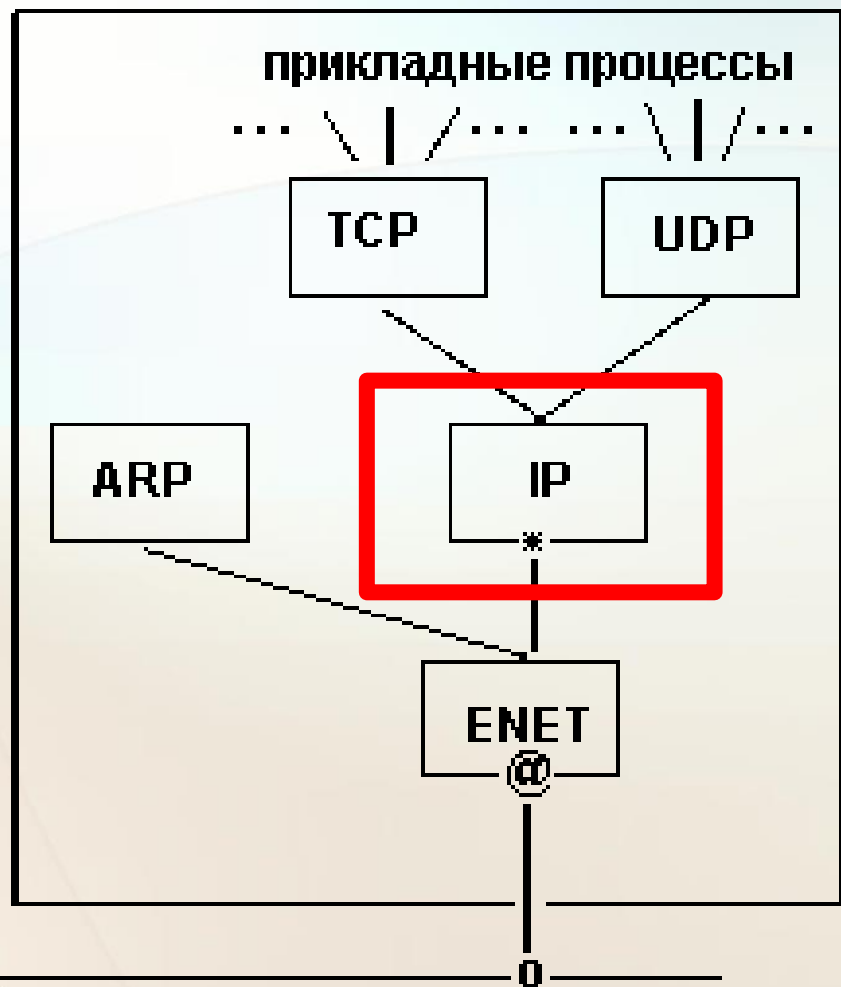


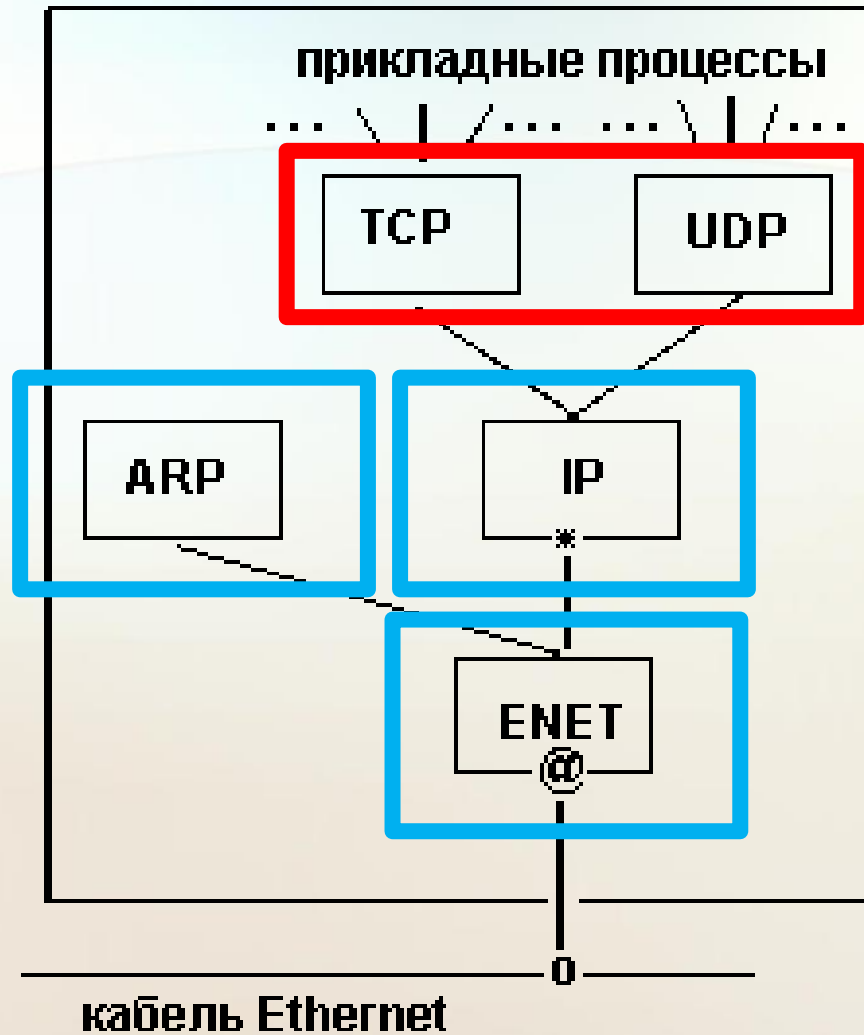
# Протоколы сетевого уровня



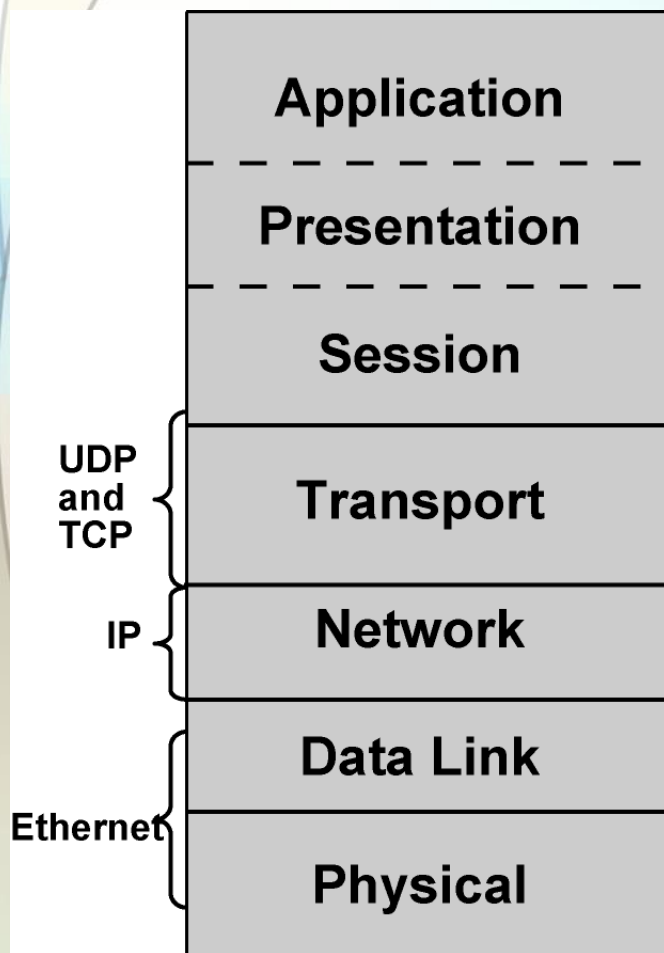
- ICMP,
- IP,
- RIP,
- OSPF,
- IGRP,
- EIGRP,
- IS-IS

