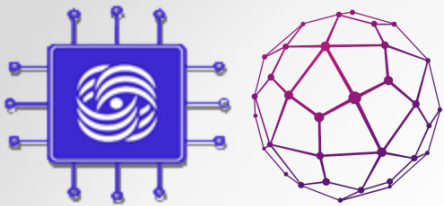


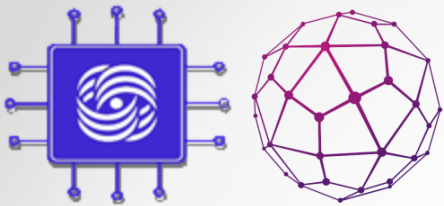
Модели распределения сетевых ресурсов

Доп. главы Компьютерных сетей и
телекоммуникации
к.ф.-м.н. Чемерицкий Е.В.



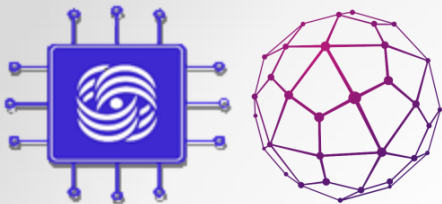
Вопросы:

- Каковы преимущества и недостатки архитектуры коммутатора с буферизацией на выходе?
- Сколько очередей внутри коммутатора с N портами, виртуальной буферизацией на входе, WRR из 5 очередей и 3-х уровневый Random Early Detection необходимо для поддержания требований качества сервиса?



План лекции

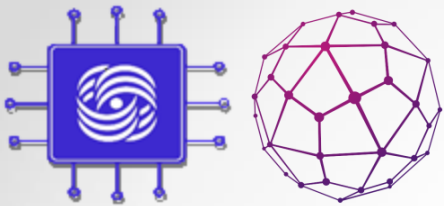
- Модель Интегрированных Сервисов
- Протокол Резервирования Ресурсов (RSVP)
- Модель Дифференцированных Сервисов
- Маршрутизация с учётом качества сервиса



Качество Сервиса в сети Интернет

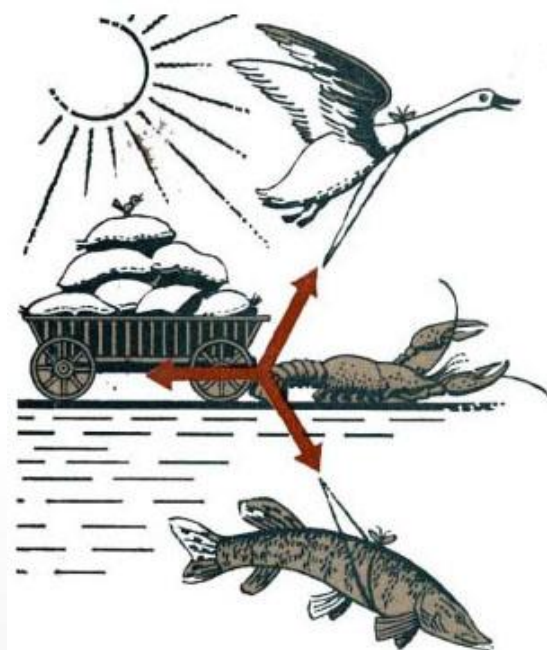
- Если управление качеством сервиса не поддерживается, то сетевые соединения обслуживаются по принципу ***best effort***
- Потоки данных обслуживаются одинаково
 - Зачем нужно разное качество сервиса?
 - Дифференцирование потоков по качеству: всегда ли это хорошая идея?

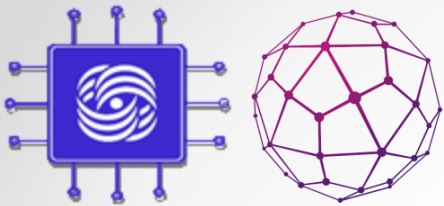
David D. Clark, K. Sollins, J. Wroclawski, R. Braden,
Tussle in Cyberspace: Defining Tomorrow's Internet
Proceedings of SIGCOMM 2002, ACM Press, 2002



Методы управления качеством сервиса в сети

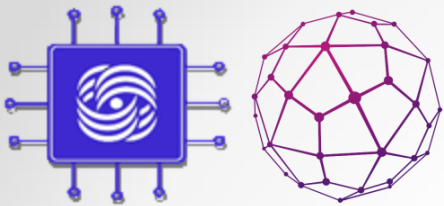
- Выполнение потребностей приложений
 - Какие параметры соединения можно задавать?
 - Даётся ли гарантия выполнения требований?
- Сложность реализации и эксплуатации
 - Какими дополнительными функциями должно обладать оборудование?
 - Какая поддержка требуется от хостов?
- Накладные расходы на использование
 - Насколько эффективна работа сети?
 - Сколько ресурсов будет задействовано?
 - Сколько будет простаивать?





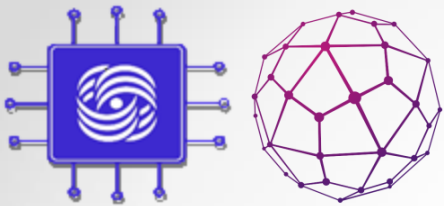
Проблема эффективного распределения ресурсов

- Проблема обеспечения качества связана с проблемой распределения сетевых ресурсов между потоками данных
 - Проблему распределения ресурсов можно формализовать как задачу оптимизации
 - Чем больше ресурсов вовлечено в обслуживание потока, тем выше качество его соединения
 - Чем большее количество ресурсов позволяет задействовать модель распределения, тем выше эффективность сети, и тем больший **уровень утилизации** достигается



Почему высокий уровень утилизации – хорошо?

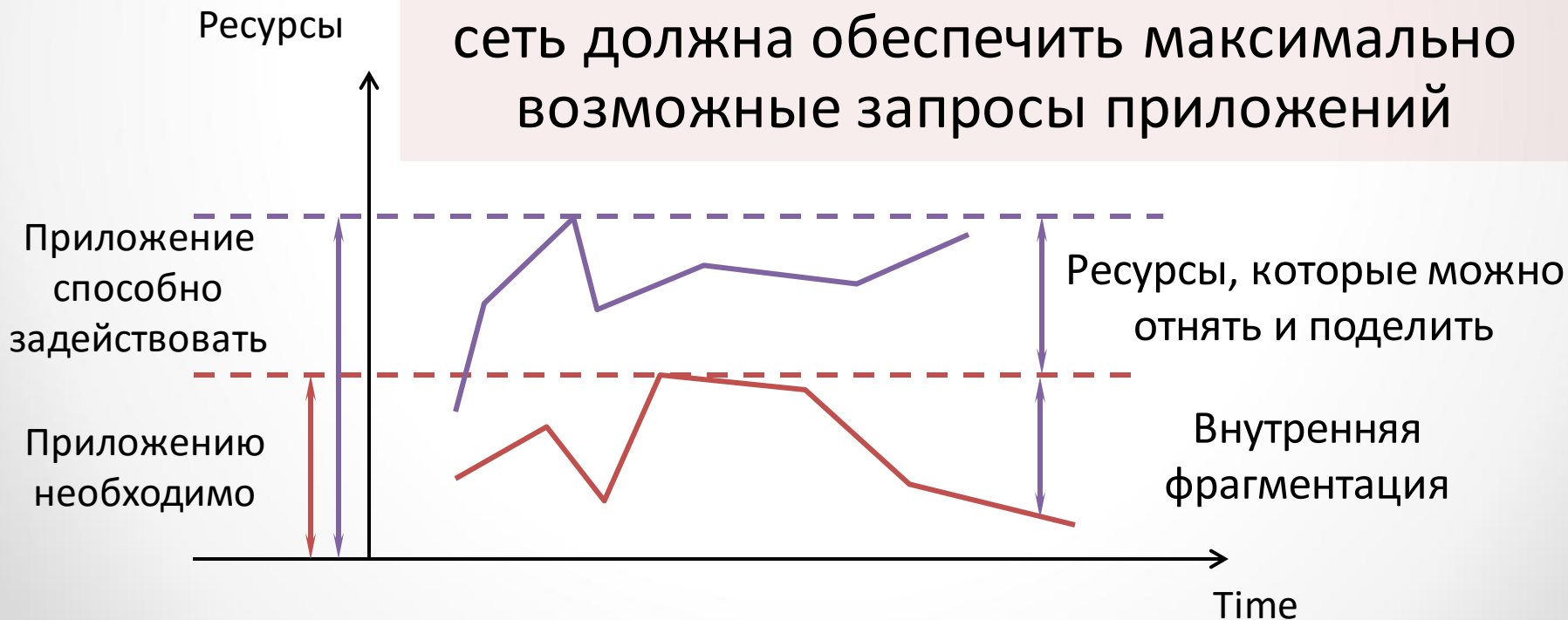
- Приложения не всегда способны задействовать все ресурсы сети
 - Spanning Tree Protocol – в топологии Fat Tree
 - Backbone networks – запас производительности
 - Разница в характеристиках аппаратуры
- Чем выше уровень утилизации, тем лучше отношение производительности сети к стоимости сетевой инфраструктуры
 - Уменьшение затрат провайдеров

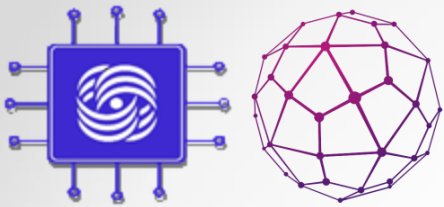


Низкая утилизация при отсутствии управления QoS

Сеть работает корректно, если она предоставляет каждому приложению необходимые ему ресурсы

При отсутствии управления качеством сеть должна обеспечить максимально возможные запросы приложений



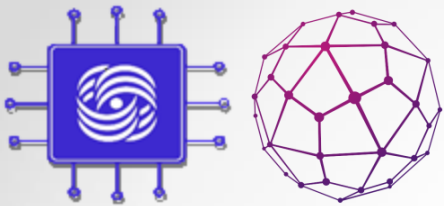


Модель Интегрированных Сервисов (IntServ)

Мультимедийный трафик в сети:

- Как оградить TCP трафик от мультимедийных данных, передающихся через UDP?
- Как обеспечить качество соединения?

