Интернет: перегрузка

Введение в компьютерные сети

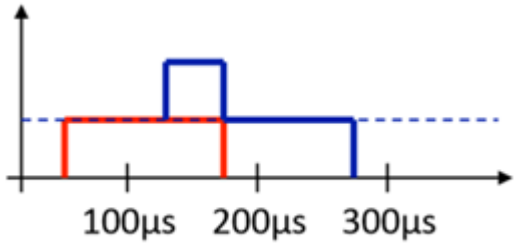
чл.-корр. РАН Смелянский Р.Л.
Кафедра АСВК

План

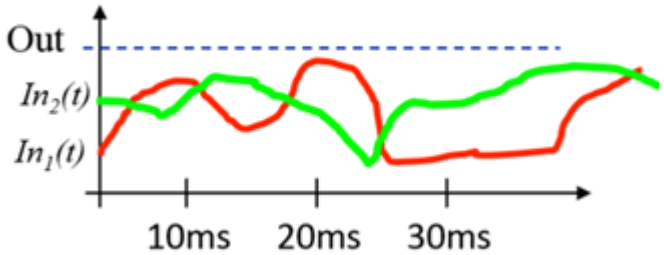
- Что такое перегрузка?
- Почему она возникает и как ею можно управлять
- Где располагать управление перегрузкой: Основные подходы
  - В сети
  - Со стороны получателя
- Управление перегрузкой в TCP
  - TCP Tahoe
  - TCP Reno
  - TCP RTT измерение
  - Управление производительностью на практике

Причины перегрузок

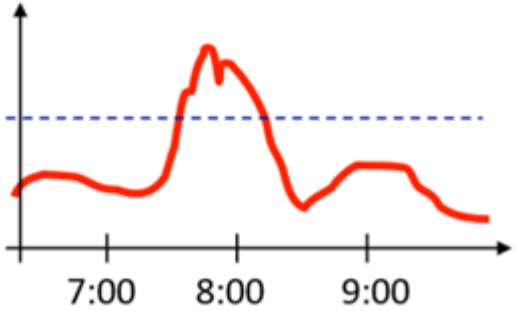
Столкновение двух пакетов в маршрутизаторе



Потоки превысили пропускную способность канала



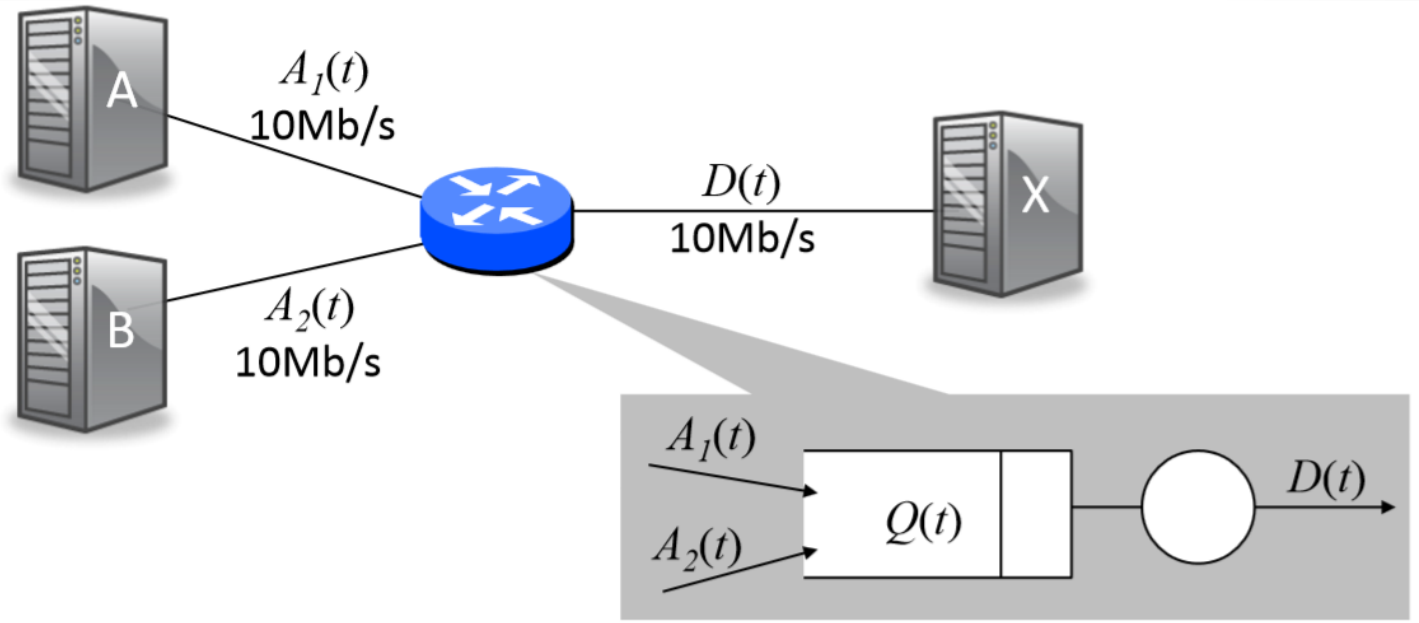
Слишком много пользователей используют один и тот же канал в одно и то же время



Потеря UDP дейтаграммы

Потеря TCP сегмента

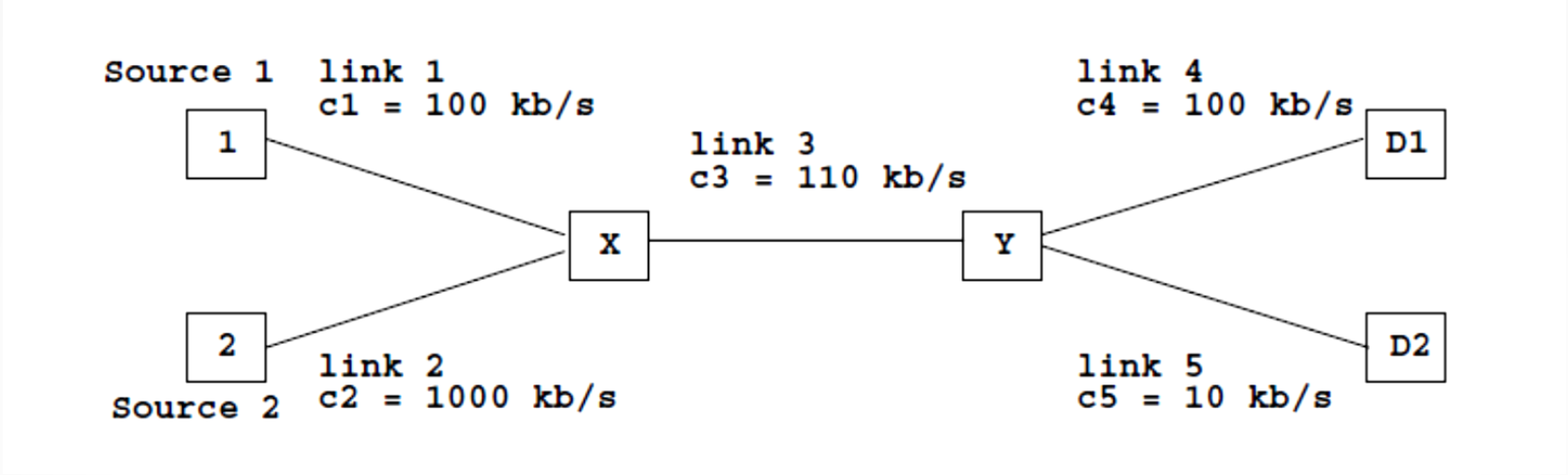
Что вызывает перегрузку?



