Многопоточные протоколы транспортного уровня

м.н.с. Степанов Евгений Павлович

Программа курса

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

Модели:

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

Примеры:

- HTTP3/QUIC
- Управление сетевыми ресурсами в Центрах Обработки Данных
- Обеспечение качества сервиса в сетях доставки контента
- Пропускная способность по требованию

Зачем нужны такие протоколы?

- Повышение отказоустойчивости
  - Если отказывает одна из линий, соединение остаётся работоспособным
- Увеличение пропускной способности
  - Соединение может использовать пропускные способности сразу нескольких линий
  - Снижение задержки на передачу
- Повышение утилизации сети
  - Балансировка нагрузки по нескольким маршрутам

В чём разница с балансировкой нагрузки на других уровнях TCP/IP?

- Потоки многопоточного соединения не обязательно идут разными маршрутами
  - В силу малой эффективности алгоритмов управления перегрузками при больших задержках выгодно создавать больше потоков
- Балансировка данных внутри одного транспортного соединения
  - L2/L3 балансировка не может разделять данные транспортного потока из-за проблемы нарушения порядка доставки
  - Балансирока на L5 требует изменения логики работы прикладных программ

Stream Control Transmission Protocol

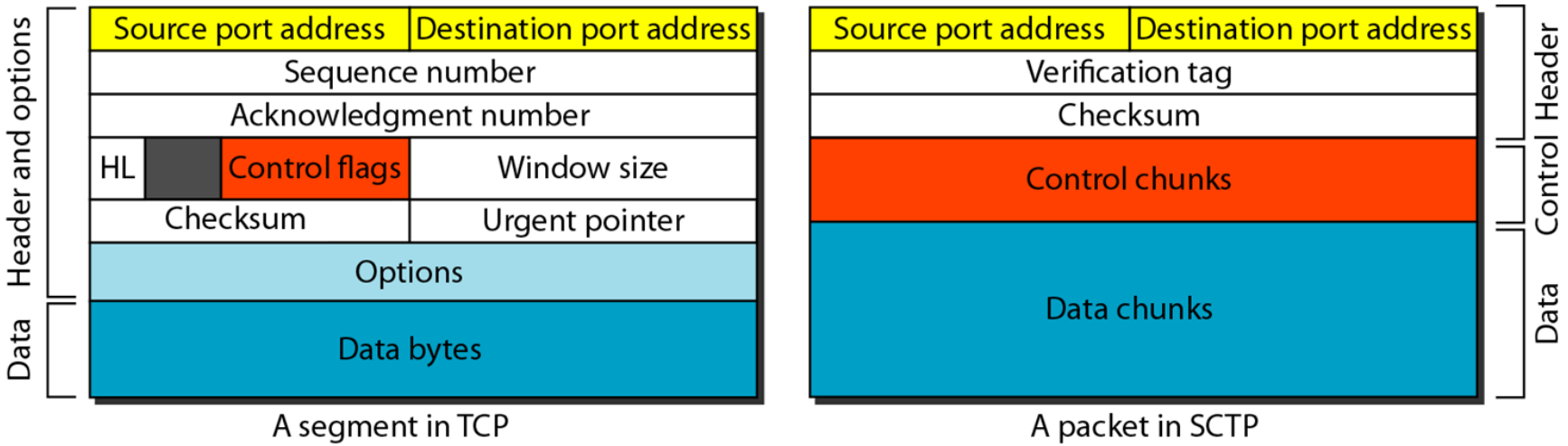
какие потребности не охватывает UDP и TCP?

- TCP
  - Абстракция потока байтов
  - Сохранение последовательности передачи
  - Нет поддержки Multi-homing
  - Head-of-Line Blocking
  - Подвержен атакам SYN-Flooding
- UDP
  - Ненадёжная передача данных
  - Нет встроенных механизмов управления перегрузкой потока

Stream Control Transmission Protocol

- Надёжная передача данных
- Множество потоков на каждое соединение
- Поддержка Multi-homing:
  - Поддержка хостов с несколькими адресами
  - Сохранение работоспособности при отказе
- Абстракция сообщений
- Возможность неупорядоченной доставки
- Встроенные механизмы управления перегрузкой (аналогичные TCP)

Сегменты TCP vs Пакеты SCTP



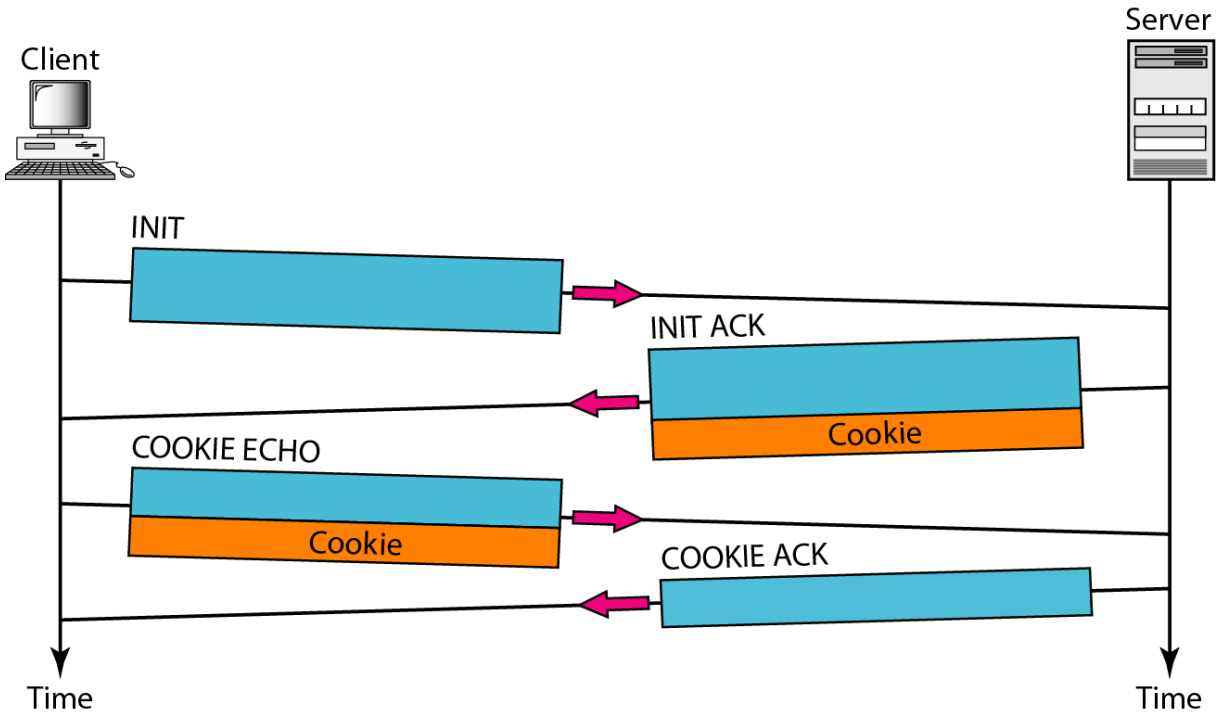
- Состоит из заголовка и множества чанков
- Чанки делятся на несколько типов
  - Payload Data, Acknowledgment, Heartbeat, etc
- Могут хранить как пользовательские данных, так и служебную информацию
- Служебные чанки всегда хранятся первыми

Типы чанков SCTP

Type	*Chunk*	*Description*
0	**DATA**	User data
1	**INIT**	Sets up an association
2	**INIT ACK**	Acknowledges INIT chunk
3	**SACK**	Selective acknowledgment
4	**HEARTBEAT**	Probes the peer for liveliness
5	**HEARTBEAT ACK**	Acknowledges HEARTBEAT chunk
6	**ABORT**	Aborts an association
7	**SHUTDOWN**	Terminates an association
8	**SHUTDOWN ACK**	Acknowledges SHUTDOWN chunk
9	**ERROR**	Reports errors without shutting down
10	**COOKIE ECHO**	Third packet in association establishment
11	**COOKIE ACK**	Acknowledges COOKIE ECHO chunk
14	**SHUTDOWN COMPLETE**	Third packet in association termination
192	**FORWARD TSN**	For adjusting cumulative TSN

