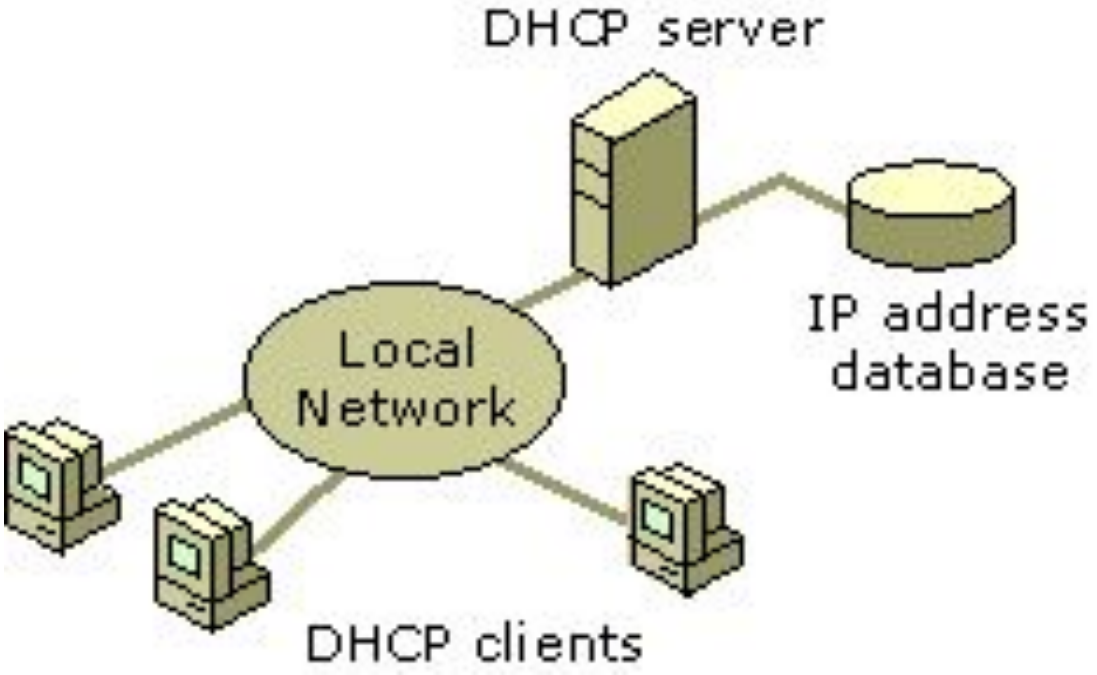
Протокол DHCP. Отобрбажение локальных адресов. Служба DNS. Фрагментация IP пакетов

Лекция №12

Протокол DHCP

Dynamic Host Configuration Protocol (DHCP) - протокол динамической конфигурации хостов

RFC 2132



Основные понятия DHCP

- **Область** — это полный последовательный диапазон допустимых IP-адресов в сети.
- **Исключаемый диапазон** — это ограниченная последовательность IP-адресов в области, которая исключается из числа адресов, предлагаемых службой DHCP
- **Пул адресов** - адреса, оставшиеся после определения области DHCP и исключаемых диапазонов, образуют доступный пул адресов в области.
- **Аренда** — это интервал времени, задаваемый DHCP-сервером, в течение которого компьютер- клиент может использовать назначенный IP-адрес.

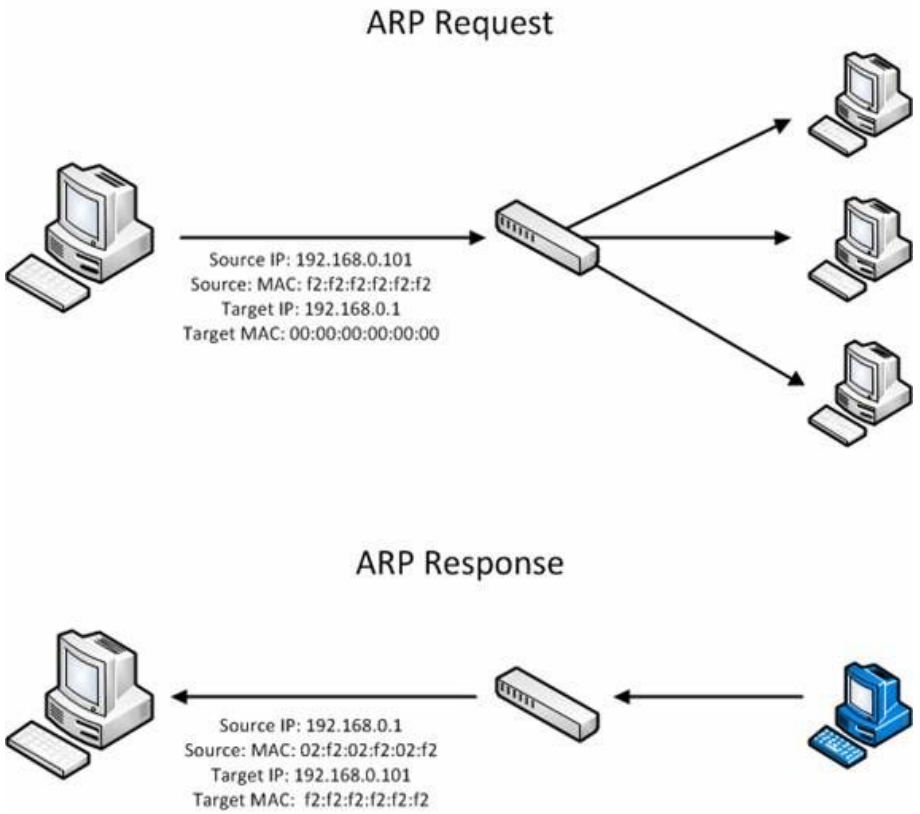
Механизм динамического выделения адресов