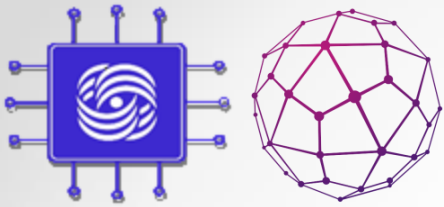
Гарантированный уровень качества можно обеспечить лишь с помощью **резервирования ресурсов** – закрепления части ресурсов сети за конкретным потоком данных



Модель Интегрированных Сервисов (IntServ)

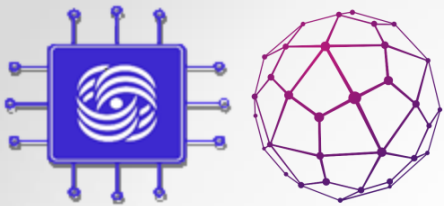
Основная идея – прокладывание маршрутов с заданным качеством путём предварительного резервирования ресурсов на оборудовании

- Модель лишь расширяет архитектуру Интернета, и способна работать по принципу *best-effort*
- Модель особенно эффективна при многоадресной передаче данных
- Допускаются накладные расходы на предварительное прокладывание маршрута
- ***всё или ничего*** – модель или гарантирует соединение нужного качества, или отказывается предоставить какое-либо соединение



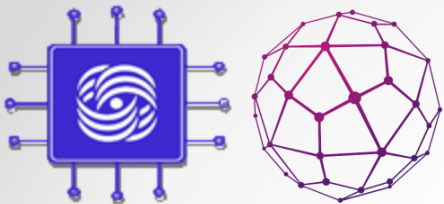
Основные компоненты модели IntServ

- Классификатор (classifier)
 - Разделение пакетов на классы обслуживания
- Планировщик (scheduler)
 - Обеспечение выполнения требований QoS
- Контроль доступа (admission control)
 - Оценка возможности добавления потоков
- Протокол резервирования ресурсов
 - Резервирование ресурсов вдоль маршрута

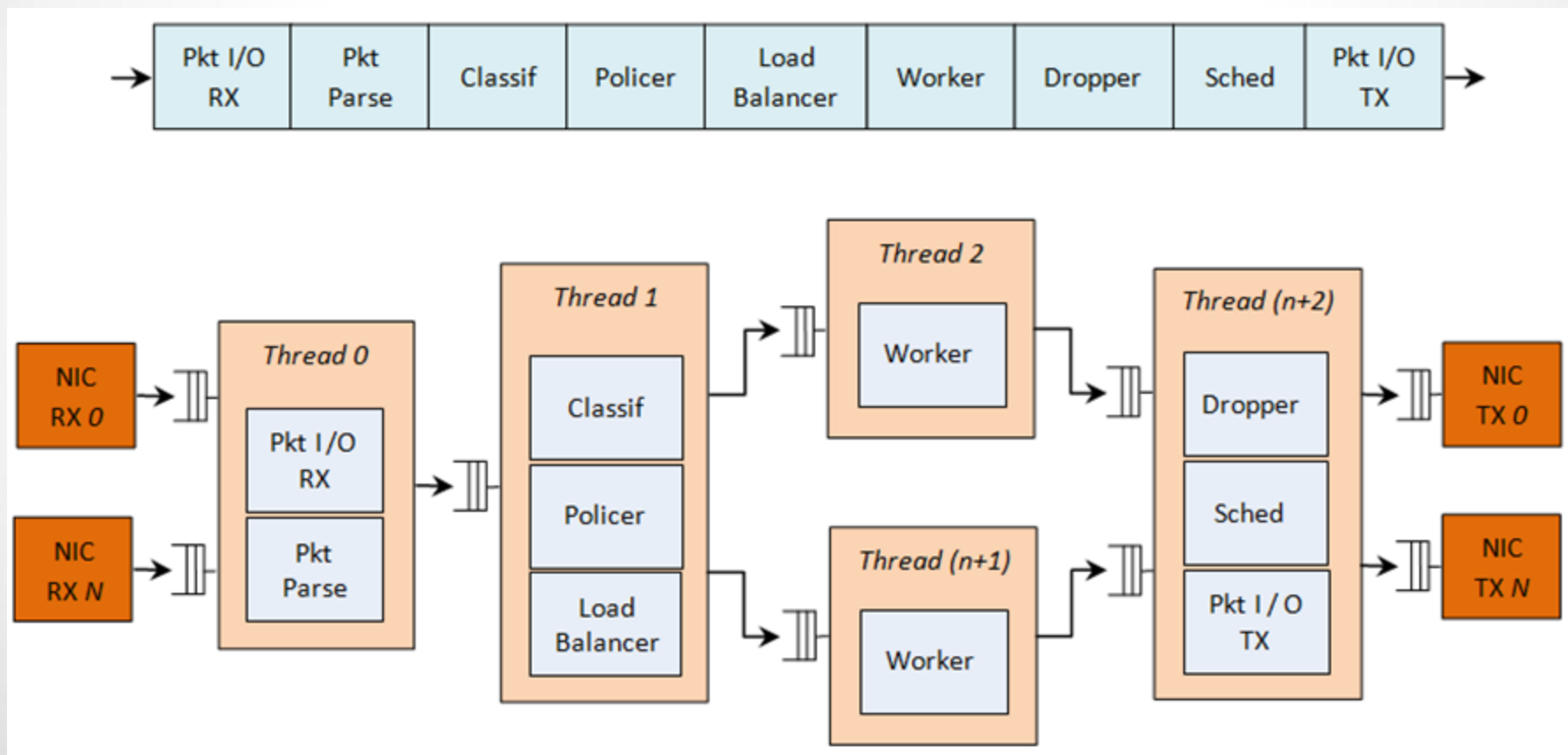


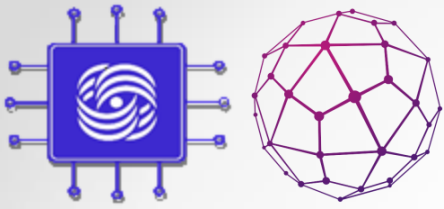
Расчёт и обеспечение качества соединений

- Распределение по очередям на входящих и исходящих интерфейсах коммутатора
- Использование policing & shaping для формирования нужного профиля потоков
- Установка надлежащих дисциплин сброса пакетов при их постановке в очереди и выборки пакетов из очередей
- Настройка алгоритмов планирования коммутационной матрицы



Расчёт и обеспечение качества соединений

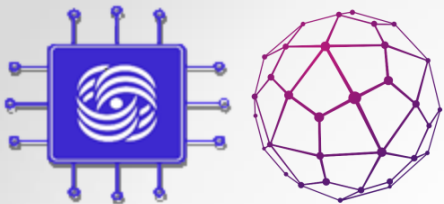




Протокол резервирования ресурсов RSVP

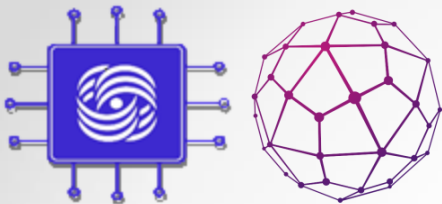
Поддерживаемые типы трафика:

- Регулярный (*best-effort*)
 - Передача файлов, просмотр почты, etc
- Чувствительный к скорости (*rate-sensitive*)
 - Потокковое вещание аудио и видео
- Чувствительный к задержке (*delay-sensitive*)
 - Voice Over IP, online игры

