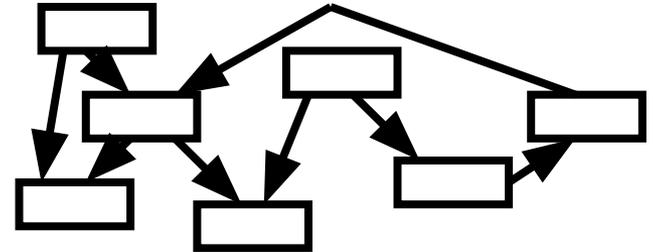


# **Иерархическая организация СВЯЗИ**

# Решение сложных задач

## Декомпозиция:

Разбиение на более простые подзадачи  
Четкое определение функций  
Входные и выходные интерфейсы  
Координация



- Иерархический подход  
строгая последовательность  
связь только со смежным уровнем)



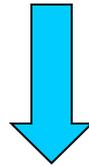
# Специфика систем связи:

- Взаимодействие 2х и более абонентов
- Территориальная распределенность
- Разнотипное оборудование
- Наличие промежуточных устройств

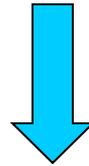
Необходимо большое количество согласований □ (стандартизация)

# Многоуровневый подход

Потребность объединения разнотипных ЭВМ



Разработка идеологической концепции  
(Многоуровневый подход к стандартизации)



Универсальные правила взаимодействия ЭВМ  
(Принцип открытых систем)

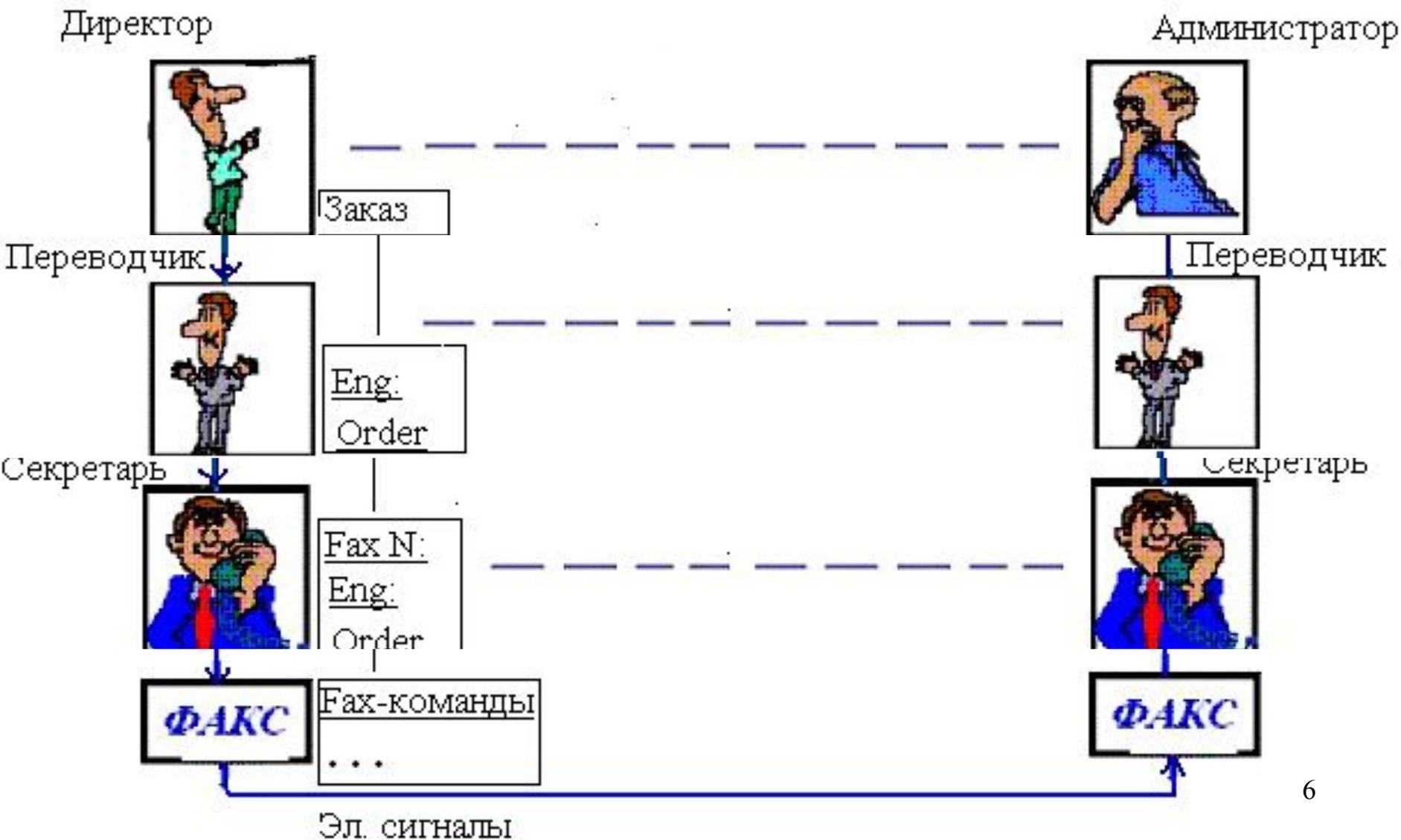
Эталонная модель ВОО/МОС, ЭМВОС, ISO/OSI

# Открытые системы

Системы, построенные в соответствии с открытыми общепринятыми спецификациями

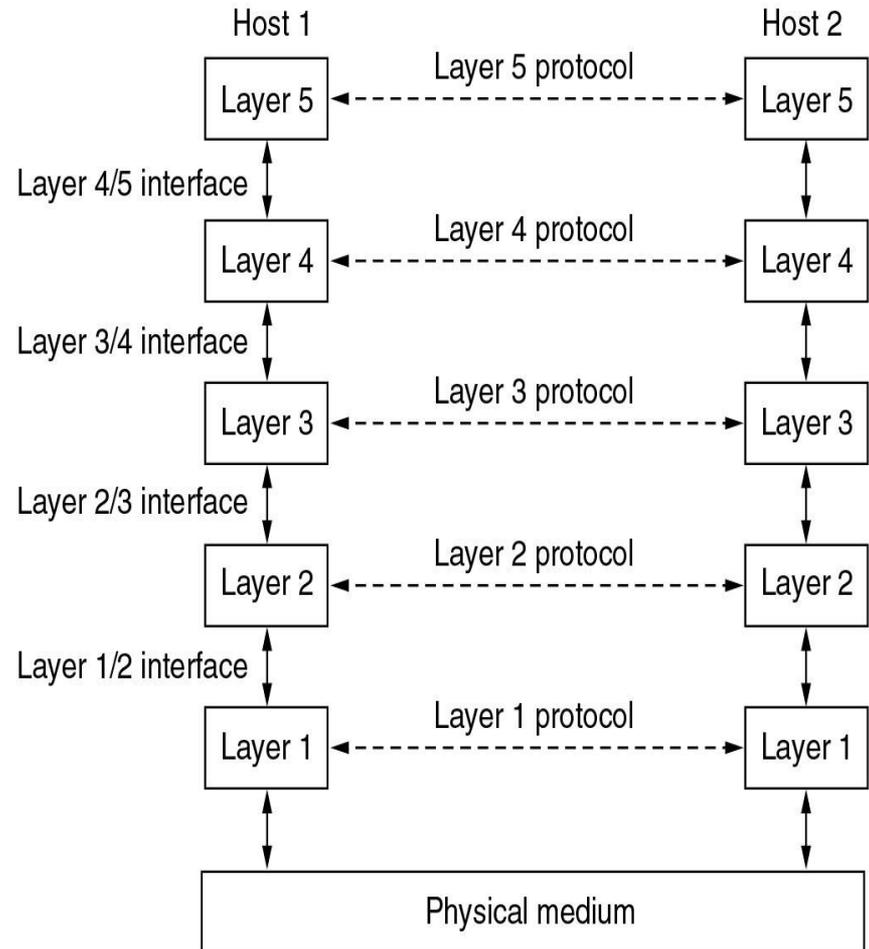
- Совместимость
- Модернизация
- Простота сопряжения с другими системами
- Простота освоения и обслуживания
- Высокий темп совершенствования

