

# Компьютерные сети (NET101)

Транспортный уровень TCP/IP

# Содержание лекции

---

- Обзор транспортного уровня
  - Функции транспортного уровня
  - Порты, мультиплексирование и демультимплексирование
- Протоколы транспортного уровня TCP/IP
  - Протокол UDP
  - Протокол TCP
    - Описание протокола
    - Общие принципы обеспечения надёжной доставки
- Примеры приложений использующих TCP и UDP



# Функции транспортного уровня

---

Прикладно й
Представи тельный
Сессия
Транспортн ый
Сетевой
Звено данных
Физический

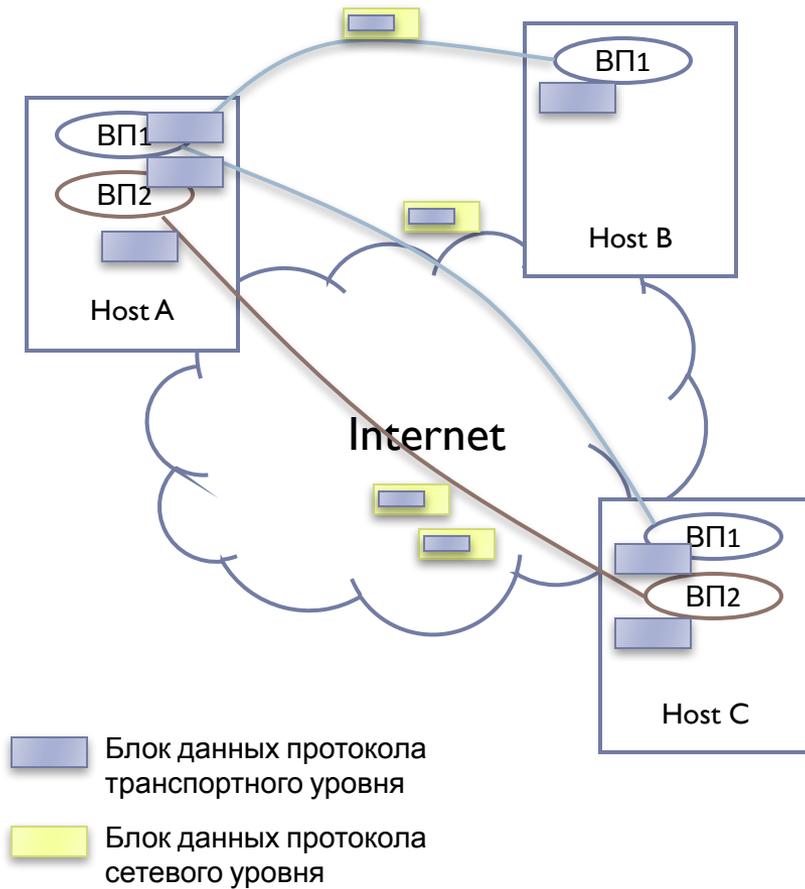
Обеспечивает передачу данных  
между вычислительными  
процессами

## Функции:

- [Де]мультиплексирование данных от и к ВП
- Обеспечение передачи данных
  - дейтаграммно
  - с установлением логического соединения
    - Сегментация
    - Обеспечение надёжной доставки



# Взаимодействующие вычислительные процессы

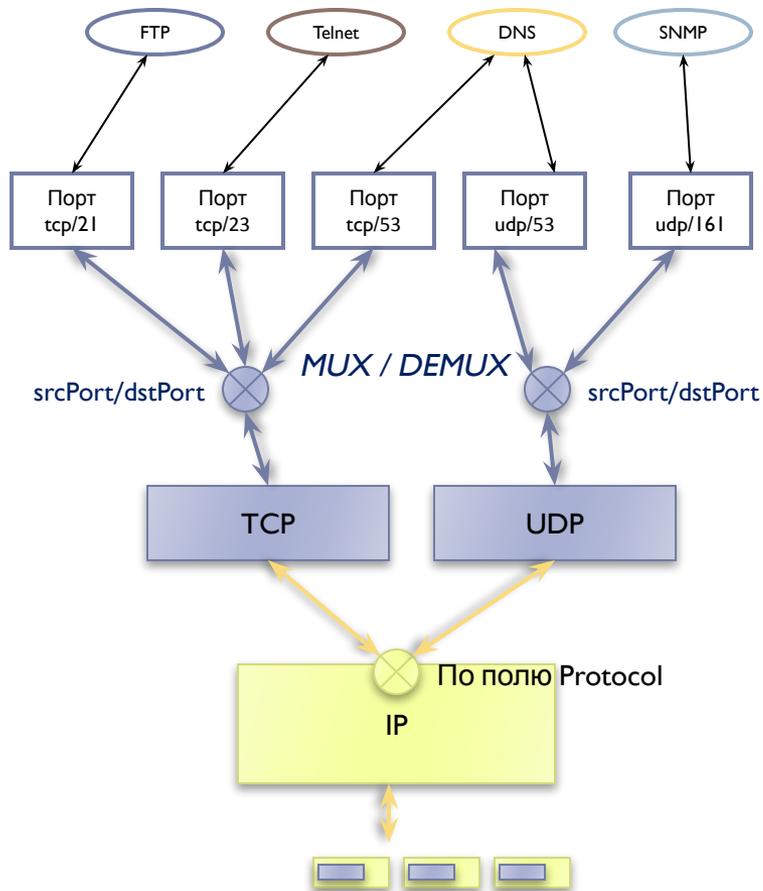


Как указать какому процессу предназначены данные?



