Управление качеством сервиса в сетях

Применение методов машинного обучения для обеспечения требуемого качества сервиса

Степанов Евгений Павлович,

к.ф.-м.н., м.н.с.


NEXT!

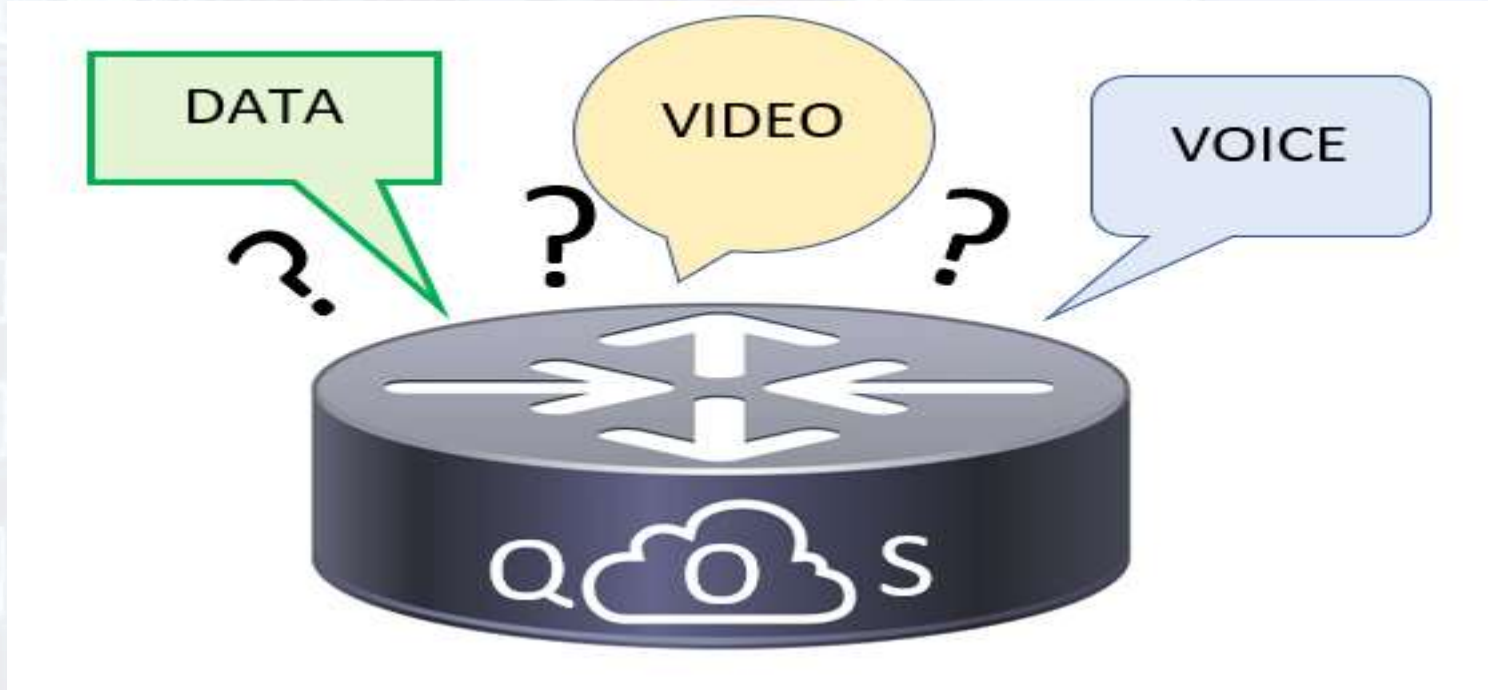
Best effort

- скорость соединения с Интернет в самом лучшем случае В Мбит/с.

- если не нравится, переключайся на другого провайдера

QoS

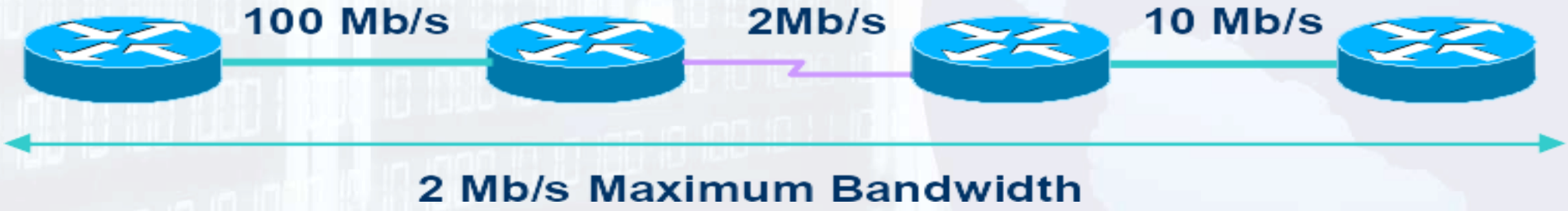
- некоторые гарантии по пропускной способности, задержке…



Метрики качества

Пропускная способность

Количество данных, которые может быть передано в единицу времени [байты в секунду]



Метрики качества

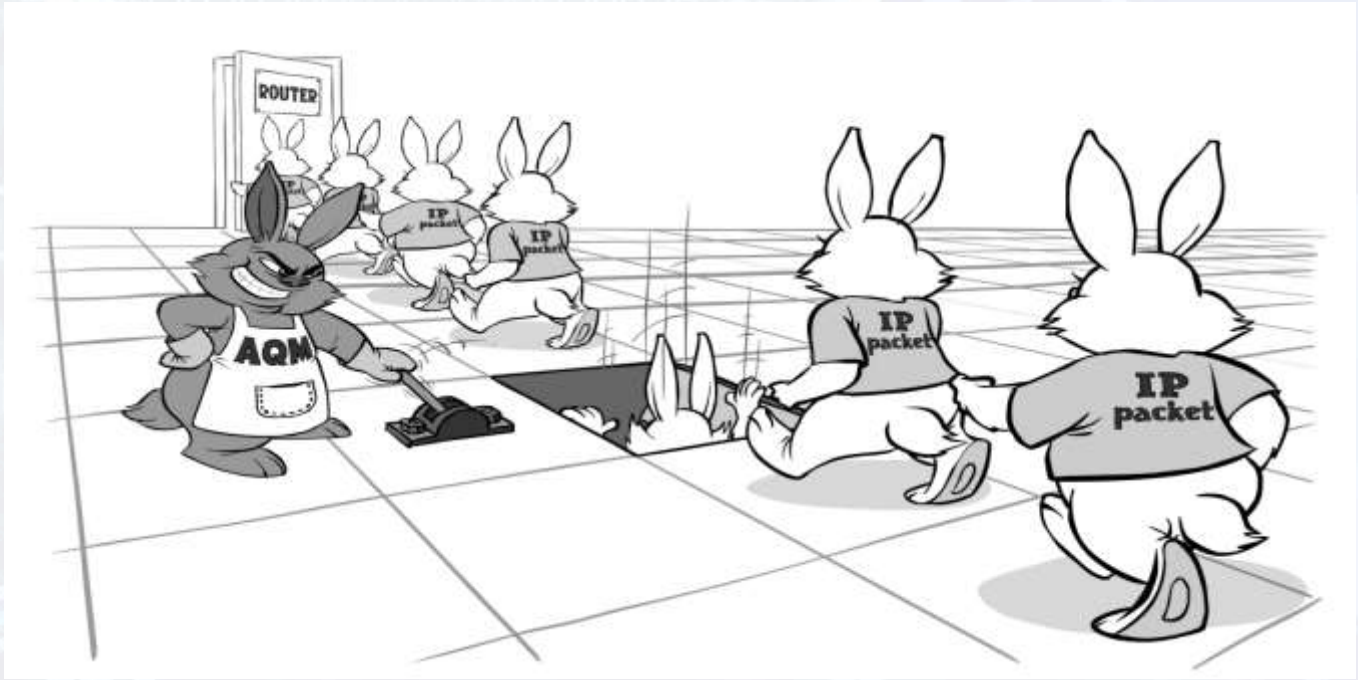
Задержка

Время передачи единицы данных по направлению от отправителя к получателю [секунды]

Метрики качества

Уровень потерь

Доля пакетов, которые были отправлены, но не были досталены получателю [проценты]



Почему QoS становится всё более важной проблемой?

Почему QoS становится всё более важной проблемой?

Почему QoS становится всё более важной проблемой?

Почему QoS становится всё более важной проблемой?

Почему QoS становится всё более важной проблемой?

How to shorten OTT TV Lag - Ops Perspective

IPTV OTT Lag 4/5 – QoS and Ops Analytics

33% Abandonment rate immediately with poor quality *Akamai

50% Abandonment rate with 5 startup sec delay *Akamai

66% Consumers are frustrated with video buffering *Research Now

21% Severely irritated when a video didn't load quickly *Research Now

38% Users heart rate increase when buffering *Ericsson

Very High Demanding Customers, Public Market Rating and Posts, Low Barriers to Churn

Path

Customer Care:

OTT Users won´t call u as before, they do public complains in Public Stores, blogs, foruns and social media. **Be proactive in capturing the Voice of Customer** and react as fast as possible in your devops cycle.

Tech Level :

Measure, Measure, Measure Experience, Engagement and Audience in Real Time to automaticaly prevent problems and monetize!

We´ve done the path in IPTV. In OTT it´s harder as you don´t own the entire chain.

Local Operators can do it better because they own the network and have direct access to the customers

David Murargi, Portugal Telecom - 09.11.2016

Управление качеством сервиса

Как управлять качеством сервиса?