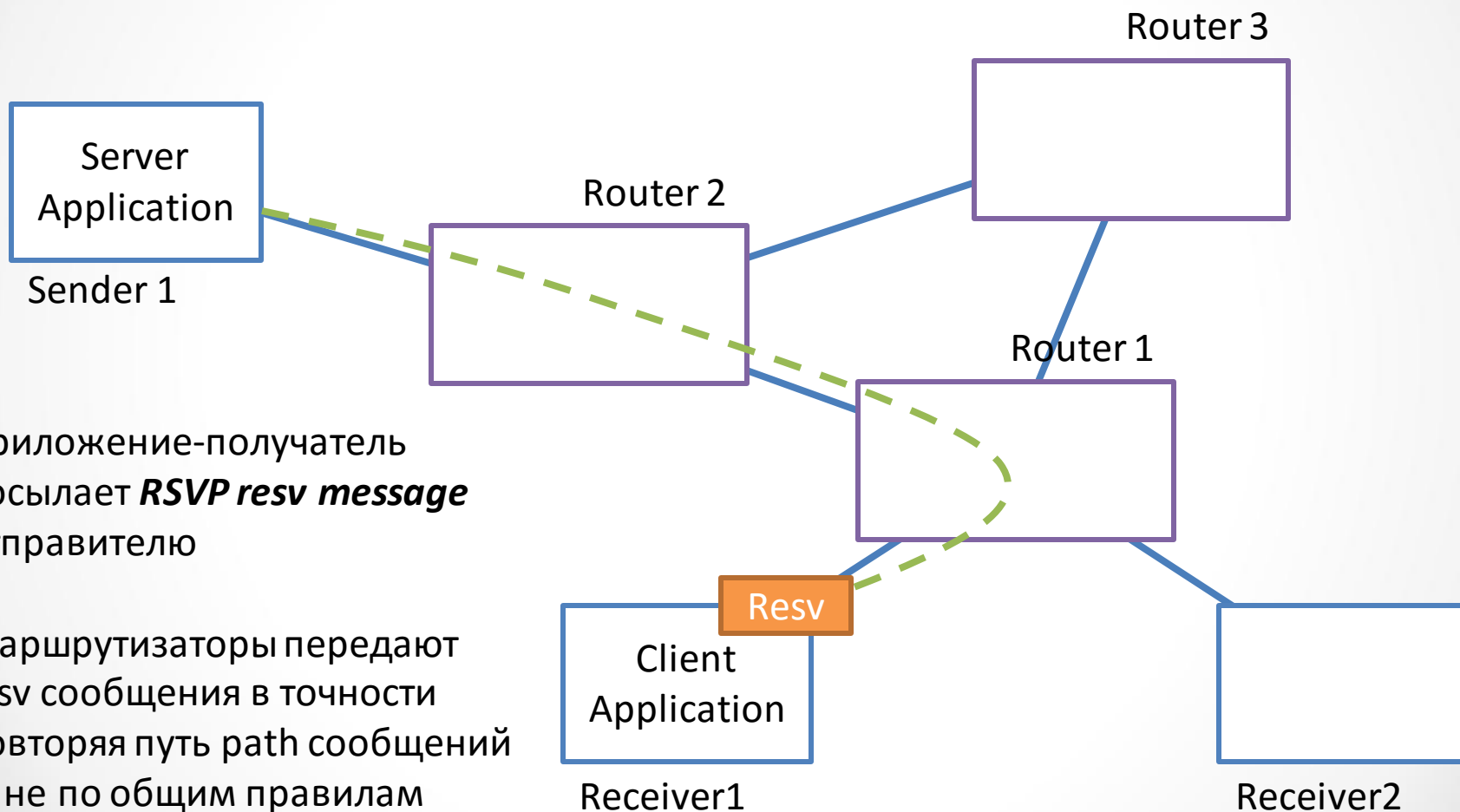


Протокол резервирования ресурсов RSVP

- **Поток данных** – последовательность пакетов, имеющих общего отправителя, единый набор получателей, и требующих одинакового уровня качества сервиса
- **Спецификация потока (*flowspec*)**
 - определяет поток данных, тип трафика, требования к качеству соединения
- **Спецификация фильтра (*filterspec*)**
 - Определяет правила доступа пакетов к зарезервированным ресурсам
- RSVP работает на уровне **сессий** – множеств потоков с одинаковым набором получателей

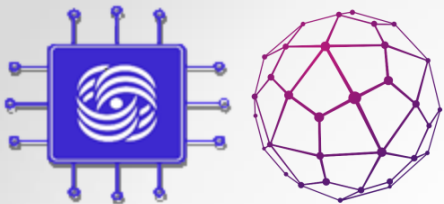


Построение маршрута

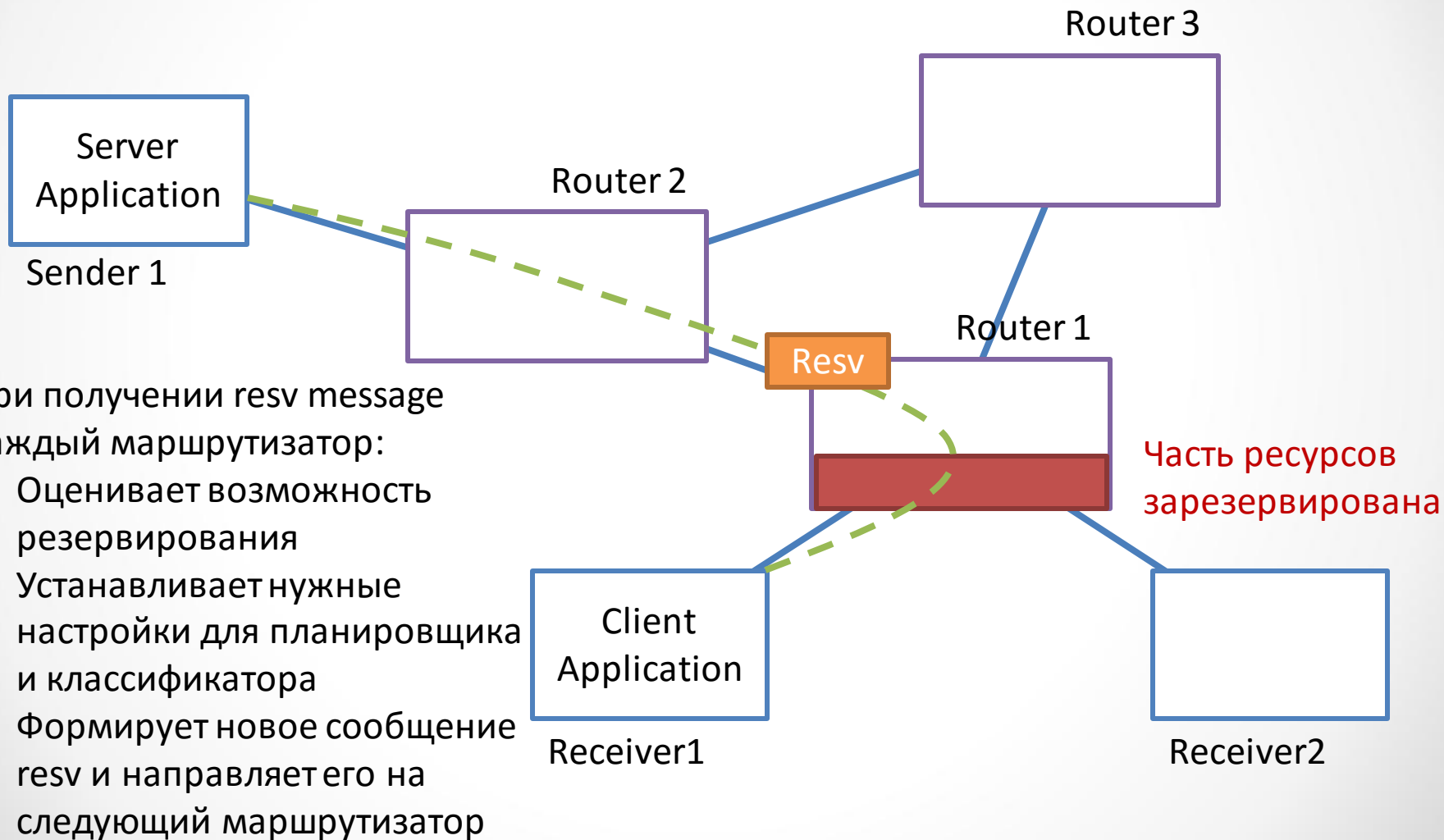


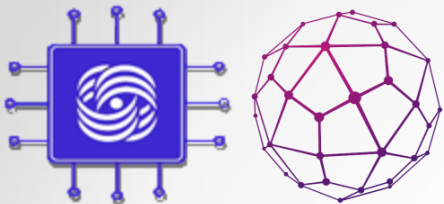
Приложение-получатель
посылает ***RSVP resv message***
отправителю

Маршрутизаторы передают resv сообщения в точности повторяя путь path сообщений (а не по общим правилам маршрутизации пакетов)

