Infrastructure

Адаптивный битрейт для видеоконференций

Александр Алексеев,
разработчик

Александр Алексеев

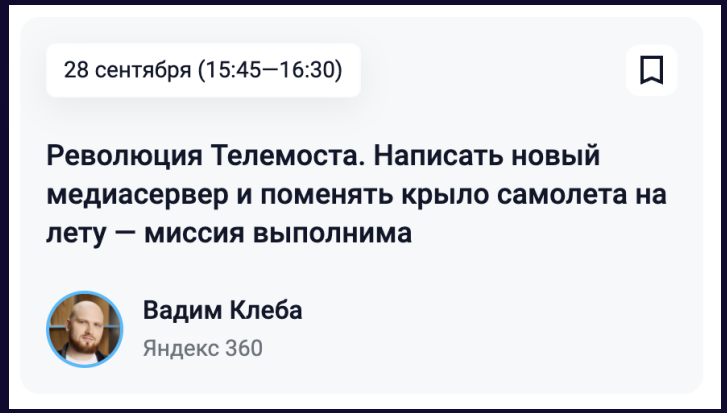
- 1.5 года занимаюсь WebRTC
- 2.5 года работаю с видео в Yandex Infrastructure
- Учусь в магистратуре МФТИ

WebRTC платформа Yandex Infrastructure

Единый стек технологий Яндекса для работы с видео в real-time

Встречи на неограниченное число участников

«Революция Телемоста»



Проблема

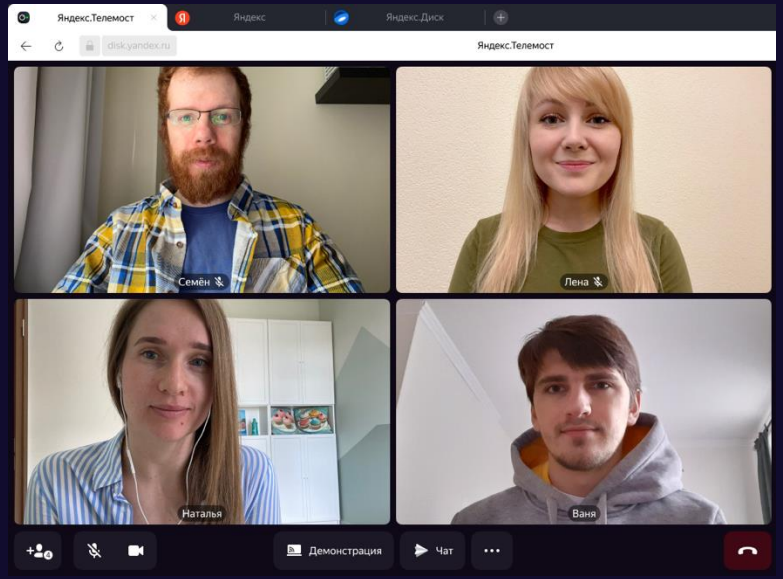
Хороший интернет – высокое качество связи
Плохой интернет – отсутствие связи

ИЛИ

Среднее качество связи при любом интернете

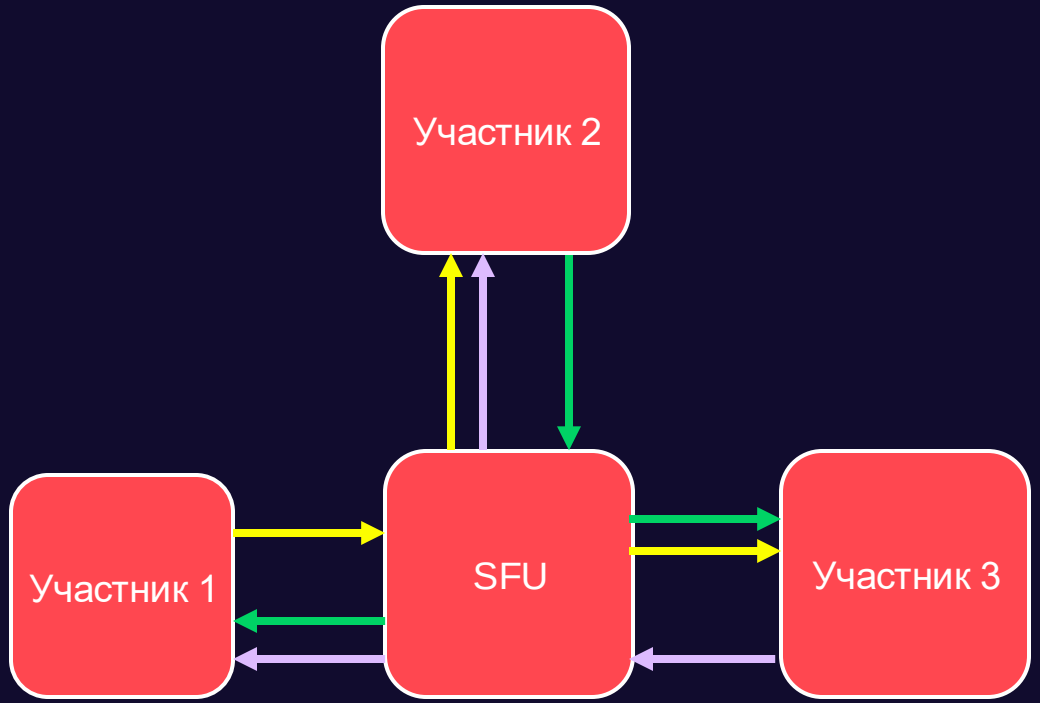
ИЛИ

Адаптивное качество!

