Введение в Сетевое Исчисление

Доп. главы Компьютерных сетей и телекоммуникации

к.ф.-м.н. Чемерицкий Е.В.

План лекции

- Основные термины и определения
- Оценка задержки передачи данных
- Алгоритм Separated Flow Analysis
- Примеры задач вычисления задержки
- Достижимые оценки для задержки с помощью линейного программирования

Литература

Le Boudec J.-Y. and Thiran P.

Network Calculus: A Theory of Deterministic Queuing Systems for the Internet

Fidler M.

Survey of deterministic and stochastic service curve models in the network calculus

Сетевое Исчисление

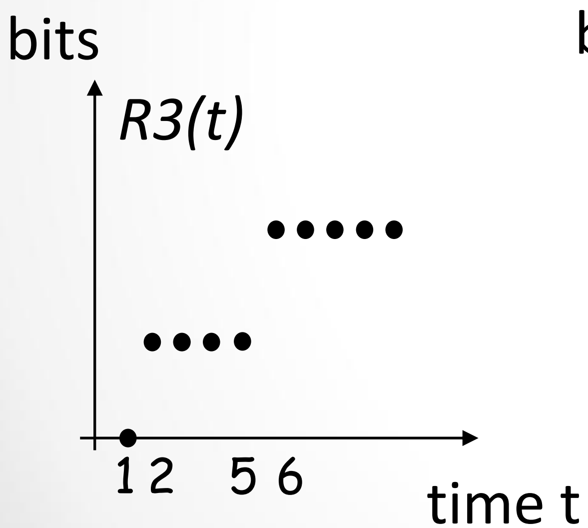
- Queueing Theory, Agner Krarup Erlang, 1909
- Queueing Networks, James R. Jackson, 1957
  - Предсказывает отставание и задержку для системы из обработчиков и буферов
  - Вероятностная модель – не годится для анализа систем реального времени
- Scheduling Theory, Liu & Layland, 1972
  - Оценка худшего случая (***worst-case analysis***)
- Network Calculus, Rene Cruz, 1991
  - Расширение теории расписаний до системы из обработчиков и буферов

Компоненты модели

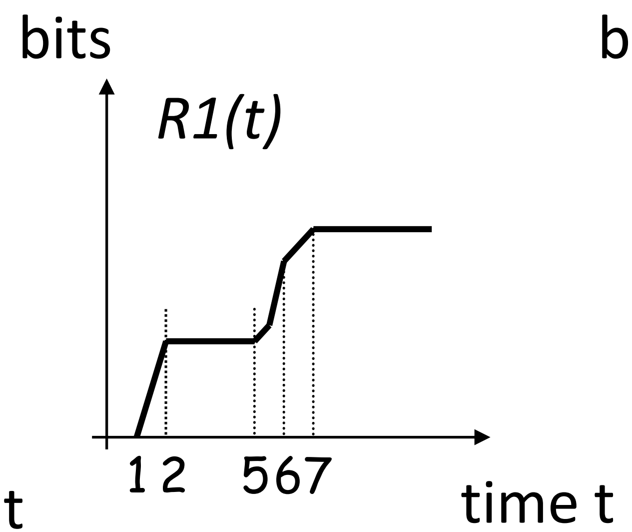
- Представление в виде ***сети обработчиков***
  – Обработчик – логически целостный компонент, выполняющий преобразования потоков данных
- Описание нагрузки – множества потоков данных, поступающих в систему
  – Маршрут передачи данных
  – Характеристики интенсивности
- Описание обработчиков
  – Характеристики производительности
  – Принципы мультиплексирования

Накопительные функции

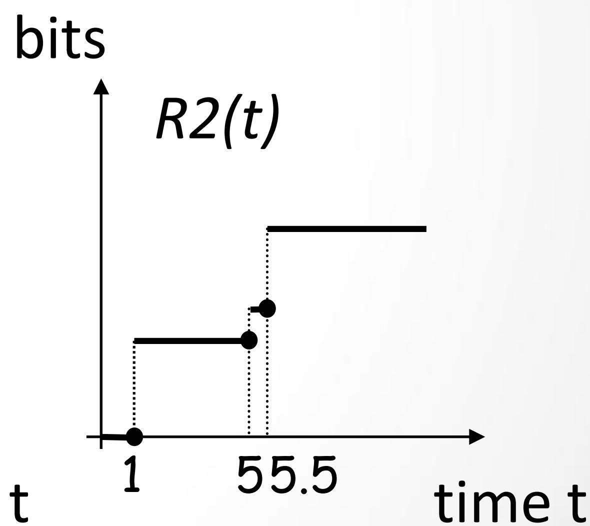
- Зависимость количества переданных данных от времени



Дисретная модель

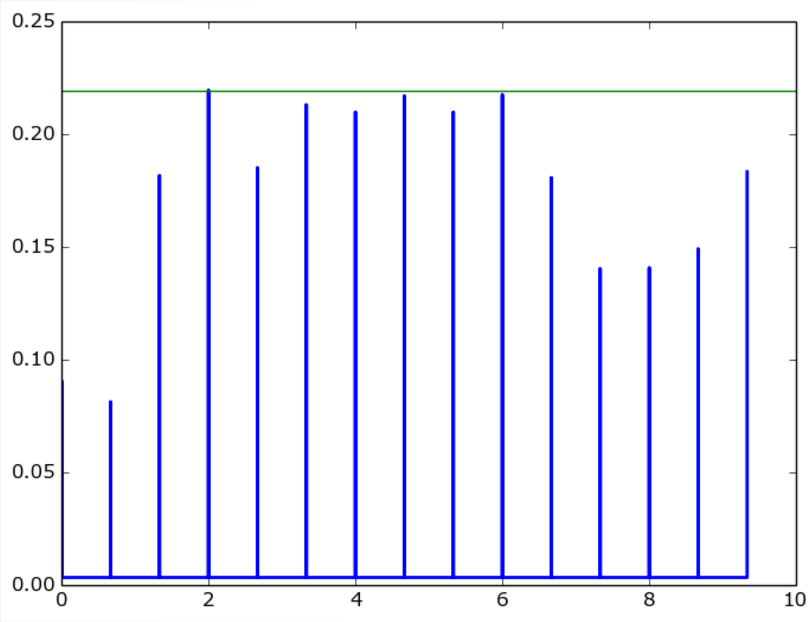


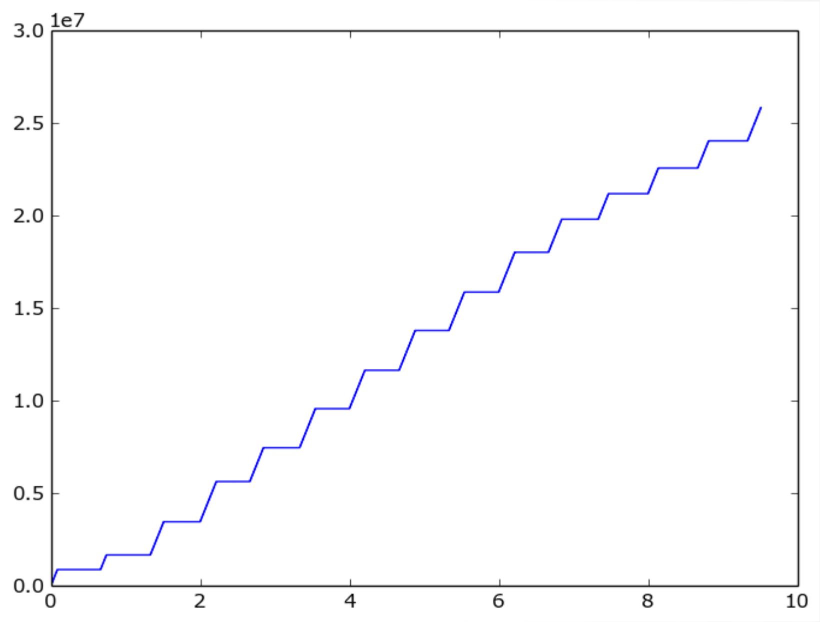
Жидкостная модель



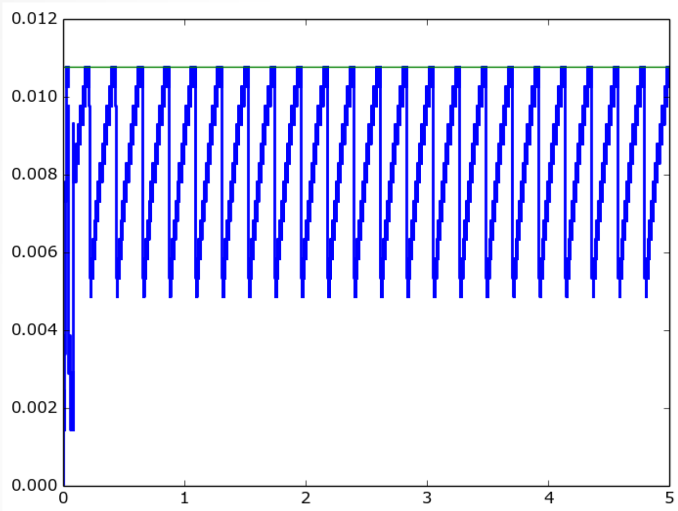
Модель без ограничений

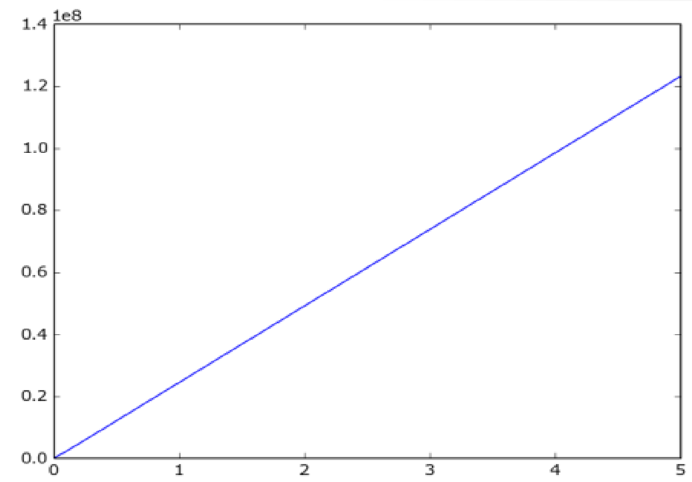
Пример: видео в MPEG

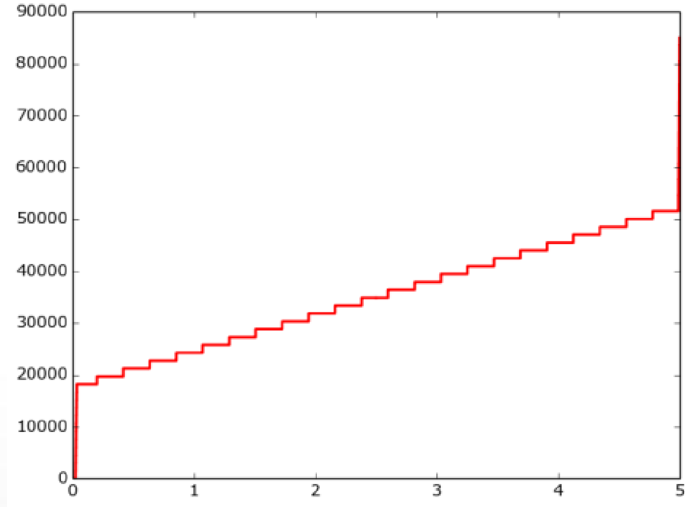




Пример: TCP трафик



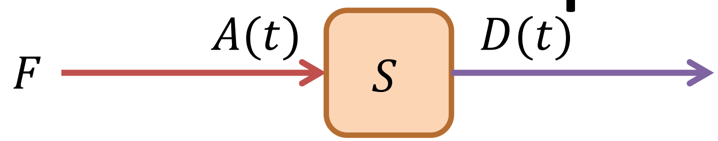




Основные определения

- ***Функция прибытия*** $A(t)$ описывает зависимость суммарного количества данных, поступивших на обработчик от времени
- ***Функция отправки*** $D(t)$ – зависимость количества переданных данных потока от времени
- Каждый обработчик может быть описан перечислением пар вида $\langle A(t), D(t) \rangle$
- ***Отставание (backlog)*** $b(t)$ – выражает количество данных, находящихся *внутри* обработчика
- ***Период отставания*** – промежуток в течение которого функция отставания строго положительна
- ***Задержка (delay)*** $d(t)$ – время прохождения через обработчик той порции данных, которая поступила на него в момент времени t

Функции поступления и отправки



$\mathcal{F}$ - множество накопительных функций

$\mathcal{F} = \{f: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \cup \{+\infty\} | \Psi_1 \wedge \Psi_2 \wedge \Psi_3\}$

$A(t) \in \mathcal{F}$ – функция поступления

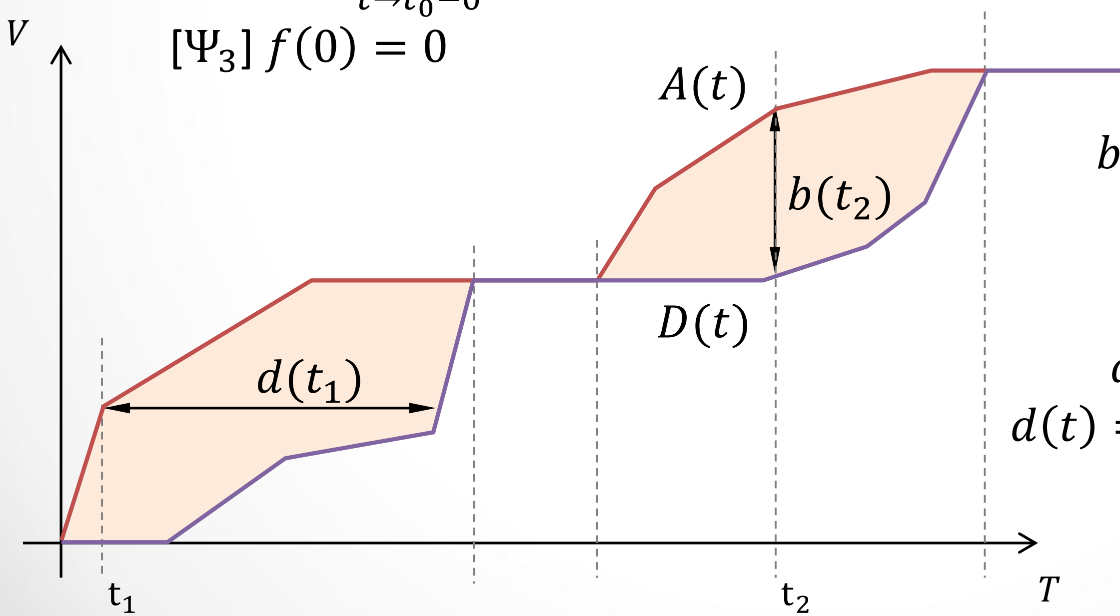
$D(t) \in \mathcal{F}$ – функция отправки

$[\Psi_1]\ \forall t_1 \leq t_2: f(t_1) \leq f(t_2)$

$[\Psi_2]\ \forall t_0: \lim\limits_{t \to t_0 - 0} f(t) = f(t_0)$

$[\Psi_3]\ f(0) = 0$

$$\forall \mathrm{t}:\ \mathrm{D(t)} \geq \inf_{0 \leq \mathrm{s} \leq \mathrm{t}} \{\mathrm{A(s)} + \beta(\mathrm{t} - \mathrm{s})\}$$



$b: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - отставание

$b(t) = A(t) - D(t)$

$b(t) \leq v(A, D)$

$d: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - задержка

$d(t) = \inf\{\tau \geq 0\ | A(t) \leq D(t + \tau)\}$

$d(t) \leq h(A, D)$

Кривые нагрузки и сервиса

- Вид функций поступления и отправки в большинстве случаев неизвестен
  - Необходима аппроксимация
- ***Кривая нагрузки*** $\alpha(t)$ -- накопительная функция, для которой выполнено условие

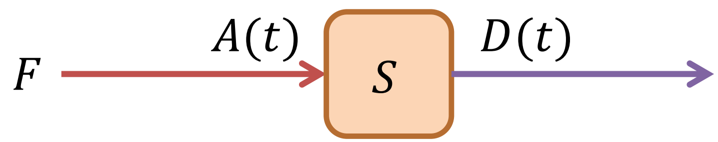
$$\forall t, \tau: A(t+\tau) - A(t) \leq \alpha(\tau)$$

  - за время τ поступает не больше $\alpha(\tau)$ данных
- ***Кривая сервиса*** $\beta(t)$ – накопительная функция, для которой внутри каждого периода оставание выполняется условие

$$\forall t, \tau: D(t+\tau) - D(t) \geq \beta(\tau)$$

  - за время τ обслуживается не меньше $\beta(\tau)$ данных

Кривые нагрузки и сервиса



$\mathcal{F}$ - множество накопительных функций

$\mathcal{F} = \{f: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \cup \{+\infty\} | \Psi_1 \wedge \Psi_2 \wedge \Psi_3\}$