5

Пример

Источники ничего не знают о распределении пропускных способностей каналов



$\lambda_1 = 100$kb/s $\lambda_2 = 1000$kb/s

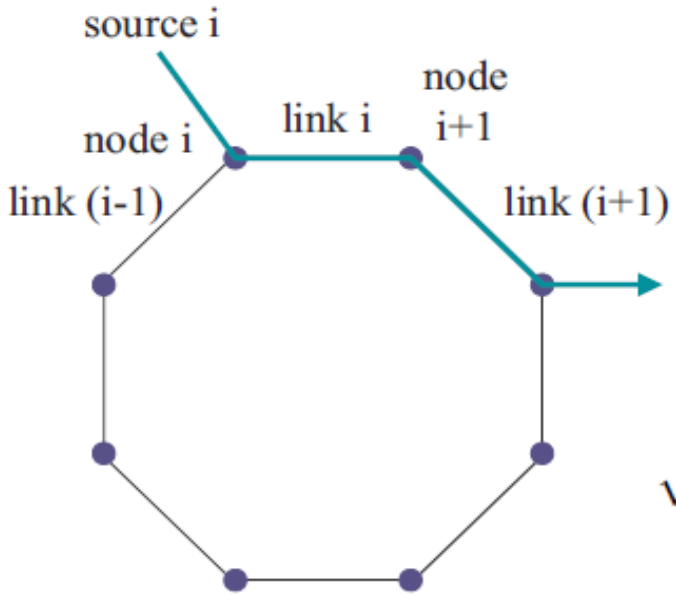
Буфер у X будет переполняться!
Интенсивность трафиков не соответствует распределению пропускных способностей каналов .

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

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



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

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

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

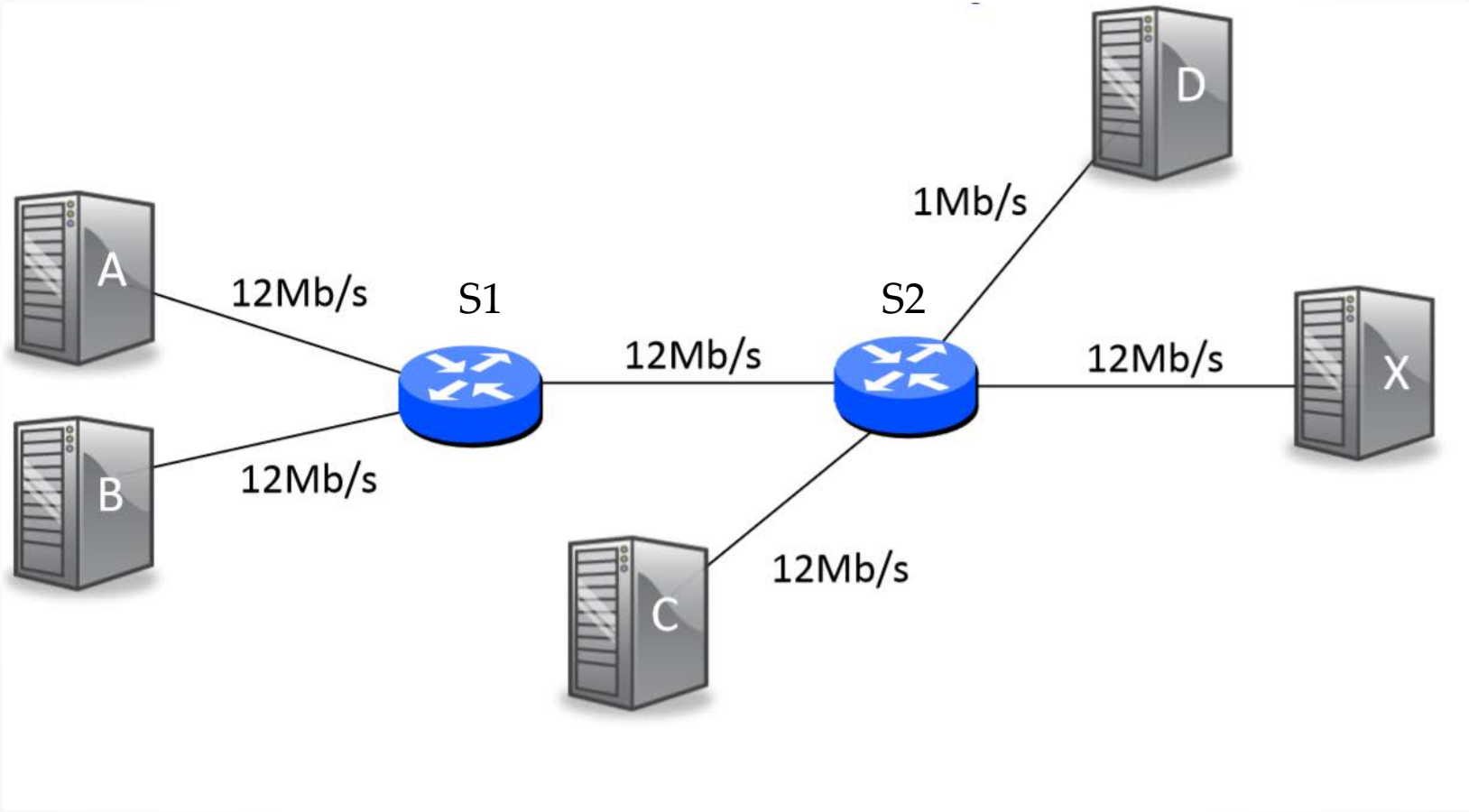
при $u \to 0$

$$\lambda'' = \frac{c^2}{\lambda} + o\left(\frac{1}{\lambda}\right)$$

При λ -> ∞, λ'' -> 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)

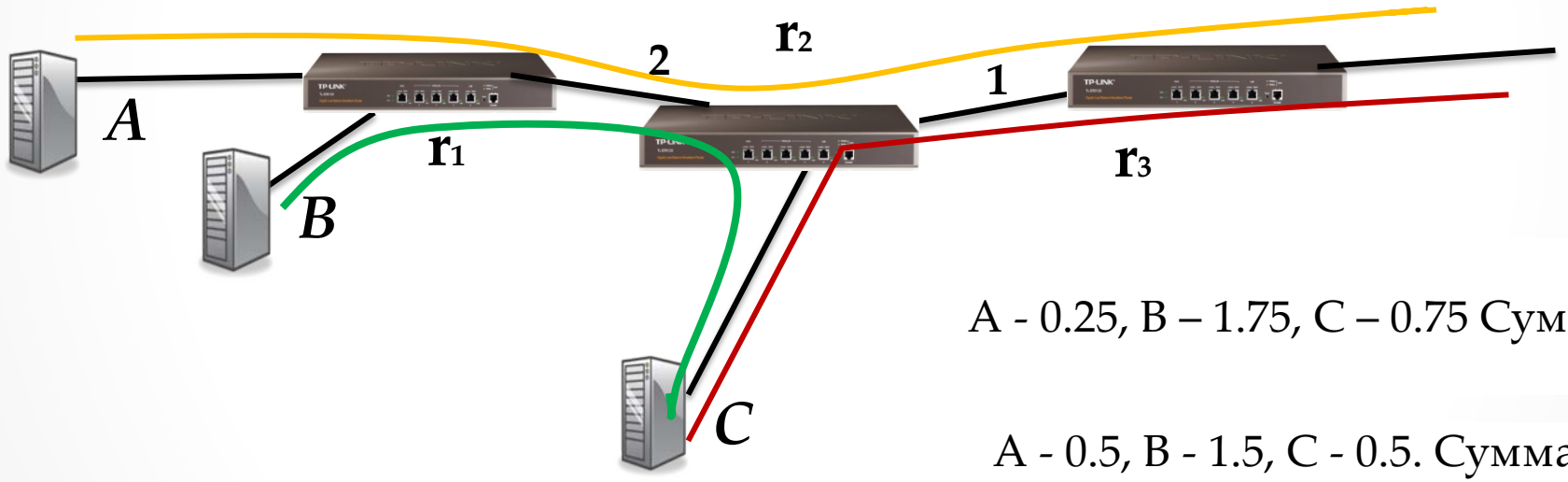
- Коммутацию пакетов используют потому, что она позволяет эффективно использовать пропускную способность каналов. Поэтому буферы в маршрутизаторах часто заполнены.
- Если буферы пусты, задержки малы, но интенсивность использования сети низкая.
- Если буферы постоянно заполнены, задержки возрастают, но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

Перегрузки неизбежны и даже желательны!

