



# Интернет: перегрузка

Введение в компьютерные сети

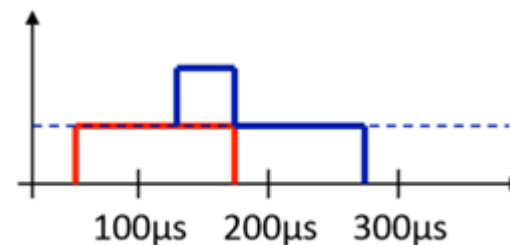
проф. Смелянский Р.А.  
Лаборатория Вычислительных комплексов  
ф-т ВМК МГУ

# План

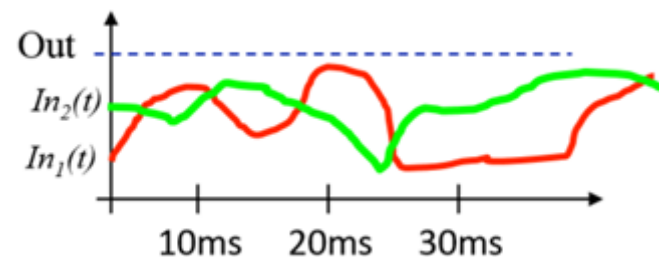
- Что такое перегрузка и как ею можно управлять
- Где располагать управление перегрузкой: Основные подходы
  - В сети
  - Со стороны получателя
- Управление перегрузкой в TCP
  - TCP Tahoe
  - TCP Reno
  - TCP RTT измерение
  - Управление производительностью на практике

# Причины перегрузок

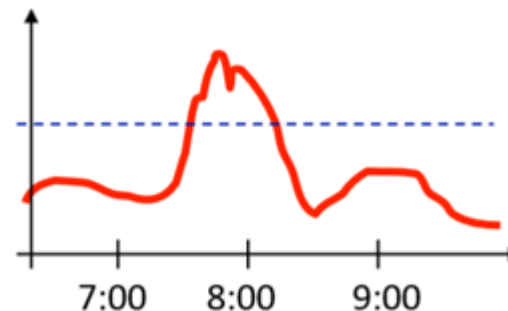
Столкновение двух пакетов в маршрутизаторе



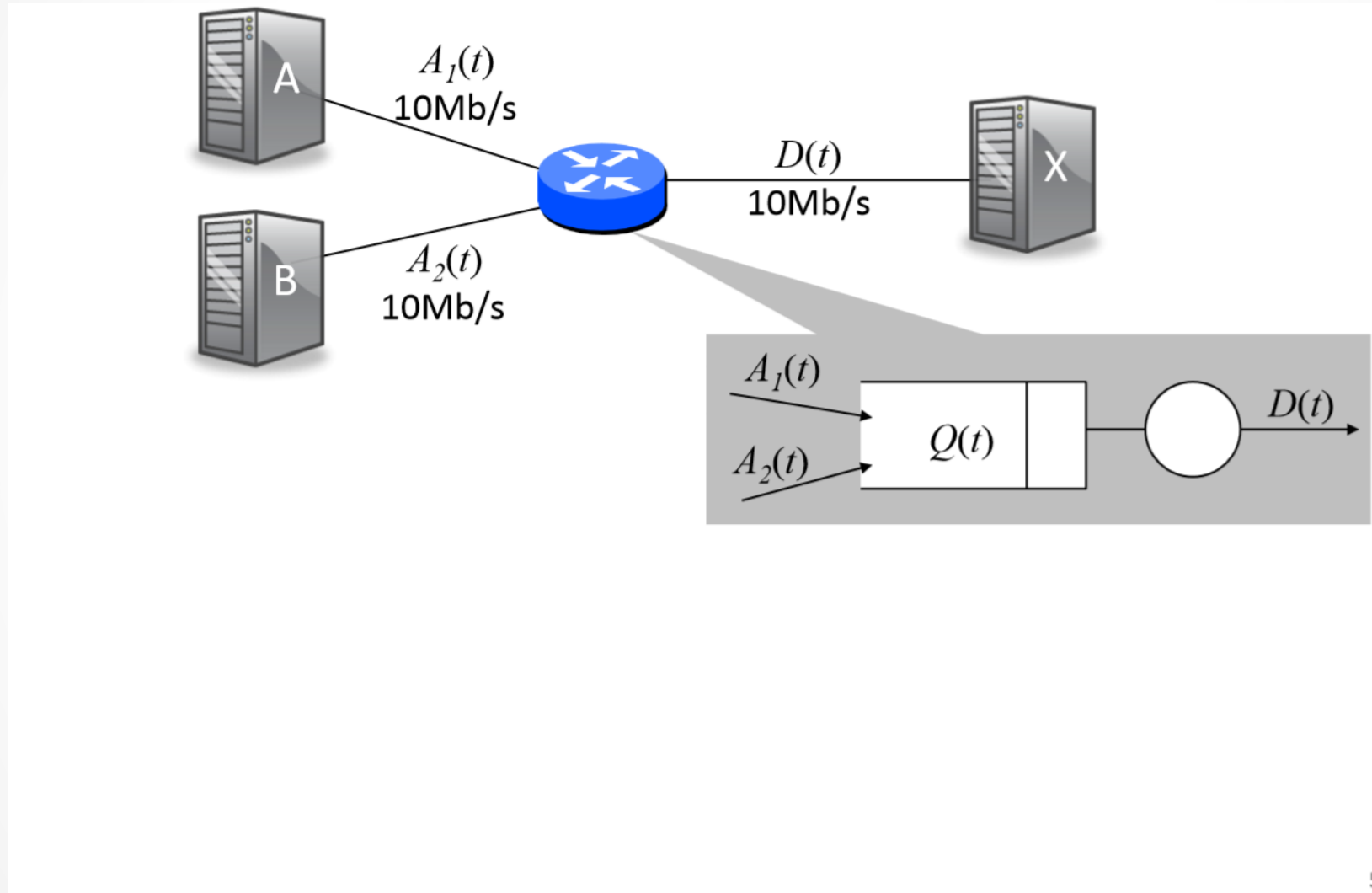
Потоки превысили пропускную способность канала