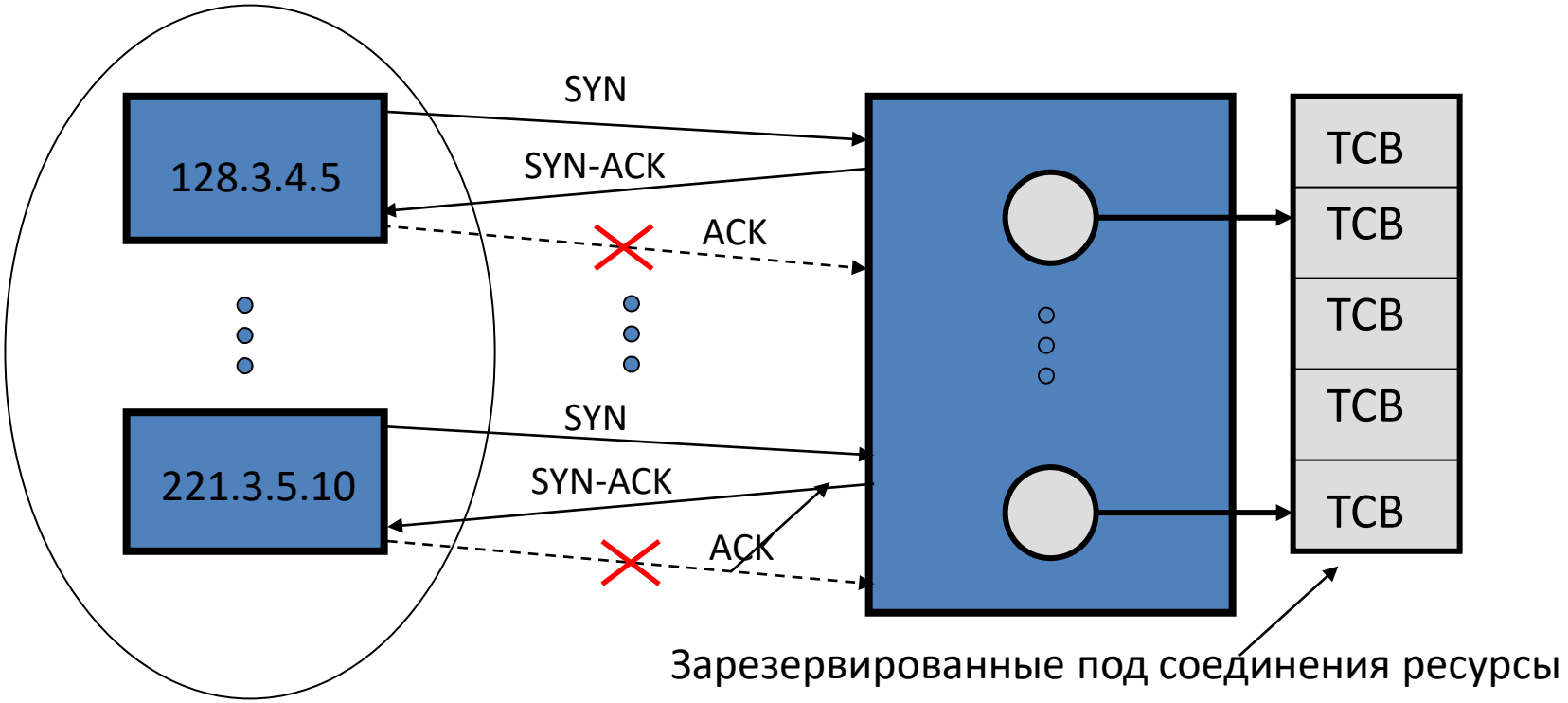
Установка SCTP соединения

В SCTP соединения называются ассоциациями



- Init & Init ACK могут включать в себя дополнительные адреса
- Init & Init ACK не могут использоваться для передачи данных
- Cookie Echo & Cookie Ack могут содержать чанки с данными

Атака типа SYN-Flood

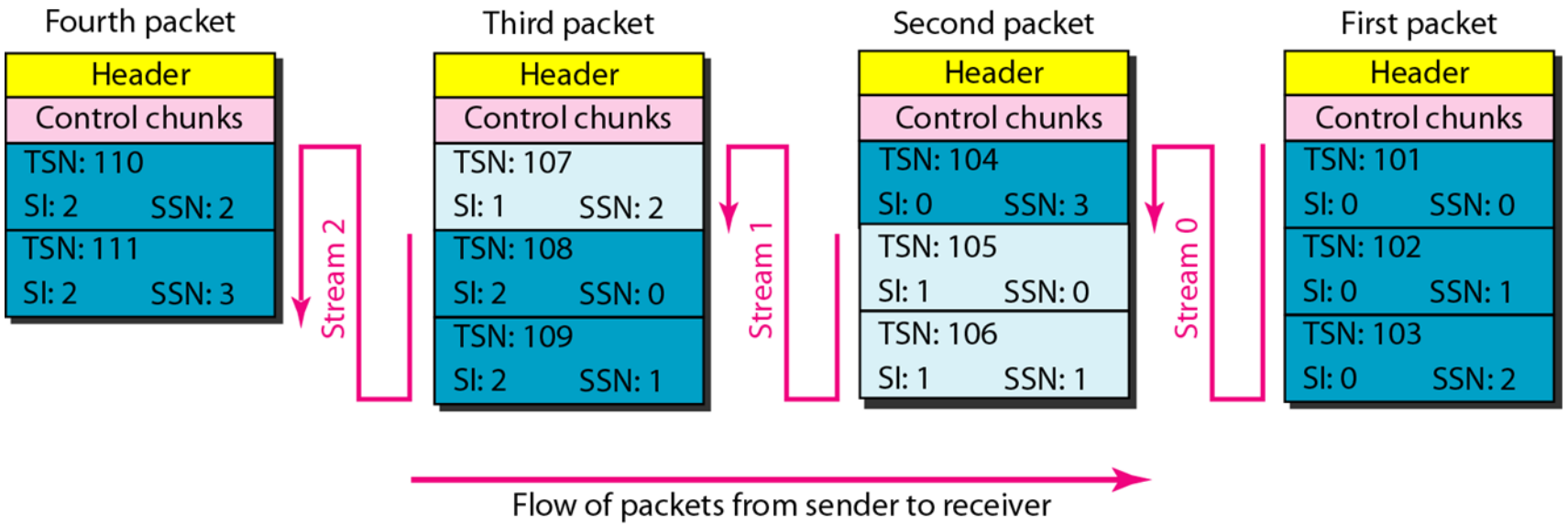


- Злоумышленники инициируют открытие большого количества соединений, но не подтверждают их установку
- Сервер резервирует под эти соединения ресурсы
- Обычне пользователи не могут подключиться к серверу, потому что для их обслуживания ресурсов не остаётся

Идентификация данных в SCTP

Каждый чанк с данными имеет три идентификатора:

- Transmission Sequence Number (TSN) – номер внутри ассоциации
  - Подтверждение доставки и исключение дублирования
- Stream Identifier (SI) – задействованный для передачи поток
- Stream Sequence Number (SSN) – идентификатор внутри потока
  - Решение проблемы Head of Line Blocking

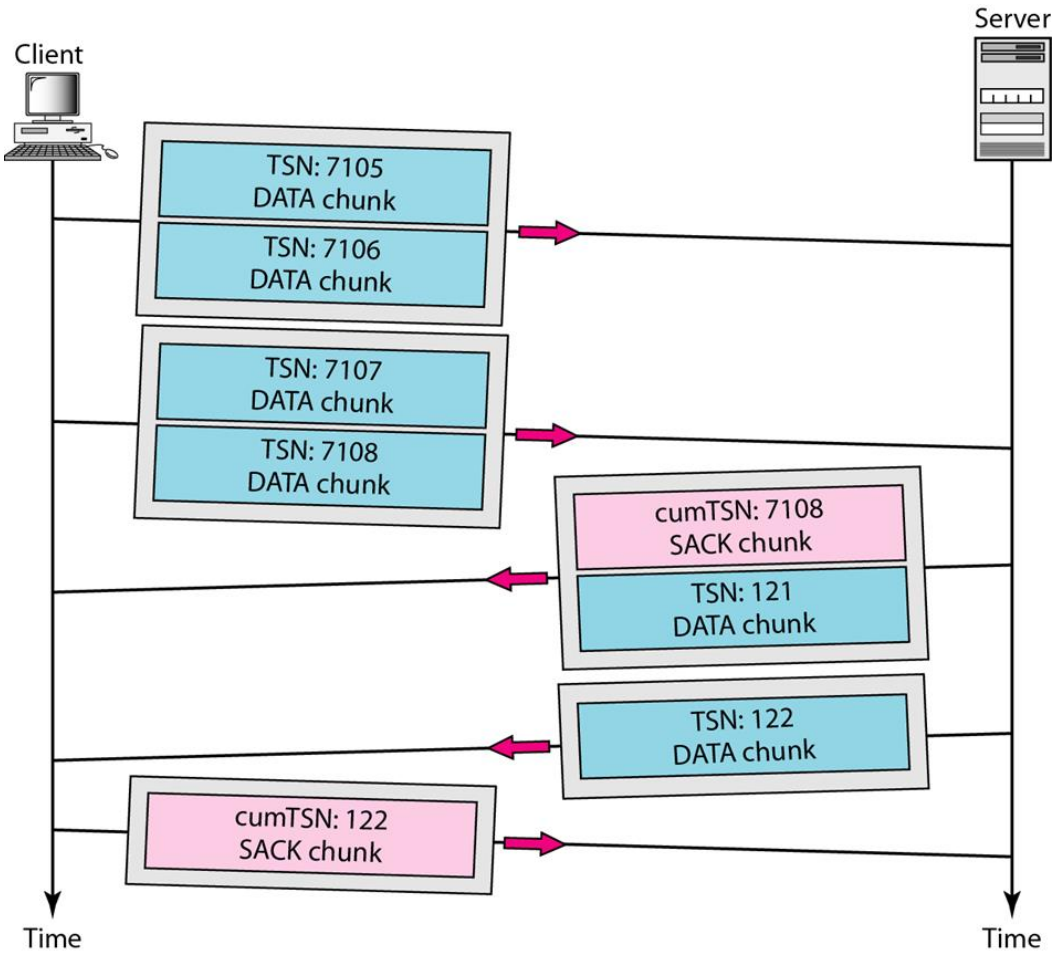


Подтверждение данных

- Подтверждаются только чанки с пользовательскими данными
  - Только у них есть идентификаторы
- Для этого используется чанк типа SACK
  - Содержит CumTSN – идентификатор последнего чанка, полученного по порядку, и
  - Множество Gap блоков, перечисляющих диапазоны с чанками, доставленными не по порядку
- Для подтверждения служебных чанков используются чанки других типов

