

Многопоточные протоколы транспортного уровня

Чемерицкий Евгений Викторович

к.ф.-м.н.

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
 - Данных можно передать быстрее
 - Перераспределение отправки во времени
- 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
 - Балансировка на L5 требует изменения логики работы прикладных программ

Stream Control Transmission Protocol

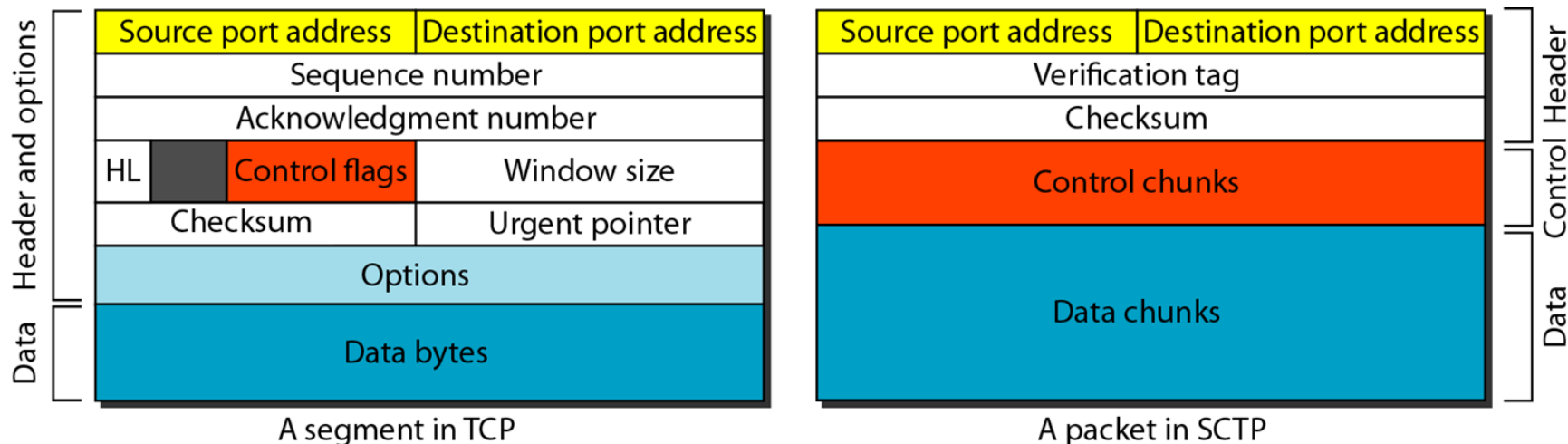
какие потребности не охватывает UDP и TCP?

- 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



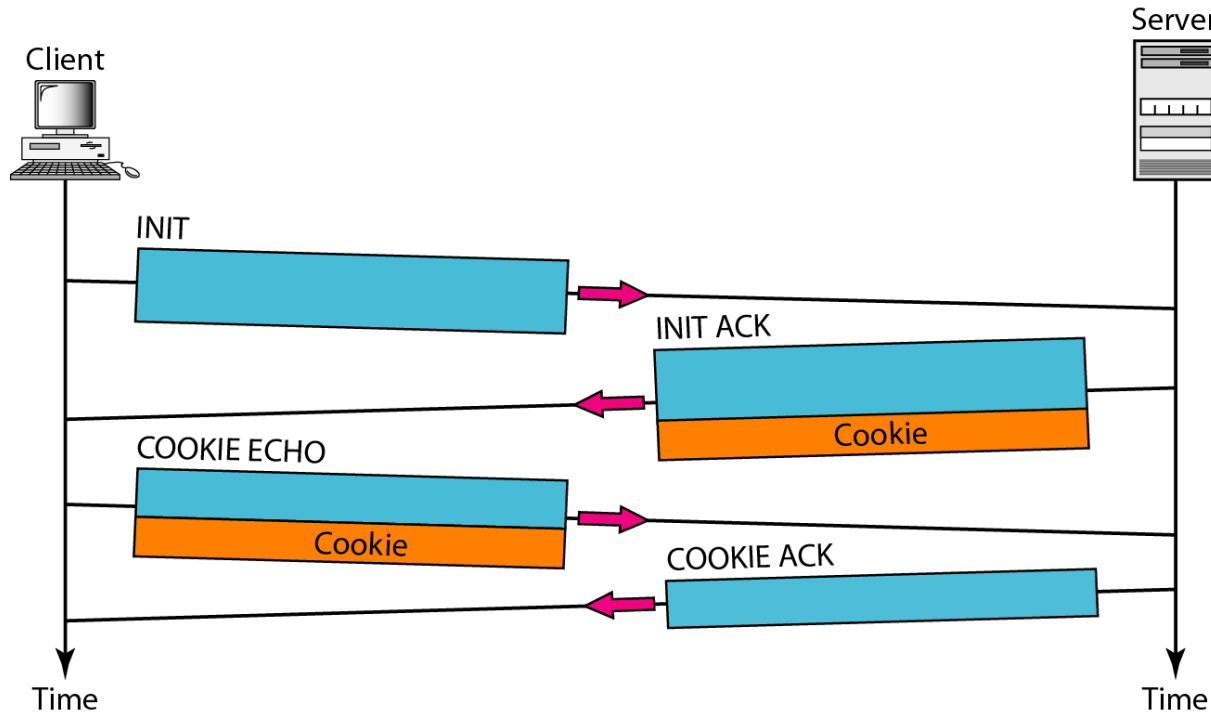
- Состоит из заголовка и множества чанков
- Чанки делятся на несколько типов
 - Payload Data, Acknowledgment, Heartbeat, etc
- Могут хранить как пользовательские данных, так и служебную информацию
- Служебные чанки всегда хранятся первыми

Типы чанков SCTP

<i>Type</i>	<i>Chunk</i>	<i>Description</i>
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 liveness
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