кабель Ethernet

# Протоколы верхних уровней



# Транспортный уровень



- Мультиплексирование сессий
- Сегментация данных
- Контроль передачи (когда требуется)
- Поддержка соединений (когда требуется)
- Надёжность (когда требуется)

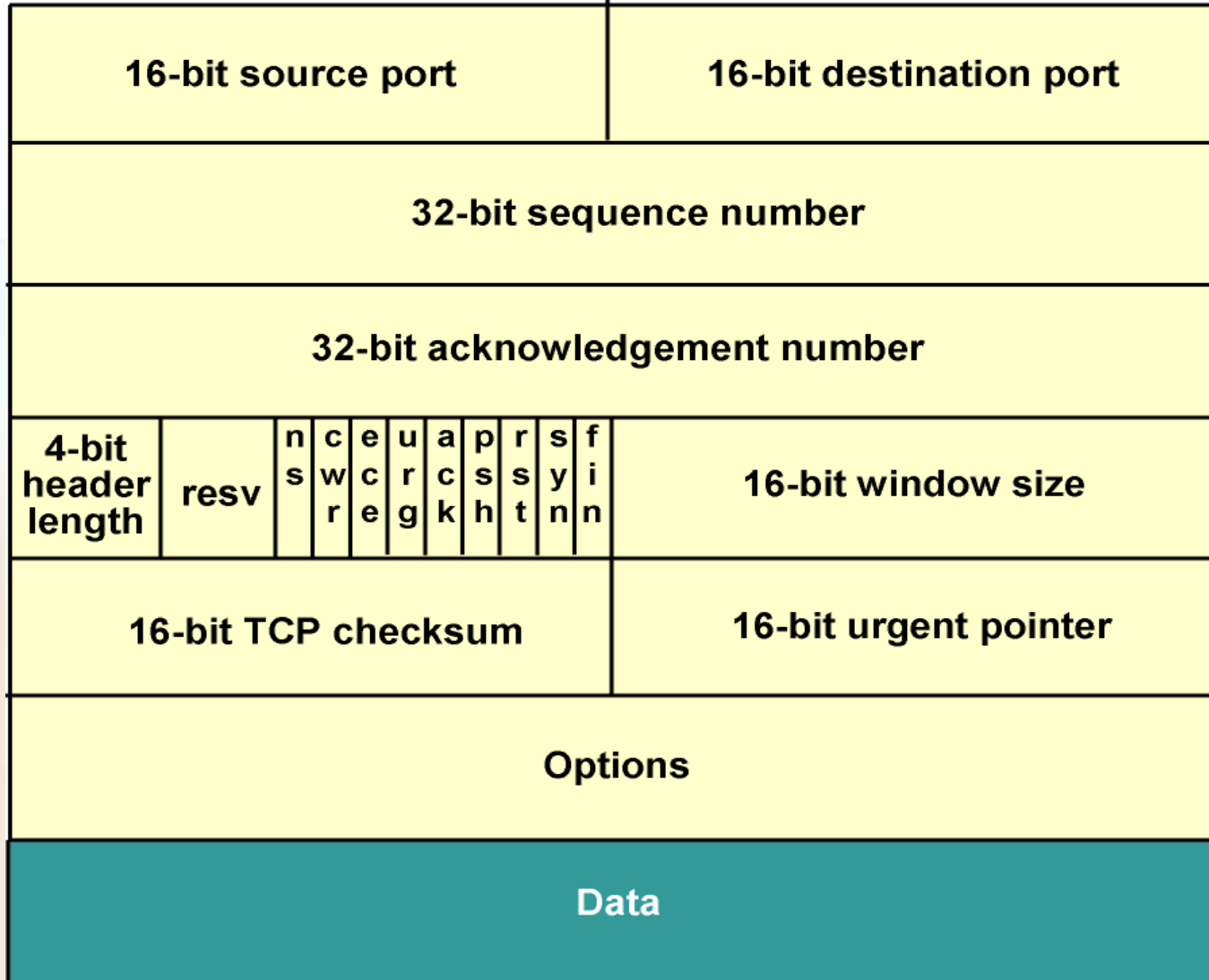
# ТСР и UDP

	Надежность	Скорость
Протокол	ТСР	UDP
Соединения	Есть	Нет
Последовательность пакетов	Есть	Нет
Используется	Почта, передача файлов	Потоковое видео и аудио

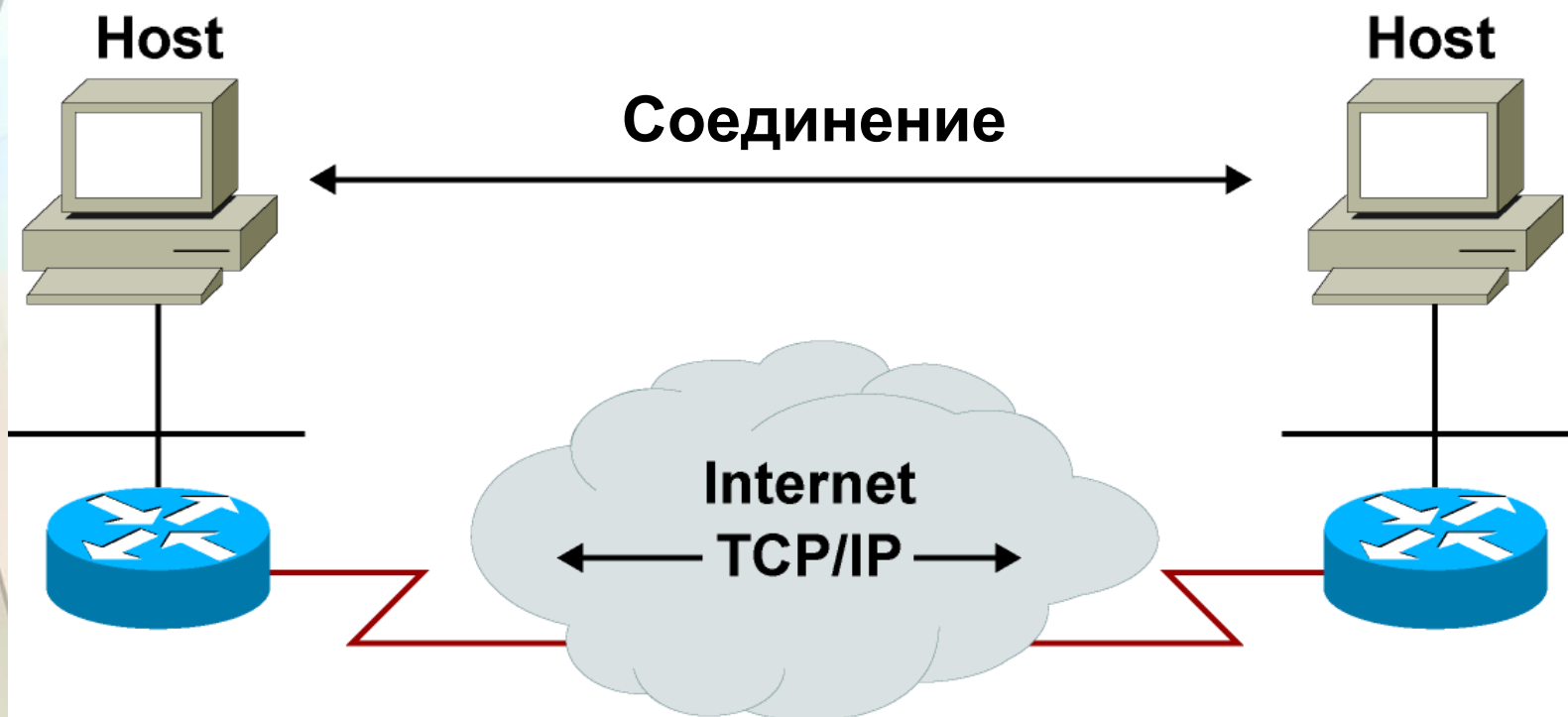
# **Протокол TCP (Transmission Control Protocol)**