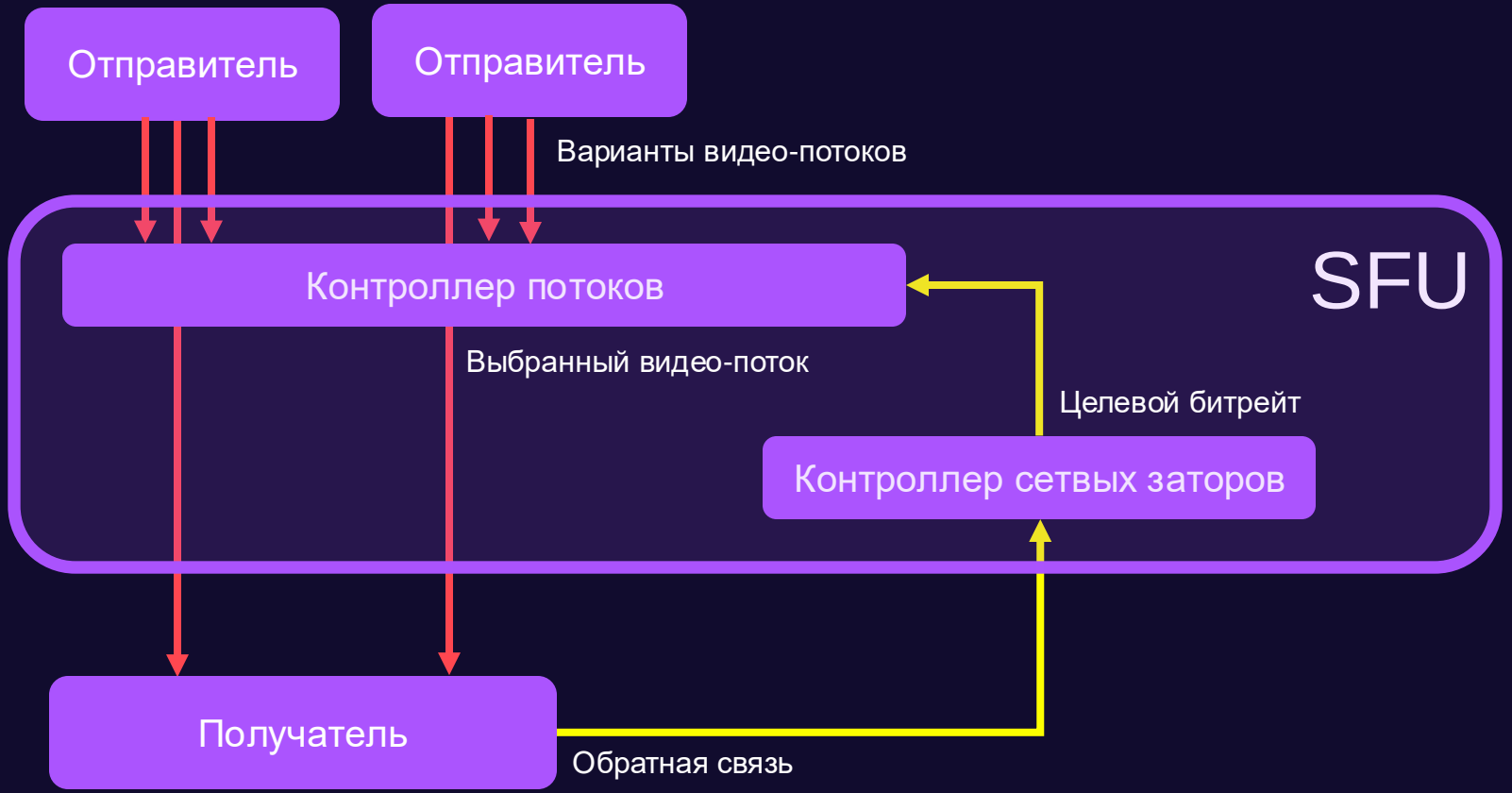


Selective forwarding unit

Основной серверный компонент, работающий с медиа



Архитектура решения



1. Архитектура решения
2. **Способы достижения вариативности битрейта**
3. Оценка полосы пропускания
4. Распределение полосы