Слишком много пользователей используют один и тот же канал в одно и то же время

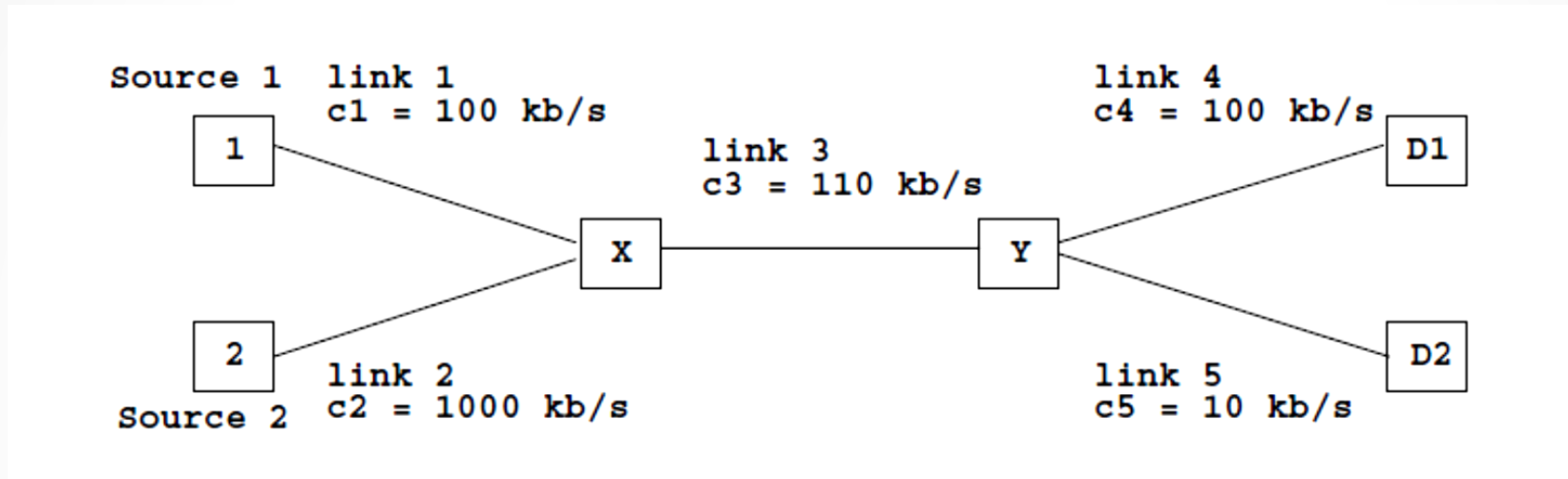


# Что вызывает перегрузку?



# Пример

Источники ничего не знают о распределении пропускных способностей каналов



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

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

**Необходимо, чтобы источник имел обратную связь с сетью**

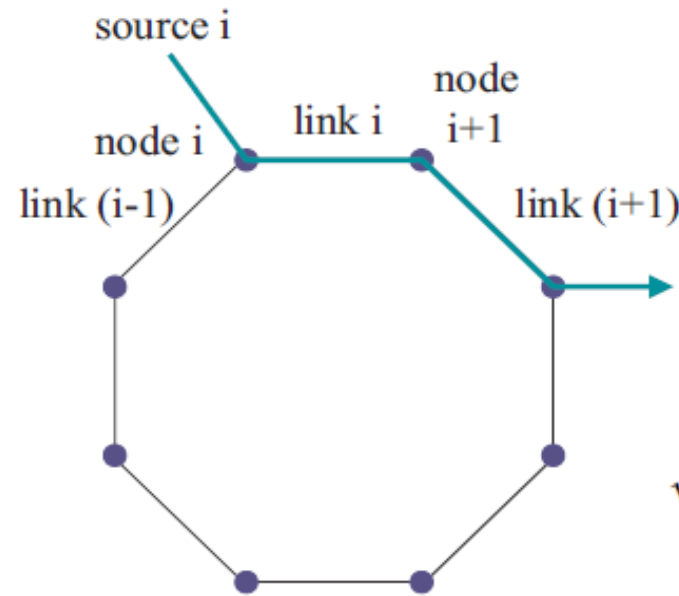
# Пример

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

Пусть для любого  $i$

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

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



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

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

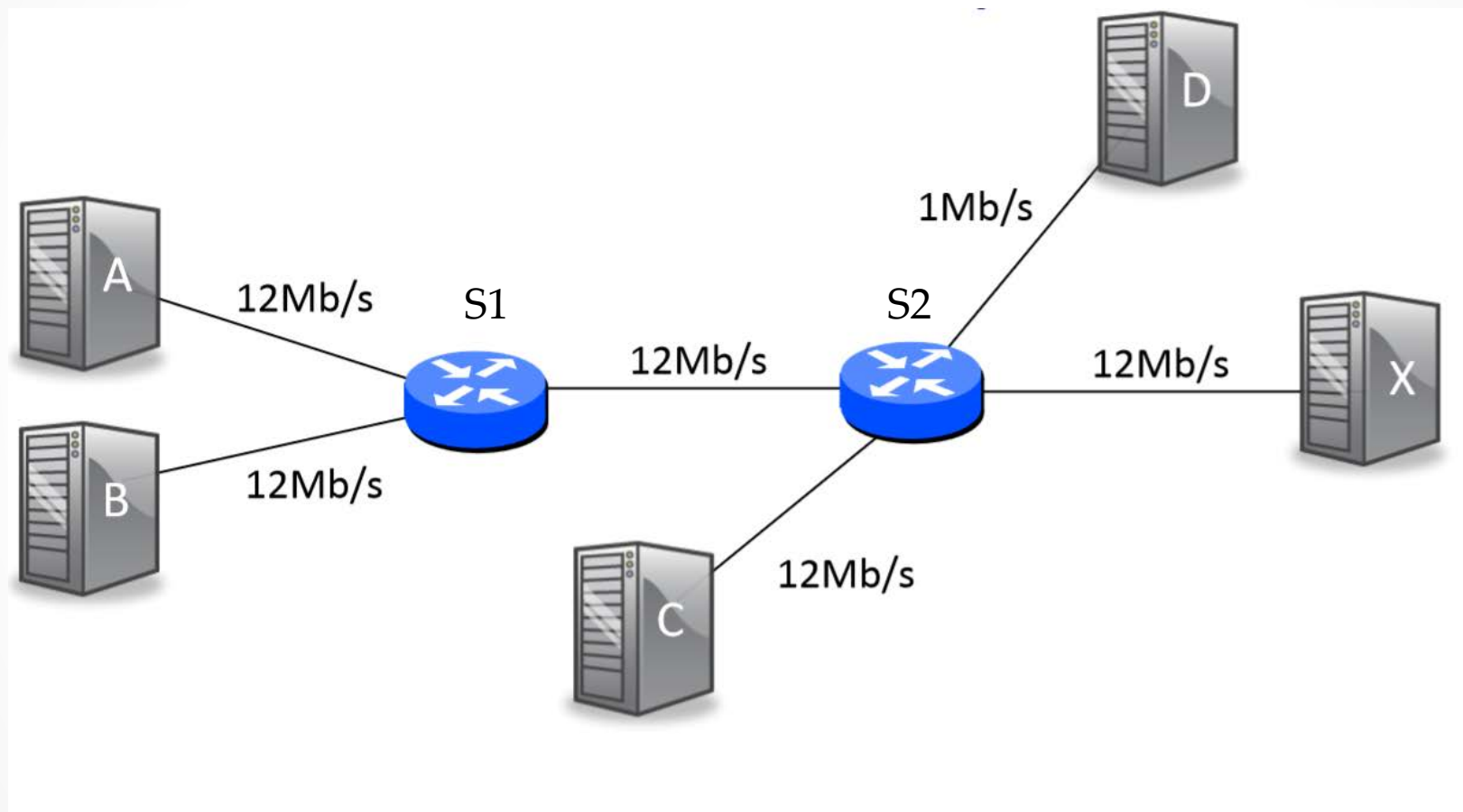
$$\sqrt{1+u} = 1 + \frac{1}{2}u - \frac{1}{8}u^2 + o(u^2)$$

при  $u \rightarrow 0$

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

**Вывод: в сети с коммутацией пакетов источник должен регулировать свою скорость вброса пакетов в сеть в зависимости от состояния сети. В противном случае наступит перегрузка.**

# Попробуем ограничить источники





# Перегрузки неизбежны!

(может быть это и хорошо)

- Коммутацию пакетов используют потому, что она позволяет эффективно использовать пропускную способность каналов. Поэтому буферы в маршрутизаторах часто заполнены.
- Если буферы пусты, задержки малы, но интенсивность использования сети низкая.
- Если буферы постоянно заполнены, задержки возрастают, но интенсивность использования сети также возрастает

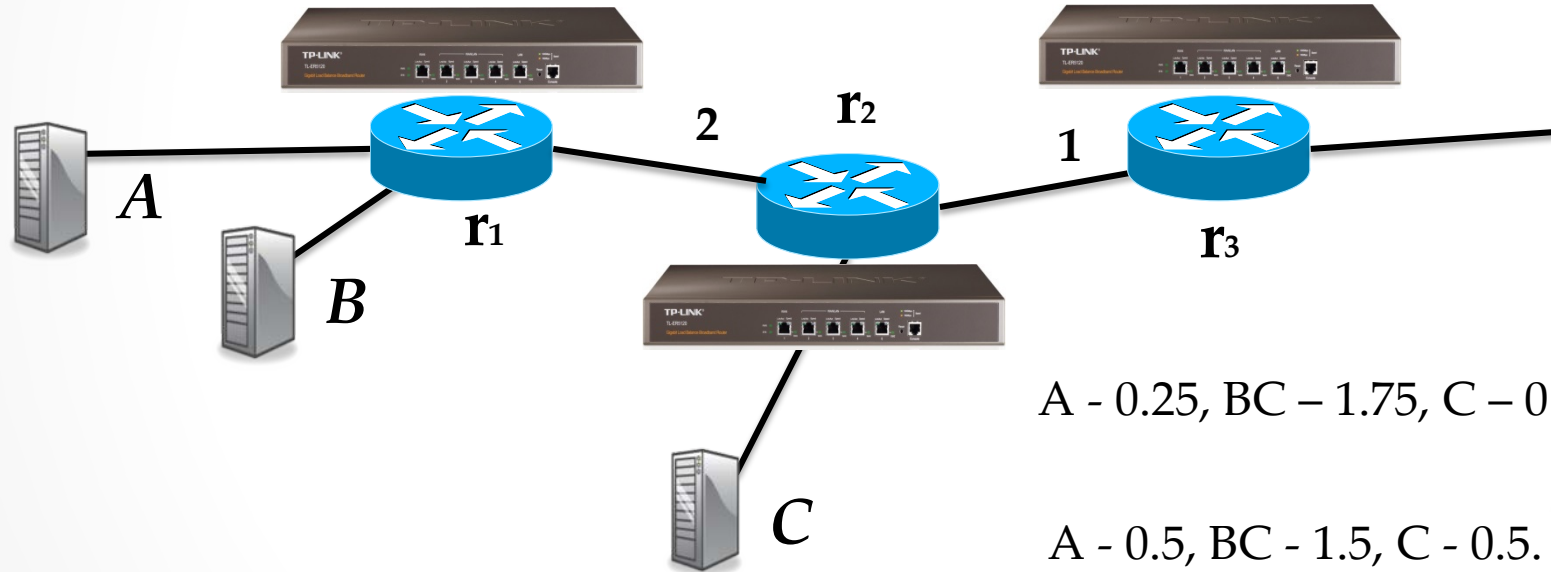
**Перегрузки неизбежны и даже желательны!**



# Промежуточный итог

1. Перегрузки неизбежны и даже желательны
2. Перегрузки возможны в разных случаях
  - Коллизия пакетов в маршрутизаторе
  - Некоторые потоки шлют данные с очень большой скоростью
  - Толпа пользователей объявилась в сети
3. Когда пакет сброшен, то все ресурсы, которые были потрачены до того чтобы его доставить к месту сброса, потрачены зря
4. Если пакет был сброшен, то его ретрансмиссия может усугубить перегрузку
5. Необходимо ввести понятие справедливости, чтобы решить как потоки будут разделять ресурсы перегруженного канала

