Образовательный комплекс

Компьютерные сети

Лекция 14

Уровень Хост-Хост TCP/IP

Microsoft®

Содержание

- Уровень Хост-Хост модели TCP/IP
  - Протокол UDP
  - Протокол TCP
- Программный интерфейс сокетов

Уровень Хост-Хост

- Межсетевой уровень (IP) позволяет передавать данные между узлами через интерсеть
- Уровень Хост-Хост обеспечивает сервисы, которые могут использоваться приложениями для доставки данных
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- Протоколы TCP и UDP выполняют подмножество функций более сложного транспортного уровня модели ISO/OSI

Уровень Хост-Хост

- Процесс, который хочет взаимодействовать с другим процессом, должен зарегистрироваться на каком-либо порту
- Порт – это 32-битное число, которое используется протоколами уровня Хост-Хост для определения протокола (сервиса или приложения) прикладного уровня, которому предназначается сообщение
- Механизм портов используют протоколы TCP, UDP. Также механизм портов реализован в протоколе ISO-4.
- Пространства портов протоколов TCP и UDP различны
  - то есть, один процесс может использовать 10-й порт TCP, а другой в то же самое время – 10-й порт UDP

Уровень Хост-Хост Стандартные сервисы

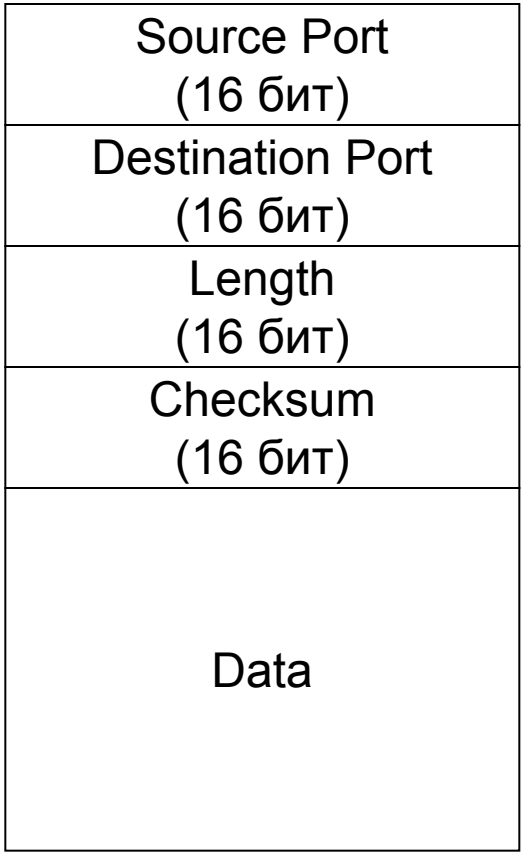
- Порты с номерами 0-1023 предназначены для регистрации серверных компонент стандартных сервисов (протоколов прикладного уровня) TCP/IP
- Порты с номерами 1024-65535 используются любыми программами, в том числе клиентскими частями стандартных протоколов
- Ниже перечислены некоторые стандартные сервисы, используемые ими протоколы и номера портов
  - 21/TCP – FTP (20/TCP – FTP-DATA)
  - 22/TCP – SSH (Secure SHell)
  - 23/TCP – TELNET
  - 25/TCP – SMTP
  - 53/UDP – NAMESERVER (DNS)
  - 80/TCP – HTTP
  - 110/TCP – POP3
  - …
- В Linux в файле /etc/services для большого числа сервисов указаны используемый протокол и номер порта

Уровень Хост-Хост Протокол UDP

- UDP – ненадежный датаграммный протокол
  - Обеспечивает прикладным программам возможность посылать данные другим программам с минимальными накладными расходами
  - Не добавляет надежности нижележащим уровням
  - Не выполняет контроль трафика
  - Приложения, требующие надежной доставки потоков данных, должны использовать TCP

Уровень Хост-Хост
Формат UDP-датаграммы

- Source Port (SP) – номер порта источника
- Destination Port (DP) – номер порта получателя
- Length – длина датаграммы в байтах
- Checksum – контрольная сумма датаграммы
- Data – передаваемые данные