$[\Psi_1]\ \forall t_1 \leq t_2: f(t_1) \leq f(t_2)$

$[\Psi_2]\ \forall t_0: \lim\limits_{t \to t_0 - 0} f(t) = f(t_0)$

$[\Psi_3]\ f(0) = 0$

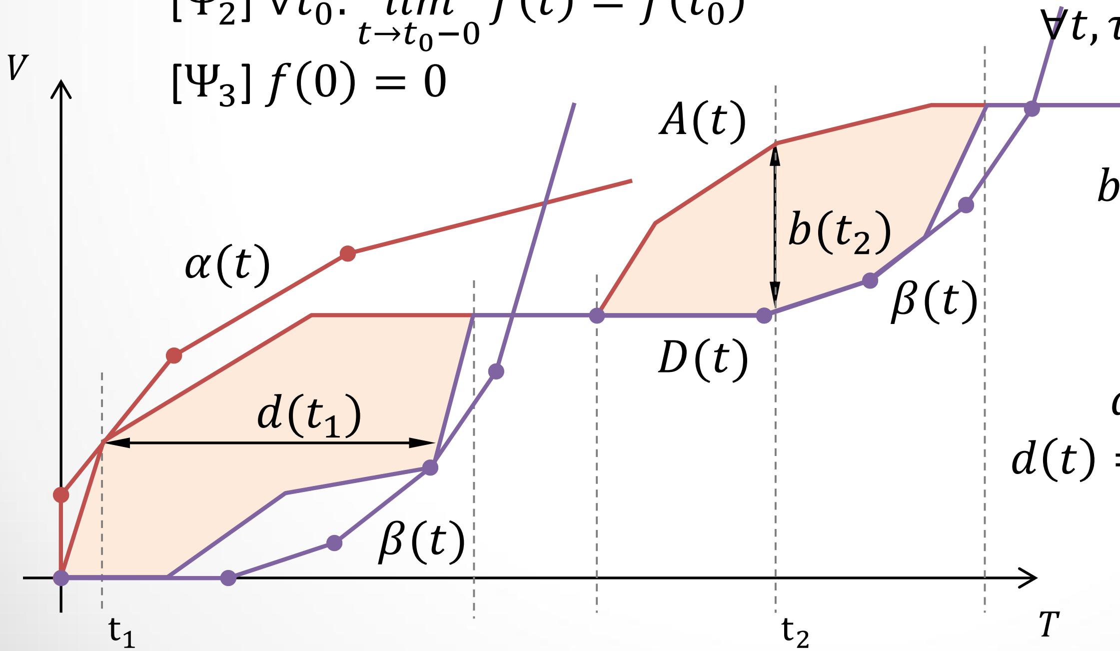
$\alpha(t) \in \mathcal{F}$ – кривая нагрузки

$\forall t, \tau: A(t+\tau) - A(t) \leq \alpha(\tau)$

$\beta(t) \in \mathcal{F}$ – кривая сервиса

внутри периода отставания:

$\forall t, \tau: D(t+\tau) - D(t) \geq \beta(\tau)$



$b: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - отставание

$b(t) = A(t) - D(t)$

$b(t) \leq v(A, D)$

$d: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - задержка

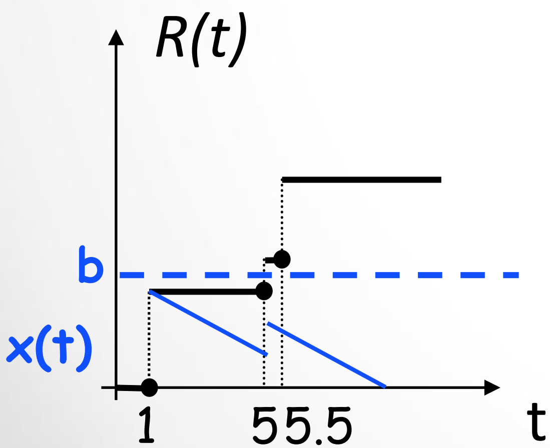
$d(t) = \inf\{\tau \geq 0\ | A(t) \leq D(t+\tau)\}$

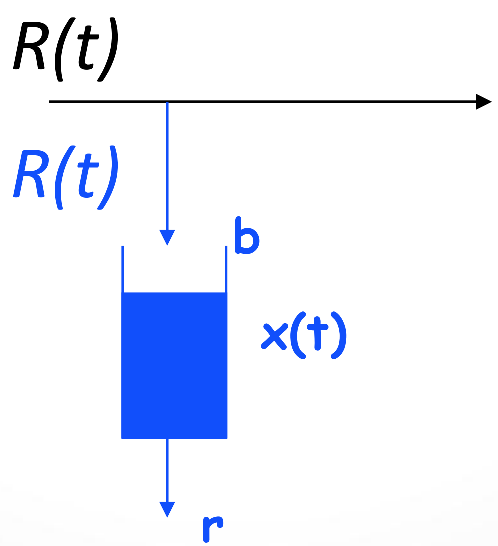
$d(t) \leq h(A, D)$

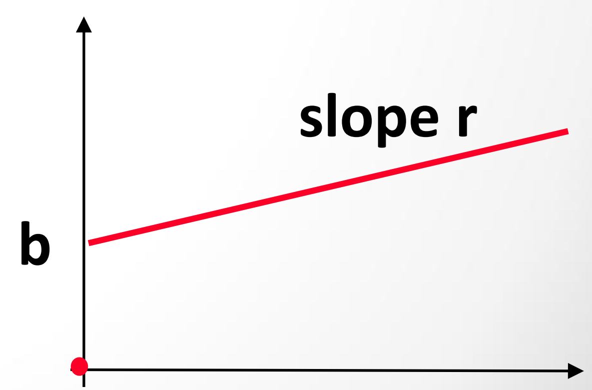
Алгоритм текущего ведра

Кривые нагрузки определяются профилями трафика после shaping’а & policing’а

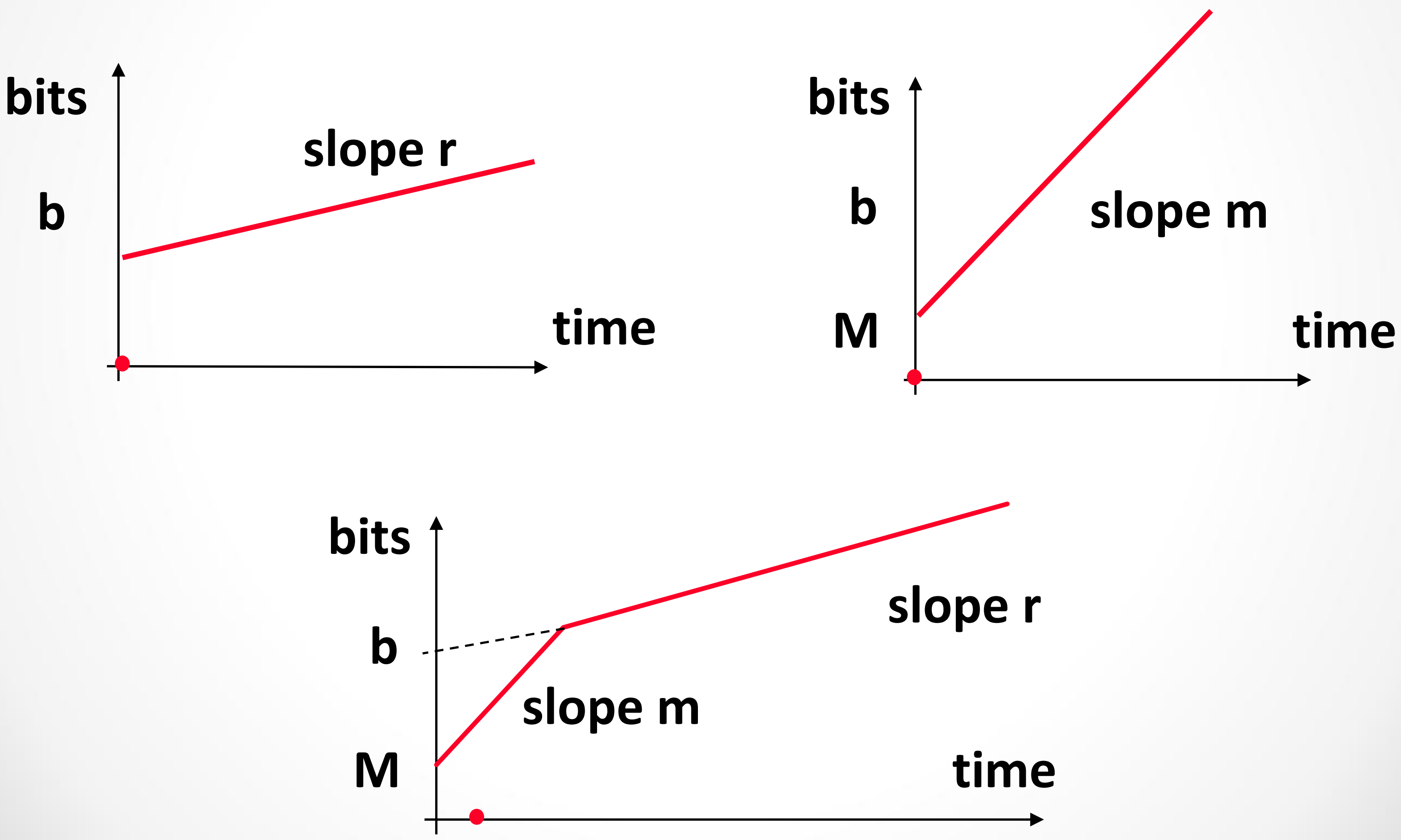
Кривые ограничения (envelop curve)







Комбинирование нескольких шейперов



Базовые результаты сетевого исчисления

Теорема 1 (оценка отставания). Пусть поток данных с кривой нагрузки $\alpha \in \mathcal{F}$, обслуживается обработчиком с кривой сервиса β.

Тогда значение отставание обработчика не превышает вертикального отклонения между кривыми прибытия α и сервиса β*:*

$$\forall t \in \mathbb{R}\colon\ b(t) \leq v(\alpha, \beta)$$

$$v(\alpha, \beta) = \sup_{t \geq s \geq 0} \{\alpha(s) - \beta(s)\}$$

Базовые результаты сетевого исчисления

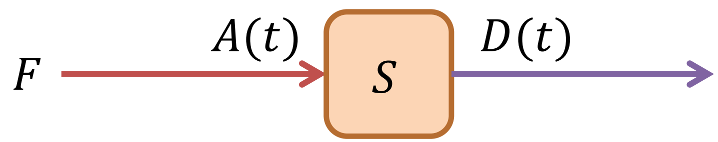
Теорема 2 (оценка задержки). Пусть поток с кривой нагрузки $\alpha \in \mathcal{F}$, обслуживается обработчиком с кривой сервиса $\beta \in \mathcal{F}$ по дисциплине FIFO. Тогда $\forall t \in \mathbb{R}$: $d(t) \le h(\alpha, \beta)$

$$h(\alpha, \beta) = \sup_t \{\inf\{\tau \ge 0 \mid \alpha(t) \le \beta(t+\tau)\}\}$$

Если же дисциплина обслуживания неизвестна, то для задержки $d(t)$ справедлива оценка:

$$d(t) \le \inf\{\tau \ge 0 | \alpha(\tau) \le \beta(\tau)\}$$

Кривые нагрузки и сервиса



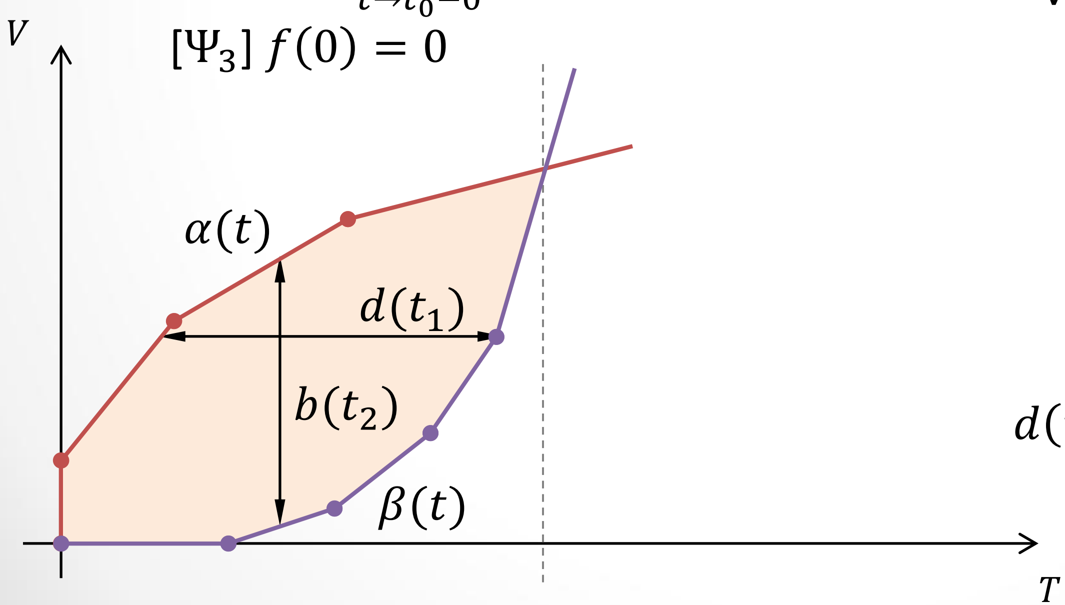
$\mathcal{F}$ - множество накопительных функций

$\mathcal{F} = \{f: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \cup \{+\infty\} | \Psi_1 \wedge \Psi_2 \wedge \Psi_3\}$

$[\Psi_1]\ \forall t_1 \leq t_2: f(t_1) \leq f(t_2)$

$[\Psi_2]\ \forall t_0: \lim\limits_{t \to t_0 - 0} f(t) = f(t_0)$

$[\Psi_3]\ f(0) = 0$



$\alpha(t) \in \mathcal{F}$ – кривая нагрузки

$\forall t, \tau: A(t+\tau) - A(t) \leq \alpha(\tau)$

$\beta(t) \in \mathcal{F}$ – кривая сервиса

внутри периода отставания:

$\forall t, \tau: D(t+\tau) - D(t) \geq \beta(\tau)$

$b: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - отставание

$$b(t) = \alpha(t) - \beta(t)$$
$$b(t) \leq v(\alpha, \beta)$$

$d: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ - задержка

$$d(t) = \inf\{\tau \geq 0\ |\alpha(t) \leq \beta(t+\tau)\}$$
$$d(t) \leq h(\alpha, \beta)$$

Базовые результаты сетевого исчисления

Теорема 3 (оценка выходного потока).

Если поток данных с кривой нагрузки $\alpha \in \mathcal{F}$, поступает на элемент с кривой сервиса $\beta \in \mathcal{F}$, то выходной поток, полученный в результате его обработки, ограничен кривой $\alpha' \in \mathcal{F}$:

$$\alpha'(s) = \sup_{\tau \geq 0}\{\alpha(s+\tau) - \beta(\tau)\}$$

Зная кривую прибытия и оценку выходного потока, можно получить оценку отставания и задержки

Магия свёрток

В линейной алгебре *свёрткой* функций f и g называется выражение:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{+\infty} f(s)g(t-s)ds$$

Если заменить операции суммы и умножения на операции поточечной минимизации и суммы, то свёрткой будет называться:

$$(f \otimes g)(t) \stackrel{\text{def}}{=} \inf_{0 \le s \le t} \{f(s) + g(t-s)\}$$

Алгебра Min-Plus

В min-plus алгебре операторы ***свёртки (convolution)*** $\otimes$ и ***обратной свёрткой (deconvolution)*** $\oslash$ функций f и g имеют следующий вид:

$$(f \otimes g)(t) \stackrel{\text{def}}{=} \inf_{0 \le s \le t} \{f(s) + g(t-s)\}$$

$$(w \oslash g)(t) \stackrel{\text{def}}{=} \sup_{t \ge u \ge 0} \{w(t+u) - g(u)\}$$

Преимущества алгебры Min-Plus

Кривые нагрузки и сервиса:

$$\forall t{:}\,\forall s \le t{:}\; A(t) - A(s) \le \alpha(t-s) \Leftrightarrow A = A \otimes \alpha$$

$$\forall t{:}\; D(t) \ge \inf_{0\le s\le t}\{A(s) + \beta(t-s)\} \Leftrightarrow D \ge A \otimes \beta$$

Оценки для оставания и задержки:

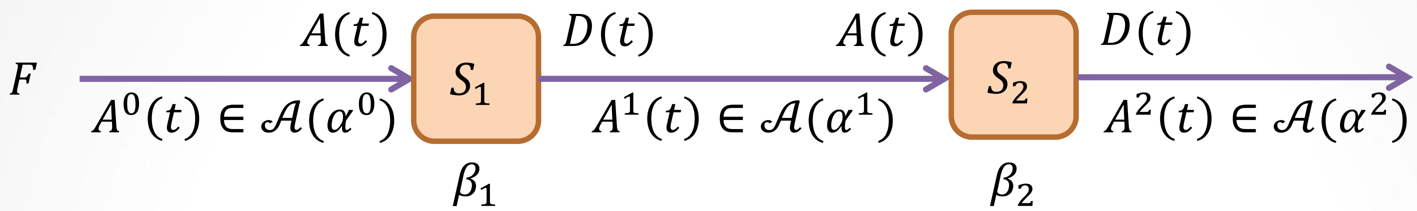
$$b(t) \le v(\alpha,\beta) = \sup_{s\ge 0}\{\alpha(s) - \beta(s)\} = (\alpha \oslash \beta)(0)$$

$$d(t) \le h(\alpha,\beta) = \sup_{t}\{\inf\{\tau \ge 0 \mid \alpha(t) \le \beta(t+\tau)\}\}$$

Оценка выходного потока:

$$\alpha'(t) = \sup_{u\ge 0}\{\,\alpha(t+u) - \beta(u)\} = (\alpha \oslash \beta)(t)$$