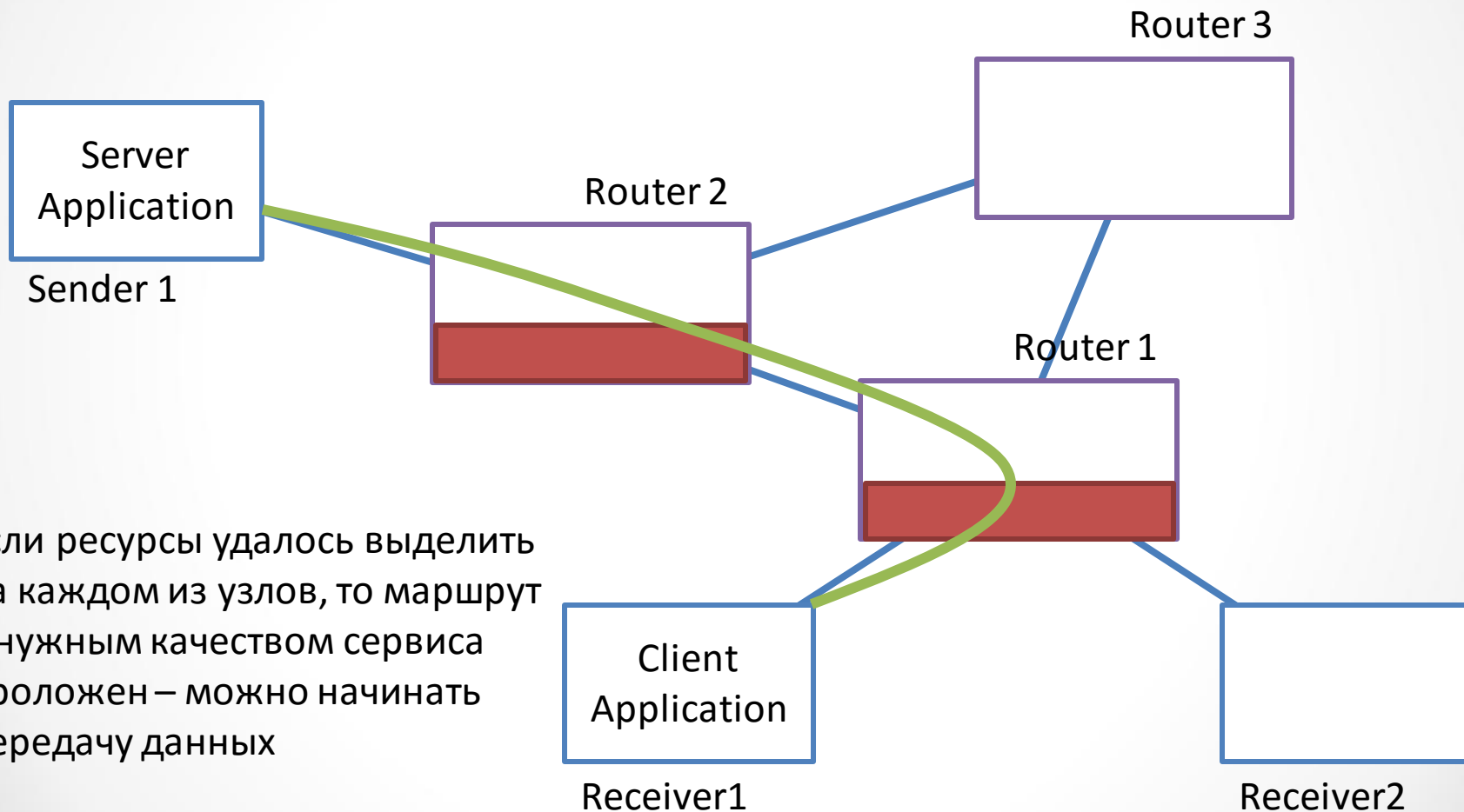


Резервирование ресурсов

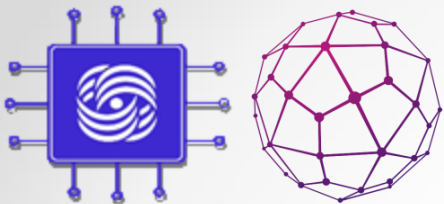




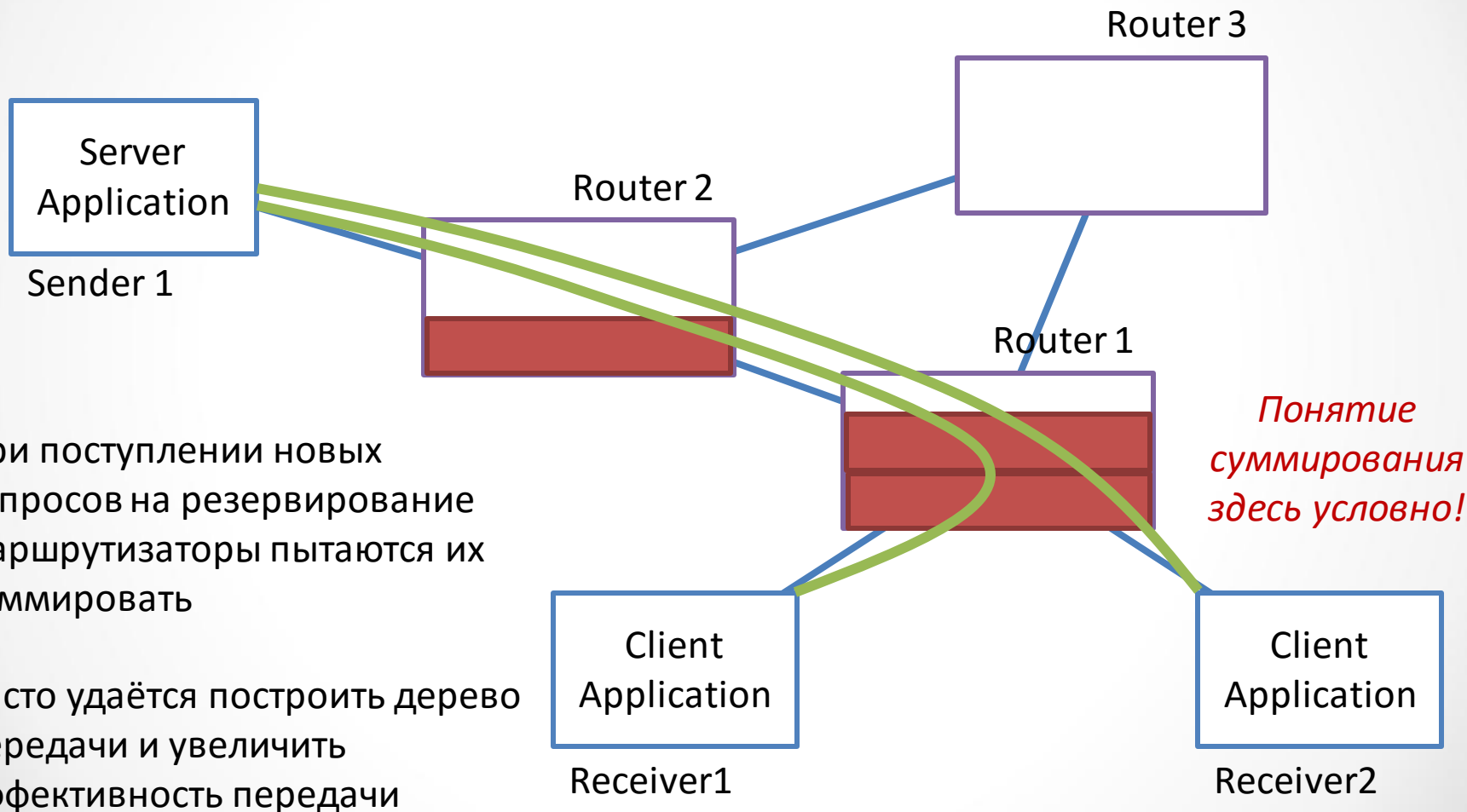
Начало передачи данных



Если ресурсы удалось выделить на каждом из узлов, то маршрут с нужным качеством сервиса проложен – можно начинать передачу данных

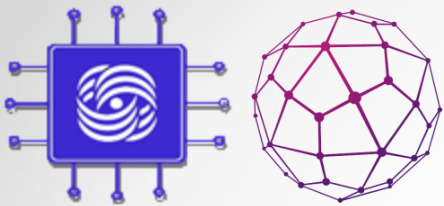


Использование filterspec



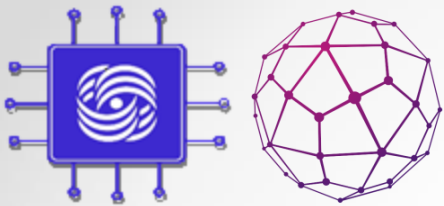
При поступлении новых запросов на резервирование маршрутизаторы пытаются их суммировать

Часто удаётся построить дерево передачи и увеличить эффективность передачи



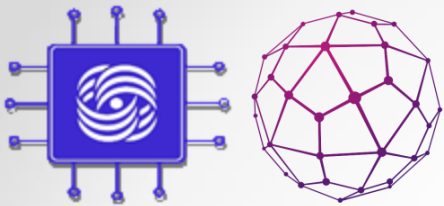
Стили резервирования FilterSpec

- Заданным фильтром (Fixed Filter)
 - Ресурсы выделяются отправителю для индивидуального пользования
 - Обычный видео стриминг
- Шаблонное (Wildcarded Filter)
 - Ресурсы разделяются между группой отправителей по заданному предикату
 - Во время аудио конференции одновременная передача данных маловероятна
- Разделяемое явно (Shared-Explicit)
 - Ресурсы разделяются между группой отправителей
 - Члены группа могут изменяться со временем