- **Ориентирован на установку соединений**
- **Управляет процессом передачи пакетов**
- **Маркирует передаваемые пакеты**
- **Уведомляет о доставке пакетов**
- **Поддерживает обнаружение ошибок передачи**
- **Повторно передаёт утерянные / испорченные пакеты**

# Заголовок TCP-пакета

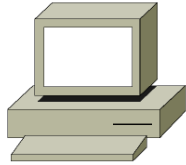


# Процесс установки соединений

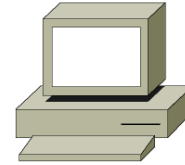


# «Тройное рукопожатие»

Host A



Host B



1

Send SYN  
(SEQ=100 CTL=SYN)

SYN received

2

Send SYN, ACK  
(SEQ=300 ACK=101 CTL=SYN, ACK)

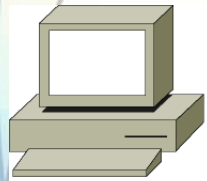
SYN received

3

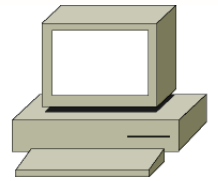
Established  
(SEQ=101 ACK=301 CTL=ACK)



# Управление потоком передачи



**Отправитель**



**Получатель**

**Передача**



**Не готов к приёму!**

**Остановка**



**Буфер приёма  
полон!**

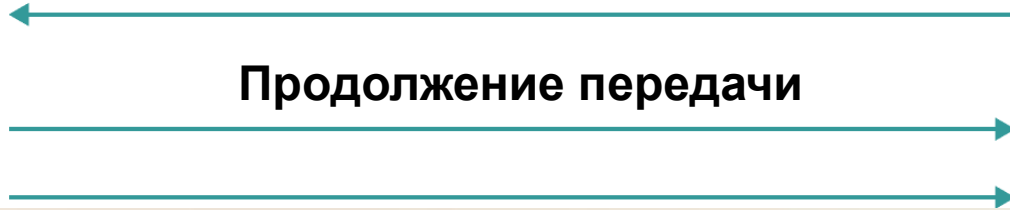
**Обработка  
сегментов**



**Буфер приёма  
готов к работе**

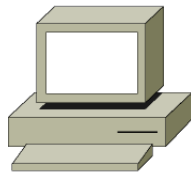
**Можно  
продолжать!**

**Продолжение передачи**



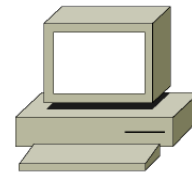
# Уведомления о доставке

Отправитель



Размер «окна» = 1

Получатель



Отправлен сегмент 1

Receive ACK 2

Отправлен сегмент 2

Receive ACK 3

Отправлен сегмент 3

Receive ACK 4

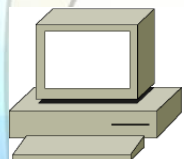
Принят сегмент 1  
Send ACK 2

Принят сегмент 2  
Send ACK 3

Принят сегмент 3  
Send ACK 4

# Фиксированный размер «окна»

Отправитель



Отправка:

сегмент 1  
сегмент 2  
сегмент 3

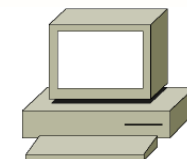
Receive ACK

сегмент 4  
сегмент 5  
сегмент 6

Receive ACK  
Send 7

Размер «окна» = 3

Получатель

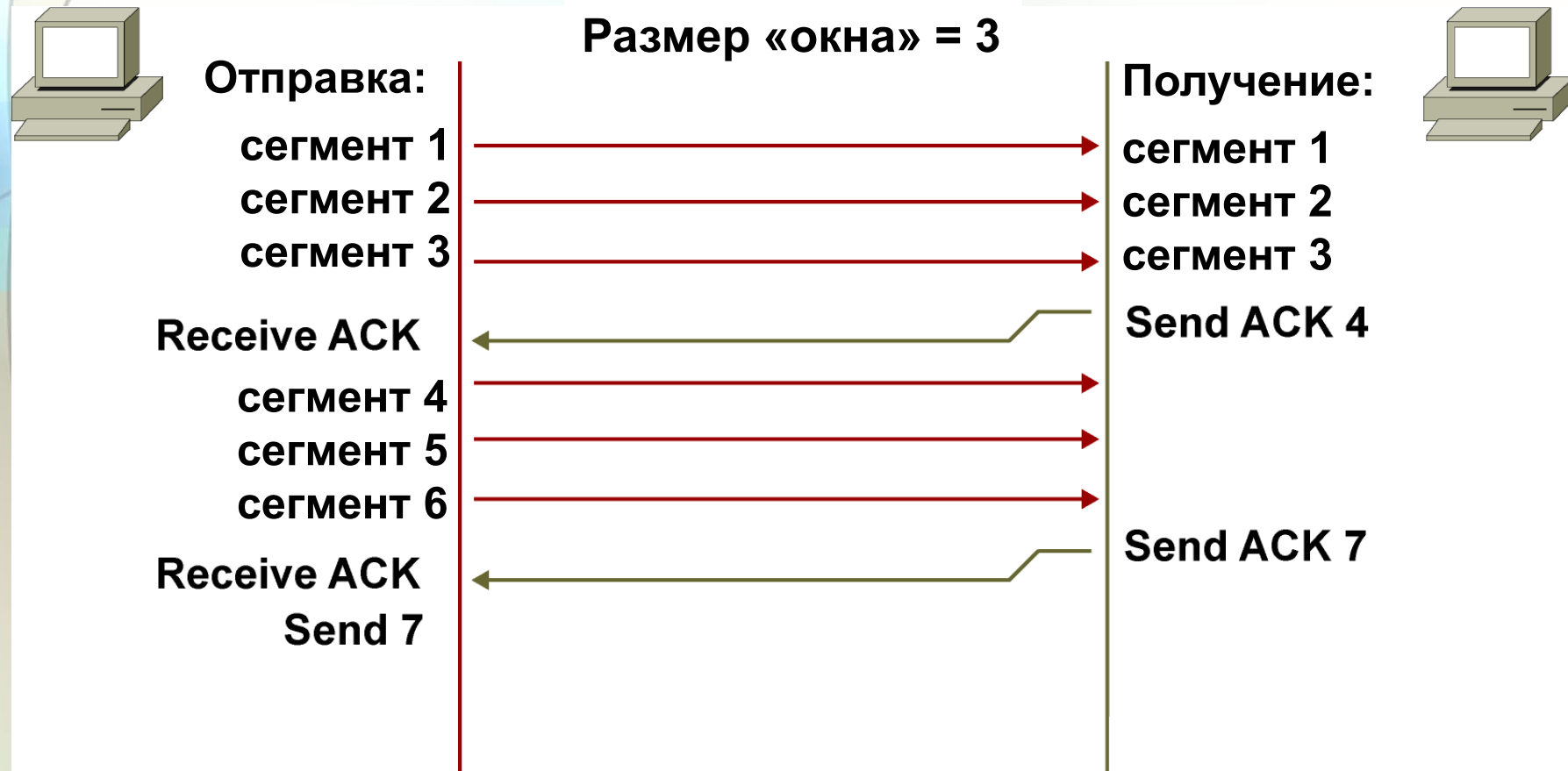


Получение:

сегмент 1  
сегмент 2  
сегмент 3

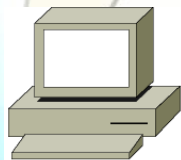
Send ACK 4

Send ACK 7



# «Плавающий» размер окна

Отправитель



Window size = 3  
Send 1

Window size = 3  
Send 2

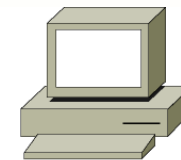
Window size = 3  
Send 3

Window size = 3  
Send 3

Window size = 3  
Send 4



Получатель



ACK 3  
Window size = 2

Сегмент 3  
был  
потерян

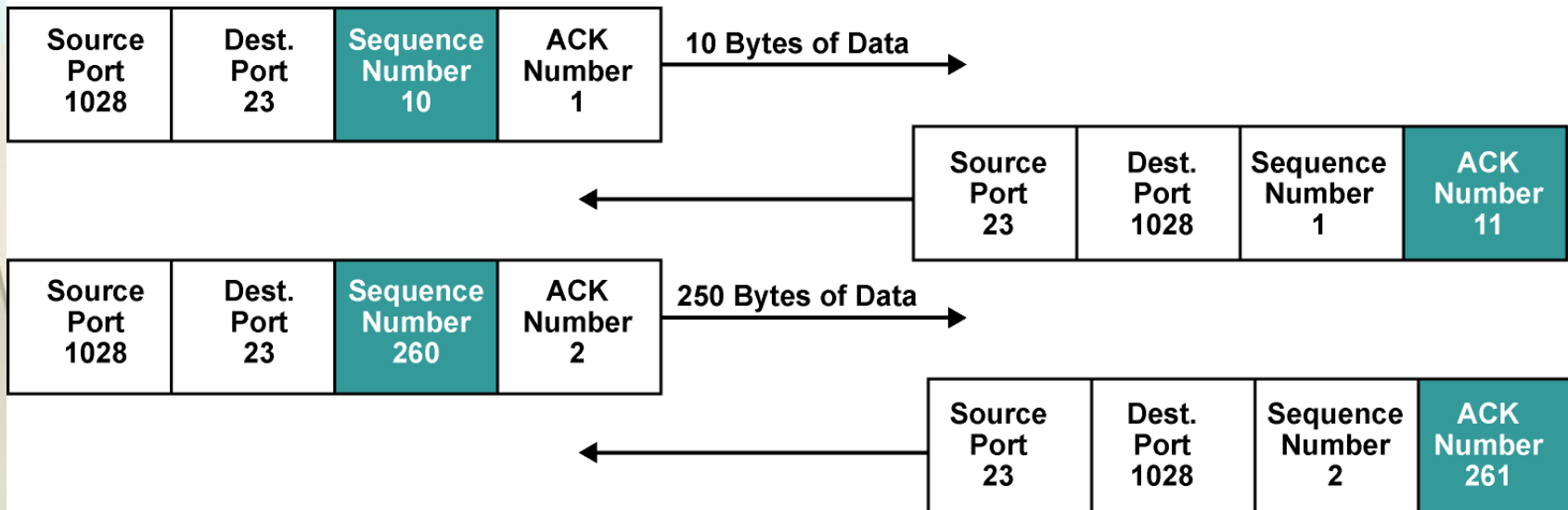
ACK 5  
Window size = 2

# Номера пакетов и уведомлений



Отправитель

Получатель



# Итоги по протоколу ТСР

- **Используя ТСР, как протокол транспортного уровня, приложения могут передавать данные. Соединение устанавливается однократно и далее поддерживается в процессе передачи данных.**
- **Контроль передачи позволяет избегать проблем с переполнением буфера получателя.**
- **ТСР поддерживает нумерацию пакетов и уведомления. Когда очередной пакет доставлен, получатель уведомляет об этом отправителя и начинается передача следующих пакетов.**
- **Размер ТСР-«окна» определяет число пакетов, которые можно передавать без подтверждения.**

# **Протокол UDP (User Datagram Protocol)**

- **Не использует соединения**
- **Пакеты (дейтаграммы) не связаны друг с другом**
- **Доставка пакетов (дейтаграмм) не проверяется и не гарантируется**
- **Утерянные пакеты (дейтаграммы) не восстанавливаются**

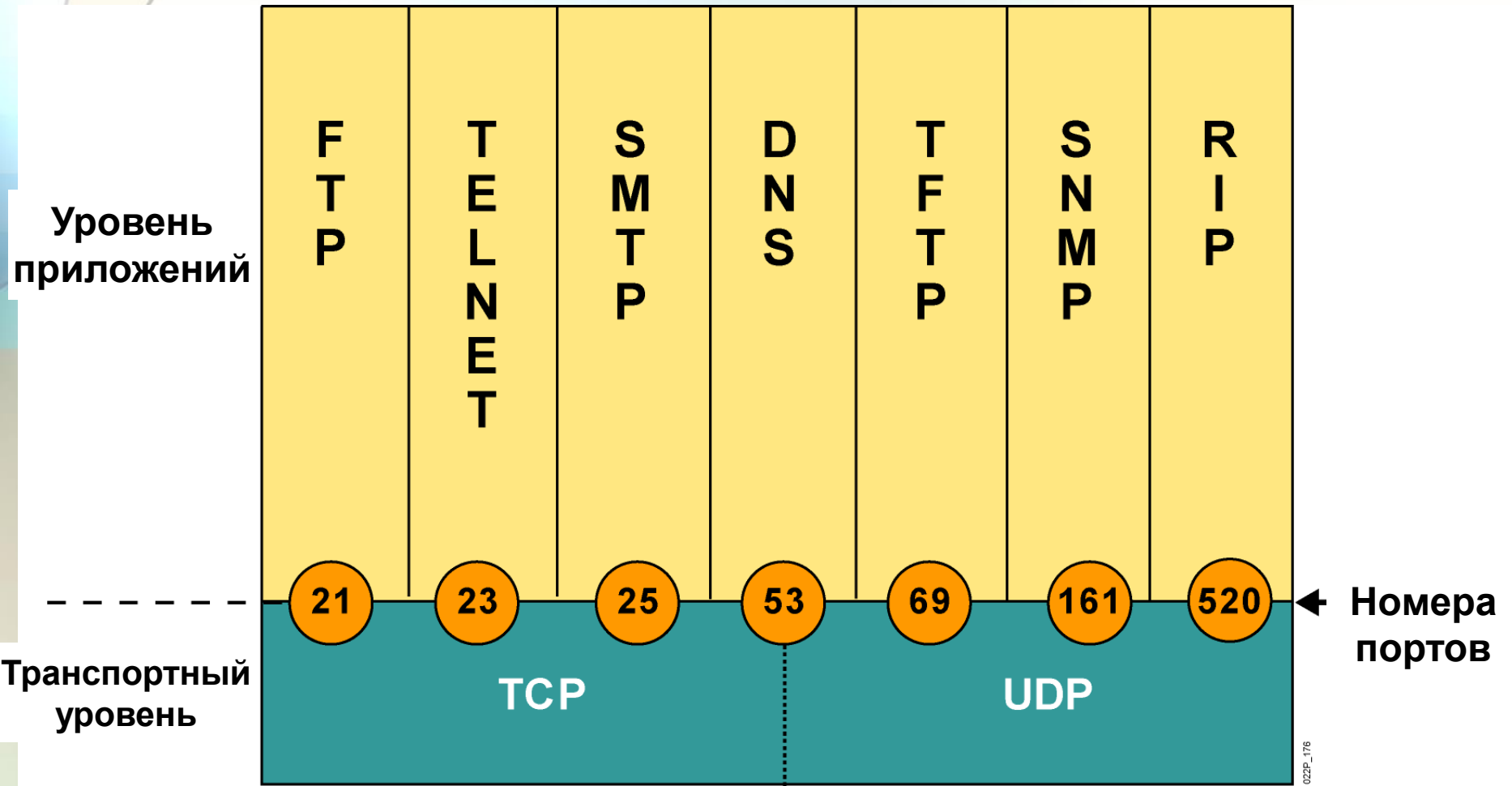
# Заголовок UDP-дейтаграммы

<b>16-bit source port</b>	<b>16-bit destination port</b>
<b>16-bit UDP length</b>	<b>16-bit UDP checksum</b>
<b>Data</b>	