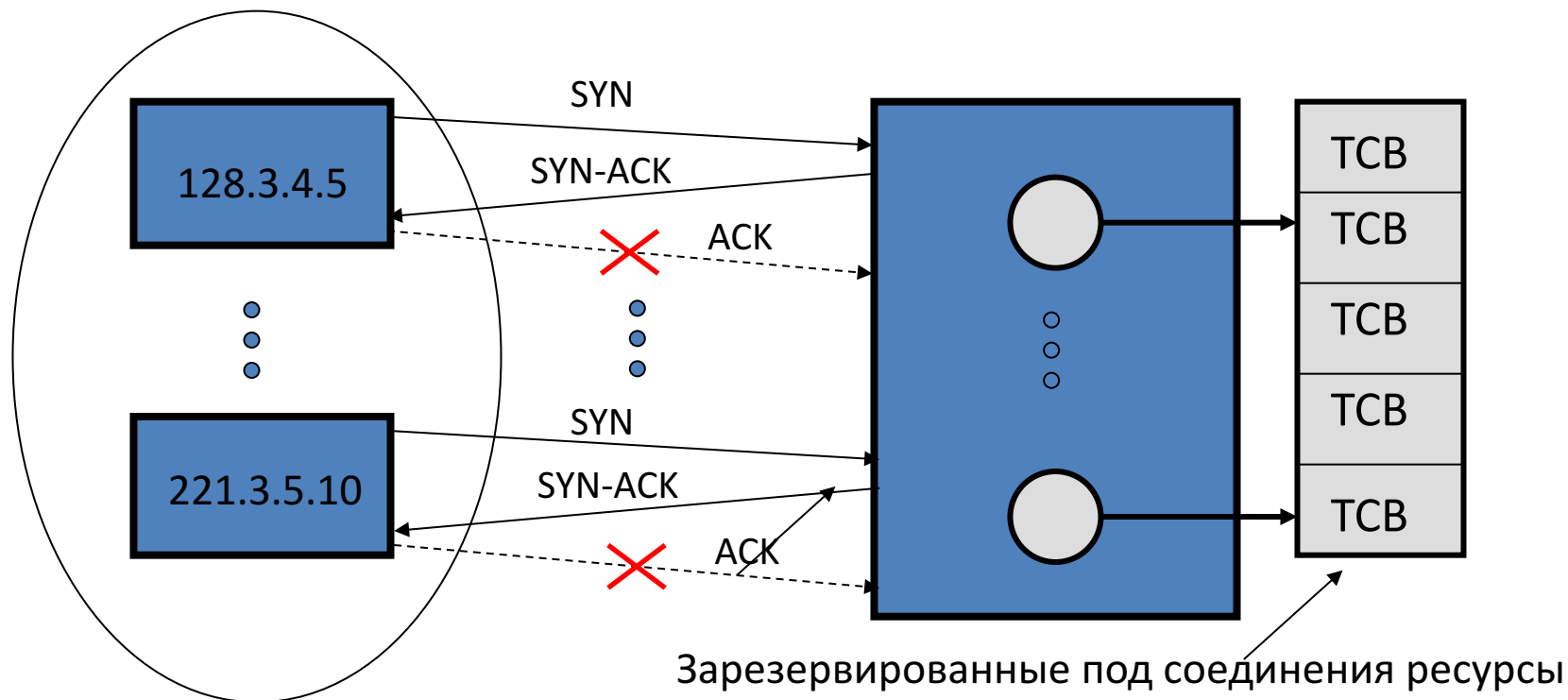
Установка SCTP соединения

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

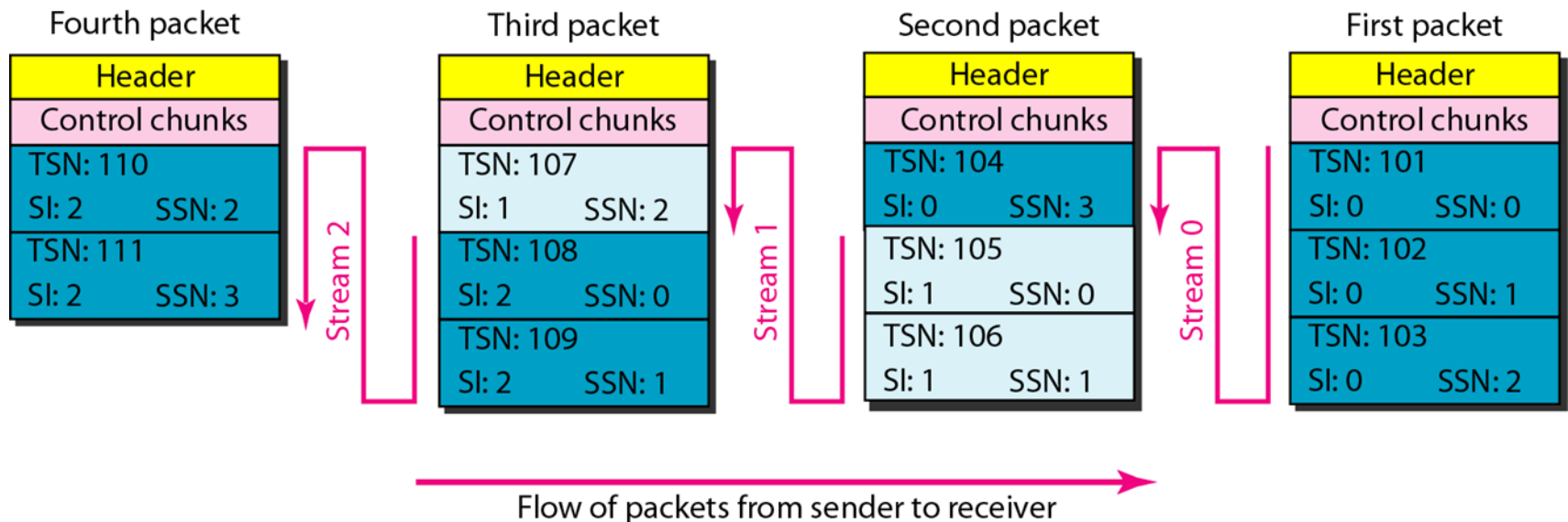


- Злоумышленники инициируют открытие большого количества соединений, но не подтверждают их установку
- Сервер резервирует под эти соединения ресурсы
- Обычно пользователи не могут подключиться к серверу, потому что для их обслуживания ресурсов не остаётся