Подходы:

1. ***Управление перегрузкой***
   - Современные протоколы управления перегрузкой TCP
2. ***Демультиплексирование/мультиплексирование***
   - Многопоточные транспортные протоколы
   - Маршрутизация на уровне интернет провайдеров
   - Network Coding
3. ***Сегментация***
   - TCP Proxy
4. ***Балансировка***
   - Балансировка нагрузки и управление трафиком
5. ***Преобразование сообщений***
   - FEC
   - Сжатие

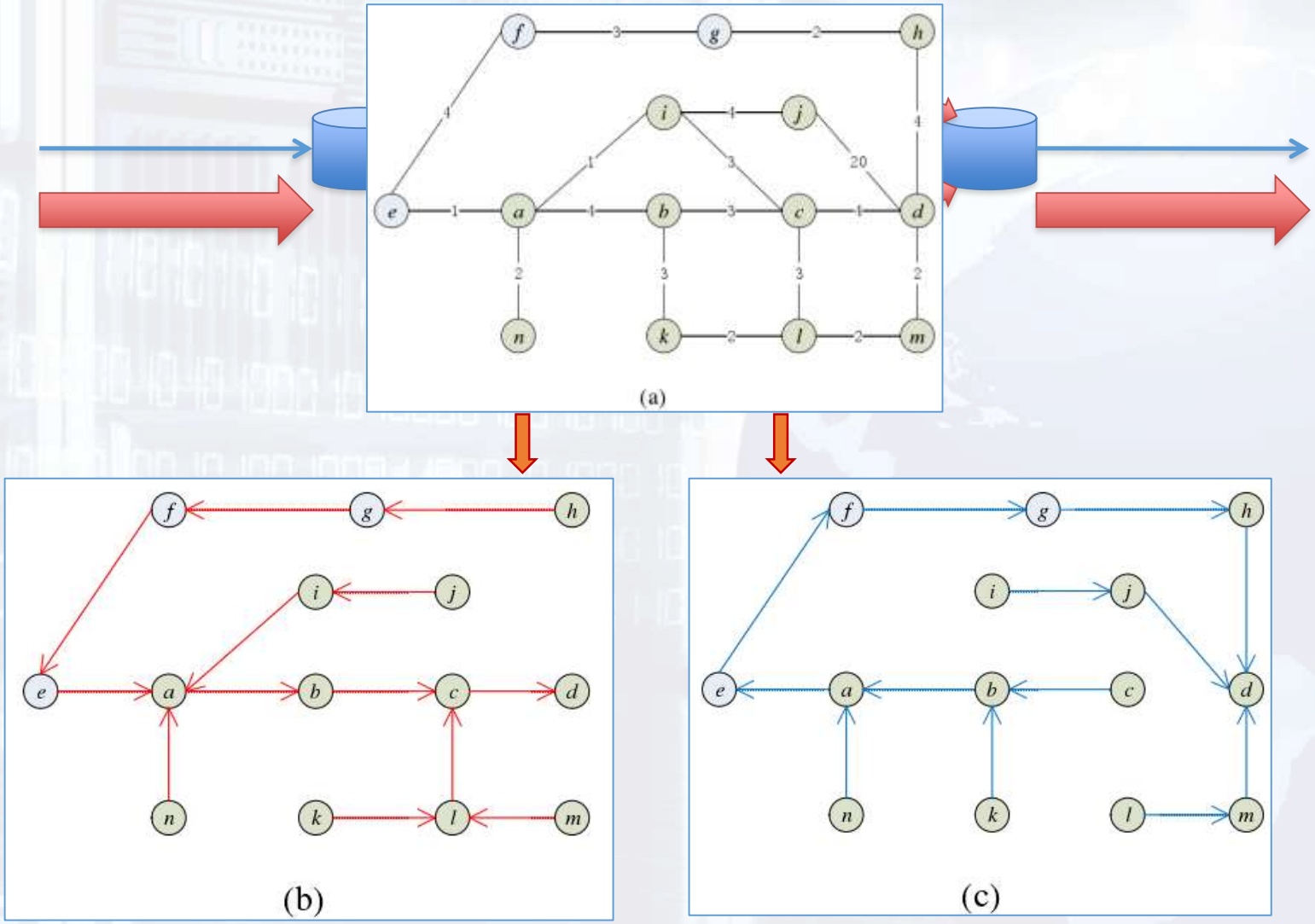
Модели оценки качества сервиса:

- Активные и пассивные измерения
- Сетевое исчисление: математический подход к качеству сервиса
- NS3: моделирование поведения сети с высокой точностью

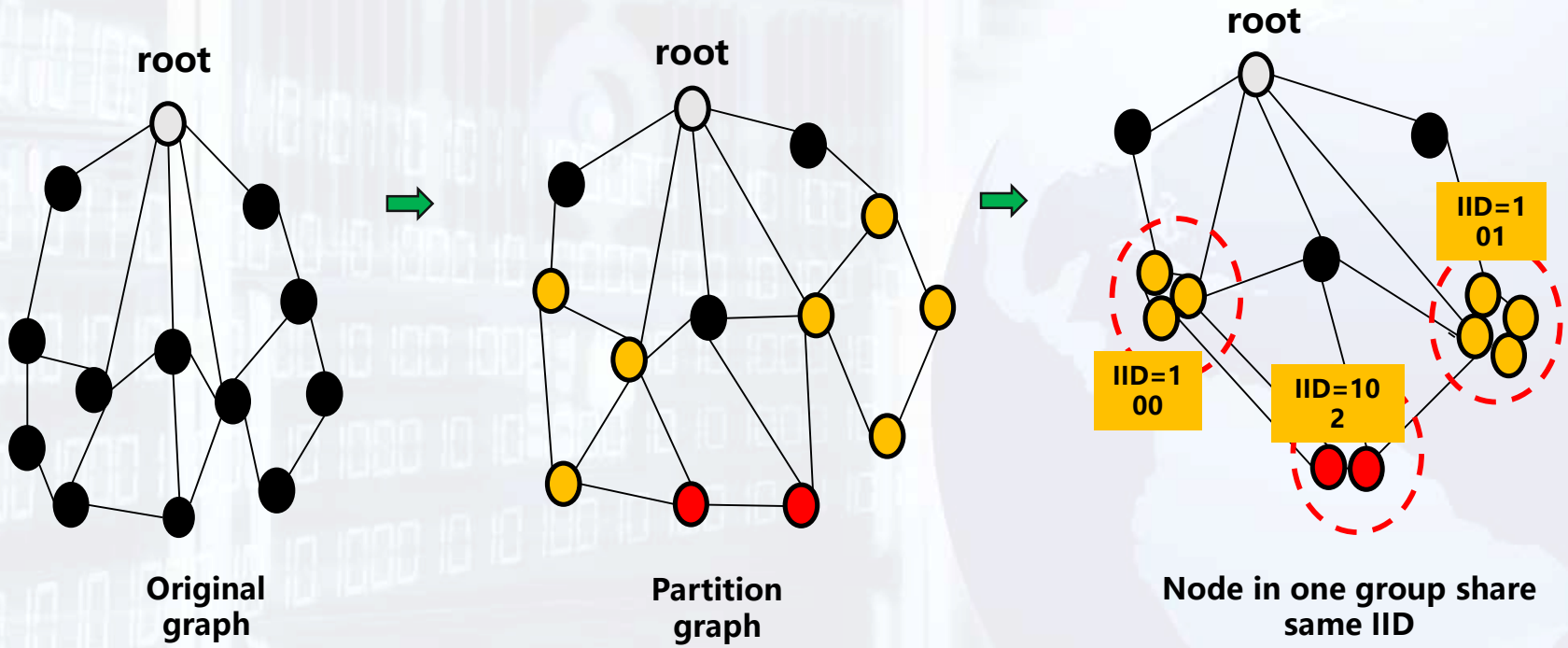
Примеры:

- Управление сетевыми ресурсами в Центрах Обработки Данных
- Обеспечение качества сервиса в сетях доставки контента
- Пропускная способность по требованию