022P\_177



# Протоколы работающие с TCP и UDP



# Маршрутизация

- **Маршрутизаторы**
- **Маршрутизаторы и процесс доставки ip-пакетов**
- **Выбор пути**
- **Таблицы маршрутизации**
- **Статические, динамические маршруты**
- **Протоколы динамической маршрутизации**

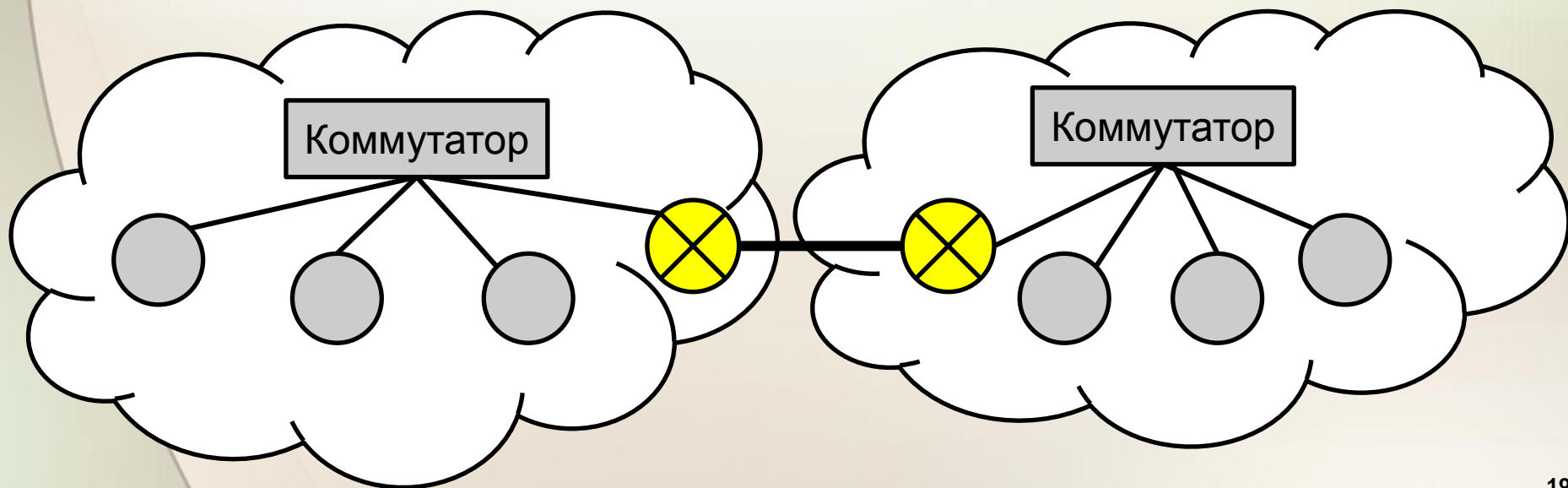
# Маршрутизаторы



Linksys



Cisco 2610 Router



# Задачи маршрутизатора

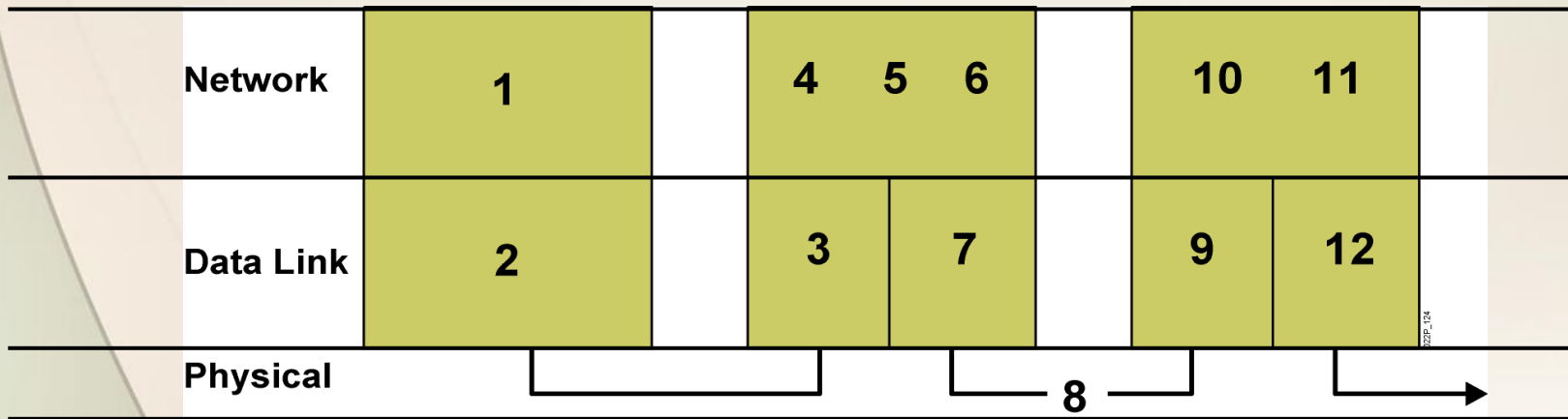
1. Обладать информацией о достижимых сетях
2. Выбирать куда отправить пакет далее

```
router# show ip route
```

```
1 { D 192.168.1.0/24 [90/25789217] via 10.1.1.1  
R 192.168.2.0/24 [120/4] via 10.1.1.2  
O 192.168.3.0/14 [110/229840] via 10.1.1.3 } 2
```

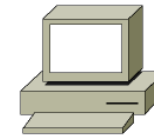
# Доставка IP-пакетов

1. Хост-отправитель собирается отправить IP-пакет на хост-получатель.
2. IP-пакет инкапсулируется в кадр Ethernet и отправляется на маршрутизатор
3. Маршрутизатор принимает кадр, извлекает из него IP-пакет.
4. Маршрутизатор находит сеть хоста-получателя в таблице маршрутизации.
5. Маршрутизатор выбирает через какой ближайший маршрутизатор достижима сеть хоста-получателя
6. Определяется MAC-адрес маршрутизатора
7. Шаги 2 – 6 повторяются сколько нужно для достижения сети хоста получателя.
9. Маршрутизатор сети хоста-получателя принимает кадр.
10. Маршрутизатор видит что нужная сеть достижима непосредственно
11. С помощью ARP выявляется MAC адрес хоста получателя
12. IP-пакет инкапсулируется в кадр Ethernet и направляется к хосту получателю.