«Передай эти данные процессу на узле А,  
который подключен к порту номер N»

# Понятие порта



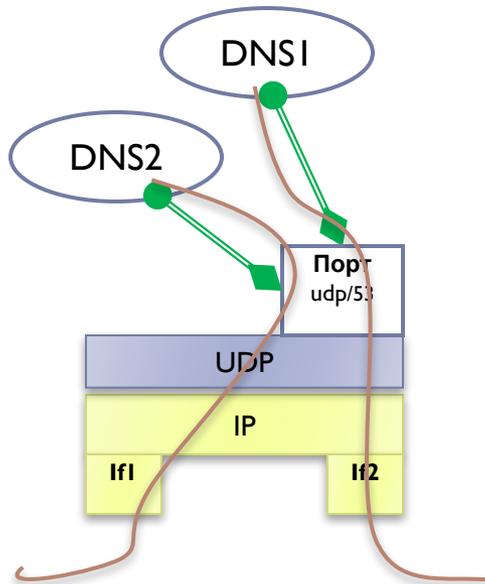
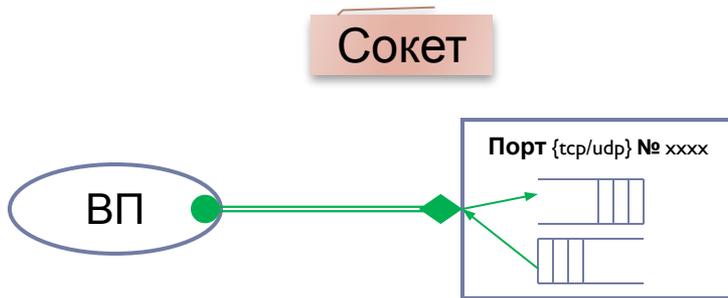
□ *Порт* – это ...

- средство двустороннего сообщения ВП с внешним миром
- пара очередей ОС (приёма и передачи), к которым подключается процесс, открывая порт

□ **Номера портов**

- передаются в заголовке транспортного уровня
- делятся на
  - well-known (RFC1700, RFC3232)
  - динамически назначаемые

# На самом деле, цепочка длиннее...



- *Сокет* – программный объект ВП, связывающий его с портом.
- А точнее:
  - для UDP-сокетов:
    - {local\_ip, local\_port}
  - для TCP-сокетов:
    - {local\_ip, local\_port} (listen)
    - {local\_ip, local\_port, remote\_ip, remote\_port} (connected)

# Протоколы транспортного уровня TCP/IP

---

- Протокол пользовательских дейтаграмм (UDP, User Datagram Protocol)
- Протокол управления передачей (TCP, Transmission Control Protocol)



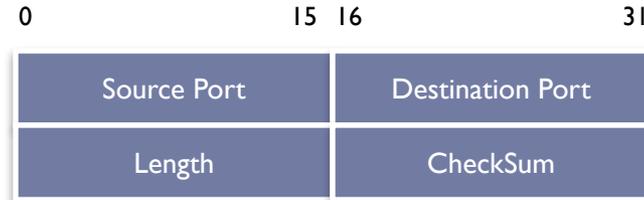
# Протокол UDP

- Реализует сервис дейтаграммной передачи сообщений (best effort)
- Одно сообщение пользователя – одна UDP-дейтаграмма
- Возможна многоадресная рассылка (mcast, bcast)