1. Клиент посылает широковещательный (BROADCAST-255.255.255.255) запрос DHCPDISCOVER всем серверам DHCP
2. Все активные серверы посылают широковещательный ответ DHCPOFFER. Клиент принимает все ответы, инициализацию делает по адресу канального уровня (MAC-адрес для Ethrnet)
3. Клиент выбирает один из предложенных адресов и посылает широковещательно DHCPREQUEST, которое должно содержать параметр Server Identifier, чтобы указать, какой сервер им выбран
4. Сервер посылает широковещательно DHCPACK
5. Клиент может работать

Протокол ARP

Address Resolution Protocol (ARP) - протокол разрешения локальных адресов

RFC 826



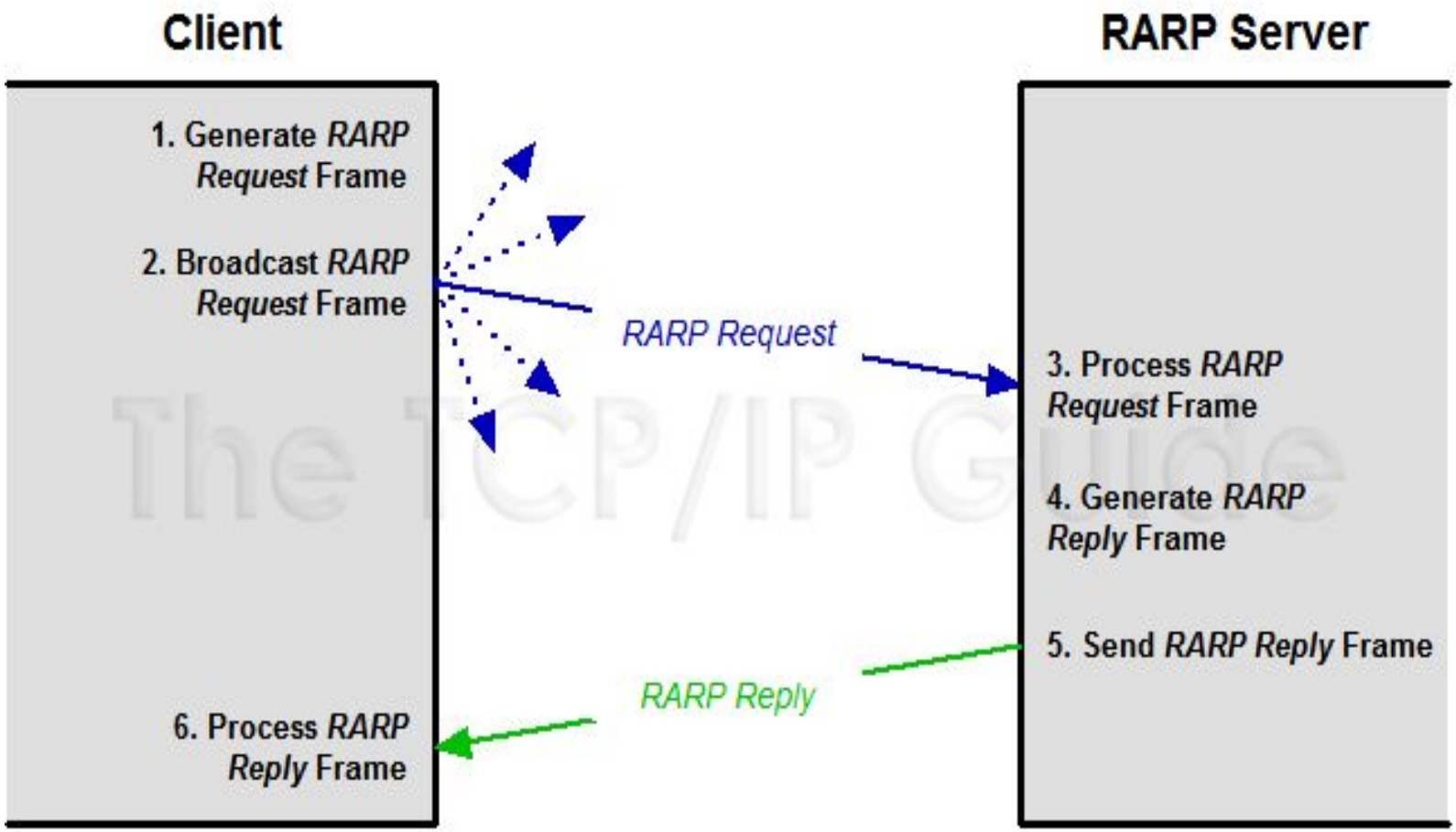
Формат пакета ARP

Протокол RARP

Reverse Address Resolution Protocol (RARP) - протокол сетевого уровня модели OSI, выполняет обратное отображение адресов, то есть преобразует физический адрес в IP-адрес.