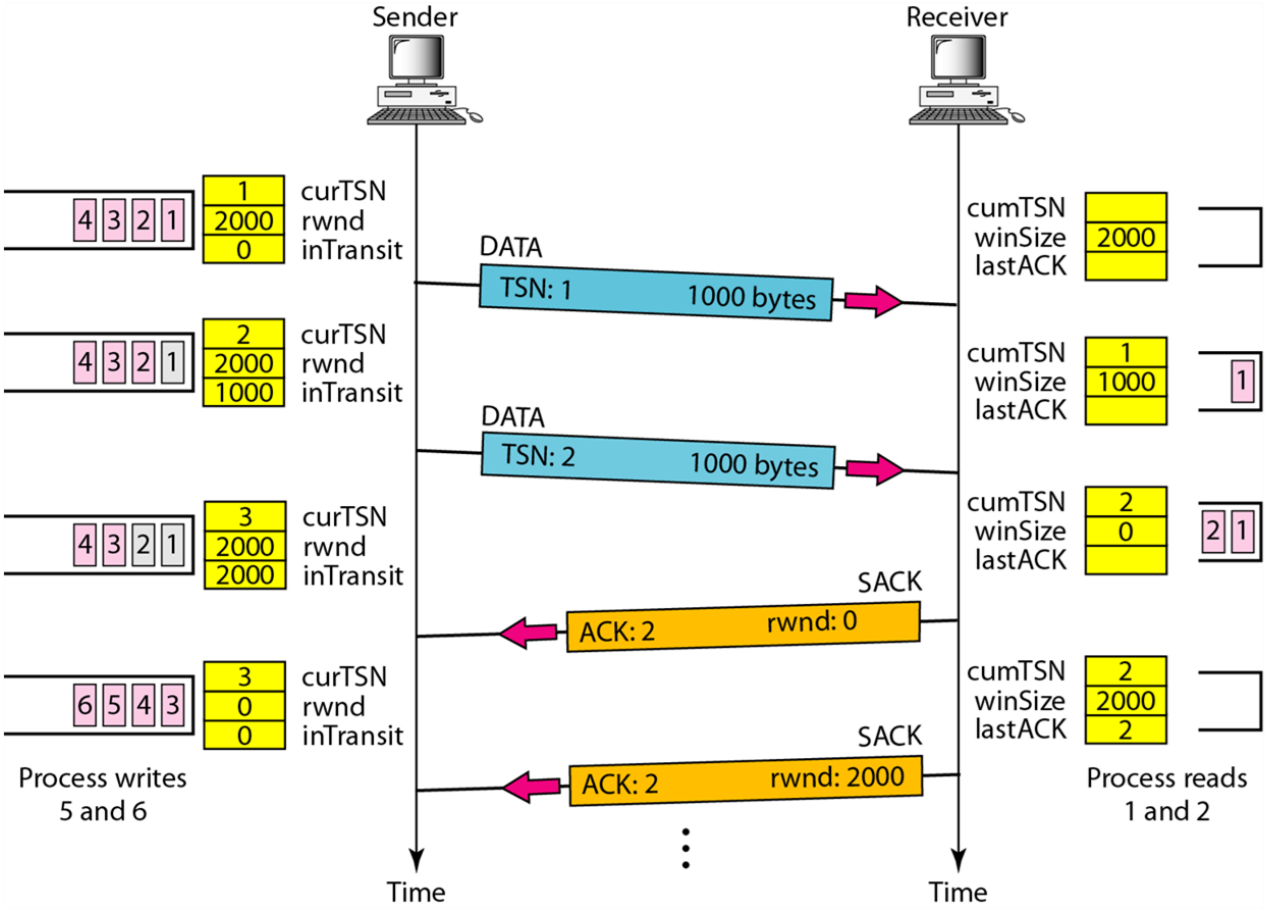
Пример передачи данных через установленное SCTP соединение

Подтверждение доставки



Пример передачи данных через установленное SCTP соединение

Управление перегрузкой получателя



Работа в сетях с Multihoming

- Абонент SCTP ассоциации может иметь сразу несколько адресов
- Один из них считается основным (*primary*)
- По умолчанию все данные отправляются на основной адрес (через *primary path)*
- В случае повторной передачи данные отправляются по другим путям
  - Если в сети возникла перегрузка, то повторная передача не усугубит ситуацию

Проблемы SCTP

- Данные передаются в несколько потоков только при потере, или при явном указании пользователя
- Можно ли передавать данные одновременно?
  - Concurrent Multipath Transfer using Stream Control Transmission Protocol (**CMT-SMTP**)
  - Multipath Transmission Control Protocol (**MPTCP**)

SCTP vs MPTCP

SCTP

- Многопоточный протокол
- Поддержка multihoming
- Поддержка отказа потока
- Надёжная доставка

- Передача сообщений
- Возможность организации неупорядоченной доставки сообщений

- Необходимо изменение логики работы программы

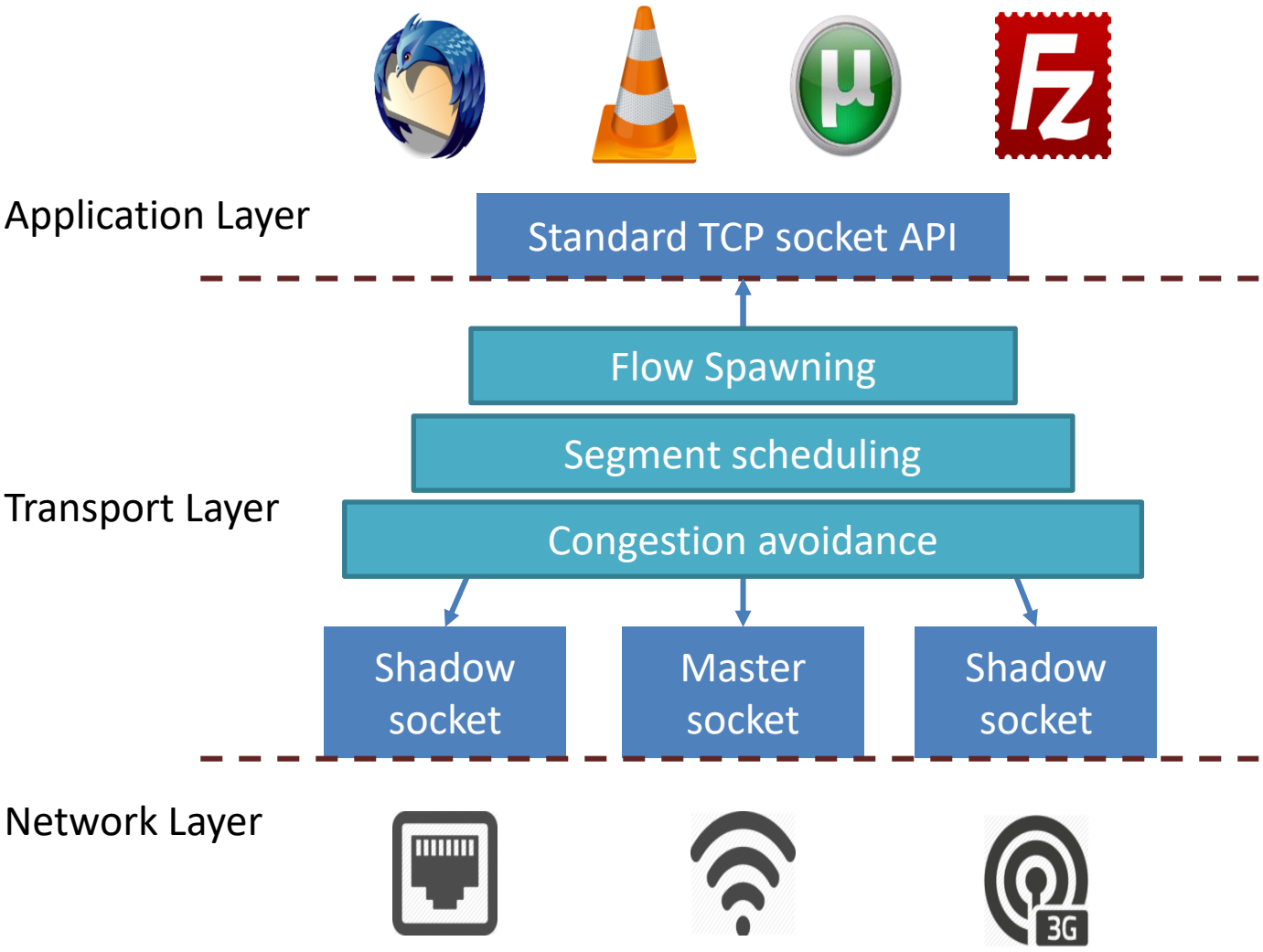
MPTCP

- Многопоточный протокол
- Поддержка multihoming
- Поддержка отказа потока
- Надёжная доставка

- Автоматрическая балансировка по потокам
- Обратная совместимость со стандартным TCP

