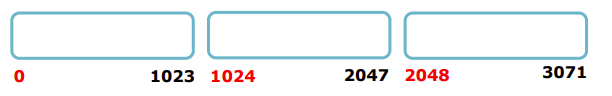
# 12. Транспортный протокол TCP. Формат пакета TCP.

## Общая информация:

* TCP – Transmission Control Protocol
* Стандарт – RFC793
* По сравнению с UDP имеет:
  + Более низкую скорость
  + Большую надежность
* Зарезервирован номер 6 в IP-пакете
* Адресация приложений осуществляется с помощью TCP-портов
* Передача – потоковая
* Данные для передачи хранятся в буфере:
  + Данные от приложения добавляются конец буфера
  + Данные для передачи в сеть берутся из начала буфера
* Пересылаемая порция данных называется – сегмент.
* Каждый передаваемый байт – пронумерован
* Сегменту присваивается номер его первого байта (номер очереди)  
  
* При посылке в сеть сегмента:
  + Сегмент копируется в буфер повторной передачи
  + Взводится таймаут

## Передача сегмента в сеть:

Изображение выглядит как текст, небо

Автоматически созданное описание

* Каждый посланный в сеть байт должен быть подтвержден
* При получении подтверждения от сегмента подтвержденными считаются все байты сегмента
* Если подтверждение не получено в течение определенного времени – сегмент из буфера повторной передачи посылается заново
* Подтверждение содержит номер следующего ожидаемого байта
* В TCP отрицательные квитанции не посылаются

## Формат пакета:

Изображение выглядит как стол

Автоматически созданное описание

* Номер очереди
  + Номер посланного сегмента при обмене
  + Синхронизация номеров сегментов при установлении соединения
* Номер подтверждения
  + Подтверждение принятого сегмента
  + Синхронизация номеров сегментов при установлении соединения
* Смещение данных – длина заголовка TCP
* Окно – величина скользящего окна
* Контрольная сумма – сегмента
* Указатель срочности – объем срочных данных

**Порт источника**

Порт источника идентифицирует приложение клиента, с которого отправлены пакеты. По возвращении данные передаются клиенту на основании номера порта источника.

**Порт назначения**

Порт назначения идентифицирует порт, на который отправлен пакет.

**TCP-порты**

Существует набор служб (использующих для передачи данных TCP), за которыми закреплены определенные порты.

Номер последовательности

Номер последовательности выполняет две задачи:

1. Если установлен флаг SYN, то это начальное значение номера последовательности — ISN (Initial Sequence Number), и первый байт данных, которые будут переданы в следующем пакете, будет иметь номер последовательности, равный ISN + 1.
2. В противном случае, если SYN не установлен, первый байт данных, передаваемый в данном пакете, имеет этот номер последовательности.

**Номер подтверждения**

Если установлен флаг ACK, то это поле содержит номер последовательности, ожидаемый получателем в следующий раз. Помечает этот сегмент как подтверждение получения.

**Смещение данных**

Это поле определяет размер заголовка пакета TCP в 4-байтных (4-октетных) словах. Минимальный размер составляет 5 слов, а максимальный — 15, что составляет 20 и 60 байт соответственно. Смещение считается от начала заголовка TCP.