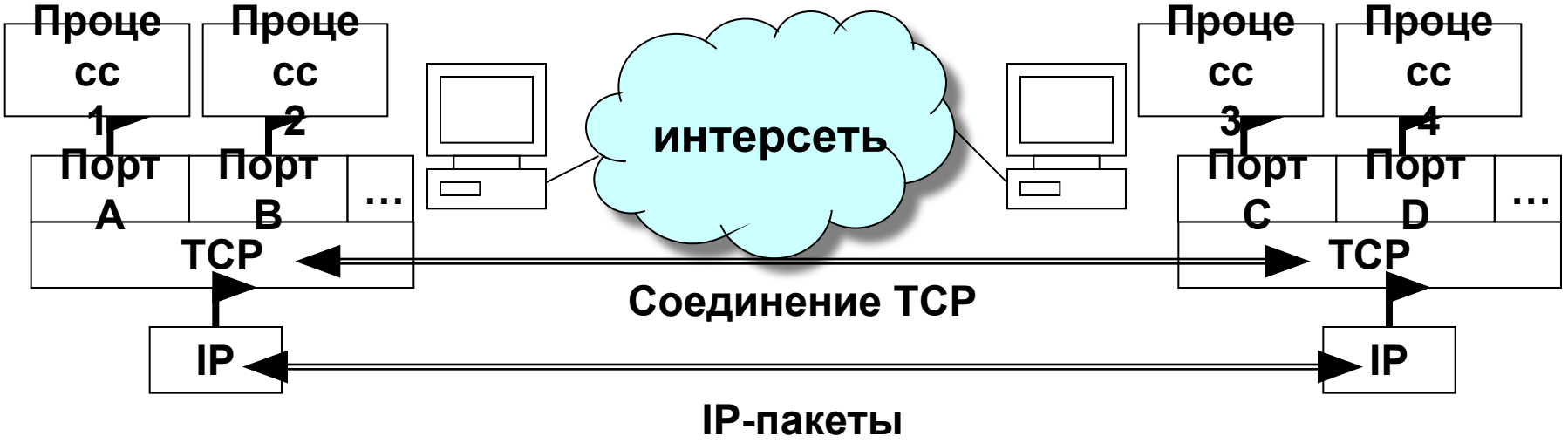
Уровень Хост-Хост
Формат UDP-датаграммы

- Для внутреннего использования перед UDP-заголовком размещается псевдозаголовок, который не входит в UDP-датаграмму и содержит информацию из IP-заголовка
  - IP-адрес отправителя
  - IP-адрес получателя
  - Протокол
  - Длина UDP-датаграммы
- Длина псевдозаголовка UDP не учитывается в общей длине датаграммы, но его содержимое используется при вычислении контрольной суммы

Уровень Хост-Хост Использование UDP

- Некоторые стандартные приложения, использующие UDP
  - Trivial File Transfer Protocol (TFTP) – тривиальный протокол передачи файлов (используется при удаленной загрузке)
  - Domain Name Server (DNS) – служба доменных имён
  - Remote Procedure Call (RPC) – механизм удаленного вызова процедур (используется многими программами, например, сервисом Network File System, NFS)
  - Simple Network Message Protocol (SNMP) – простой протокол управления сетью

Уровень Хост-Хост Протокол TCP



- TCP – протокол, обеспечивающий сервис, ориентированный на соединение, для пары взаимодействующих процессов, и включающий надежность, контроль трафика и исправление ошибок

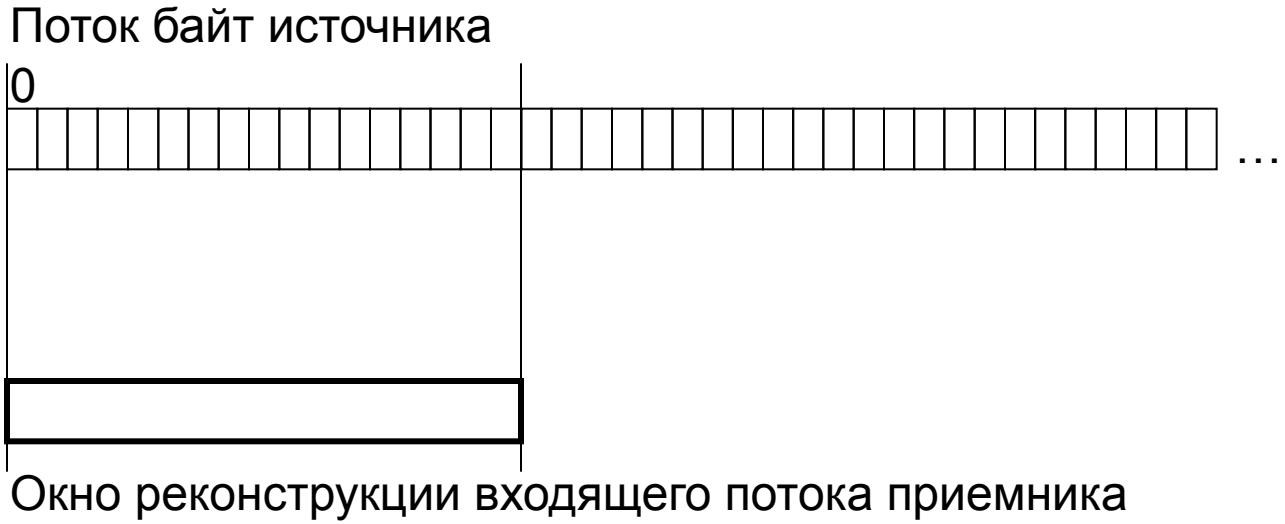
Уровень Хост-Хост Протокол TCP

- TCP устанавливает логическое соединение между парой процессов. Логическое соединение идентифицируется парой адресов сокетов.
  - Сокет – объект межпроцессного взаимодействия, который может быть использован для передачи данных между процессами через сеть
  - Адрес сокета – комбинация IP-адреса и номера порта, используемых для коммуникации
- TCP хранит всю информацию о логическом соединении в специальной структуре, называемой блоком управления передачей