- Требует внедрения нового функционала в стек ОС

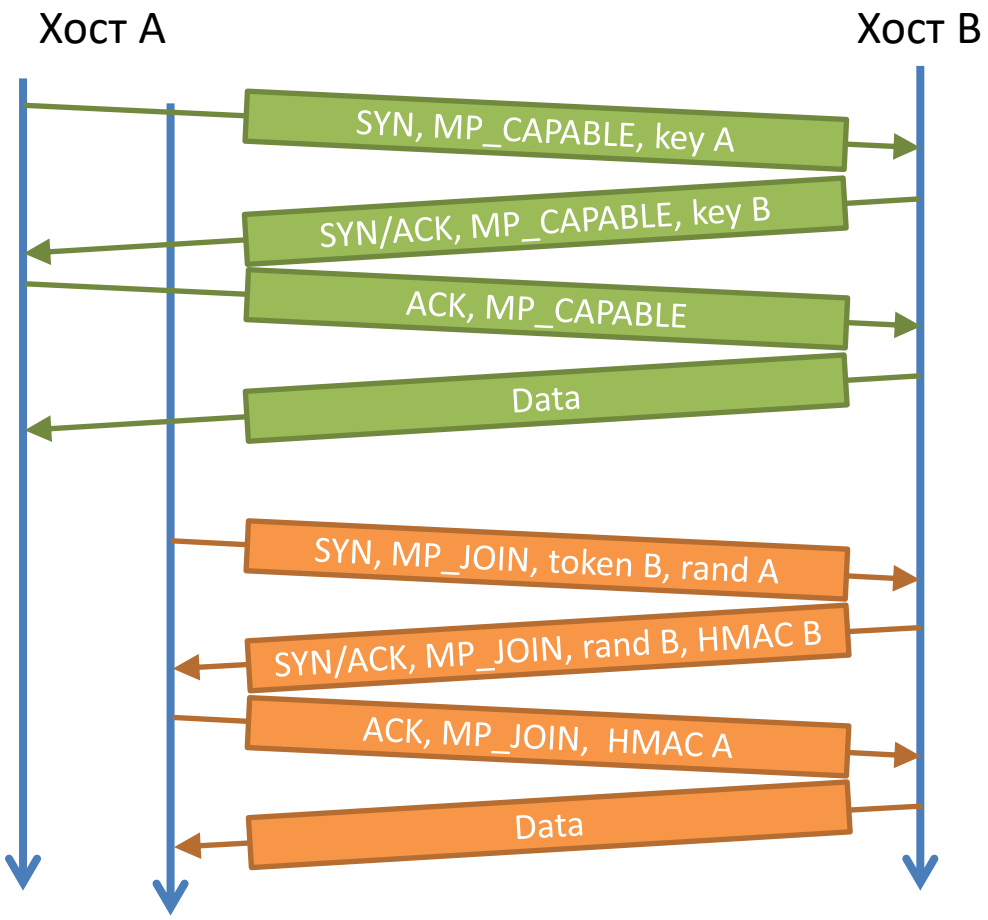
Multi Path TCP (MPTCP)



Организация MPTCP

- Протокол является расширением TCP и использует стандартный TCP socket API
  - Не требует изменения логики приложений
- Для синхронизации используются опциональные поля заголовка TCP
  - Опасность потери данных
  - Накладные расходы
- Потоки могут добавляться и удаляться динамически в процессе работы протокола

Установка MPTCP соединения



- Как открыть MPTCP вместо обычного TCP?
  - Опция MP_CAPABLE
- Что будет, если middlebox вырежет опцию MP_CAPABLE из SYN или SYN/ACK?
  - Сработает Fallback на TCP
- Как добавить поток к существующему соединению?
  - Опция MP_JOIN
  - Идентификация соединения по ключу
- Как не позволить злоумышленнику подключиться к открытому соединению?
  - Аутенрификация HMAC

MPTCP: Управление Перегрузкой

- Принцип справедливости требует уважать однопоточные TCP соединения
  - На алгоритмы управления перегрузкой накладываются дополнительные ограничения
  - Для большинства популярных алгоритмов управления перегрузкой есть MPTCP версии

MPTCP: автоматическое планирование пакетов по потокам

- Round Robin
  - Плохо работает на потоках разного качества
- Минимальное время доставки
  - Приводит к проблеме увеличения времени отклика соединения и переполнению буфера получателя
- Сохранение изначальной последовательности доставки пакетов на стороне получателя
  - Необходим сбор и анализ большого количества статистической информации

MPTCP: управление количеством потоков

- Количество открытых потоков должно соотвествовать состоянию сети
  - Если слишком много – накладные расходы приведут к падению производительности
  - Если слишком мало – сеть работает неоптимально, скорость может увеличиться
- Существующие подходы:
  - Открывать N потоков на старте соединения
  - Открывать по 1 потоку на каждом интерфейсе
  - Изменять количество потоков динамически

MPTCP: маршрутизация потоков

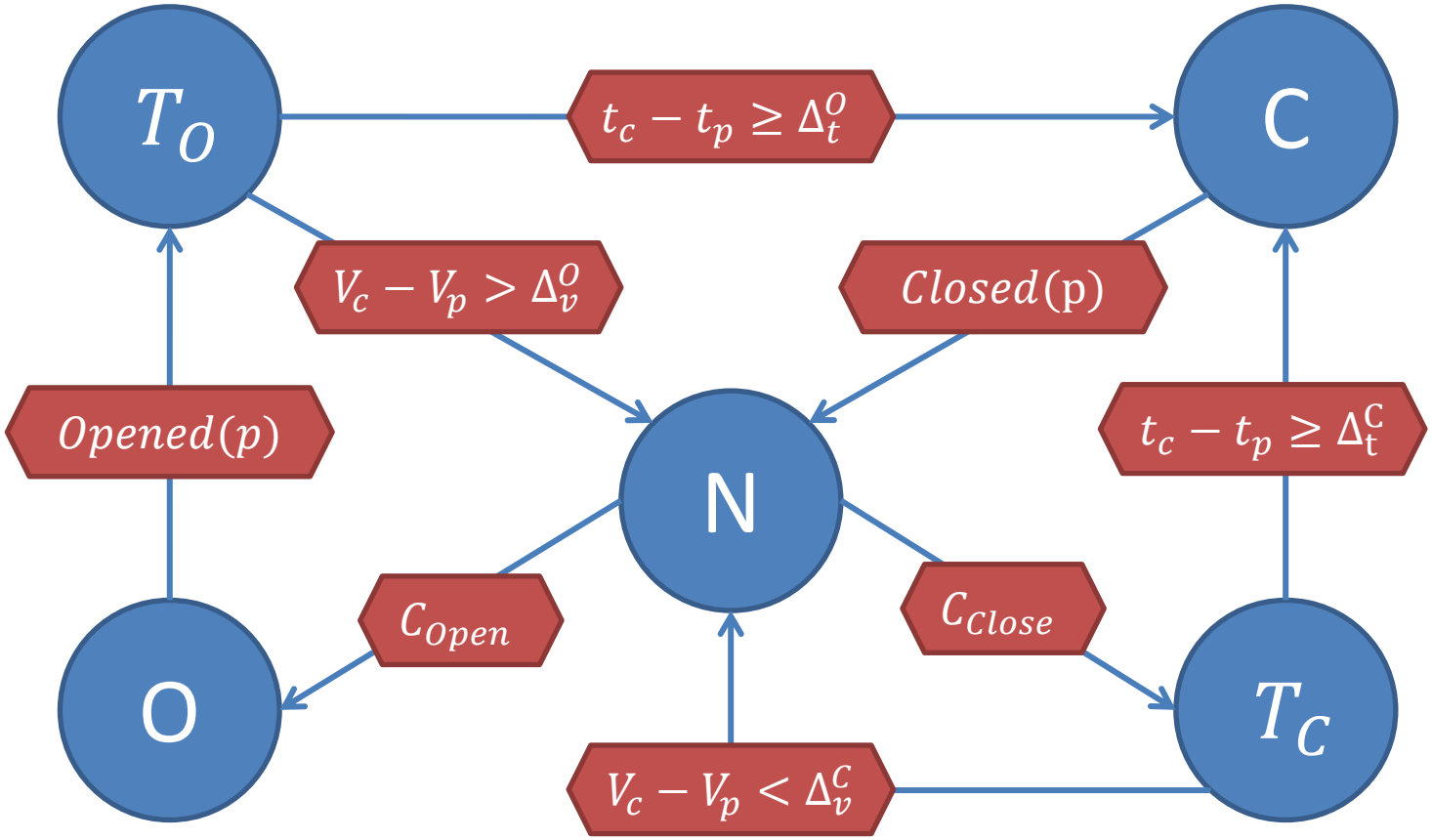
- Базовая эвристика – разные потоки соединения должны проходить по непересекающимся маршрутам
- В традиционных сетях управление маршрутизацией на уровне отдельных потоков затруднительно
  - ECMP распределяет потоки случайно
- В программно-конфигурируемых сетях приводит к увеличению времени отклика и чрезмерной нагрузке на контроллер
  - Flow (De)Multiplexing Protocol

Где используется MPTCP?