# Понятие справедливости



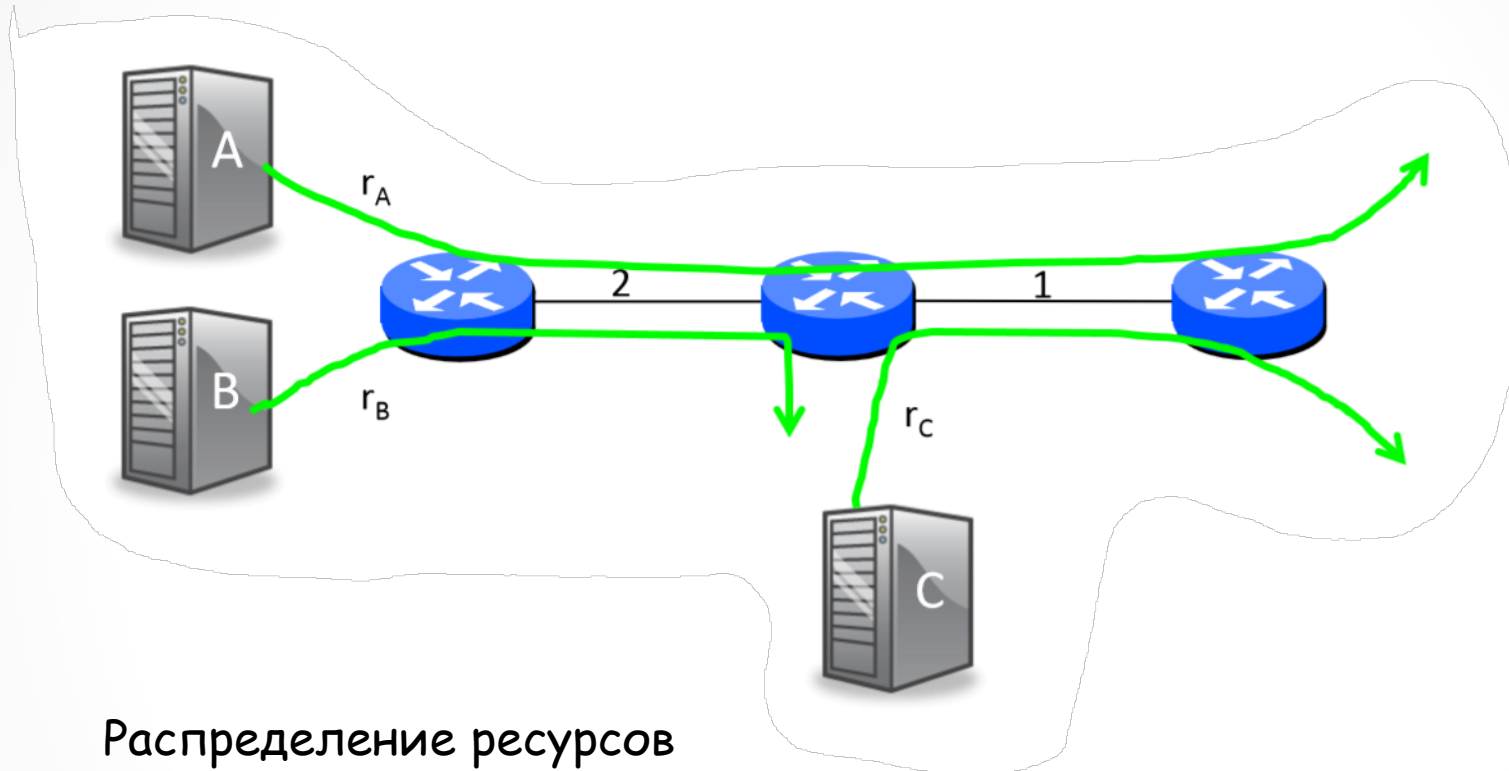
A - 0.25, BC - 1.75, C - 0.75 Суммарно = 2.75

A - 0.5, BC - 1.5, C - 0.5. Суммарно - 2.5

Здесь и A и C на r2r3 по 0.5

Что такое справедливость и как ее измерить ?

# Справедливость и пропускная способность



## Распределение ресурсов

1.  $r_A = 0.25, r_B = 1.75, r_C = 0.75$  Общая пропускная способность = 2.75
2.  $r_A = 0.5, r_B = 1.5, r_C = 0.5$  Общая пропускная способность = 2.5

Случай 2 «справедливее».

Как мы можем определить справедливость?

# Теория

Пусть есть:

- $s = 1, \dots, S$  - множество источников
- $l = 1, \dots, L$  - множество линий
- $A_{l,s}$  – доля потока от  $s$  на линии  $l$

Модель сети  $M = (x, A)$ , где  $x$  –  
распределение потоков от источников  $s$

*Распределение потоков допустимо*

- $\forall s: 1 \leq s \leq S \rightarrow x_s \geq 0$
- $\forall l: 1 \div L: \sum_{i=1}^S A_{l,i} x_i \leq c_l$

*Линия  $l$  – насыщена если  $c_l = \sum_{i=1}^S A_{l,i} x_i$*

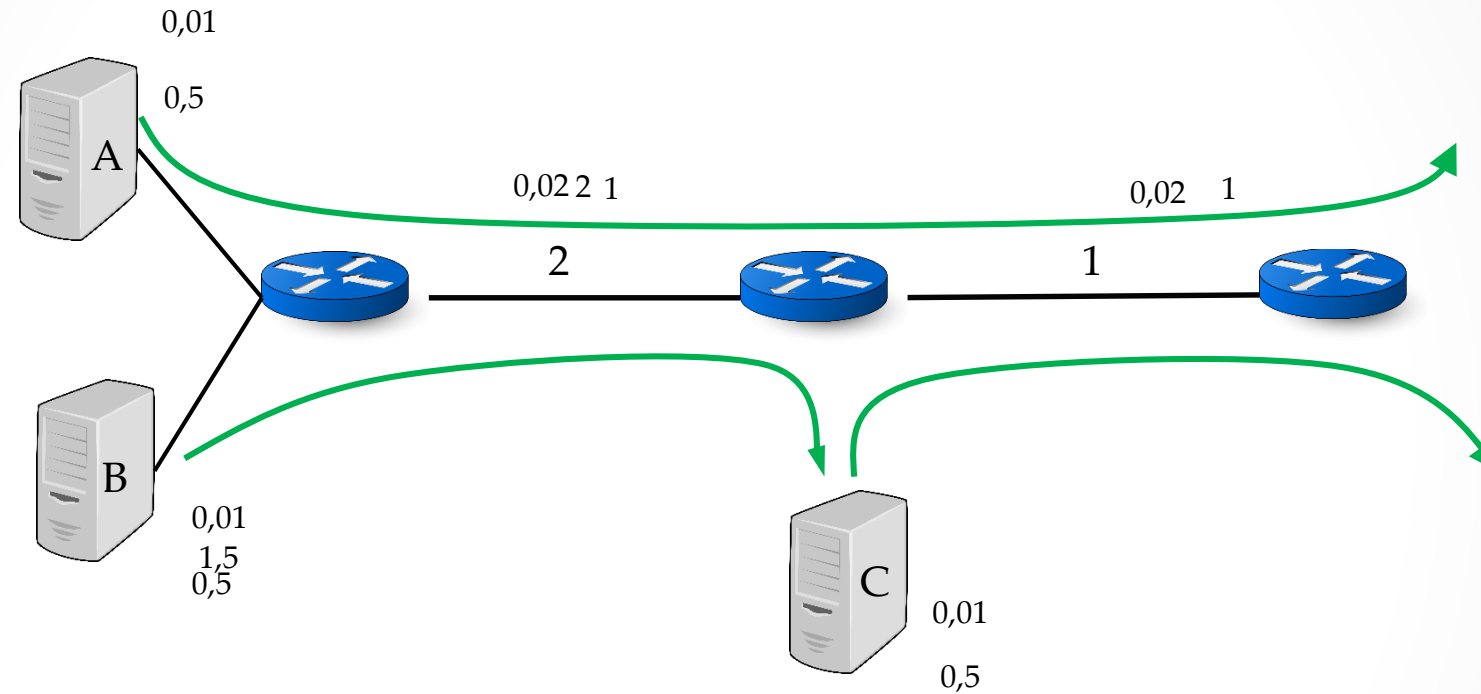
*Линия  $l$  – критична для  $s$  тогда и только тогда*

- $l$  – насыщена
- $s$  имеет максимальный поток среди всех источников, использующих  $l$ :
- $\forall s': A_{l,s} \geq 0 \rightarrow x_s \geq x_{s'}$

**Определение:** допустимое распределение потоков является *max-min* распределением если существует такой источник потока, что любое другое допустимое распределение  $y$  такое, что  $y_s > x_s$ , то  $\exists s': x_{s'} \leq x_s$  и  $y_{s'} < x_{s'}$

**Теорема 1:** *Распределение потоков  $x$  есть max-min тогда и только тогда когда у каждого источника есть критичная линия*

# Алгоритм построения max-min распределения (постепенного заполнения)



## Распределение ресурсов

1.  $r_A = 0.25, r_B = 1.75, r_C = 0.75$  Общая пропускная способность = 2.75
2.  $r_A = 0.5, r_B = 1.5, r_C = 0.5$  Общая пропускная способность = 2.5

Случай 2 «справедливее».