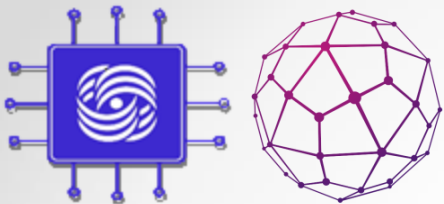
Особенности RSVP

- RSVP – **сигнальный протокол**: обеспечивает лишь резервирование вдоль маршрута: первичный выбор маршрута – забота протоколов маршрутизации
- Протокол экономно расходует ресурсы при частичном совпадении маршрутов
- Иницирует резервирование получатель
- Одни и те же ресурсы могут использоваться сразу несколькими отправителями
- Резервирование поддерживается отправителем и получателем (***Soft State***)



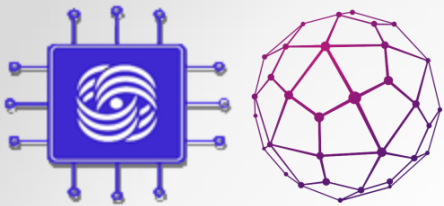
Недостатки RSVP

- Необходимо вести маршрутизацию на уровне потоков – для каждого потока нужно хранить его состояние (soft state)
- Выделение ресурсов в индивидуальном порядке слишком требовательно к железу
- Плохая масштабируемость решения
- ***Внутренняя фрагментация*** ресурсов при статическом резервировании – низкая утилизация оборудования



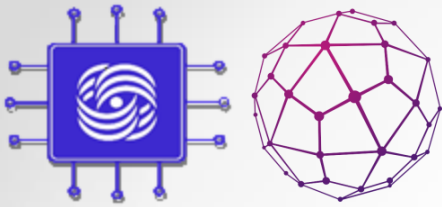
Наследие RSVP

- Способность RSVP прокладывать стабильные маршруты послужила основой для разработки **коммутации потоков**
- Протокол IFMP (1996, Ipsilon Networks) – **коммутация по тегам** в сетях ATM
- Протокол MPLS (1997, Cisco Systems) – **коммутация по меткам** в сетях IP
- Проект Ethane & протокол OpenFlow (2008)



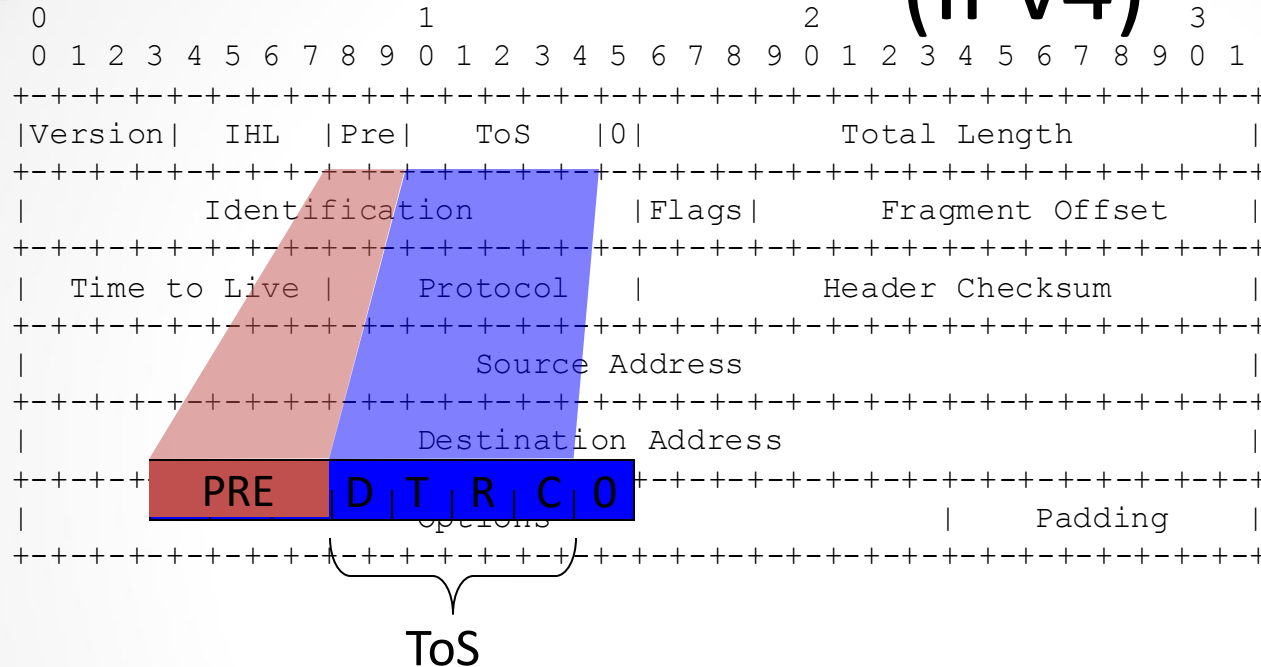
Модель Дифференцированных Сервисов (DiffServ)

- Каждый маршрутизатор имеет несколько predetermined классов обслуживания
- Пограничные маршрутизаторы определяют класс потока, маркируют его пакеты **dscp метками** и проводят **traffic conditioning** – используют инструменты policing & shaping для установки нужного профиля трафика
- На внутренних маршрутизаторах пакеты с более высоким приоритетом получают большую долю ресурсов, и наоборот



Internet Protocol version 4 (IPv4)

[RFC1349]



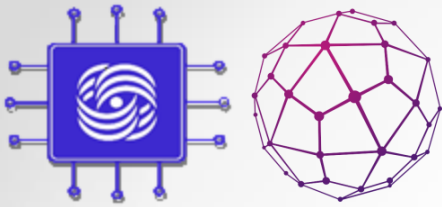
ToS

☐ Type of Service

- ☐ D – minimize delay
- ☐ T – maximize throughput
- ☐ R – maximize reliability
- ☐ C – minimize cost

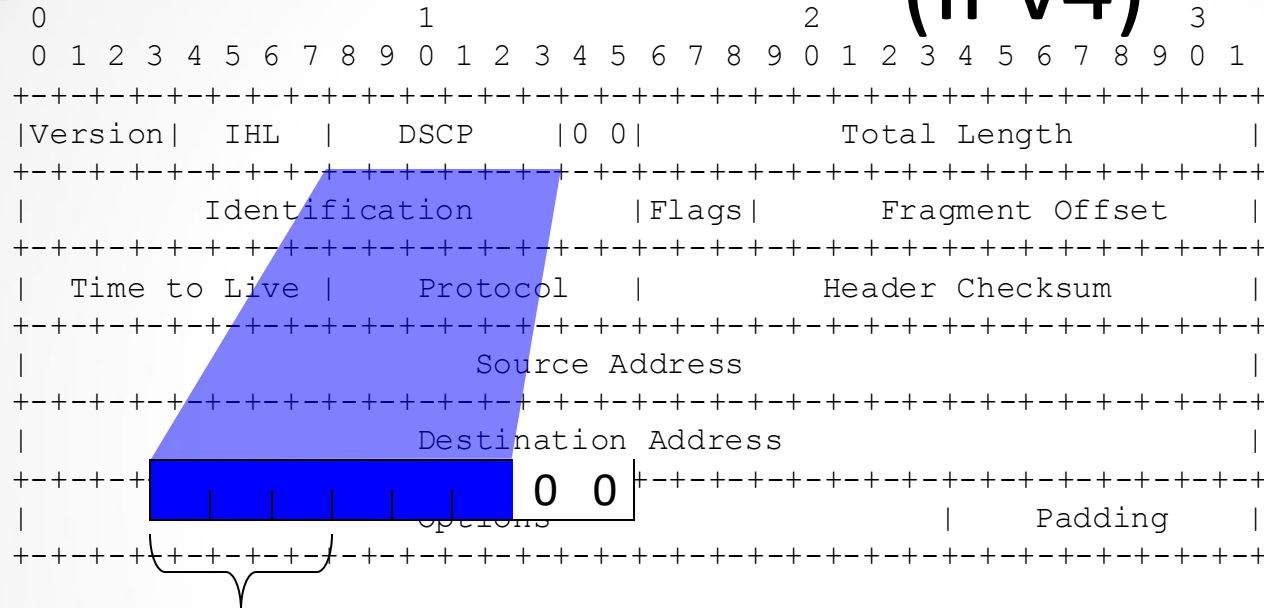
PRE

- Precedence Field
 - Priority of the packet



Internet Protocol version 4 (IPv4)

[RFC2474]