Уровень Хост-Хост Протокол TCP

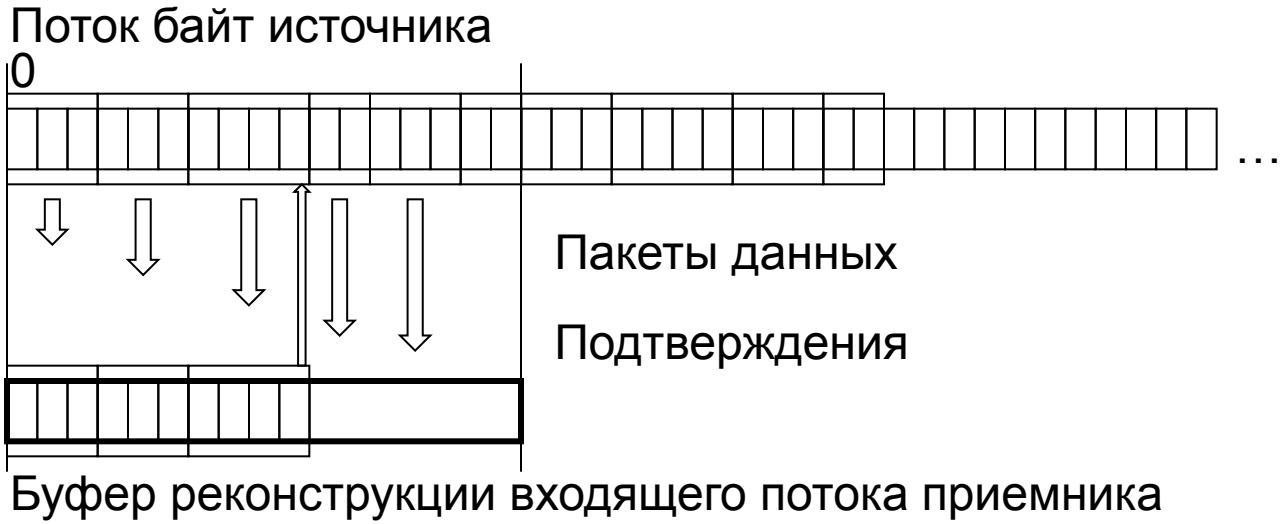
- Поля блока управления передачей
  - Локальный IP-адрес
  - Локальный номер порта
  - Протокол
  - Удаленный IP-адрес
  - Удаленный номер порта
  - Размер буфера передачи
  - Размер буфера приема
  - Текущее состояние TCP
  - Текущее значение интервала тайм-аута
  - Количество осуществленных повторных передач
  - Текущий размер окна передачи
  - Максимальный размер передаваемого сегмента
  - Номер последнего из подтвержденных байтов
  - Максимальный размер принимаемого сегмента
  - Номер байта, который должен быть послан

Уровень Хост-Хост
Протокол TCP – Механизм окон…



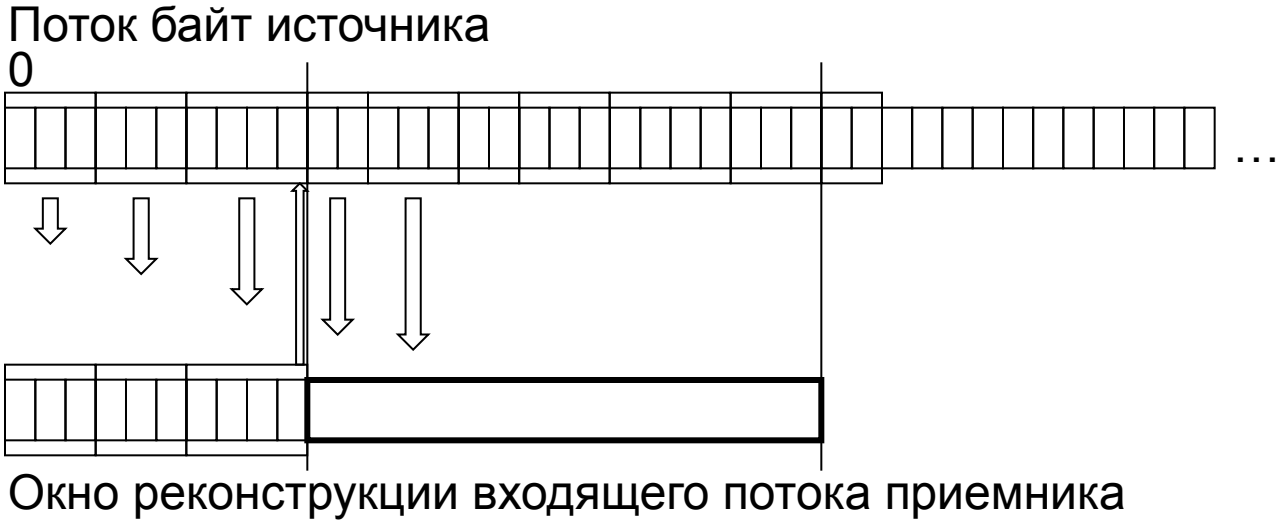
- Протокол TCP обеспечивает передачу потоков данных, при этом он использует механизм окон
  - Все байты исходящего потока последовательно нумеруются
  - Размер окна задается получателем в момент установления соединения, но может изменяться им в процессе передачи
  - На стороне приемника окно – это фактически буфер приема, на стороне источника – абстракция, определяющая порядок передачи
  - В исходный момент окно расположено в начале потока