Балансировка нагрузки



Балансировка нагрузки / Indirect Next Hop

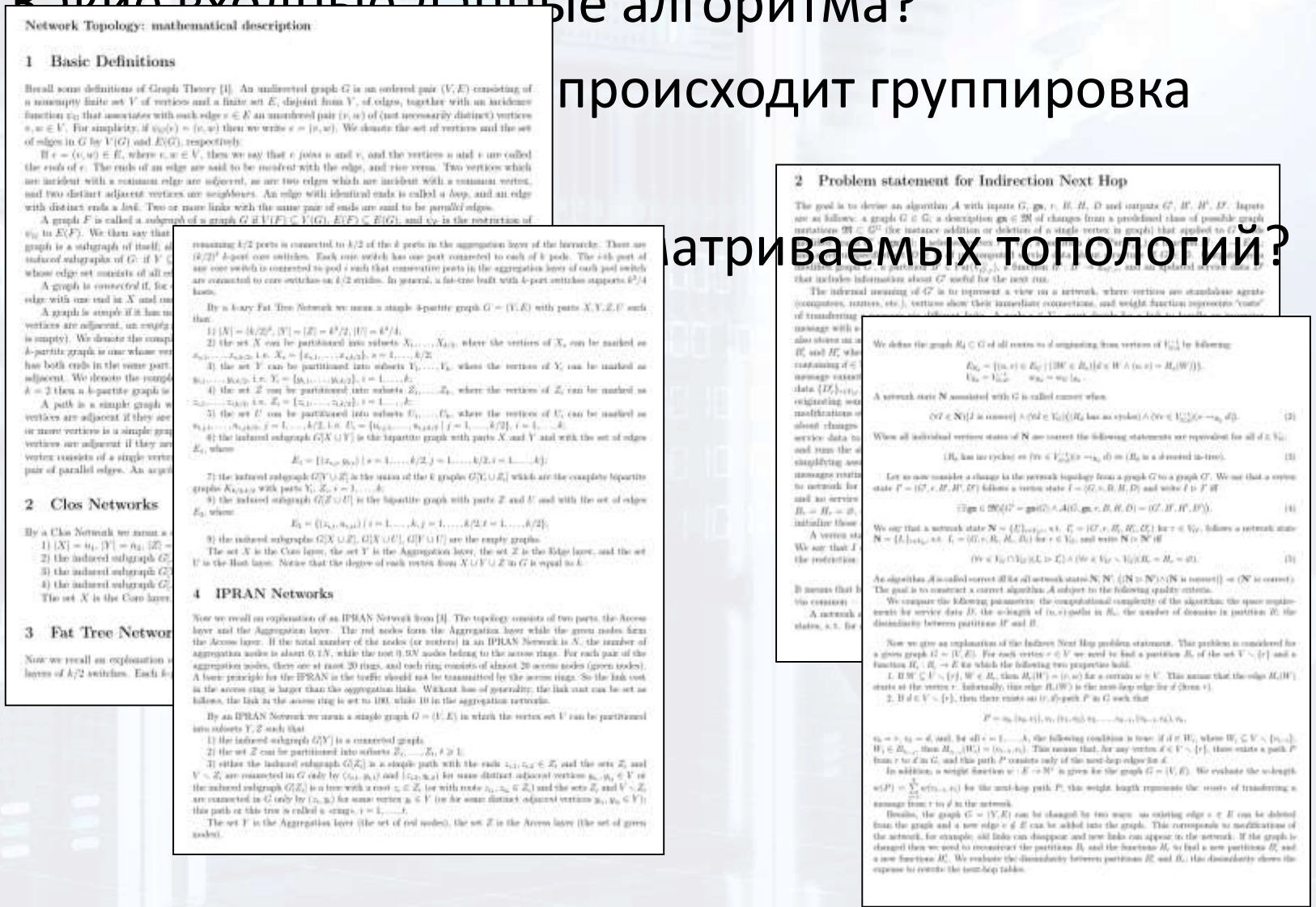


Этапы исследования

- Математическая постановка задачи
- Проведение обзора схожих работ
- Разработка алгоритма решения поставленной задачи
- Реализация разработанного алгоритма
- Проведение экспериментального исследования

Постановка задачи

- Какие входные данные алгоритма?
- [illegible] происходит группировка
- [illegible] атриваемых топологий?



Network Topology: mathematical description

1 Basic Definitions

2 Clos Networks

3 Fat Tree Networ

4 IPRAN Networks

2 Problem statement for Indirection Next Hop

Обзор схожих работ

- The problem statement si
- The applicability to the gi
- The algorithm complexity
- The accuracy of the soluti

The correspondence of articles to selected overview criteria

Article	The problem statement similarity	The applicability to the given topologies	The algorithm complexity	The accuracy of the solution
Ramalingam G., Reps T., On the computational complexity of dynamic graph problems (1996) [6]	+	+	+	+
Frigioni D., Marchetti-Spaccamela A., Nanni U. Fully dinamic output bounded single sourse shortest path problem (1996) [7]	+	+	+	+
Bin Xiao, Jiannong Cao, Zili Shao, Edwin H.-M. Sha. An efficient algorithm for dynamic shortest path tree update in network routing (2007) [8]	+	+	+	+
Khuller S., Raghavachari B., Young N., Balancing minimum spanning trees and shortest-path trees (1995) [9]	−	+	+	+
Nardelli E., Proietti G., Widmayer P., How to swap a failing edge of a single source shortest paths tree (1999) [4]	+	+	+	−
Nardelli E., Proietti G., Widmayer P., Swapping a failing edge of a single source shortest paths tree is good and fast (2003) [5]	+	+	+	−
Di Salvo A., Proietti G. Swapping a failing edge of a shortest paths tree by minimizing the average stretch factor (2007) [10]	+	+	+	−
Spira P. M., Pan A., On finding and updating spanning trees and shortest paths (1975) [11]	+	+	+	−
Roditty L., Zwick U., On dynamic shortest paths problems (2004) [12]	+	+	−	+
Williams V. V., Faster replacement paths (2011) [13]	−	+	−	+
Guo Y., Qin Z., Chang Y., A novel hybrid algorithm for the dynamic shortest path problem (2010) [14]	+	+	+	−
Cong J., Kahng A. B., Robins G., Sarrafzadeh M., Wong C. K., Provably good performance-driven global routing (1992) [15]	−	+	+	−
Eppstein D., Italiano G. F., Tamassia R., Tarjan R. E., Westbrook J., Yung M., Maintenance of a minimum spanning forest in a dynamic plane graph (1992) [16]	−	+	+	−
Nagamochi H., Ibaraki T., A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph (1992) [17]	−	+	+	+

Разработка алгоритма

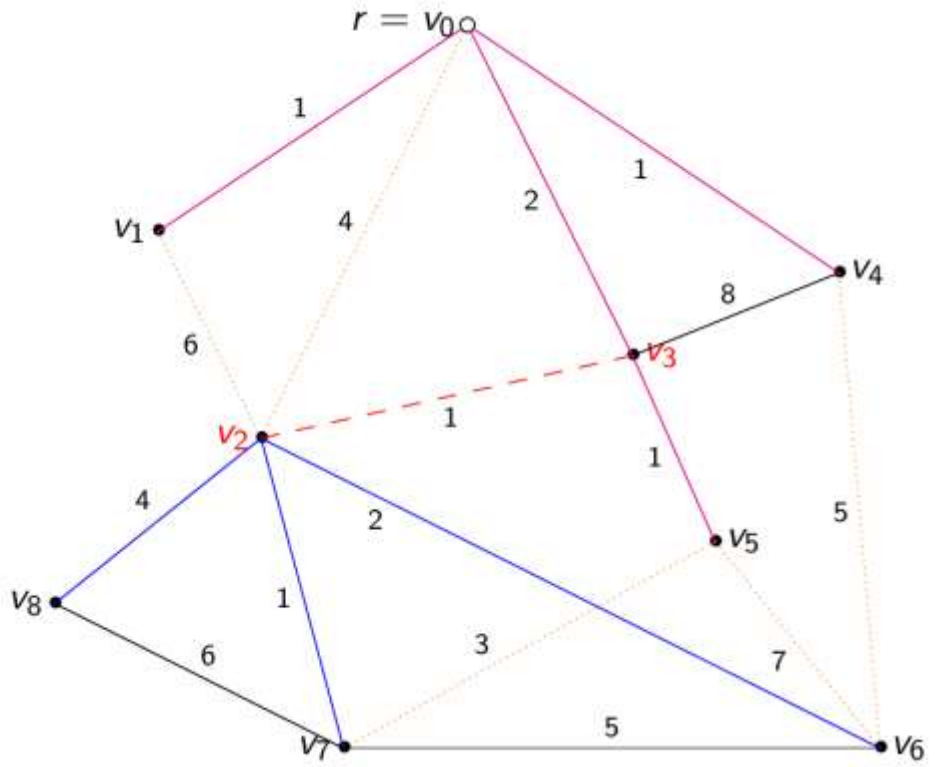
- Swap-edge based algorithm

$e = (v_3, v_2)$ — disconnected edge

S_{v_2} — 'lower' e-subtree

$S^{(v_3,v_2)}$ — 'upper' e-subtree

$C_{(v_3,v_2)}$ — set of all swap edges for e



- Dynamic algorithm

Реализация алгоритма

Структуры данных

1. Матрица смежности
2. Список смежности (список)
3. Список смежности (хэш-таблица)
4. Список смежности (дерево поиска - будут приведены значения для AVL, но можно использовать и любое другое)

Структура	Место	get e	add e	del e	add v	del v
Mc	O(V^2) (value/8 if no weight)	O(1)	O(1)	O(1)	O(V) * max(O(array copy), O(array alloc))	O(1) if do not care about space, O(V) * max(O(array copy), O(array alloc)) if reallocating
Cc(c)	O(E+V)	O(V)	O(1)	O(V)	O(1)	O(1)
Cc(x)	O(E+V) (const worse than list)	O(1) * O(hash)	O(1) * O(hash)	O(1) * O(hash)	O(1)	O(1)
Cc(д)	O(E+V) (const worse than list)	O(logV)	O(logV)	O(logV)	O(1)	O(1)

Экспериментальное исследование

Эффективность: $\frac{\sum_i Time_{nh,i} - Time_{def,i}}{|V|}$ $\frac{\sum_{i,j} d_{G_{nh}}(v_i, v_j) - \sum_{i,j} d_{G_{def}} v_i, v_j}{|V|}$

Параметры эксперимента:

- Topology family: Clos or IPRAN.
- Overall number of vertices (routers) in topology graph. Possible values: 500,1000,2500,5000,6000,7000,8000,9000,10000
- Distribution of subnetworks that defines initial routing table:
  - Static: 5,10,15 subnetworks per vertex.
  - Uniform distribution U(1,20).
- Sequence of graph mutations gm:
  - Trivial sequences of length 1.
  - Sequences where after serveral removal of edge/vertex follows addition of same edge/vertex.
- Graph structure used in algorithm implementation:
  - Adjacency list with edges stored in list.
  - Adjacency list with edges stored in array.
  - Adjacency list with edges stored in set based on hash table.
  - Adjacency list with edges stored in set based on red-black tree.
- Used algorithm: Dijkstra, proposed dynamic Dijkstra, proposed swap-edge based algorithm

ML в балансировке трафика

$G(t_i) = (V(t_i), E(t_i))$

- $c_{u,v}$ - the bandwidth of the corresponding link; assume $c_{u,v} = c_{v,u}$.
- $b_{u,v}$ - the currently occupied bandwidth of the link. Note, $b_{u,v}$ and $b_{v,u}$ can differ from each other.