Class selector codepoints
of the form xxx000

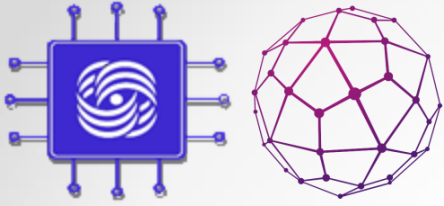
DSCP

❑ Differentiated Services Codepoint

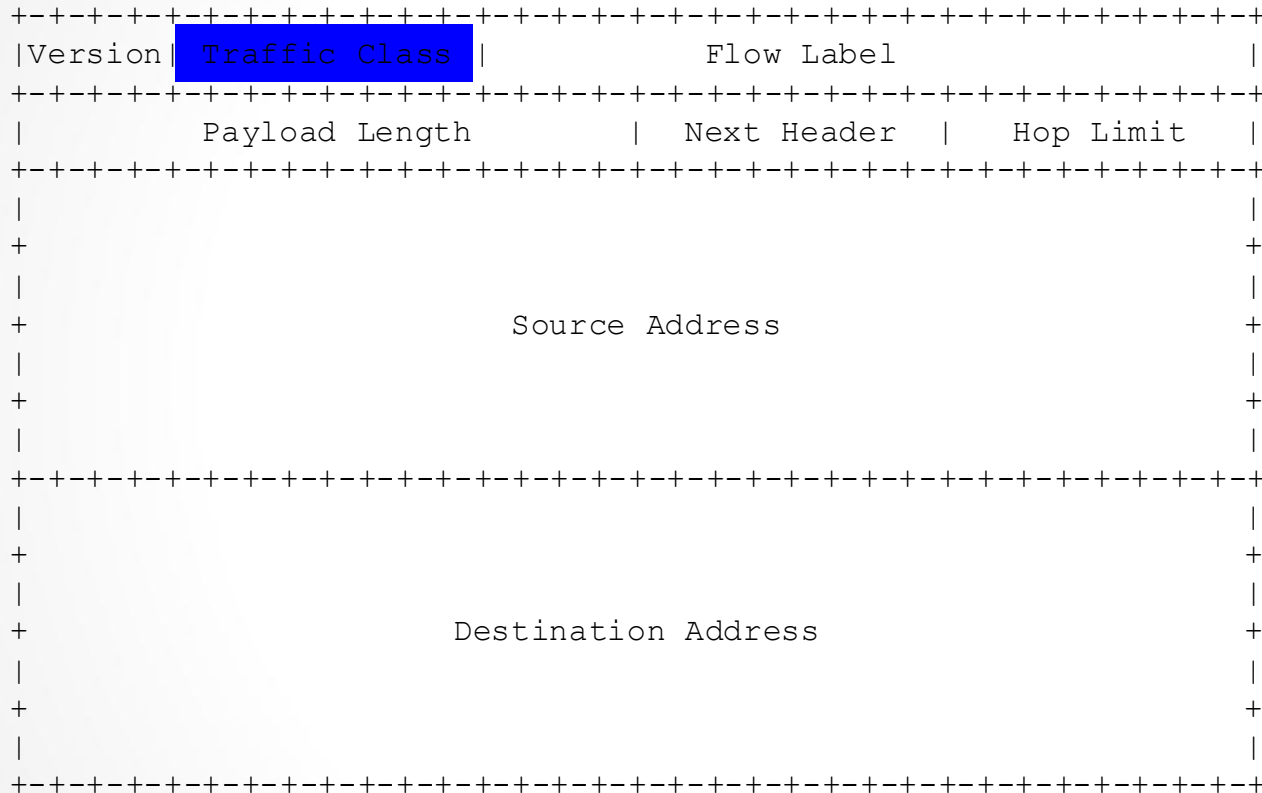
xxxxx0 reserved for standardization

xxxx11 reserved for local use

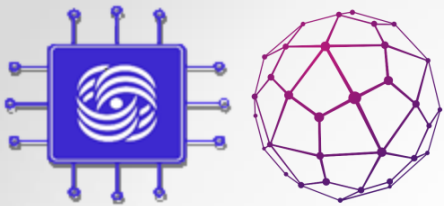
xxxx01 open for local use, may be standardized later



Internet Protocol version 6 (IPv6)

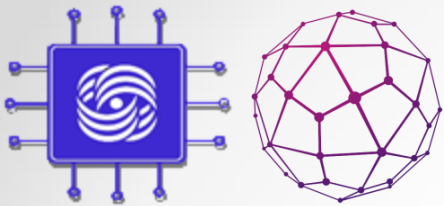


- Traffic class
 - Interpret like IPv4's DS field



Стандартные классы обслуживания DiffServ

- Default Forwarding (DF)
 - Обычно обслуживается по best-effort
- Expedient Forwarding (EF)
 - Идёт через очередь с высшим приоритетом
 - Маленькие delay, jitter & loss
- Assured Forwarding (AF)
 - Идёт через менее приоритетную очередь
 - Охватывает несколько классов с разной политикой сброса при заполнении очереди



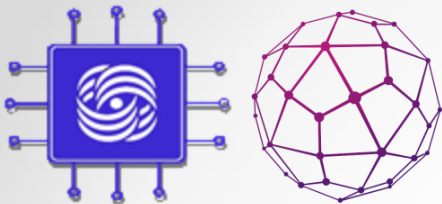
Модель DiffServ

Преимущества

- Отсутствие внутренней фрагментации
- Высокая степень утилизации оборудования
- Простота реализации в аппаратуре
- Хорошая масштабируемость

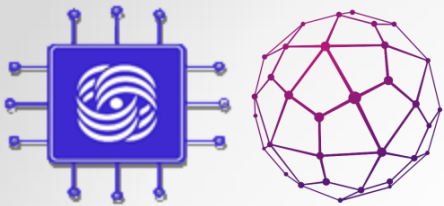
Недостатки

- Не различает метрик качества
- Не предоставляет гарантий качества
- Распределяет ресурсы пропорционально



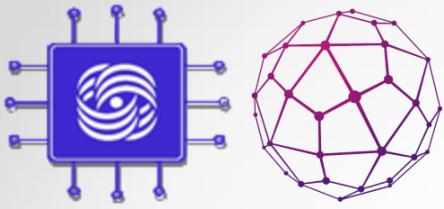
QoS Routing

- Прокладывание индивидуальных маршрутов для потоков с учётом их требований качества сервиса
 - Маршруты между двумя точками в сети могут быть построены разными способами, если их требования качества для потоков не совпадают
- Появляется возможность вовлечь ресурсы, расположенные вне основных маршрутов передачи данных
- Метод совместим с обеими моделями управления ресурсами



Проблемы QoS Routing

- Необходим гранулярный контроль за коммутационными устройствами
 - Управление на уровне потоков
- Необходимо централизованное управление
 - Алгоритмы маршрутизации не будут сходиться
- Нужна возможность получения требований качества от приложений
- Построение маршрутов – тяжёлая задача

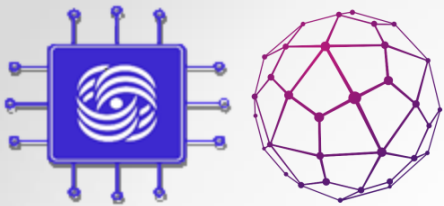


Проблема наилучшего распределения ресурсов

Управлять выполнением требований качества и уровнем утилизации сети можно за счёт особого алгоритма построения маршрутов

Критерии оптимизации:

- Увеличение устойчивости к пиковым нагрузкам – равномерное распределение
- Обслуживание потоков наименьшим количеством устройств – можно выключить часть оборудования
- Максимизация надёжности сети



Управление QoS с помощью Flow (De) Multiplexing Protocol

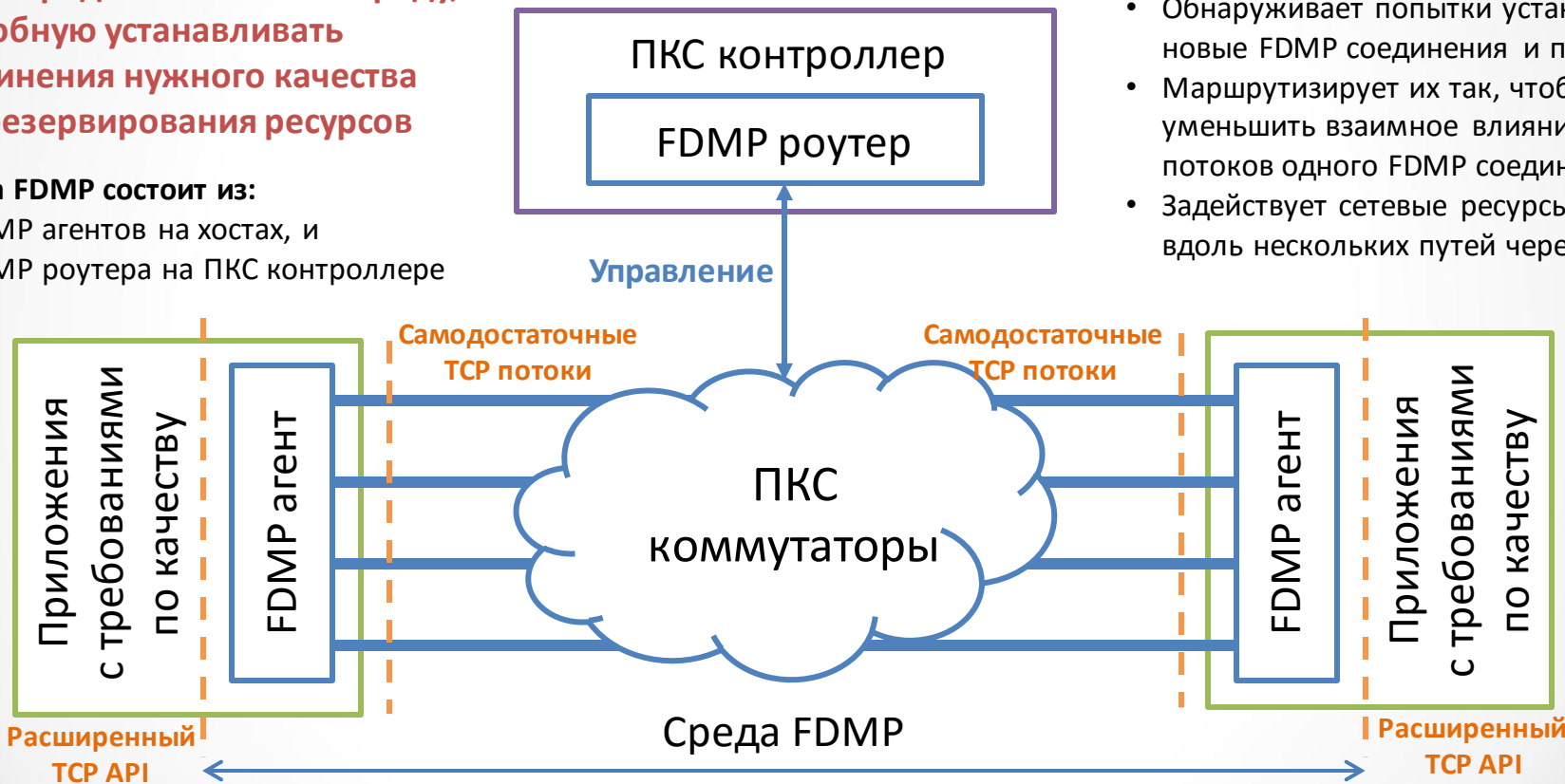
FDMP представляет собой среду, способную устанавливать соединения нужного качества без резервирования ресурсов