- Центры обработки данных
  - Копирование больших объёмов данных
  - Инфраструктура с большой избыточностью
- Удалённые населённые пункты
  - Несколько медленных каналов
  - Ненадёжная связь
- Мобильные телефоны
  - Несколько сетевых интерфейсов
  - Поддерживается IOS начиная с версии 7

Flow (De)Multiplexing Protocol (FDMP)

Алгоритм адаптации количества потоков



V_c - current speed

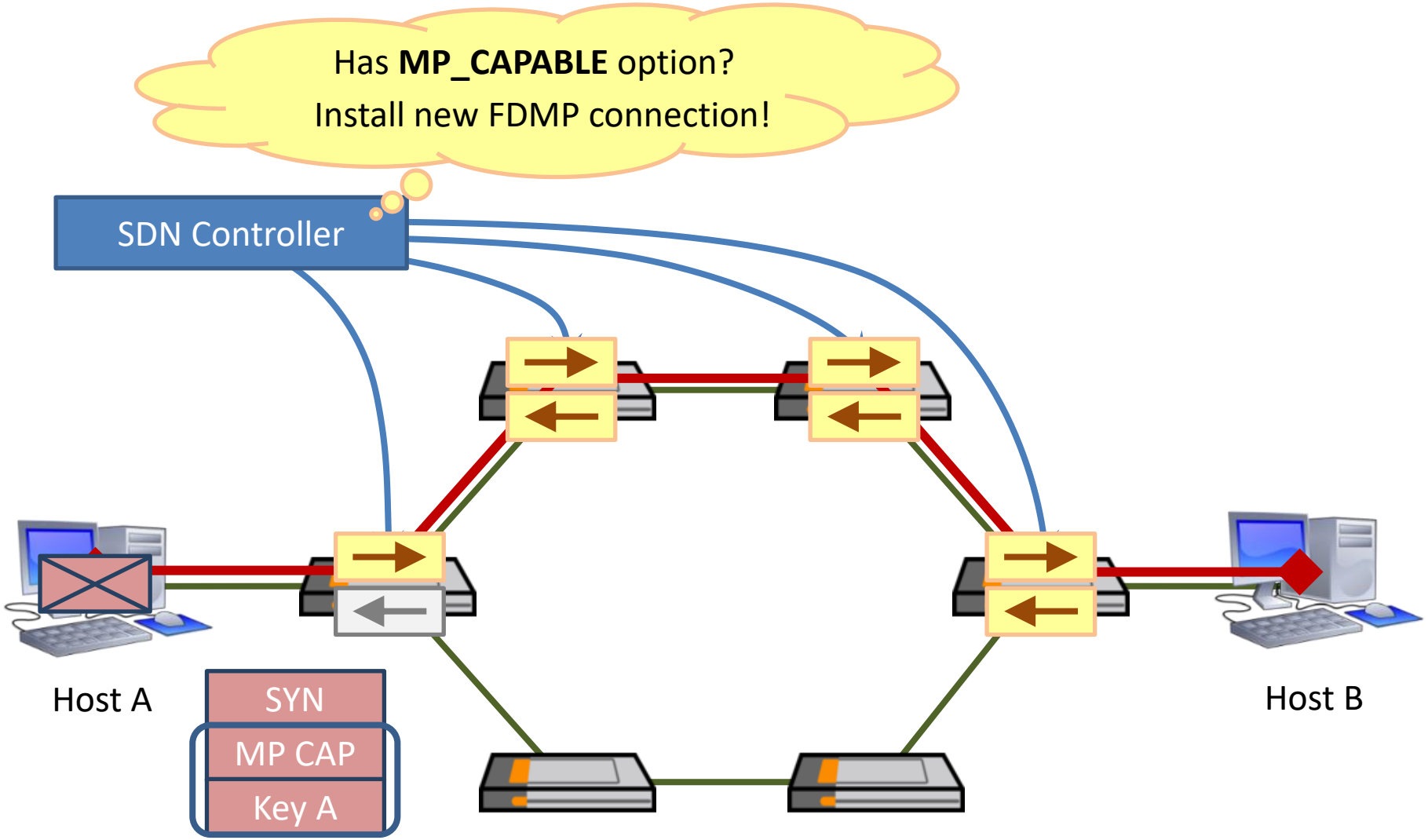
V_p - speed at previous phase

t_c - current time

t_p - time at previous phase

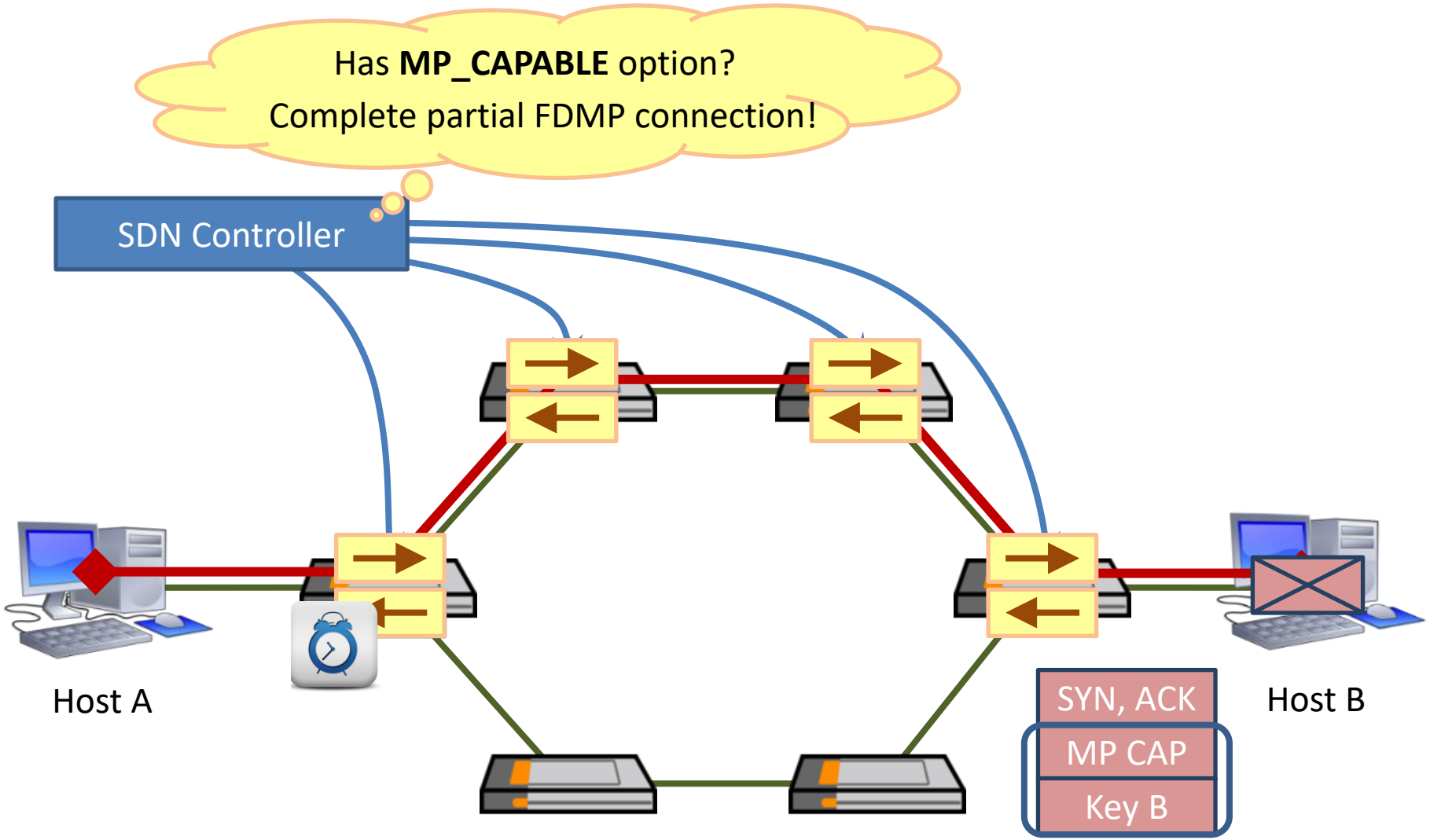
Маршрутизация потков FDMP в ПКС