# Бронирование отеля в Бразилии

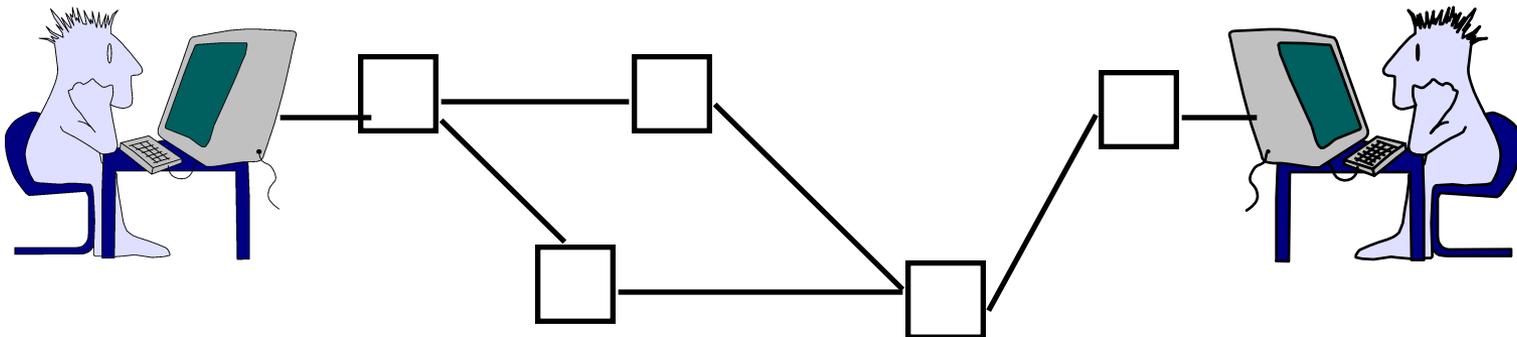
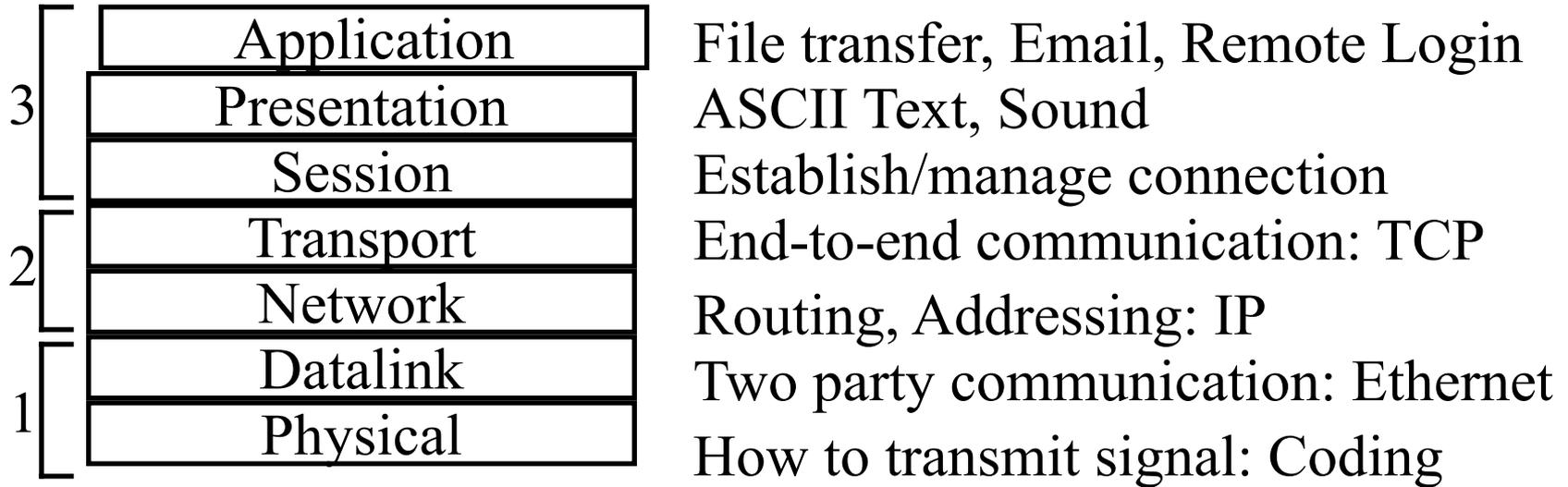


# Уровни и протоколы

- **Функциональные уровни**
- **Правила взаимодействия уровней - протоколы**
- **Правила взаимодействия смежных уровней - (межуровневый) интерфейс**



# ISO/OSI Reference Model





# Физический уровень

- Во всех устройствах сети
- Физическая среда, дальность, скорость, кодирование, уровни сигналов, разъемы
- Аппаратная реализация (сетевые адаптеры)

10 BaseT - UTP кат 3, 100 ом, 100м,  
10Мбит/с, Манчестерский код, RJ-45

# Канальный уровень

- Управление доступом к среде (МАС)
- Обработка ошибок (Кадры, контрольная сумма, повторная передача)
- Адресация
- Управление потоком

Хосты, мосты, коммутаторы, маршрутизаторы

Хост : Сетевые карты + драйверы

LAN: Ethernet, Token Ring, FDDI (типовые топологии)

WAN: “точка-точка”, PPP, LAP-B

# Сетевой уровень

“Сеть” - типовая топология, стандартный канальный протокол, один администратор

Внутри сети – канальный уровень

Между сетями – сетевой уровень

- Маршрутизация,
- Согласование технологий
- Адресация (“сеть” – хосты с единым номером сети)
- Программный модуль ОС, ПО маршрутизатора  
IP, IPX

# Транспортный уровень

- Протоколы END-to-END
- Качество передачи  
(достоверность, очередность)
- Управление передачей
- Мультиплексирование
- ПО хоста

TCP, UDP, SPX

# Сеансовый уровень

- Управление диалогом
- Установление/разъединение соединения
- Синхронизация (точки отката)
- ПО хоста