Композиция обработчиков



$\forall t{:}\ A^1(t) \geq \inf_{0 \leq s \leq t}\{A^0(s) + \beta_1(t-s)\}$

$\forall t{:}\ \alpha^1(t) = \sup_{\tau \geq 0}\{\alpha^0(t+\tau) - \beta_1(\tau)\}$

Множество функций $\mathcal{F}$ с операциями поточечной суммы и минимума образует алгебру $\langle \mathcal{F}, min, plus \rangle$

$\forall t{:}\ A^1(t) \geq (A^0 \otimes \beta_1)(t)$

$\forall t{:}\ \alpha^1(t) = (\alpha^0 \oslash \beta_1)(t)$

$\otimes$ - оператор свёртки

$(f \otimes g)(t) \stackrel{\text{def}}{=} \inf_{0 \leq s \leq t}\{f(s) + g(t-s)\}$

$\forall t{:}\ A^2(t) \geq \big((A^0 \otimes \beta_1) \otimes \beta_2\big)(t)$

$\forall t{:}\ \alpha^2(t) \geq ((\alpha^0 \oslash \beta_1)\ \oslash \beta_2)(t)$

$\oslash$ - оператор обратной свёртки

$(w \oslash g)(t) \stackrel{\text{def}}{=} \sup_{t \geq u \geq 0}\{w(t+u) - g(u)\}$

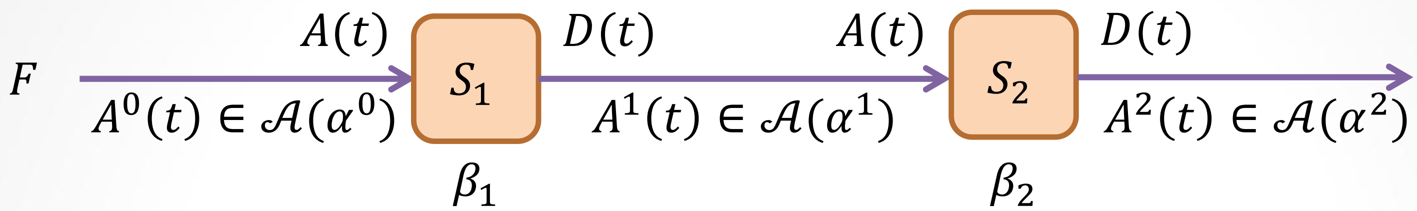
Теорема о композиции обработчиков

Если обработчики S_1 и S_2 с кривыми сервиса β_1 и β_2 образуют ***тандем***, то их систему описывает кривая сервиса $\beta = \beta_1 \otimes \beta_2$

Доказательство:

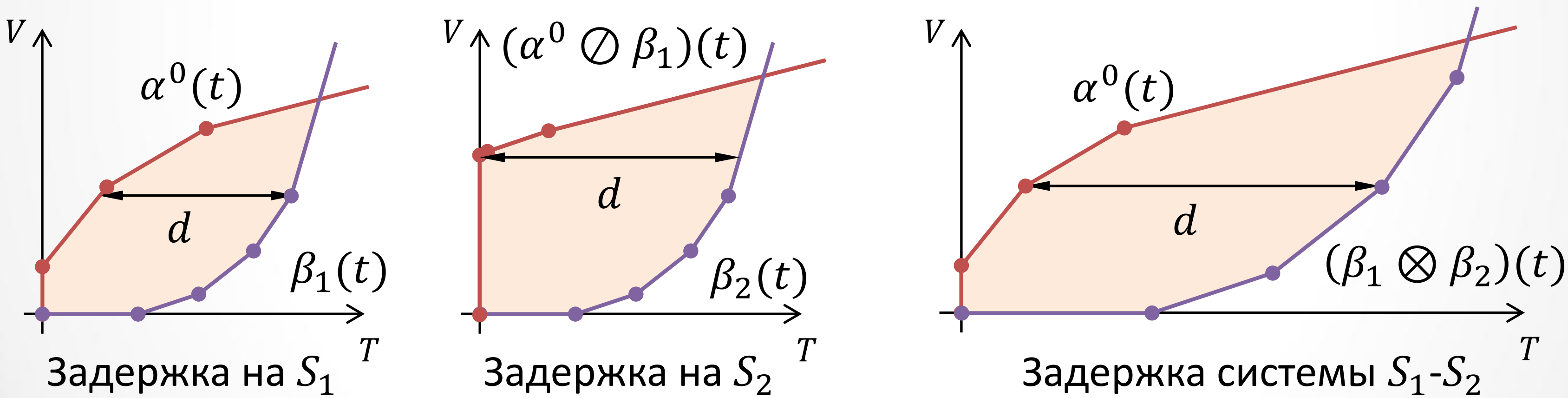
$$D_2 \geq A_2 \otimes \beta_2 = D_1 \otimes \beta_2 \geq (A_1 \otimes \beta_1) \otimes \beta_2$$
$$= A_1 \otimes (\beta_1 \otimes \beta_2) = A_1 \otimes \beta$$

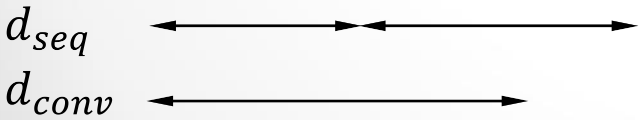
Pay Burst Only Once (PBOO)



$d_{seq}(t) = h(\alpha^0, \beta_1) + h(\alpha^0 \oslash \beta_1, \beta_2)$

$d_{conv}(t) = h(\alpha^0, \beta_1 \otimes \beta_2)$



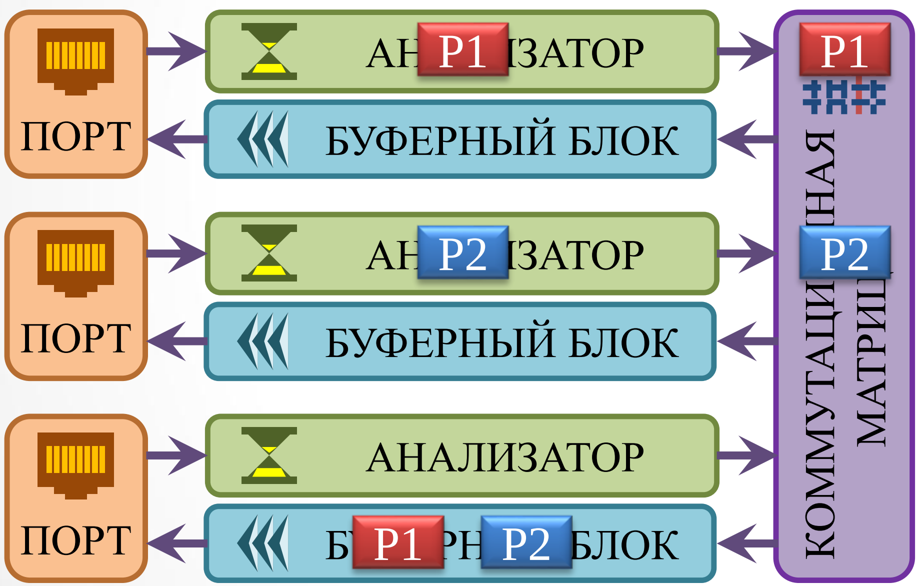


Операция свёртки позволяет учитывать всплеск кривой нагрузки один раз

Блочная модель коммутатора

Анализатор пакетов:
- Разбирает заголовки пакетов
- Вычисляет выходной порт

Задержка d_A анализатора пакетов мало колеблется
$d_A \leq \Delta_A$



Коммутационная матрица перемещает пакеты между портами коммутатора

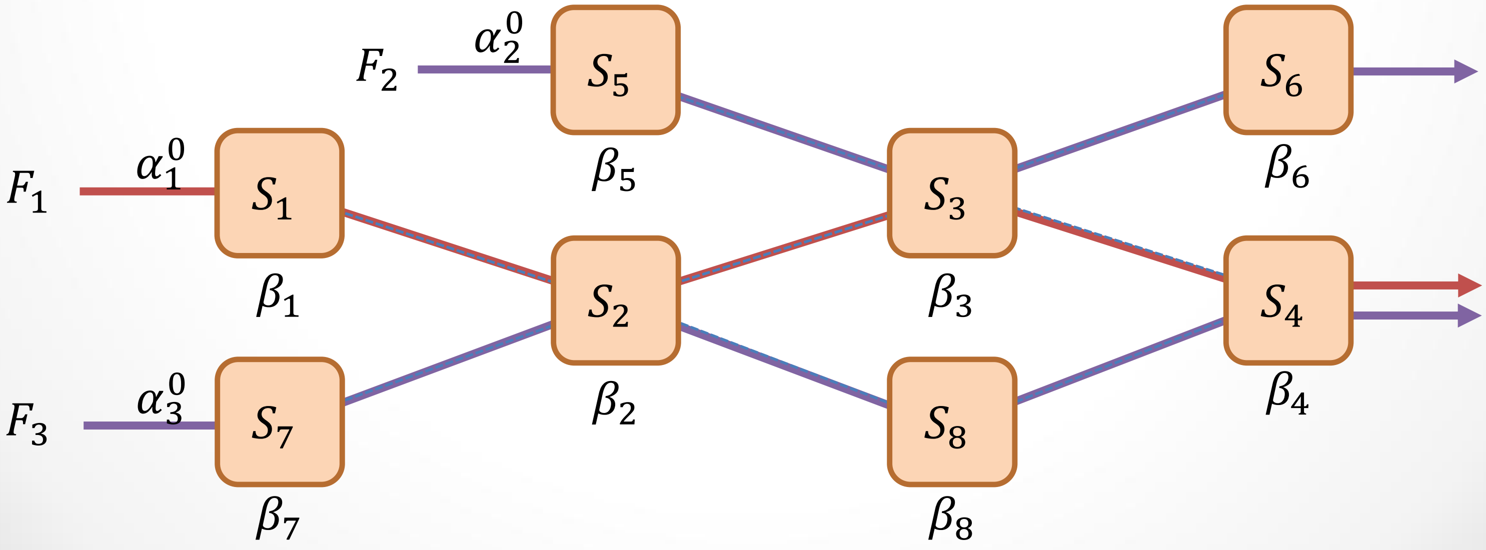
Задержка d_F коммутационной матрицы невелика
$d_F \leq \Delta_F$

Буферный блок
- Сохраняет пакеты пока загружен канал
- Распределяет канал между потоками

Задержка d_B буферного блока зависит от интенсивности потоков данных

Модель ПКС

- Сеть представляется графом обработчиков
- Известны кривые нагрузки для каждого из потоков и кривые сервиса для каждого из обработчиков
- Дисциплины мультиплексирования потоков неизвестны
- Нагрузки не превышают возможности обработчиков
- Граф обработчиков не содержит циклов



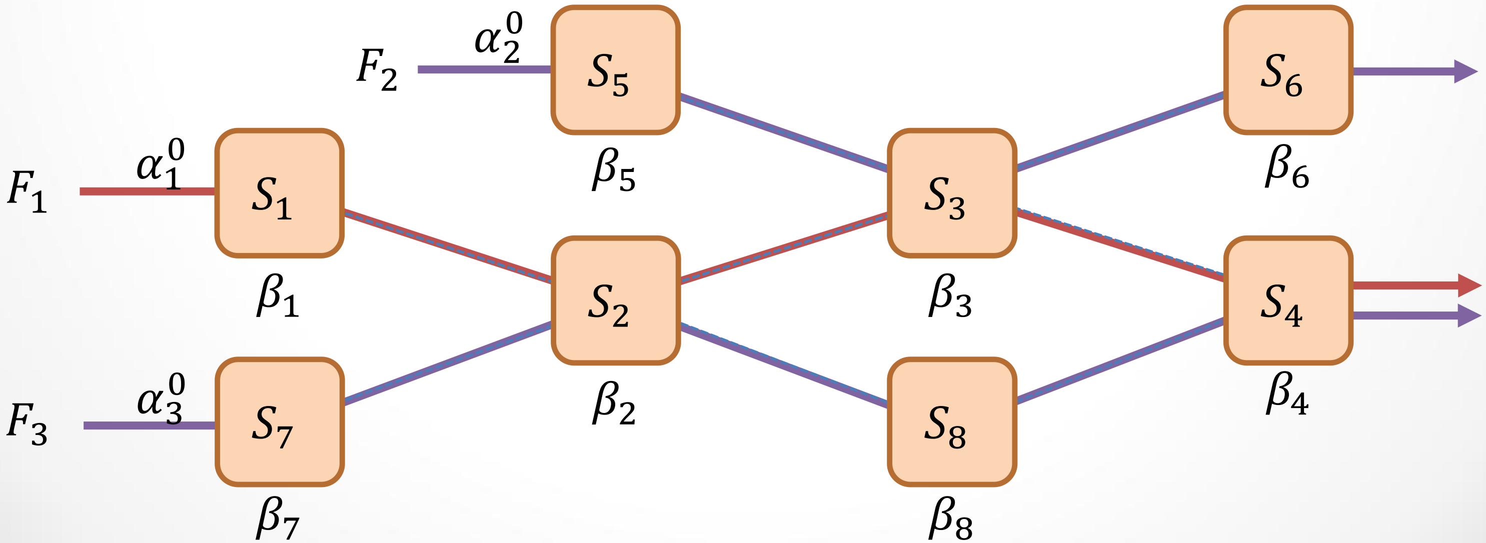
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



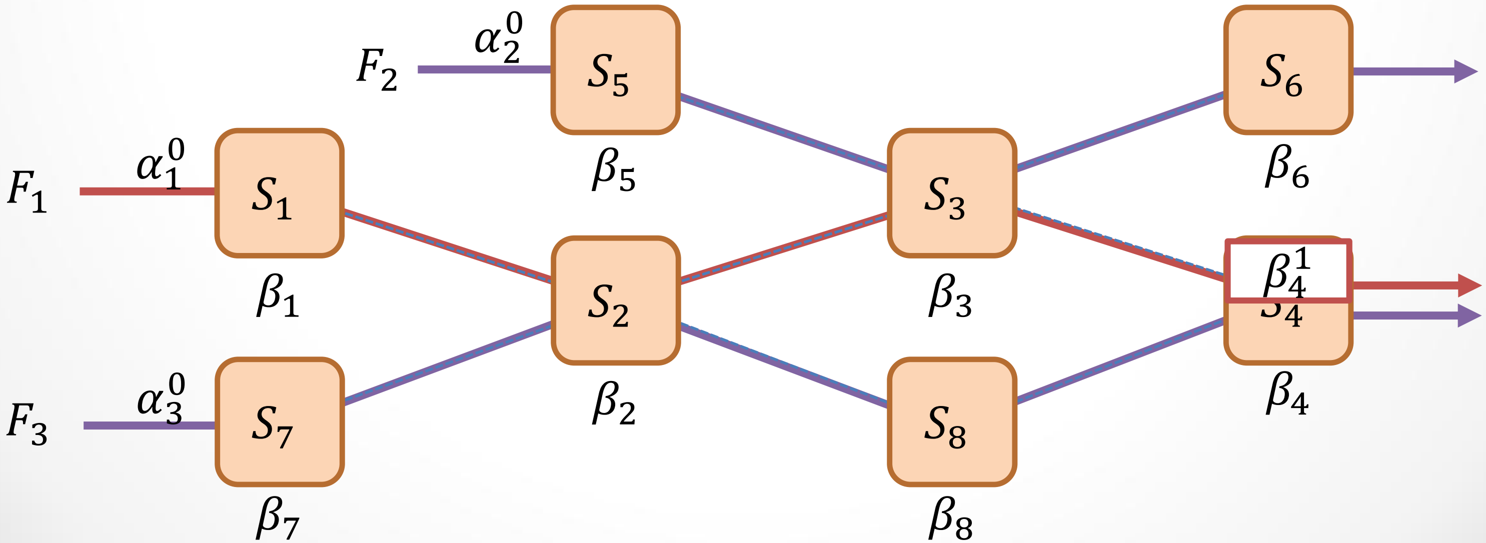
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



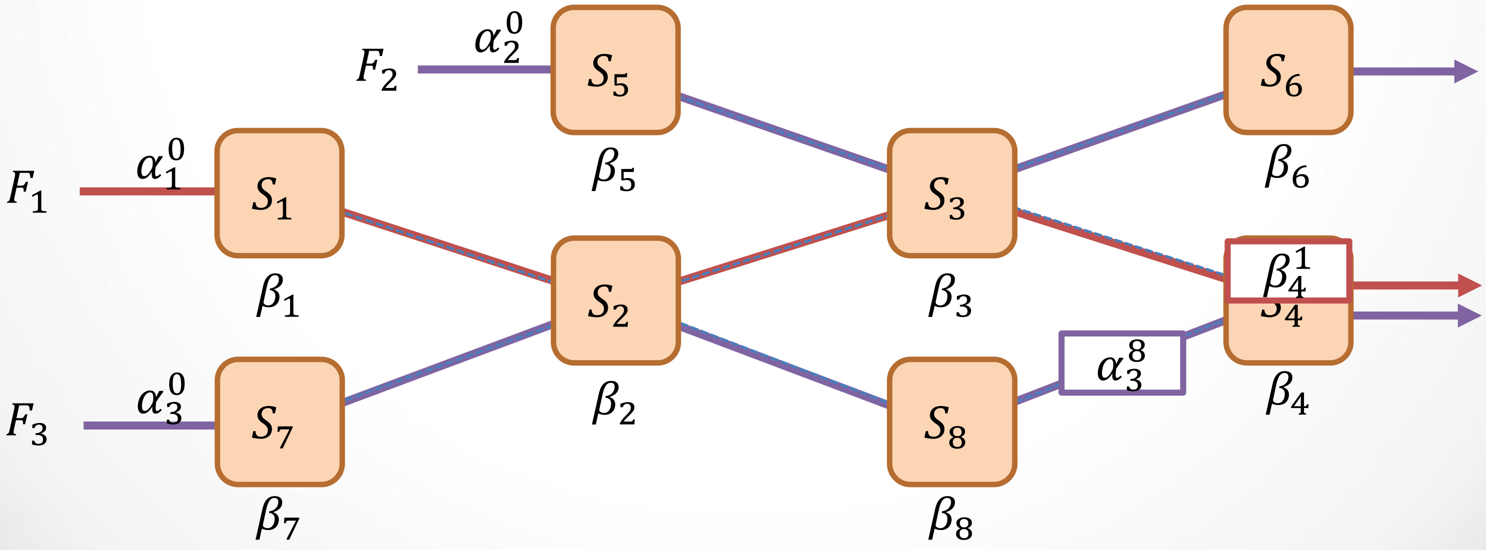
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