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

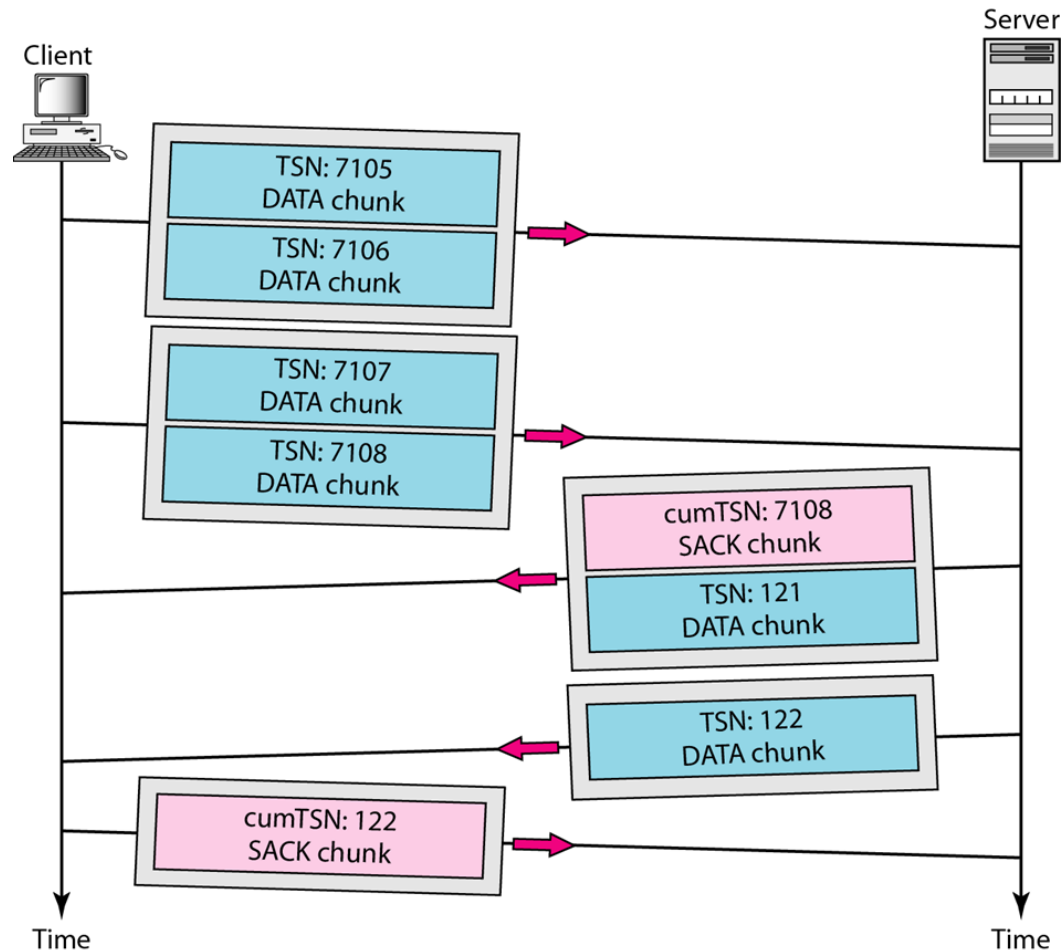


Подтверждение данных

- 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

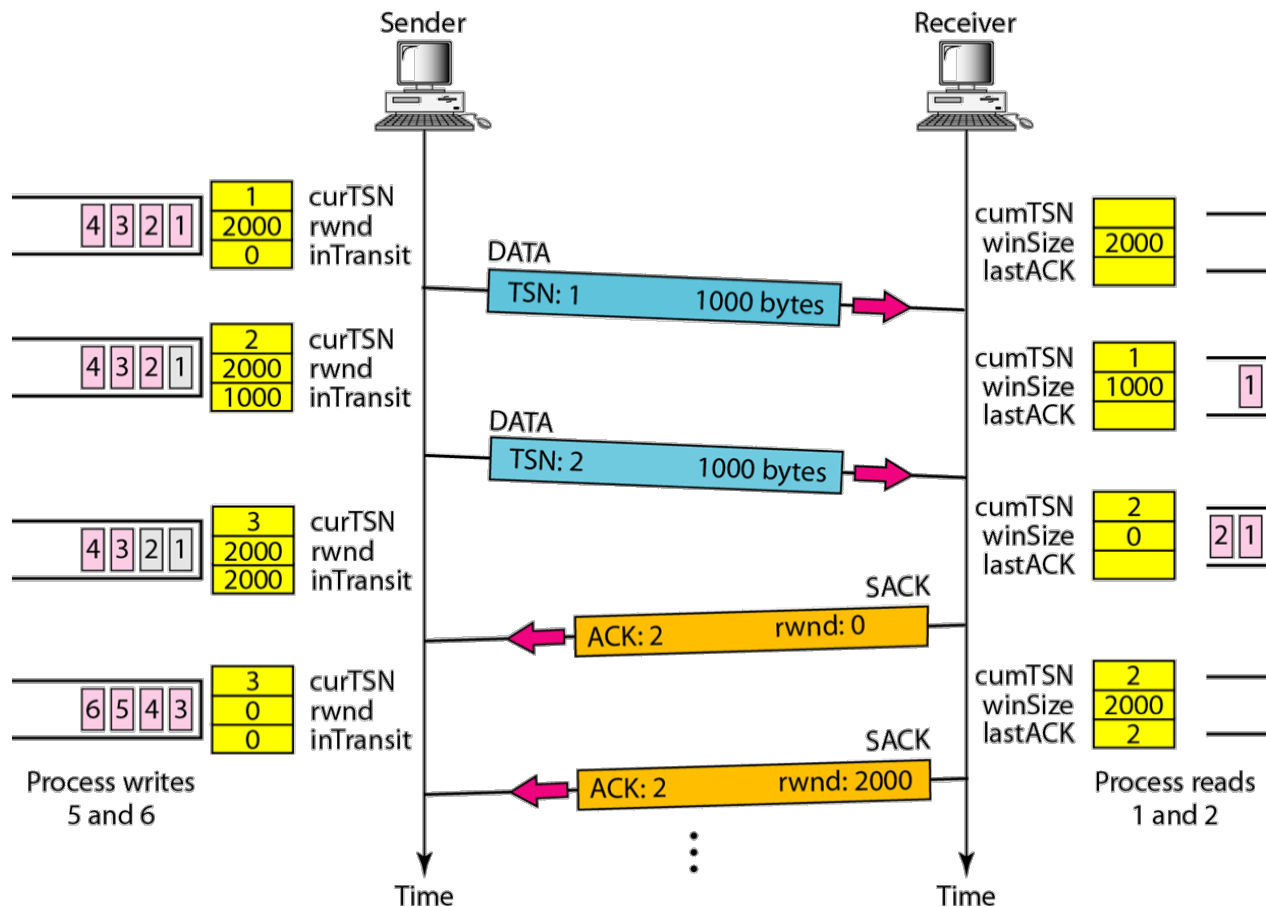
Пример передачи данных через установленное SCTP соединение

