

Семейство протоколов TCP/IP

Транспортный Уровень

Транспортный уровень

- Межсетевой уровень не предоставляет гарантий успешной доставки всех пакетов.
- Транспортный уровень (Transport layer) предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы.
- Протоколы этого уровня предназначены для взаимодействия типа точка-точка.
- «Точкой» на каждой из сторон-участников передачи является порт.
- На транспортном уровне работают протоколы TCP и UDP.

Порт

- Порт — идентифицируемый номер системный ресурс, выделяемый приложению, выполняемому на некотором сетевом хосте, для связи с приложениями, выполняемыми на других сетевых хостах
- Для каждого из протоколов TCP и UDP стандарт определяет возможность одновременного выделения на хосте до 65536 уникальных портов, идентифицирующихся номерами от 0 до 65535.
- Номер порта в совокупности с номером сети и номером конечного узла однозначно определяет прикладной процесс в сети; такая совокупность называется сокетом.

Порты

- В клиент-серверной модели приложение либо ожидает входящих данных (или запроса на соединение; «слушает порт»; роль сервера), либо посылает данные (или запрос на соединение) на известный порт, открытый приложением-сервером (роль клиента).
- Для приложений-клиентов номер порта выдается драйвером транспортного уровня автоматически в верхнем диапазоне (>1023 , >10000 для некоторых реализаций).
- Порты TCP не пересекаются с портами UDP. То есть, порт 1234 протокола TCP не будет мешать обмену по UDP через порт 1234.

Назначение портов

- При необходимости приложение может запросить конкретный (предопределённый) номер порта.
- Приложения-сервера как правило «слушают» на определенном порту.
- Номера портов наиболее распространенным службам (например: FTP – 21 или telnet – 23) назначаются в Internet Assigned Numbers Authority (IANA). Эти номера затем закрепляются и публикуются в стандартах Internet (RFC 1700).

Сокеты

- Сокеты (socket) — название программного интерфейса для обеспечения обмена данными между процессами.
- Сокет — абстрактный объект, представляющий конечную точку соединения.
- Выделяют клиентские и серверные (слушающие) сокеты.
- Интерфейс сокетов впервые появился в BSD Unix и на настоящий момент поддерживается практически всеми сетевыми операционными системами.

Сокеты и Порты



Узел 1
1.0.0.2

Web-браузер



Узел 2
1.0.0.1

Web-сервер

14: 1.0.0.2:1028:1.0.0.1:80
клиентский сокет

99: 1.0.0.1:80:1.0.0.2:1028
серверный сокет

In

Out

выходной
поток

порт 1028

TCP

IP

входной
поток

In

Out

порт 80

TCP

IP

взаимодействие на
уровне сетевого доступа

Очереди передачи



Узел
1.0.0.2

Web-браузер

FTP-клиент

ICQ

14

сокет

In

Out

порт 1028

21

In

Out

порт 1111

47

In

Out

порт 5467

Транспортный уровень

входная очередь
сегментов/дейтаграмм

выходная очередь
сегментов/дейтаграмм

Межсетевой уровень

Уровень сетевого взаимодействия

Протокол UDP

- Протокол User Datagram Protocol (UDP) обеспечивает неориентированную на соединение службу доставки дейтаграмм по принципу «максимального усилия». Это означает, что получение всей дейтаграммы или правильной последовательности не гарантируется.
- Протокол UDP используется приложениями, не требующими подтверждения (DNS, SNMP, TFTP) или стремящиеся минимизировать «нагрузочный» трафик (передача видео- и аудио- информации, bittorrent, и т.п.).
- UDP-порт – это мультиплексная очередь сообщений.
- Протокол UDP не разбивает полученные от приложения данные на отдельные дейтаграммы, а отправляет их в одной.

Структура пакета UDP



Протокол TCP

- Transmission Control Protocol (TCP) (протокол управления передачей) — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета.
- TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.
- TCP-порт – это мультиплексная очередь сообщений.

Сегменты и потоки

- Сегмент - единица данных TCP.
- TCP рассматривает данные приложений как неструктурированный поток байт. Поступающие данные буферизуются. Для передачи на сетевой уровень из буфера вырезается некоторая часть данных, – это и есть сегмент.
- Все сегменты не обязательно имеют одинаковый размер, но существует ограничение — максимальный размер сегмента (MSS).
- MSS не должен превышать значение (MTU — размер заголовка IP).