Промежуточный итог

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

Понятие справедливости



A - 0.25, B – 1.75, C – 0.75 Суммарно = 2.75

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

Здесь и A и C на $r_2 r_3$ по 0.5

Что такое справедливость и как ее измерить ?

Max-min справедливость

Определение:

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

A - 0.25, B – 1.75, C – 0.75
Суммарно = 2.75

A - 0.5, B - 1.5, C - 0.5. Суммарно - 2.5
Здесь и A и C на r2r3 по 0.5

Необходимое и достаточное условие min-max справедливости

Пусть есть:

- $s = 1, \ldots, S$ - *множество источников*
- $l = 1, \ldots, L$ - *множество линий*
- $A_{l,s}$ – *доля потока от s на линии l*
- $\{c_i\}_{i=1}^{L}$ - пропускная способность c_i

Модель сети $M = (\boldsymbol{x}, A)$, где

$\boldsymbol{x}$ – распределение потоков от s (вектор),

A – матрица S*L

Распределение потоков допустимо

- $\forall s: 1 \le s \le S \Rightarrow x_s \ge 0$
- $\forall l: 1 \div L: \ \sum_{i=1}^{S} A_{l,i}\, x_i \le c_l$

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

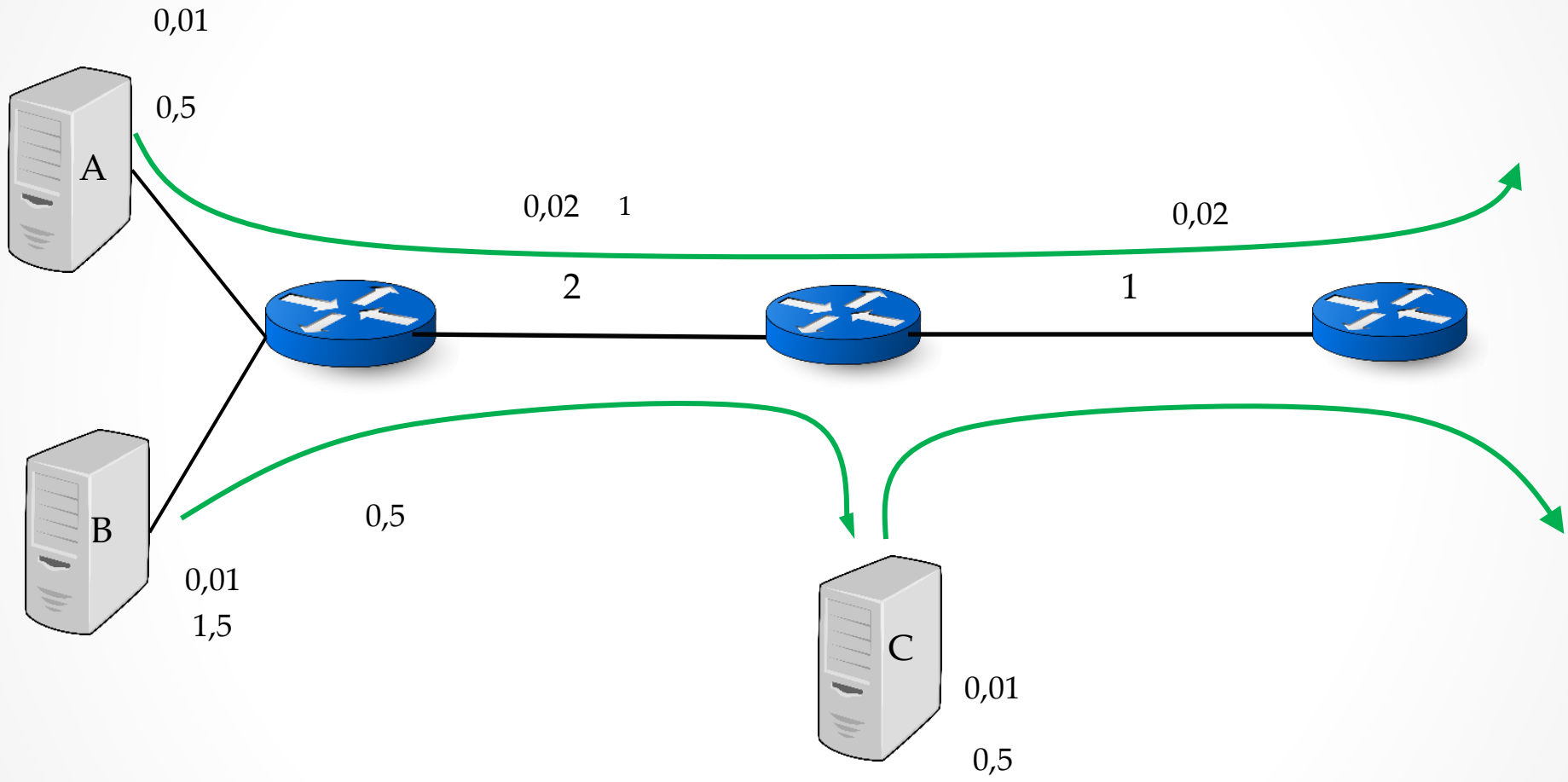
Линия l – ***критична*** *для s тогда и только тогда*

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

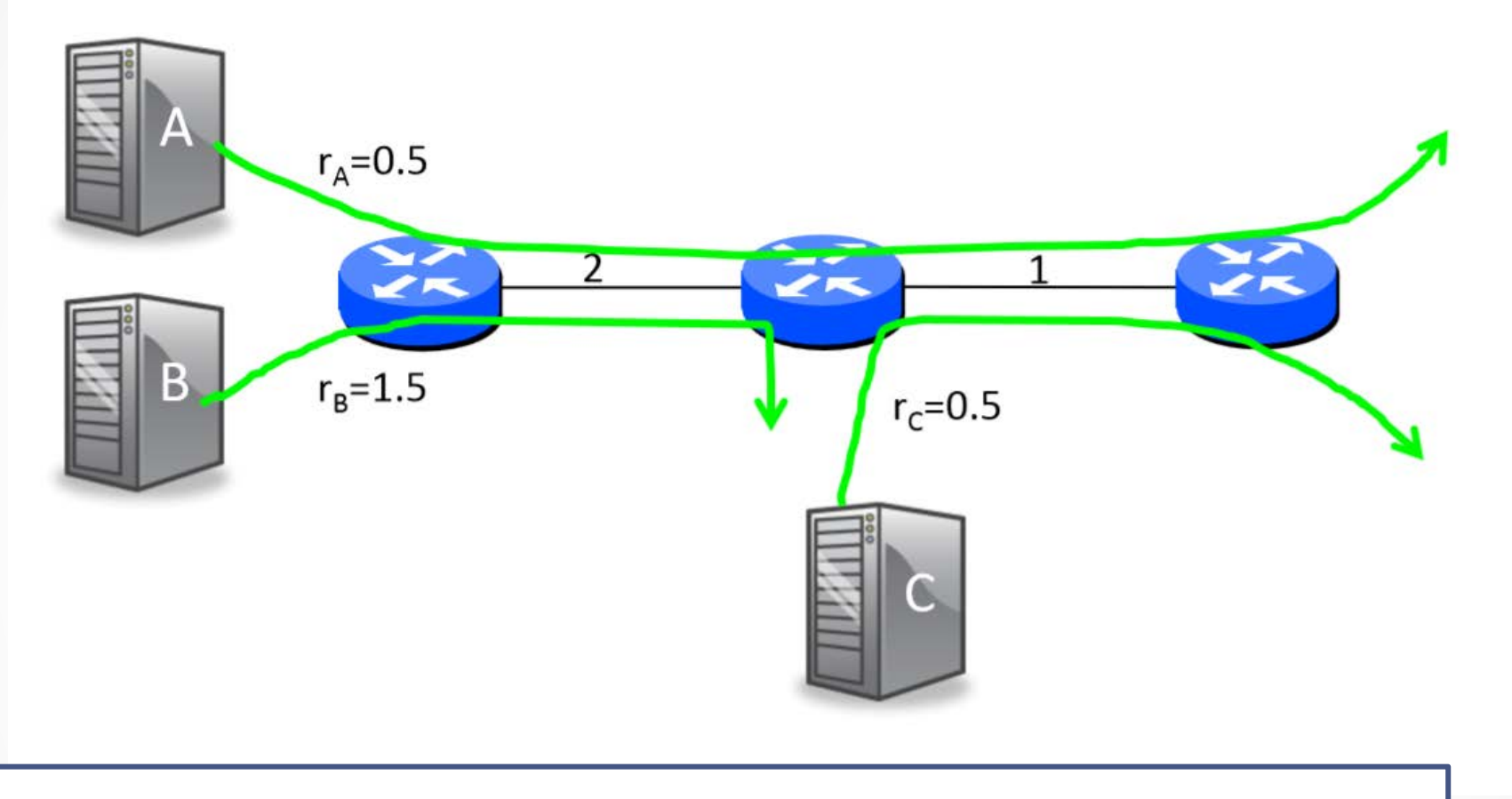
Определение: допустимое распределение потоков является max-min распределением если существует такой источник потока, что любое другое допустимое распределение $\boldsymbol{y}$ *такое , что* $y_s > x_s$, *то* $\exists\, s': x_{s'} \le x_s$ *и* $y_{s'} < x_{s'}$