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

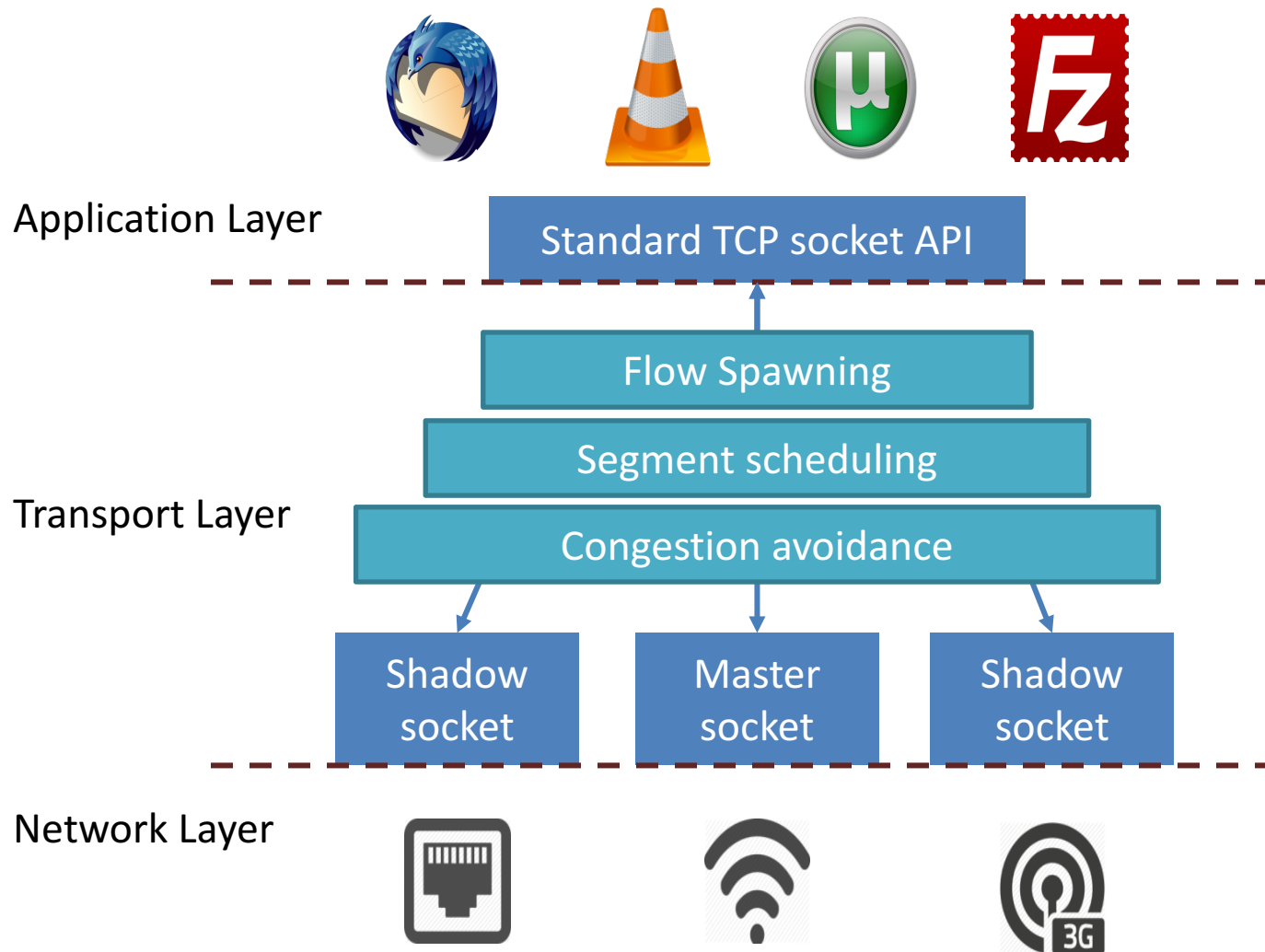
SCTP

- 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
- Автоматическая балансировка по потокам
- Обратная совместимость со стандартным TCP
- Требуется внедрения нового функционала в стек ОС

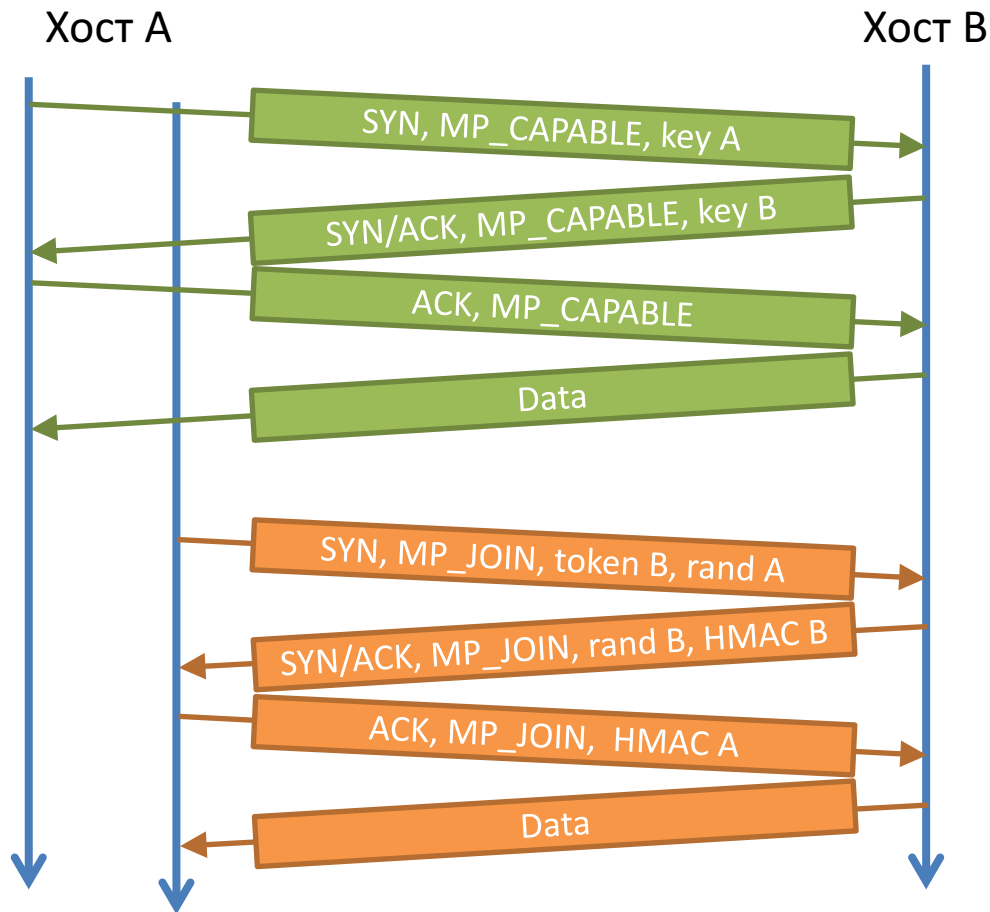
Multi Path TCP (MPTCP)