Как мы можем определить справедливость?

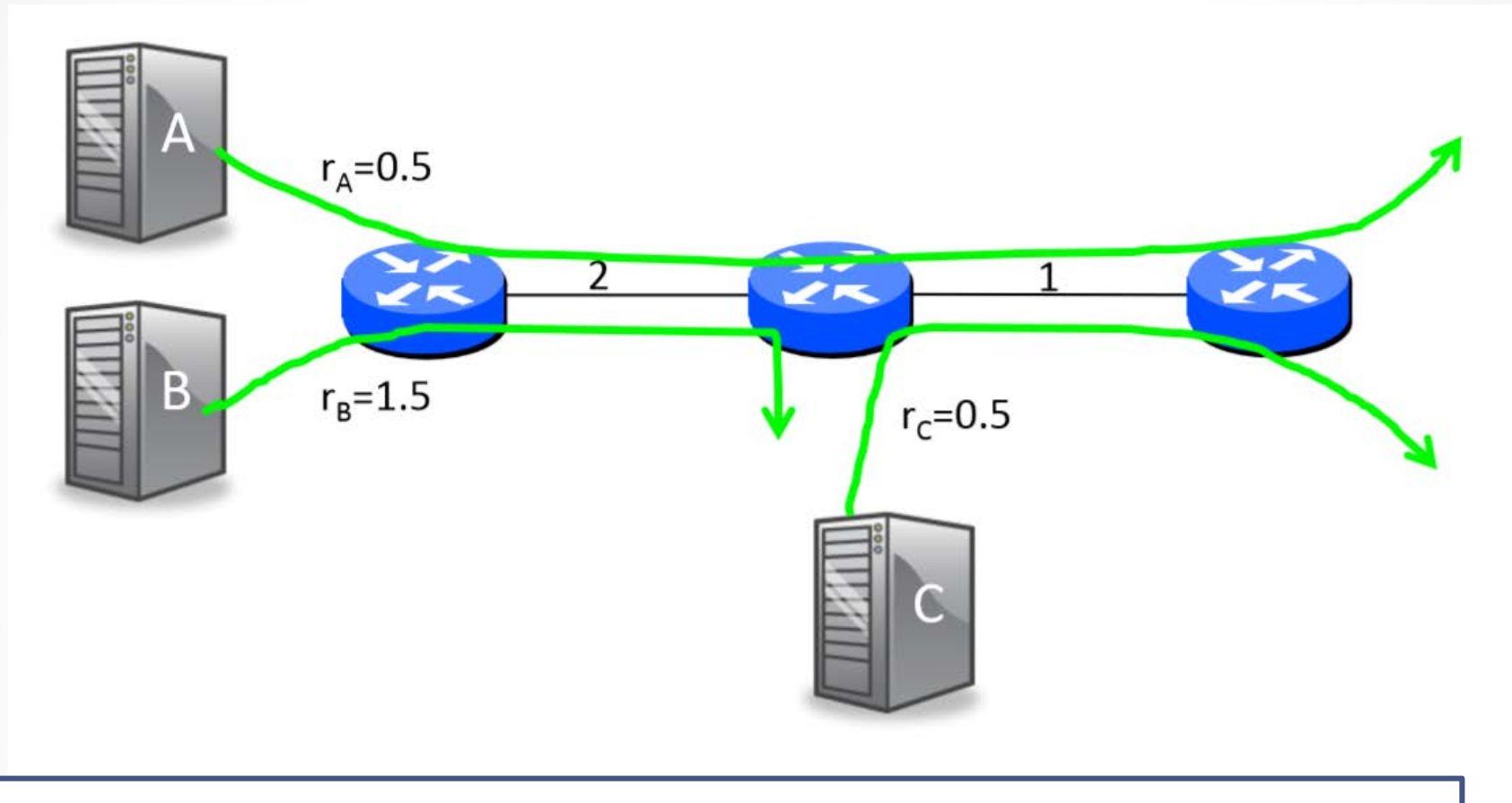


# Max-min справедливость

Определение:

Распределение *max-min справедливо* если нельзя увеличить скорость какого-нибудь потока, не понизив скорости другого, меньшего потока

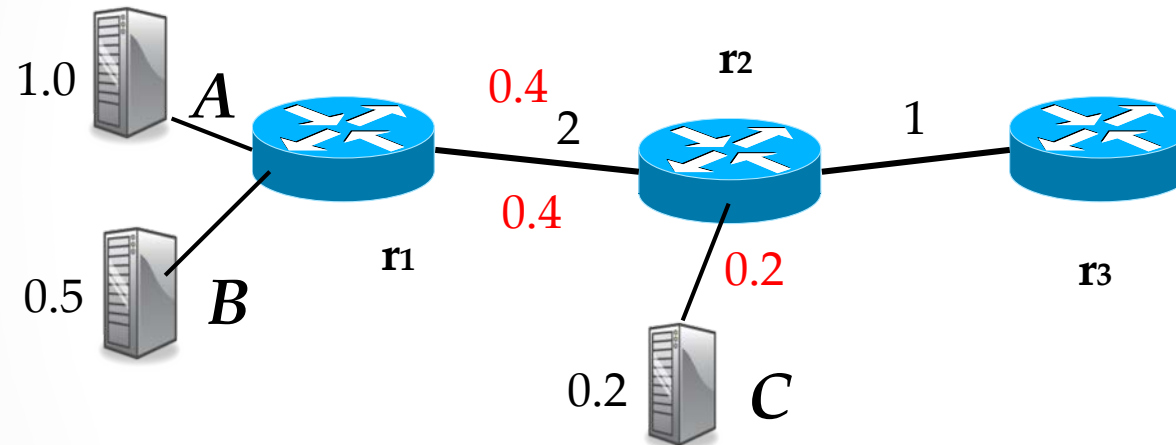
# Max-min справедливое распределение



**Если max-min справедливое распределение существует, то оно единственное для заданной ТОПОЛОГИИ.**

# Max-min справедливое распределение на одном канале

- Определение интуитивно верно для одного канала





# Цели управления перегрузкой

1. **Высокая пропускная способность - каналы загружены максимально, чтобы обеспечить высокую скорость потоков**
2. **Быстрая реакция на изменения распределения потоков в сети**
3. **Распределенное управление**



# Где и как управлять перегрузкой?

Введение в компьютерные сети

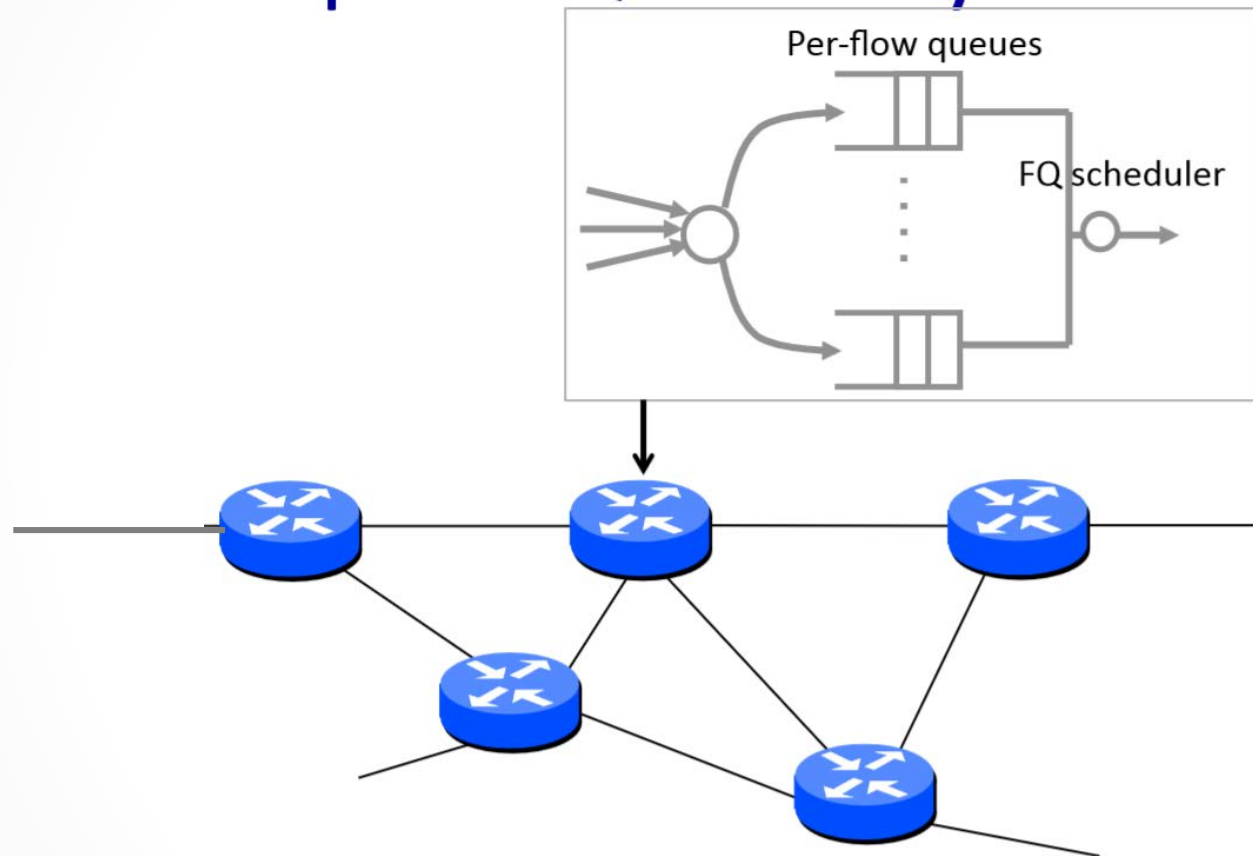
проф. Смелянский Р.Л.  
Лаборатория Вычислительных комплексов  
ф-т ВМК МГУ