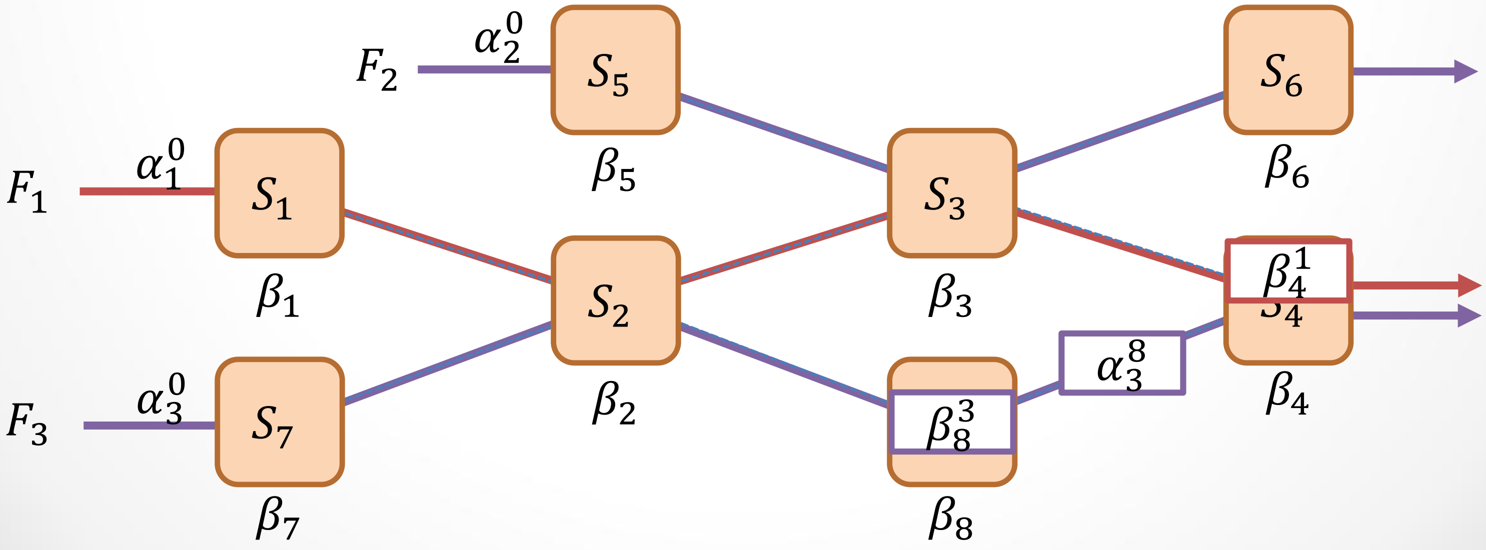
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



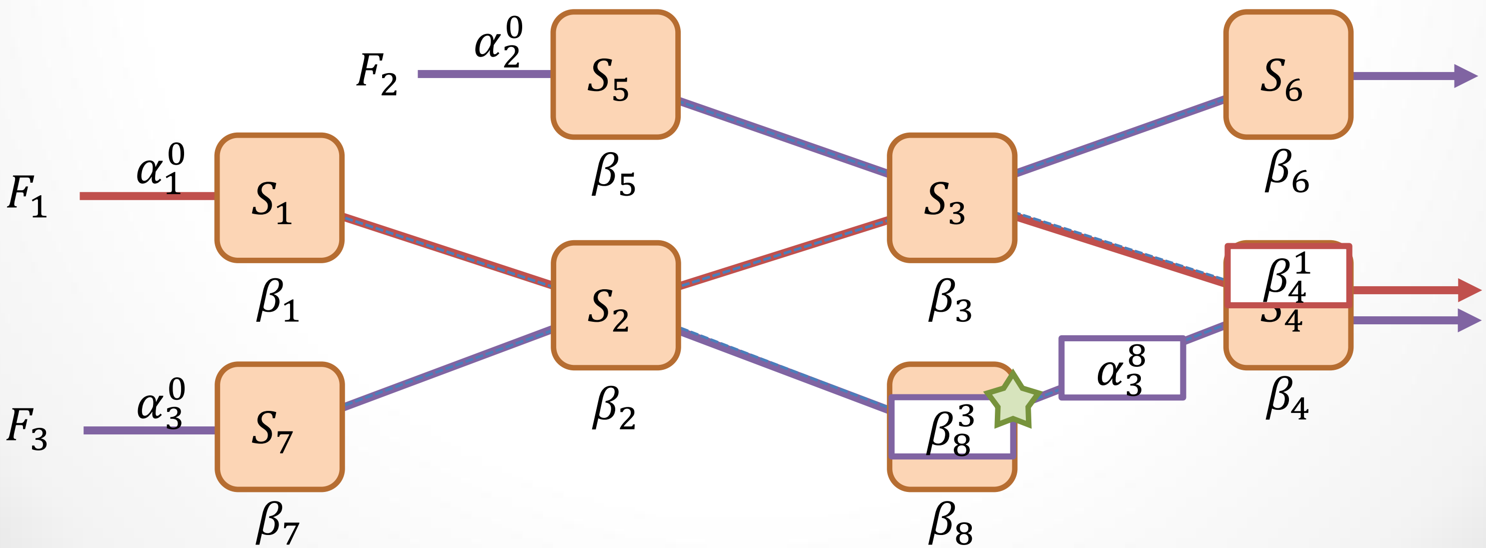
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



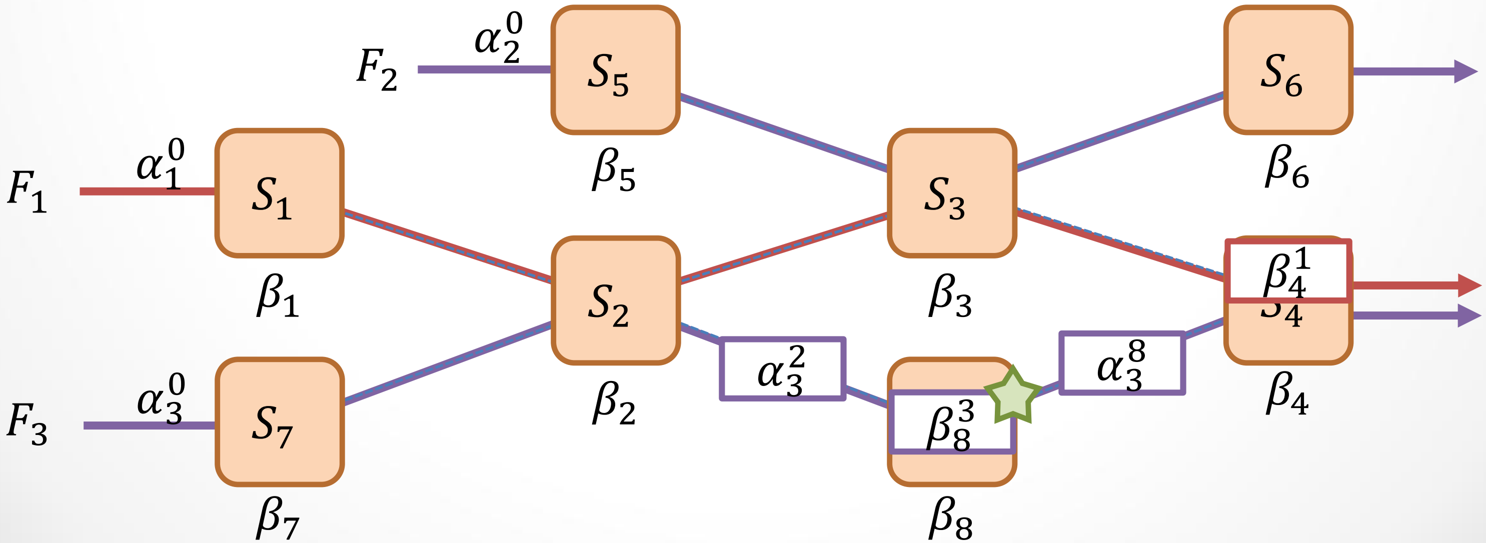
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



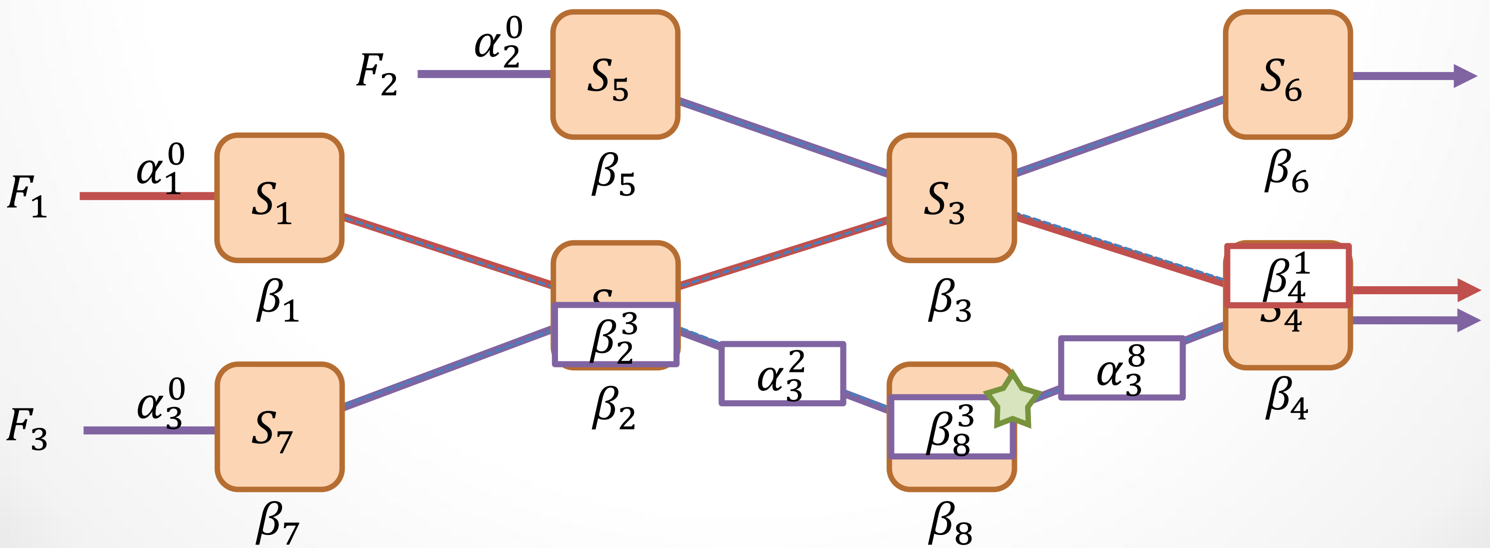
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



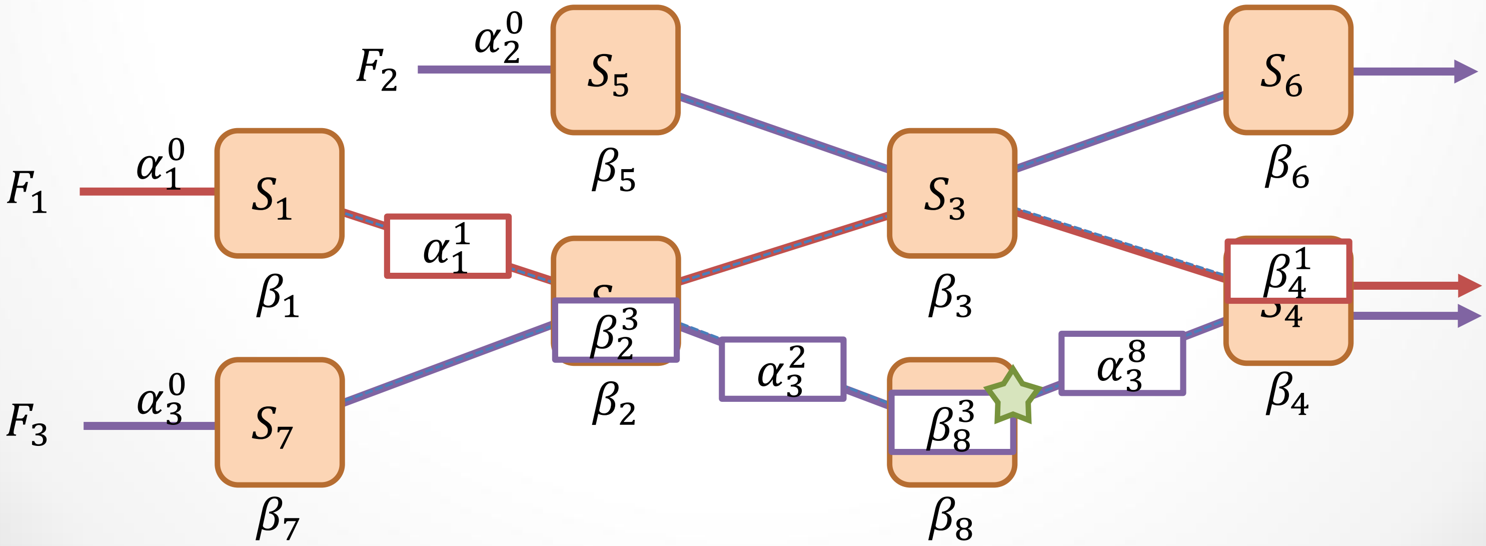
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



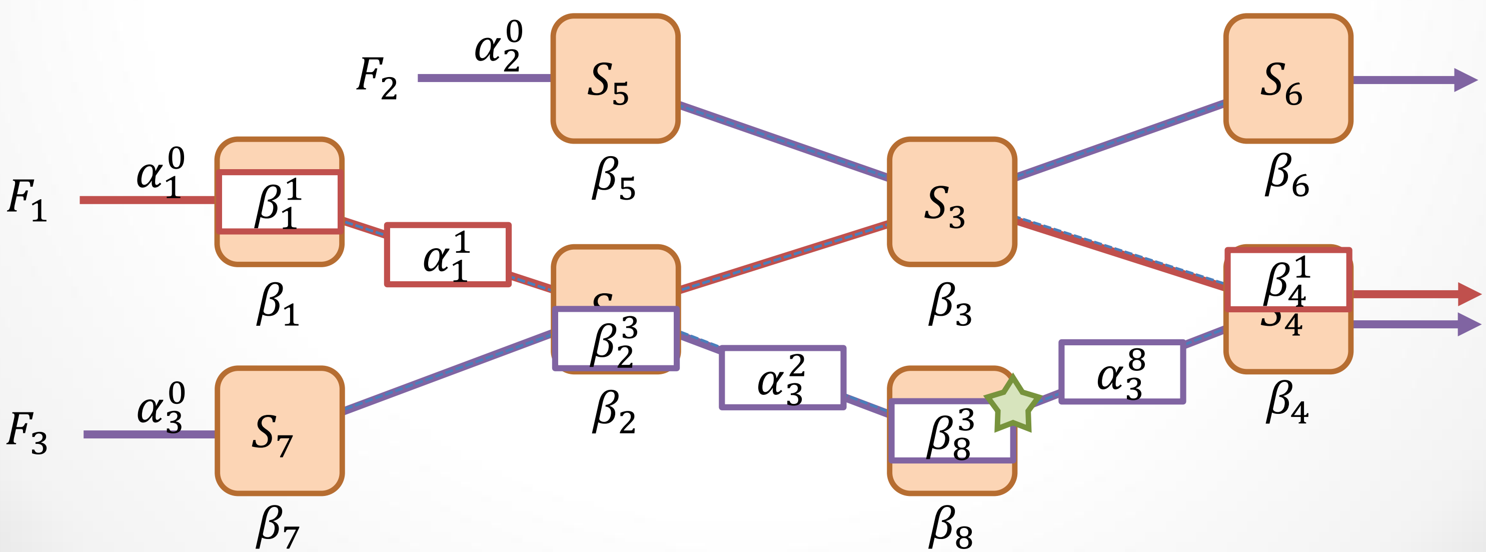
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