Организация МРТСП

- Протокол является расширением ТСР и использует стандартный ТСР socket API
 - Не требует изменения логики приложений
- Для синхронизации используются опциональные поля заголовка ТСР
 - 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?
 - Аутентификация 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

МРТСП: 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

МРТСП: управление количеством потоков

- Количество открытых потоков должно соответствовать 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
 - Сколько времени нужно для адаптации соединения?

MPTCP: маршрутизация потоков

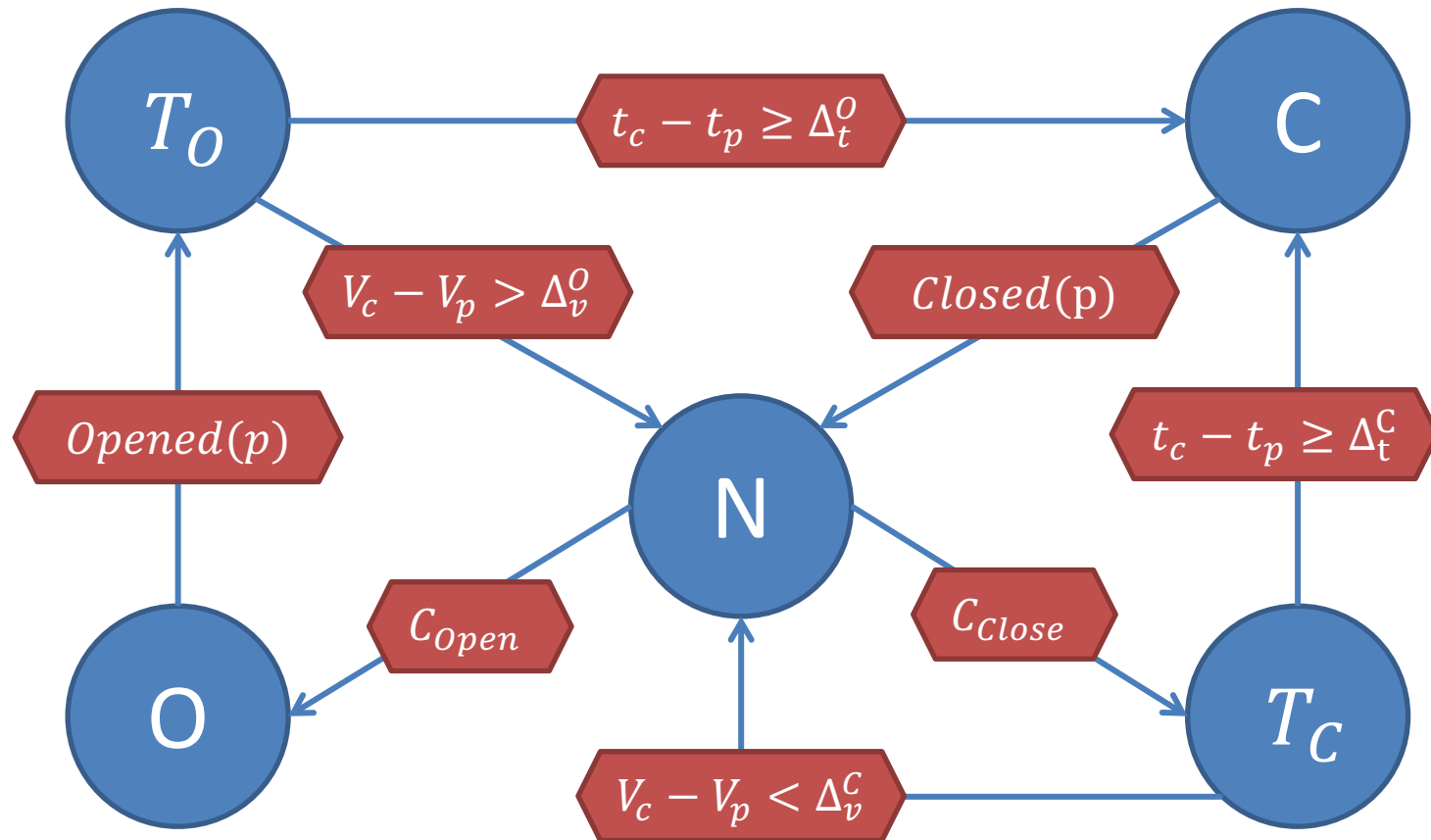
- Базовая эвристика – разные потоки соединения должны проходить по непересекающимся маршрутам
- В традиционных сетях управление маршрутизацией на уровне отдельных потоков затруднительно
 - ECMP распределяет потоки случайно
- В программно-конфигурируемых сетях приводит к увеличению времени отклика и чрезмерной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

Где используется МРТСР?