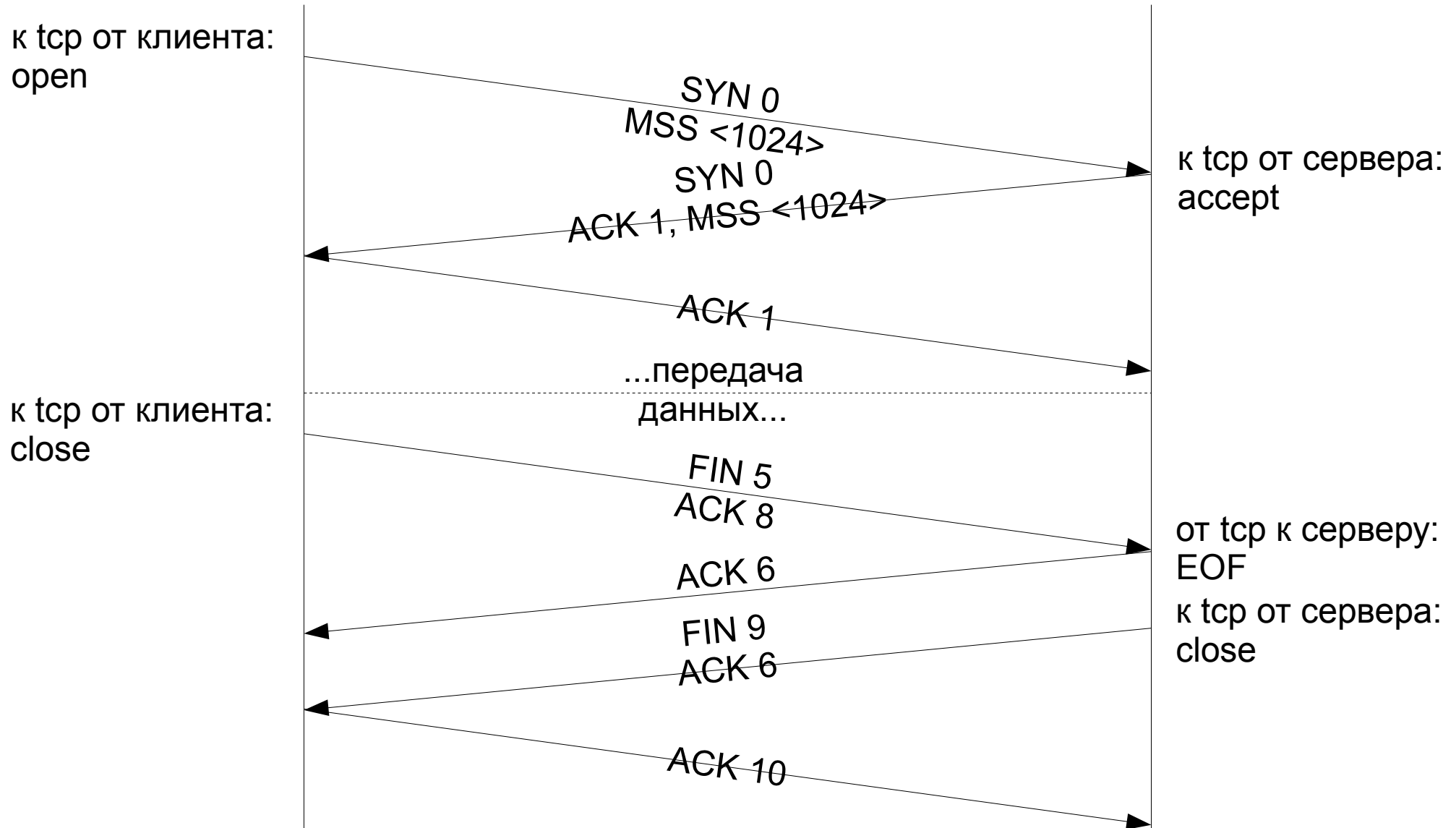
Соединения в TCP

- В TCP определены многошаговые процедуры подтверждения и завершения связи.
- При соединении согласовываются MSS, объемы данных, которые разрешено передавать без подтверждения и начальные/текущие номера передаваемых байтов.
- Процесс установки соединения:
 - Клиент запрашивает соединение посылая сегмент с флагом SYN;
 - Сервер отвечает сегментом SYN/ACK с порядковым номером байта сегмента, который он может послать и подтверждением, включающим порядковый номер сегмента который он ожидает получить;
 - Клиент отправляет обратно сегмент ACK с подтверждением номера последовательности и номером своего подтверждения.