Теорема 1: $\boldsymbol{x}$ *есть max-min распределение потоков тогда и только тогда когда у каждого источника есть критичная линия*

Алгоритм построения max-min распределения (постепенного заполнения)



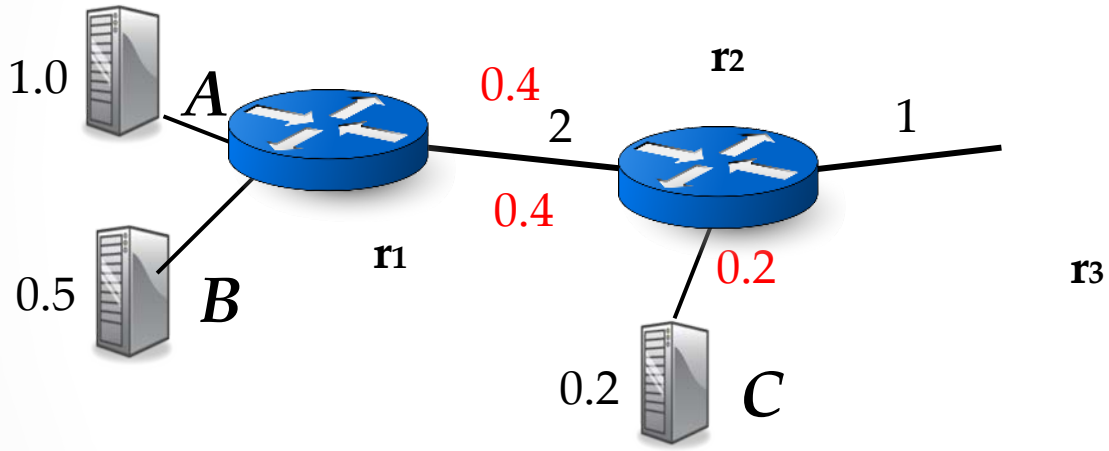
Max-min справедливое распределение



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

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

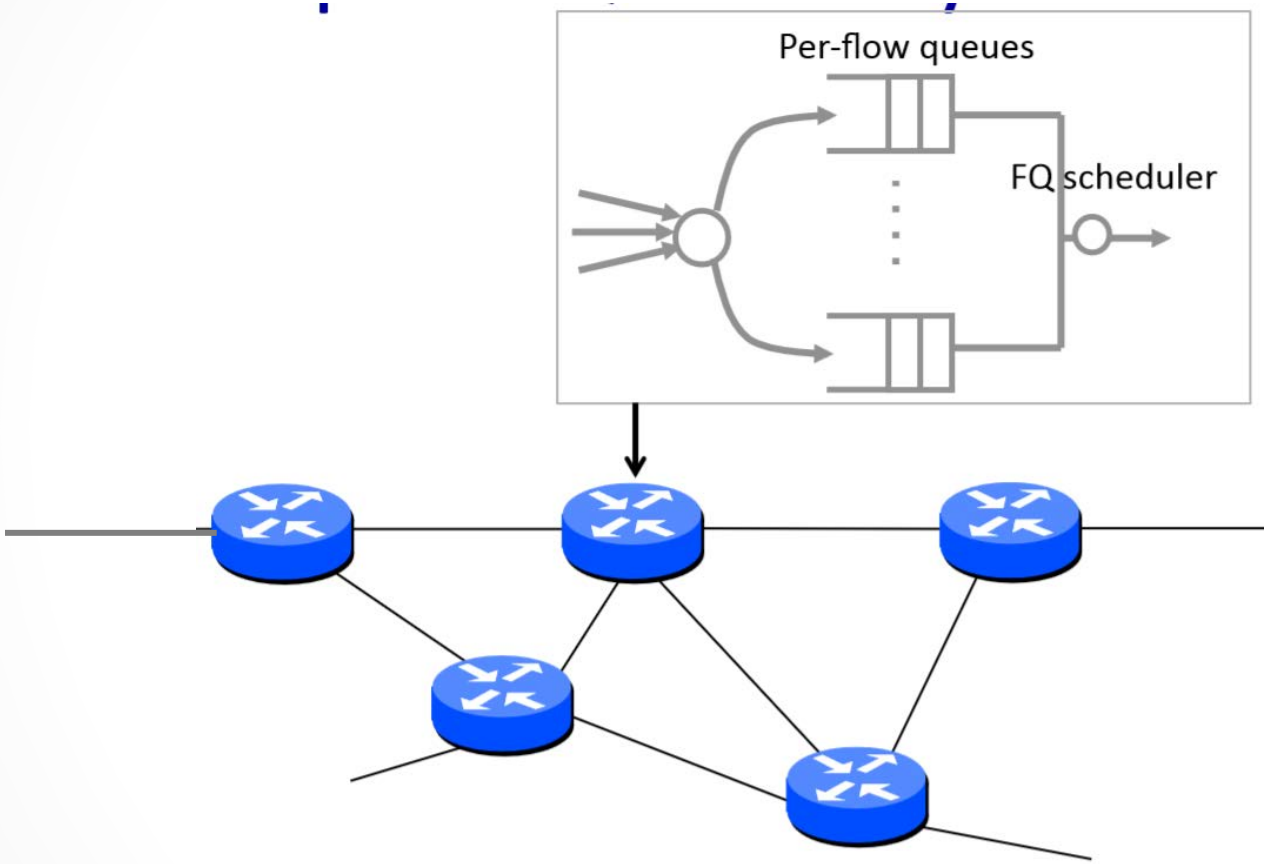
План

Мы хотим построить алгоритм управления перегрузками такой, чтобы:

1. Высокая пропускная способность: каналы загружены, скорость потоков высокая
2. Распределение пропускной способности каналов в соответствии с max-min справедливостью
3. Быстрая реакция на изменения состояния сети
4. Распределенное управление