Обычно совмещается с транспортным или прикладным уровнем

# Представительский уровень

- Унификация формы представления информации
- Шифрация

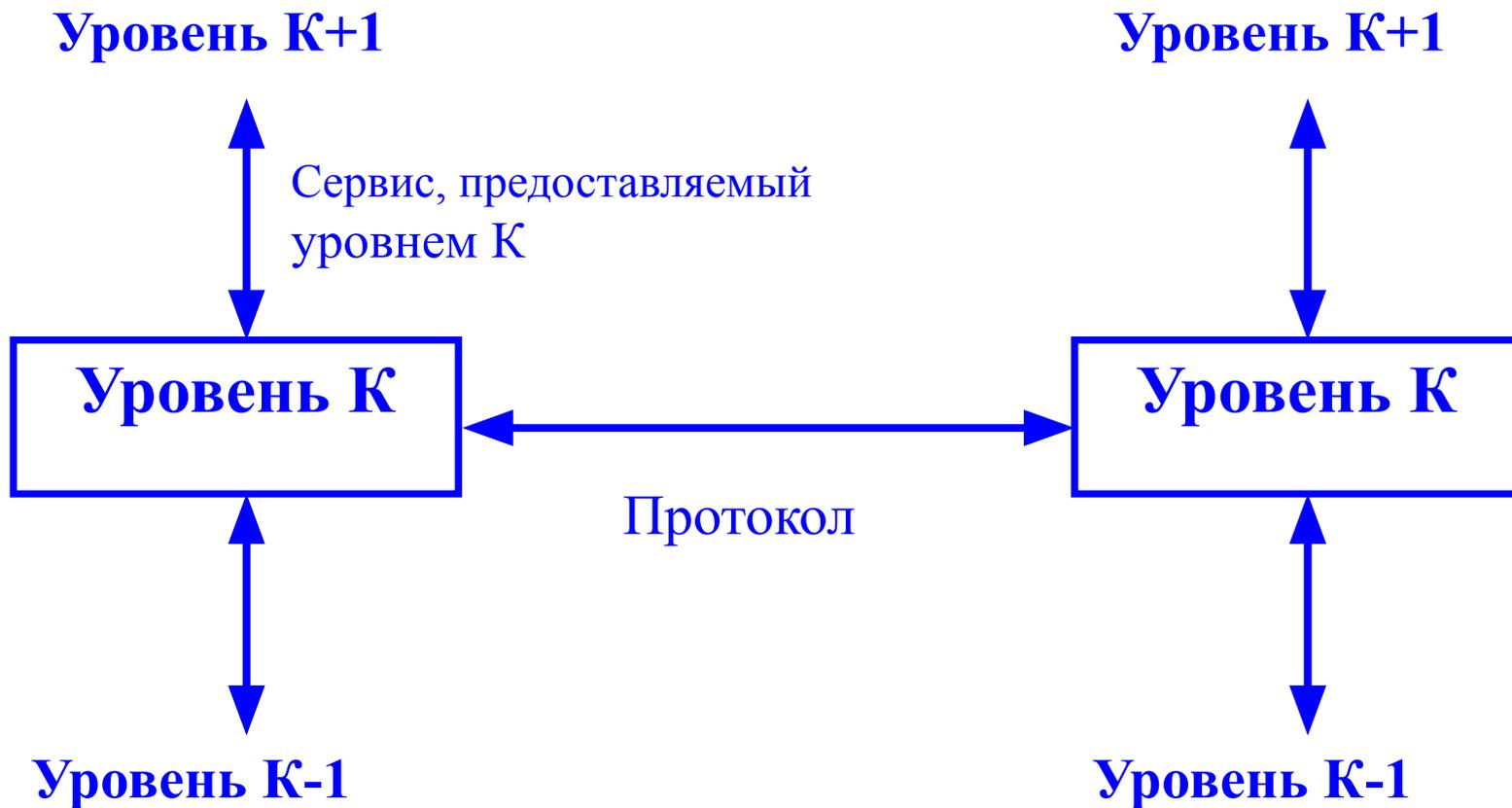
SSL

Обычно совмещается с прикладным уровнем

# Прикладной уровень

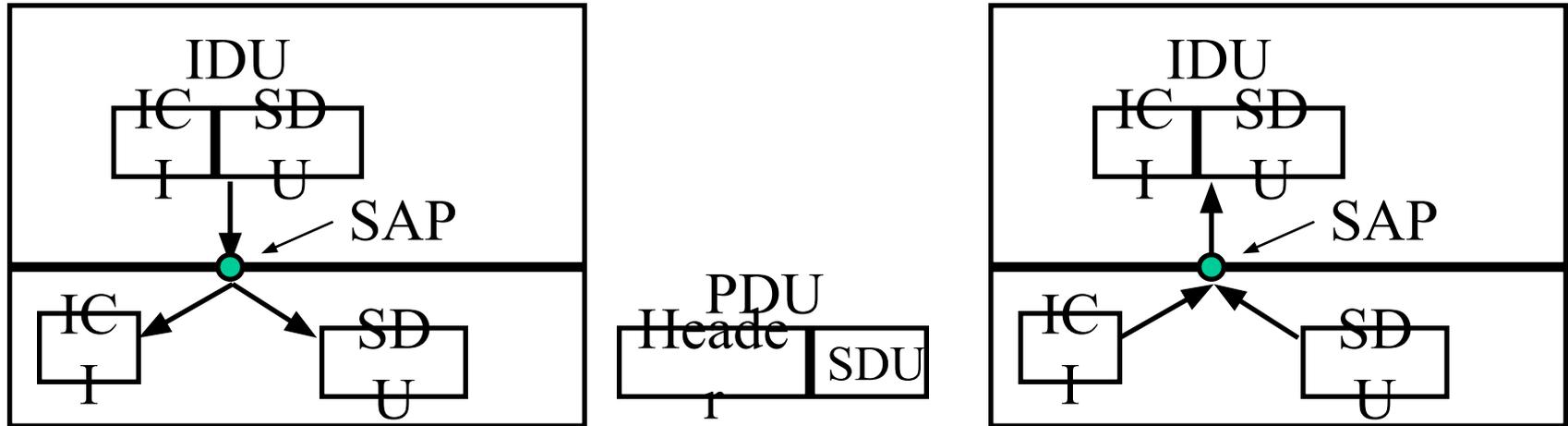
- Доступ к распределенным ресурсам
- Большое число протоколов
- Модель “клиент-сервер”
- HTTP, FTP, NFS, SMTP, Telnet...
- Служебные протоколы DNS, LDAP