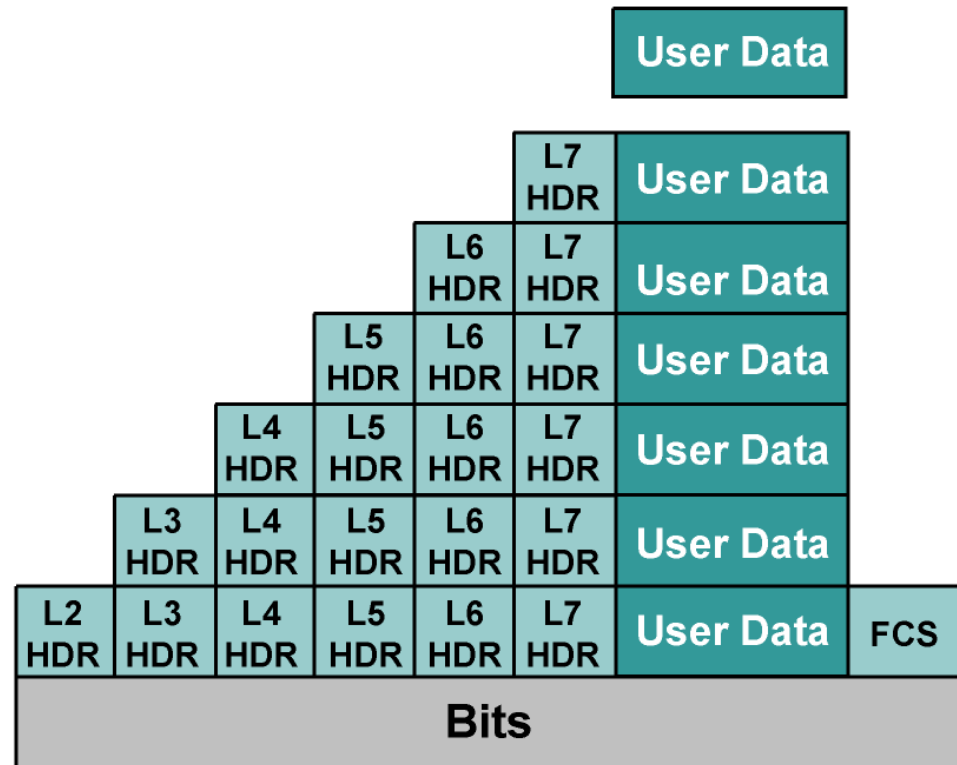
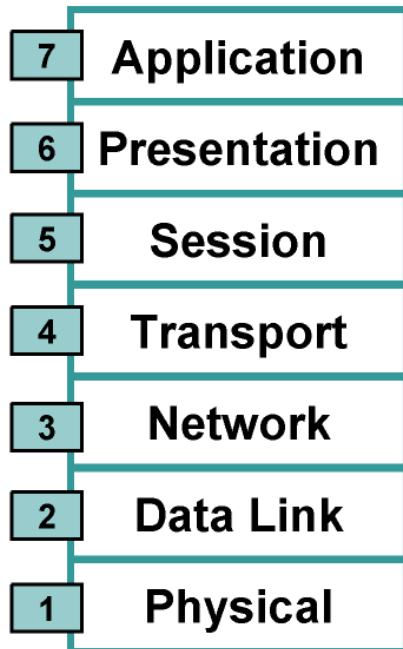


# Инкапсуляция данных при отправке

Sender



User Data

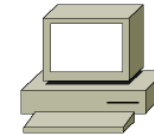


HDR = header

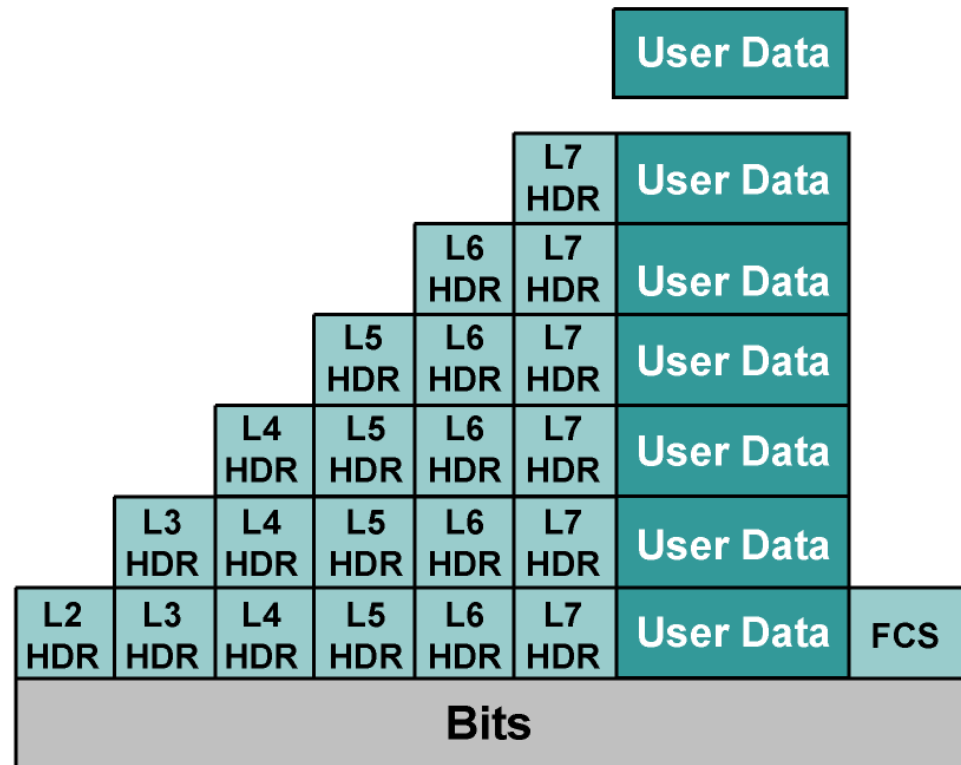
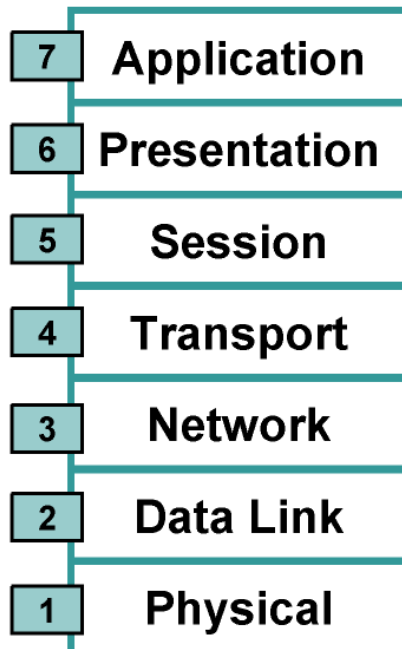
022P\_063

