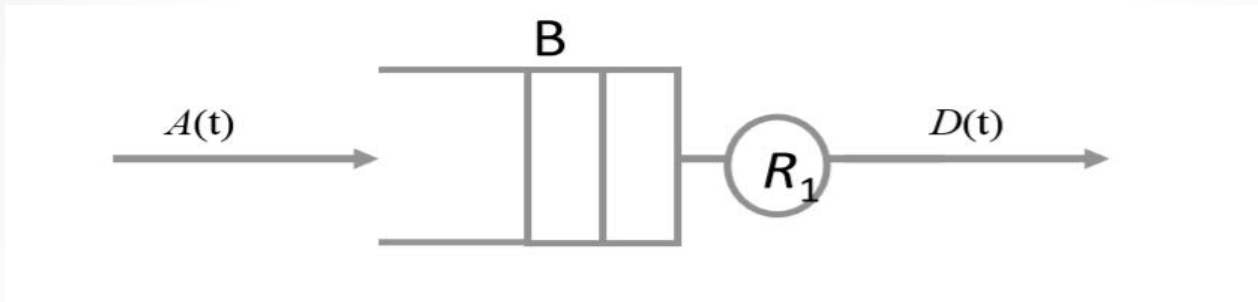
Пакетный коммутатор: управление задержкой пакета