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

Пример: FQ на каждом маршрутизаторе



Управление перегрузками в TCP

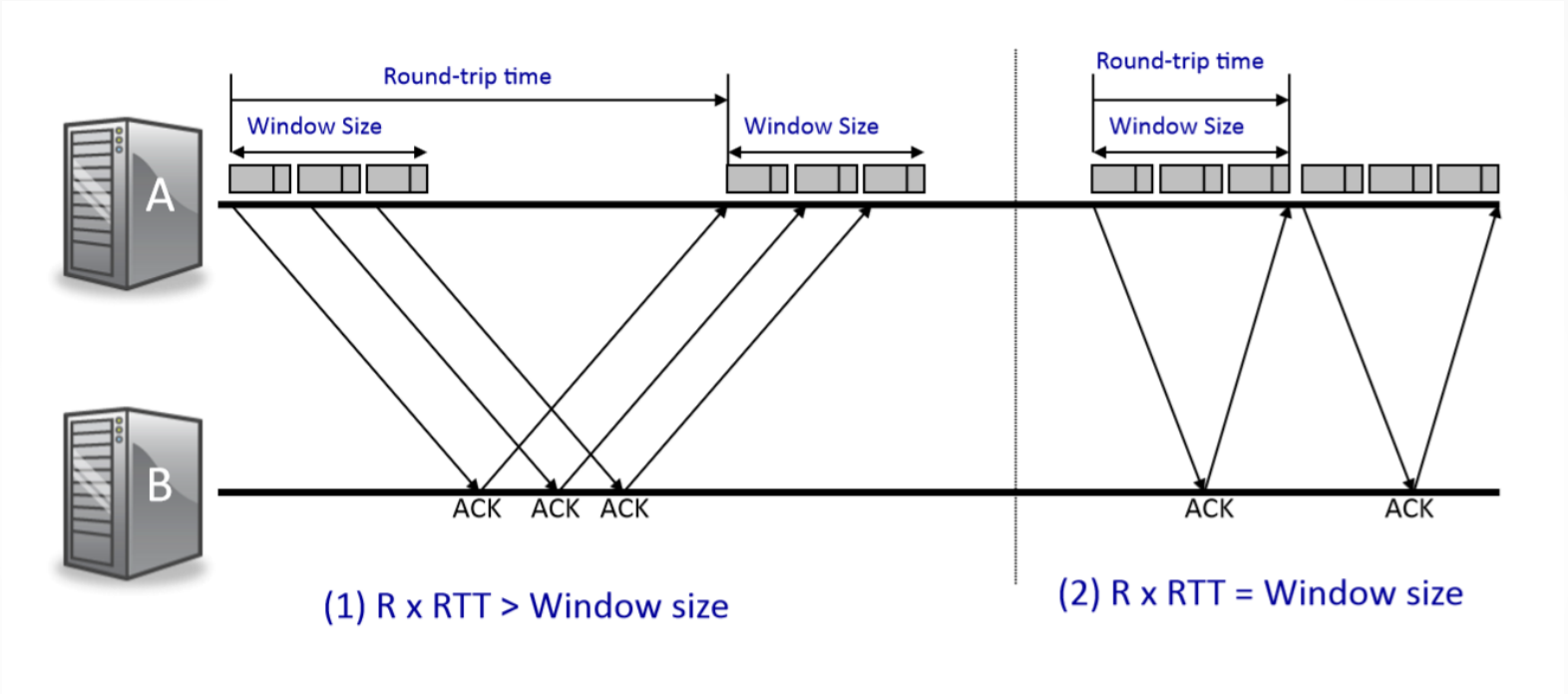
(управление на хостах)

В TCP управление перегрузкой размещается на конечном хосте

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

Скользящее окно

Скользящее окно в TCP



Скользящее окно перегрузки в TCP

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

размер окна=min{ Объявленное окно, Окно перегрузки }

получатель — отправитель (cwnd)

Как определить размер cwnd?