Мы хотим построить алгоритм управления перегрузками такой, чтобы:

1. Высокая пропускная способность: каналы загружены, скорость потоков высокая
2. max-min справедливость
3. Быстрая реакция на изменения состояния сети
4. Распределенное управление

- Где надо размещать управление перегрузкой:
  - Очередь со справедливой дисциплиной (FWQ) в каждом маршрутизаторе
  - В сети
  - На каждом хосте
- Скользящее окно и AIMD

# Пример: FQ на каждом маршрутизаторе





# Управление перегрузками в ТСП

## (управление на хостах)

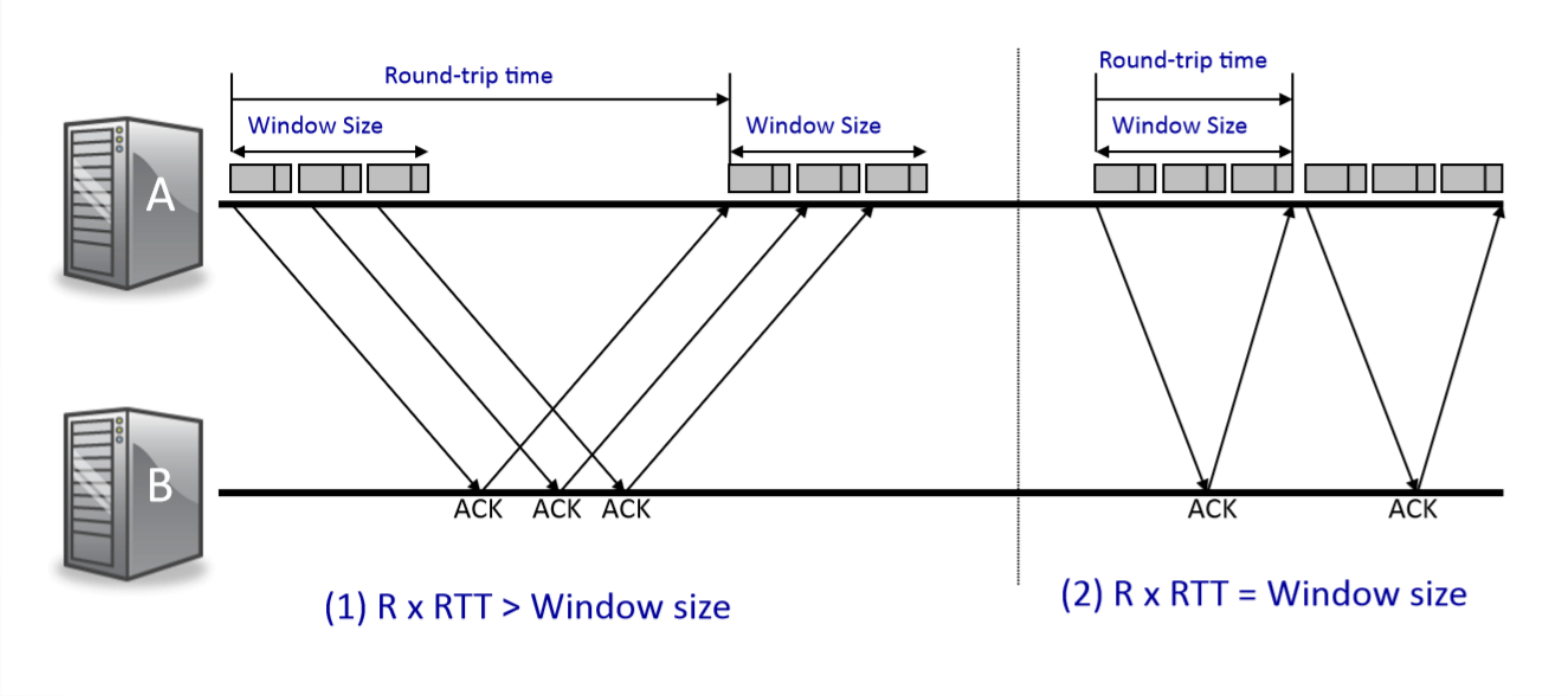
В ТСП управление перегрузкой размещается на  
конечном хосте

- Реакция на события, наблюдаемые на конечном хосте (например, потеря пакета).
- Использование скользящего окна, предназначенного для управления потоком в ТСП
- Постараться оценить сколько пакетов можно безопасно отправить в сеть одновременно.

# Скользящее окно



# Скользящее окно в ТСР



# Скользящее окно в ТСР

ТСР варьирует число пакетов отправляемых в сеть, изменяя размер скользящего окна:

$$\text{размер окна} = \min\left\{ \underbrace{\text{Объявленное окно}}_{\text{получатель}}, \underbrace{\text{Окно перегрузки}}_{\text{отправитель (cwnd)}} \right\}$$

Как определить размер cwnd?





