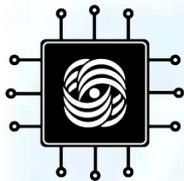


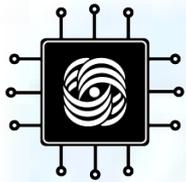
# **АРХИТЕКТУРА КОМПЬЮТЕРНЫХ СИСТЕМ**

## **Лекция 11: Компьютерные Сети**



# План лекции

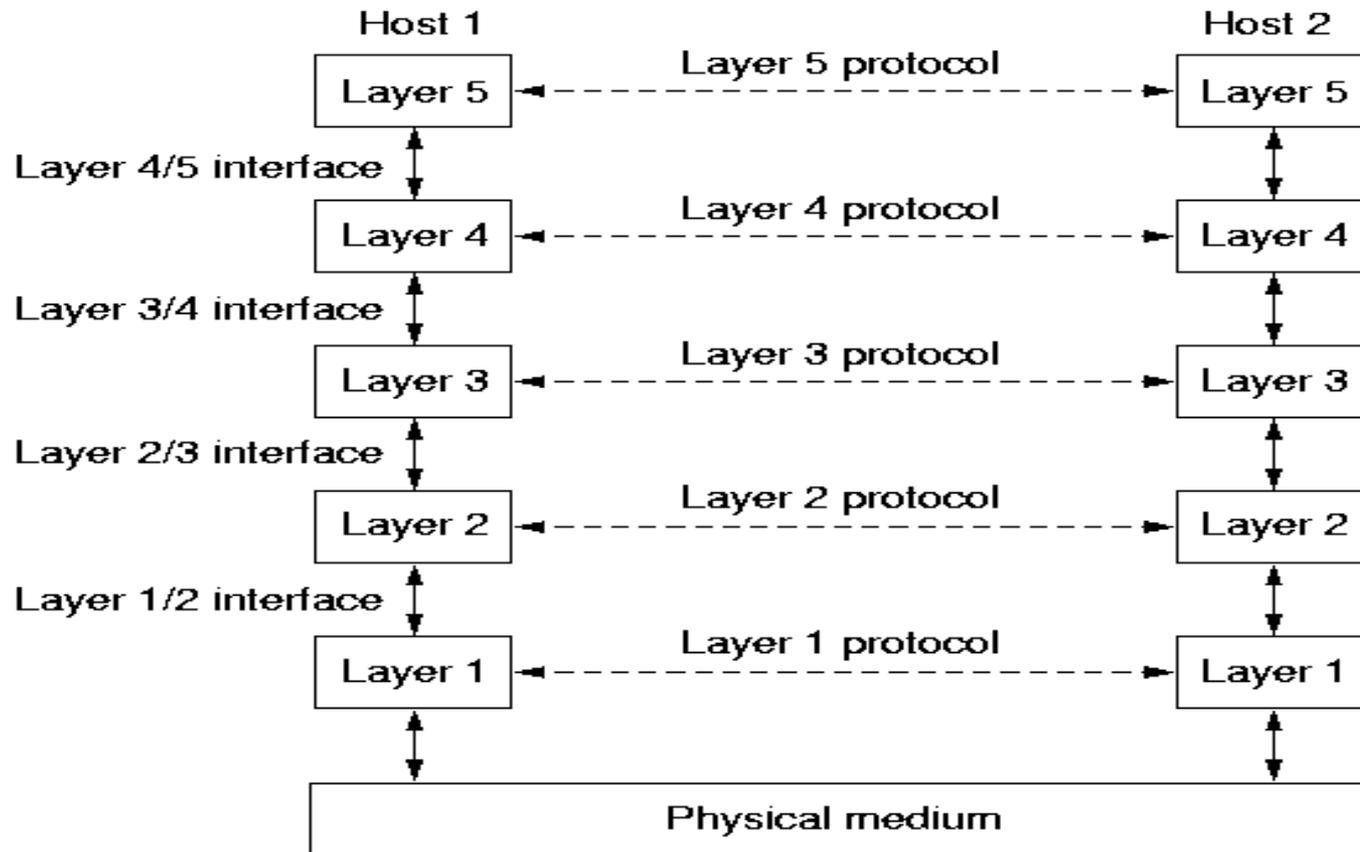
- Иерархия протоколов
- Сервисы и интерфейсы
- Модель ISO/OSI
- Модель TCP/IP
- Примеры сетей

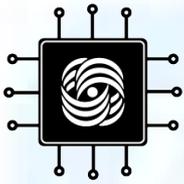


# Организация вычислительных сетей

- *Сеть = абонентские машины + транспортная среда.*
- *Абонентские машины* обеспечивают интерфейс пользователей и работу приложений в сети.
- *Транспортная среда* = Коммутирующие элементы + СПД
- *Коммутирующие элементы* - маршрутизаторы, мосты, шлюзы.
- *СПД* - коммуникационные каналы и переключатели.
- *Коммуникационные каналы* - это линии связи самой различной природы и каналообразующая аппаратура