# Взаимодействие уровней



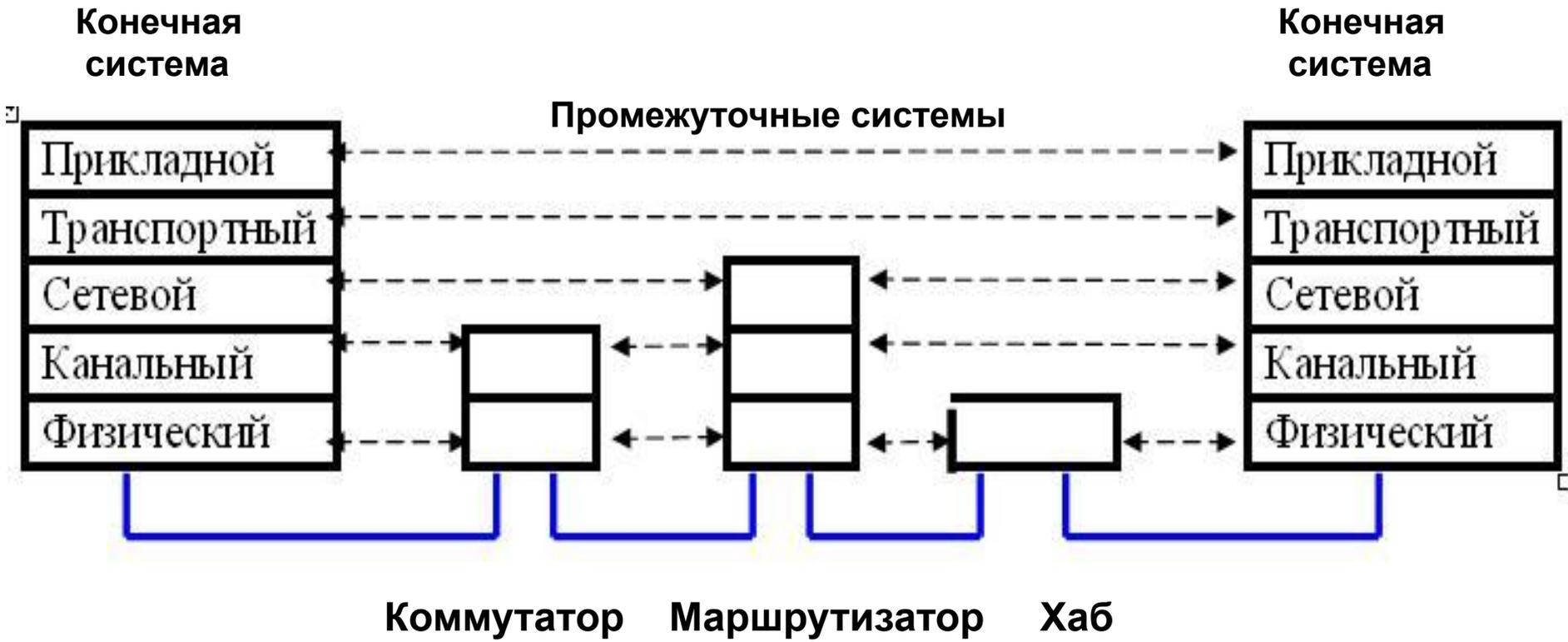
# Модель ISO/OSI

## Интерфейсы и сервисы

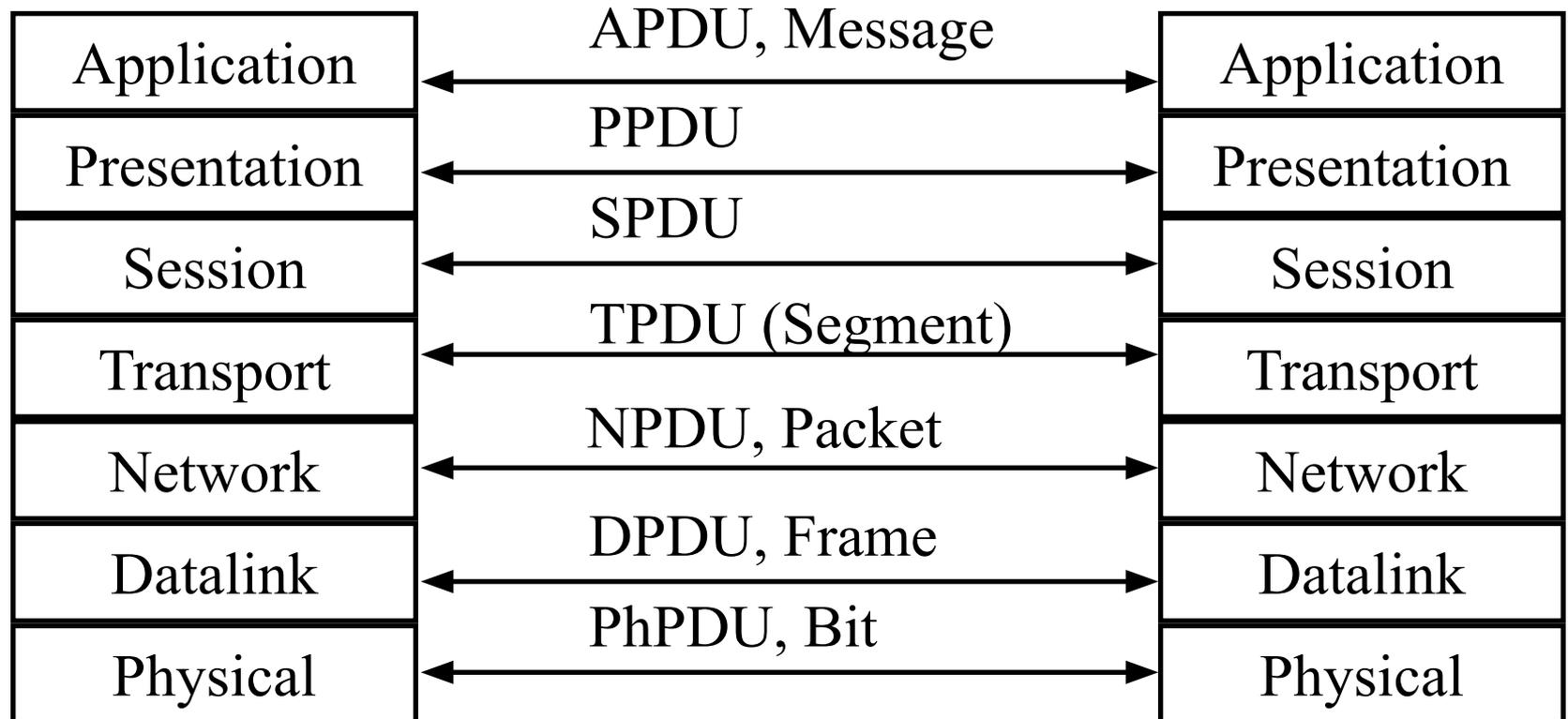


- **IDU - Interface Data Unit = ICI + SDU**
- **ICI - Interface Control Information**
- **SDU - Service Data Unit**
- **PDU - Protocol Data Unit = Fragments of SDU + Header or Several SDUs + Header (blocking)**
- **SAP = Service Access Point**

# Модель взаимодействия двух узлов ( Промежуточные системы )



# Protocol Data Unit (PDU)



# Эталонная модель ISO/OSI

- Единый перечень понятий
- Единый способ расщепления функций
- Единые правила – совместимость
- Независимость модулей – модификация
- Сетезависимые и сетезависимые уровни
- Протоколы транспортного уровня и выше работают на конечных узлах
- Протоколы сетевого уровня определяют тип сети (например, IP-сети) и работают на всех узлах сети