AIMD управление перегрузкой для одного потока

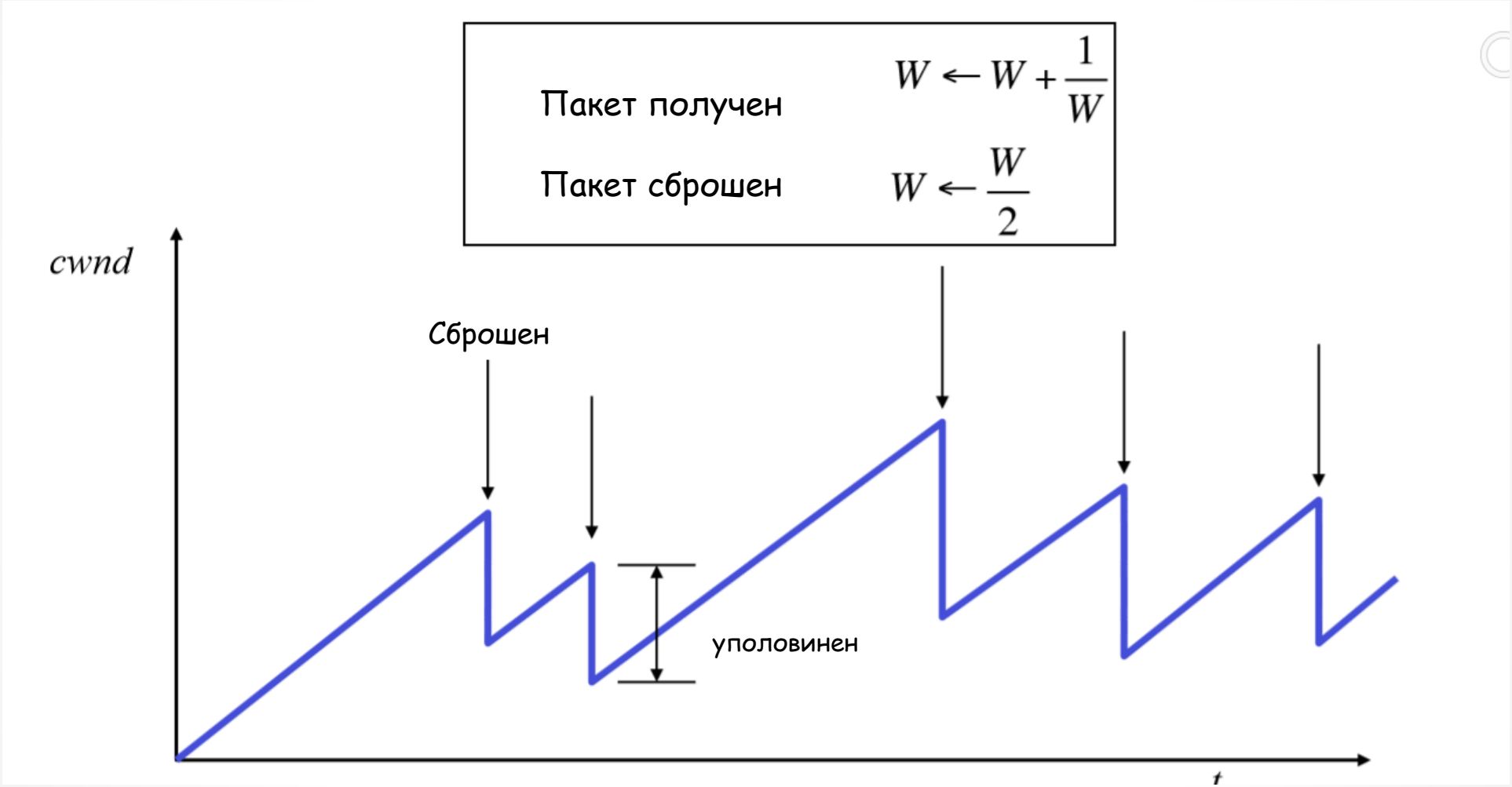
Введение в компьютерные сети

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

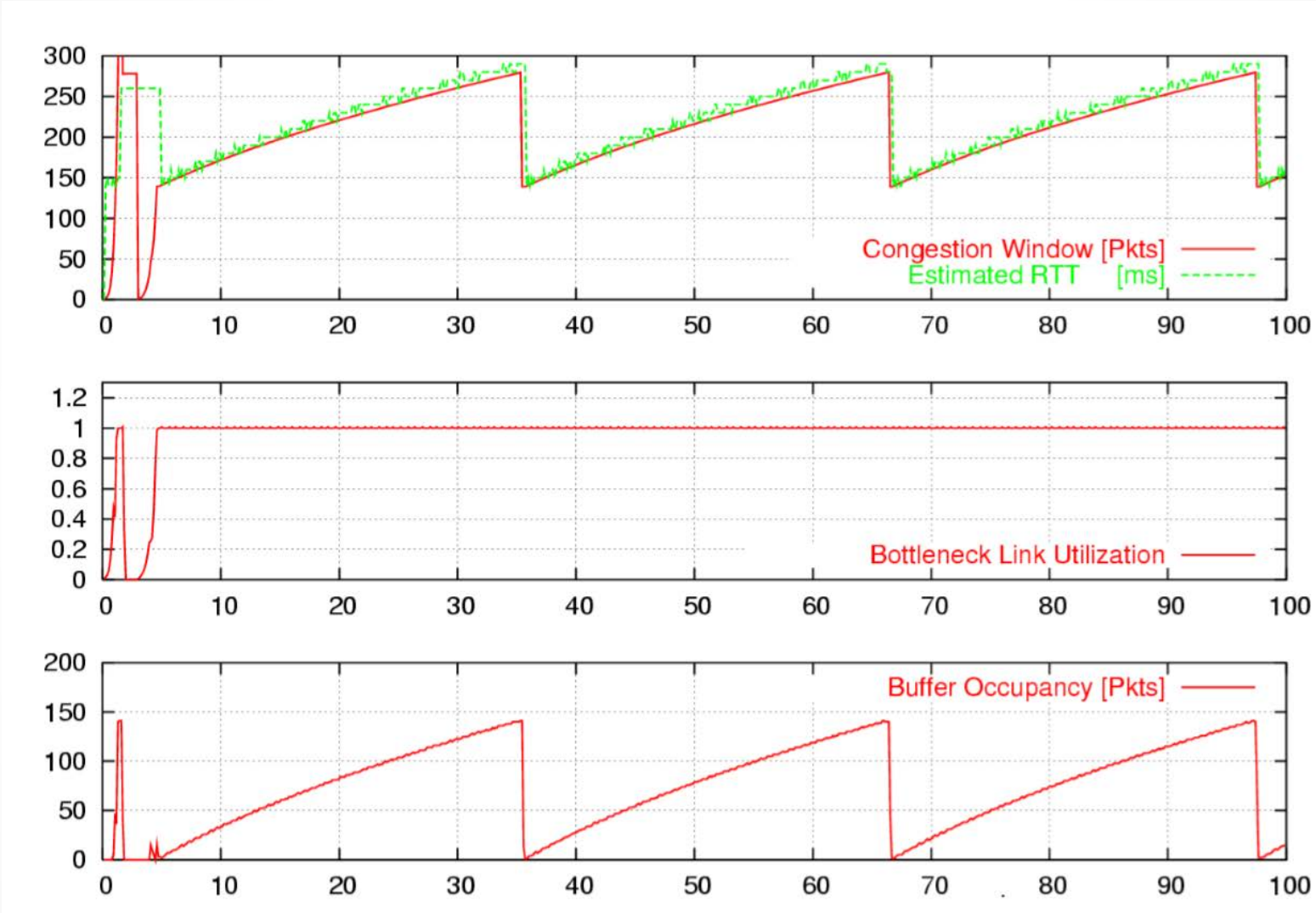
AIMD
(Additive Increase Multiple Decrease)

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

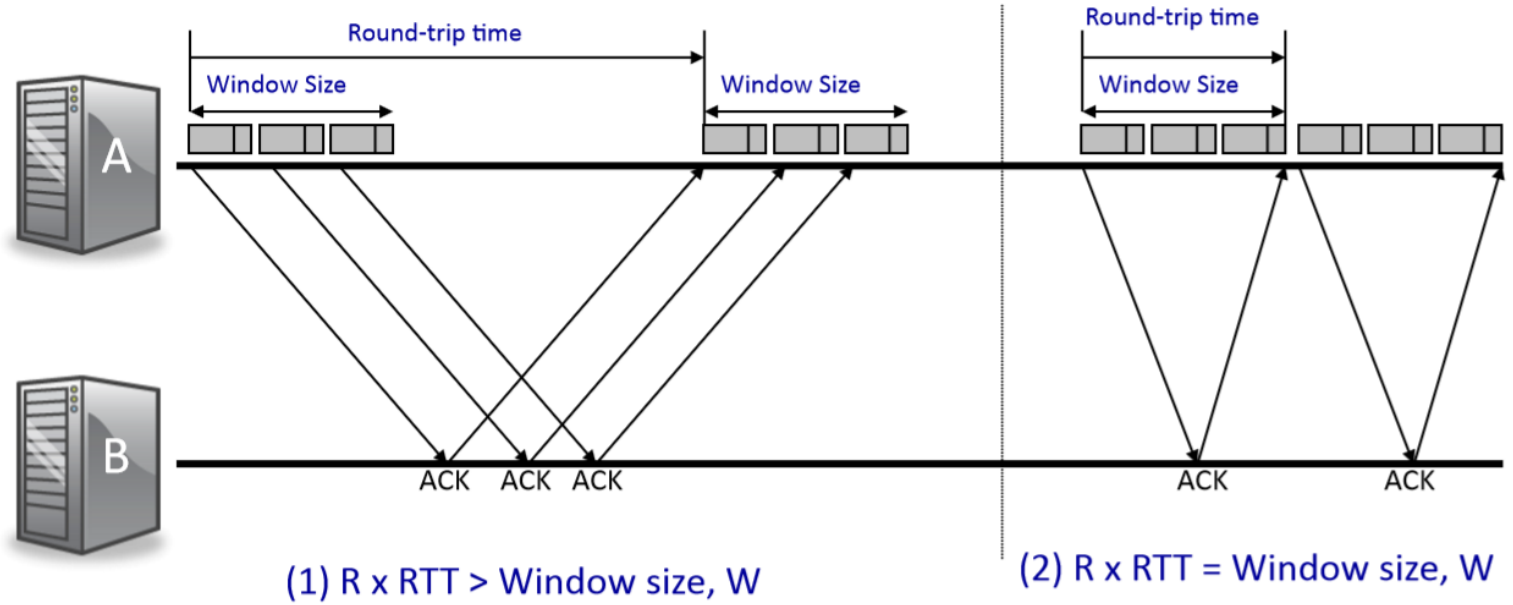
Пила AIMD


Пакет получен
$W \leftarrow W + \frac{1}{W}$
Пакет сброшен
$W \leftarrow \frac{W}{2}$
cwnd
Сброшен
уполовинен
t