# Формат UDP-заголовка

---

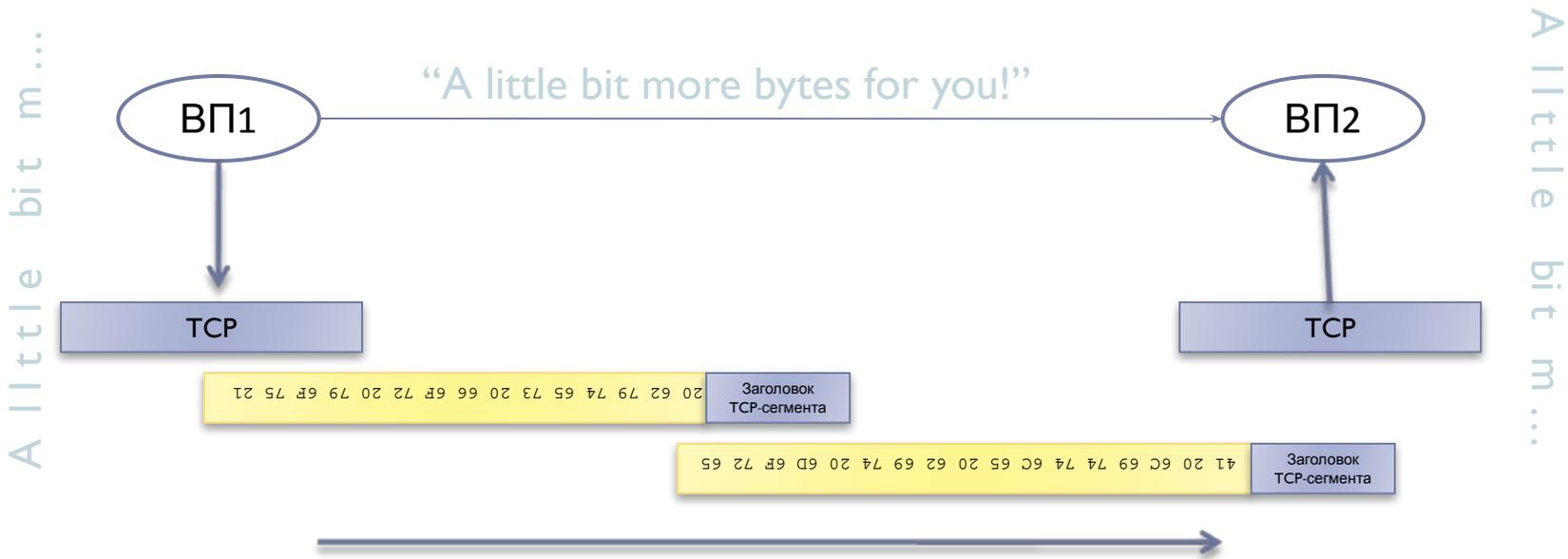


- Source Port и Destination Port – указывают на отправляющий и принимающий процессы.
- Поле Length содержит длину в байтах UDP заголовка и UDP данных.
- CheckSum – контрольная сумма UDP охватывает UDP заголовков и UDP данные.



# Протокол TCP

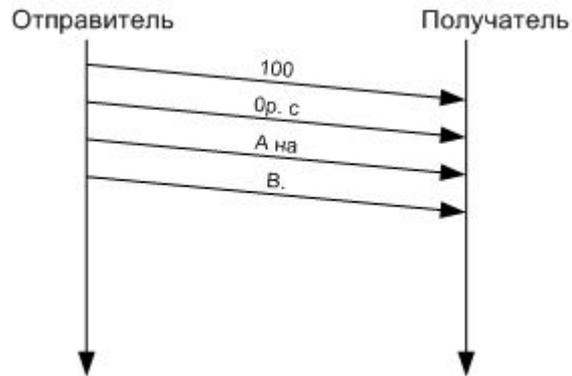
- Предоставляет основанный на логическом соединении надежный сервис **потока байтов**
- Логическое соединение «точка-точка» – многоадресная рассылка невозможна



# Общие вопросы надёжной доставки

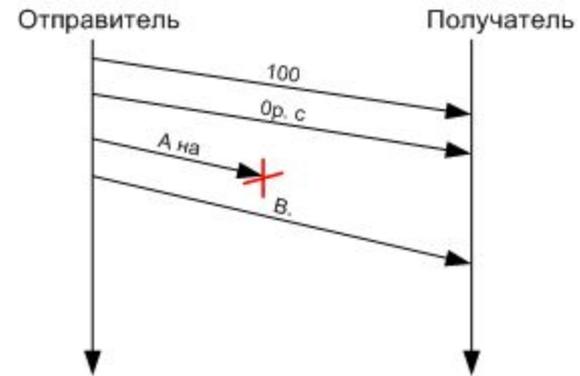
**Задача:** передать сообщение «1000р. с А на В.»

Нет ошибок



Принято: «1000р. с А на В.»

Потеря сегмента

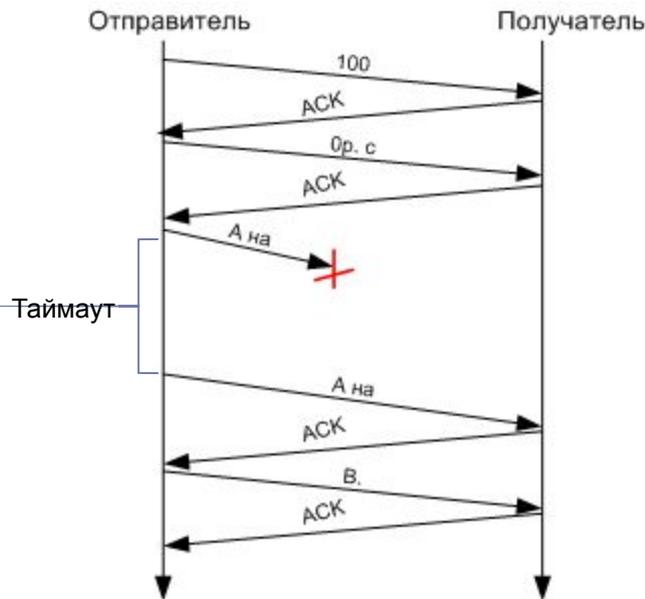


Принято: «1000р. сВ.»

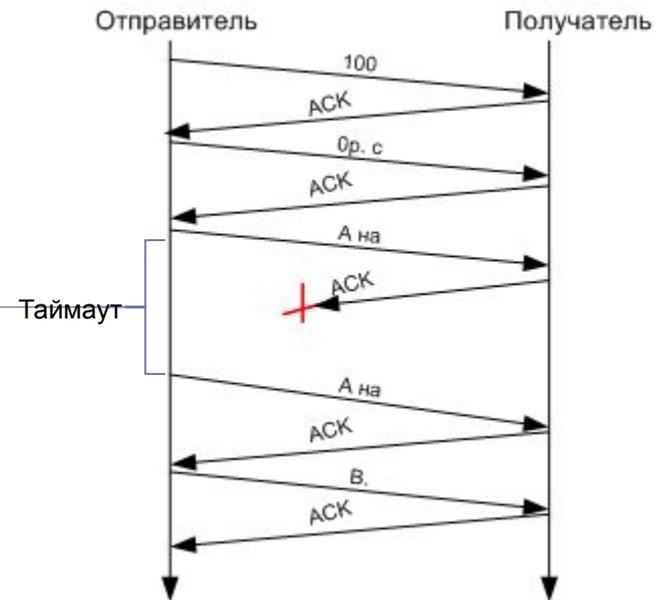


# Обеспечение надёжной доставки

- **Использование подтверждений, таймаутов и повторной передачи** – решает проблему потери сегментов