$PATHS(u, v) = \{(d_i, w_i)\}, \forall u, v \in V, u \neq v$

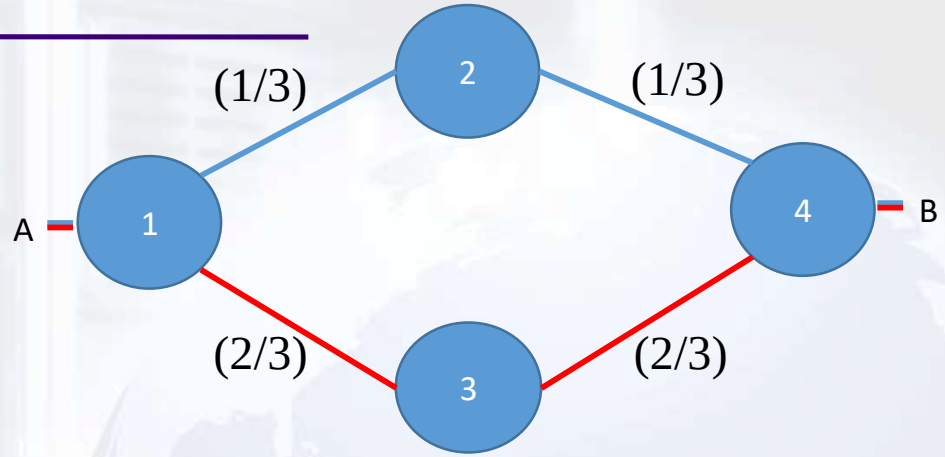
- IGP-path from vertex k to vertex v – (d, w)
- PATHS(1,4) = {(2, 2), (3,2)}

TM

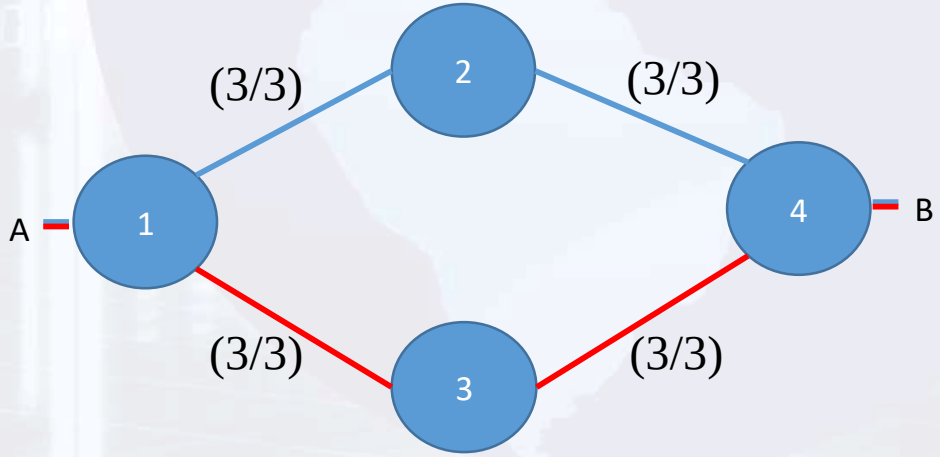
- Flow is <*fdst*, *fsrc*, *fr*>
- TM = $\{f_1 < 4,1,1 >, f_2 < 4,1,1 >, f_3 < 4,1,1 >\}$

$HASH: \quad G(t_i) \times R(t_i) \times TM(t_{i+1}) \rightarrow G(t_{i+1})$

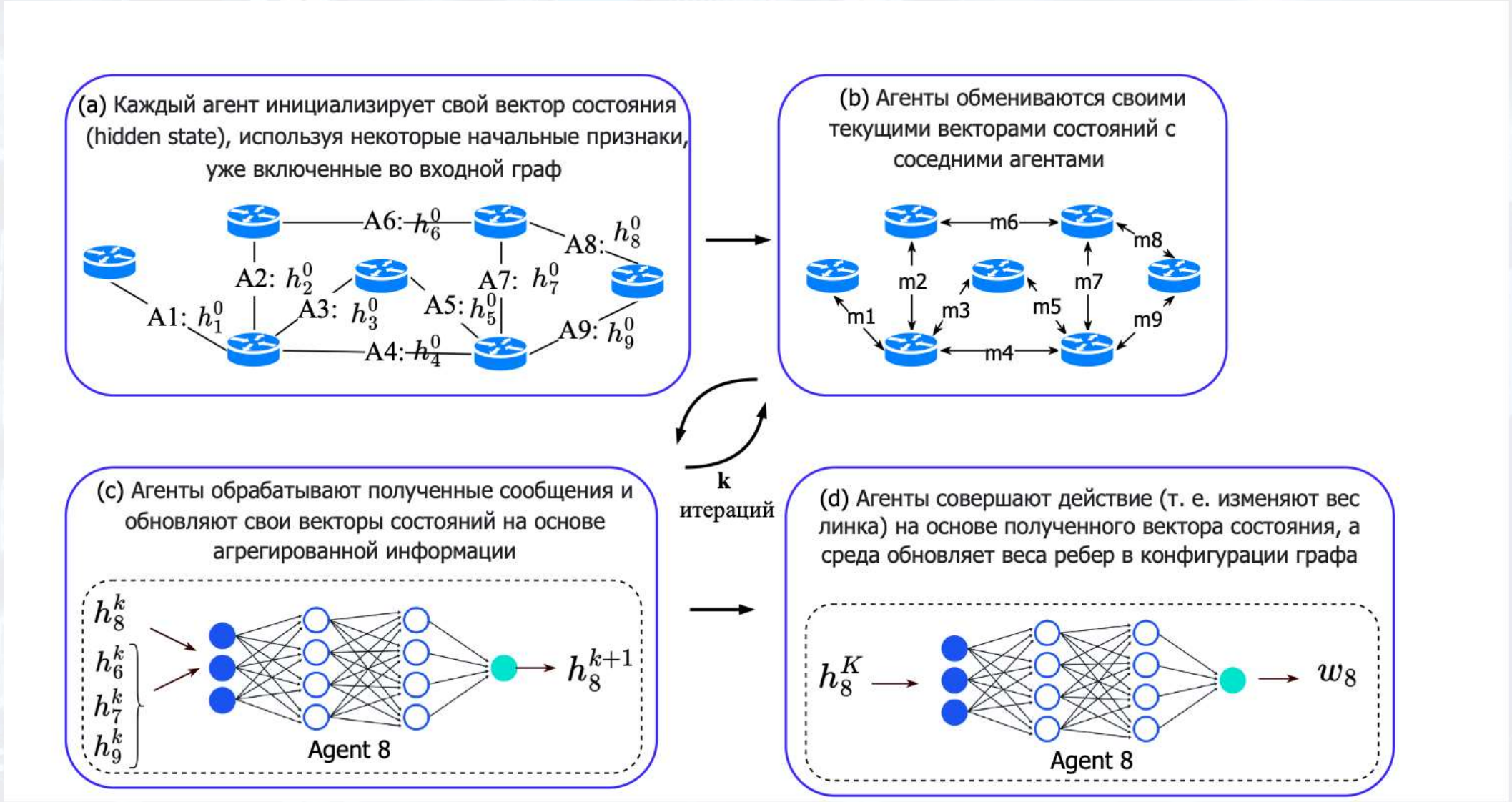
- $R(t_i)$ – *hash-weights (h-weights)*
- R = {2, 1}