Установка нового соединения



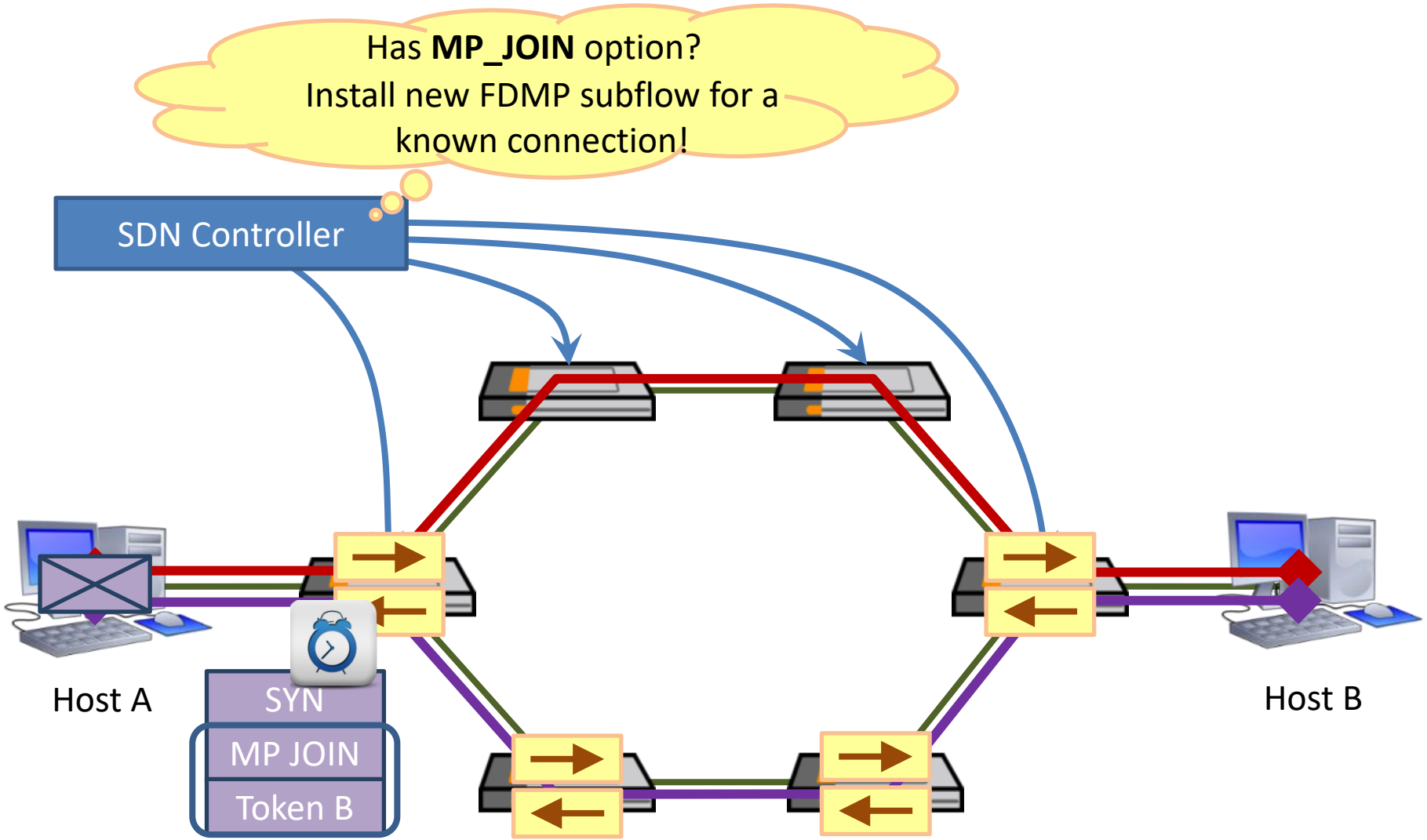
Маршрутизация потков FDMP в ПКС

Установка нового соединения



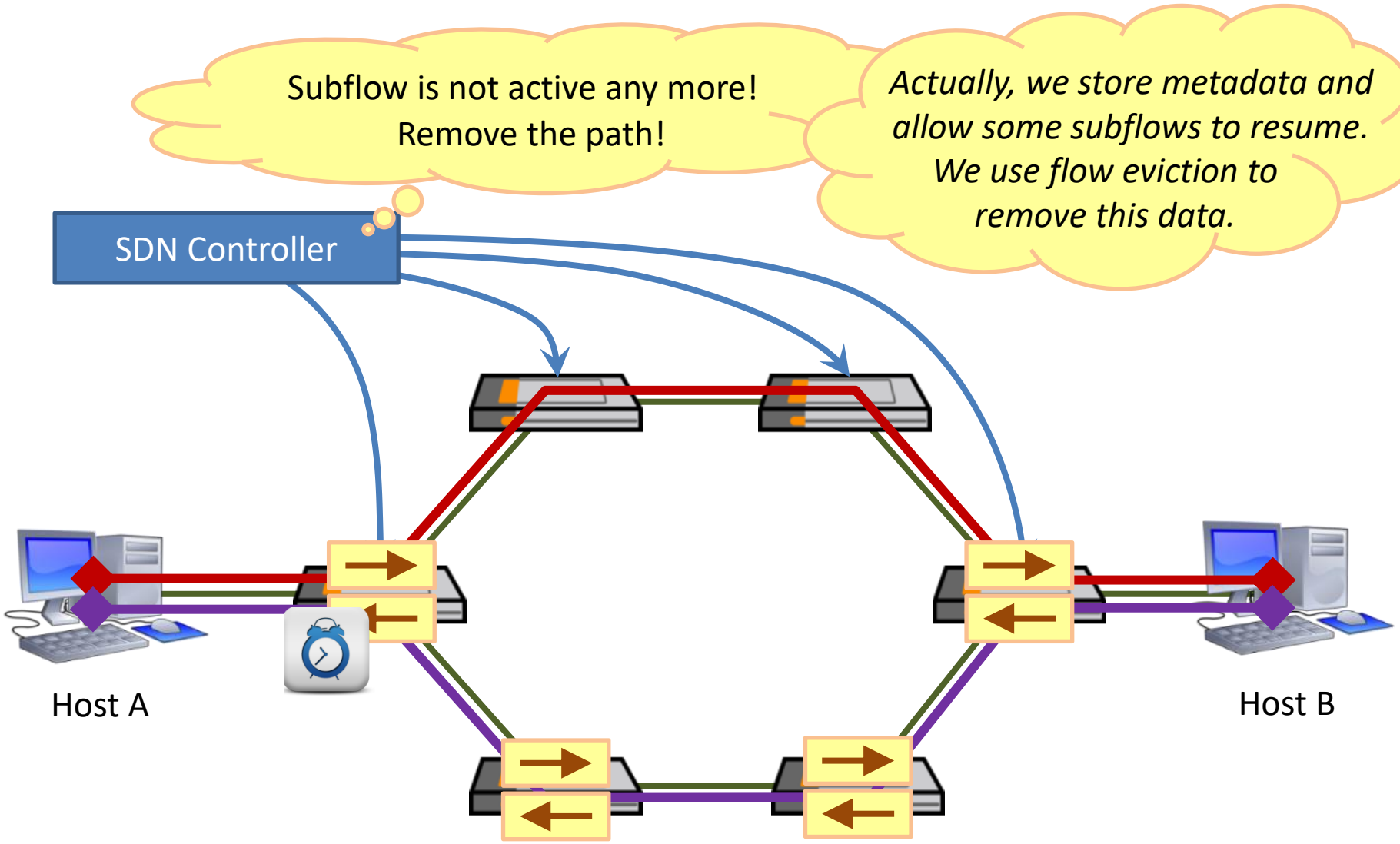
Маршрутизация потков FDMP в ПКС

Установка нового соединения



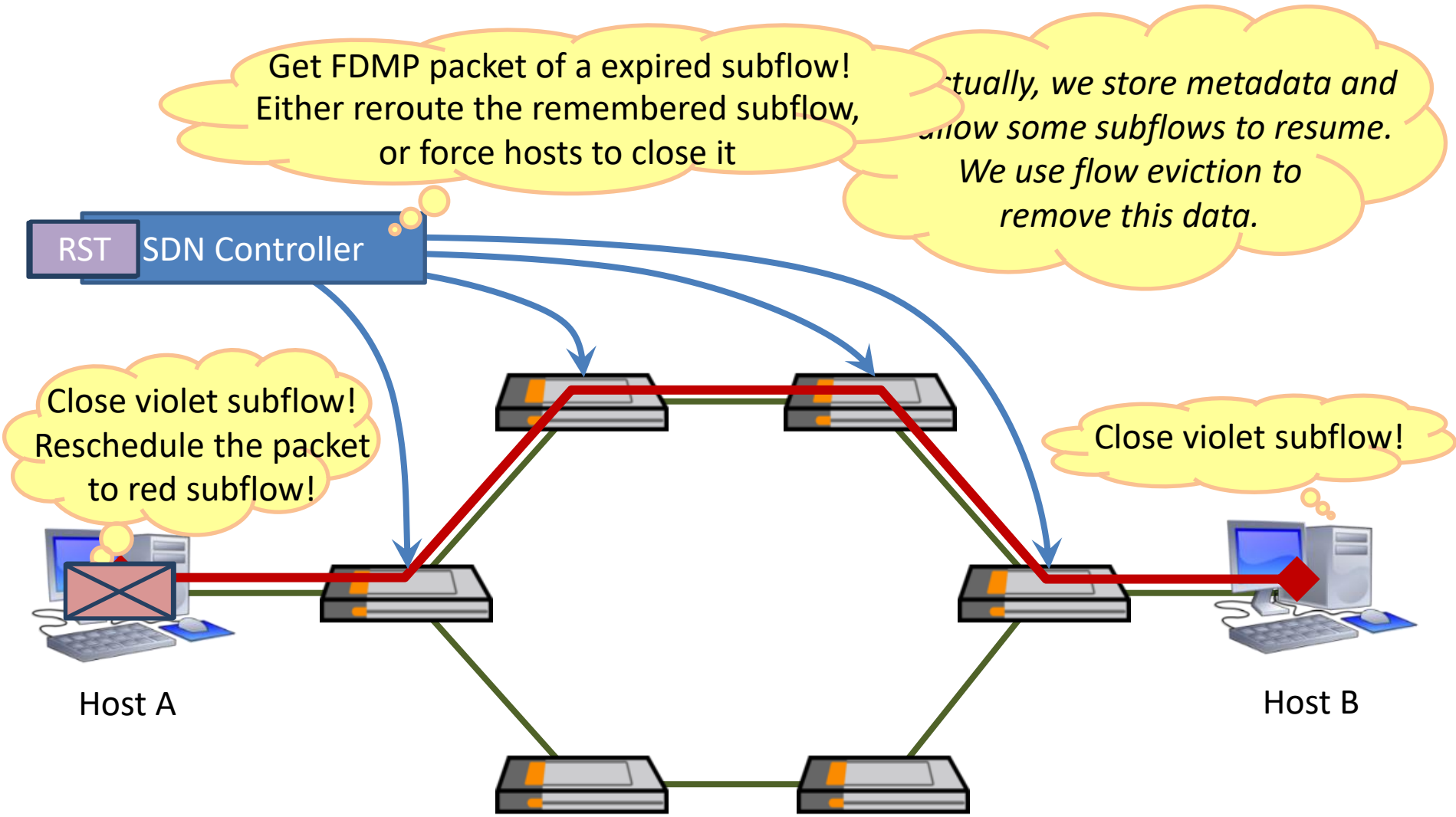
Маршрутизация потков FDMP в ПКС

Добавление потока к открытому соединению



Маршрутизация потков FDMP в ПКС

Добавление потока к открытому соединению



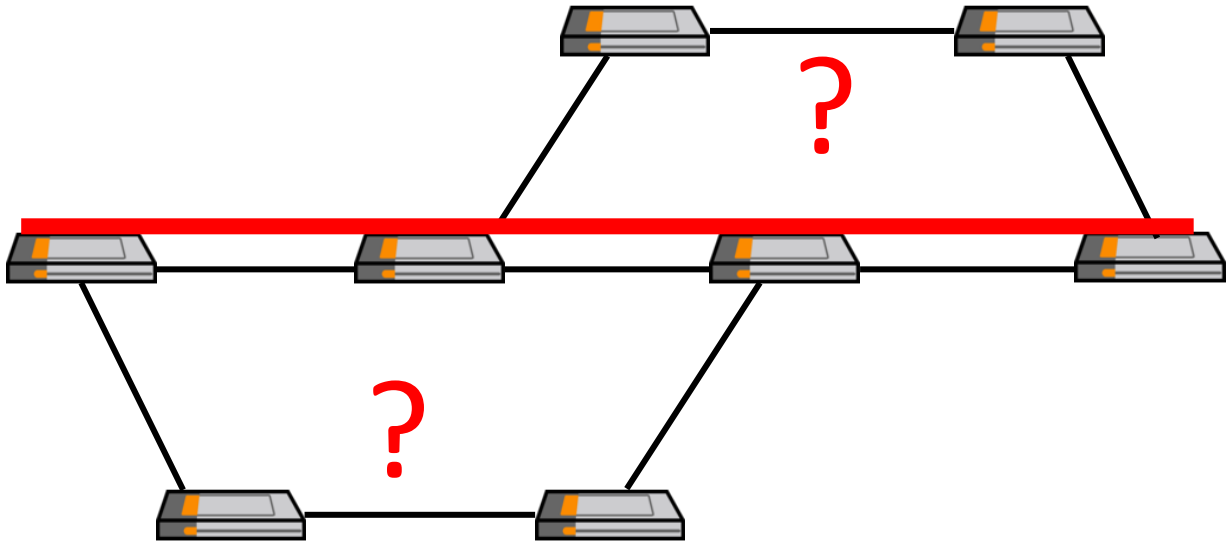
Массовая многопоточность

Маршрутизация

- Недостатки реактивной маршрутизации:
  - Огромный объем правил
  - Интенсивный трафик на контроллер
  - Задержка на установление соединения
- Переход на проактивную маршрутизацию
  - Нужно обеспечить распределение подпотоков по разным маршрутам (идея с DSCP, IPv6)

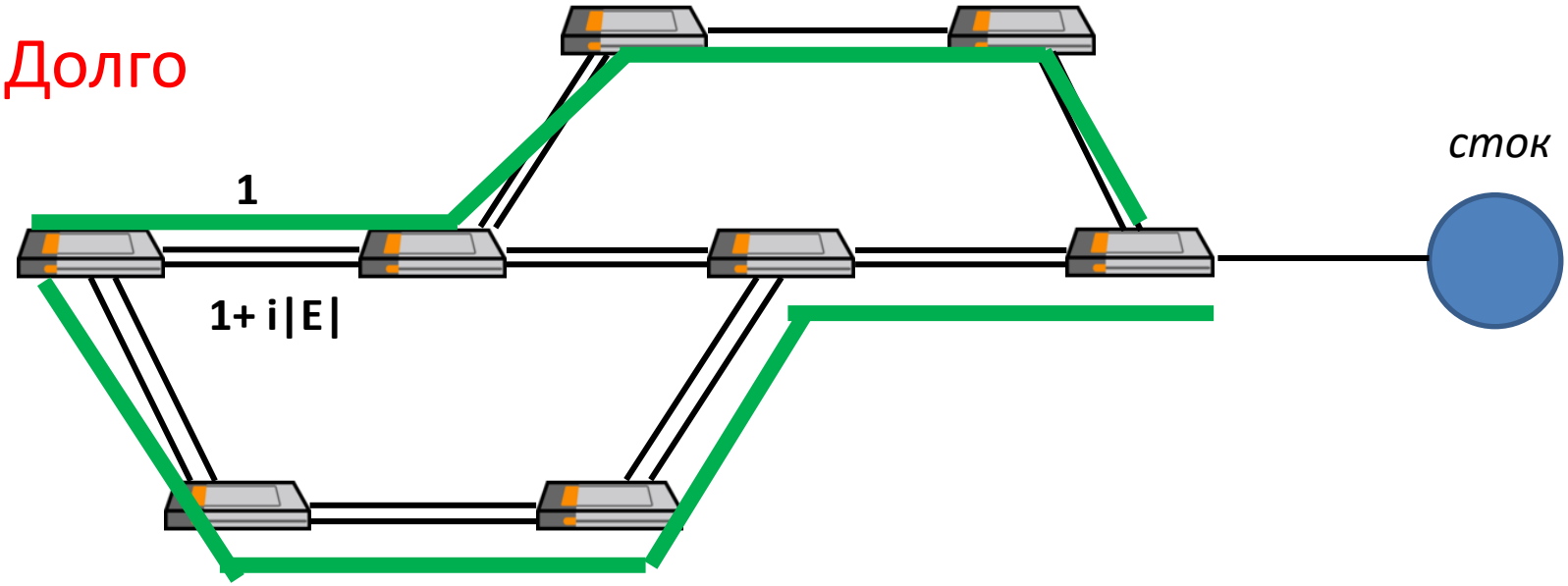
Маршрутизация

- Greedy Shortest Path First (GSPF)
  - Легко реализовать
  - Не дает оптимальных маршрутов

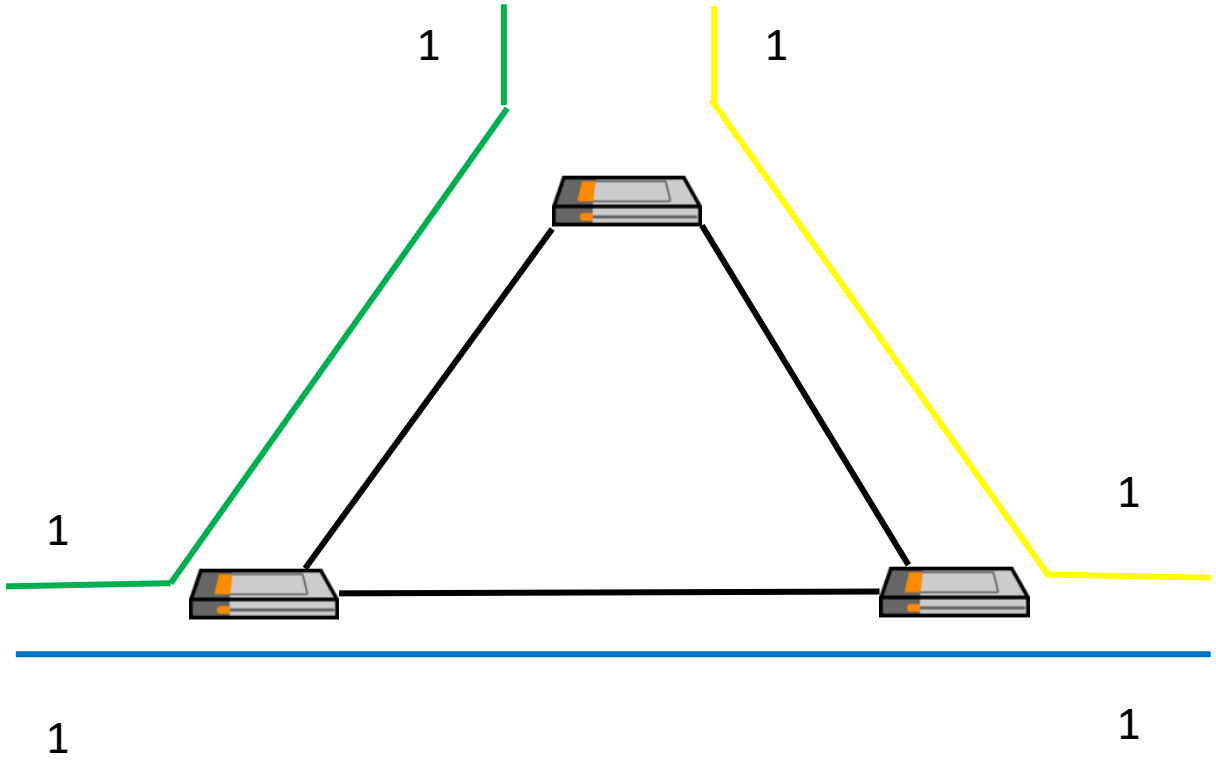


Маршрутизация

- Min-Cost Max-Flow(MCMF)
  - Сводим задачу к нахождению максимального потока минимальной стоимости (дает оптимальное решение)
  - Долго