# AIMD управление перегрузкой для одного потока

Введение в компьютерные сети

проф. Смелянский Р.Л.  
Лаборатория Вычислительных комплексов  
ф-т ВМК МГУ

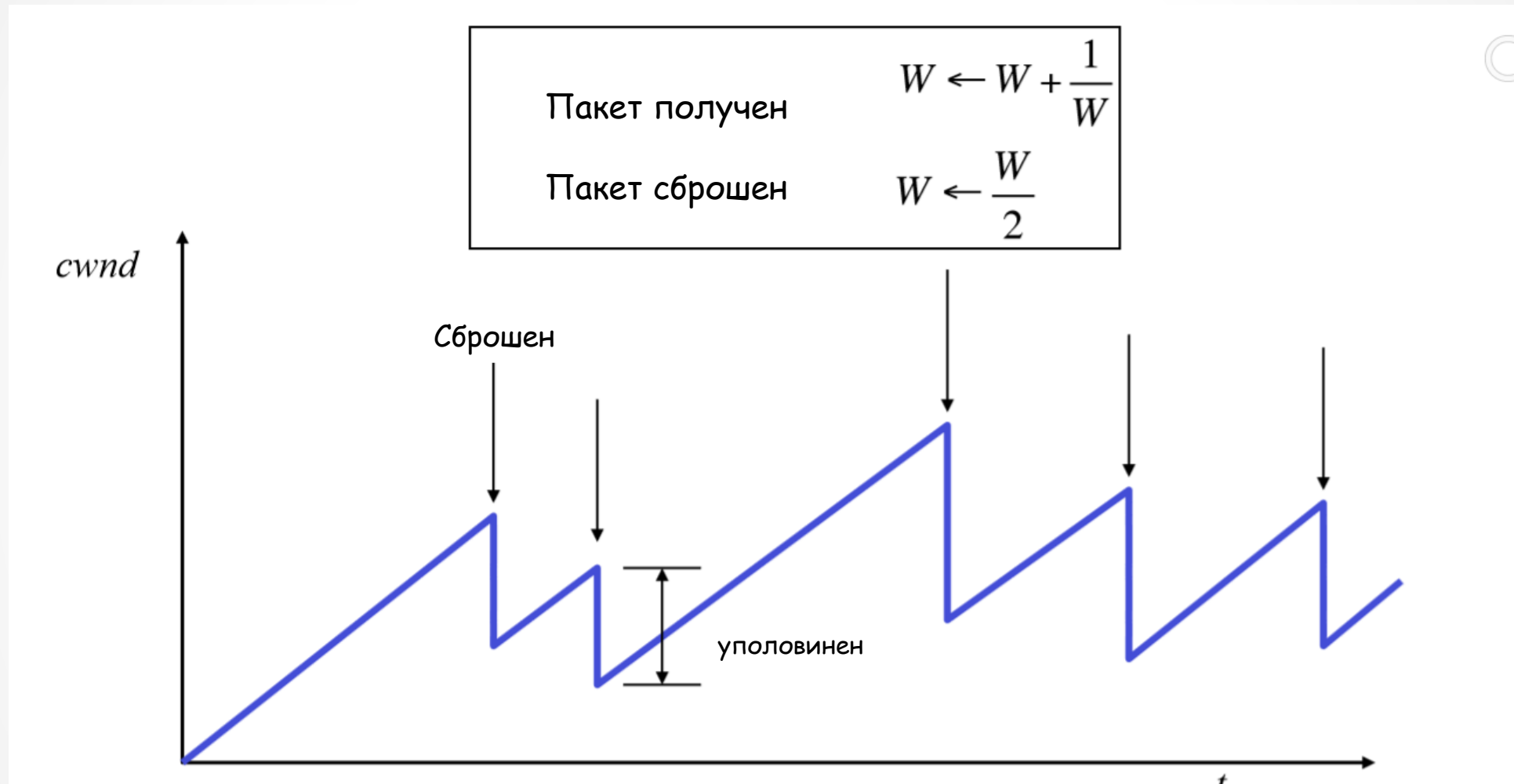


# AIMD

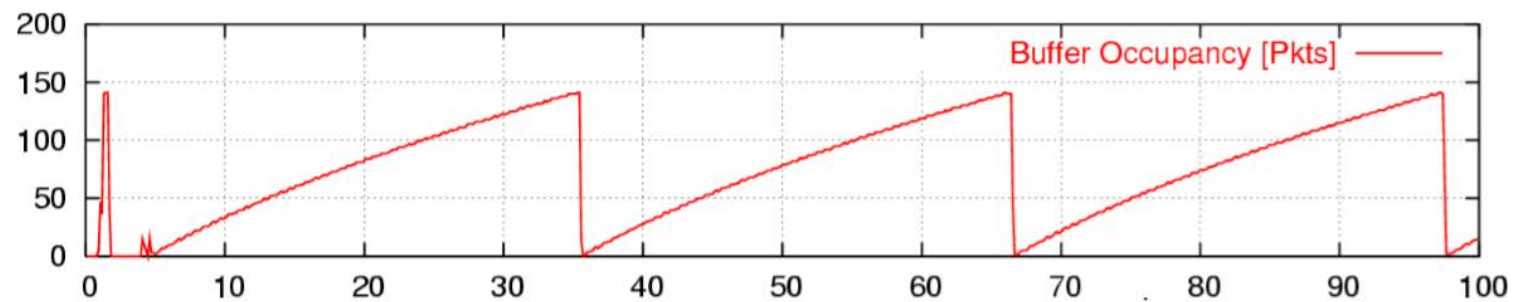
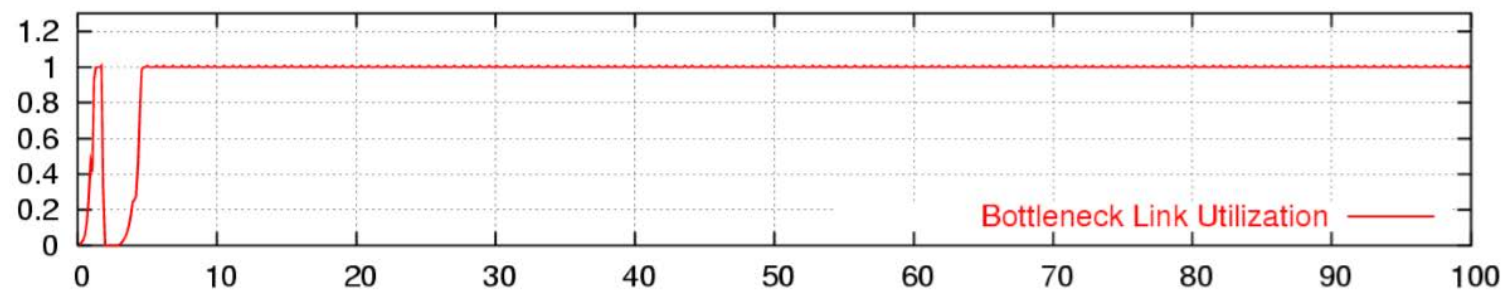
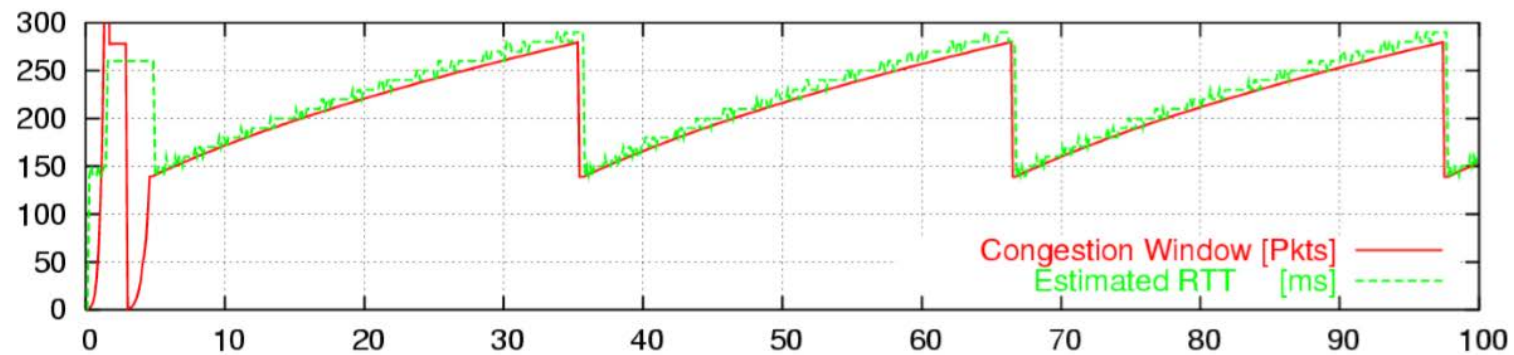
(Additive Increase Multiple Decrease)

- Если пакет получен успешно:  $W \leftarrow W + \frac{1}{W}$
- Если пакет был сброшен:  $W \leftarrow \frac{W}{2}$