- 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

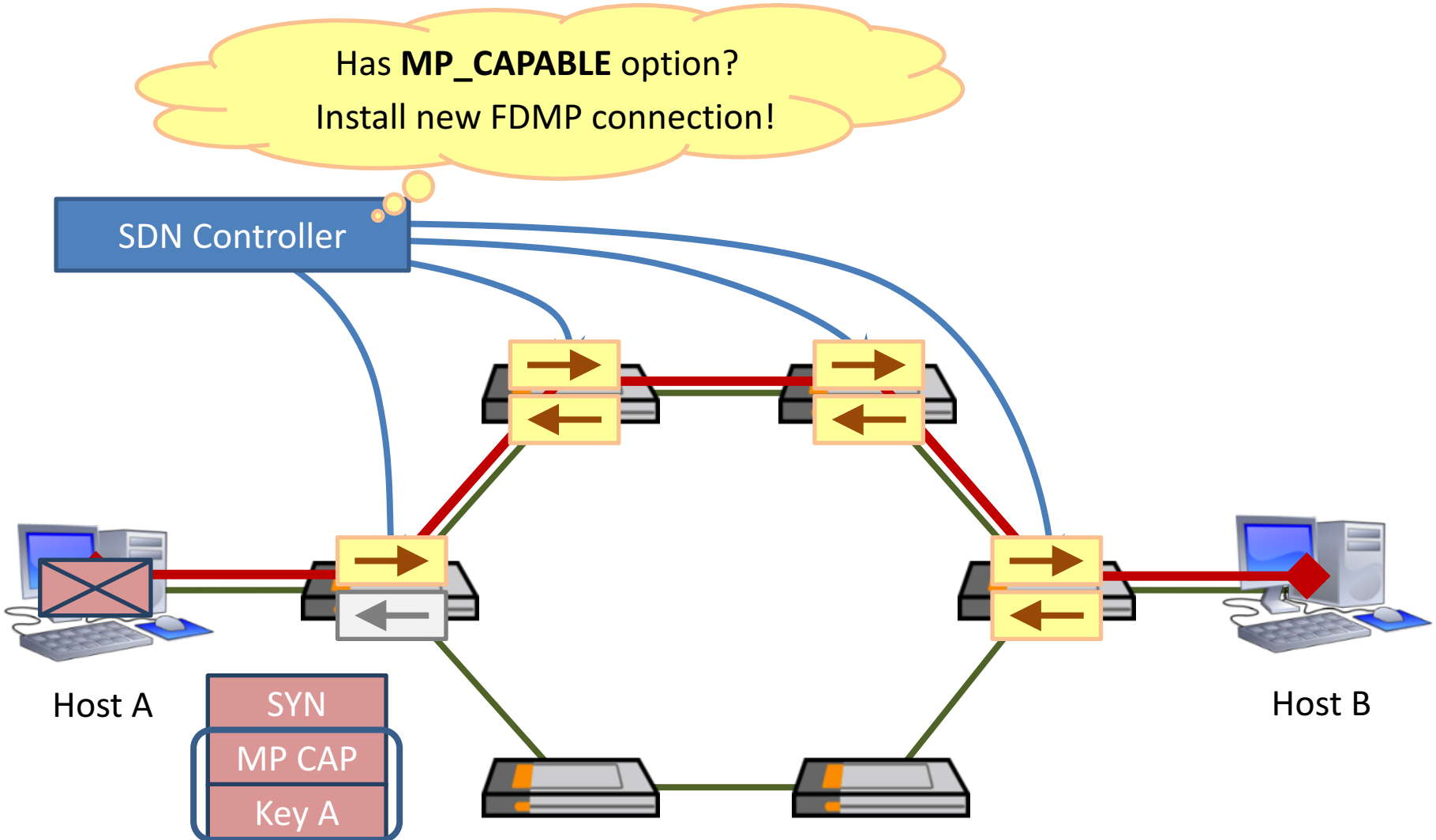
V_p - speed at previous phase

t_c - current time

t_p - time at previous phase

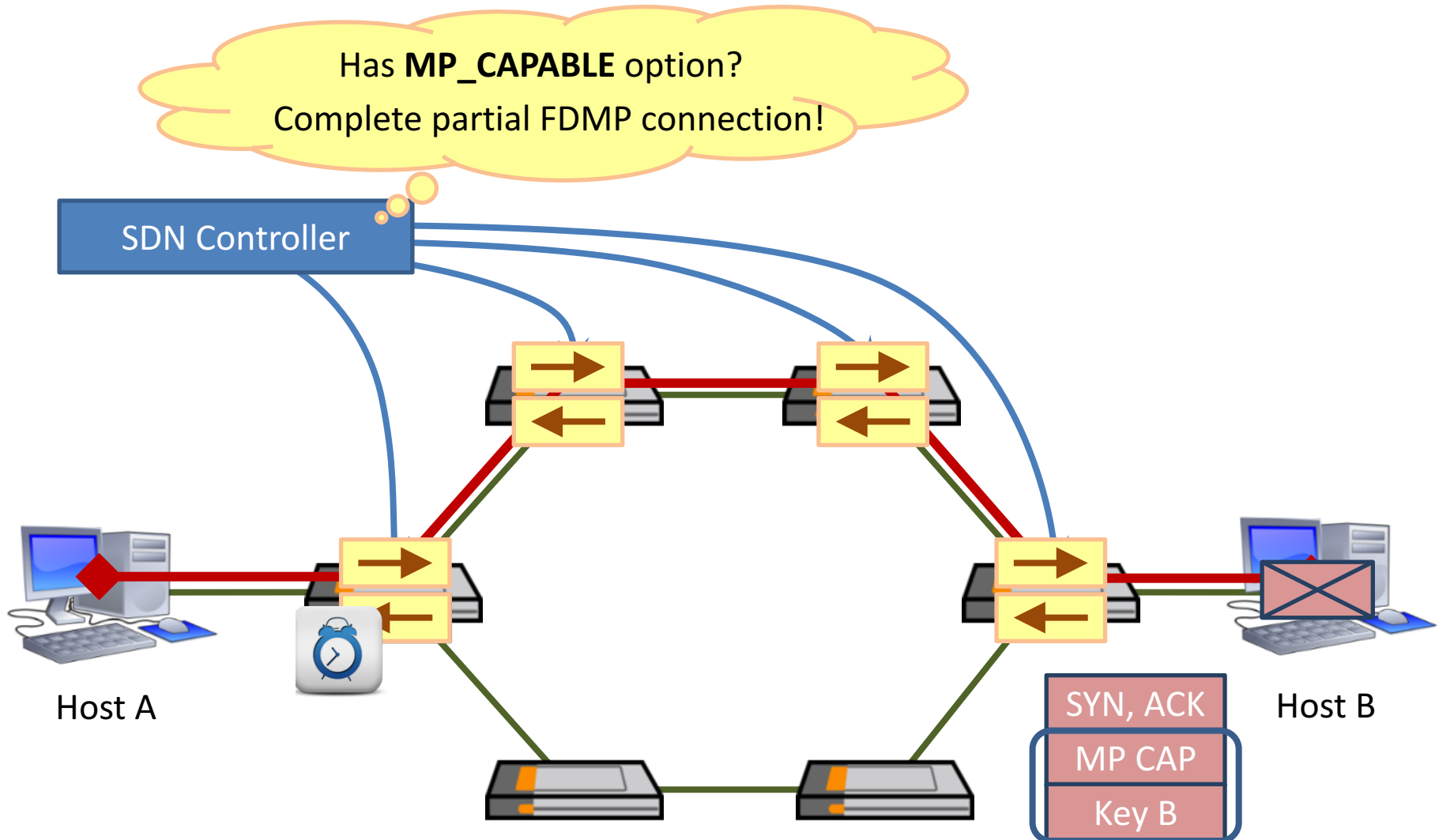
Маршрутизация потков FDMP в ПКС