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

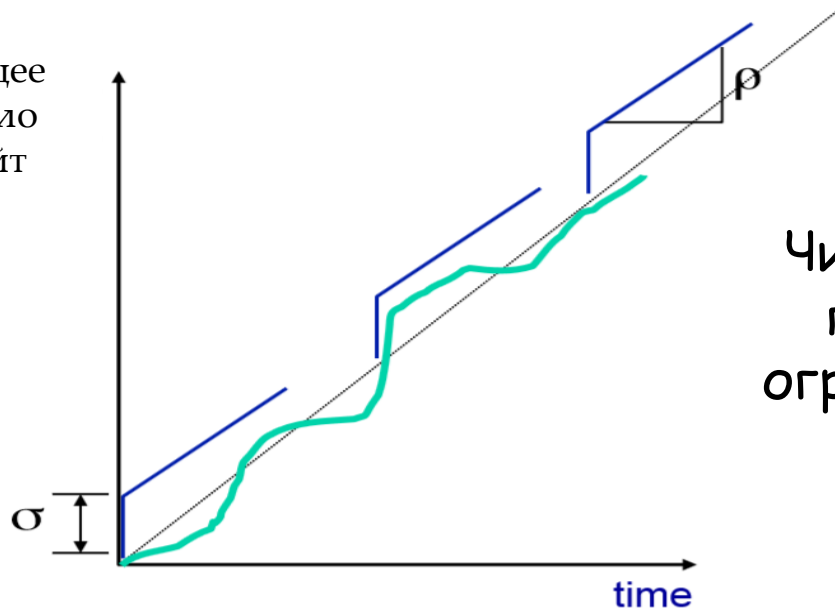


Идея:
Мы не можем управлять процессом поступления, давайте его ограничим



Ограничение трафика

Общее
число
байт



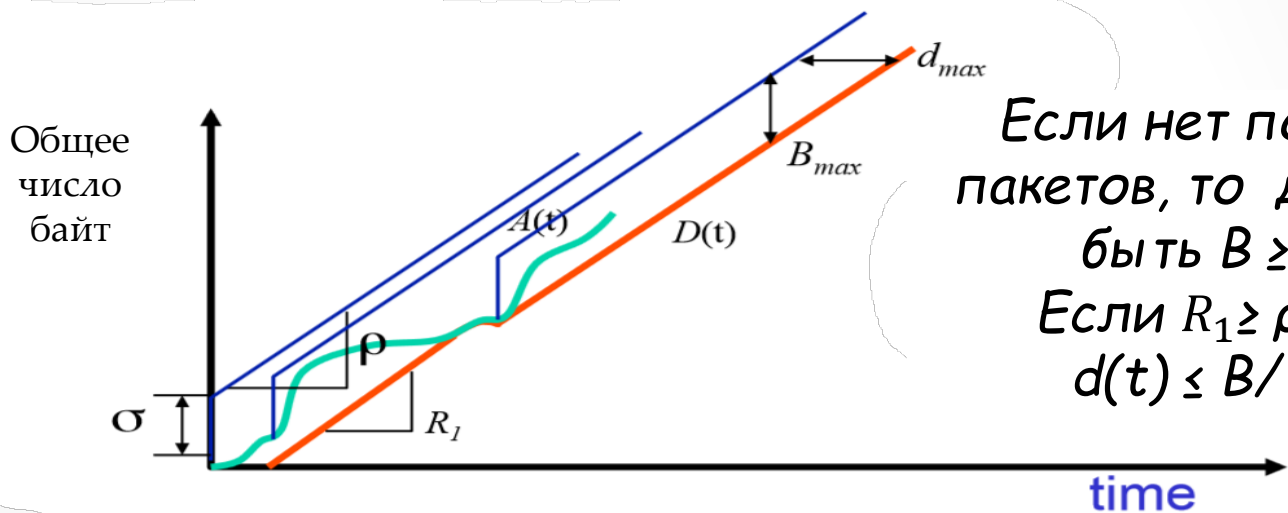
Число бит, которые могут
поступить за период t ,
ограничено величиной $\sigma + \rho t$

В нашем примере

$$\sigma = B, \rho = R_1$$



(σ, ρ) ограничение поступления и минимальная скорость обслуживания

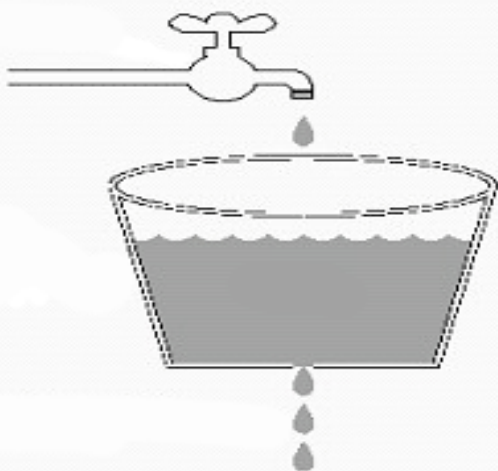


Если нет потери пакетов, то должно быть $B \geq \sigma$.
Если $R_1 \geq \rho$, то $d(t) \leq B / R_1$

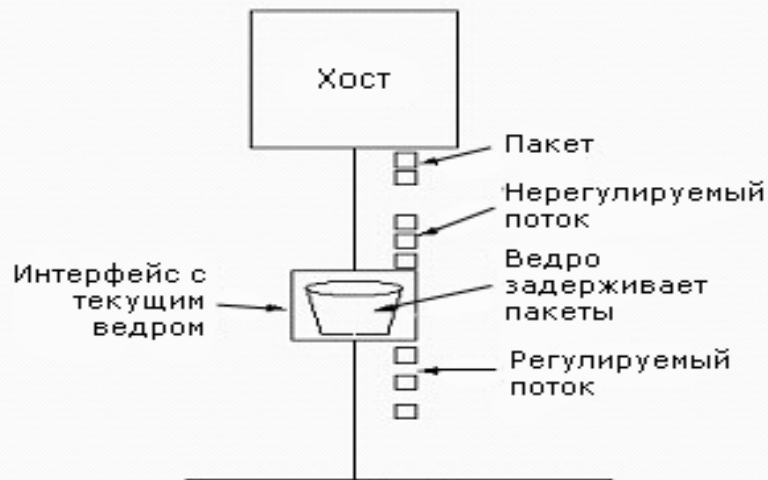
Если поток ограничен текущим буфером и маршрутизатор использует дисциплину обслуживания WFQ, то можно гарантировать e2e задержку.