Уровень Хост-Хост
Протокол TCP – Механизм окон…



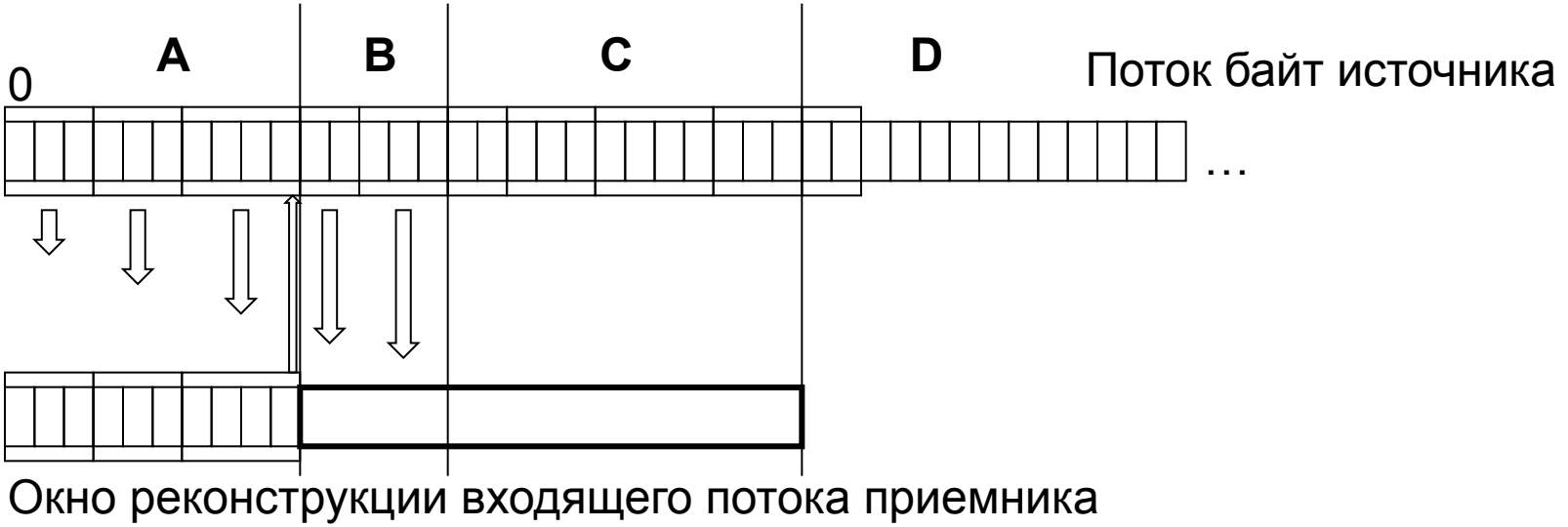
- Источник разбивает исходящий поток на пакеты (сегменты)
- Источник может послать все пакеты в окне без подтверждения, но должен запускать таймер для каждого из них
- Получатель подтверждает номер последнего принятого байта

Уровень Хост-Хост
Протокол TCP – Механизм окон…



- Получатель в ходе реконструкции потока данных передвигает начало окна на первый еще не полученный байт
- Источник продвигает окно после получения очередного подтверждения