$\{f_1 < 4,1,1 >, f_2 < 4,1,1 >, f_3 < 4,1,1 >\}$

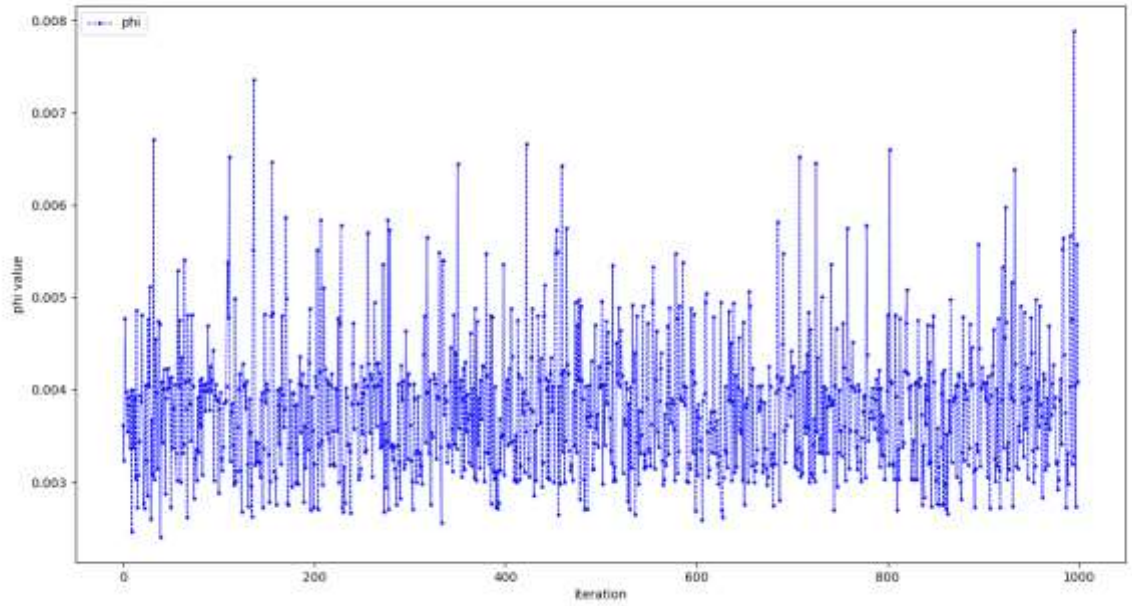


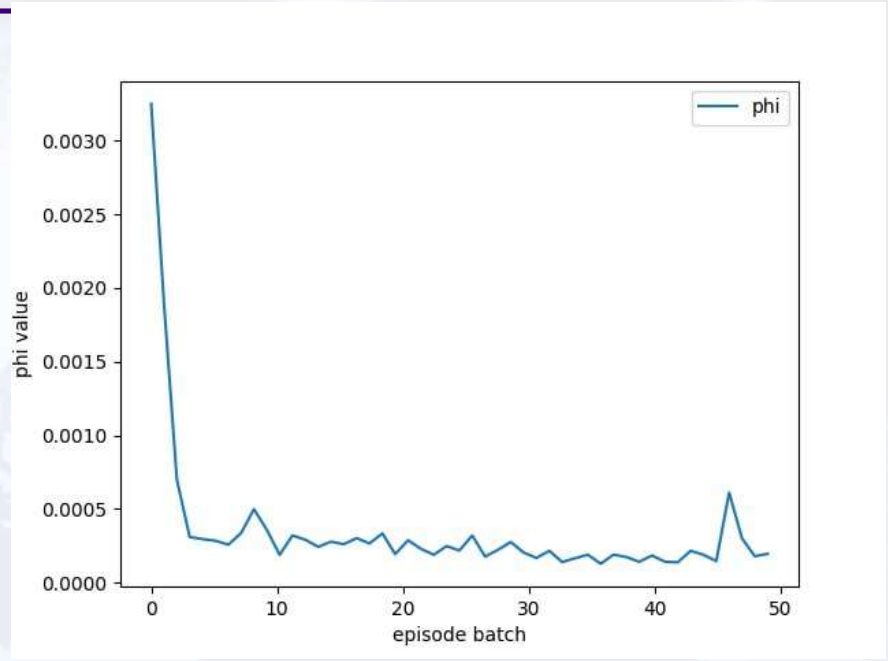
MARL-GNN подход



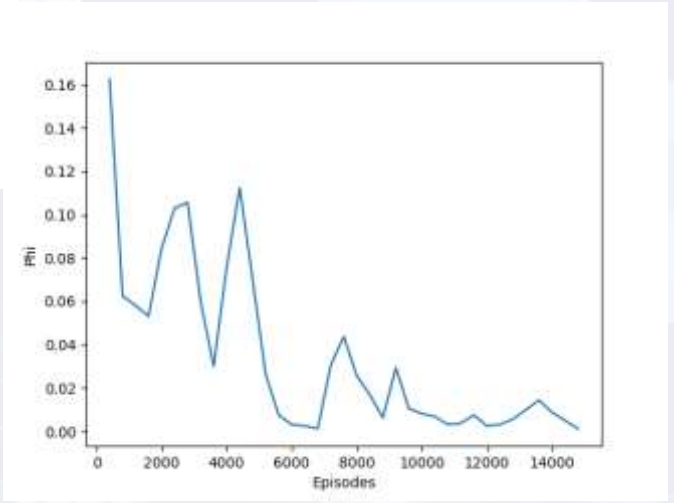
Экспериментальное исследование

Проблема с осцилляцией:



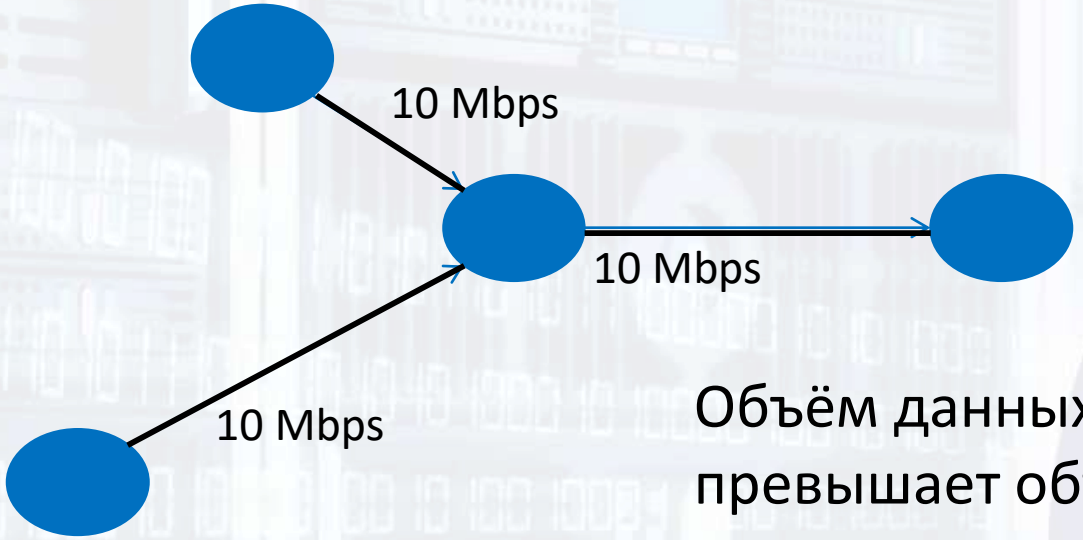


50 flows



2 flows

Перегрузка

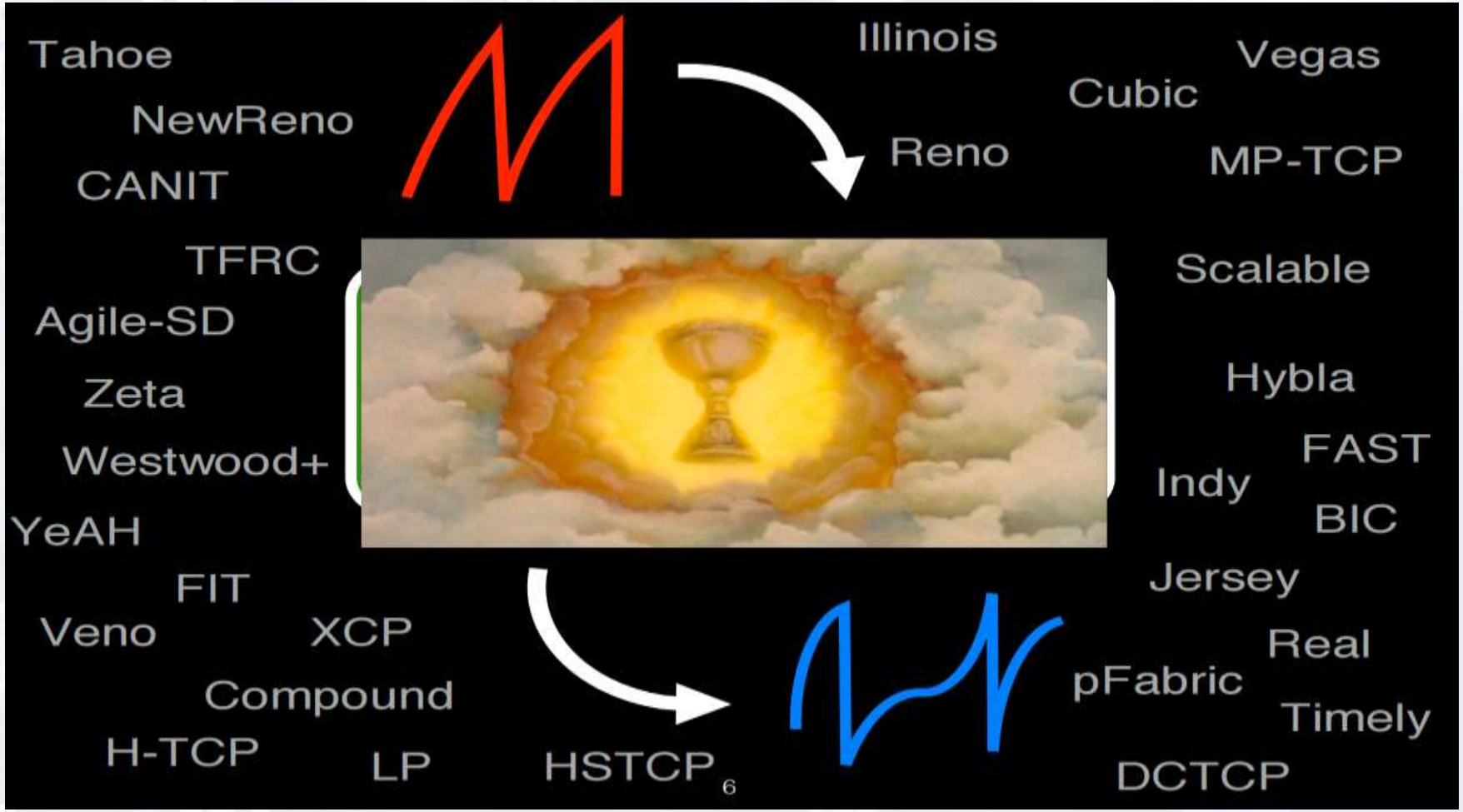


Объём данных, поступающих на коммутатор, превышает объём данных, которые он может передать