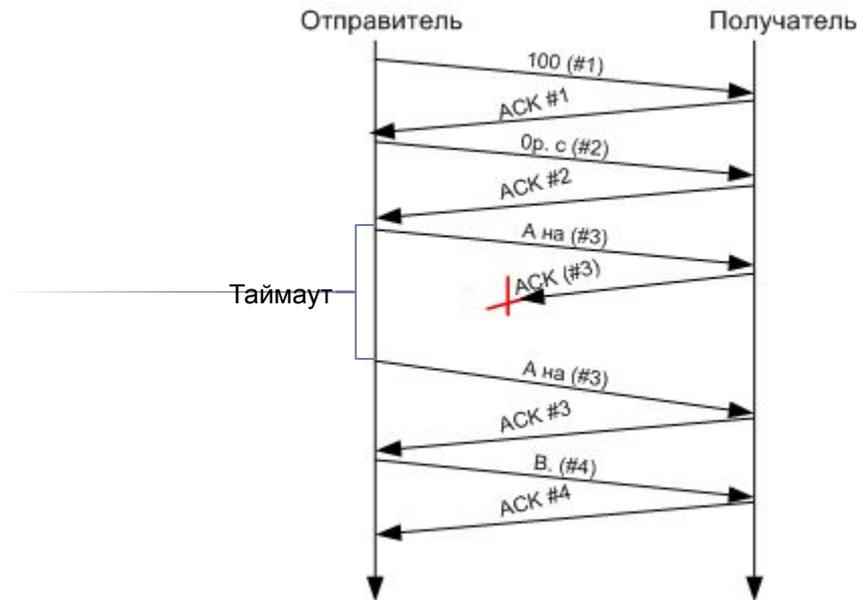
Принято: «1000р. с А на В.»



Принято: «1000р. с А на А на В.»

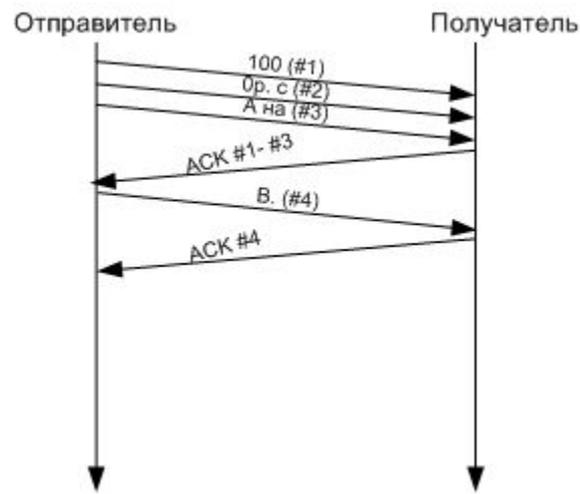
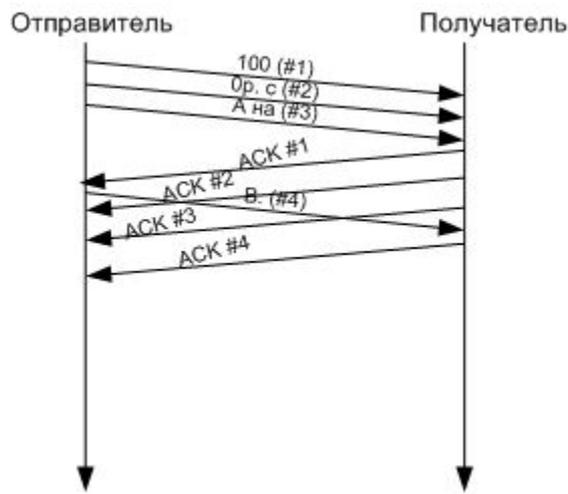
# Обеспечение надёжной доставки

- **Нумерация сегментов** и упорядочение их на принимающей стороне – решает проблему нарушения порядка доставки и дублирования сегментов



# Обеспечение производительной надёжной доставки

- Использование техники конвейеризации (pipelining)
  - Можно передать несколько сегментов, но не более окна доступности, ещё до того как будет получено первое подтверждение
- Использование групповых подтверждений



# Надёжность доставки в ТСР

---

- При отправке сегмента, ТСР **устанавливает таймер, ожидая**, что с удаленного конца придет **подтверждение на этот сегмент**. Если подтверждение не получено по истечении времени, **сегмент передается повторно**.
- Когда ТСР принимает данные от удаленной стороны соединения, он **отправляет подтверждение**.
- Если принят **сегмент с неверной контрольной суммой**, он **отбрасывается** и подтверждение не генерируется.
- ТСР **упорядочивает** принимаемые **сегменты** и отбрасывает дубликаты.
- ТСР осуществляет **контроль потока данных\***.