Способы достижения вариативности битрейта

1. **Серверное транскодирование** в несколько разрешений

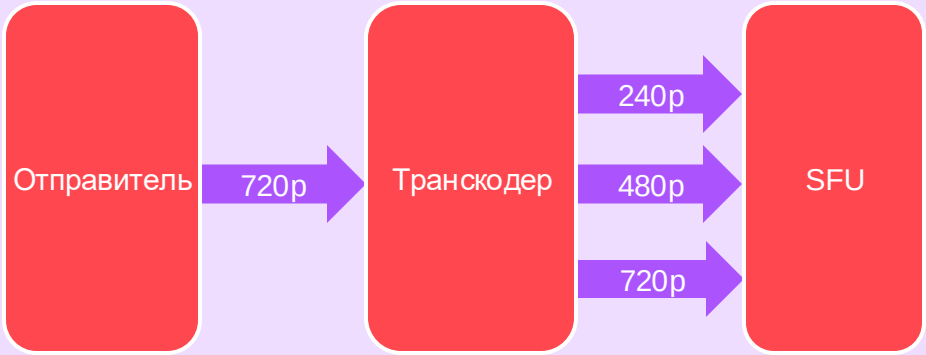
Транскодирование

Преимущества:

- Наилучшая утилизация исходящего канала пользователя

Недостатки:

- Транскодирование – ресурсоёмкая задача
- Увеличиние задержки



Способы достижения вариативности битрейта

1. Серверное транскодирование в несколько разрешений
2. **Simulcast**

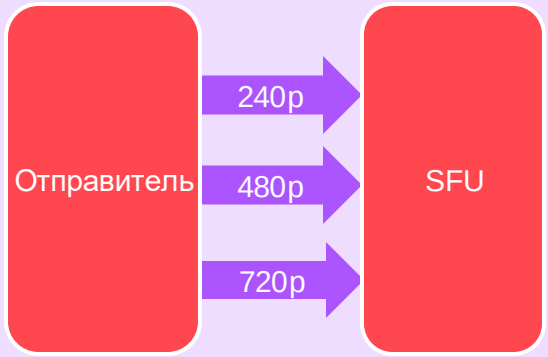
Simulcast

Преимущества:

- Экономия серверных ресурсов
- Широкая поддержка среди браузеров

Недостатки:

- Большой оверхед на кодирование и передачу по сети независимых потоков



Способы достижения вариативности битрейта

1. Серверное транскодирование в несколько разрешений
2. Simulcast
3. **Scalable Video Coding**

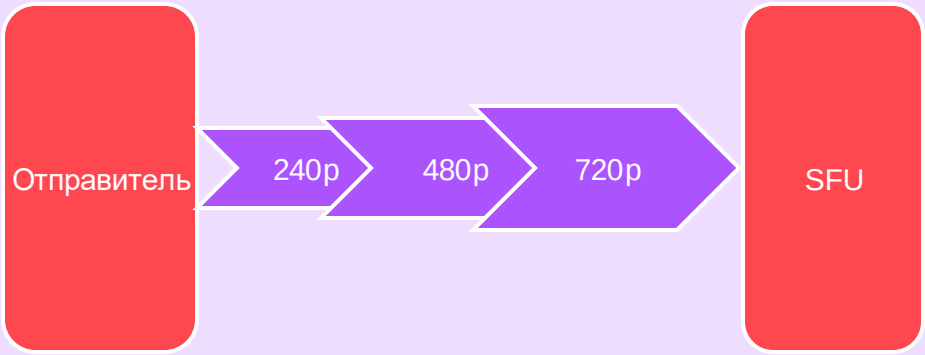
Scalable Video Coding

Преимущества:

- Экономия серверных ресурсов

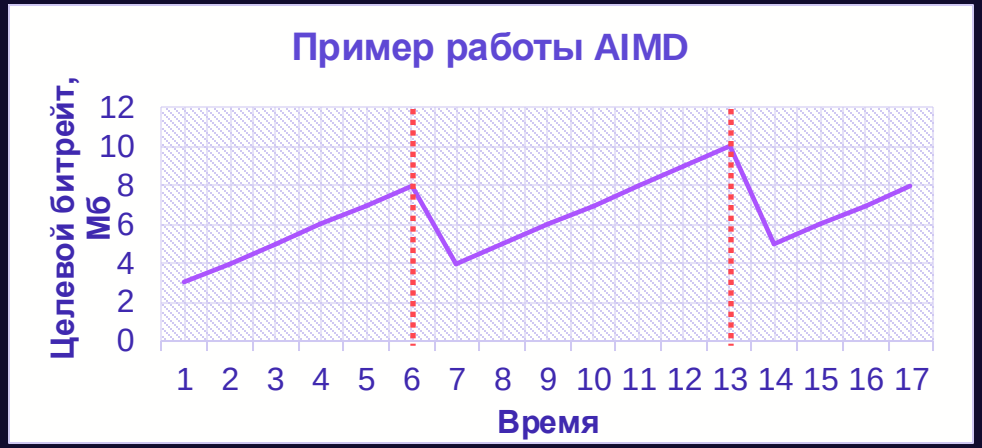
Недостатки:

- Узкая поддержка среди браузеров



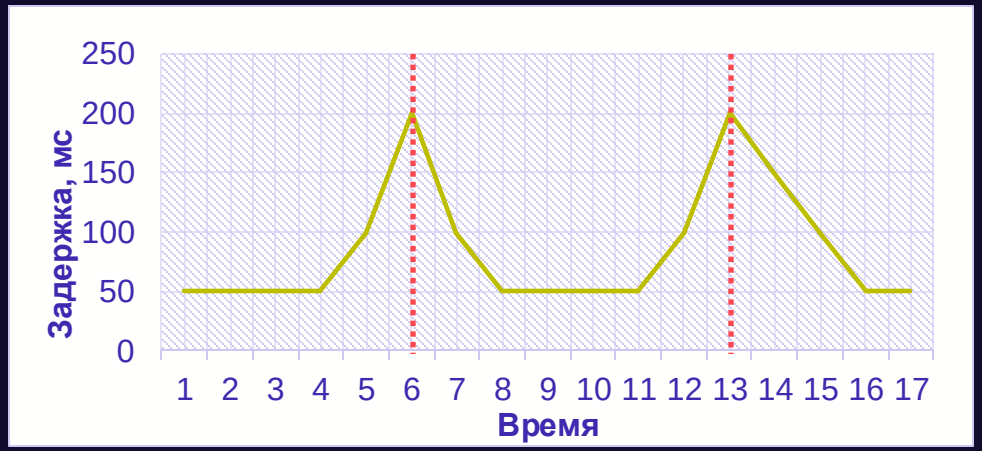
1. Архитектура решения
2. Способы достижения вариативности битрейта
3. **Оценка полосы пропускания**
4. Распределение полосы

Почему нужны media-friendly алгоритмы



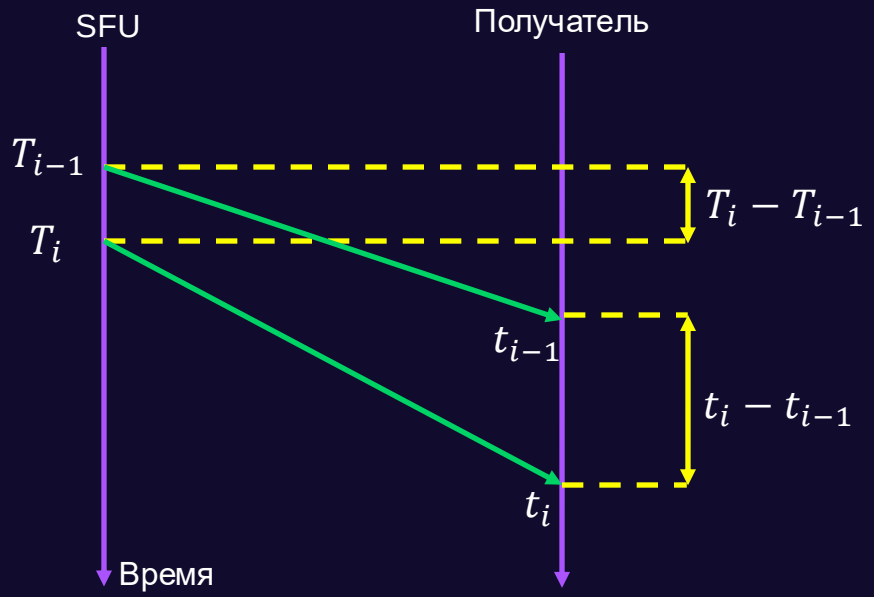
c – целевой битрейт

$$c\,(t+1) = \begin{cases} c(t) + a, & \text{нет затора} \\ c(t) \div b, & \text{есть затор} \end{cases}$$



Потерям предшествует накоплние задержки

Измерение производной задержки



$$d_i = (t_i - t_{i-1}) - (T_i - T_{i-1})$$

Carlucci, Gaetano & De Cicco, Luca & Holmer, Stefan & Mascolo, Saverio. (2016). Analysis and design of the google congestion control for web real-time communication (WebRTC). 1-12. 10.1145/2910017.2910605.

Избавляемся от шума в измерениях

Воспользуемся фильтром Калмана для $d_i = (t_i - t_{i-1}) - (T_i - T_{i-1})$

K_i – степень уверенности в d_i от 0 до 1 (Kalman gain)