Передача данных и завершение

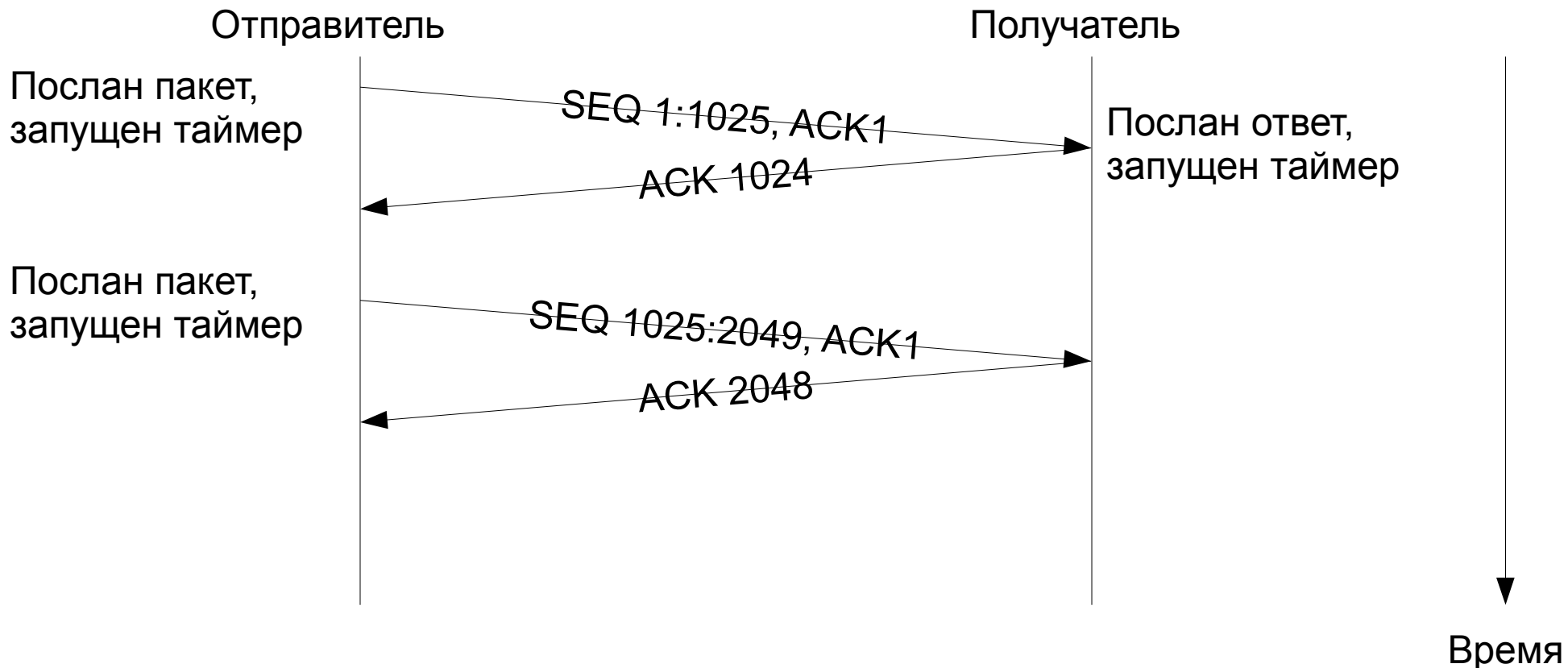
- При обмене данными приемник использует номер последовательности, содержащийся в получаемых сегментах, для восстановления их исходного порядка. Приемник уведомляет передающую сторону о номере последовательности, до которой он успешно получил данные, включая его в поле «номер подтверждения». На любую оговоренную последовательность данных приемник обязан ответить подтверждением (ACK).
- Процесс завершения соединения:
 - Инициатор посылает сегмент с флагом FIN/ACK;
 - Получатель подтверждает запрос на закрытие и отправляет сегмент ACK;
 - Получатель отправляет сегмент FIN/ACK;
 - Инициатор закрывает соединение и в подтверждение отправляет сегмент ACK, что соединение закрыто.