RFC 903

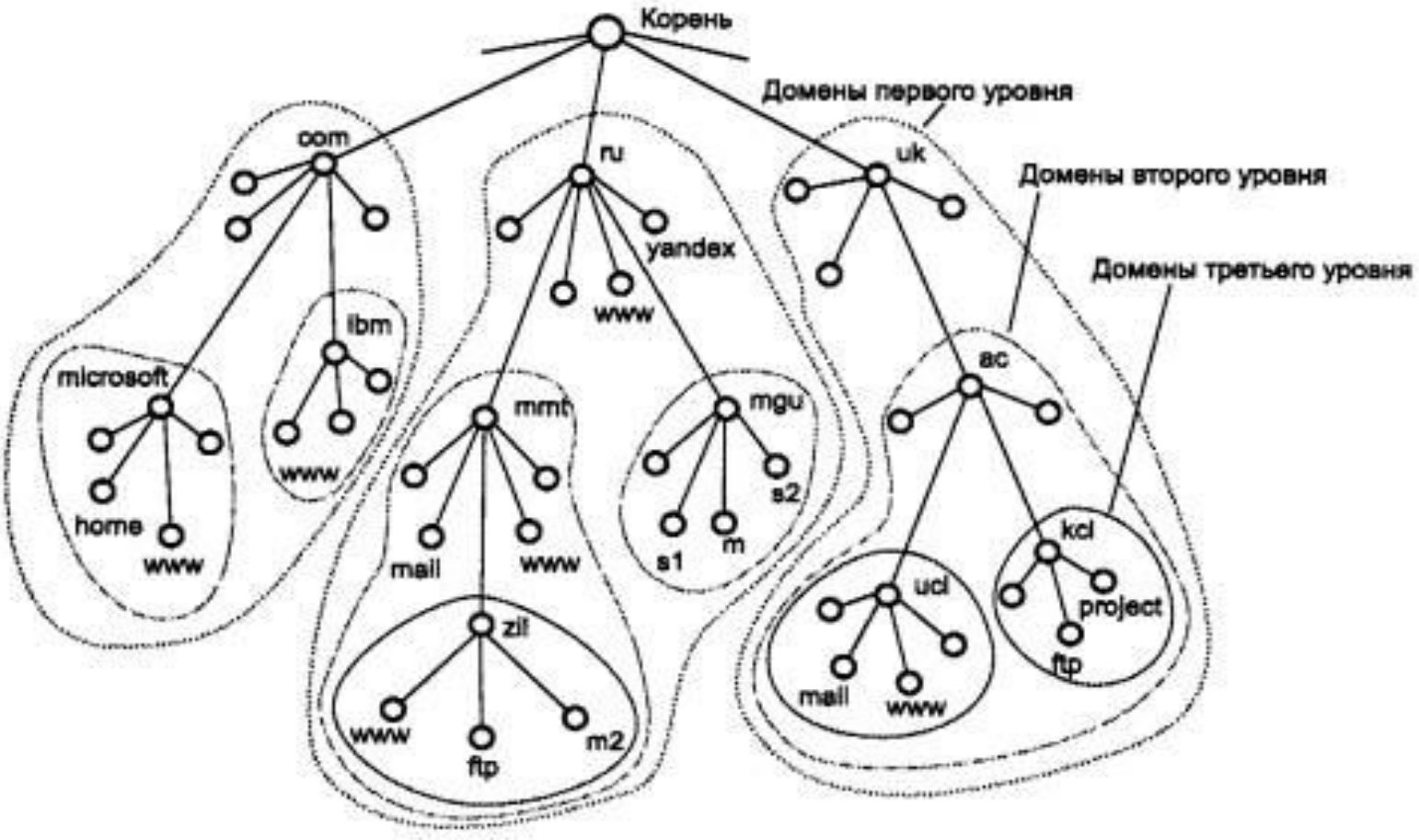
Работа протокола RARP



Система доменных имен DNS

DNS (Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста

Древовидная структура DNS



Домены первого уровня

Итеративная схема разрешения доменного имени

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени;

2. DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени;

3. DNS-клиент делает запрос следующего DNS-сервера, который ото его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Рекурсивная схема разрешения доменного имени