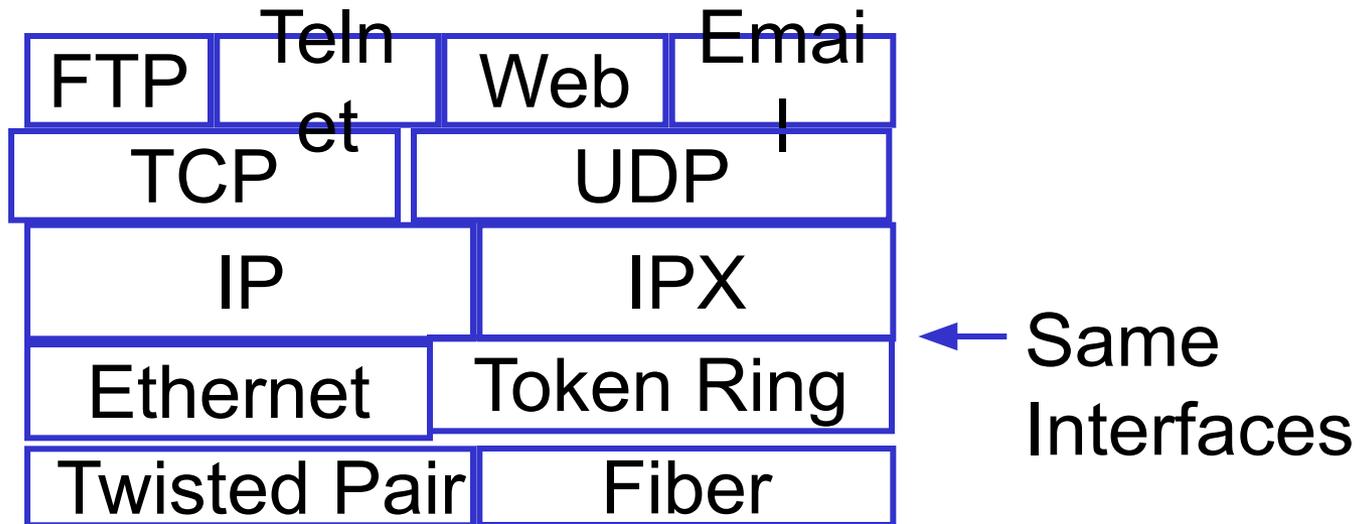
# Примитивы служб

- 8 примитивов
- CONNECT.request – запрос на установку соединения
- CONNECT.indication – сигнал вызываемой стороне
- CONNECT.response – вызываемая сторона принимает/откл. запрос
- CONNECT.confirm – вызывающей стороне сообщает, что вызов принят
- DATA.request – запрос на пересылку данных
- DATA.indication – сигнал о прибытии данных
- DISCONNECT.request – запрос об освобождении соединения
- DISCONNECT.indication - сигнал вызываемой стороне

# Примитивы служб. Пример

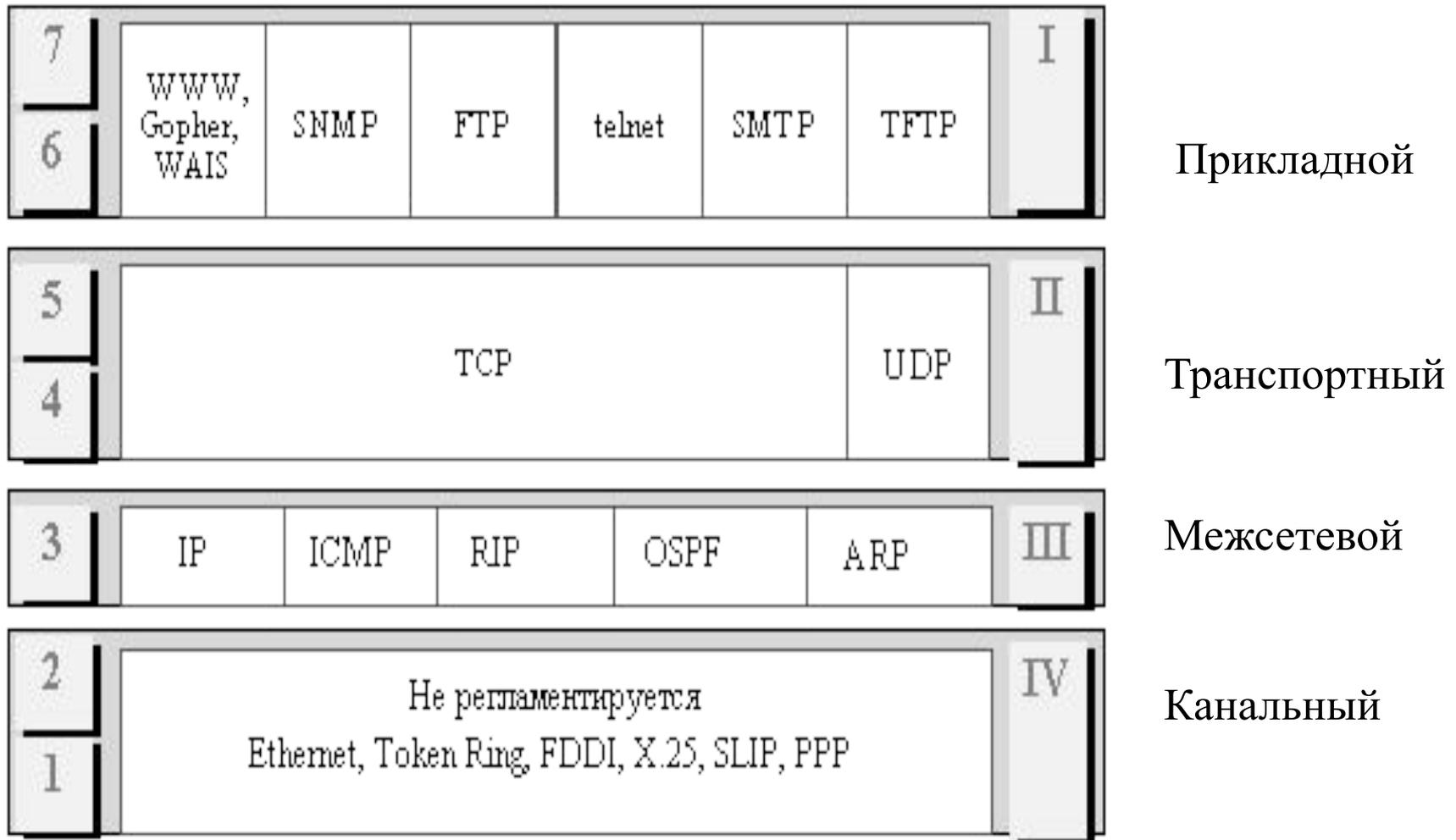
- 8 примитивов
- CONNECT.request – набрать номер т. Маши
- CONNECT.indication – ее телефон звонит
- CONNECT.response – она поднимает трубку
- CONNECT.confirm – Вы слышите, что гудки прекратились
- DATA.request – Вы приглашаете на чай
- DATA.indication – она слышит ваше приглашение
- DATA.request – она говорит, что будет рада прийти
- DATA.indication – Вы слышите ее ответ
- DISCONNC.request – Вы кладете трубку
- DISCONNECT.indication - она слышит гудки и тоже кладет трубку

# Layering



- Protocols@each layer perform a set of functions
- All alternatives for a row have the same interfaces
- Choice of protocols at each layer is independent of those of at other layers.
- May not be the most efficient implementation<sup>24</sup>