Установка и завершение соединений



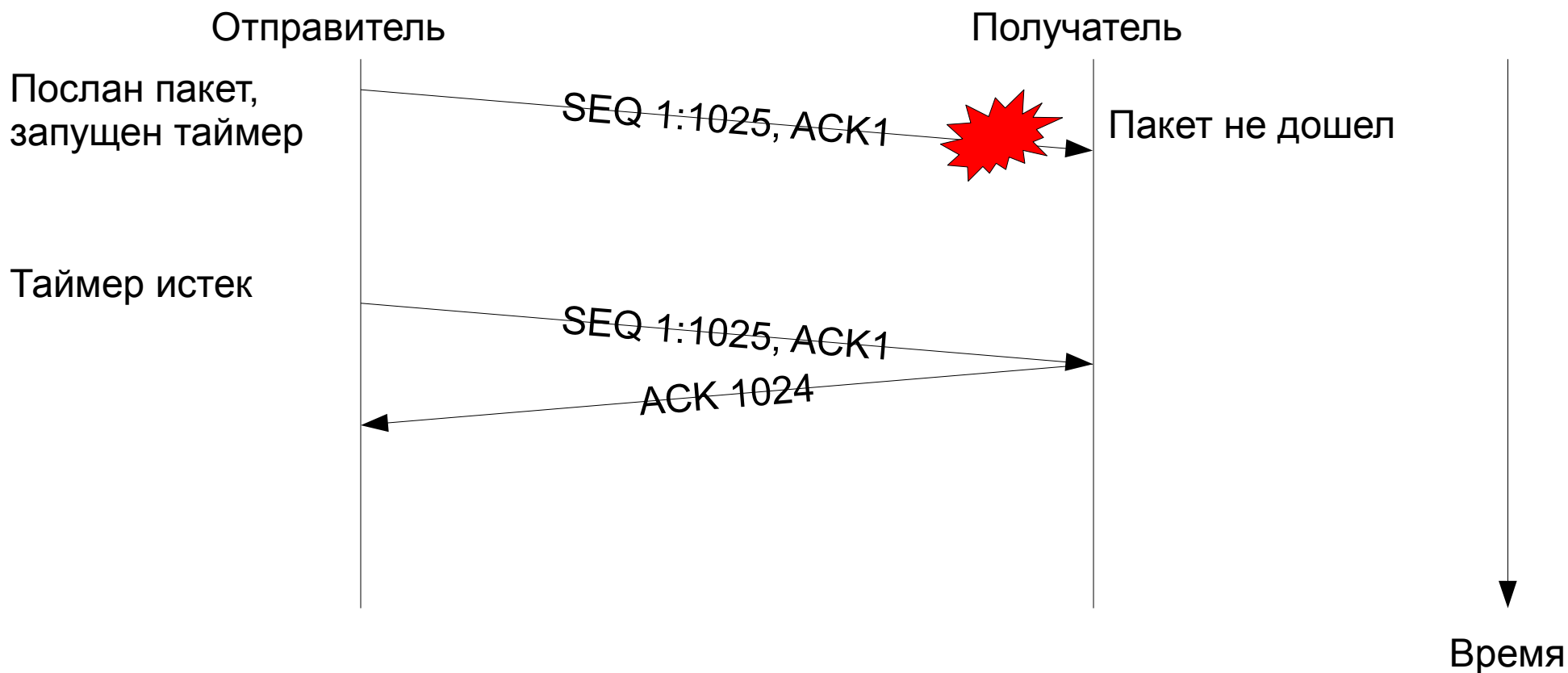
Надежность в ТСР

- Для обеспечения надежности передачи используется механизм подтверждений (ACK) и таймеров повторных передач.