Среда FDMP состоит из:

- FDMP агентов на хостах, и
- FDMP роутера на ПКС контроллере

FDMP роутер:

- Обнаруживает попытки установить новые FDMP соединения и потоки
- Маршрутизирует их так, чтобы уменьшить взаимное влияние потоков одного FDMP соединения
- Задействует сетевые ресурсы вдоль нескольких путей через сеть



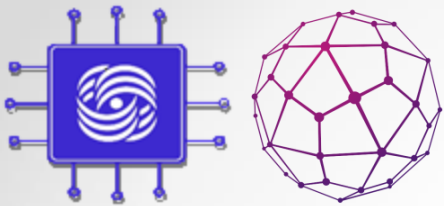
FDMP агент на стороне отправителя:

Разделяет поток байтов от отправителя на несколько TCP потоков и передаёт их независимо друг от друга

Число TCP потоков изменяется динамически в зависимости от актуального состояния сети и текущих требований приложений

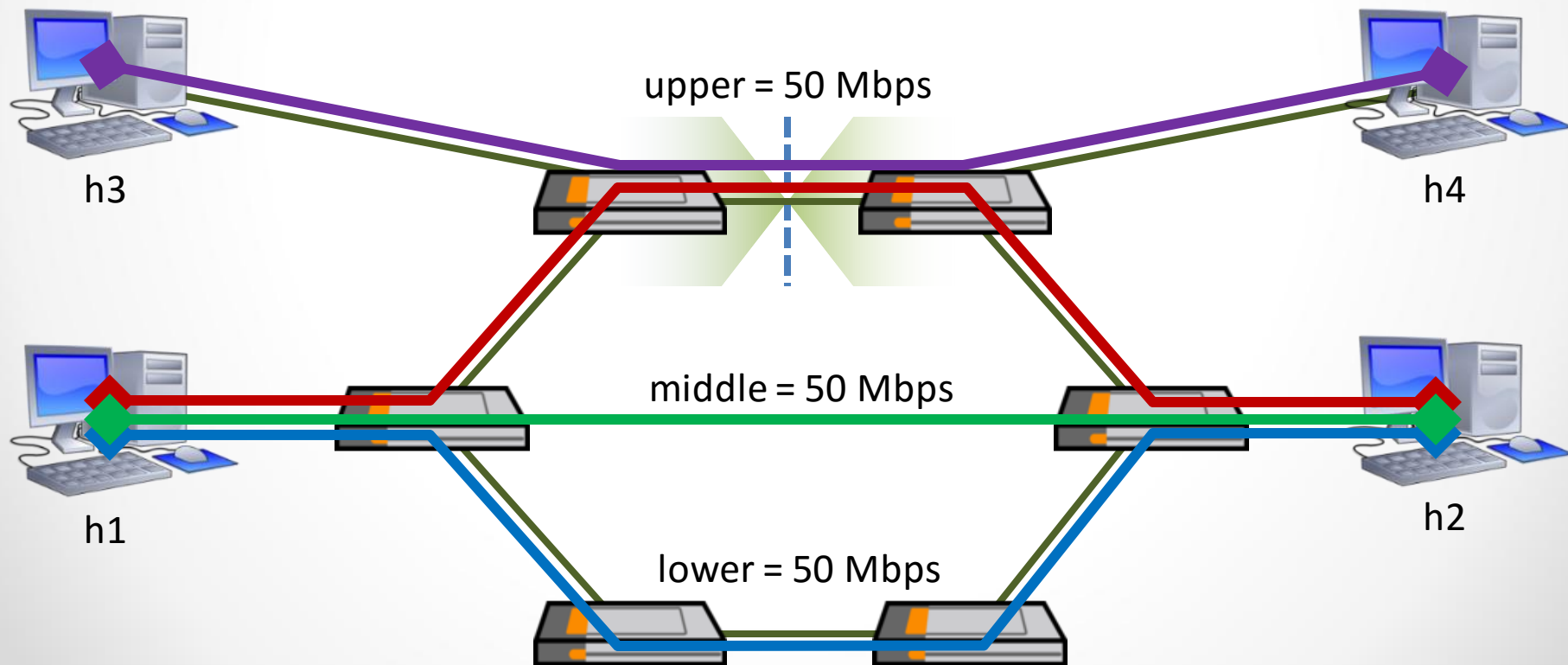
FDMP агент на стороне получателя:

Собирает из пакетов, приходящих через TCP потоки от отправителя, оригинальный поток байтов



Демонстрация FDMP

Динамическая адаптация соединения к перегрузкам в сети:
пропускная способность от h1 к h2 не меньше 80 Mbps



MP TCP connection

```
TCP window size: 85.3 KByte (default)
```

Iperf сервер на h2
(основной получатель)

Iperf клиент на h1
(основной отправитель)

```
>iperf -c 192.168.1.1 -i1 -t50
```

Subflows

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

Индикаторы скорости
подпотоков соединения на
маршрутах upper, middle и
lower (слева направо)

Extra flow

```
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)
```

Iperf сервер на h4
(побочный отправитель)

Iperf клиент на h3
(побочный отправитель)

MP TCP connection

```
TCP window size: 85.3 KByte (default)
```

Запускаем генератор трафика
на 50 секунд

Соединение использует
единственный подпоток,
проходящий по пути upper

Пропускной способности не
достаточно чтобы выполнить
требование:
 $50 \text{ Mbps} < 80 \text{ Mbps}$

Многопоточный агент
детектирует нехватку скорости
и пытается открыть новый
подпоток

Subflows

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

Extra flow

```
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)
```

Приложение контроллера
проводит его по такому
маршруту, который имеет как
можно меньше пересечений с
другими подпотоками этого же
соединения

MP TCP connection

```
-----  
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562  
-----  
TCP window size: 21.0 KByte (default)  
-----  
[ 3] local 192.168.1.3 port 50562 connected with 192.168.1.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 1.0 sec   6.12 MBytes   51.4 Mbits/sec  
[ 3]  1.0- 2.0 sec   6.00 MBytes   50.3 Mbits/sec  
[ 3]  2.0- 3.0 sec   6.00 MBytes   50.3 Mbits/sec
```

Subflows

```
V = 0 bps  
V = 0 bps  
V = 172800 bps  
V = 6264932 bps  
V = 6252820 bps
```

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

```
V = 0 bps
```

Теперь используются пути
upper и middle

```
V = 0 bps
```

При использовании двух
подпотоков требования к
пропускной способности
выполнены

Extra flow

```
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)  
-----
```

Node: h4

```
V  
V  
V  
V  
V  
V  
V
```

MP TCP connection

```
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562
```

ID	Interval	Transfer	Bandwidth
[3]	0.0- 1.0 sec	6.12 MBytes	51.4 Mb
[3]	1.0- 2.0 sec	6.00 MBytes	50.3 Mb
[3]	2.0- 3.0 sec	6.00 MBytes	50.3 Mbits/sec
[3]	3.0- 4.0 sec	11.5 MBytes	96.5 Mb
[3]	4.0- 5.0 sec	11.2 MBytes	94.4 Mb
[3]	5.0- 6.0 sec	11.1 MBytes	93.3 Mb

Почему колеблется скорость соединения?