Пакетный коммутатор: текущий буфер (идея)



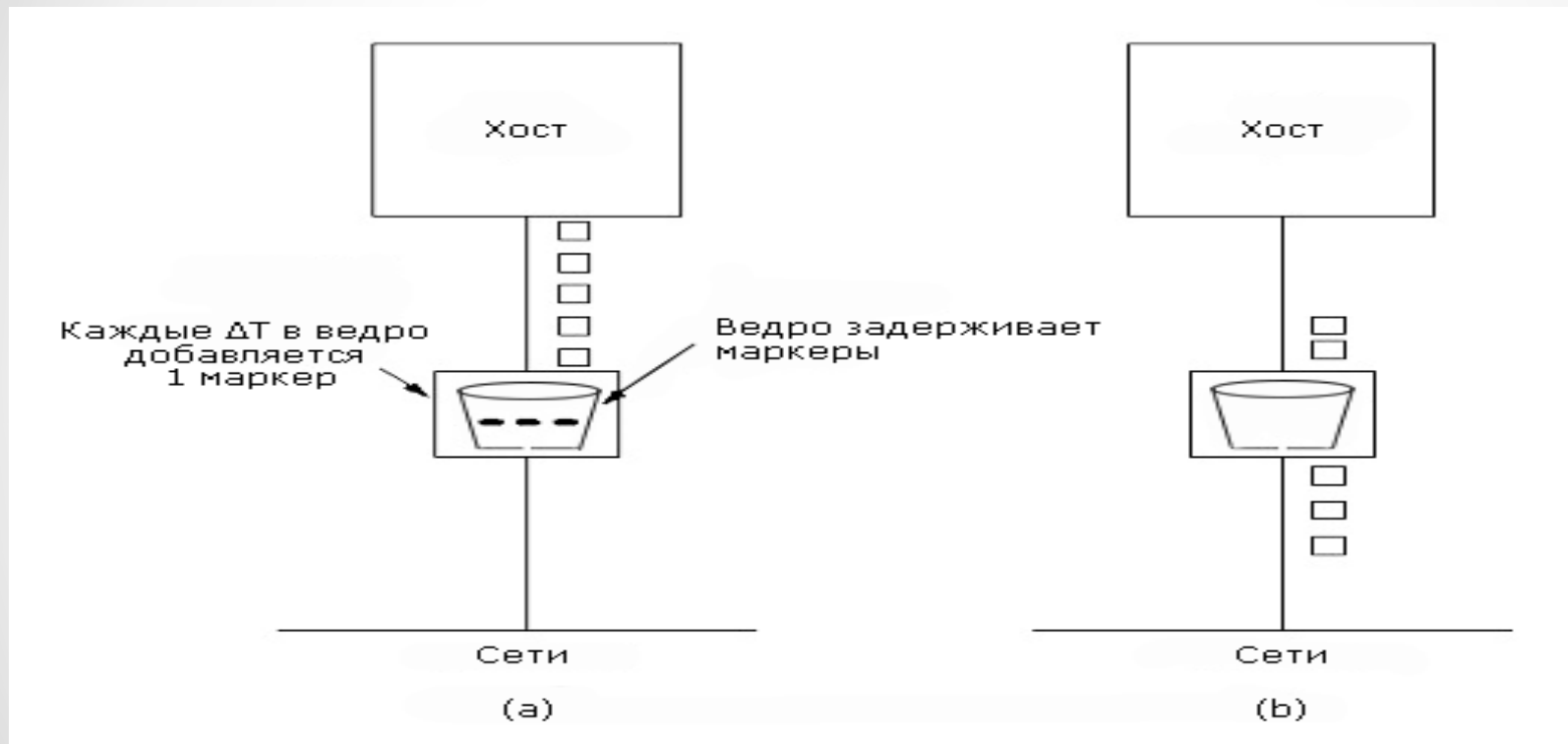
(a)



(b)



Текущий буфер с маркерами





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.
- Самое главное не допустить сброса пакетов. Для этого можно использовать метод «текущего ведра».
- Так можно гарантировать максимальную 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.