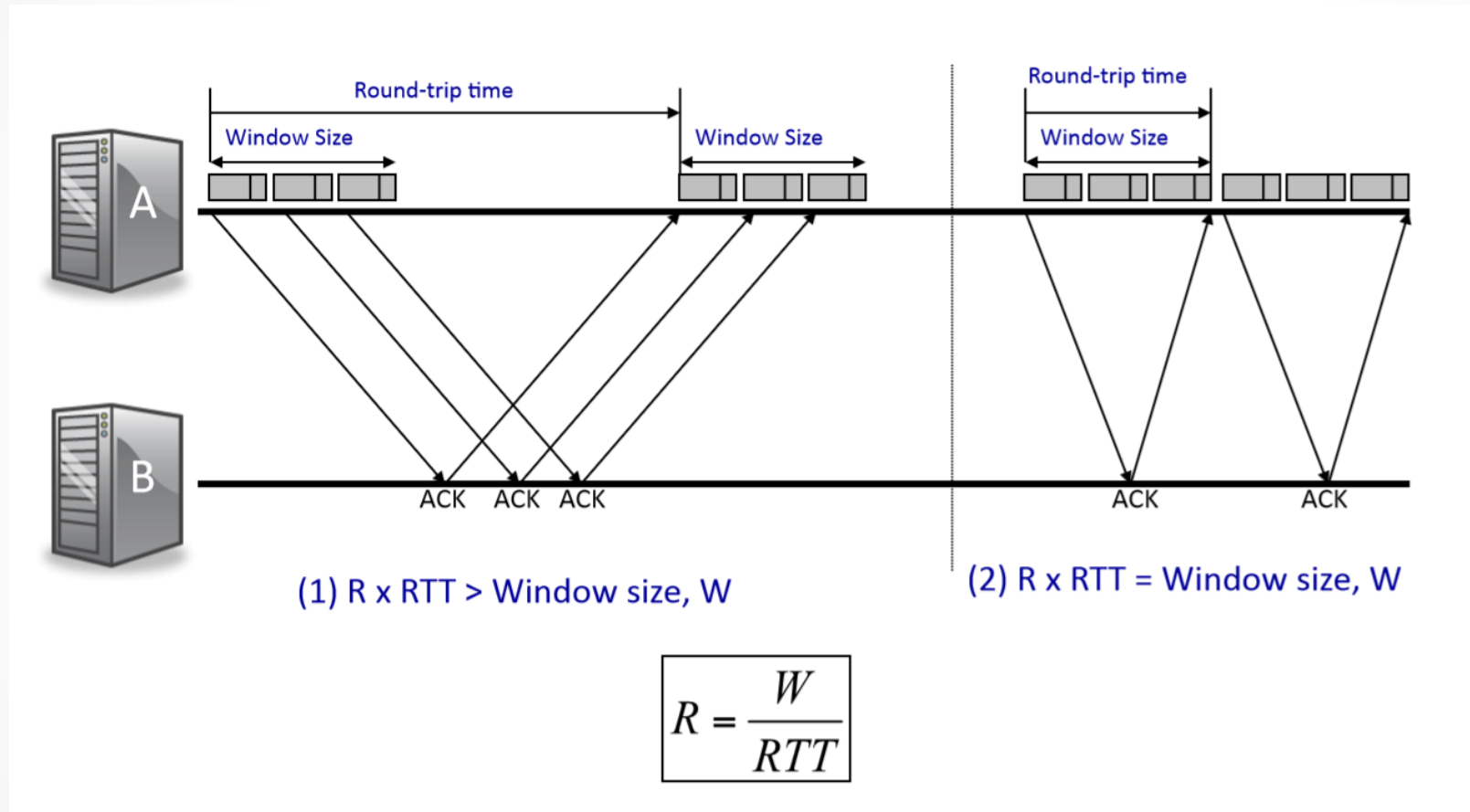
# Пила AIMD



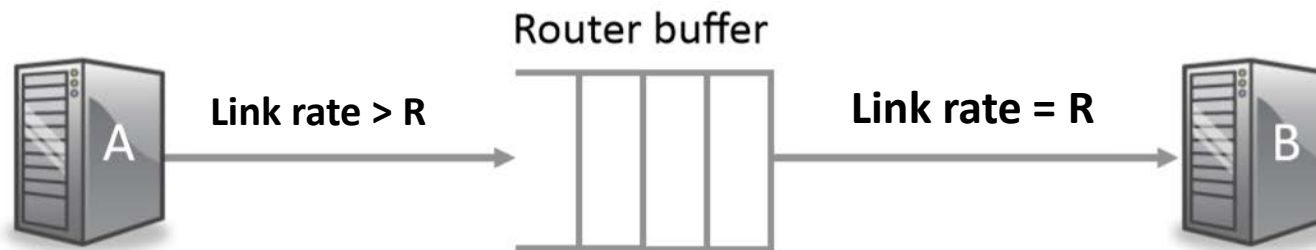
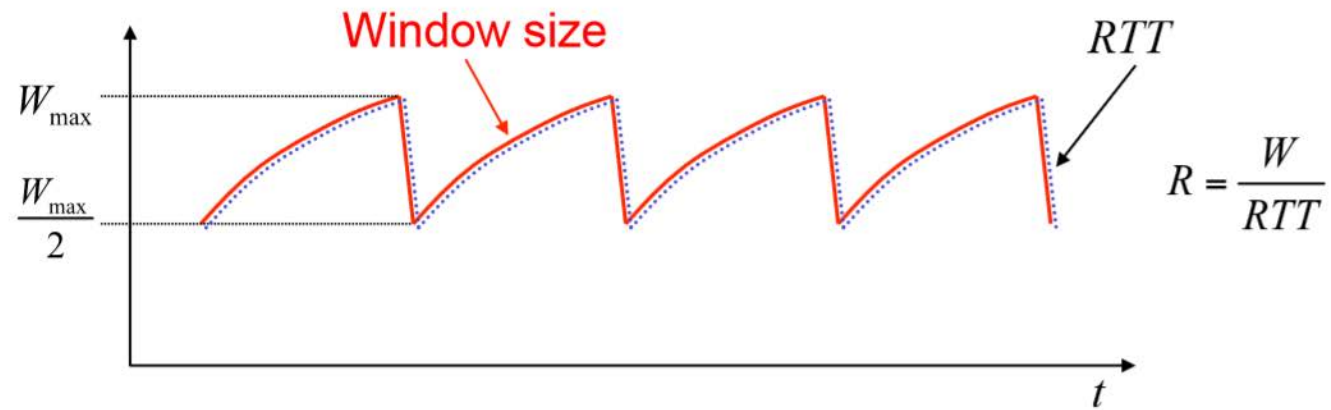
# Примеры динамики одиночного потока



# Скорость отправки для одиночного потока

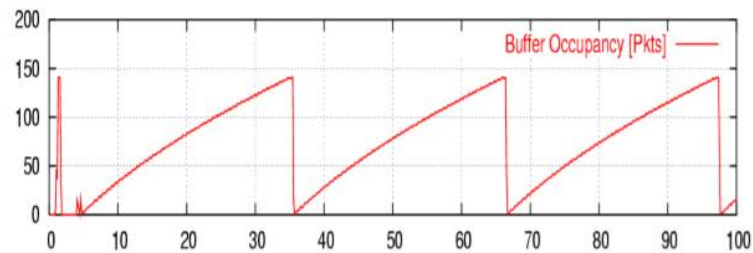
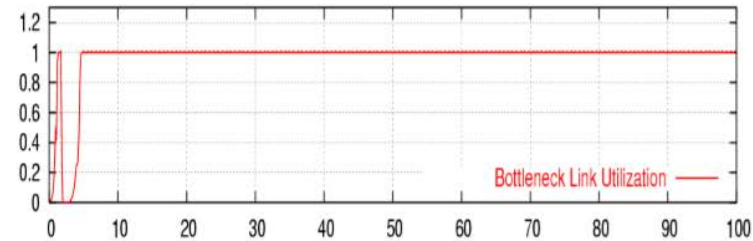
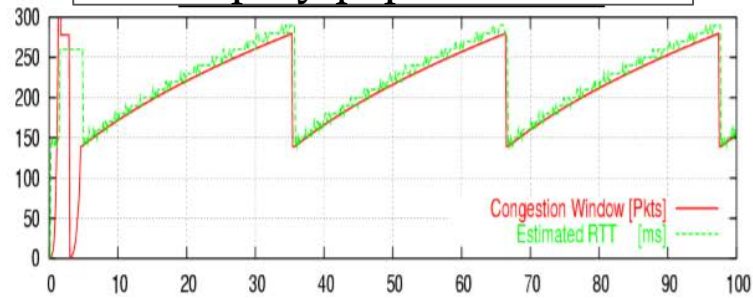


# Скорость отправки для одиночного потока

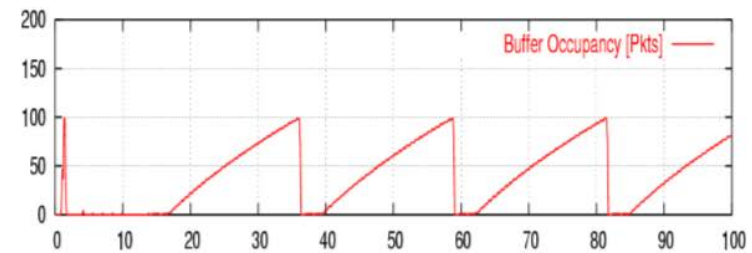
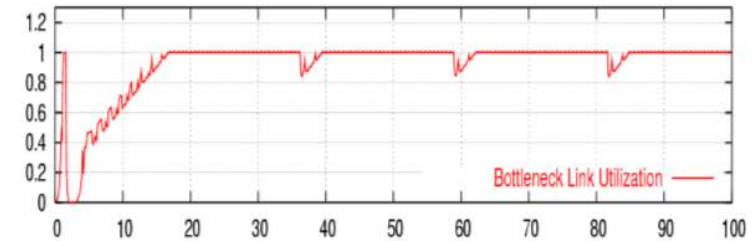
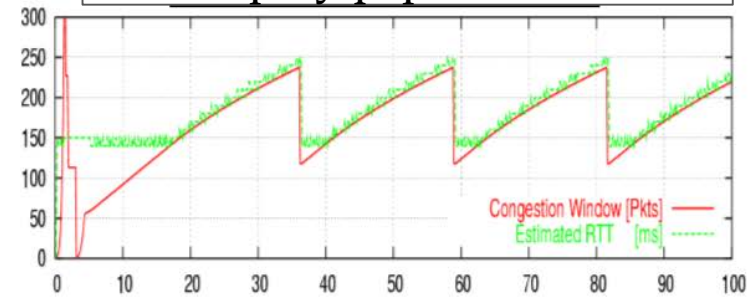


# Насколько большим должен быть буфер?

Размер буфера  $W = RTT * R$



Размер буфера  $W < RTT * R$



# Промежуточные выводы

## 1. В ТСП управление перегрузкой располагается на хосте

- Реакция на события, наблюдаемые на конечном хосте (например, потеря пакета).
- Использование скользящего окна, предназначенного для управления потоком в ТСП
- Постараться оценить сколько пакетов можно безопасно отправить в сеть одновременно.
- Изменять размер окна в соответствии с алгоритмом AIMD



# Комментарии для одиночного потока

1. Окно увеличивают, сокращают в соответствии с AIMD
2. ... пробировать как много байт канал еще может вместить
3. Пилообразное поведение - нормальная форма динамики
4. Скорость отправки постоянная
5. Размер буферного пространства определяет соотношение -  $RTT \times R$