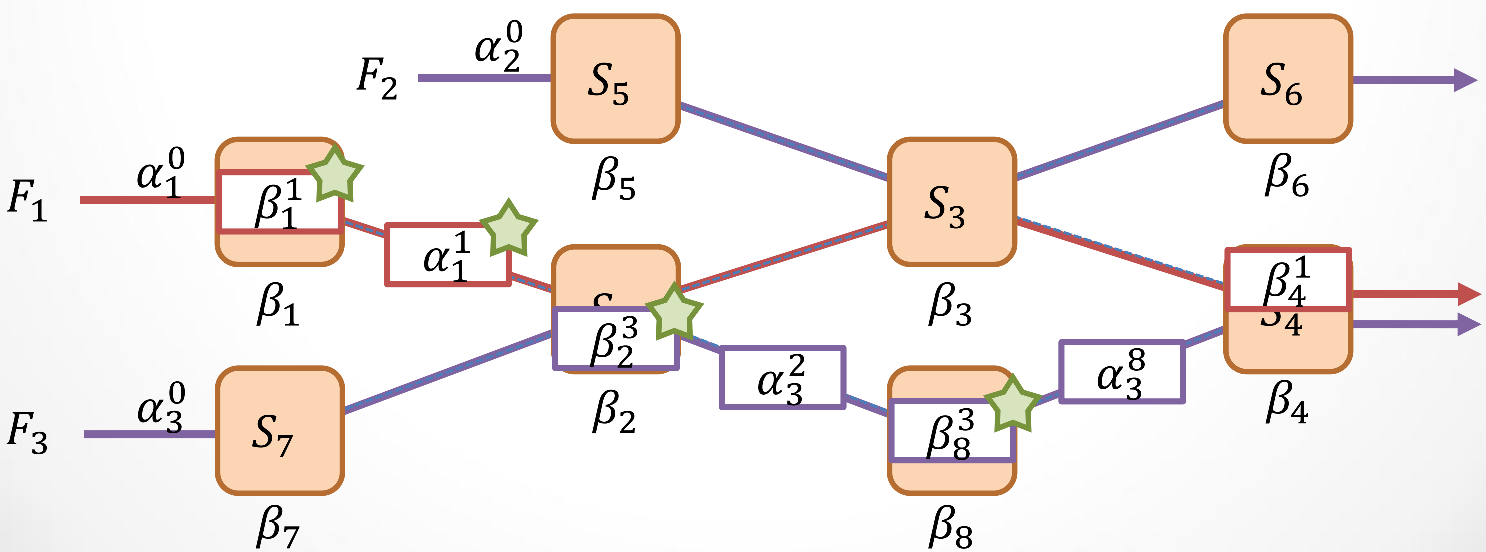
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



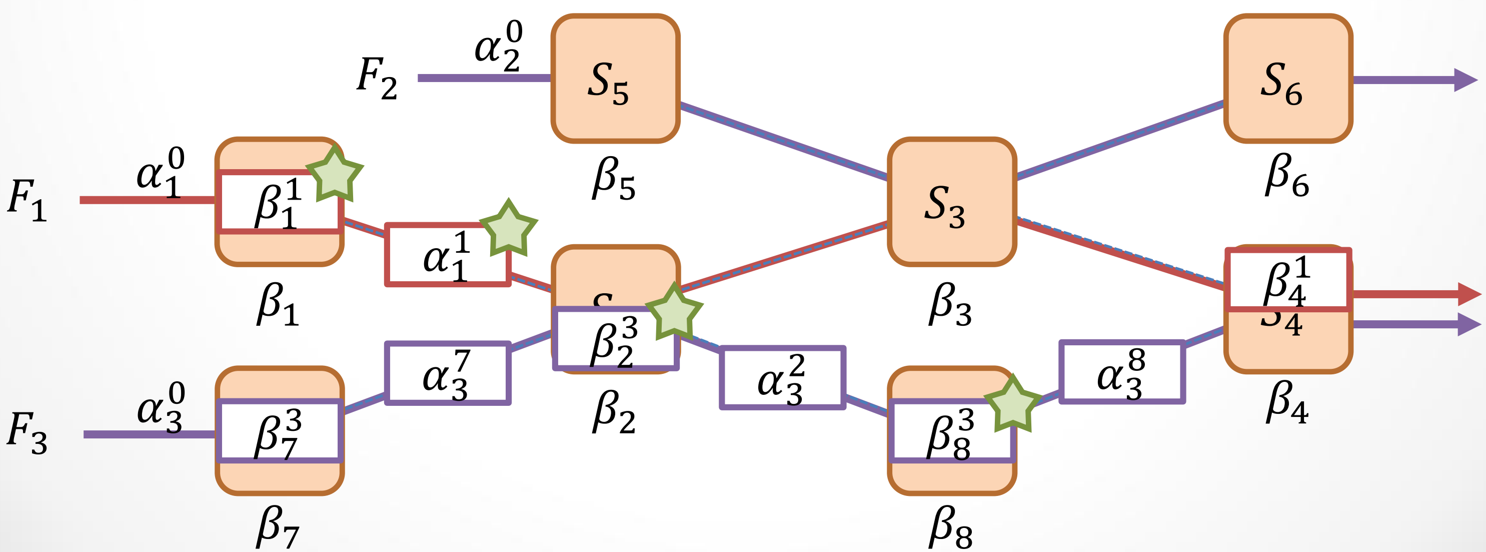
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



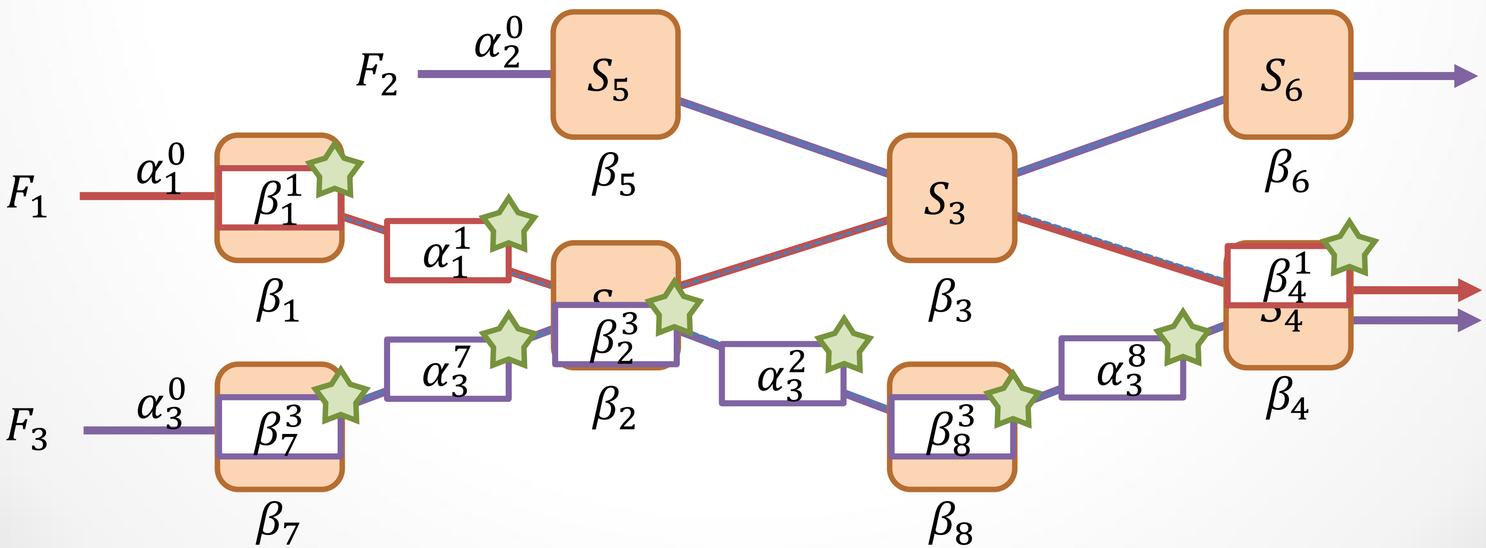
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



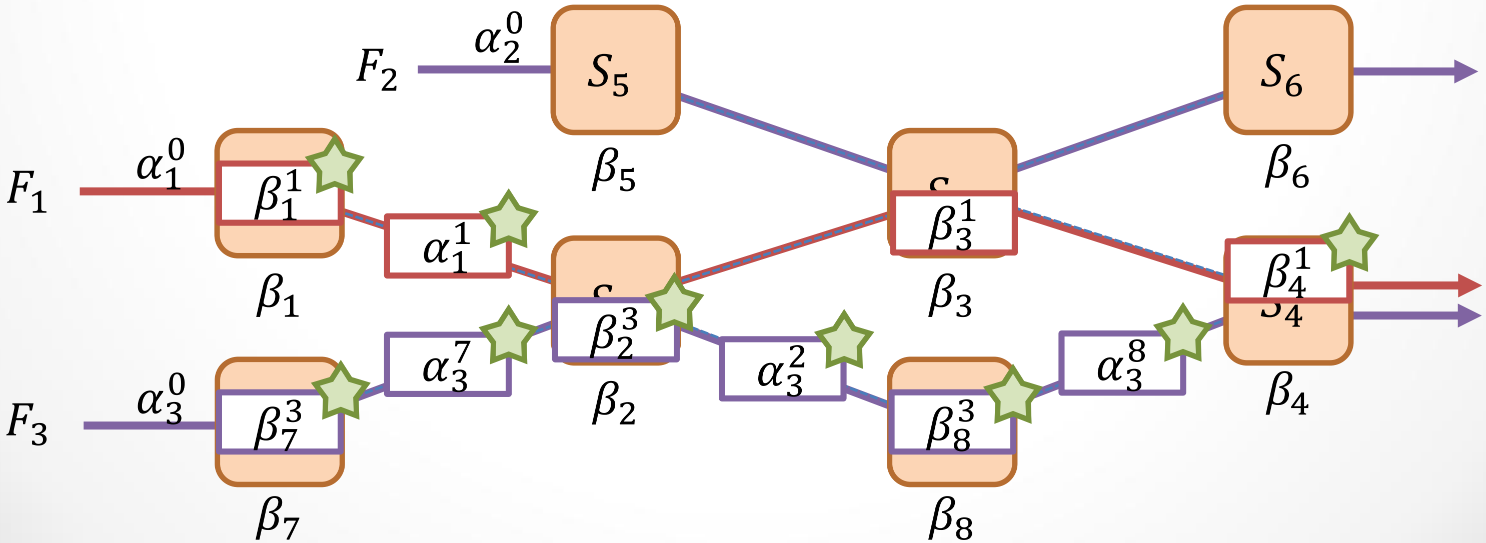
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



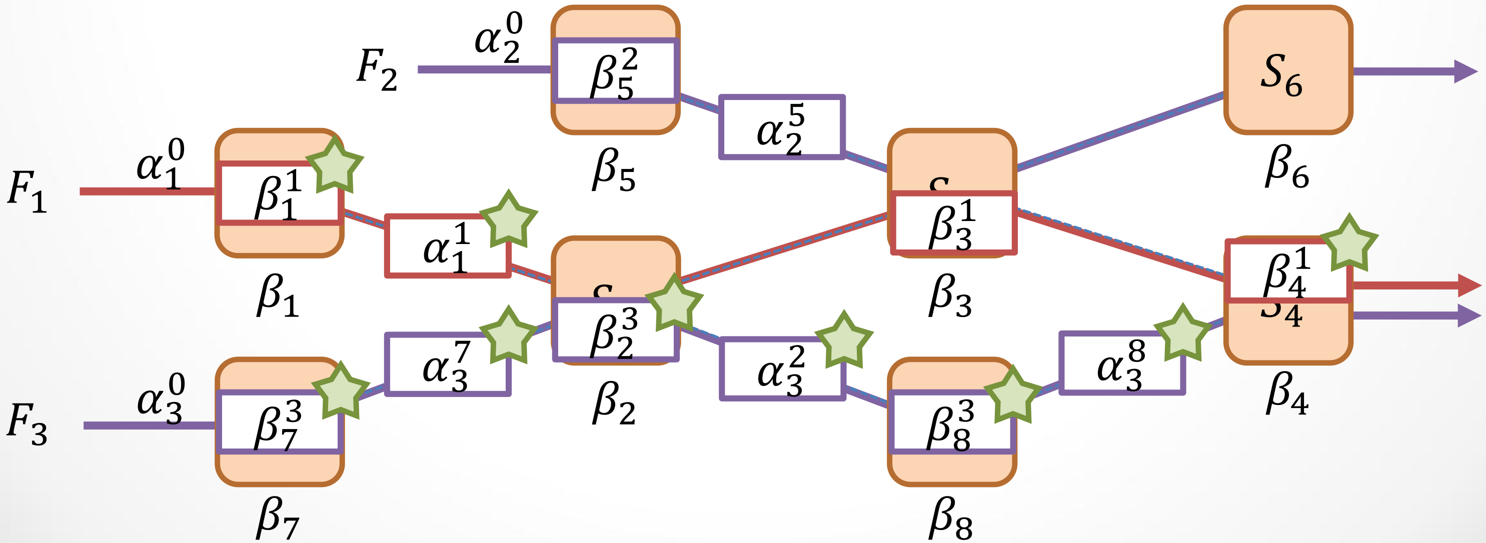
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



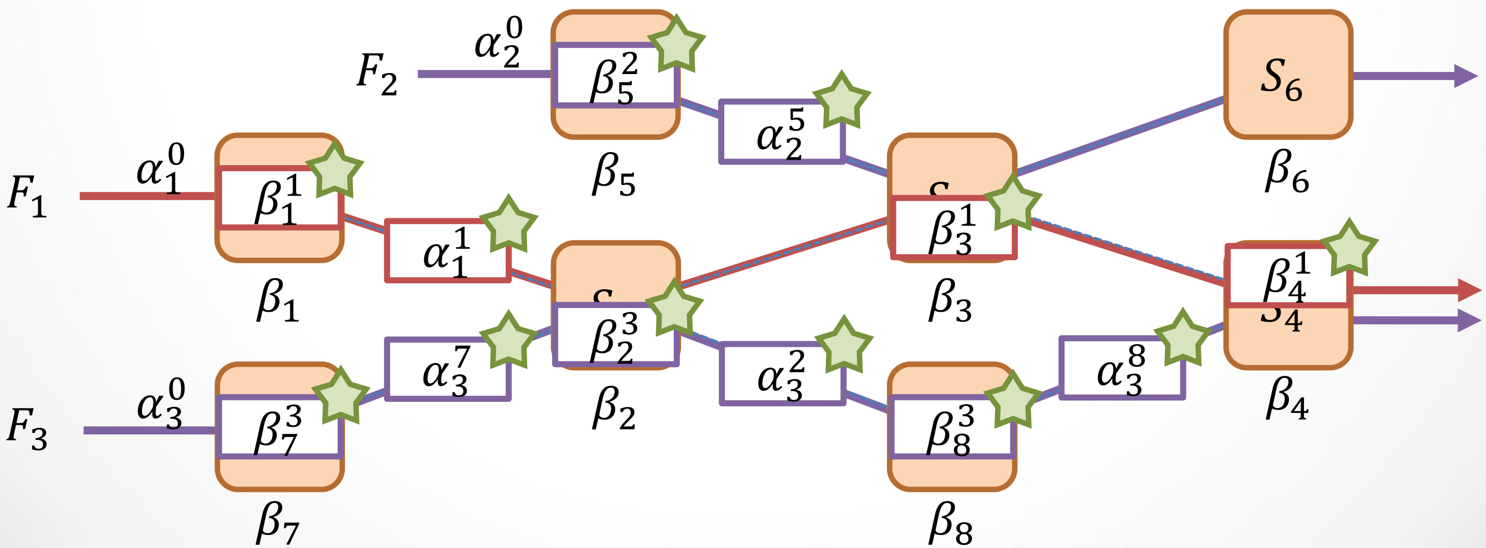
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



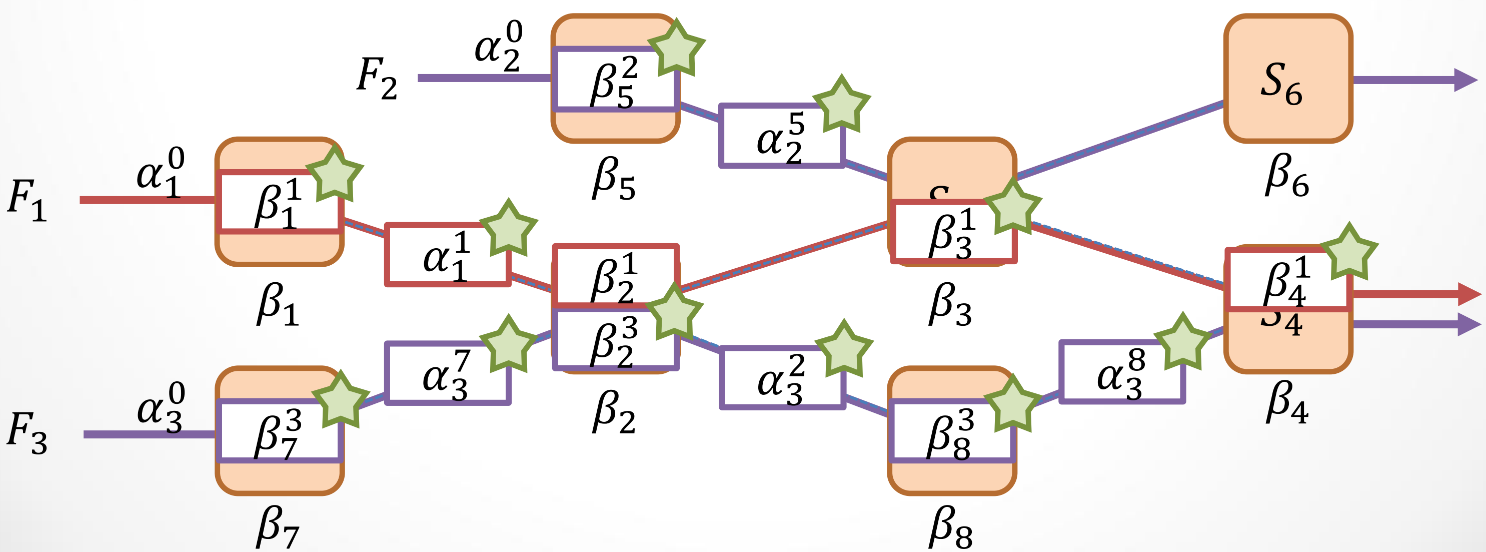
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