# Формат TCP-заголовка

---

<b>Source Port</b>		<b>Destination Port</b>	
<b>Sequence Number</b>			
<b>Acknowledge Number</b>			
<b>HdrLen</b> 4bit	<b>Reserved</b> 6bit	<b>Flags</b>	<b>Window Size</b>
<b>Checksum</b>		<b>Urgent Pointer</b>	
<b>Options</b>			<b>Padding</b>



# Флаги TCP

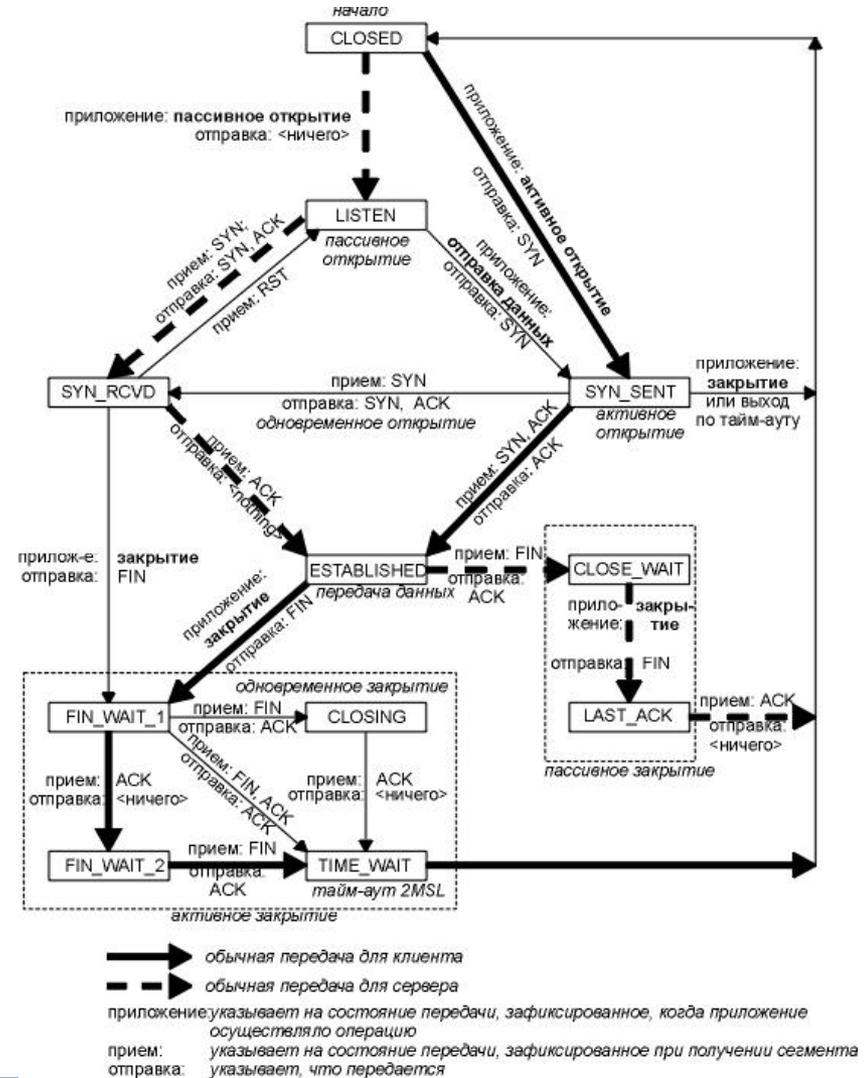
---

- URG – признак срочных данных.
- ACK – учесть номер подтверждения.
- PSH – получатель должен передать эти данные приложению как можно скорее.
- RST – сбросить соединение.
- SYN – синхронизирующий номер последовательности для установления соединения.
- FIN – отправитель заканчивает посылку данных.



# Фазы TCP-соединения

- Установление соединения
- Обмен данными
- Разрыв соединения

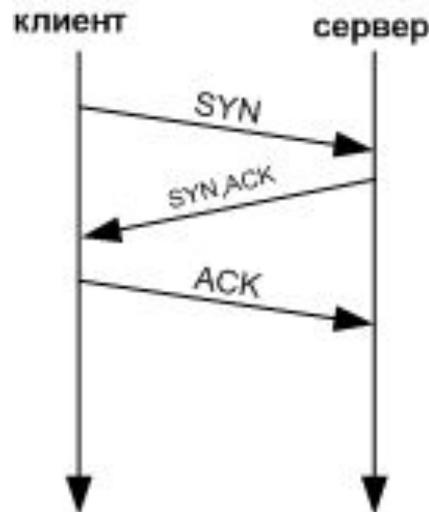


netstat -na -p tcp

# Установление соединения (handshake)

1. Запрашивающая сторона отправляет SYN сегмент, указывая номер порта сервера и исходный номер последовательности клиента. Это сегмент номер 1.
2. Сервер отвечает своим сегментом SYN, содержащим исходный номер последовательности сервера (сегмент 2) и подтверждает приход SYN клиента с использованием ACK (ISN клиента плюс один).
3. Клиент должен подтвердить приход SYN от сервера с использованием ACK (ISN сервера плюс один, сегмент 3).