# Извлечение данных при получении

Receiver



User Data



HDR = header

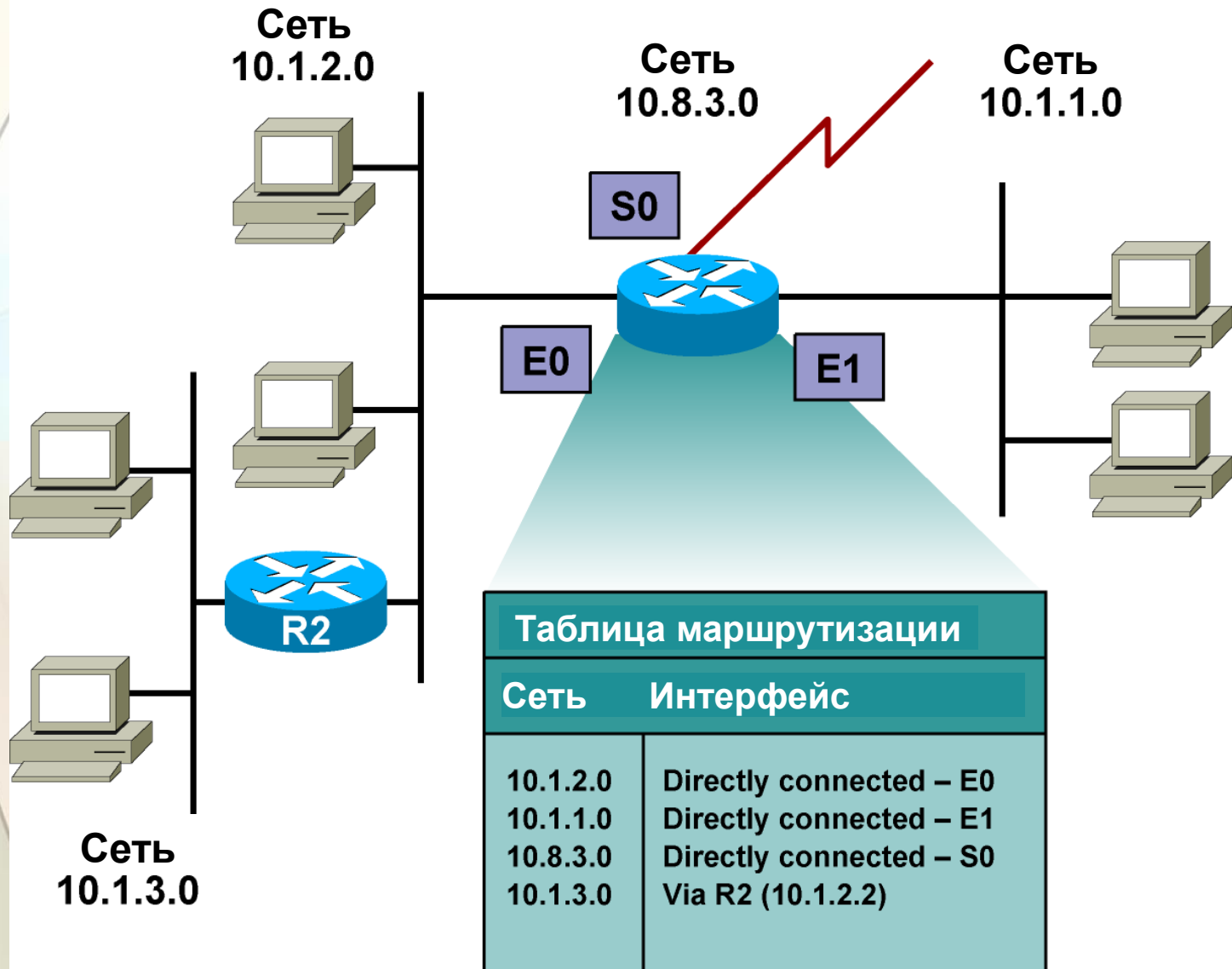
022P\_064

# Основная задача маршрутизатора – выбор пути следования

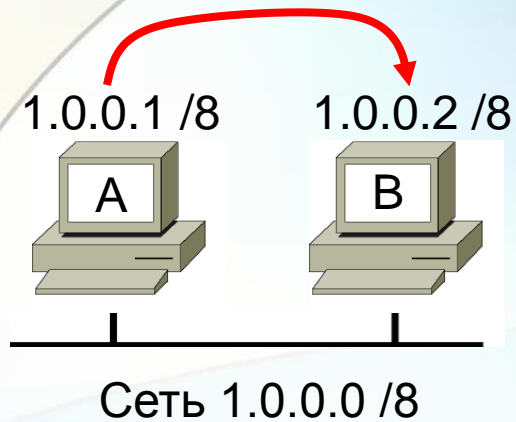




# Таблицы маршрутизации



# Таблицы маршрутизации



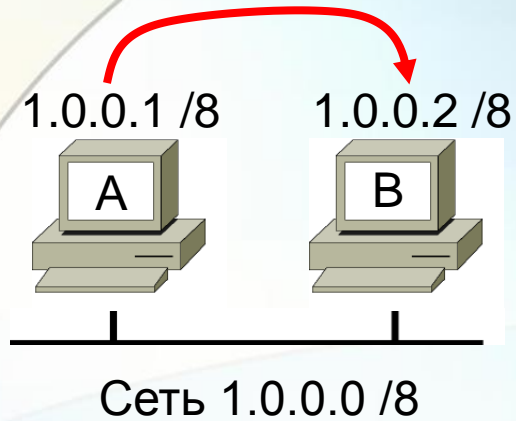
Отправляем IP-датаграмму с 1.0.0.1 на 1.0.0.2  
Модуль протокола IP должен выбрать сетевой интерфейс отправки (на основе «таблицы маршрутизации»)

Адрес сети получателя	Маска	Интерфейс	Шлюз	Метрика
1.0.0.0	255.0.0.0	1	-	1

## 1 этап

1. Берём очередной маршрут
2. Побитно умножаем IP-адрес получателя на «маску» маршрута
3. Находим «сеть получателя»
4. Сравниваем «сеть получателя» и «адрес сети получателя»
5. Если равны – отправляем датаграмму через «интерфейс» на «шлюз»
6. Если не равны – возврат на п.1

# Таблицы маршрутизации



Отправляем IP-датаграмму с 1.0.0.1 на 1.0.0.2  
Модуль протокола IP должен выбрать сетевой интерфейс отправки (на основе «таблицы маршрутизации»)

Адрес сети получателя	Маска	Интерфейс	Шлюз	Метрика
1.0.0.0	255.0.0.0	1	-	1

Если в таблице маршрут найден, то:

- Программный модуль сетевого уровня (IP) готовит пакет для передачи в модуль протокола канального уровня (Ethernet)
- Для поиска Ethernet-адреса получателя используется ARP
- Приготовленный кадр отправляется в сеть через интерфейс №1