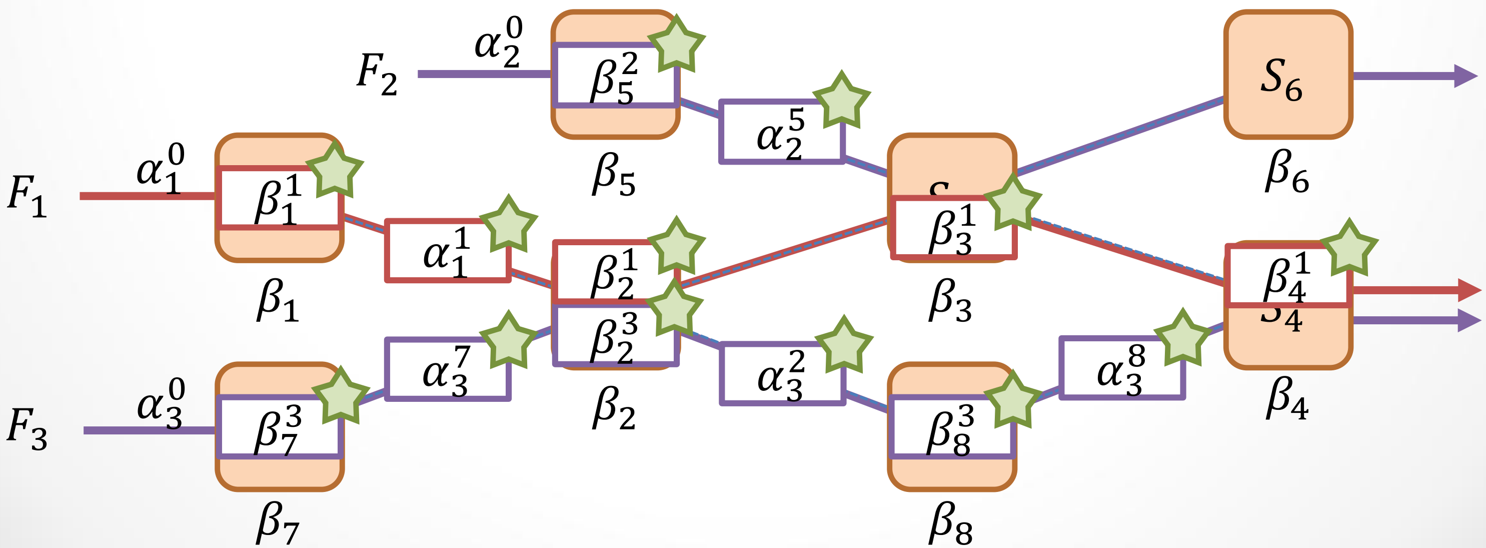
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



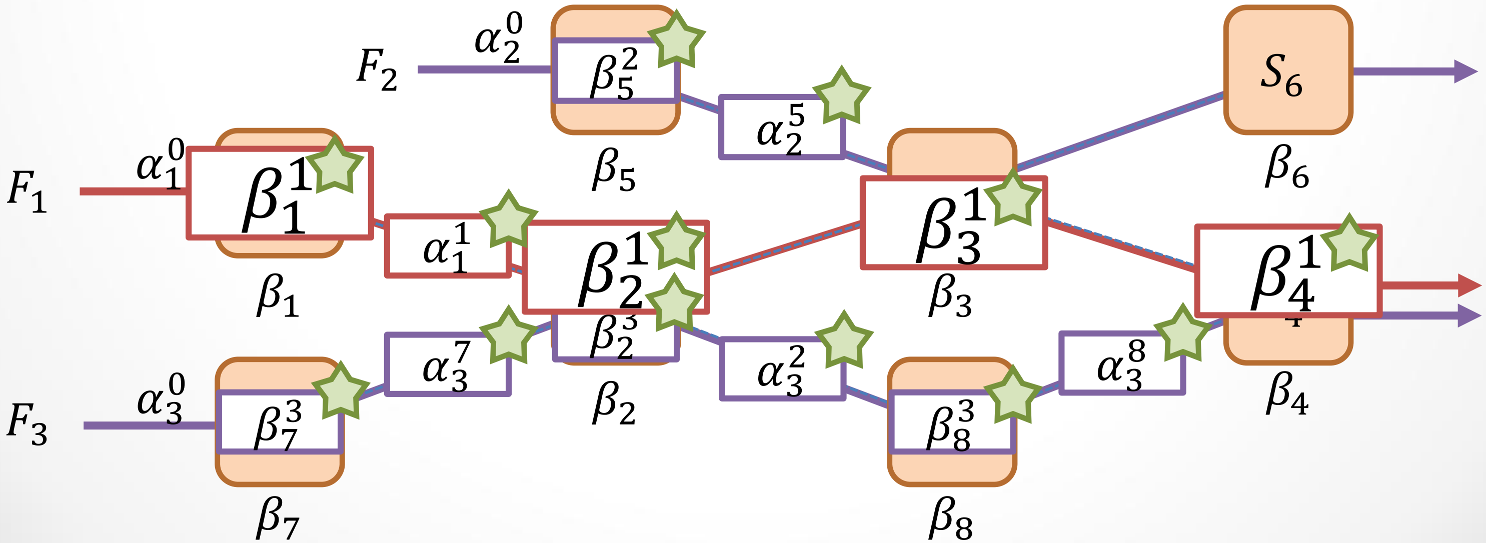
Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки



Алгоритм вычисления задержки

J. B. Schmitt, F. A. Zdarsky, and M. Fidler,
Delay bounds under arbitrary multiplexing:
When network calculus leaves you in the lurch ...
Procedings of INFOCOM 2008

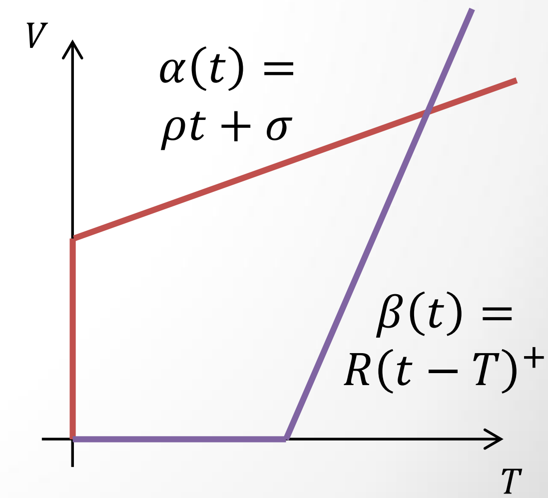
Separated Flow Analysis

Для заданного целевого потока:

1. Построить *остаточные кривые сервиса* обработчиков
2. Вычислить общую кривую сервиса системы
3. Задержка потока равна максимальному расстоянию между кривой сервиса системы и его кривой нагрузки

Нужно 4 операции над функциями:

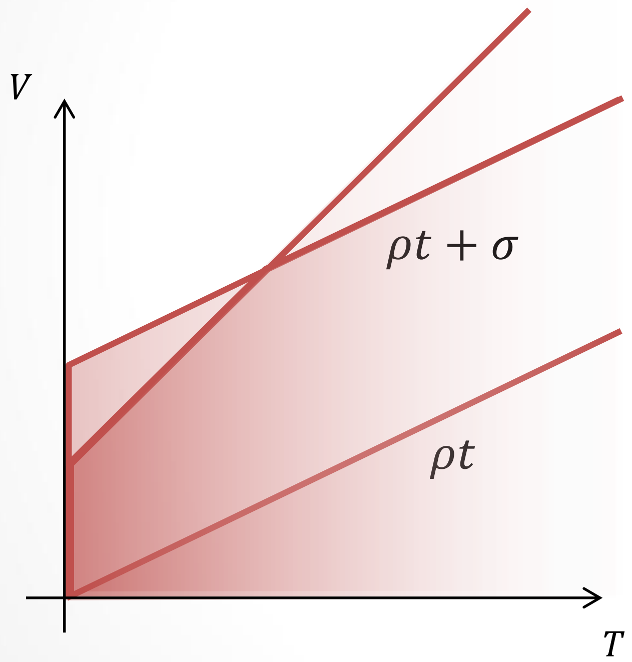
1. Поточечная разность $[\beta - \alpha]^+$
2. Обратная min-plus свёртка $\alpha \oslash \beta$
3. Min-plus свёртка $\beta_1 \otimes \beta_2$
4. Горизонтальное расстояние $h(\alpha, \beta)$



Используются функции простого вида

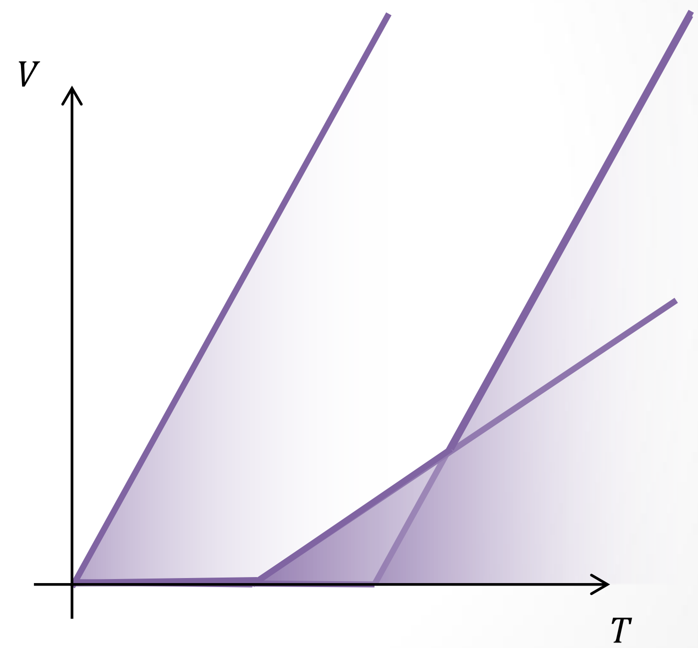
Вид кривых нагрузки и сервиса в сетях

Кривая нагрузки



$\alpha \in \mathcal{F}$ - вогнутая кусочно-линейная

Кривая сервиса



$\beta \in \mathcal{F}$ - выпуклая кусочно-линейная

Операции над функциями из $\mathcal{F}$: min-plus свёртка

$\otimes$ - оператор свёртки

$(f \otimes g)(t) \stackrel{\text{def}}{=} \inf_{0 \le s \le t}\{f(s) + g(t-s)\}$

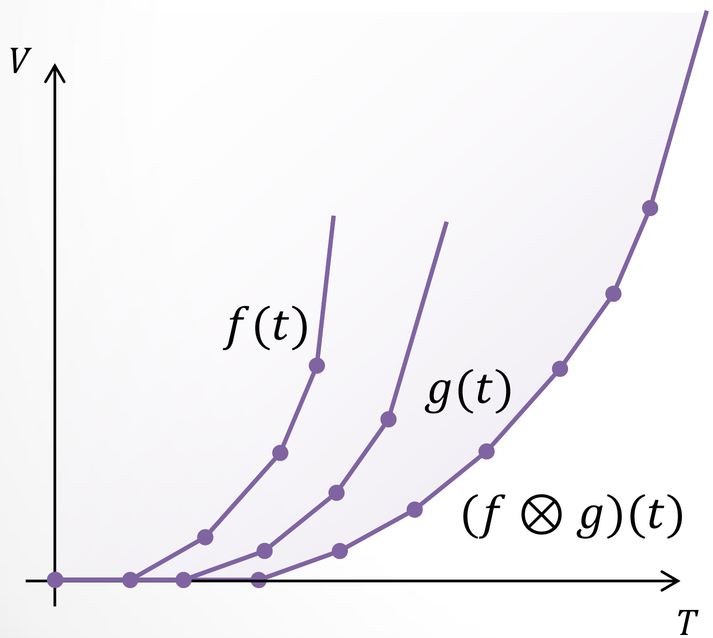


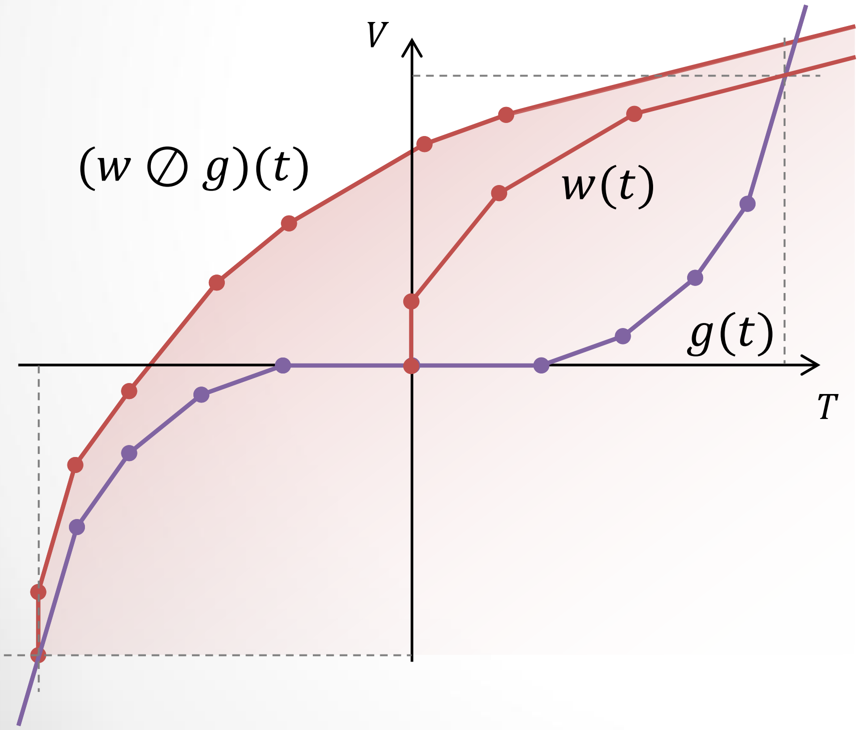
График min-plus свертки выпуклых кусочно-линейных функций из $\mathcal{F}$ можно построить из начала координат путём поочерёдного соединения их сегментов в порядке увеличения их наклонных коэффициентов

Boudec, J-Y., Thiran, P. *Network calculus: a theory of deterministic queuing systems for the internet*, Springer-Verlag, 2001, vol. LNCS 2050, revised version 4, May 10, 2004.

Операции над функциями из $\mathcal{F}$: обратная min-plus свёртка

$\oslash$ - оператор обратной свёртки

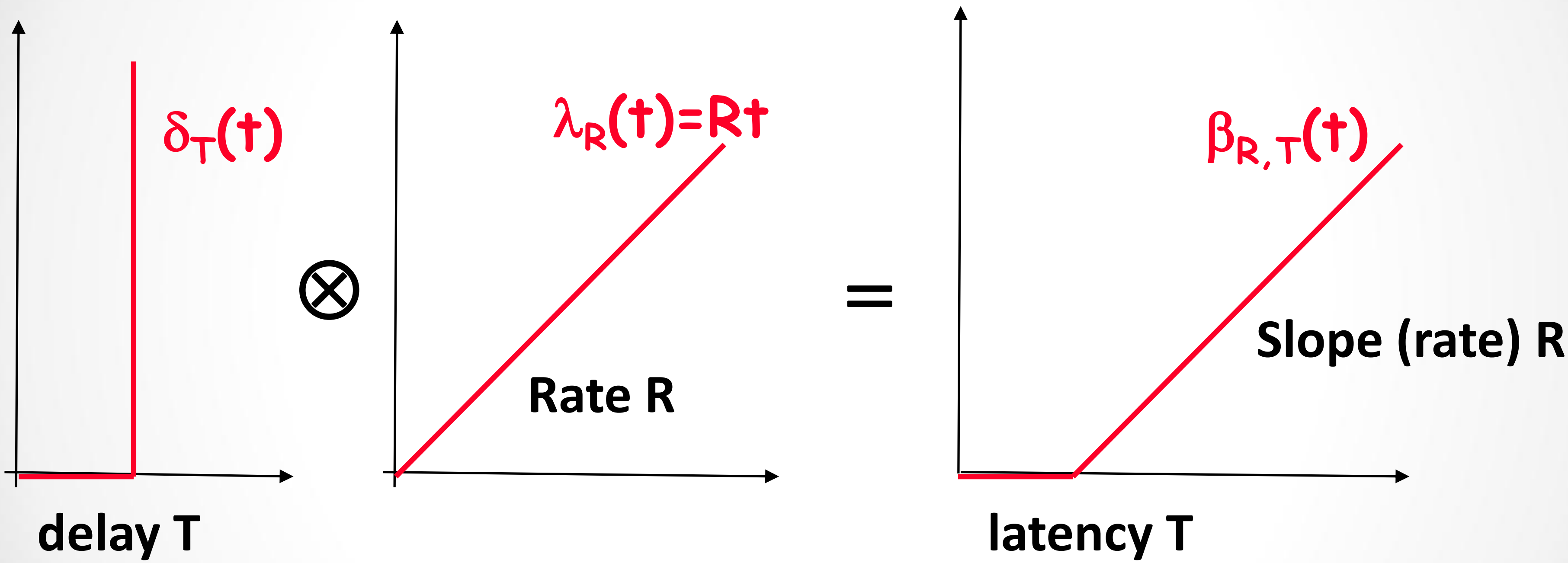
$$(w \oslash g)(t) \stackrel{\text{def}}{=} \sup_{t \ge u \ge 0} \{w(t+u) - g(u)\}$$



Если $w, g \in \mathcal{F}$ - вогнутая и выпуклая функции, причём $w(t) = g(t) \Leftrightarrow t \in \{0, t_0\}$, а график $f(t)$ построен из точки $\langle t_0, -g(t_0)\rangle$ путём соединения сегментов функций $w(t)$ и $g'(t) = \min\big(g(t), g(t_0)\big)$ в порядке уменьшения их наклонных коэффициентов, то функция обратной свёртки $(w \oslash g)(t)$ совпадает с $f(t)$ в положительной полуплоскости: $\forall t > 0: (w \oslash g)(t) = f(t)$

Алгоритм SFA расширен на больший класс фукнций!