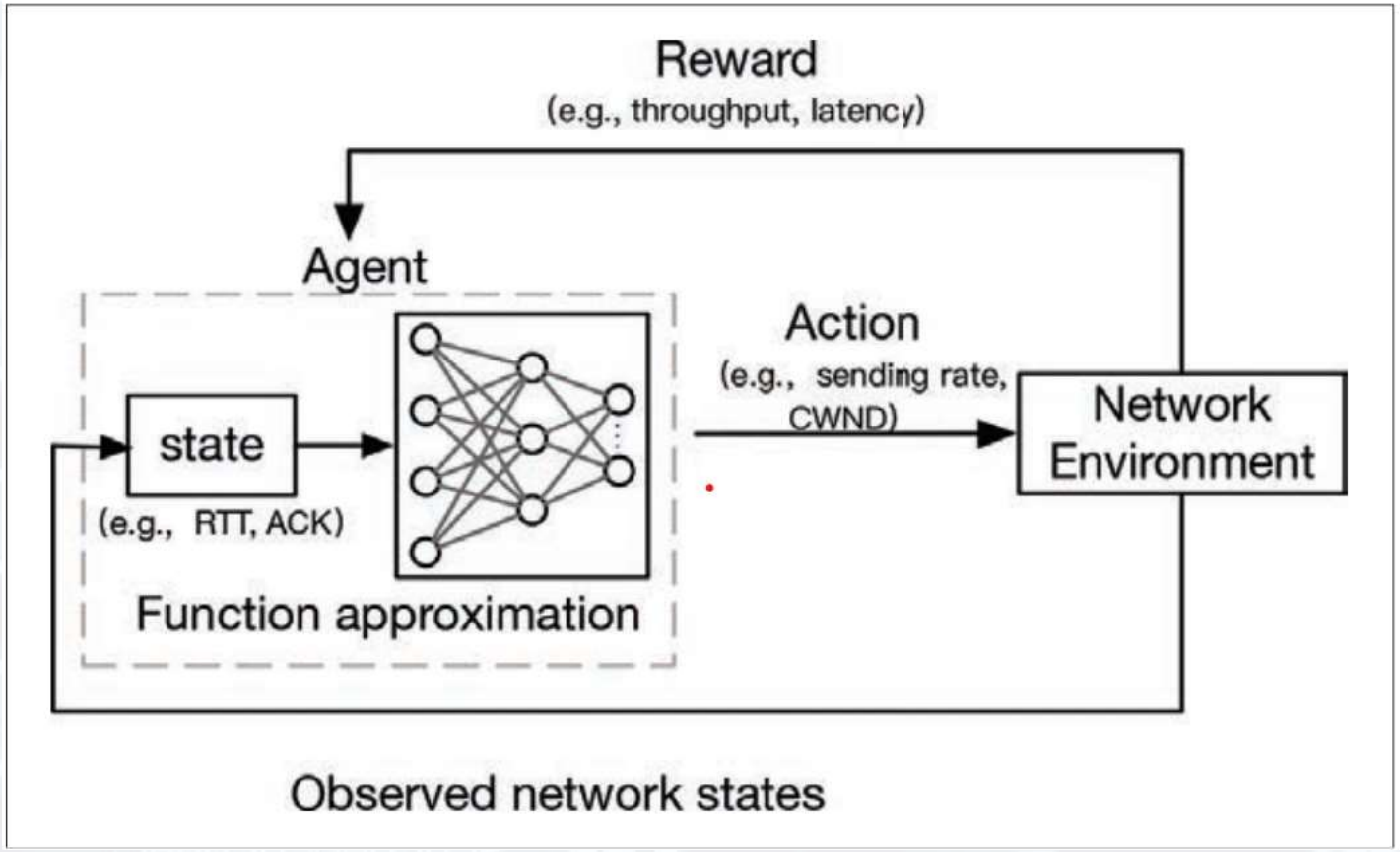
Коммутаторы буферизируют проходящий через них трафик:

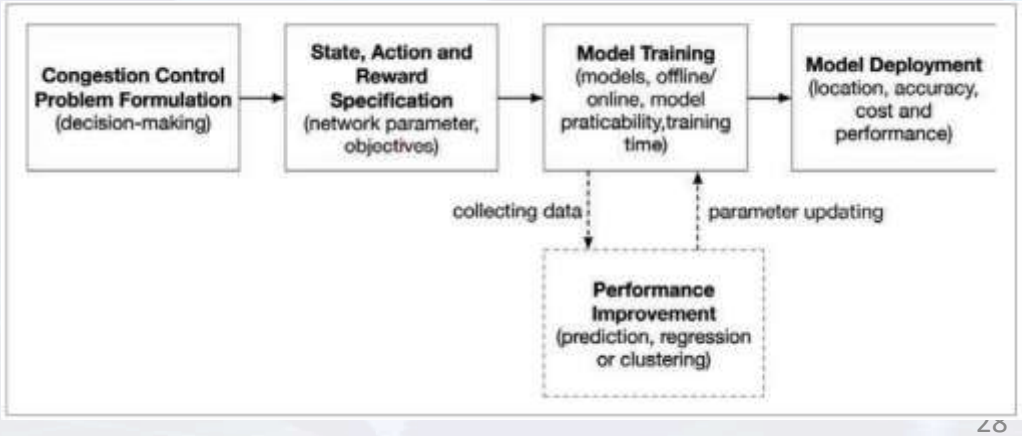
- При кратковременной перегрузке увеличивается задержка
- При достаточно долгой перегрузке пакеты сбрасываются

Алгоритмы управления перегрузкой



ML в алгоритмах управления перегрузкой

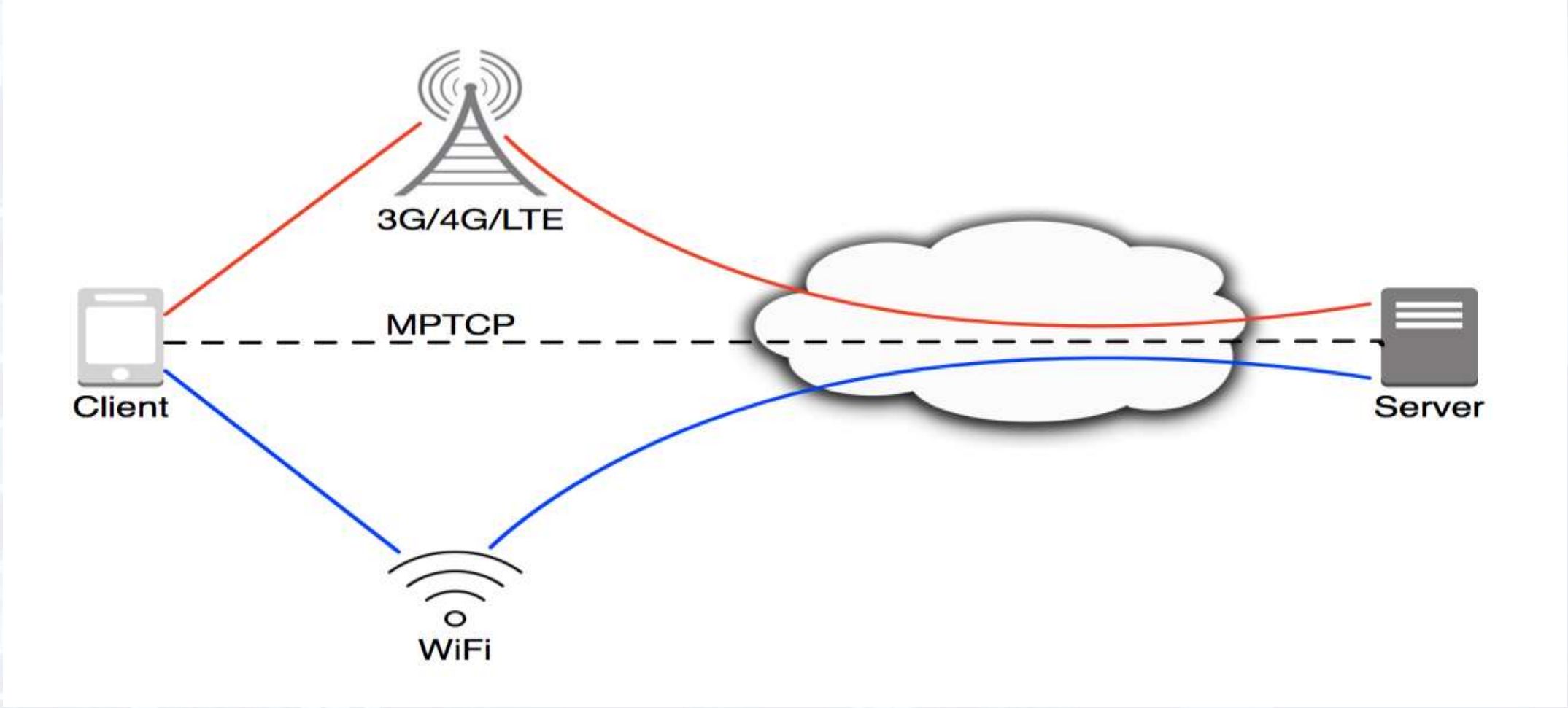




Технологии и задачи

- Машинное обучение
- Оптимизация параметров
- Разработка и анализ новых алгоритмов управления перегрузкой
  - BBR, RCP

Demultiplexed Transport Connections



Demultiplexed Transport Connections (FDMP)