## Флаги:

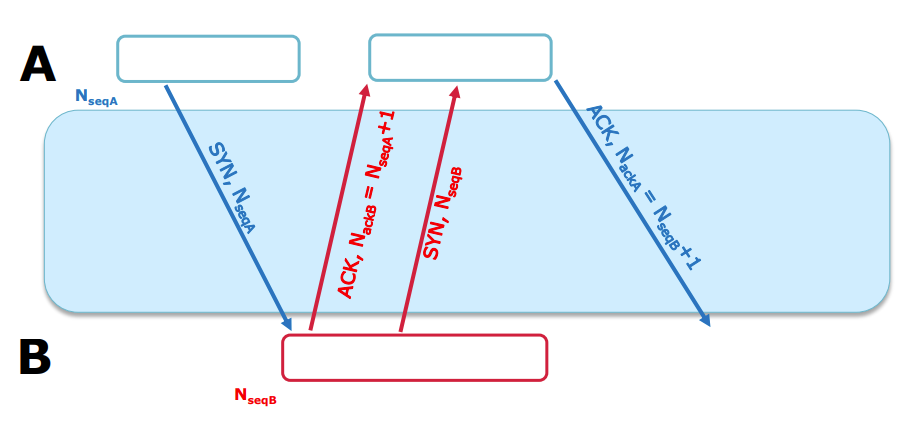
Изображение выглядит как стол

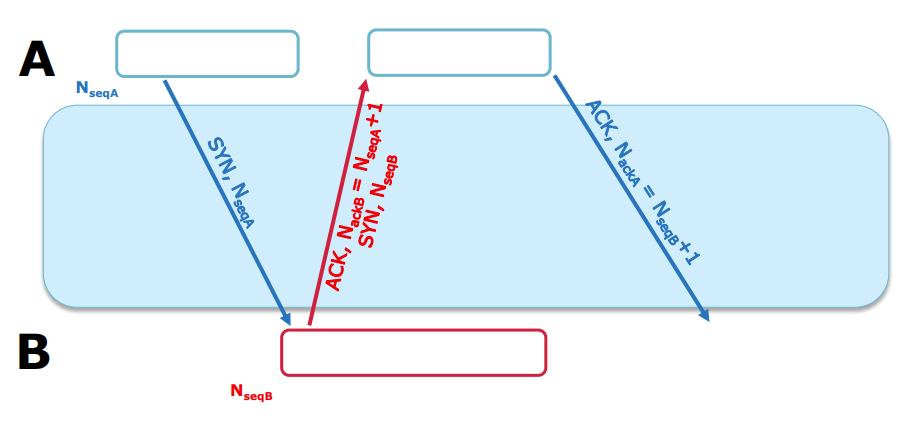
Автоматически созданное описание

* URG – задействовано поле «Указатель срочности»
* ACK - задействовано поле «Подтверждение»
* PSH – включена функция проталкивания
* RST – перезагрузка соединения
* SYN – синхронизация номеров очередей
* FIN – завершение соединения

# 13. Транспортный протокол TCP. Алгоритм функционирования.

## Установление соединения:





Процесс начала сеанса TCP - обозначаемое как "рукопожатие" (handshake), состоит из 3 шагов.

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN.

* Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента.
* В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED.
* В случае неудачи сервер посылает клиенту сегмент с флагом RST.

1. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

* Если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED.
* Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться.
* Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

1. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

* В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется "трехэтапным согласованием" ("three way handshake"), так как несмотря на то что возможен процесс установления соединения с использованием 4 сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), на практике для экономии времени используется 3 сегмента.