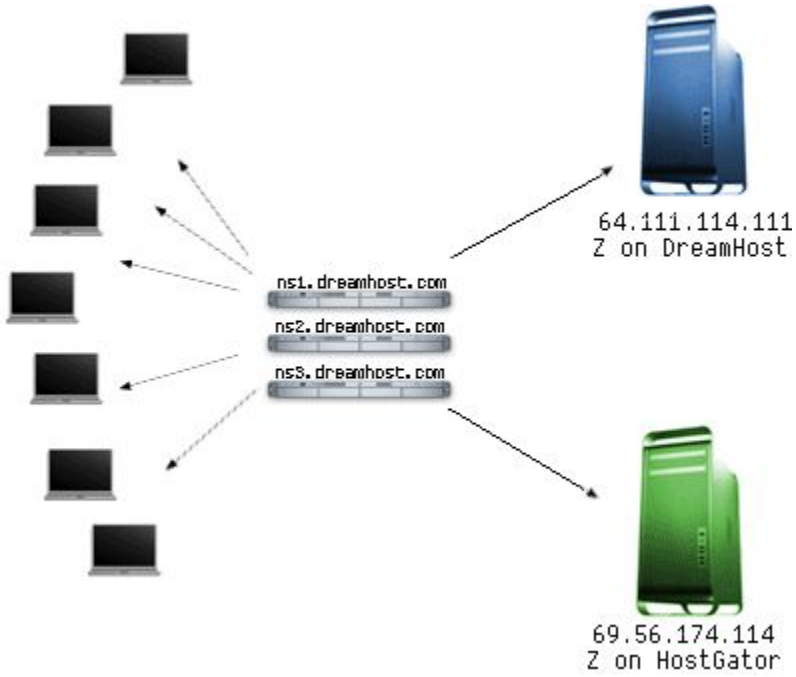
1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, который обслуживает поддомен, к которому принадлежит имя клиента;

2. Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также может соответствовать случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше;

3. Если же локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в первом варианте; получив ответ, он передает его клиенту, который все это время просто ждал его от своего локального DNS-сервера.

Алгоритм Round Robin

Round robin DNS — один из методов распределения нагрузки, или отказоустойчивости за счёт избыточности количества серверов, с помощью управления ответами DNS-сервера в соответствии с некой статистической моделью. Обычно применяется к таким Интернет-протоколам, как Веб-серверы, FTP-серверы.