Application Layer

Стандартный socket API

Transport Layer

Адаптивное управление потоками

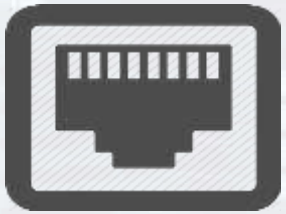
Планирование пакетов

Подпоток

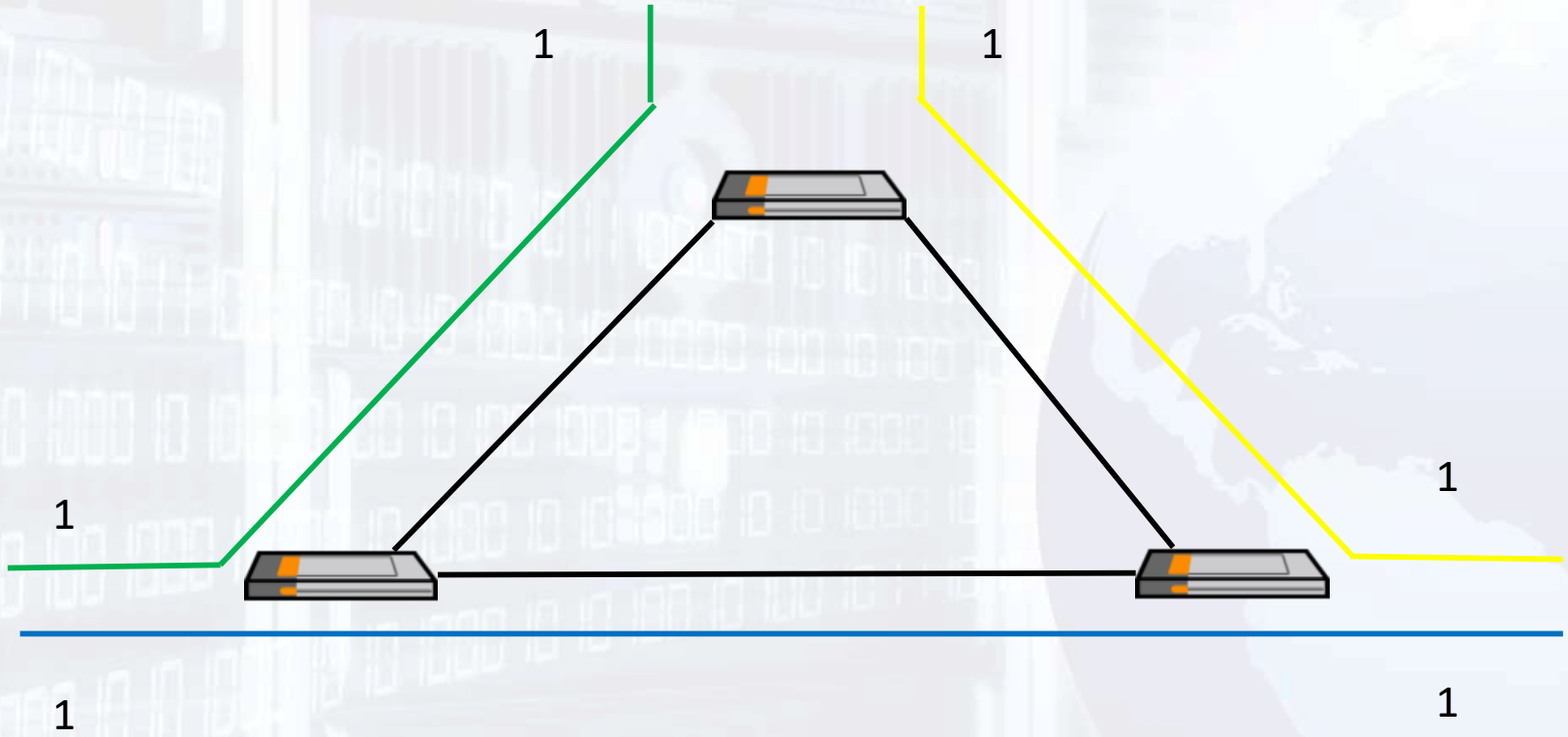
Подпоток

Подпоток

Network Layer

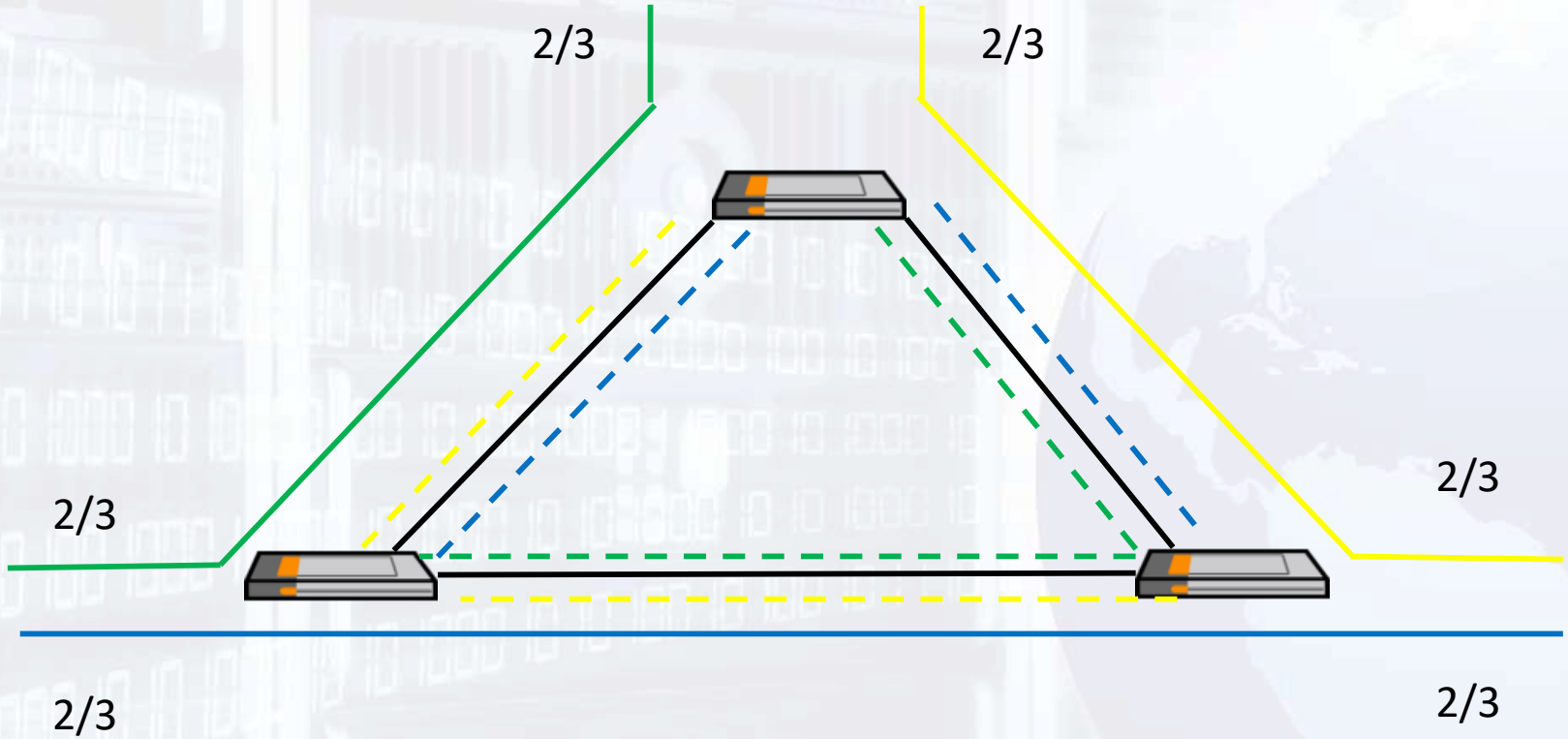


Массовая многопоточность



Общая пропускная способность сети - 6

Массовая многопоточность

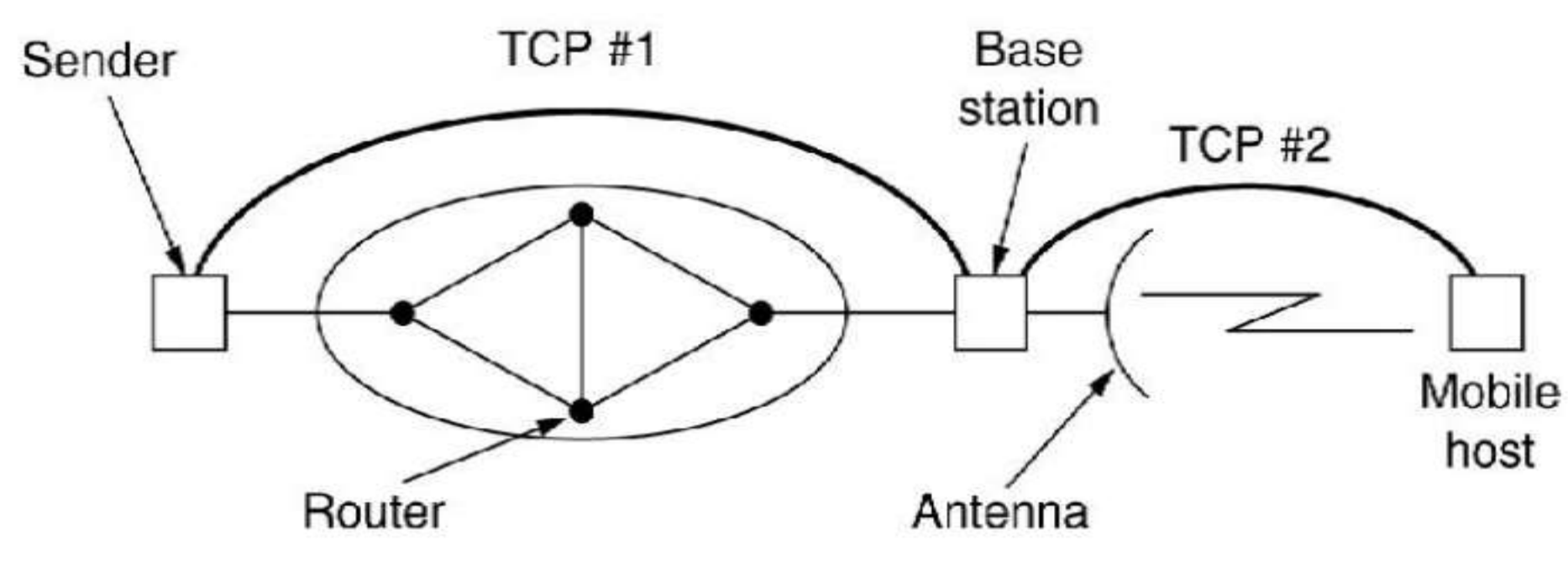


Общая пропускная способность сети - 4

Технологии и задачи

- Возможность работы с ядром Linux
- Оптимизация работы с сетевым стеком в пространстве пользователя (dpdk)
- Массовая многопоточность
- Обеспечение качества сервиса
- Умная маршрутизация
- Алгоритм управления перегрузкой