$$m_i = m_{i-1} + K_i(d_i - m_{i-1})$$

$$K_i = 0 \Rightarrow m_i = m_{i-1}$$

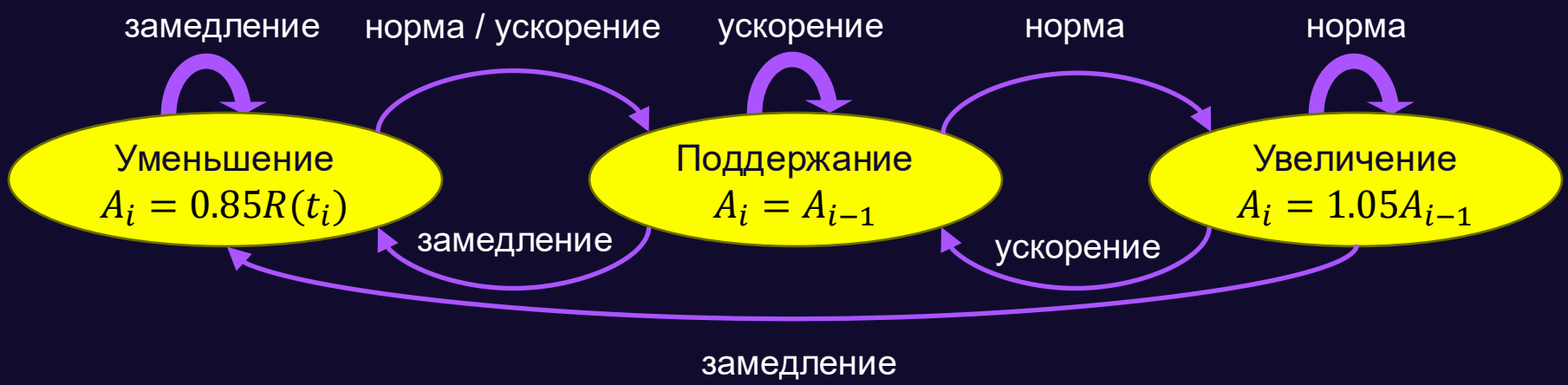
$$K_i = \frac{1}{2} \Rightarrow m_i = \frac{m_{i-1} + d_i}{2}$$

$$K_i = 1 \Rightarrow m_i = d_i$$

Carlucci, Gaetano & De Cicco, Luca & Holmer, Stefan & Mascolo, Saverio. (2016). Analysis and design of the google congestion control for web real-time communication (WebRTC). 1-12. 10.1145/2910017.2910605.

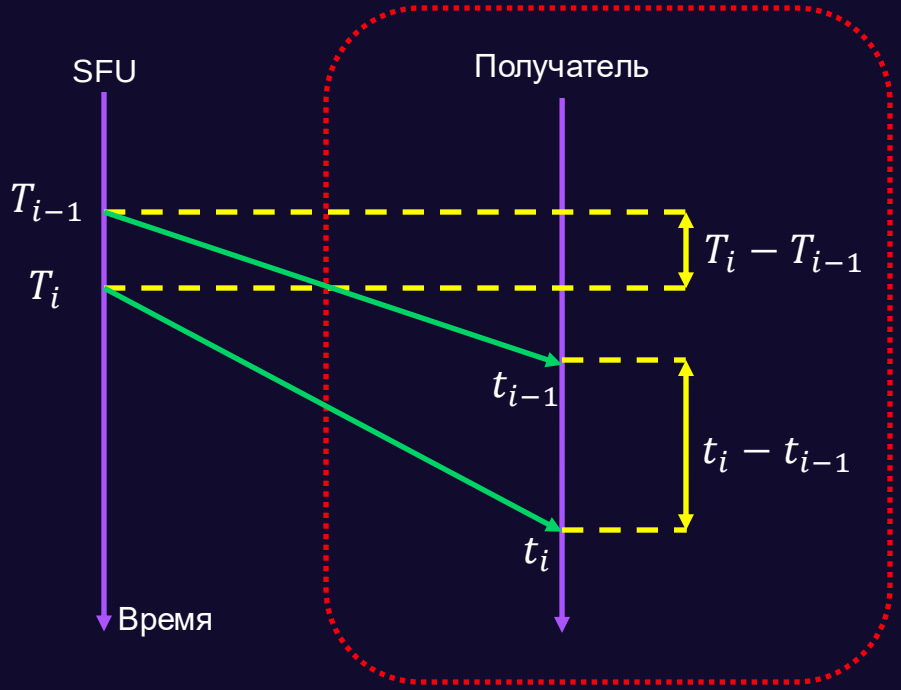
Управляем битрейтом

Сигнал	Ускорение	Норма	Замедление
Условие	$m_i < -\gamma$	$-\gamma \leq m_i \leq \gamma$	$m_i > \gamma$

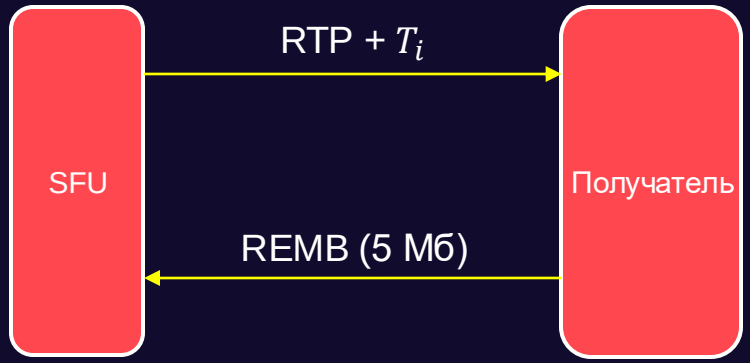


Carlucci, Gaetano & De Cicco, Luca & Holmer, Stefan & Mascolo, Saverio. (2016). Analysis and design of the google congestion control for web real-time communication (WebRTC). 1-12. 10.1145/2910017.2910605.