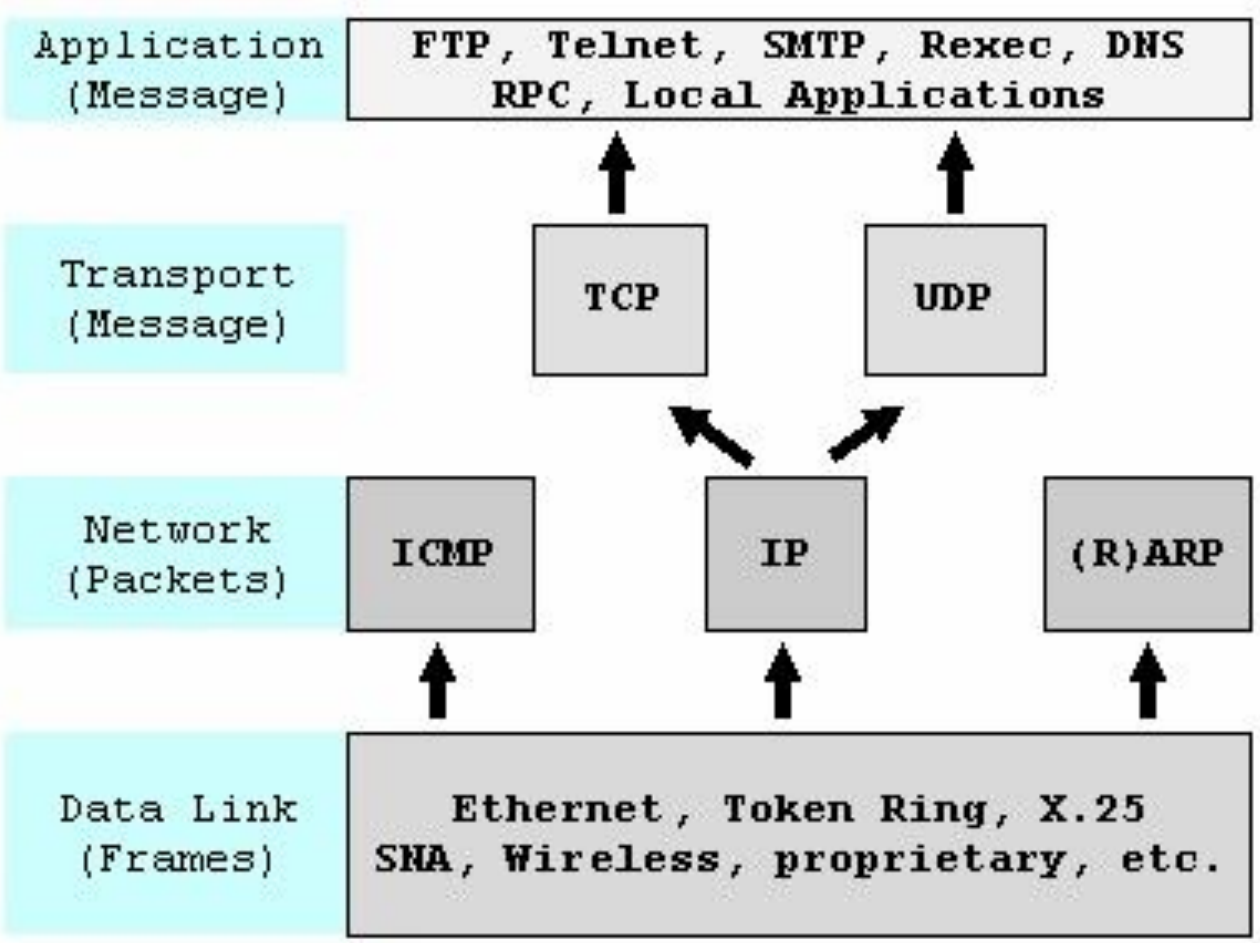


Протокол TCP

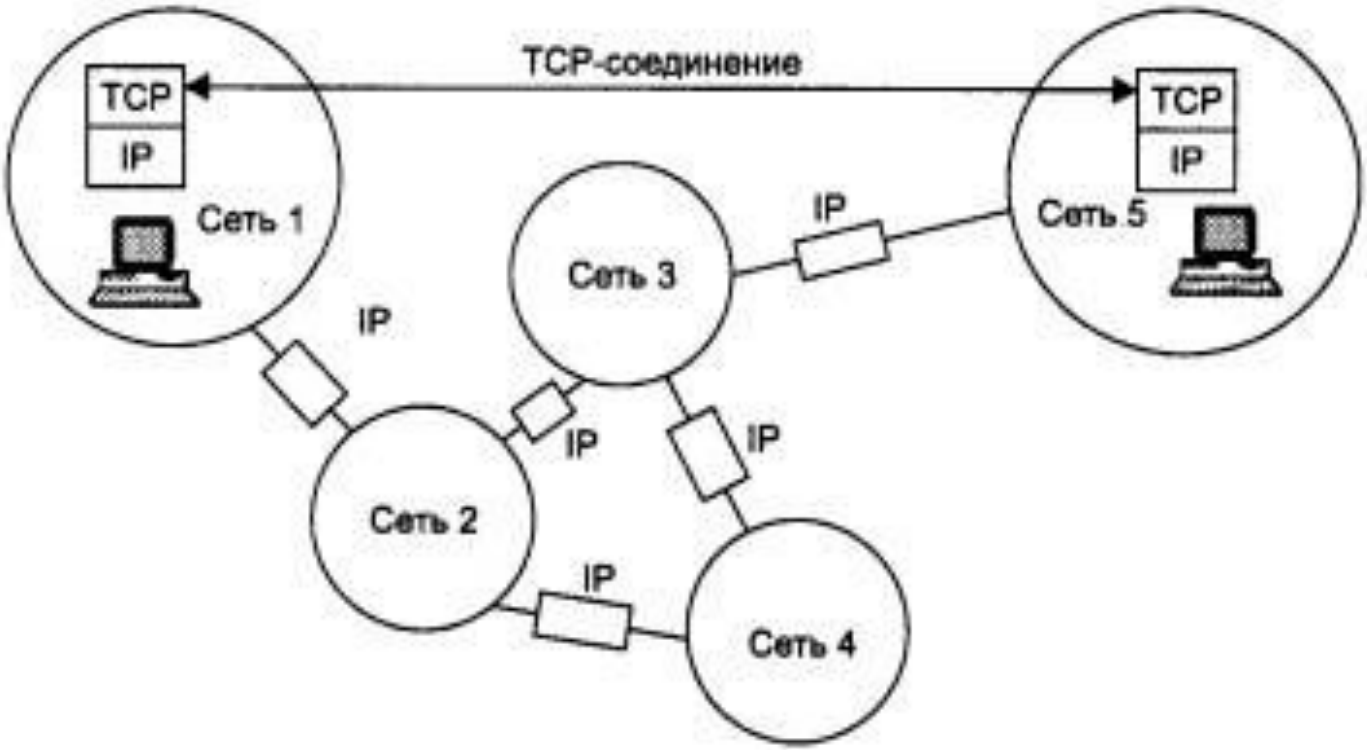
TCP (Transmission Control Protocol) — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета.

В отличие от **UDP** гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи.