# Модель протоколов TCP/IP

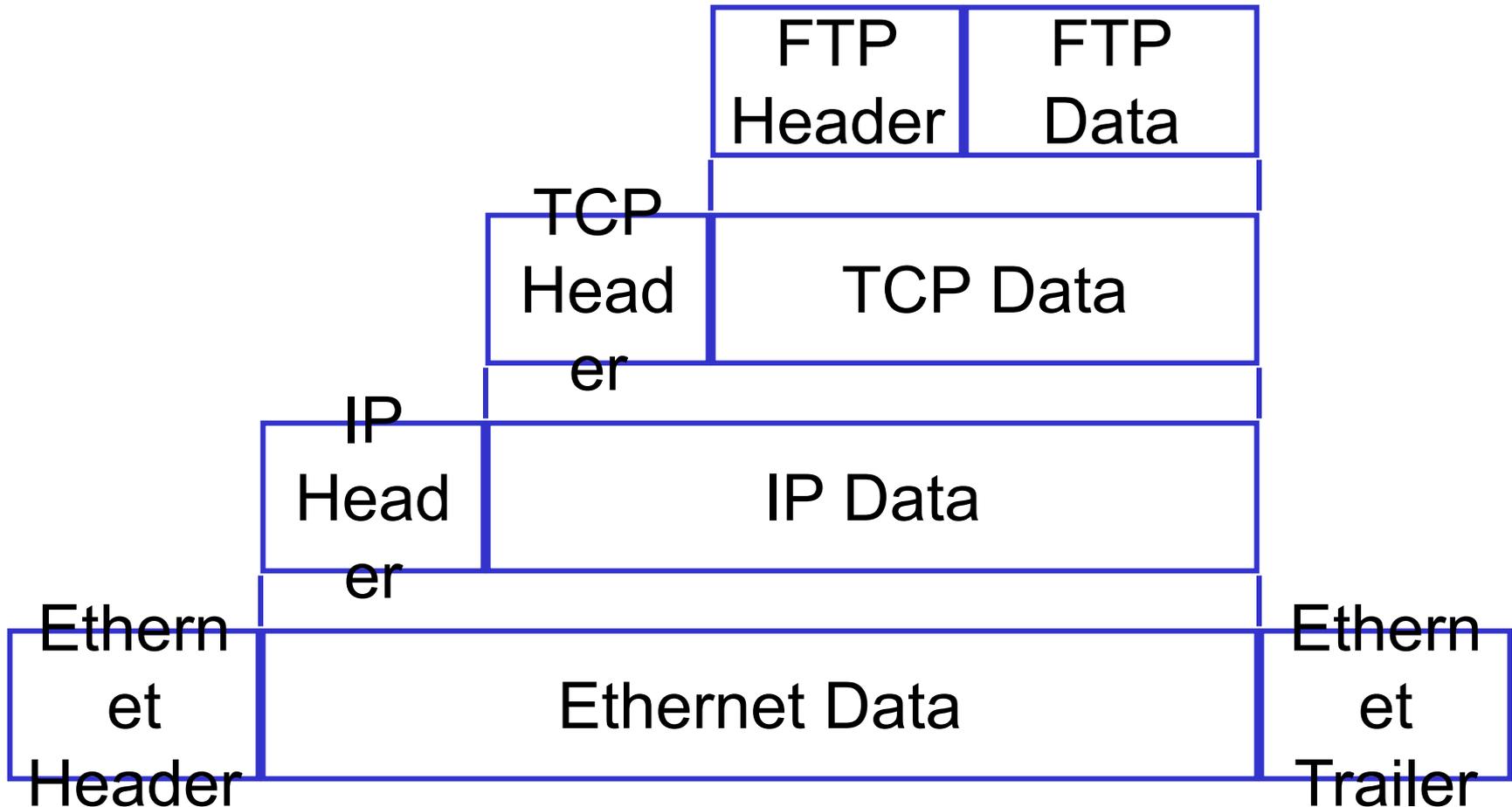


Уровни  
модели  
OSI

Уровни  
стека  
TCP/IP

# Инкапсуляция

Служебная информация N-го уровня пересылается как “данные” N-1го уровня.



# Сравнение моделей

## ISO/OSI

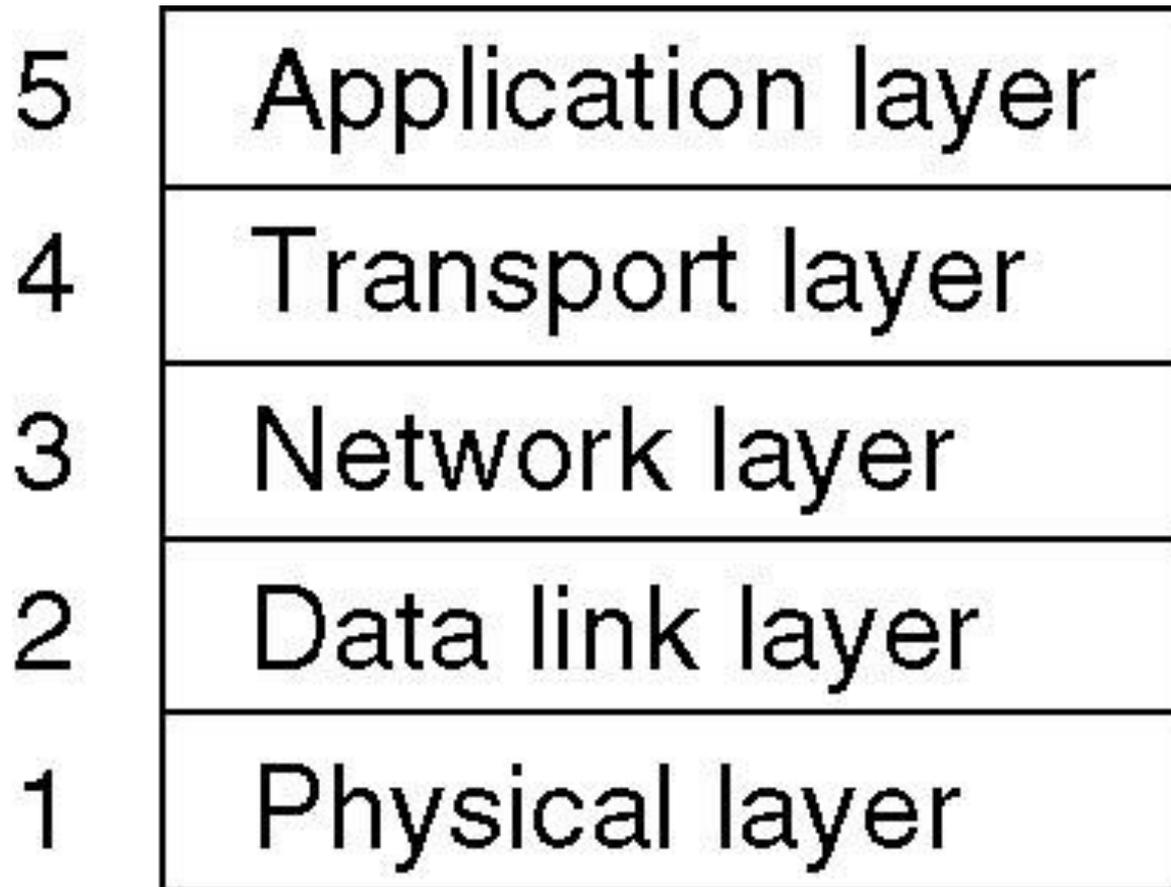
- Модель → протоколы (универсальность)
- Интерфейсы, протоколы службы - явно разделены
- Модель полезна, протоколы ?
- Сложная модель
- Менталитет связи