Кривая сервиса Rate-Latency



δ_T – имульс (Выпуклая)

λ_R – функция скорости (Выпуклая)

$\beta_{R,T}$ – rate-latency (Выпуклая)

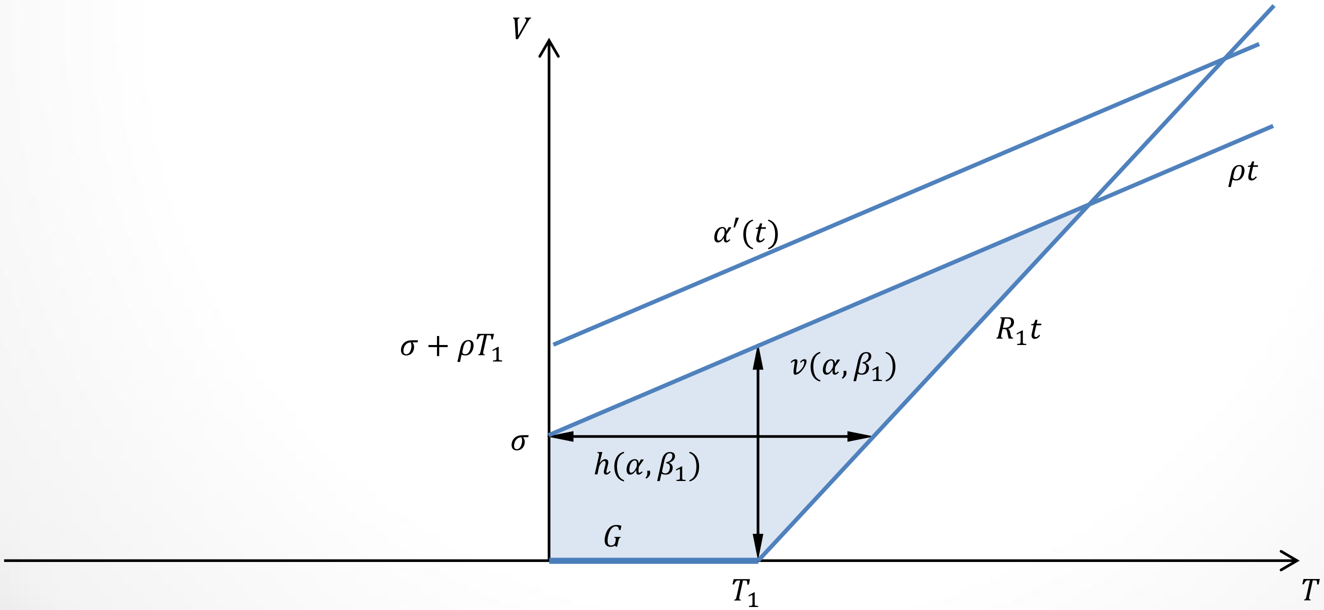
Примеры вычислений

$$\alpha^1(t) = \alpha \oslash \beta_1 = \sup_{u \geq 0}\{\alpha(t+u) - \beta_1(u)\}$$

$$\alpha^1(t) = \max\left(\sup_{0 \leq u \leq T_1}\{\rho(t+u)+\sigma\}, \sup_{u > T_1}\{\rho(t+u)+\sigma - R_1(u - T_1)\}\right)$$

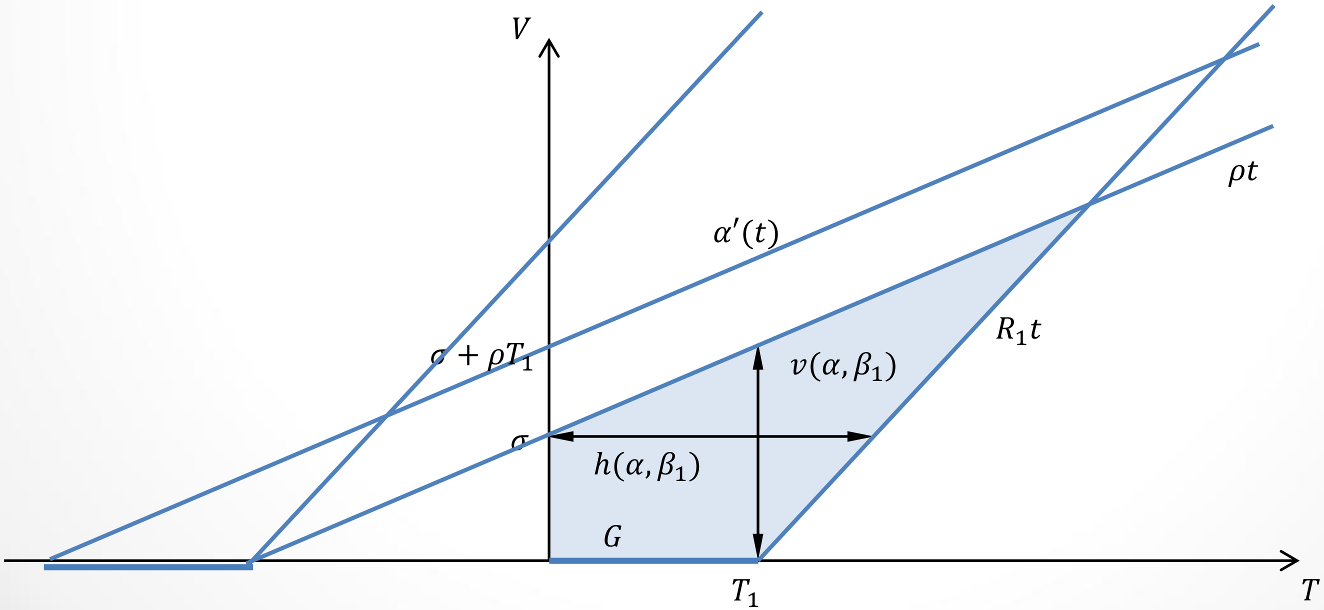
$$\alpha^1(t) = \max\left(\rho(t+T_1)+\sigma, \sup_{u > T_1}\{\rho t + \sigma + R_1 T_1 - u(R_1 - \rho)\}\right)$$

$$\alpha^1(t) = \rho(t+T_1)+\sigma$$

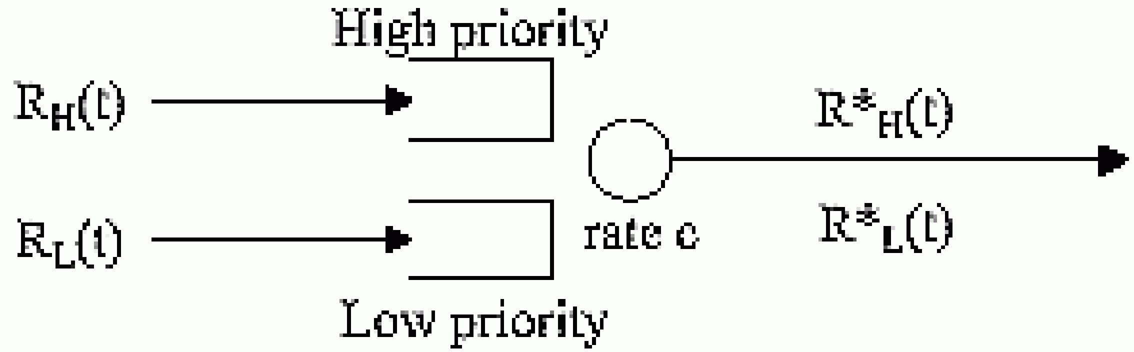


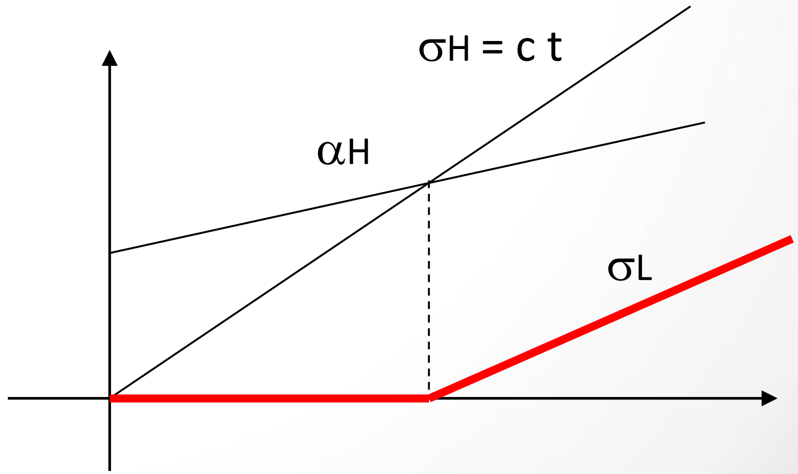
Примеры вычислений

Обратная свёртка от функций w и g – такая функция f, что при выполнении операции свёртки с кривой g получается кривая w

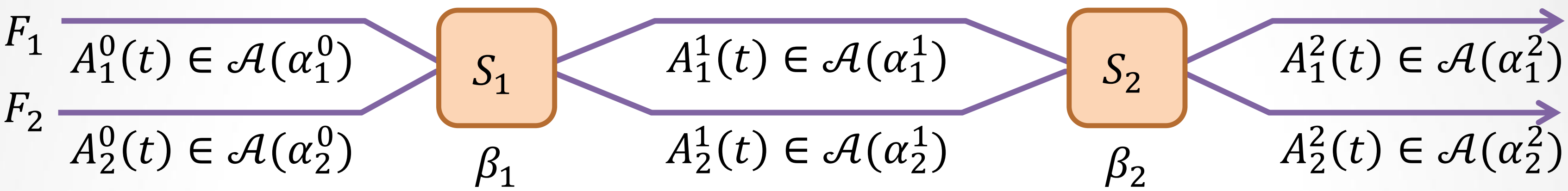


Остаточная кривая сервиса





Pay Multiplexing Only Once (PBOO)



SFA удобно применять в предположениях модели IntServ

В случае DIffServ SFA не точен из-за переучёта мультиплексирования

A. Bouillard, B. Gaujal, S. Lagrange, E. Thierry
Optimal routing for end-to-end guarantees using network calculus
Performance Evaluation, 2008

A. Bouillard, L. Jouhet, E. Thierry
Tight performance bounds in the worst-case analysis of feed-forward networks
INFOCOM 2010

Взаимное влияние потоков зависит от топологии сети -- Network Calculus нужна многокомпонентная свёртка

Задача построения многокомпонентной свёртки может быть заменена задачей линейного программирования

Оценка задержки с помощью линейного программирования

Bouillard A., Jouhet L., Thierry E. Tight performance bounds in the worst-case analysis of feed-forward networks // INFOCOM 2010. — San Diego, USA, 2010. — Pp. 1–9.

Модельные предположения можно использовать для построения такого набора систем линейных неравенств, что их совокупность учитывает особенности топологии

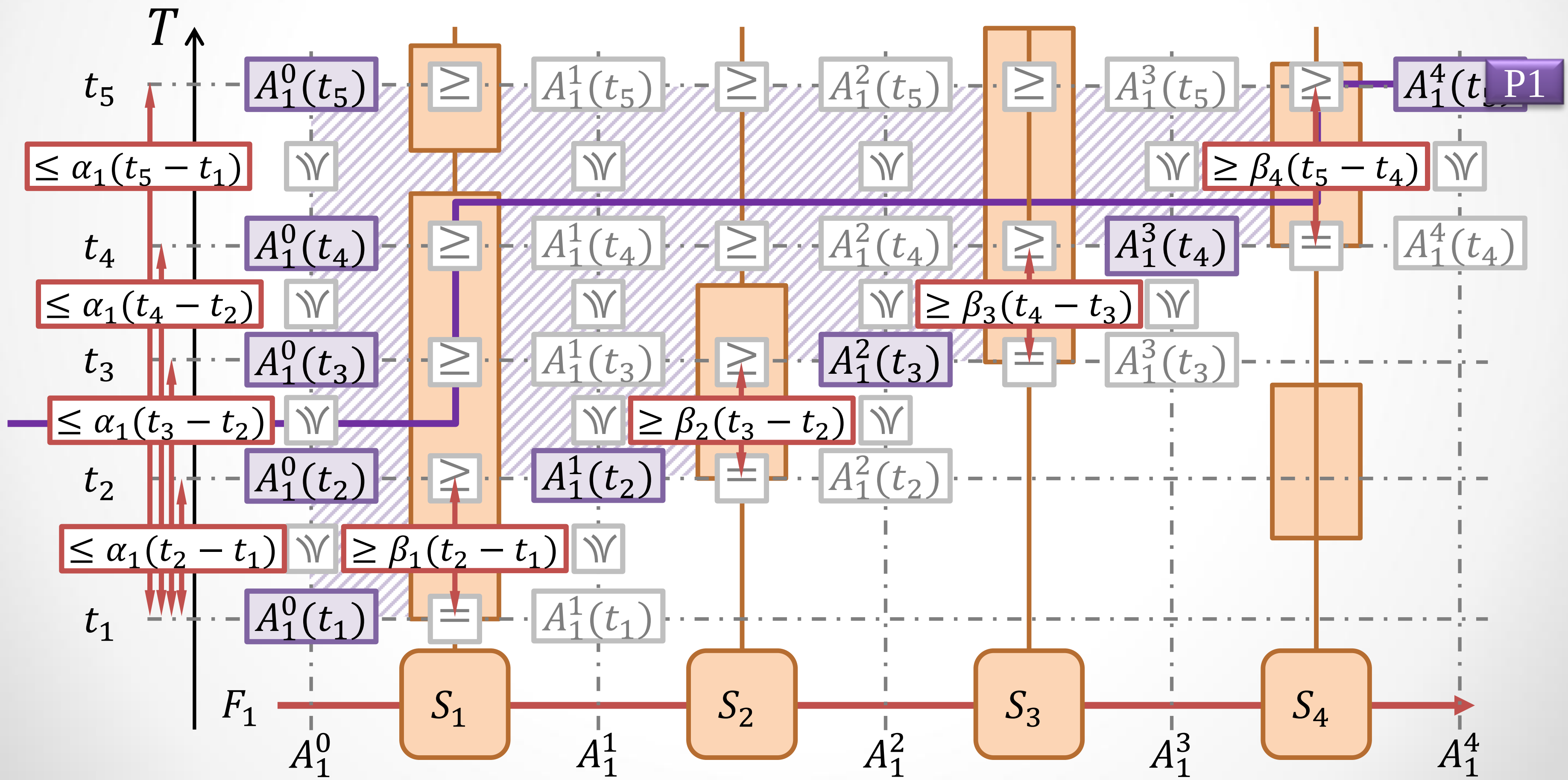
Решение этих систем позволяет получить достижимую верхнюю оценку сквозной задержки

1. Доказательство NP-полноты и алгоритм вычисления достижимой оценки для сетей обработчиков без циклов

2. Полиномиальный алгоритм вычисления достижимой оценки для сетей, представленных тандемом обработчиков

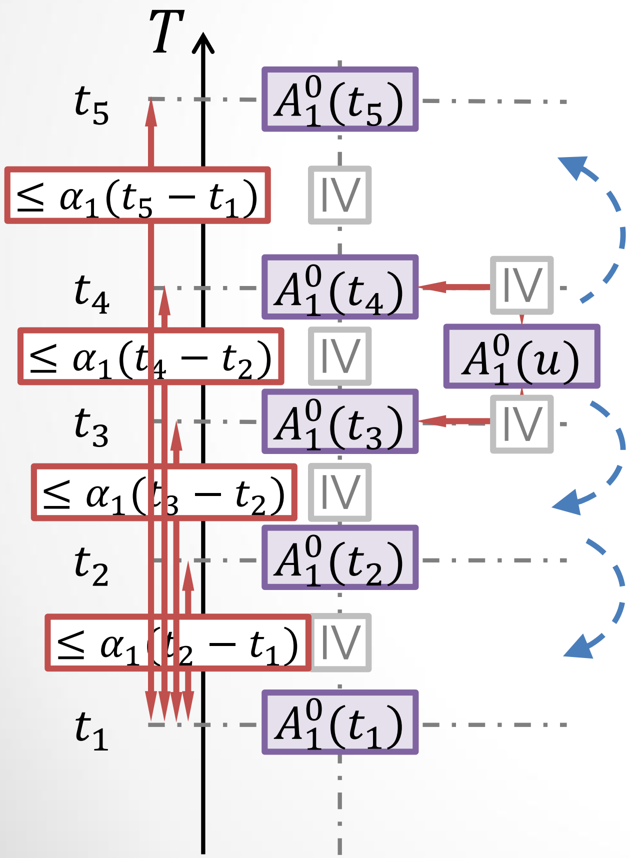
Оценка задержки как задача линейного программирования