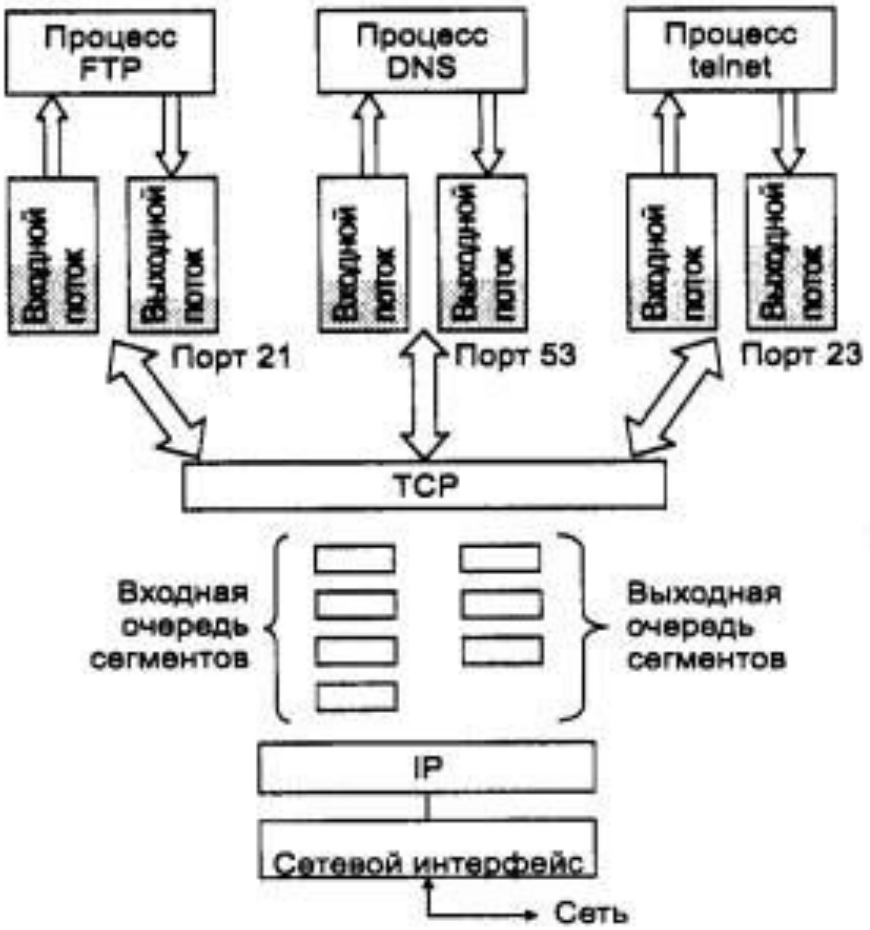
Положение TCP в стеке



Организация канала связи



Порты и потоки в TCP



Скользящее окно в TCP

Описание сетевого соединения

Сокет отправителя =

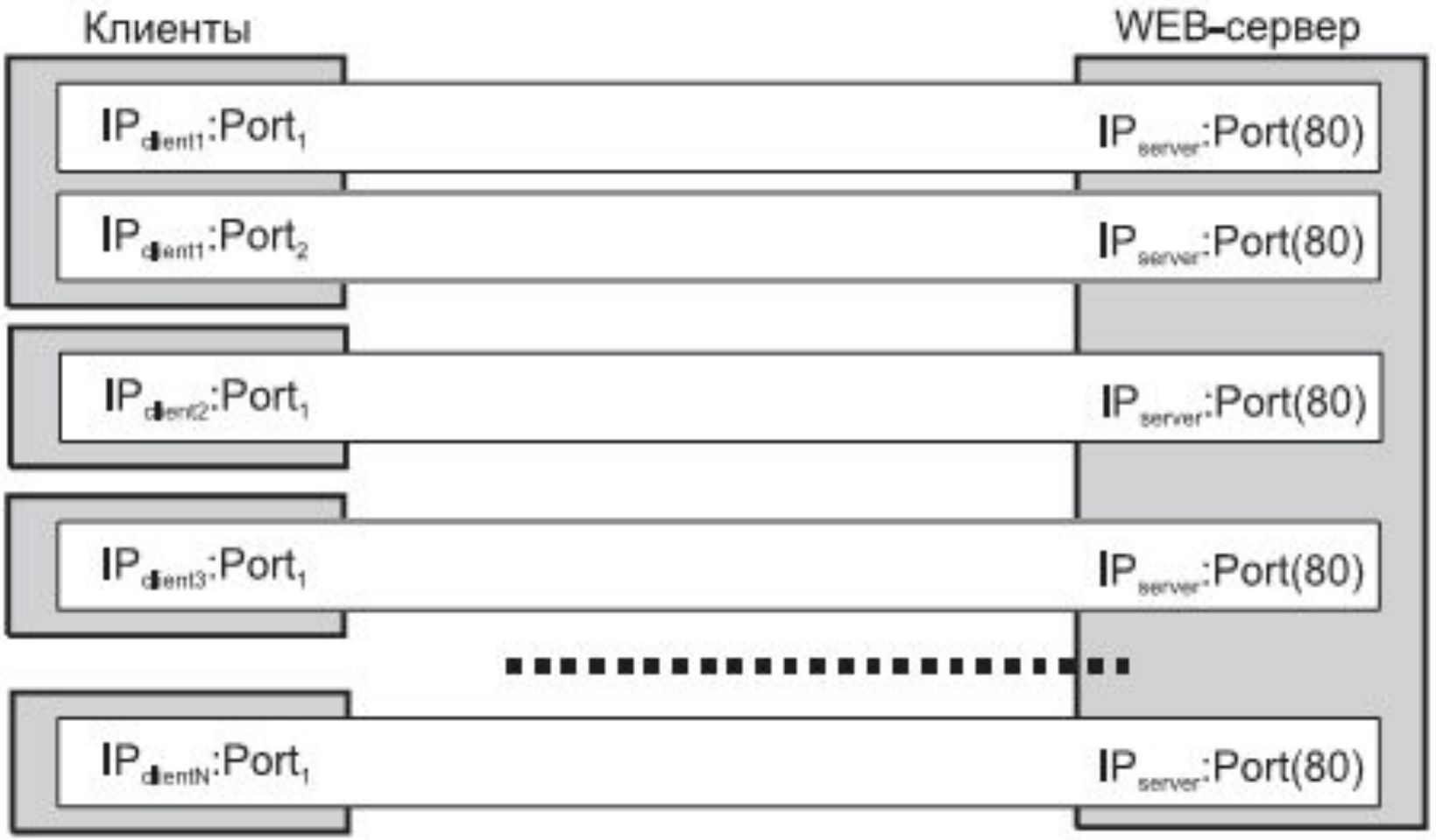
IP-адрес отправителя (IP_S) + номер порта отправителя (P_S)