Надежность в ТСР

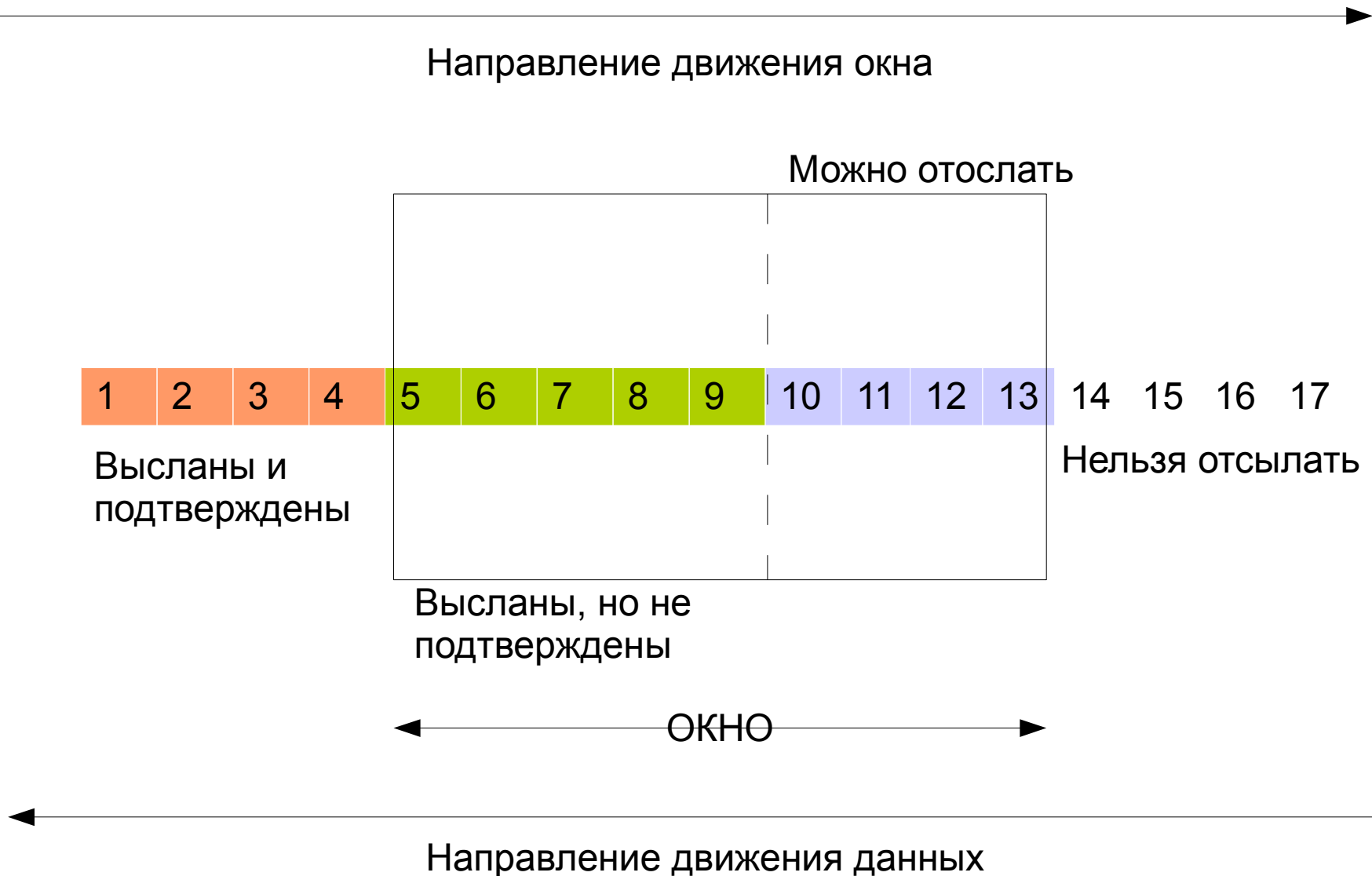
- В случае потери пакета, отправляющий хост инициирует повторную передачу



Определение RTT

- RTT – время оборота, управляющее таймаутами и повторными передачами в TCP.
- Производительность протокола во многом зависит от правильного выбора тайм-аута.
- Алгоритм определения тайм-аута
 - Узнать время оборота.
 - Усреднить времена весовыми коэффициентами.
 - В качестве тайм-аута выбирать среднее время оборота, умноженное на специальный коэффициент (на практике значение этого коэффициента должно превышать 2).

Скользящее окно



Реализация скользящего окна

- Концептуально каждому байту присваивается номер очереди – число на единицу большее, чем максимальный номер байта в полученном сегменте.
- Номер очереди для первого байта данных в сегменте передается вместе с этим сегментом и называется номером очереди для сегмента.
- Сегменты также несут номер подтверждения, который является номером для следующего ожидаемого байта данных, передаваемого в обратном направлении.

Структура пакета TCP

0		1516		32	
Порт источника (16 бит)			Порт назначения		
Номер последовательности (32 бита)					
Номер подтверждения (32 бита)					
Длина заголовка (4 бита)	Резерв (6 бит)	Флаги (6 бит)	Размер окна (16 бит)		
Контрольная сумма (16 бит)			Указатель границы срочных данных		
Опции					
Данные (если есть)					

Флаги в сегментах TCP

- URG — Поле «Указатель границы срочных данных» задействовано.
- ACK — Поле «Номер подтверждения» задействовано.
- PSH — инструктирует получателя протолкнуть данные, накопившиеся в приемном буфере, в приложение пользователя.
- RST — Оборвать соединения, сбросить буфер (очистка буфера).
- SYN — Синхронизация номеров последовательности.
- FIN — флаг, будучи установлен, указывает на завершение соединения.