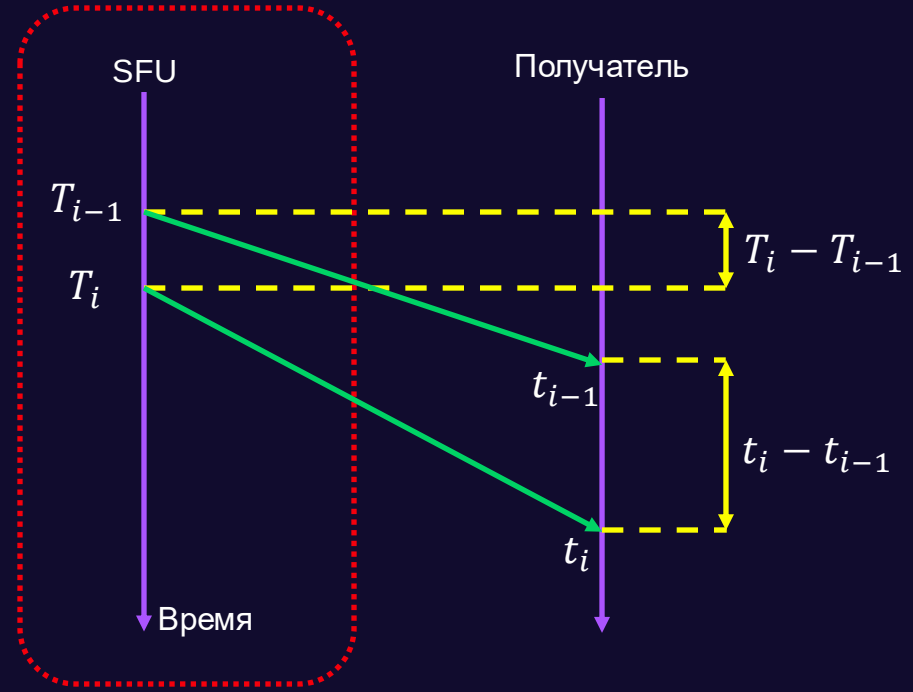
Вариант 1. Реализация на стороне получателя



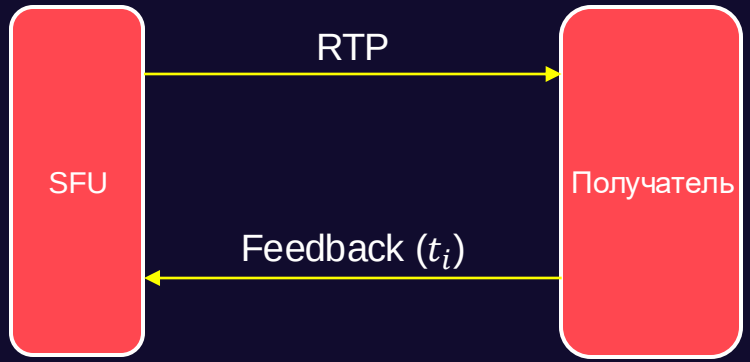
Анализ задержек на стороне получателя



Вариант 2. Реализация на стороне SFU



Анализ задержек на стороне SFU

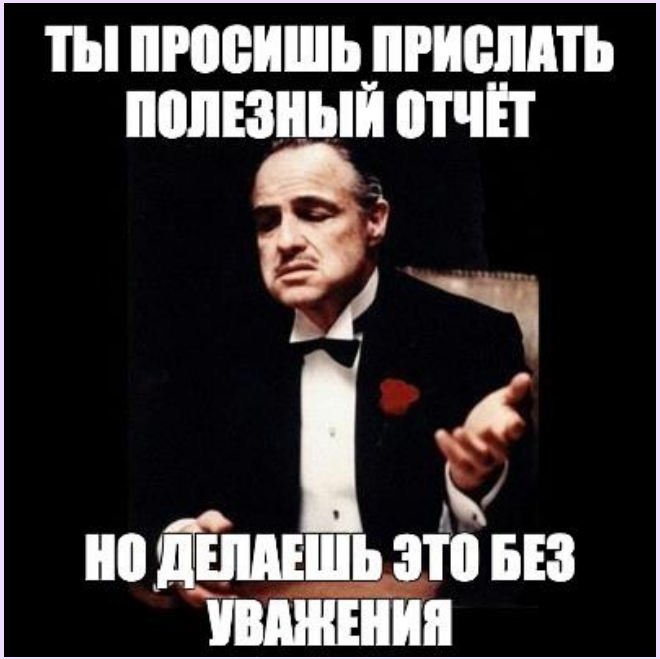


Congestion control feedback

1. Receiver and Sender reports

Receiver and Sender reports

- Широко поддержаны
- Информации недостаточно, чтобы получить точную оценки сети.