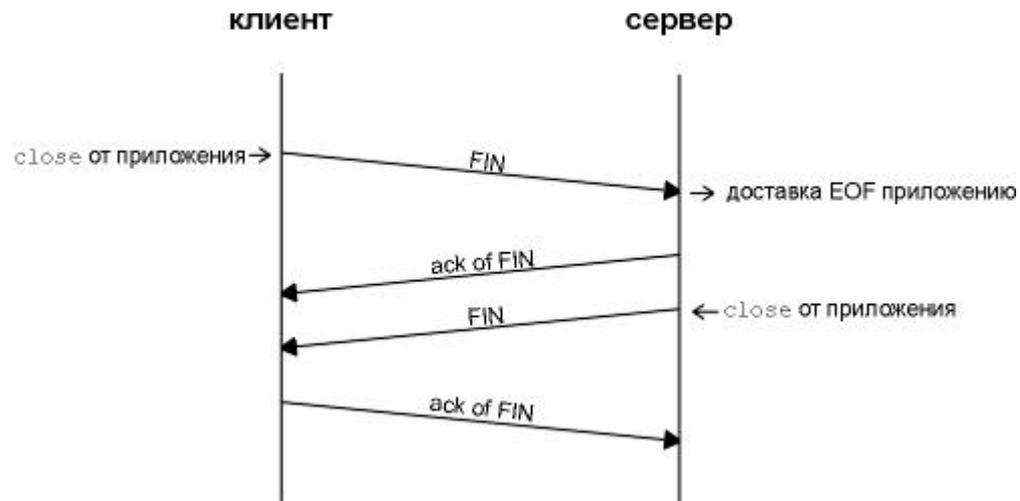
3 сегмента



# Разрыв соединения

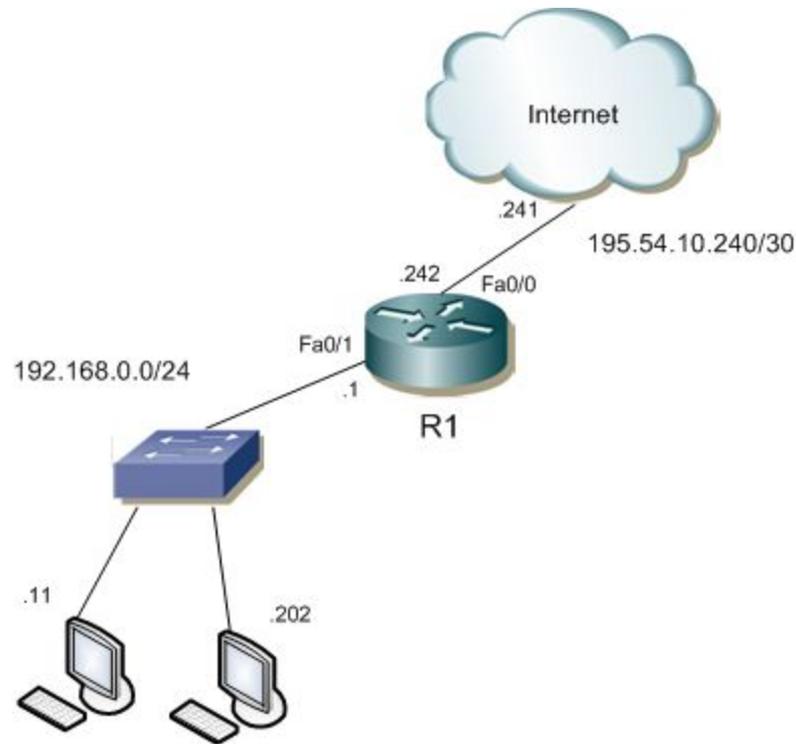
- Каждая из сторон должна закрыть соединение
  - Сторона инициирующая закрытие со своей стороны отправляет сегмент с FIN (сегмент 1).
  - Сторона принявшая FIN должна ответить на него ACK (сегмент 2).

4 сегмента



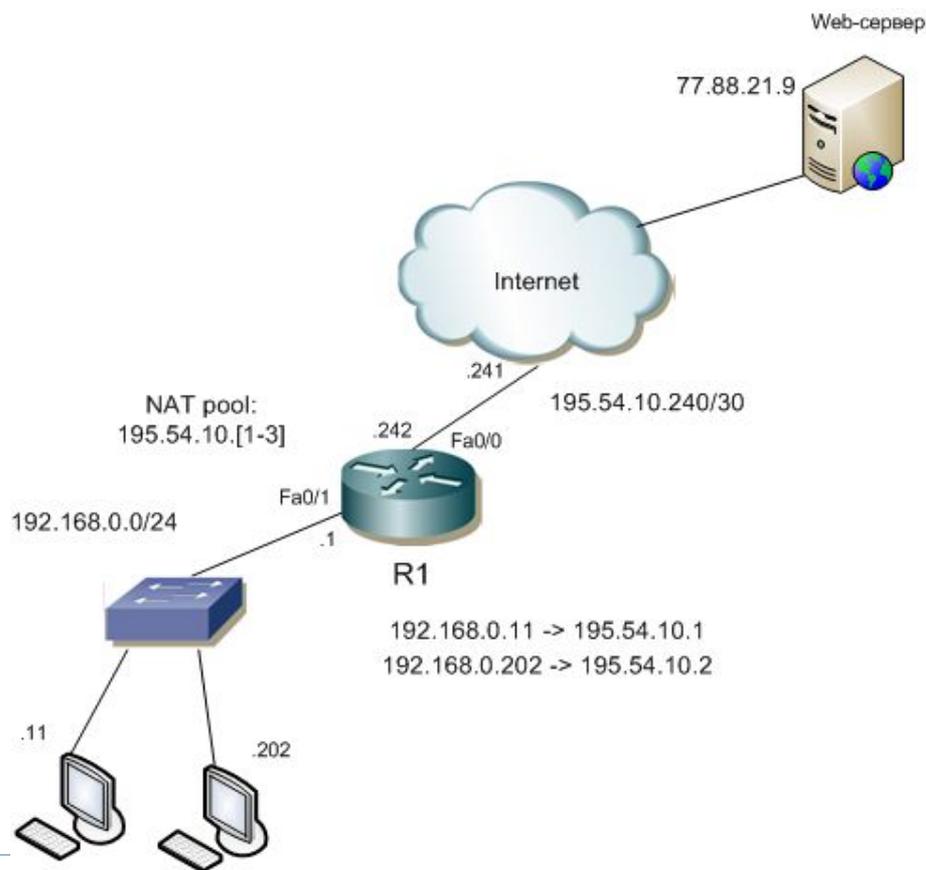
# Сетевая трансляция адресов и портов: NAT/PAT