Примеры динамики одиночного потока

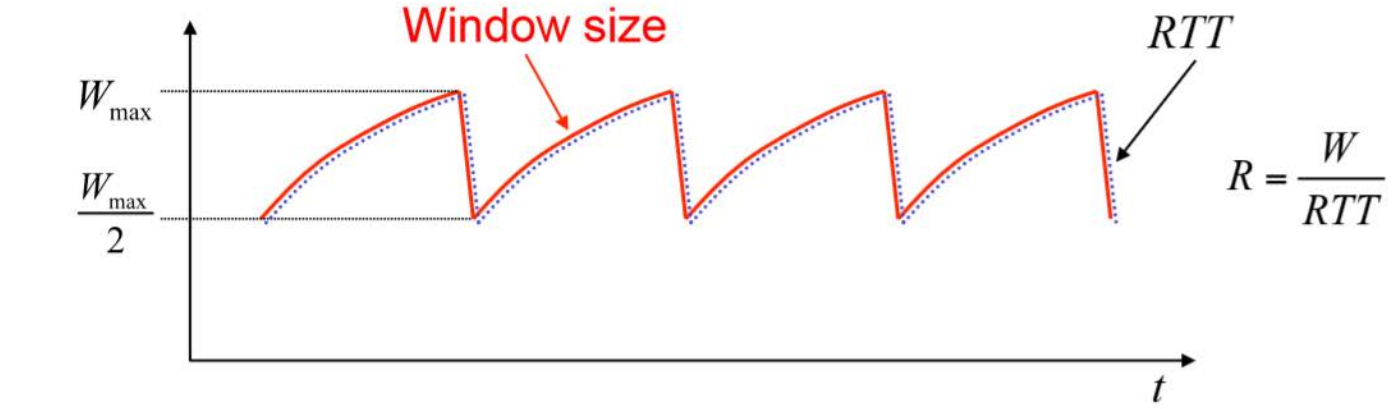


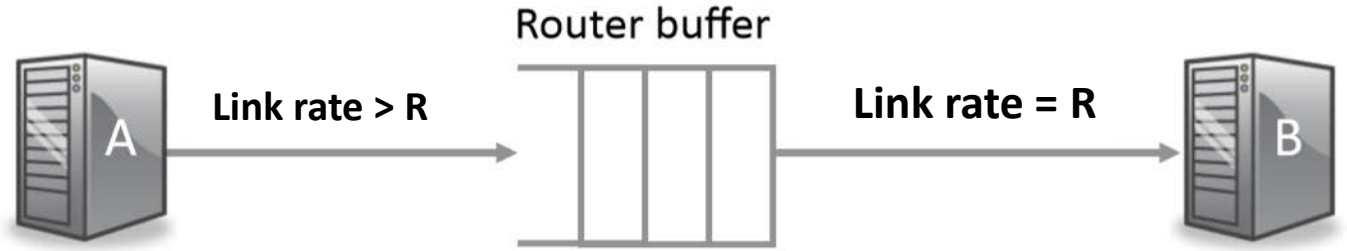
Скорость отправки для одиночного потока



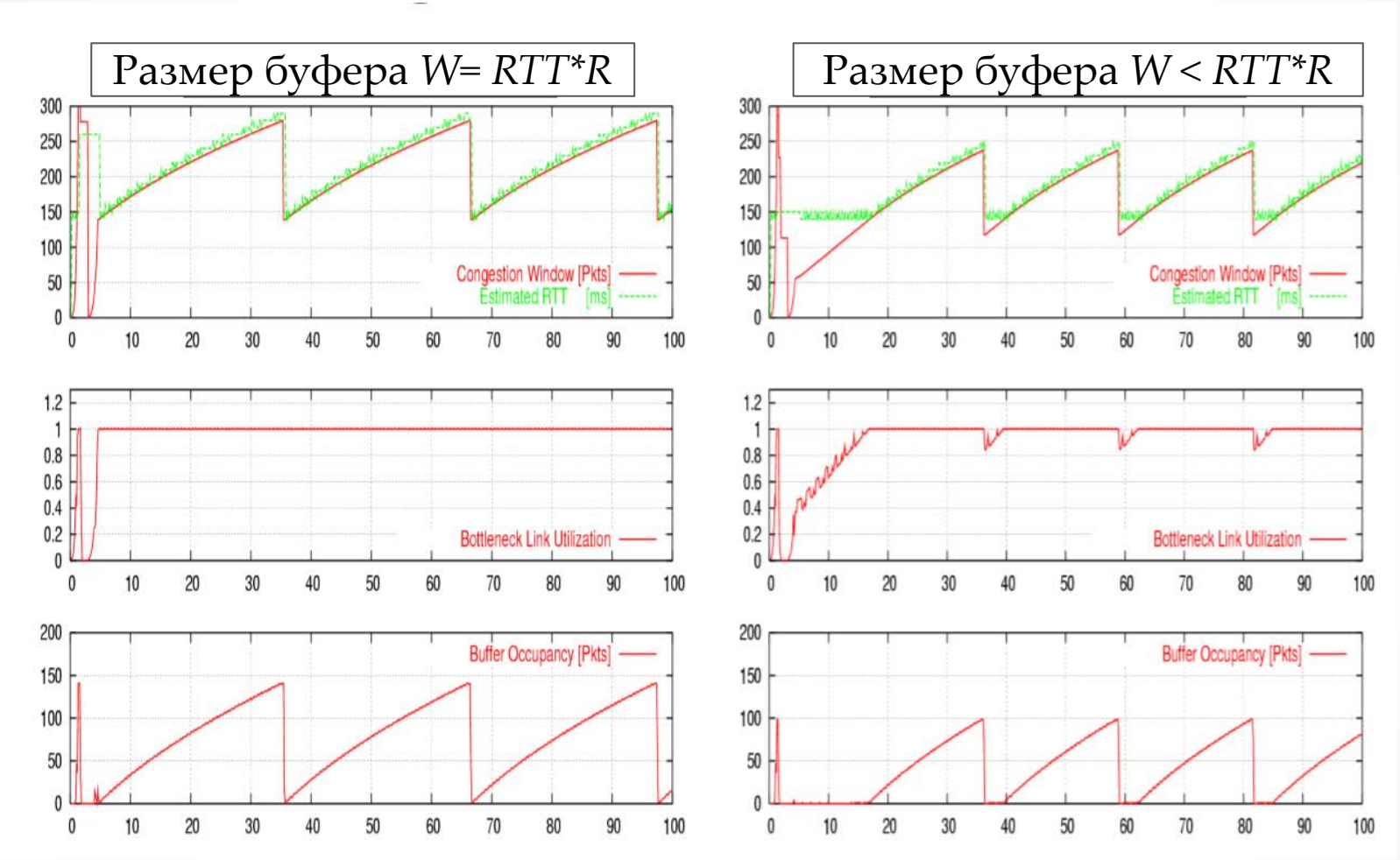
$$R = \frac{W}{RTT}$$

Скорость отправки для одиночного потока





Насколько большим должен быть буфер?



Промежуточные выводы

1. В TCP управление перегрузкой располагается на хосте

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

Комментарии для одиночного потока

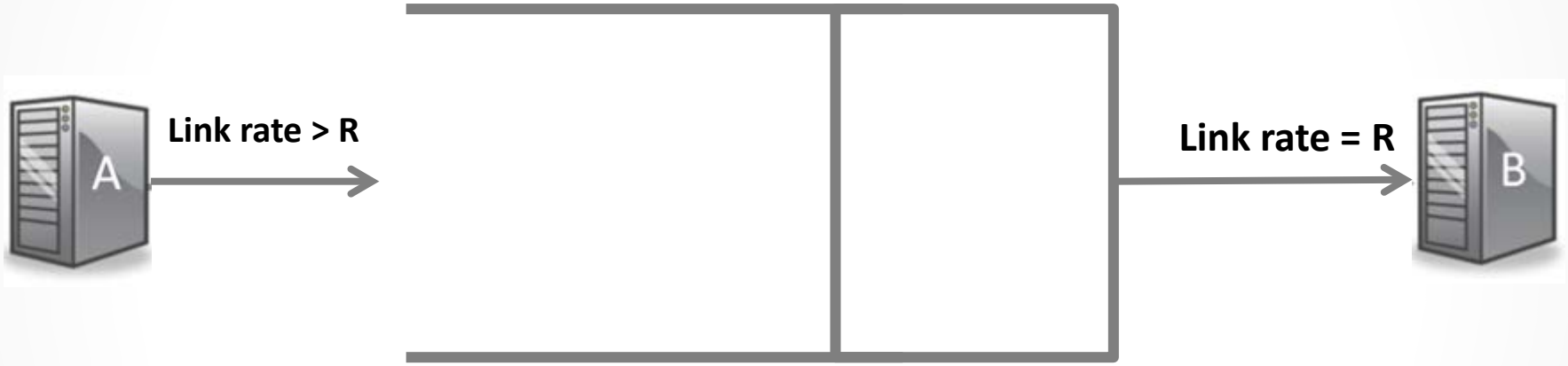
1. Окно увеличивают, сокращают в соответствии с AIMD
2. … пробировать как много байт канал еще может вместить
3. Пилообразное поведение – нормальная форма динамики
4. Скорость отправки постоянная
5. Размер буферного пространства определяет соотношение – RTT x R