Уровень Хост-Хост
Протокол TCP – Механизм окон…



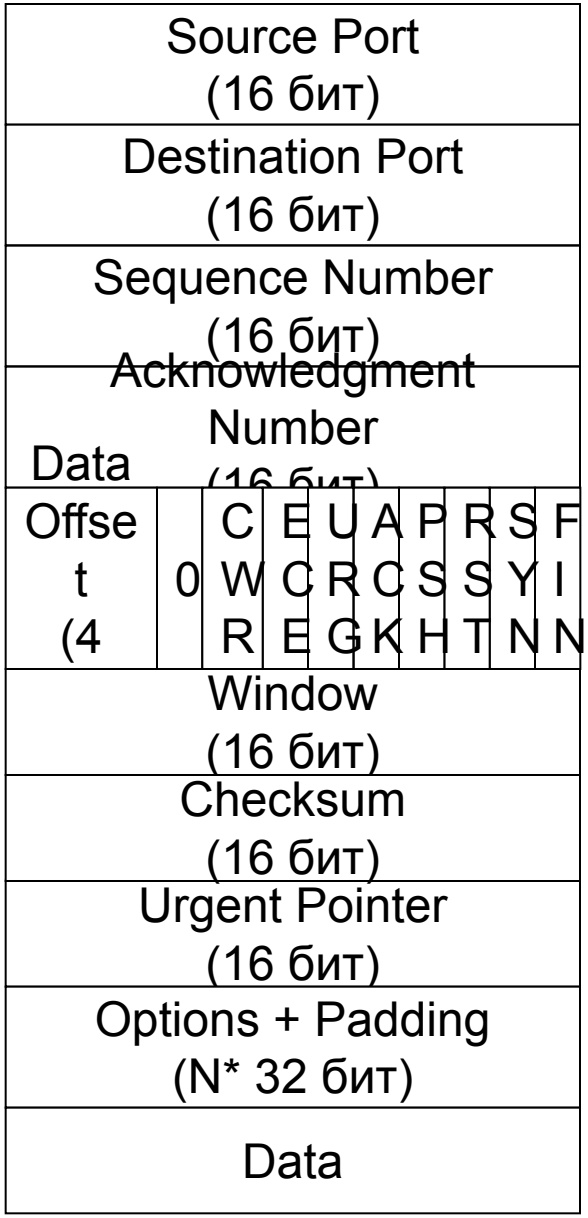
- С точки зрения источника, выходной поток байт делится на 4 части
  - A – посланные и подтвержденные байты
  - B – посланные, но еще не подтвержденные байты
  - C – байты, которые могут быть посланы без подтверждения
  - D – байты, которые еще не могу быть посланы

Уровень Хост-Хост
Протокол TCP – Механизм окон

- Если при передаче потерян какой-либо пакет, источник не получит подтверждения и спустя время тайм-аута начнет повторную передачу всех неполученных байт
  - даже если следующие пакеты были получены приемником, он посылает номер последнего байта полностью реконструированного потока
- Каждое подтверждение полностью описывает текущую ситуацию для источника
  - если получатель отправил подтверждение, но оно не дошло до источника, следующее доставленное подтверждение отменит необходимость повторной передачи данных источником
- Таким образом, механизм окон обеспечивает
  - Надежную передачу
  - Контроль трафика
  - Эффективное использование пропускной способности сети (источник может посылать данные, не дожидаясь подтверждения всех уже отправленных данных)

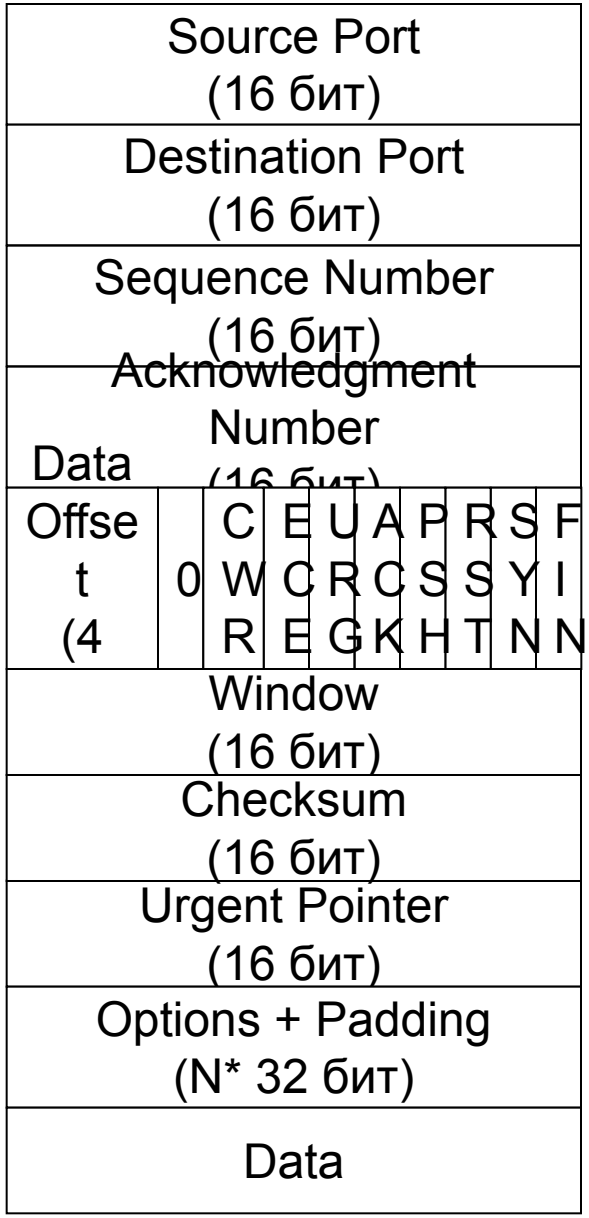
Уровень Хост-Хост Формат TCP-сегмента…

- Source Port (SP) – номер порта источника
- Destination Port (DP) – номер порта получателя
- Sequence Number – порядковый номер первого байта в сегменте
- Acknowledgment Number – если установлен флаг ACK, номер следующего байта в потоке, который узел рассчитывает принять как получатель
- Data Offset – размер заголовка TCP в 32-битных словах