## TCP/IP

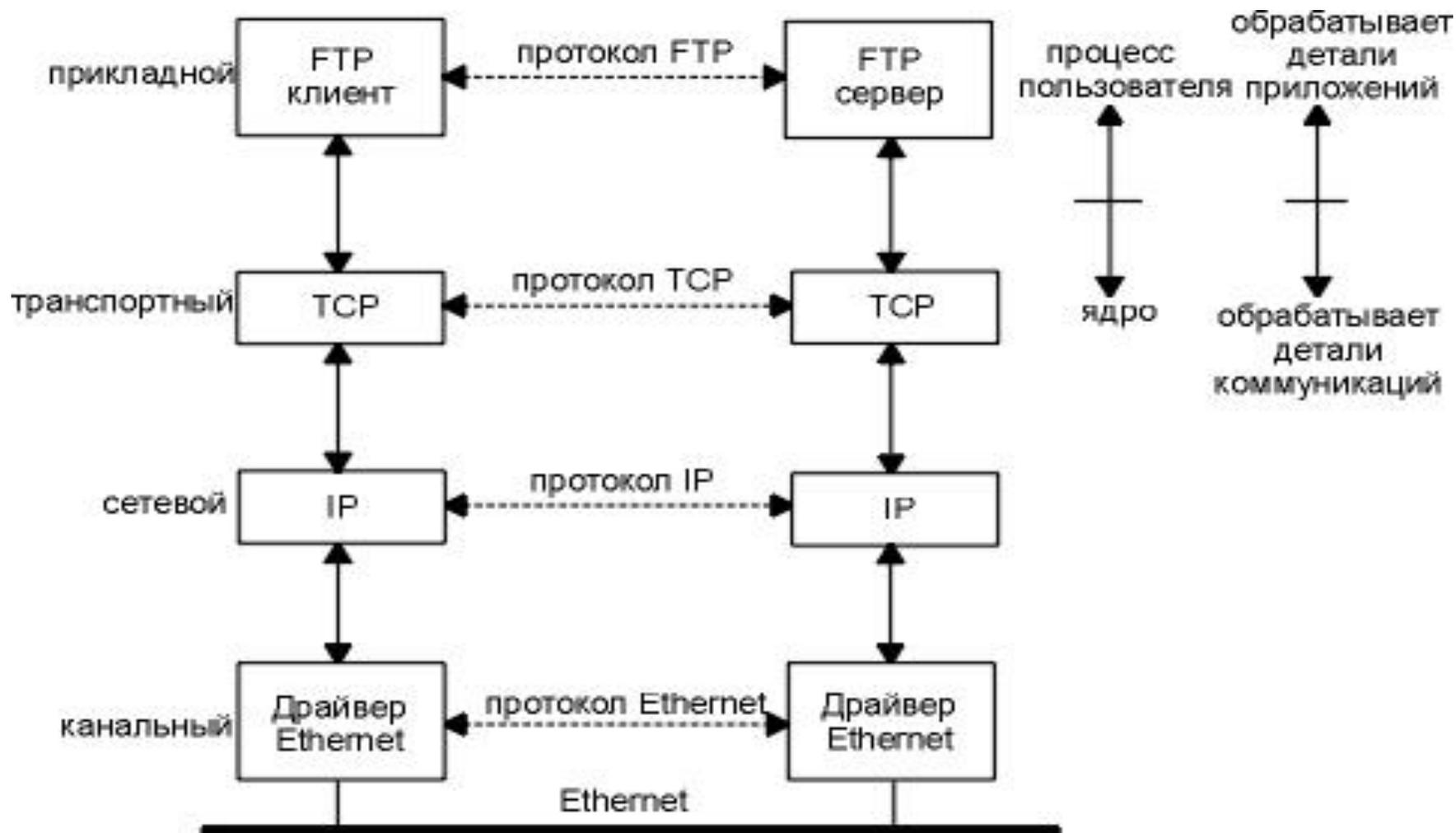
- Протоколы → модель (эффективность)
- Понятия смешаны
- Протоколы полезны, модель ?
- Простая модель
- Менталитет ВТ