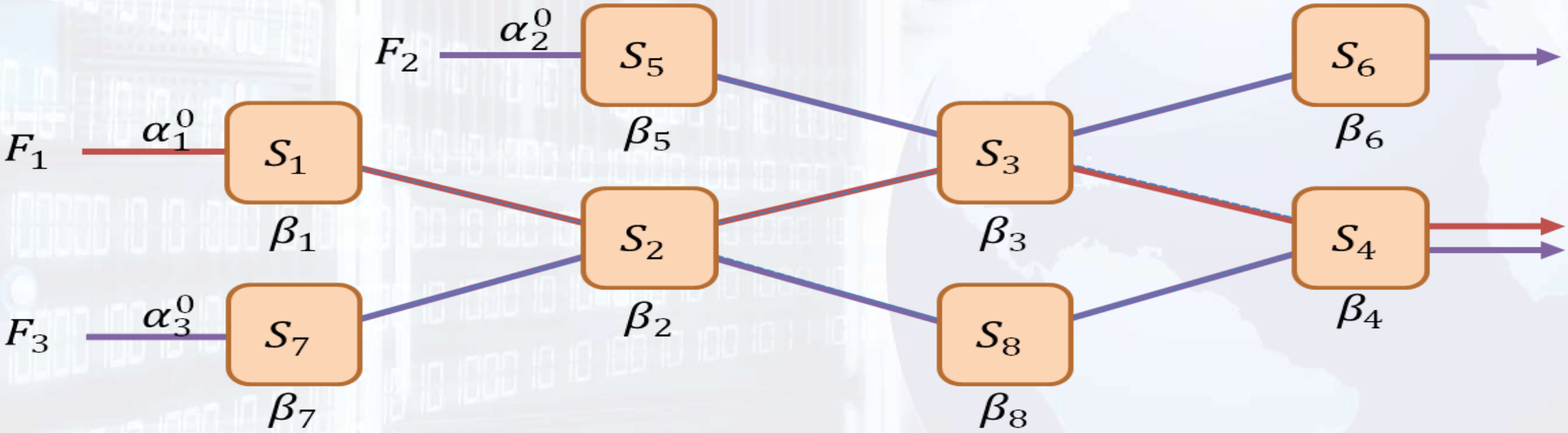
Сегментирование - Split TCP



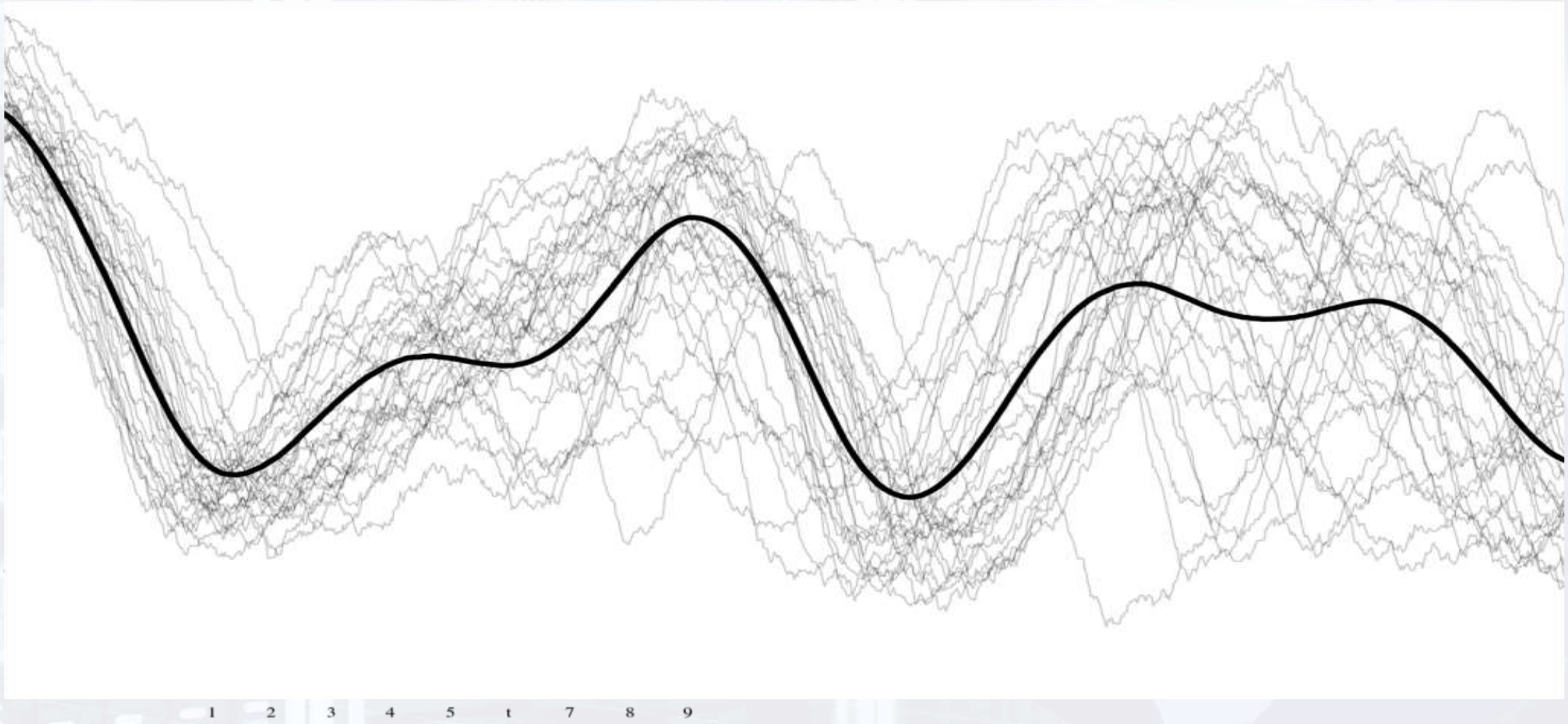
Технологии и задачи

- Разработка прокси для split TCP
- Размещение прокси в сети
- Маршрутизация в такой сети
- Оценка метрик качества сервиса для split-соединений

Сетевое исчисление

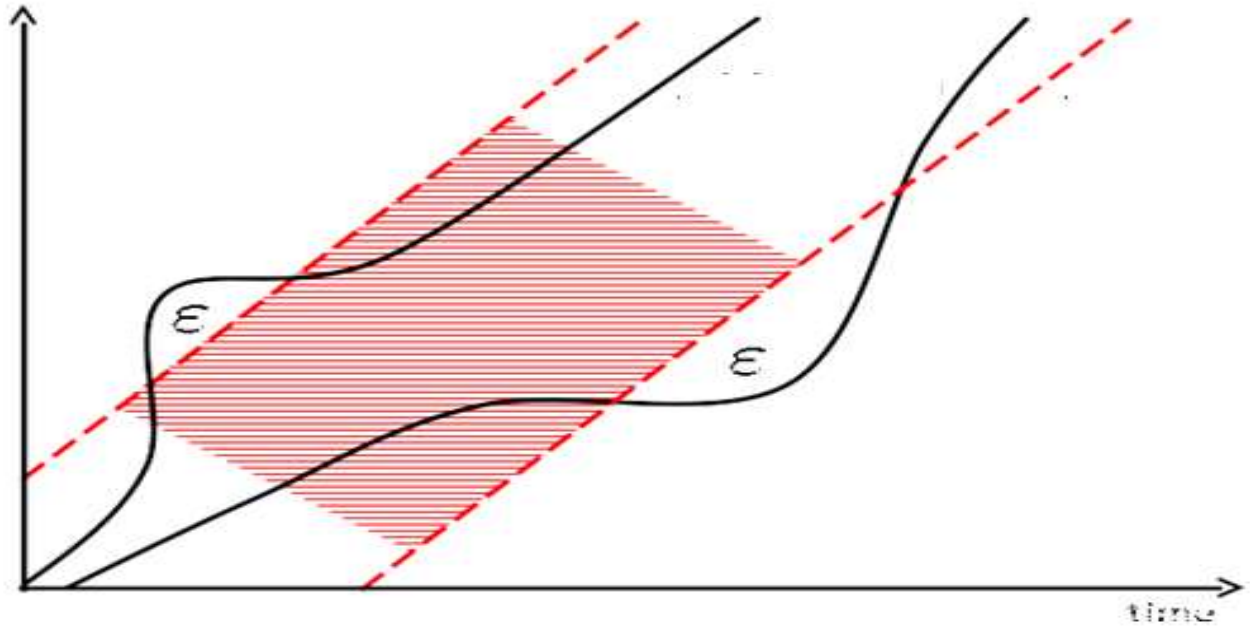


Сетевое исчисление



Стохастическое сетевое исчисление

Позволить нарушения кривых нагрузки и сервиса с некоторой вероятностью
-> теряем точность результата



Технологии и задачи

- Теория вероятностей, математическая статистика
- Линейное программирование
- Оптимизация параметров
- Учет потери пакетов
- Связь с алгоритмами управления перегрузкой

Благодарю за внимание!