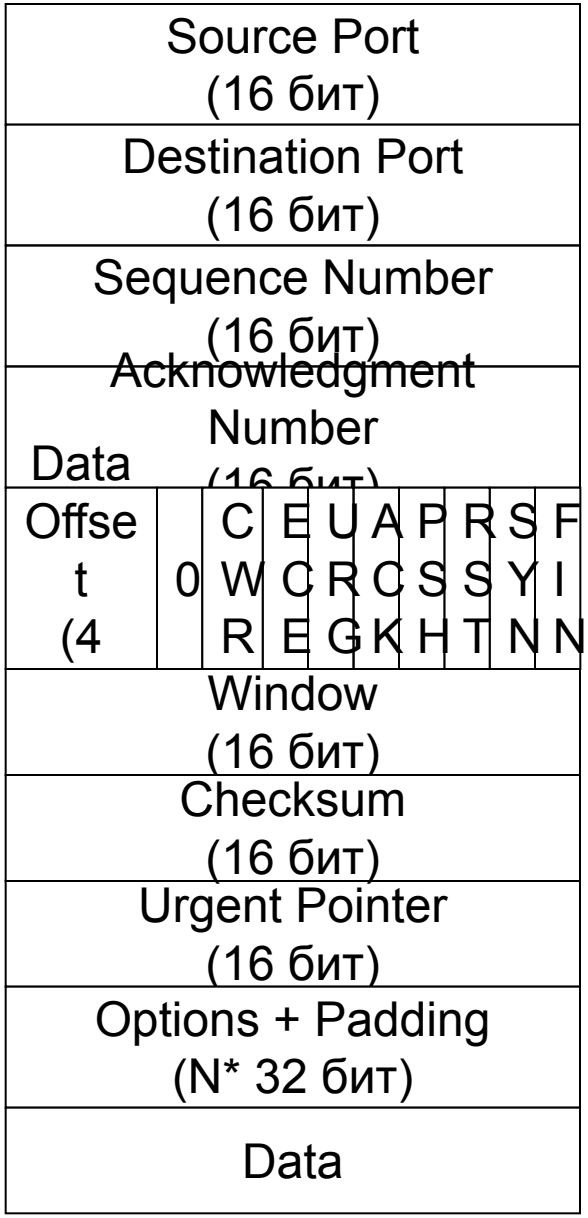
Уровень Хост-Хост Формат TCP-сегмента…

- Флаги
  - CWR (Congestion Window Reduced) – подтверждение принятия сегмента с установленным флагом ECE
  - ECE (ECN-Echo) – данный узел способен на явное уведомление о перегрузке
  - URG – если равен 1, то поле Urgent Pointer сегмента значимо
  - ACK – если равен 1, то поле Acknowledgment Number сегмента значимо
  - PSH – указание получателю "протолкнуть" данные в приложение
  - RST – сброс соединения
  - SYN – синхронизация номеров байт в потоке
  - FIN – конец данных источника



Уровень Хост-Хост Формат TCP-сегмента…

- Window – количество байт, начиная с номера из поля Acknowledgment Number, которое узел может принять как получатель
- Checksum – контрольная сумма TCP-сегмента
- Urgent Pointer – номер первого байта срочных данных (в поле данных сегмента)
- Options – опции доставки TCP-сегмента, может содержать несколько опций и имеет переменную длину
- Padding – дополнения поля Options до размера, кратного 32 битам
- Data – данные



Уровень Хост-Хост
Формат TCP-сегмента

- Для внутреннего использования перед TCP-заголовком размещается псевдозаголовок, который не входит в TCP-сегмент и содержит информацию из IP-заголовка
  - IP-адрес отправителя
  - IP-адрес получателя
  - Протокол
  - Длина TCP-сегмента
- Длина псевдозаголовка TCP не учитывается в общей длине сегмента, но его содержимое используется при вычислении контрольной суммы

Программный интерфейс сокетов

Программный интерфейс сокетов

- Сокет – объект межпроцессного взаимодействия, обеспечивающий прием и передачу данных для процесса
- Существуют следующие типы сокетов
  - Stream – обеспечивает надежную доставку потоков данных (SOCK_STREAM)
  - Datagram – обеспечивает ненадежную доставку сообщений (SOCK_DGRAM)
  - Sequential packet – обеспечивает надежную доставку пакетов длины не больше заданной (SOCK_SEQPACKET)
    - на настоящий момент отсутствует реализация данного типа сокетов
  - Raw – доступ к нижележащему протоколу (SOCK_RAW)

Программный интерфейс сокетов

- Сокет – достаточно общий интерфейс и может обеспечивать взаимодействие посредством использования различных механизмов (локальных и сетевых)
- Различные механизмы требуют использования специальных типы адресов, принадлежащие различным коммуникационным доменам
  - AF_INET – взаимодействие удаленных систем с использованием TCP/IP (адрес – пара IP-адрес + номер порта)
  - AF_INET6 – взаимодействие удаленных систем с использованием TCP/IP (IP версии 6)
  - AF_UNIX – локальное межпроцессное взаимодействие (адресом является имя на файловой системе)
  - AF_NS
  - IUCV
  - ...