# Управление перегрузкой: AIMD с несколькими потоками

Введение в компьютерные сети

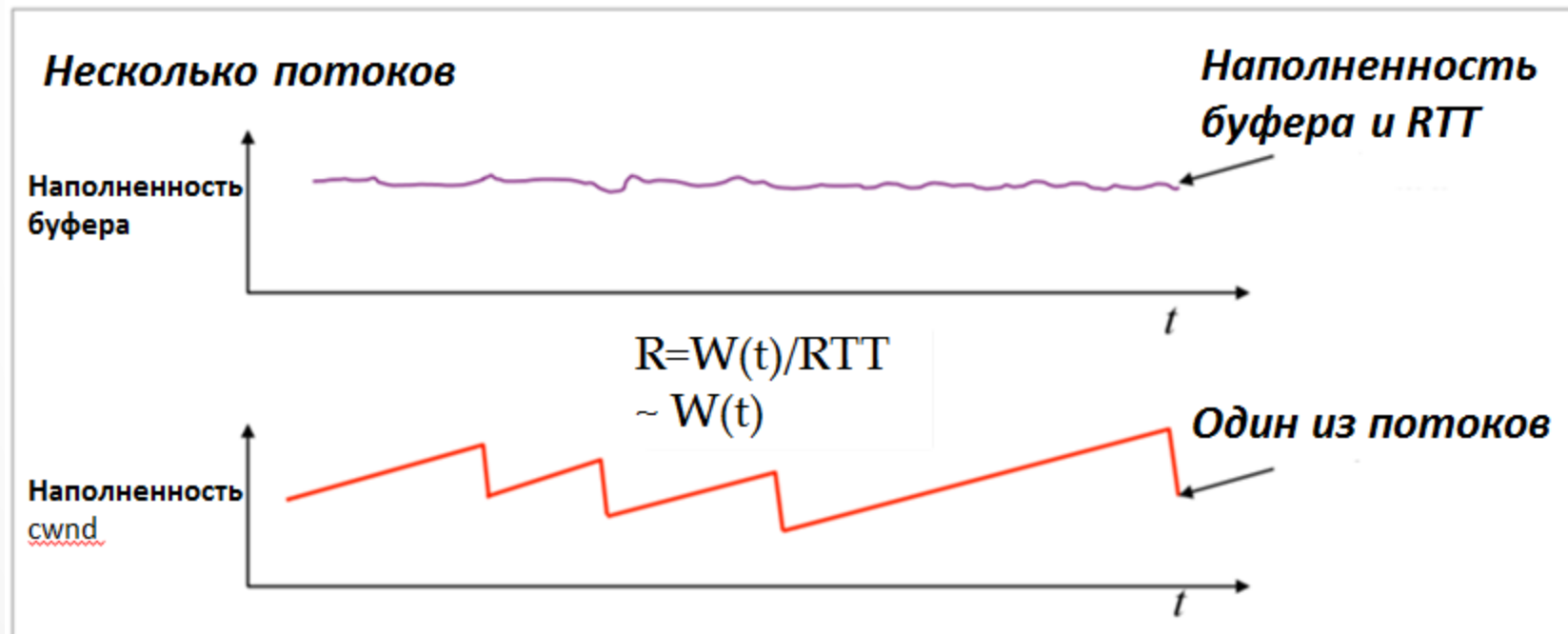
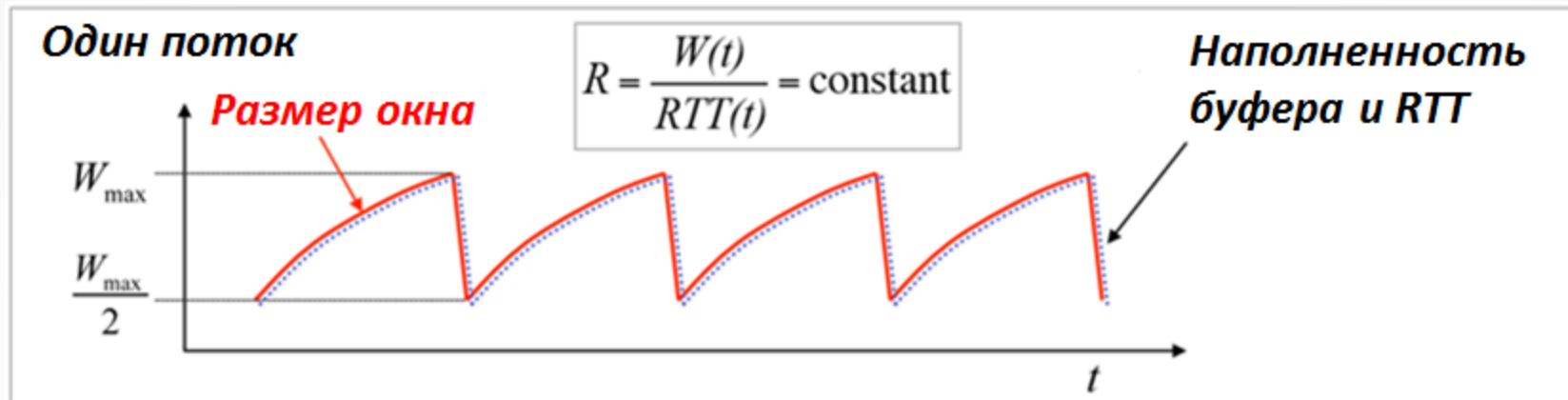
проф. Смелянский Р.А.  
Лаборатория Вычислительных комплексов  
ф-т ВМК МГУ



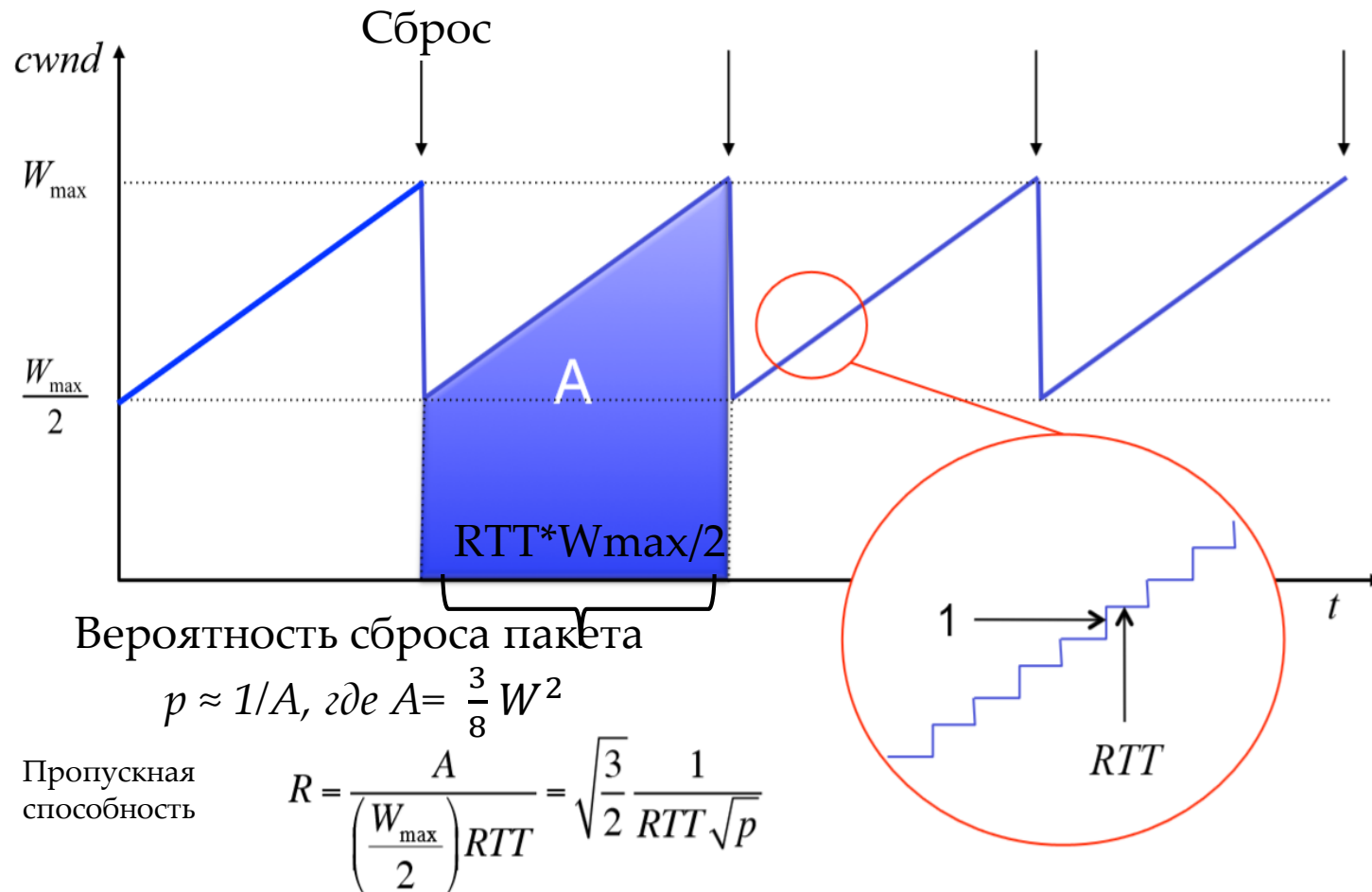
## Буфер маршрутизатора



# Один поток vs много потоков



# Интуитивная геометрическая интерпретация



# Интерпретация уравнения скорости

$$R = \sqrt{\frac{3}{2}} \frac{1}{RTT \sqrt{p}}$$

$$RTT \rightarrow 0 \Rightarrow R \rightarrow \infty$$

$$p \rightarrow 0 \Rightarrow R \rightarrow \infty$$

$$\frac{R_1}{R_2} = \sqrt{\frac{p_2}{p_1}}$$

# Комментарии для нескольких потоков

1. Окно увеличивают/сокращают в соответствии с AIMD
2. ... пробировать как много байт канал еще может вместить
3. В «узком месте» будут скапливаться пакеты разных потоков
4. Скорость отправки меняется в зависимости от размера окна
5. AIMD очень чувствителен к вероятности потери пакетов
6. AIMD ущемляет потоки с большим RTT

# Заключение:

## Управление перегрузками в ТСП

В ТСП управление перегрузкой размещается на  
конечном хосте

- Реакция на события, наблюдаемые на конечном хосте (например, потеря пакета).
- Использование скользящего окна, предназначенного для управления потоком в ТСП
- Механизм AIMD позволяет оперативно оценить сколько пакетов можно безопасно отправить в сеть одновременно.