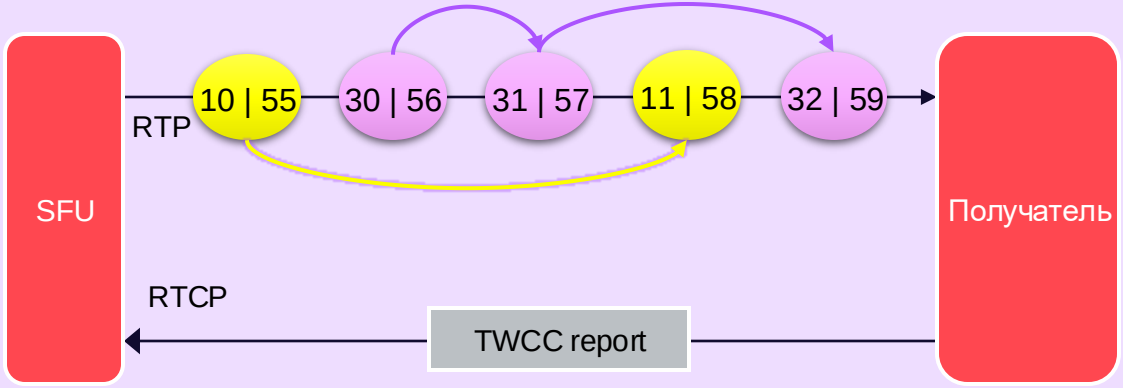


Congestion control feedback

1. Receiver and Sender reports
2. **Transport-wide Congestion Control**

Transport-wide congestion control

- Вводит номера пакетов на транспортном уровне
- Содержит информацию по каждому пакету, включая тайминги получения

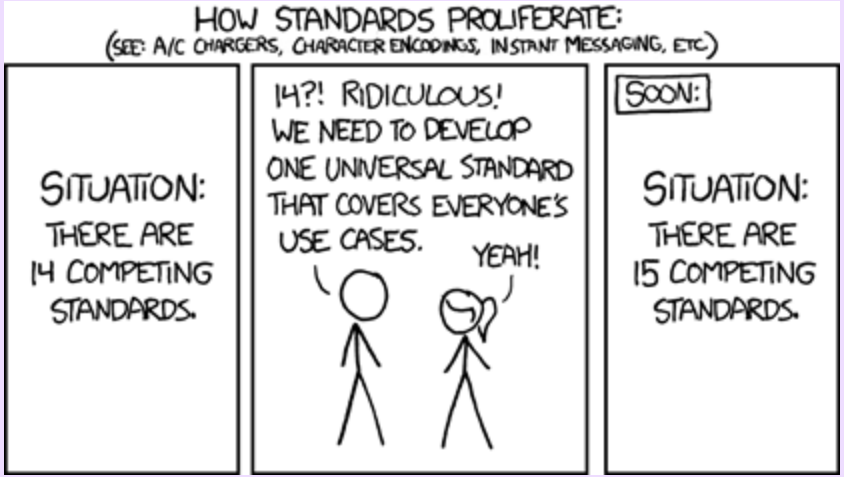


Congestion control feedback

1. Receiver and Sender reports
2. Transport-wide Congestion Control
3. **RFC 8888**

RFC 8888

- Красивый номер RFC
- Поддержан в libwebrtc в апреле 2024 г
- Совместим со всеми популярными алгоритмами
- Обратная связь на уровне RTP потоков
- Не увеличивает размер RTP пакетов



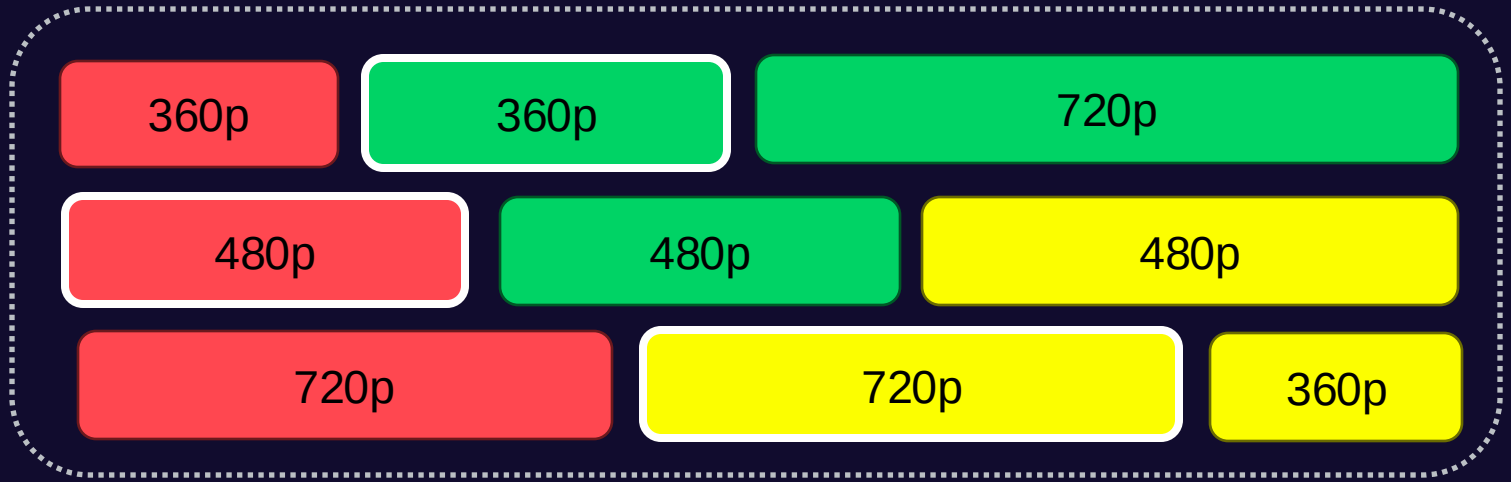
xkcd.com/927

1. Архитектура решения
2. Способы достижения вариативности битрейта
3. Оценка полосы пропускания
4. **Распределение полосы**