Программный интерфейс сокетов Датаграммное взаимодействие

- При датаграммной передаче взаимодействующие процессы выполняют один и тот же набор вызовов (далее приводится множество вызовов для сокетов в реализации Berkley)
- Создание сокета
  - int socket(int domain, int type, int protocol);
  - domain – коммуникационный домен
  - type – тип сокета
  - protocol – протокол транспортного уровня, если значение параметра равно 0, используется протокол по умолчанию для данного типа сокета и коммуникационного домена
  - Возвращаемое значение – дескриптор сокета

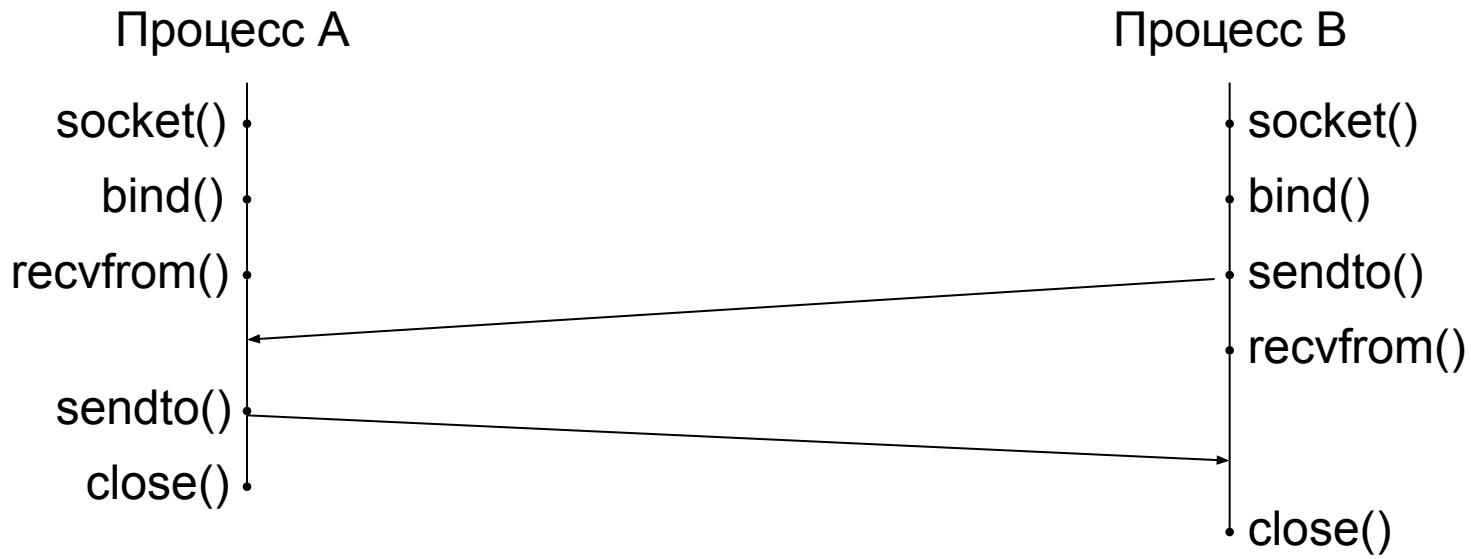
Программный интерфейс сокетов Датаграммное взаимодействие

- Назначение адреса сокету
  - int bind(int s, const struct sockaddr *name, int namelen);
  - s – дескриптор сокета
  - name – адрес буфера, содержащего адрес сокета. Адрес представляет собой структуру struct sockaddr_in; ее поля
    - sa_family_t sin_family; - коммуникационный домен
    - in_port_t sin_port; - номер порта (используется сетевой порядок следования байт в целочисленных значениях)
    - struct in_addr sin_addr; - структура, содержащая сетевой адрес
      - in_addr_t s_addr; - IP-адрес (используется сетевой порядок следования байт в целочисленных значениях)
    - unsigned char sin_zero[8]; - нули
  - namelen – длина структуры, содержащей адрес

Программный интерфейс сокетов Датаграммное взаимодействие

- Передача/прием
  - ssize_t recvfrom(int s, void *buffer, size_t length, int flags, struct sockaddr *address, socklen_t *address_len);
  - ssize_t sendto(int s, const void *message, size_t length, int flags, const struct sockaddr *address, socklen_t address_len);
  - s – дескриптор сокета
  - buffer – адрес буфера приема/передачи
  - length – размер буфера в байтах
  - flags – флаги операции
  - address – адрес буфера адреса источника/получателя
  - address_len – размер адреса получателя в sendto(), адрес объекта целого типа, содержащего размер буфера address в recvfrom()

Программный интерфейс сокетов Датаграммное взаимодействие



- Закрытие сокета
  - int close(int s);
  - s – дескриптор сокета

Программный интерфейс сокетов
Взаимодействие с установлением соединения

- При использовании взаимодействия, ориентированного на соединение, клиент и сервер выполняют разные последовательности вызовов
- Сервер
  - Создание сокета
    - int socket(int domain, int type, int protocol);
  - Назначение адреса сокету
    - int bind(int s, const struct sockaddr *name, int namelen);