Сокет адресата =

IP-адрес адресата (IP_D) + номер порта адресата (P_D)

Ансамбль $\mathbf{IP_SP_S + IP_DP_D}$ уникально описывает сокет

Пример соединений



Клиент-серверное приложение

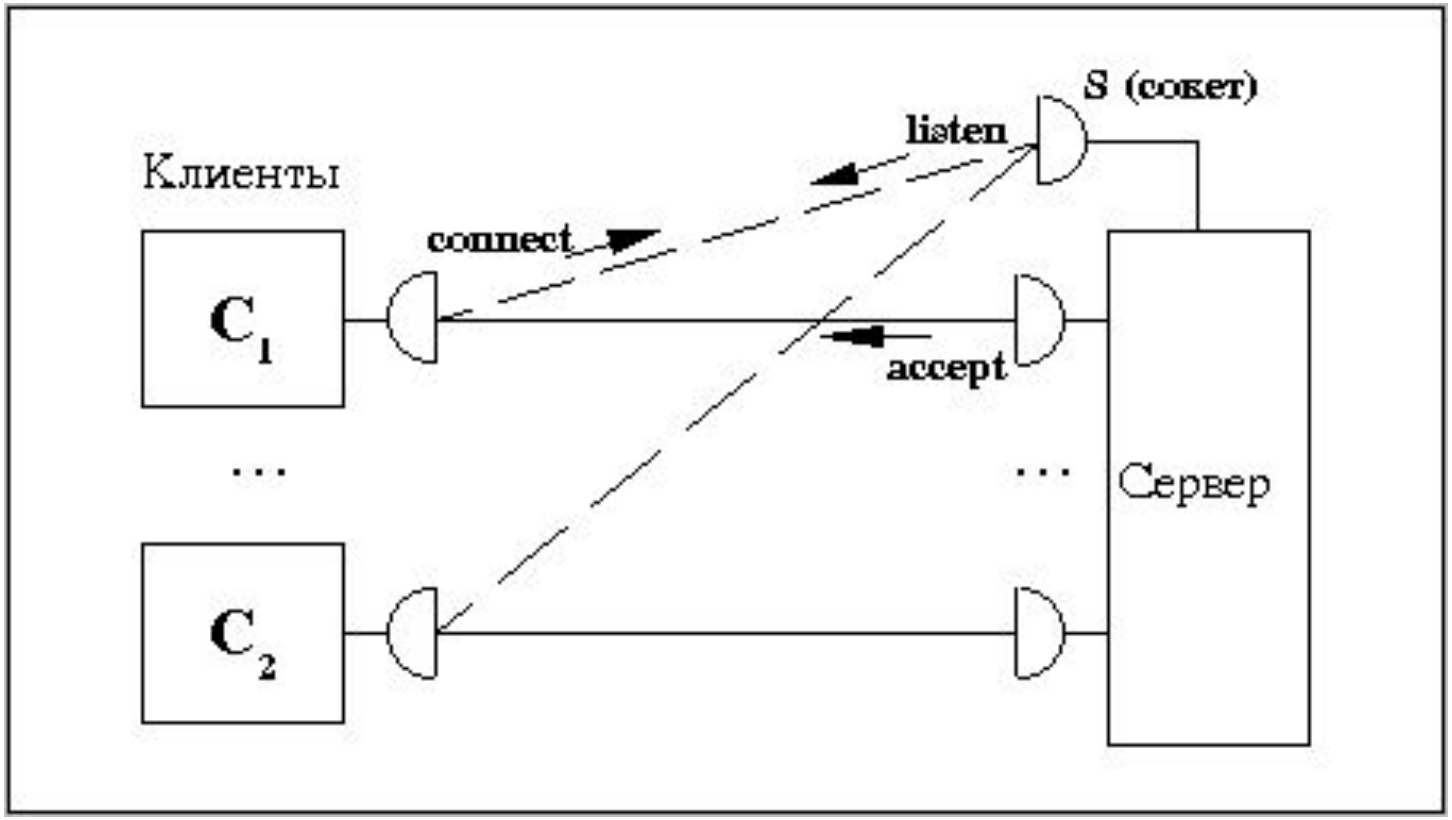
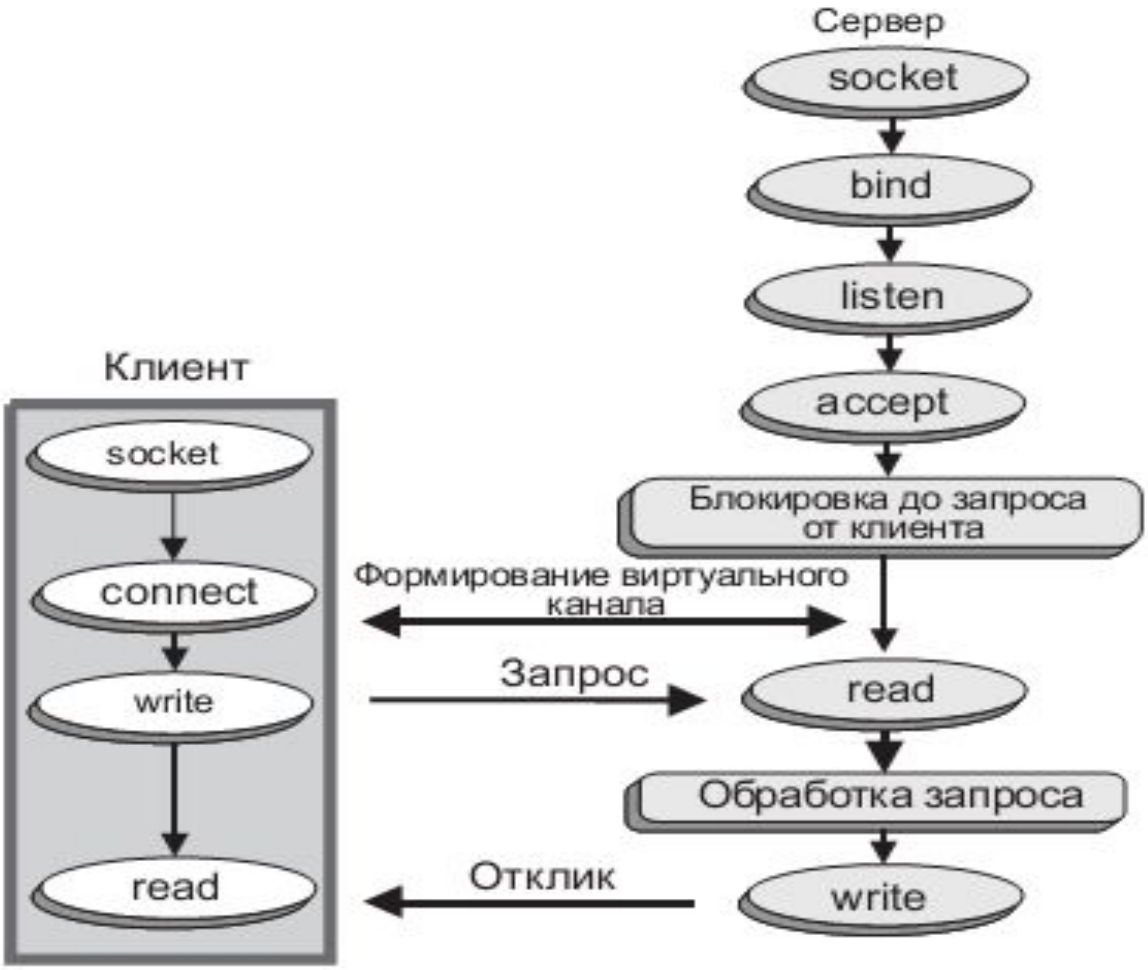