Управление перегрузкой: AIMD с несколькими потоками

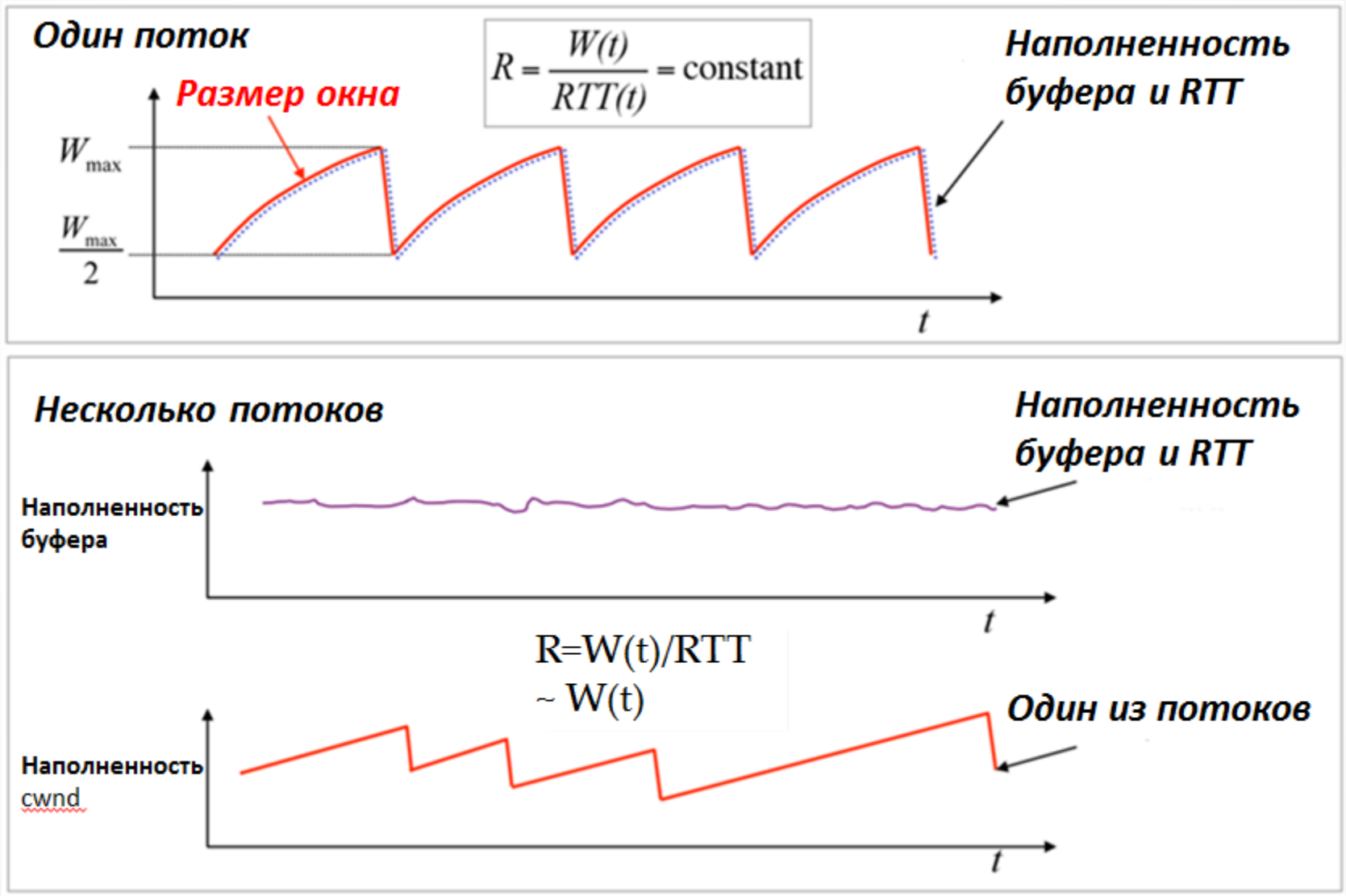
Введение в компьютерные сети

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

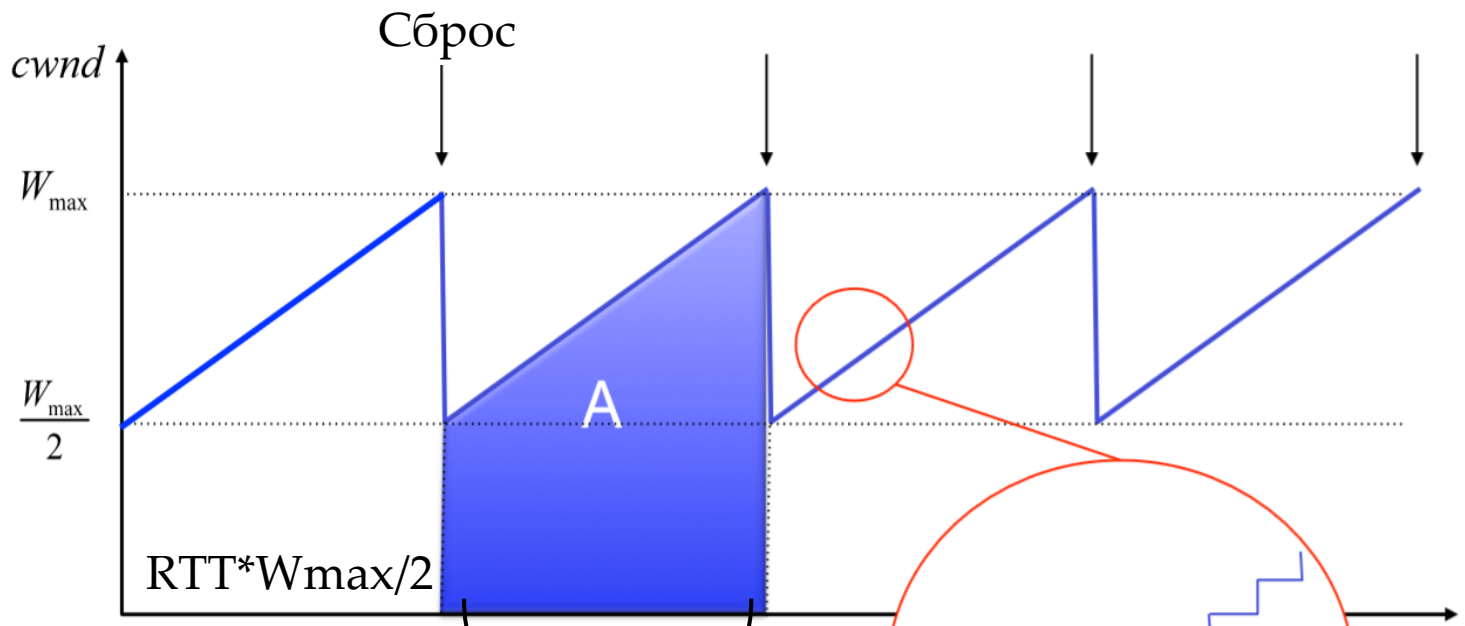
Буфер маршрутизатора



Один поток vs много потоков



Интуитивная геометрическая интерпретация



Вероятность сброса пакета

$p \approx 1/A$, где $A = \frac{3}{8} W^2$

Пропускная способность

$$R = \frac{A}{\left(\frac{W_{max}}{2}\right) RTT} = \sqrt{\frac{3}{2}} \frac{1}{RTT\sqrt{p}}$$

Интерпретация уравнения скорости

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

$$RTT \to 0 \Rightarrow R \to \infty$$

$$p \to 0 \Rightarrow R \to \infty$$

$$\frac{R1}{R2} = \sqrt[2]{\frac{p_2}{p_1}}$$

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 TCP

В TCP управление перегрузкой размещается на конечном хосте

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