Программный интерфейс сокетов
Взаимодействие с установлением соединения

- Сервер
  - Регистрация входящих запросов на соединение
    int listen(int s, int backlog);
    - s – дескриптор сокета
    - backlog – максимальное число ожидающих запросов
  - Прием запроса на соединение
    - int accept(int s, struct sockaddr *addr, socklen_t *len);
    - s – дескриптор сокета
    - addr – адрес буфера для размещения адреса клиента
    - len – размер буфера, предназначенного для адреса клиента

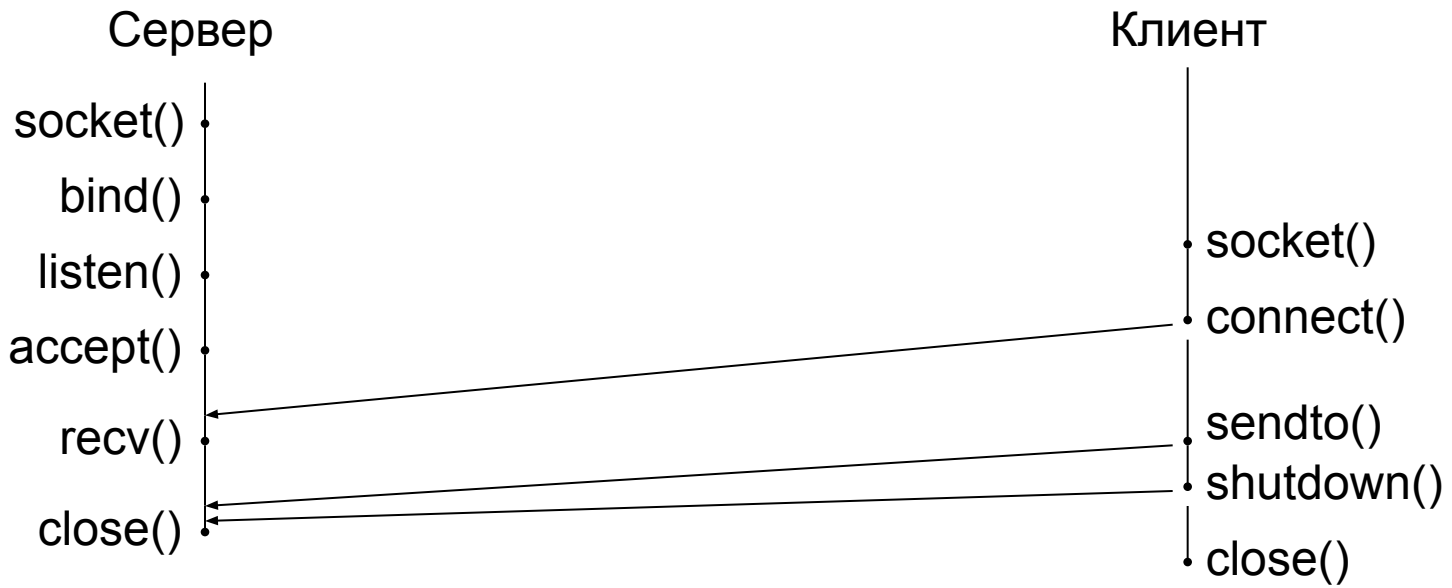
Программный интерфейс сокетов
Взаимодействие с установлением соединения

- Клиент
  - Создание сокета
    - int socket(int domain, int type, int protocol);
  - Запрос на установление соединения с сервером
    - int connect(int s, struct sockaddr *addr, socklen_t *len);
    - s – дескриптор сокета
    - addr – адрес буфера, в котором размещен адрес сервера
    - len – размер буфера addr

Программный интерфейс сокетов
Взаимодействие с установлением соединения

- Передача/прием
  - size_t recv(int s, void *buffer, size_t length, int flags);
  - size_t send(int s, const void *message, size_t length, int flags);
  - s – дескриптор сокета
  - buffer – адрес буфера приема/передачи
  - length – размер буфера в байтах
  - flags – флаги операции

Программный интерфейс сокетов
Взаимодействие с установлением соединения



- Клиент и сервер
  - Отключение сокета
    - int shutdown(int socket, int how);
    - s – дескриптор сокета
    - how – тип отключения (SHUT_RD, SHUT_WR, SHUT_RDWR)
  - Закрытие сокета
  - int close(int s);

Заключение

- Протоколы уровня Хост-Хост обеспечивают передачу данных между процессами
  - UDP – ненадежную доставку сообщений
  - TCP – надежную доставку потоков данных
- При создании приложений используется программный интерфейс сокетов

Тема следующей лекции

- Domain Name System (DNS) – система доменных имен

Вопросы для обсуждения

Литература

- Сети TCP/IP. Ресурсы Microsoft Windows 2000 Server. – М.: Русская редакция, 2001.
- В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 2001.