Изображение выглядит как текст, небо, устройство

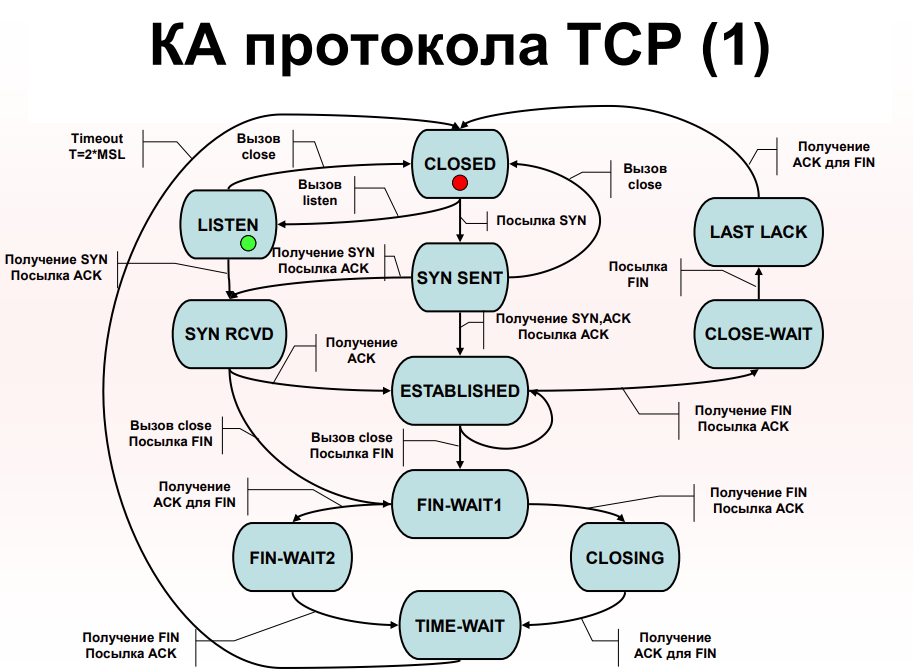
Автоматически созданное описание

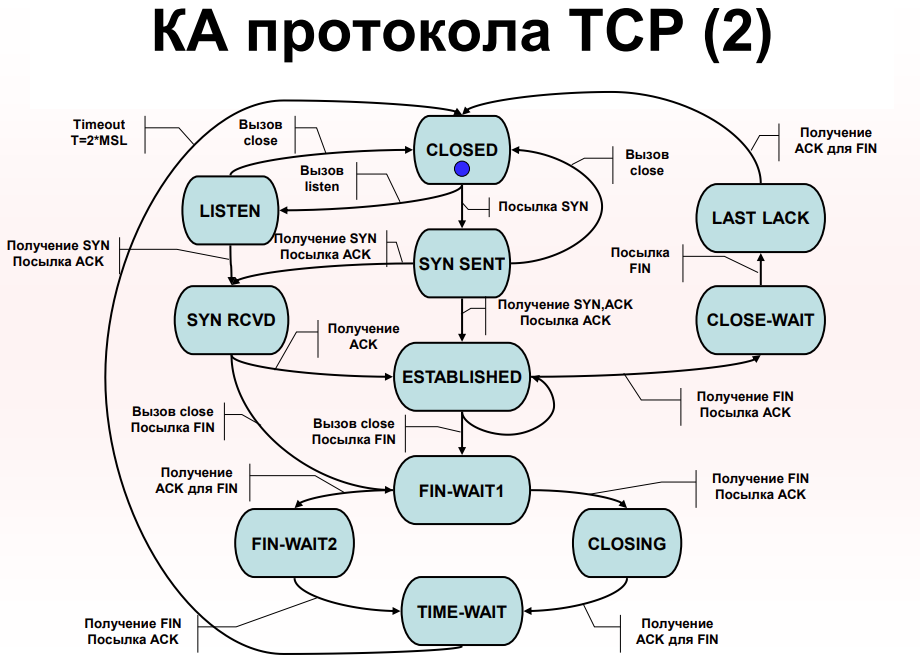
Завершение соединения можно рассмотреть в три этапа:

1. Посылка серверу от клиента флагов FIN и ACK на завершение соединения.

2. Сервер посылает клиенту флаги ответа ACK , FIN, что соединение закрыто.

3. После получения этих флагов клиент закрывает соединение и в подтверждение отправляет серверу ACK , что соединение закрыто.





## Срочные данные:

* Срочные данные
  + Используется для немедленной доставки данных приложению на приемной стороне
  + Флаг URG – признак наличия срочных данных
  + Поле Offset – указатель на первые несрочные данные
* Проталкивание данных
  + Используется для немедленной отсылки сегмента в сеть
  + Для проталкивания устанавливается флаг PSH

## Управление скоростью передачи:

* Идея: передавать в сеть больше неподтвержденных данных
* В сеть могут передаваться сегменты, которые попали в скользящее окно
* Окно сдвигается только тогда, когда приходит подтверждение на первый посланный сегмент