Установка нового соединения



Маршрутизация потков FDMP в ПКС

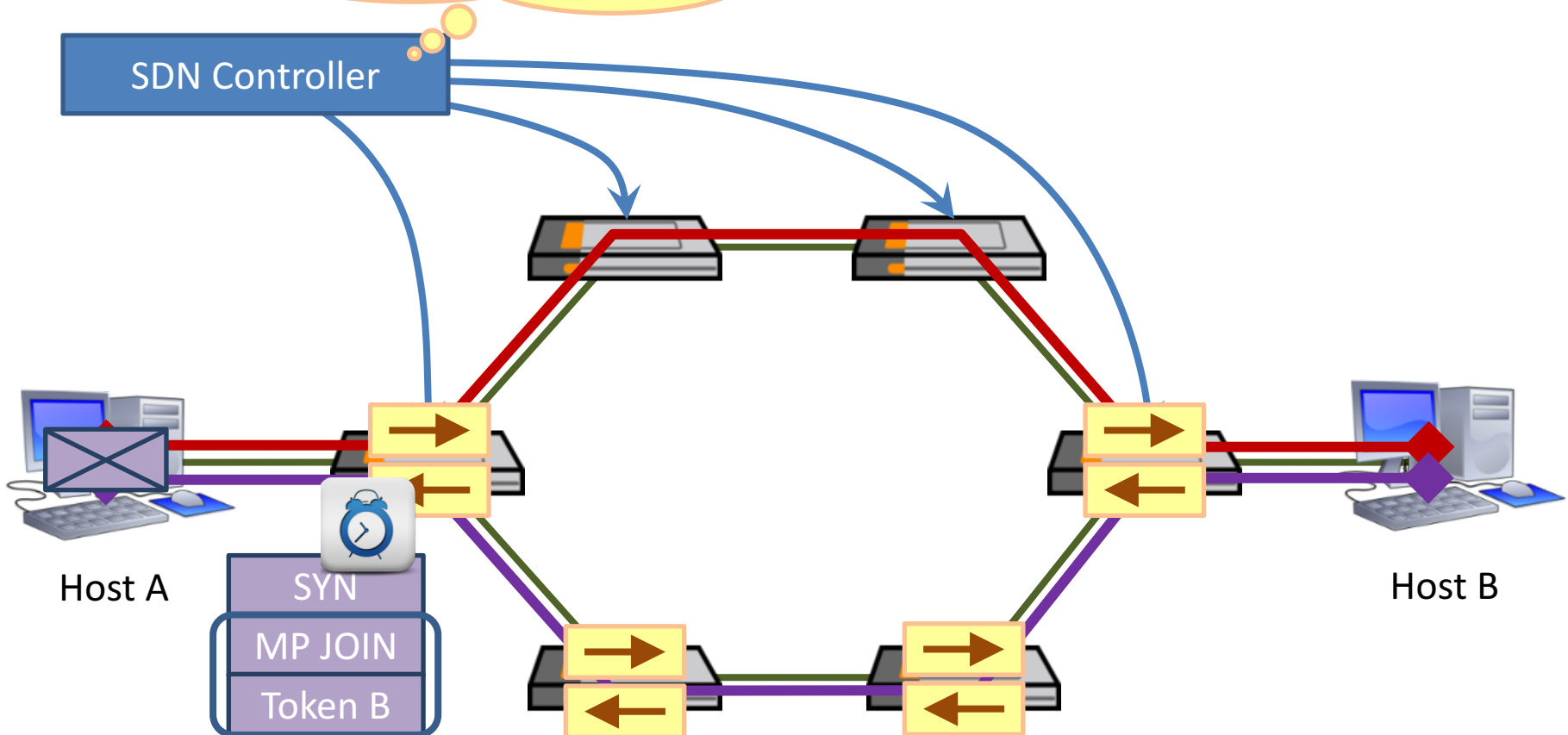
Установка нового соединения



Маршрутизация потков FDMP в ПКС

Установка нового соединения

Has **MP_JOIN** option?
Install new FDMP subflow for a
known connection!