# Таблицы маршрутизации

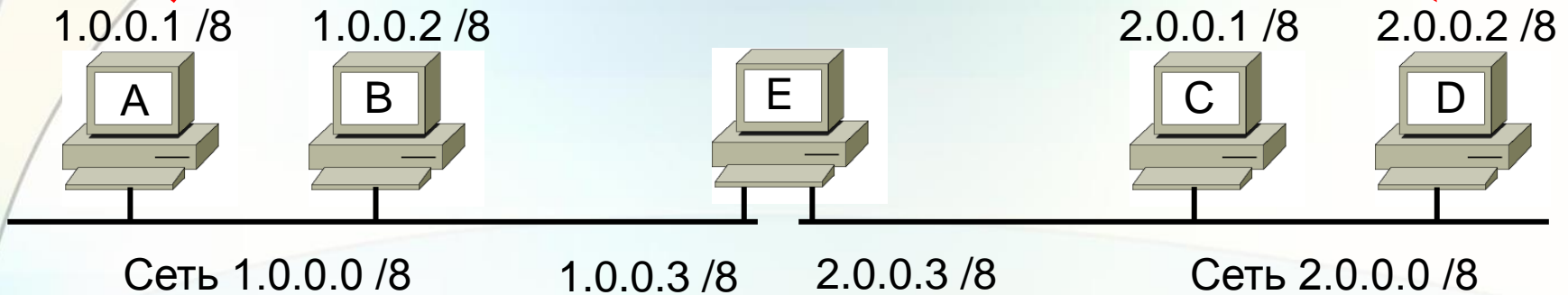


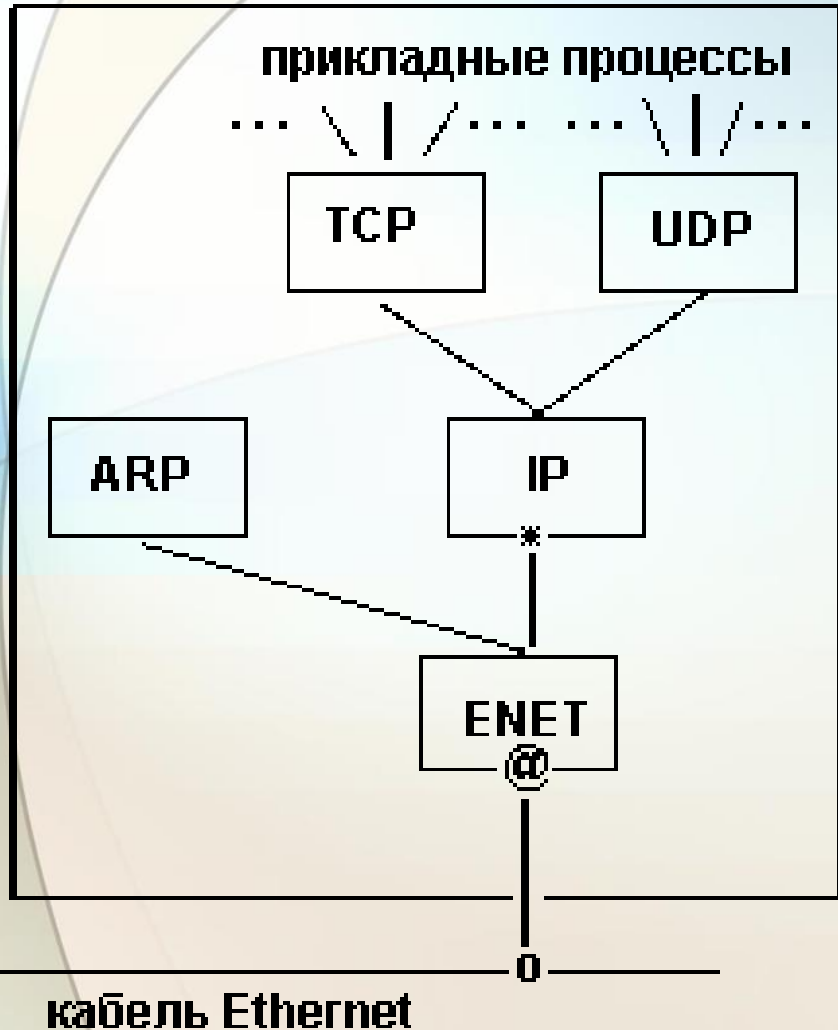
Таблица маршрутизации узла A

Адрес сети получателя	Маска	Интерфейс	Шлюз	Метрика
1.0.0.0	255.0.0.0	1	-	1
2.0.0.0	255.0.0.0	1	1.0.0.3	1

## 1 этап

1. Берём очередной маршрут
2. Побитно умножаем IP-адрес получателя на «маску» маршрута
3. Находим «сеть получателя»
4. Сравниваем «сеть получателя» и «адрес сети получателя»
5. Если равны – отправляем датаграмму через «интерфейс» на «шлюз»
6. Если не равны – возврат на п.1

# Стек протоколов



Готовится IP-датаграмма  
(датаграмма идёт от А к В)

Датаграмма упаковывается в  
кадр Ethernet

Если получатель в нашей сети,  
то MAC-адрес получателя =

- MAC-адрес получателя

Если нет:

- MAC-адрес шлюза

# Таблицы маршрутизации

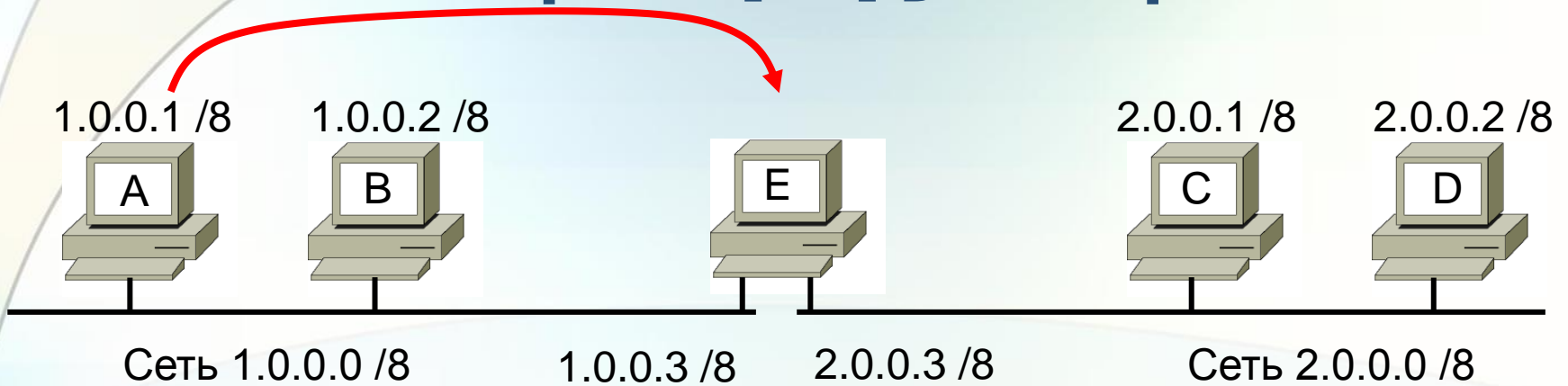


Таблица маршрутизации узла В

Адрес сети получателя	Маска	Интерфейс	Шлюз	Метрика
1.0.0.0	255.0.0.0	1	-	1
2.0.0.0	255.0.0.0	2	-	1

Так как компьютер обладает 2-мя сетевыми интерфейсам, в его таблице – два «автопрописанных» маршрута к непосредственно подключенным сетям

# Таблицы маршрутизации

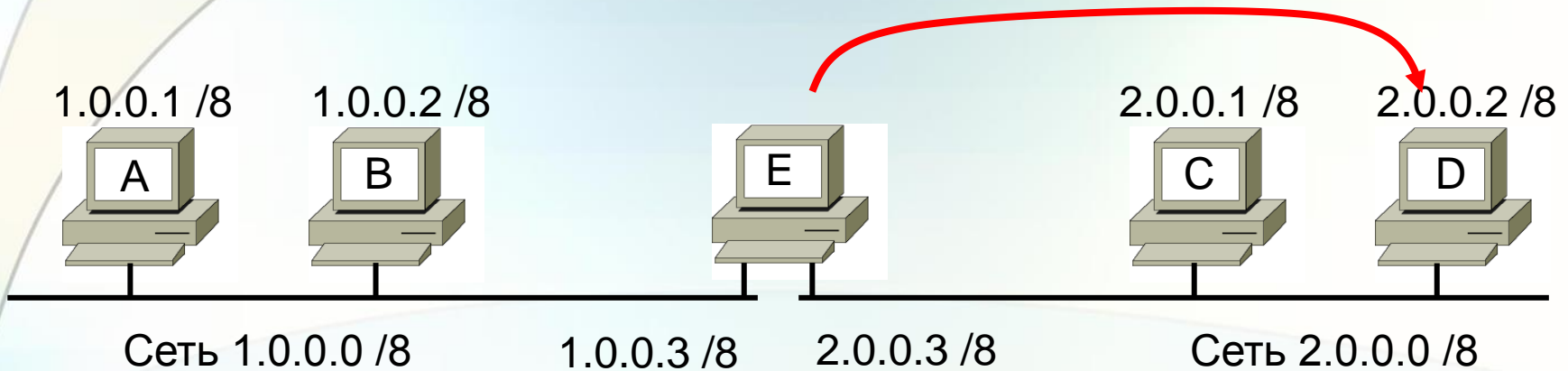


Таблица маршрутизации узла В

Адрес сети получателя	Маска	Интерфейс	Шлюз	Метрика
1.0.0.0	255.0.0.0	1	-	1
2.0.0.0	255.0.0.0	2	-	1

## 1 этап

1. Берём очередной маршрут
2. Побитно умножаем IP-адрес получателя на «маску» маршрута
3. Находим «сеть получателя»
4. Сравниваем «сеть получателя» и «адрес сети получателя»
5. Если равны – отправляем датаграмму через «интерфейс» на «шлюз»
6. Если не равны – возврат на п.1