Построение системы ограничений для траектории потока



Оценка задержки с помощью линейного программирования

Перебор решений путём дополнения системы

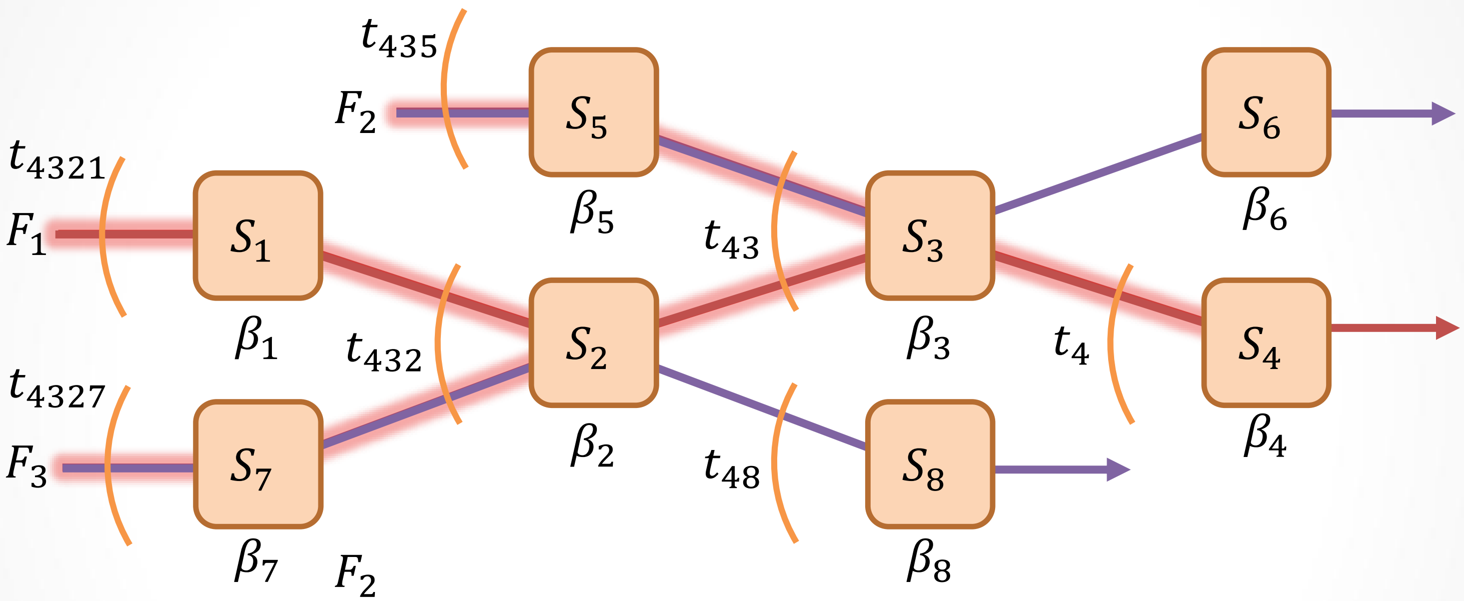


1. Решим задачу линейного программирования для каждого из предположений $t_k \leq u \leq t_{k+1}$:

$$A_1^0(t_{k+1}) - A_1^0(u) \leq \alpha_1(t_{k+1} - u)$$
$$A_1^0(u) - A_1^0(t_k) \leq \alpha_1(u - t_k)$$
$$A(u) \geq A_1^5(t_0)$$

2. Наибольшее из полученных решений является достижимой оценкой сквозной задержки сверху

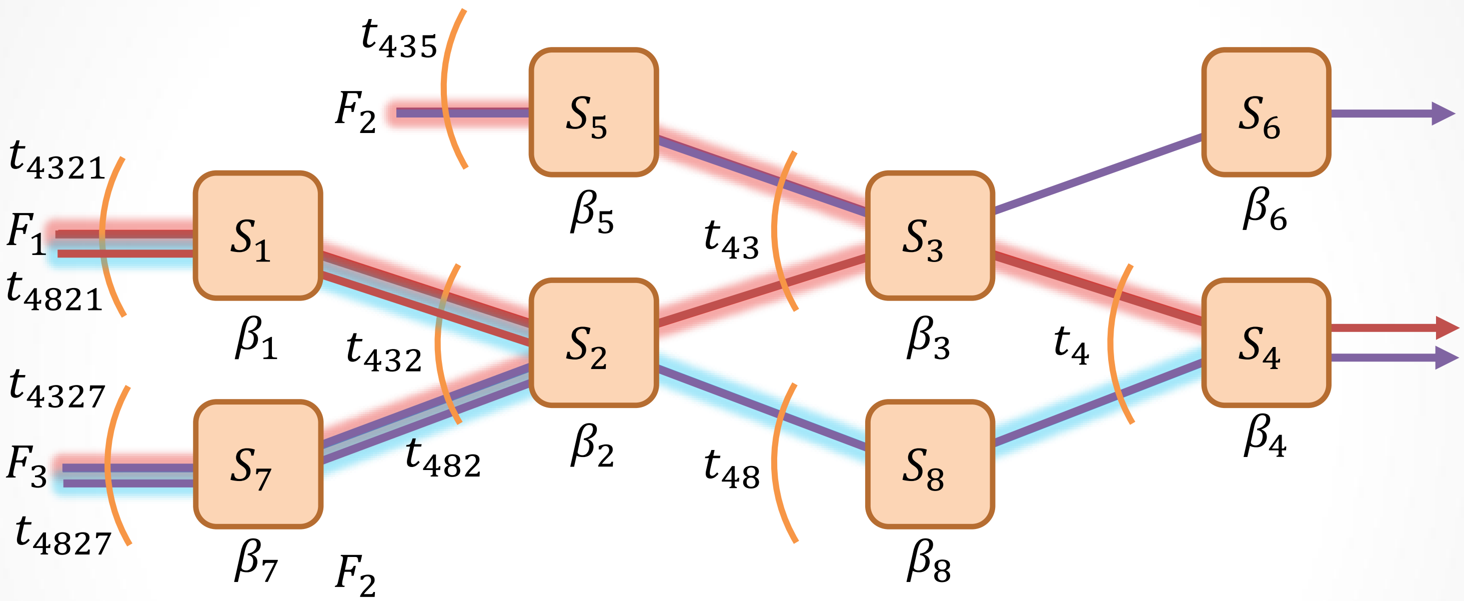
Оценка задержки с помощью линейного программирования



8. Необходимо построить сетки ограничений для всех потоков, влияющих на целевой, и связать их между собой

Получен полиномиальный (по числу потоков) алгоритм вычисления достижимой верхней оценки задержки для сетей обработчиков без альтернативных путей

Применение метода на графах с альтернативными путями

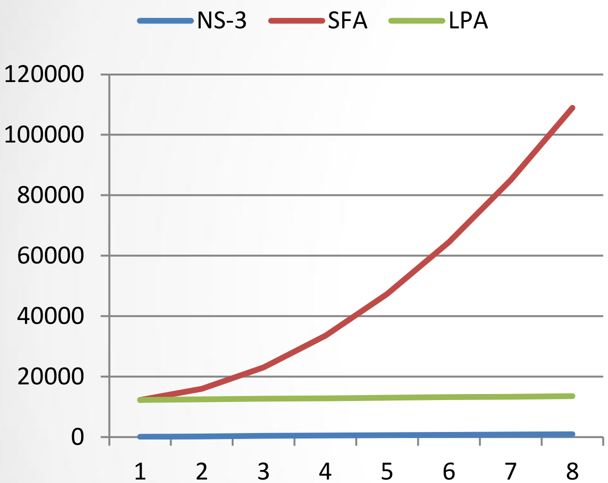


Если одному потоку соответствует несколько сеток ограничений, то для каждого обработчика их периоды отставания, или не пересекаются, или совпадают

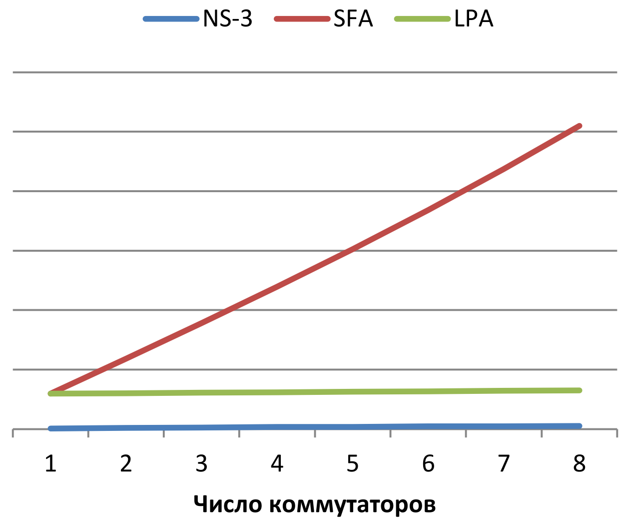
$$\Psi_4\colon \forall k, \pi_1 \neq \pi_2\colon$$
$$t_{k\pi_1} = t_{k\pi_2} \vee t_{\pi_1} < t_{k\pi_2} \vee t_{\pi_2} < t_{k\pi_1}$$

Оценки сквозной задержки (мкс) для линейной топологии

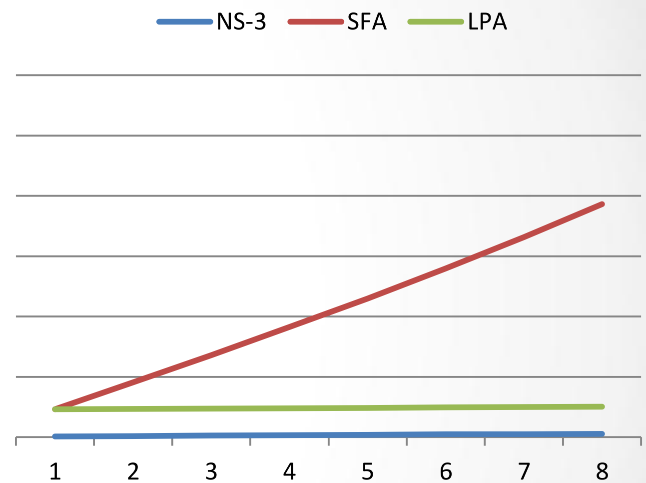
Голосовой поток

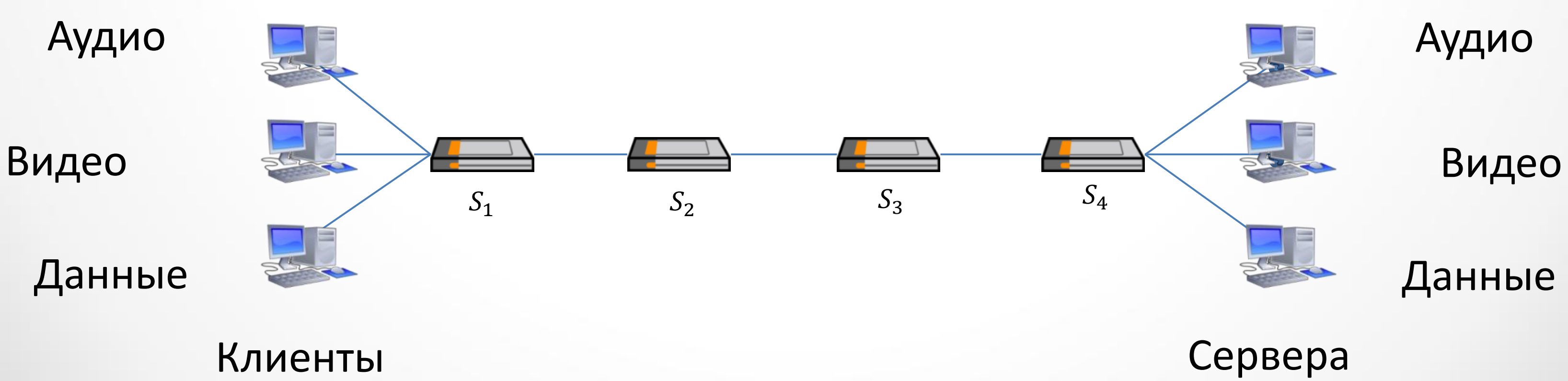


Видео потока



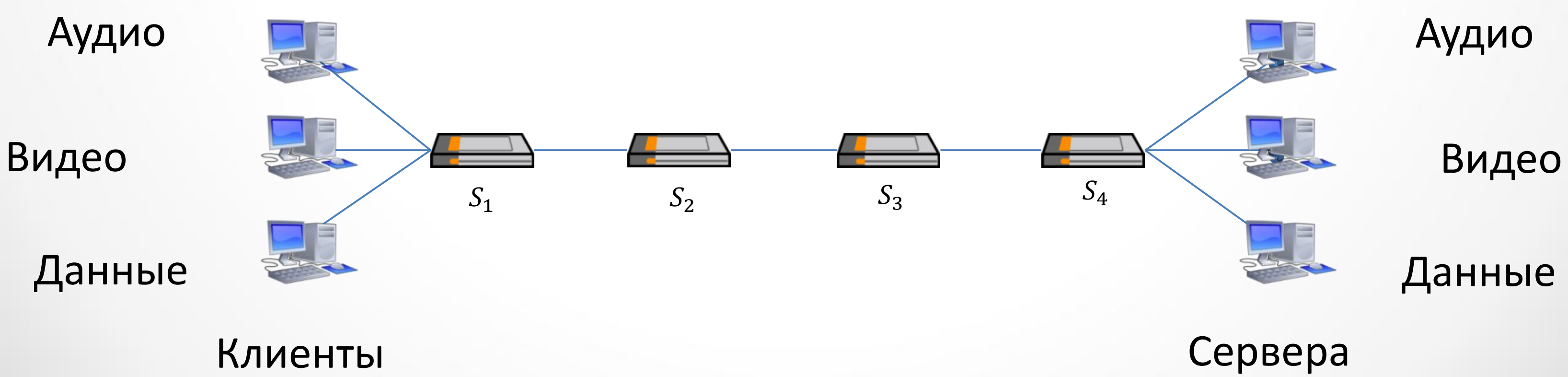
Задержка потока данных





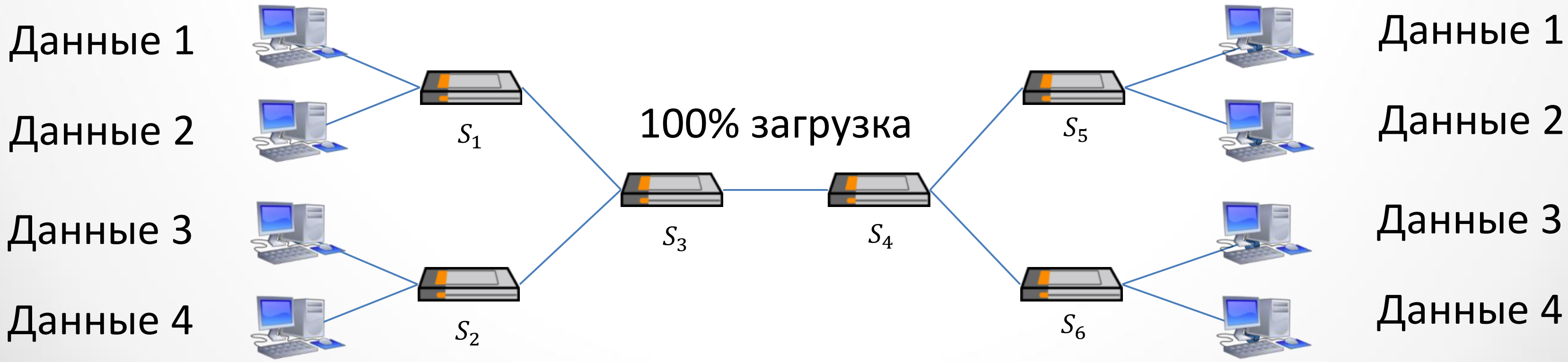
Оценки сквозной задержки (мкс) для линейной топологии

	Аудио			Видео			Данные		
Длина	NS3	SF	LP	NS3	SF	LP	NS3	SF	LP
2	159	12348	12348	244	11881	11881	244	9204	9204
3	281	16053	12519	459	23681	12045	366	18167	9334
4	403	23123	12689	488	35640	12209	515	27231	9457
5	525	33558	12860	692	47877	12373	639	36486	9584
6	647	47359	13031	733	60510	12537	733	46022	9711
7	769	64527	13201	891	73662	12701	962	55934	9838
8	891	85063	13372	998	87459	12866	977	66316	9964
9	1013	108972	13543	1099	102031	13030	1099	77270	10091



Экспериментальное исследование

Глубина	Кратность	Скорость	NS3	SF	LP
1	4	25Mbps	593	2584	2584
1	8	12.5Mbps	1058	8982	8982
1	16	6.25Mbps	1175	33590	33590
2	2	25Mbps	837	4771	3240
2	4	6.25Mbps	1535	39407	35710
3	3	12.5Mbps	1547	19186	11288
4	2	6.25Mbps	2721	63479	39998



Заключение

- Аппарат сетевого исчисления позволяет получать высокоточные оценки для задержки, но имеет ряд ограничений:
  - Неточности при мультиплексировании
  - Работа с циклическими топологиями
  - Абстракции жидкостной модели
- Незатронутые задачи:
  - Оценка отставания
  - Планирование передачи данных
  - Стохастическое сетевое исчисление