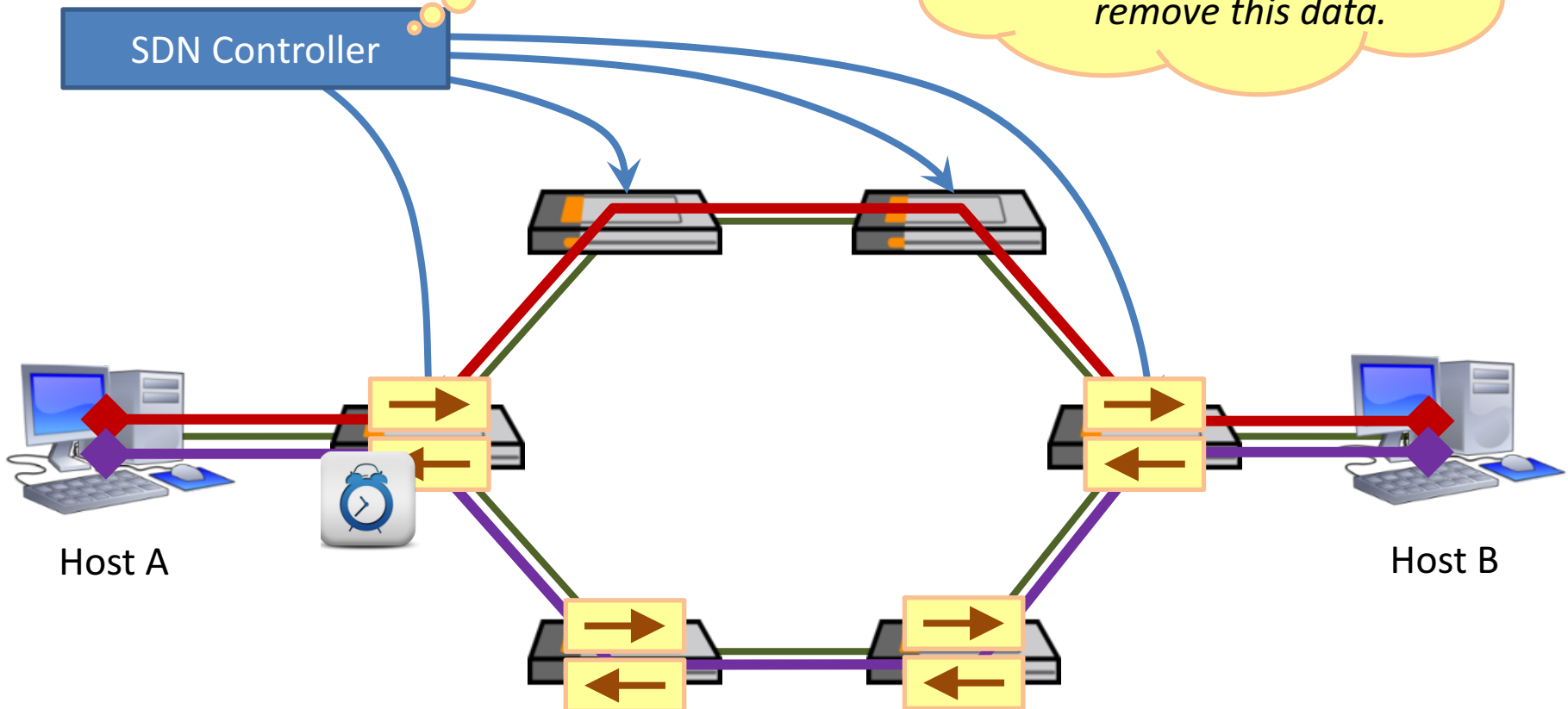


Маршрутизация потоков FDMP в ПКС

Добавление потока к открытому соединению

Subflow is not active any more!
Remove the path!

*Actually, we store metadata and
allow some subflows to resume.
We use flow eviction to
remove this data.*



Маршрутизация потоков FDMP в ПКС

Добавление потока к открытому соединению

Get FDMP packet of a expired subflow!
Either reroute the remembered subflow,
or force hosts to close it

Actually, we store metadata and
allow some subflows to resume.
We use flow eviction to
remove this data.

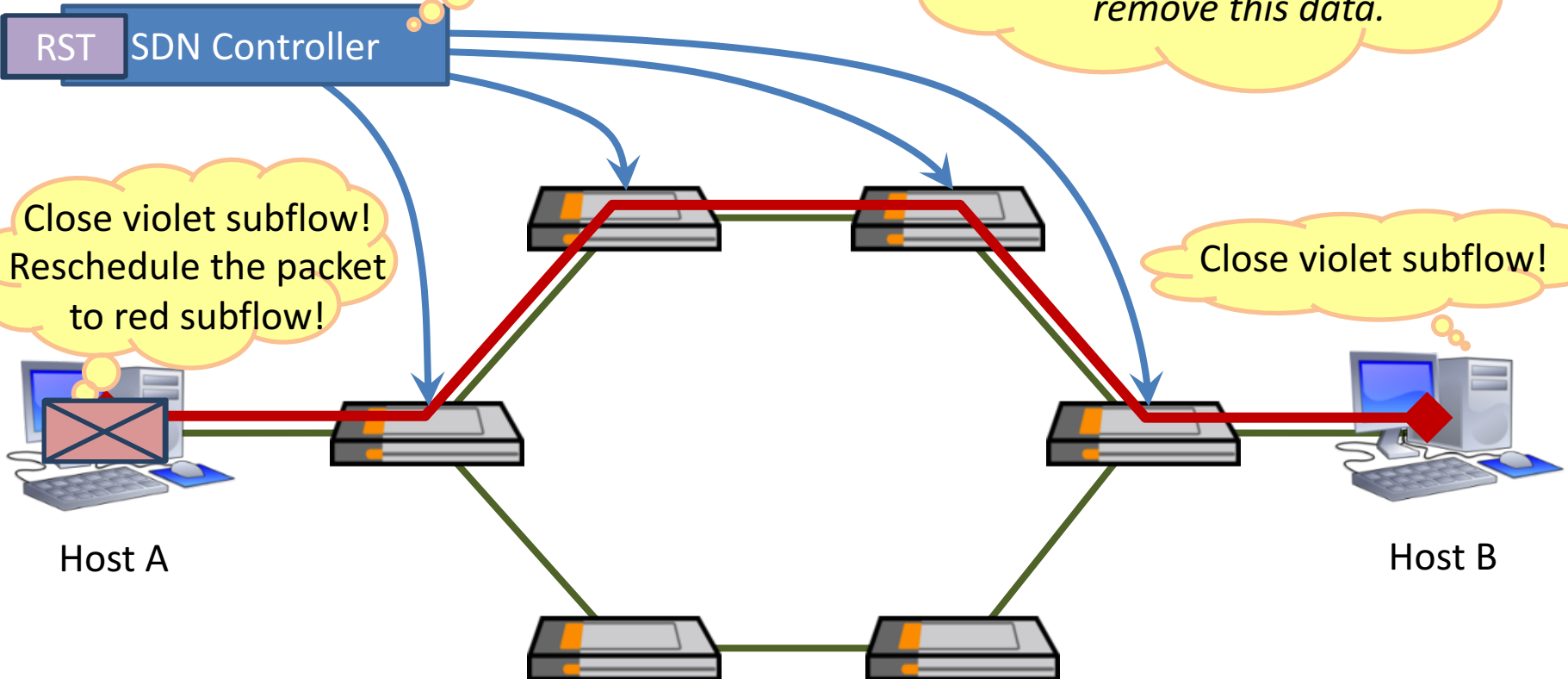
RST SDN Controller

Close violet subflow!
Reschedule the packet
to red subflow!

Close violet subflow!

Host A

Host B



Резюме

- Многопоточная маршрутизация на транспортном уровне – перспективное направление для исследований
 - SCTP открывает много возможностей, но требует изменения логики приложений
 - У стандартных TCP и UDP есть недостатки
 - MPTCP интегрируется с существующими приложениями относительно просто