Распределение полосы пропускания

Доступный битрейт, Мб

Варианты потоков



Задача о рюкзаке с множественным выбором

Вход:

N – множество id допустимых вариантов потоков

N_k – набор id вариантов от k – го участника

$$N = \{1, \dots, n\} = \bigcup_{k=1}^{r} N_k$$

c – оценка пропускной способности сети

w_j – битрейт варианта j

p_j – "полезность" варианта j

Выход:

$$x_j = \begin{cases} 1, & \text{используем поток } j \\ 0, & \text{не используем поток } j \end{cases}$$

Задача состоит в **максимизации суммарной полезности ответа**:

$$z = \sum_{j=1}^{n} p_j x_j$$

При выполнении условий:

- Сумма битрейтов выбранных вариантов не превышает оценку пропускной способности

$$\sum_{j=1}^{n} w_j x_j \le c$$

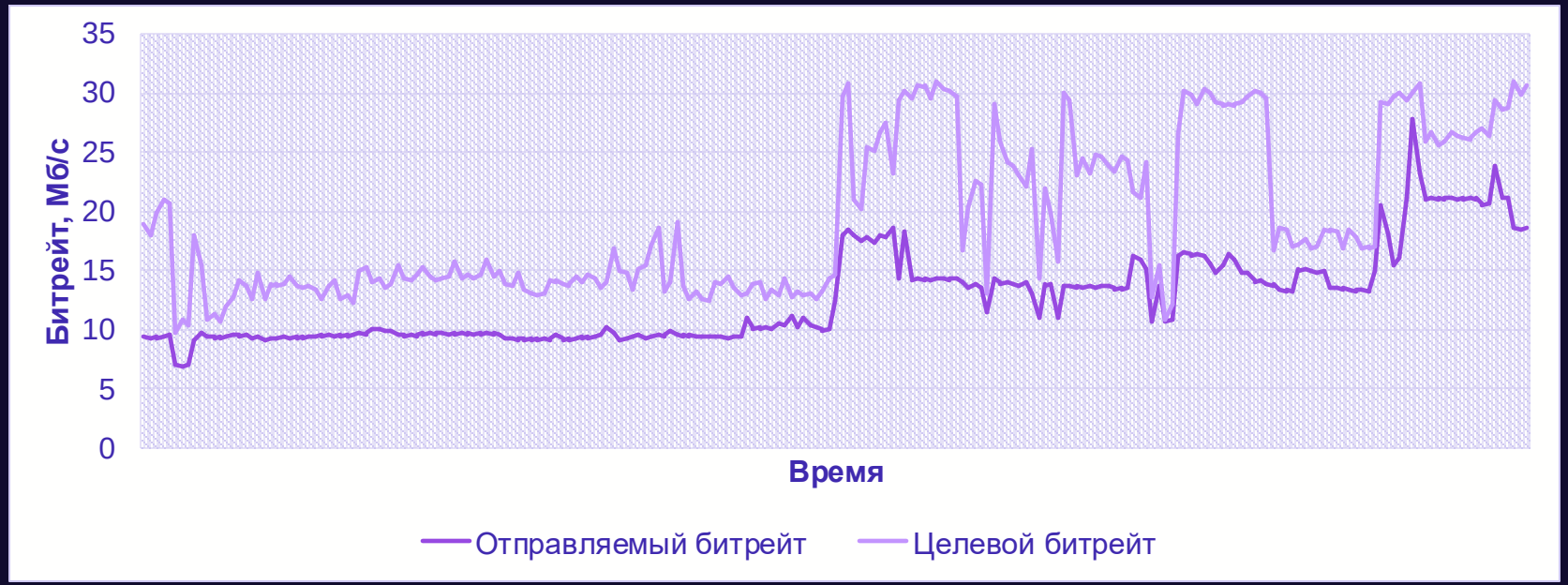
- Для каждого участника выбран ровно 1 вариант:

$$\sum_{j \in N_k} x_j = 1 \, , k = 1 \ \dots \ r$$

Kellerer, Hans & Pferschy, Ulrich & Pisinger, David. (2004). Knapsack Problems. 10.1007/978-3-540-24777-7.

Эвристика

- Максимизирует худшее воспринимаемое качество на экране
- Ограничивает качество видео размерами контейнера
- Удерживает битрейт от 70% до 90% от целевого



Итоги

- Вспомнили методы достижения вариативности битрейта
- Узнали как работают media-friendly алгоритмы оценки полосы пропускания
- Разобрались в ворохе RFC для обратной связи RTCP
- Придумали эвристику для решения задачи распределения полосы

Спасибо

Александр Алексеев

Разработчик
Yandex Infrastracture
alekseev-dev@yandex-team.ru