Многопоточный агент пробует закрыть подпоток, но требования сразу же нарушаются и подпоток быстро возобновляется

Агрессивность поведения агента можно настраивать

Subflows

```
V = 172800 bps
V = 6264932 bps
V = 6252820 bps
V = 12484444 bps
V = 6260390 bps
```

```
V = 0 bps
V = 0 bps
V = 0 bps
V = 6019066 bps
V = 6251306 bps
```

Extra flow

```
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
```

MP TCP connection

```
-----  
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562  
[ 3] 4.0- 5.0 sec 11.2 MBytes 94.4 Mbits/sec  
[ 3] 5.0- 6.0 sec 11.1 MBytes 93.3 Mb  
[ 3] 6.0- 7.0 sec 6.38 MBytes 53.5 Mb  
[ 3] 7.0- 8.0 sec 9.50 MBytes 79.7 Mb  
[ 3] 8.0- 9.0 sec 11.2 MBytes 94.4 Mb  
[ 3] 9.0-10.0 sec 11.4 MBytes 95.4 Mb  
[ 3] 10.0-11.0 sec 7.75 MBytes 65.0 Mbits/sec
```

Побочный поток мешает
передаче данных основного
потока – скорость передачи по
пути upper падает

Subflows

```
V = 12484444 bps  
V = 6260390 bps  
V = 6207400 bps  
V = 12467790 bps  
V = 13111240 bps
```

```
V = 6019066 bps  
V = 6251306 bps  
V = 6236166 bps  
V = 5562436 bps  
V = 12912906 bps
```

```
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

Запускаем генерацию
побочного потока через путь
upper

Extra flow

```
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)  
-----
```

Node: h4

```
V  
V  
V  
V  
V  
V  
V
```

I

MP TCP connection

```
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562  
[ 3] 11.0-12.0 sec 8.50 MBytes 71.3 Mbits/sec  
[ 3] 12.0-13.0 sec 10.4 MBytes 87.0 Mbits/sec  
[ 3] 13.0-14.0 sec 10.4 MBytes 87.0 Mbits/sec  
[ 3] 14.0-15.0 sec 7.25 MBytes 60.8 Mbits/sec  
[ 3] 15.0-16.0 sec 6.00 MBytes 50.3 Mbits/sec  
[ 3] 16.0-17.0 sec 7.62 MBytes 64.0 Mbits/sec  
[ 3] 17.0-18.0 sec 6.88 MBytes 57.7 Mbits/sec
```

Subflows

```
V = 4755474 bps  
V = 10540468 bps  
V = 4817548 bps  
V = 2826638 bps  
V = 1632092 bps  
V = 6249792 bps  
V = 12481416 bps  
V = 2084778 bps  
V = 5771368 bps  
V = 6254334 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps  
V = 0 bps
```

Многопоточный агент
запускает дополнительный
подпоток

Приложение контроллера
прокладывает новый маршрут
по пути lower

Соединение использует сразу
три подпотока:
upper, middle и lower

Extra flow

TCP window size: 85.3 KByte (default)

```
[ 5] local 10.0.0.3 port 5001 connected with 10.0.0.1 port 5001
```

Node: h4

```
[ 4] 1.0- 2.0 sec 1.62 MBytes 13.6 Mb  
[ 4] 2.0- 3.0 sec 1.12 MBytes 9.44 Mb  
[ 4] 3.0- 4.0 sec 4.00 MBytes 33.6 Mb  
[ 4] 4.0- 5.0 sec 4.12 MBytes 34.6 Mb  
[ 4] 5.0- 6.0 sec 4.12 MBytes 34.6 Mb
```

MP TCP connection

```
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562
```

```
[ 3] 17.0-18.0 sec 6.88 MBytes 57.7 Mbits/sec
[ 3] 18.0-19.0 sec 10.5 MBytes 88.1 Mb
[ 3] 19.0-20.0 sec 13.0 MBytes 109 Mb
[ 3] 20.0-21.0 sec 13.4 MBytes 112 Mb
[ 3] 21.0-22.0 sec 12.8 MBytes 107 Mb
[ 3] 22.0-23.0 sec 11.4 MBytes 95.4 Mb
[ 3] 23.0-24.0 sec 10.9 MBytes 91.2 Mb
```

Требования к пропускной
способности выполнены

Многопоточный агент снова
пытается закрыть один из
подпотоков

Subflows

```
V = 2826638 bps
V = 1632092 bps
V = 1795604 bps
V = 8116554 bps
V = 163512 bps
```

```
V = 5771368 bps
V = 6254334 bps
V = 6248278 bps
V = 19067316 bp
V = 6252820 bps
```

Агент временно перестаёт
передавать данные через
подпоток upper

Extra flow

```
TCP window size: 85.3 KByte (default)
```

```
[ 5] local 10.0.0.3 port 5001 connected with 10.0.0.4 port 33624
```

Node: h4

```
[ 4] 6.0- 7.0 sec 4.12 MBytes 34.6 Mbits/sec
[ 4] 7.0- 8.0 sec 3.50 MBytes 29.4 Mbits/sec
[ 4] 8.0- 9.0 sec 2.38 MBytes 19.9 Mbits/sec
[ 4] 9.0-10.0 sec 5.75 MBytes 48.2 Mbits/sec
[ 4] 10.0-11.0 sec 5.75 MBytes 48.2 Mbits/sec
```

MP TCP connection

```

-----
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562
[ 3] 33.0-34.0 sec 9.38 MBytes 78.6 Mbits/sec
[ 3] 34.0-35.0 sec 12.2 MBytes 103 Mb
[ 3] 35.0-36.0 sec 10.6 MBytes 89.1 Mb
[ 3] 36.0-37.0 sec 8.12 MBytes 68.2 Mb
[ 3] 37.0-38.0 sec 7.75 MBytes 65.0 Mbits/sec
[ 3] 38.0-39.0 sec 10.4 MBytes 87.0 Mb
[ 3] 39.0-40.0 sec 11.4 MBytes 95.4 Mb

```

Требования к пропускной способности не нарушаются

Подпоток upper закрывается

Subflows

```

V = 0 bps
V = 0 bps
V = 0 bps
V = 0 bps
V = 0 bps

```

```

V = 4597268 bps
V = 1643794 bps
V = 12492014 bps
V = 6211942 bps
V = 6464780 bps

```

```

V = 5571520 bps
V = 6254334 bps
V = 6481434 bps
V = 4479926 bps
V = 6373940 bps

```

Extra flow

```

[ 5] local 10.0.0.3 port 5001 connected with 10.0.0.4 port 33624
[ ID] Interval      Transfer      Bandwidth
[ 5]  0.0-20.2 sec  89.1 MBytes  37.0 Mbits/sec

```

Node: h4

```

[ 4] 16.0-17.0 sec  5.62 MBytes  47.2 Mbits/sec
[ 4] 17.0-18.0 sec  5.75 MBytes  48.2 Mbits/sec
[ 4] 18.0-19.0 sec  5.62 MBytes  47.2 Mbits/sec
[ 4] 19.0-20.0 sec  5.75 MBytes  48.2 Mbits/sec
[ 4]  0.0-20.0 sec  89.1 MBytes  37.3 Mbits/sec

```

MP TCP connection

```

-----
[ 4] local 192.168.1.1 port 5001 connected with 192.168.1.3 port 50562
[ 3] 38.0-39.0 sec 10.4 MBytes 87.0 Mbits/sec
[ 3] 39.0-40.0 sec 11.4 MBytes 95.4 Mbits/sec
[ 3] 40.0-41.0 sec 10.4 MBytes 87.0 Mbits/sec
[ 3] 41.0-42.0 sec 7.00 MBytes 58.7 Mbits/sec
[ 3] 42.0-43.0 sec 10.5 MBytes 88.1 Mbits/sec
[ 3] 43.0-44.0 sec 11.5 MBytes 96.5 Mbits/sec
[ 3] 44.0-45.0 sec 10.6 MBytes 89.1 Mbits/sec

```

Средняя скорость соединения в
ходе эксперимента составила
81 Mbps

Subflows

```

V = 0 bps
V = 0 bps
V = 54 bps
V = 0 bps
V = 0 bps

```

```

V = 6464780 bps
V = 7993920 bps
V = 4214204 bps
V = 12703974 bps
V = 6249792 bps

```

```

V = 6373940 bps
V = 12428426 bps
V = 6186204 bps
V = 12682778 bps
V = 2476904 bps

```

Extra flow

```

[ 5] local 10.0.0.3 port 5001 connected with 10.0.0.4 port 33624
[ ID] Interval      Transfer      Bandwidth
[ 5]  0.0-20.2 sec  89.1 MBytes  37.0 Mbits/sec

```

Node: h4

```

[ 4] 16.0-17.0 sec  5.62 MBytes  47.2 Mbits/sec
[ 4] 17.0-18.0 sec  5.75 MBytes  48.2 Mbits/sec
[ 4] 18.0-19.0 sec  5.62 MBytes  47.2 Mbits/sec
[ 4] 19.0-20.0 sec  5.75 MBytes  48.2 Mbits/sec
[ 4]  0.0-20.0 sec  89.1 MBytes  37.3 Mbits/sec

```