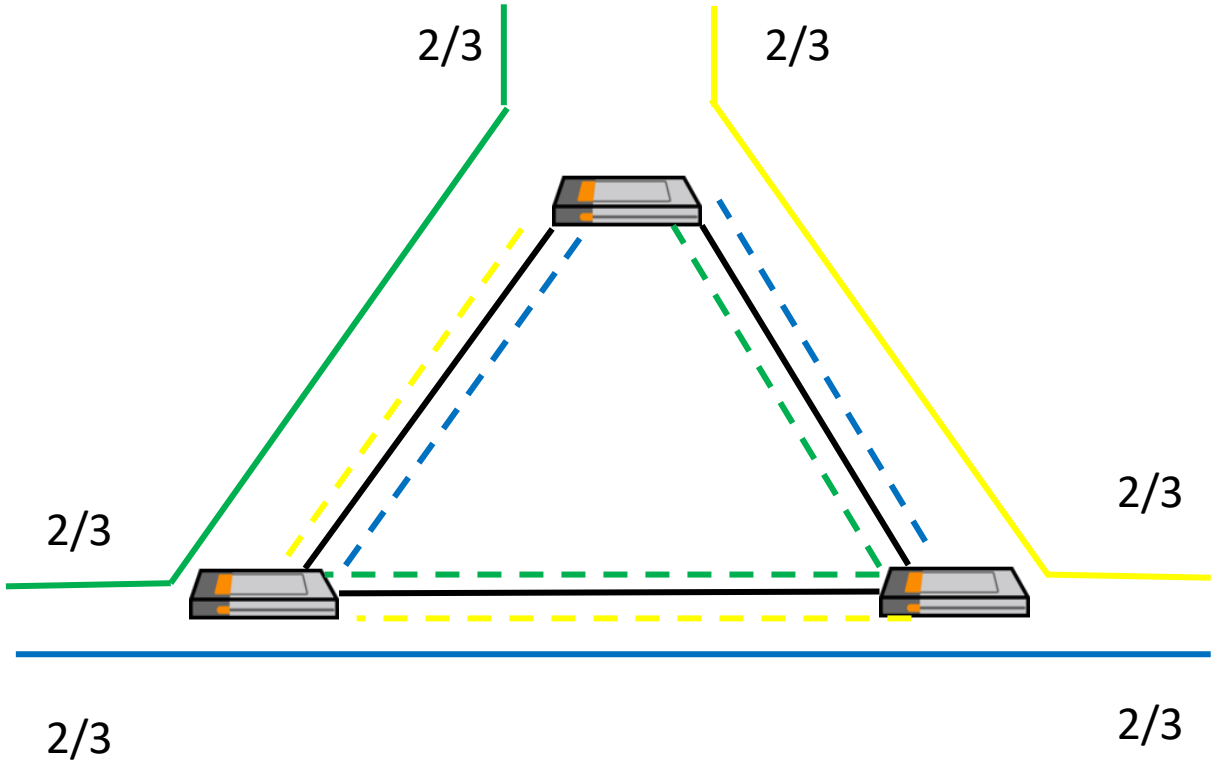


Массовая многопотчность



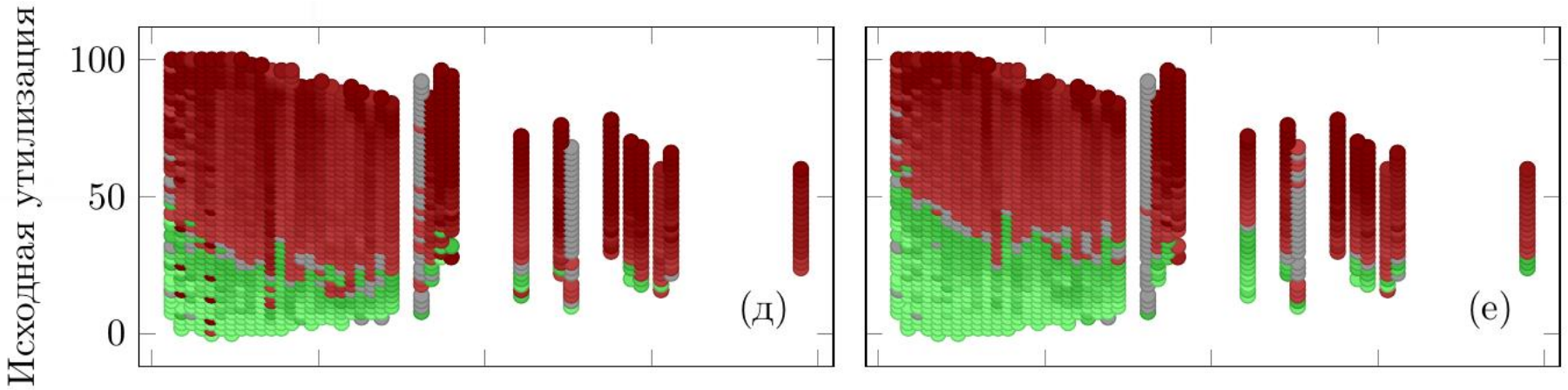
Общая пропускная способность сети - 6

Массовая многопотчность

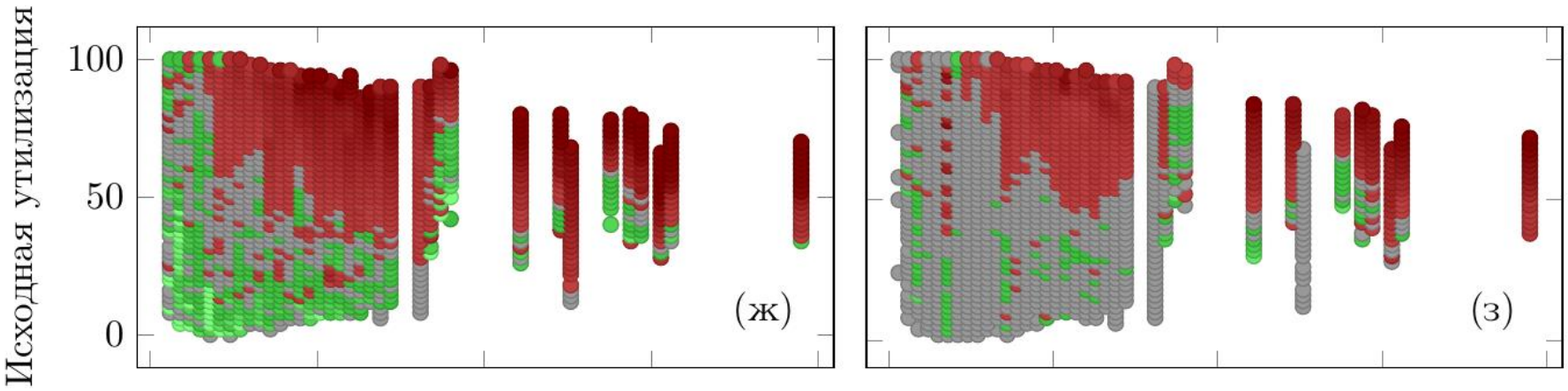


Общая пропускная способность сети - 4

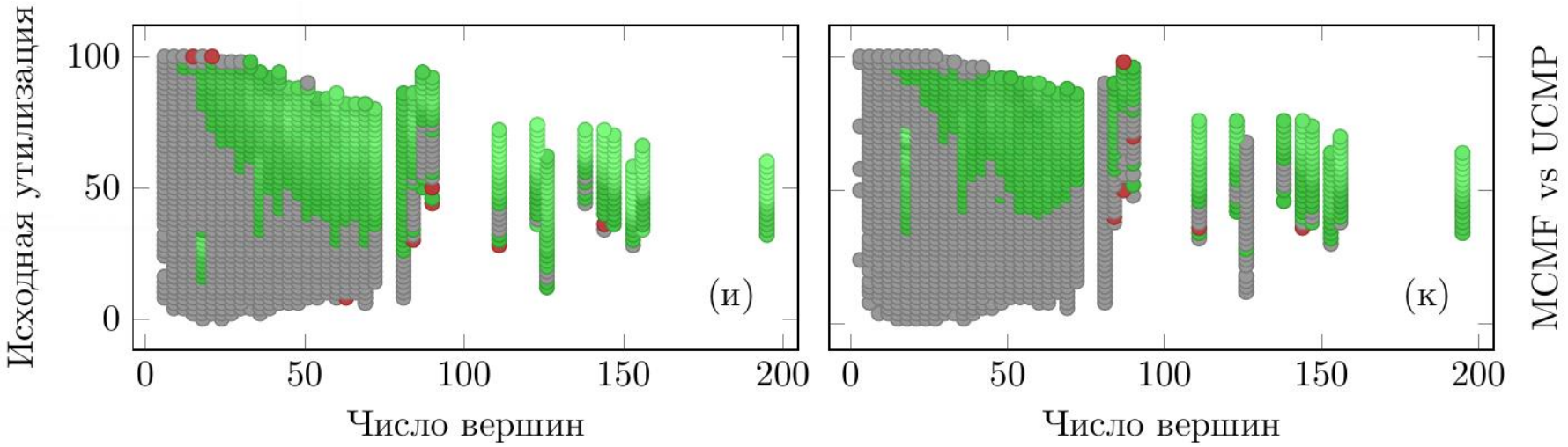
GSPF vs ECMP



MCMF vs GSPF



MCMF vs UCMP


Исходная утилизация
100
50
0
0
50
100
150
200
Число вершин
(и)
0
50
100
150
200
Число вершин
(к)
MCMF vs UCMP

Резюме

- Многопоточная маршрутизация на транспортном уровне – перспективное направление для исследований
  - SCTP октрывает много возможностей, но требует изменения логики приложений
  - У стандартных TCP и UDP есть недостатки
  - MPTCP интегрируется с существующими прилжоениями относительно просто
  - Гибкость распределения потоков и реакция на динамическую нагрузку?

Программа курса

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

Модели:

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

Примеры:

- HTTP3/QUIC
- Управление сетевыми ресурсами в Центрах Обработки Данных
- Обеспечение качества сервиса в сетях доставки контента
- Пропускная способность по требованию