Безопасность, управление сетями не представлены в моделях

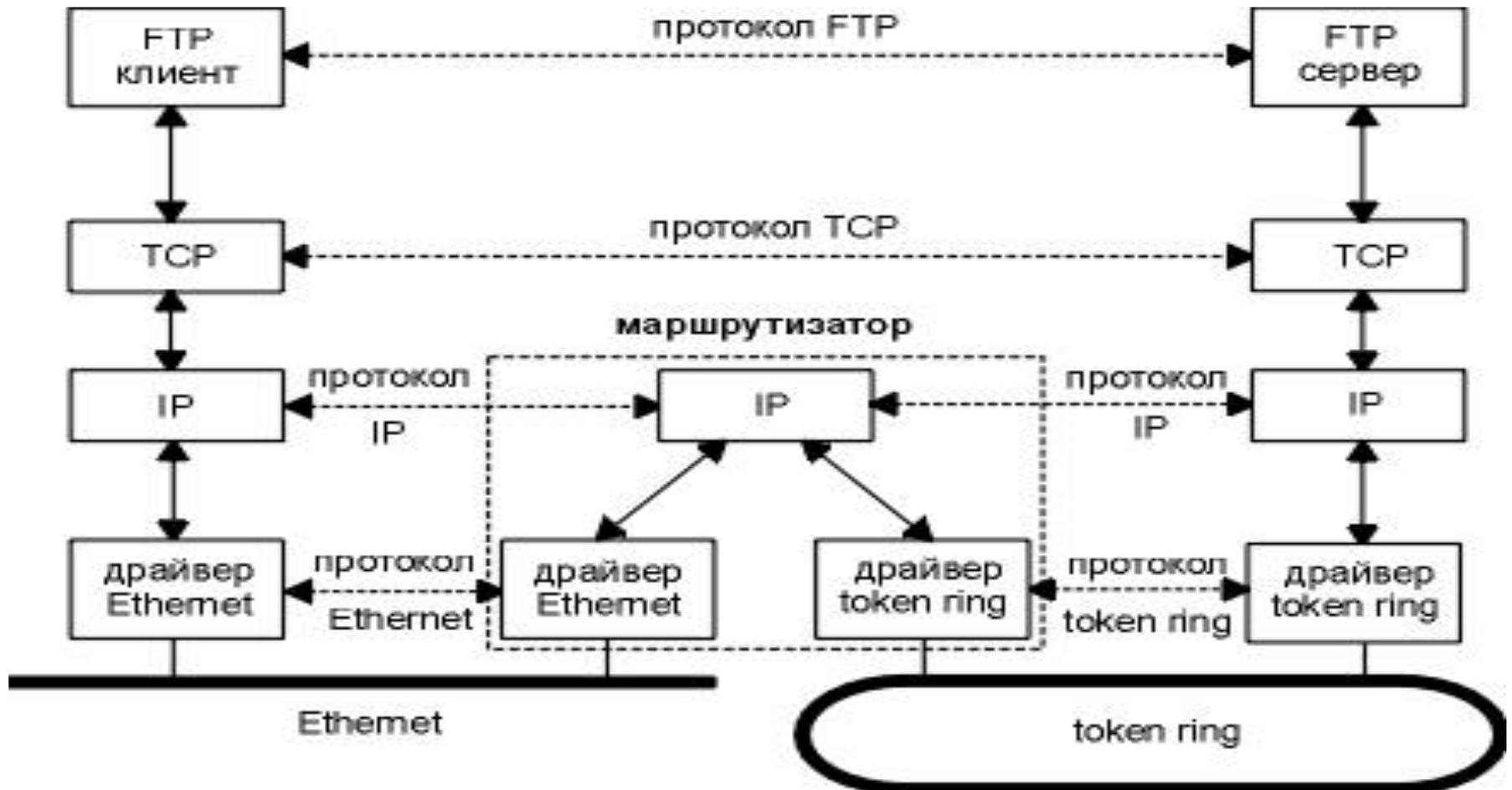
# Гибридная модель



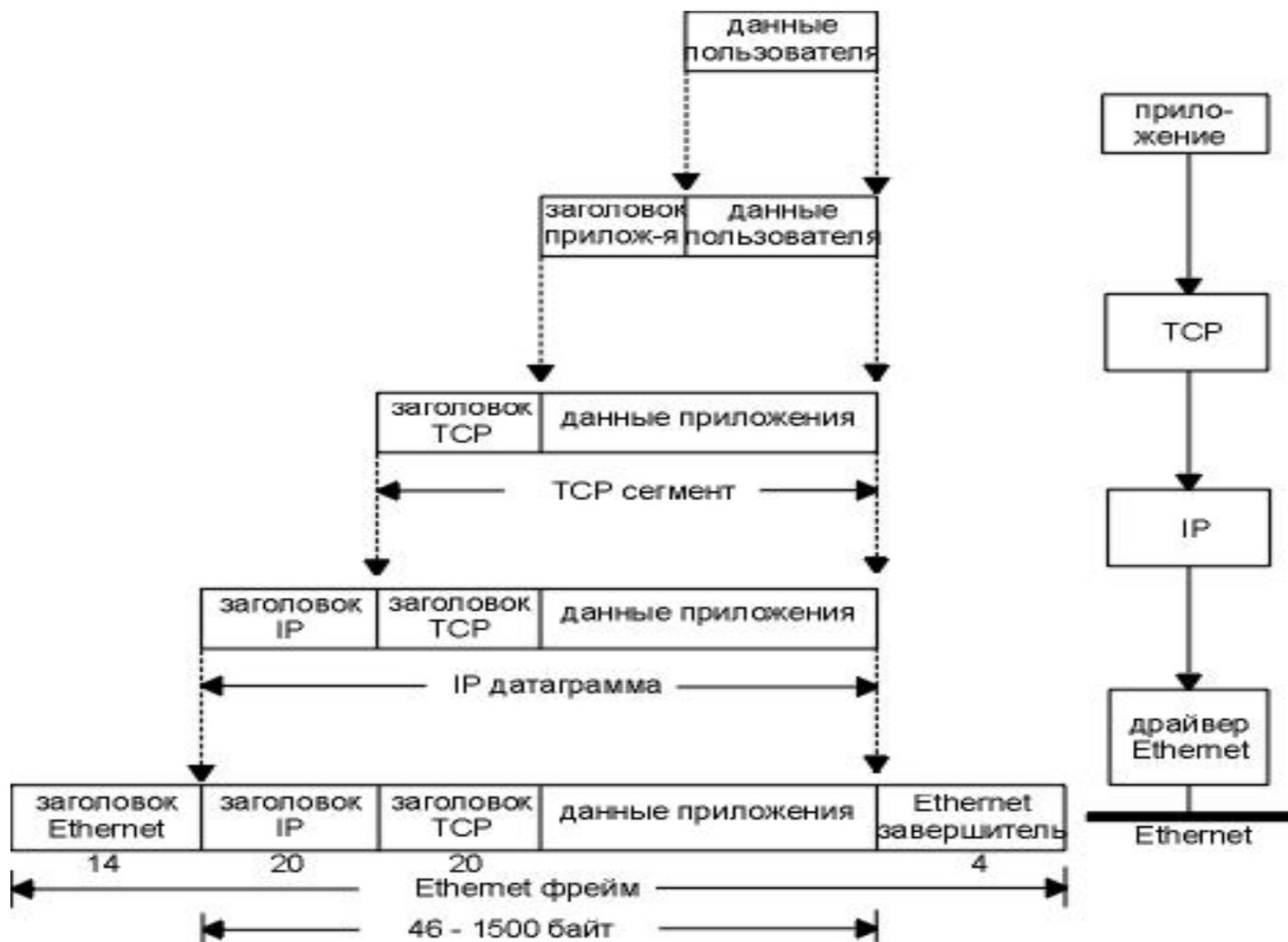
# Взаимодействия двух компьютеров в одной ЛВС



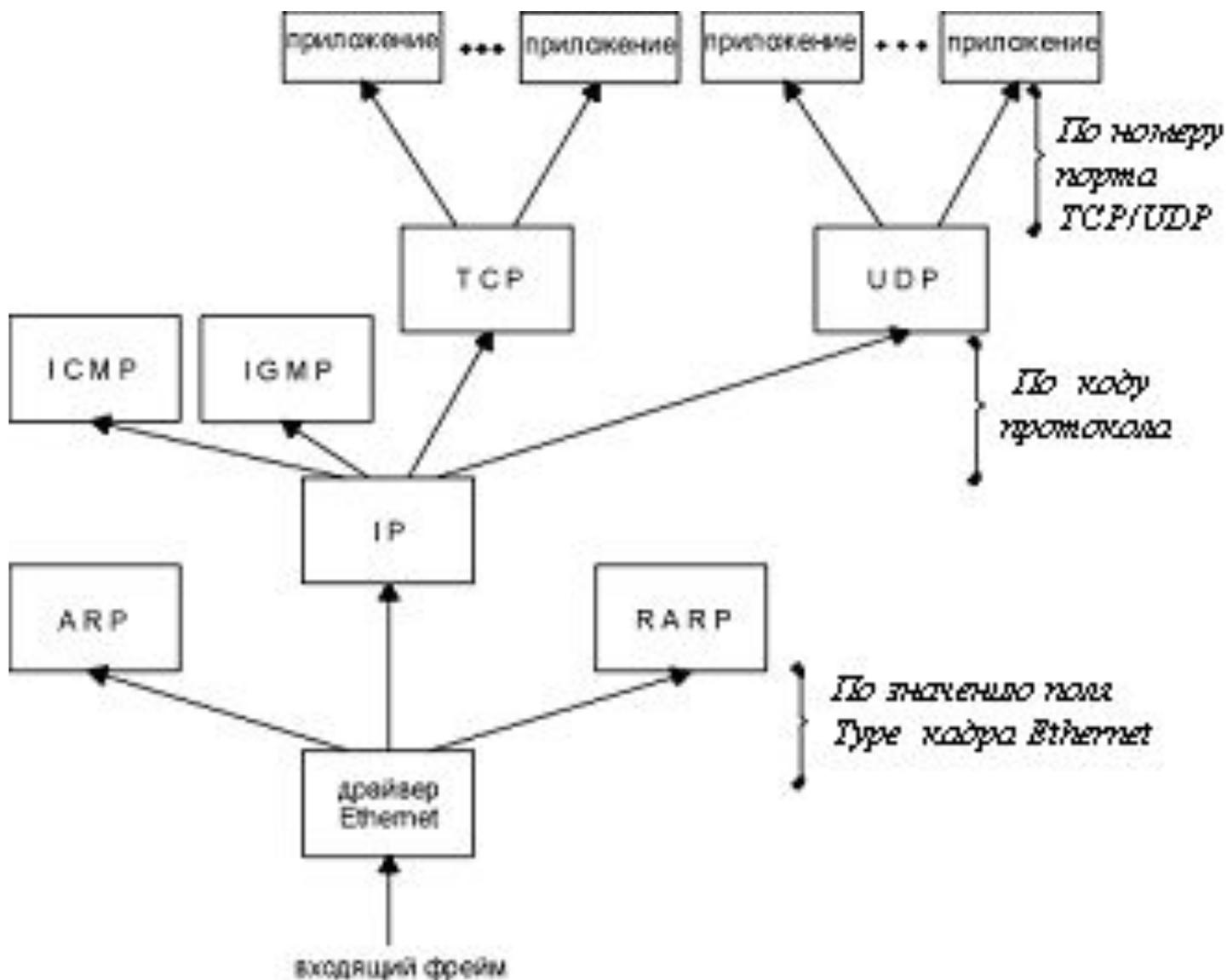
# Взаимодействие компьютеров различных ЛВС, связанных через маршрутизатор



# Инкапсуляция данных на передающей стороне



# Демультимплексирование на различных уровнях



# ЭМВОС: Выводы

- Основа построения сетевых средств связи —  
уровневая, модульная модель (ЭМВОС)
- Открытая система строится на основе доступных и  
общепринятых стандартов
- Формальные правила взаимодействия одноуровневых  
модулей – протоколы
- Формальные правила взаимодействия смежных  
уровней – межуровневые интерфейсы
- Независимость уровней (каждый уровень имеет свой  
заголовок, инкапсуляция)
- Множество стеков протоколов
- Множество технологий и приложений