- Позволяет узлам с приватными адресами взаимодействовать с узлами в Internet
- Реализуется прозрачной для конечных узлов заменой ip-адресов [и портов] отправителя и получателя



# Трансляция сетевых адресов (NAT)

- Узлам с приватными адресами по мере необходимости назначаются адреса из пула



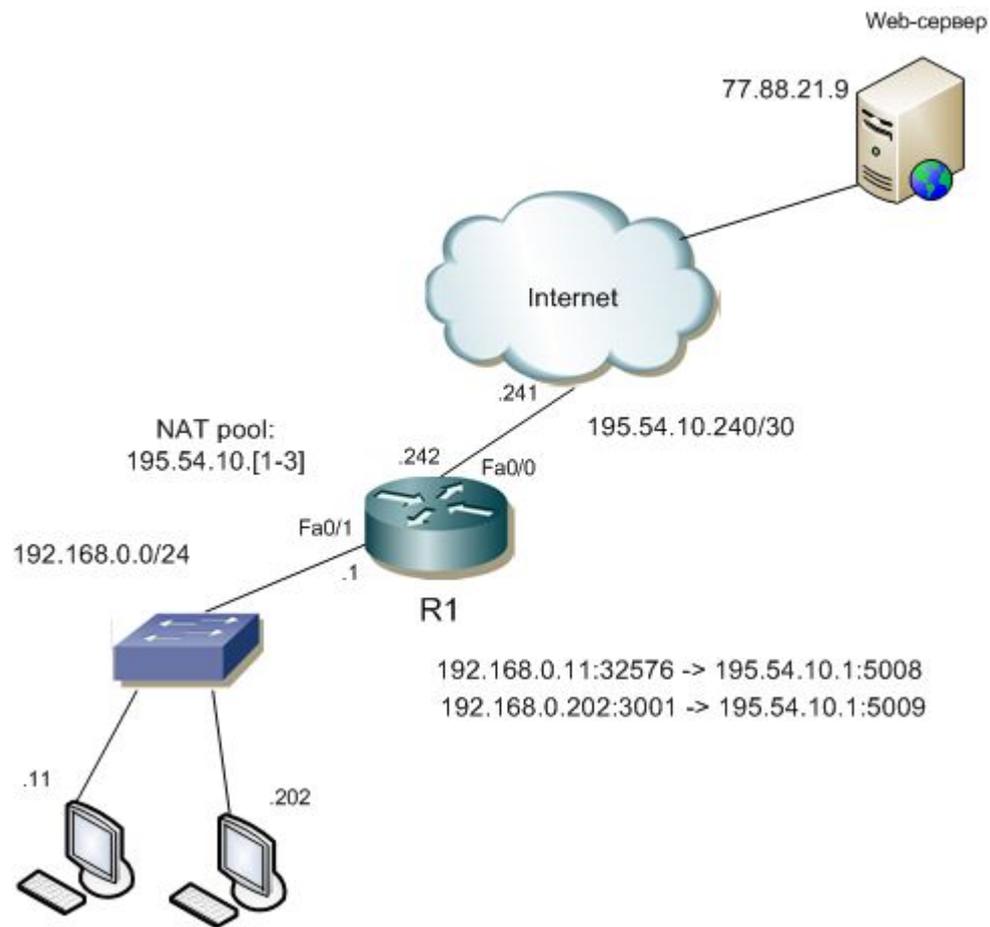
# Трансляция сетевых адресов и портов (PAT)

---

- При выходе за NAT осуществляется замена:
  - {srcip, srcport} □ {ip-from-pool, new-port}
  - {srcip, srcport}, {ip-from-pool, new-port} заносятся в таблицу трансляции
- Обратное:
  - Поиск {dstip, dstport} в таблице среди {ip-from-pool, new-port}
  - Если найдено – замена на сохранённые ранее {srcip, srcport}



# Трансляция сетевых адресов и портов (PAT)



# Приложения, использующие TCP и UDP

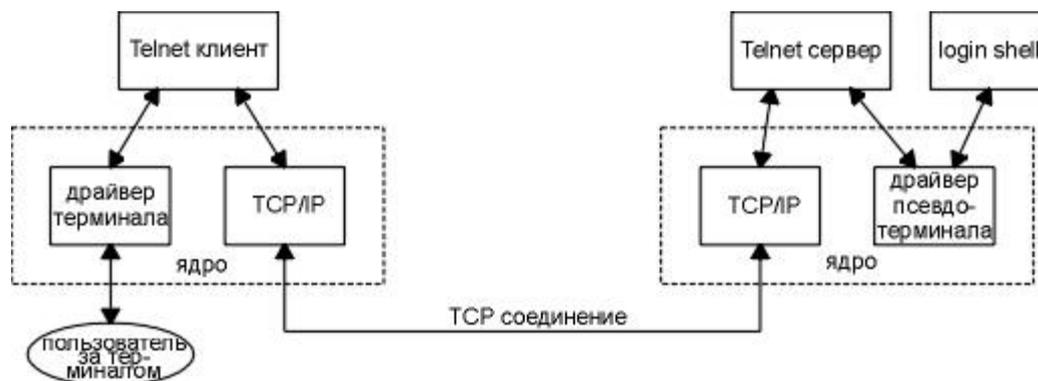
---

- Примеры приложений, использующих TCP:
  - Telnet
  - FTP
  - HTTP
  - SMTP
- Примеры приложений, использующих UDP:
  - DNS
  - NTP
  - SNMP
  - NFS