Схема взаимодействия



Создание сокета

s=**socket**(INT AF, INT type, INT protocol)

- **AF** (address_family) - PF_INET, PF_UNIX
- **type** – тип коммуникаций
  - SOCK_STREAM – Надежная доставка TCP
  - SOCK_RAW – Протоколы нижнего уровня
  - SOCK_DGRAM – Режим дейтаграмм
- **Protocol** – код используемого протокола
  - IPPROTO_TCP – протокол TCP
  - IPPROTO_UDP – протокол UDP

<sys/socket.h> - содержит описание всех типов

Таблица дескрипторов

Элемент таблицы дескрипторов:

- код семейства протоколов
- код типа сервиса
- локальный IP-адрес
- удаленный IP-адрес
- номер локального порта
- номер удаленного порта.

Присвоение IP адреса

r=**bind**(s, const struct socketaddr far*name, int namelen)

- s – дескриптор сокета
- struct sockaddr {
  u_short sa_family; // код протокола
  char sa_data[14]; // IP адрес:Порт
  };
- namelen - длина параметра name

Присваивается свой IP адрес и порт!!!

Подсоединение клиента к серверу

R=**connect**(s, const struct socketaddr
FAR*name, int namelen)

- s – дескриптор сокета
- name – идентификатор адреса места назначения
- namelen - длина адреса

Присваивается IP адрес и порт назначения

Ожидание сервером запросов

R=**listen**(s, int backlog)

- s – дескриптор сокета
- максимальный размер очереди для приходя
  запросов соединения

Извлечение запросов из очереди

R=**accept**(s, struct sockaddr FAR*addr, int FAR*addrlen)

- s – дескриптор сокета, который прослушивает соединение
- addr – указатель на структуру, которая содерж адрес
- addrlen — указатель на длину адреса

Чтение и запись

R=**write**(s, buf, len)
R=**read**(s, buf, len)

- s – дескриптор сокета
- buf - имя массива, подлежащего пересылке (и предназначенного для приема)
- len - длина этого массива

Чтение и запись

R=**send**(s, buf, len,flags)
R=**recv**(s, buf, len,flags)

- s – дескриптор сокета
- buf - имя массива, подлежащего пересылке (и
  предназначенного для приема)
- len - длина этого массива
- flags - управление передачей данных

Дейтаграмный режим