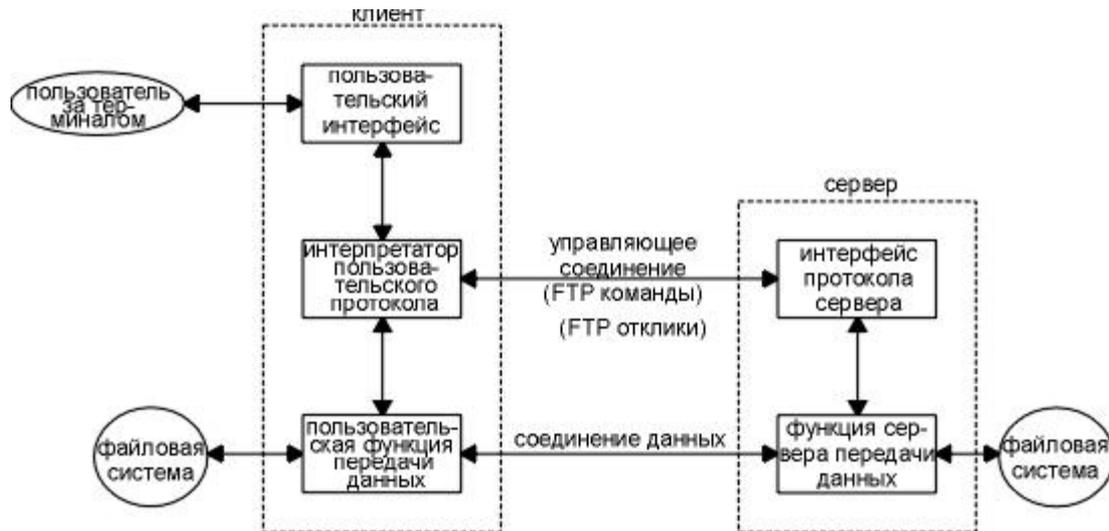
# Протокол Telnet

- Позволяет пользователю работать с удалённой системой в режиме терминала
- «In-band» – управление
- Использует TCP, порт 23



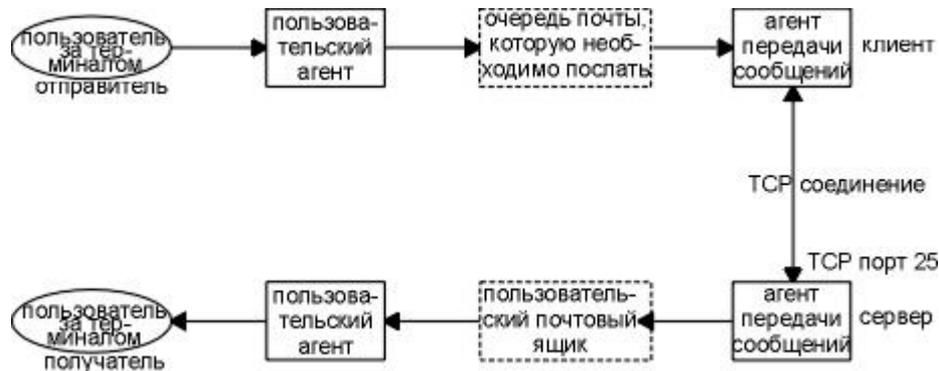
# Протокол FTP

- Реализует сервис передачи файлов через сеть
- «Out-of-band» – управление
- Использует TCP, порт 21 для команд и 20 для данных



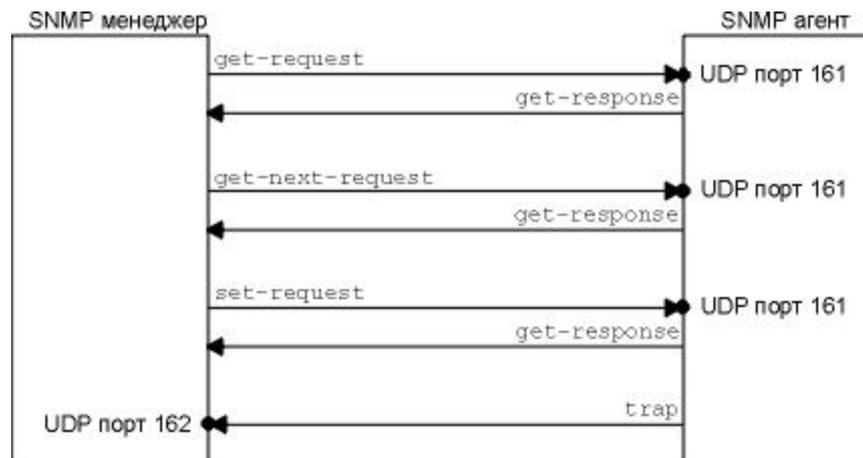
# Протокол SMTP

- Лежит в основе сервиса электронной почты
- Использует TCP, порт 25



# Протокол SNMP

- Протокол управления сетевыми устройствами
- Использует UDP, порт 161 для управления и 162 для уведомлений



# Протокол NFS

- Реализует сервис доступа к файлам удалённой системы
- Использует UDP, порт 2049