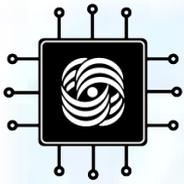
# Иерархия протоколов





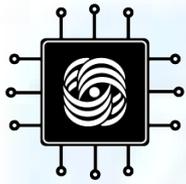
# Проблемы, решаемые на каждом уровне

- на каждом уровне нужен механизм для определения отправителей и получателей;
- правила передачи данных
  - simplex/half-duplex, duplex
  - количество виртуальных каналов через одно соединение и приоритеты между ними
- обнаружение и исправление ошибок
- сохранение исходной последовательности данных при передаче



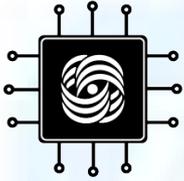
# Проблемы, решаемые на каждом уровне

- на каждом уровне нужен механизм предотвращающий ситуацию, когда получатель начинает "захлебываться"
- регулирование длины сообщения
  - разбиение и сборка сообщений
  - как быть, если процесс работает со столь короткими сообщениями, что их отдельная пересылка не эффективна
- мультиплексирование и демупльтиплексирование виртуальных каналов
- когда между получателем и отправителем есть несколько маршрутов: какой выбрать?



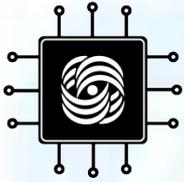
# Архитектура и интерфейс

- *интерфейс* определяет какие примитивы - элементарные операции - и какие услуги (сервис) нижележащий уровень должен обеспечивать для верхнего уровня.
- набор уровней и протоколов называется *архитектурой сети*.
  - Вопросы реализации, определения интерфейсов - не относятся к архитектуре
- *стек протоколов* - набор конкретных протоколов по одному на уровень



# Интерфейс и сервис

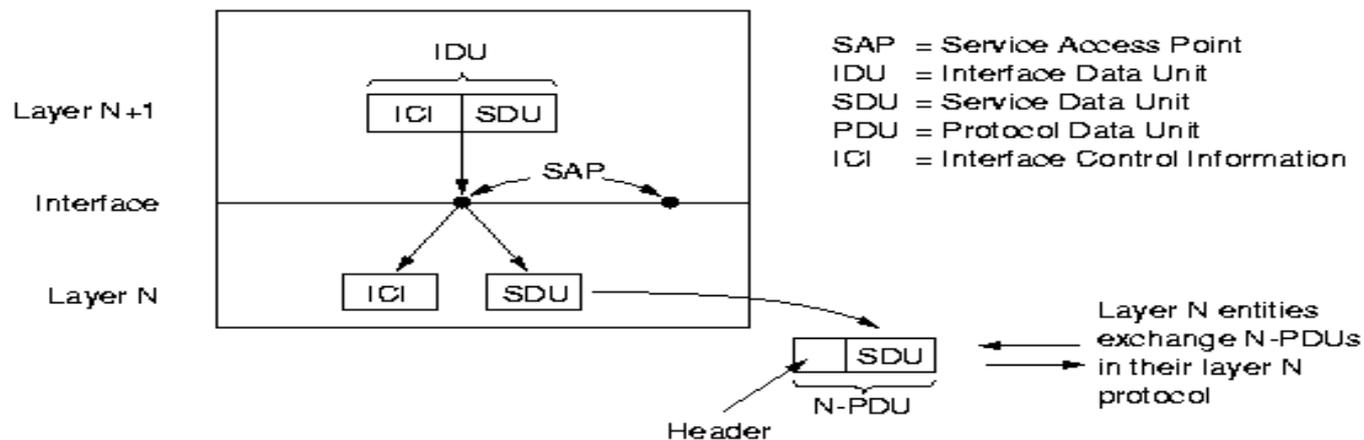
- равнозначная активность
- пользователи сервиса и поставщики сервиса
- **точки доступа к сервису - SAPs** ( service access points)
- как между двумя уровнями происходит обмен информации определяет **интерфейс** между ними
  - активности передают **IDU** (Interface Data Unit) через SAP
  - IDU состоит из **SDU** (Service Data Unit) и управляющей информации



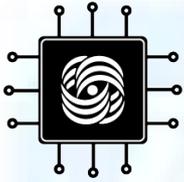
# Интерфейс и сервис (2)

- IDU состоит из **SDU** (Service Data Unit) и управляющей информации
  - SDU передает по сети равнозначной сущности.
  - управляющая информация нужна нижележащему уровню, чтобы правильно передать SDU, но она не является частью

# Взаимосвязь уровней через интерфейс

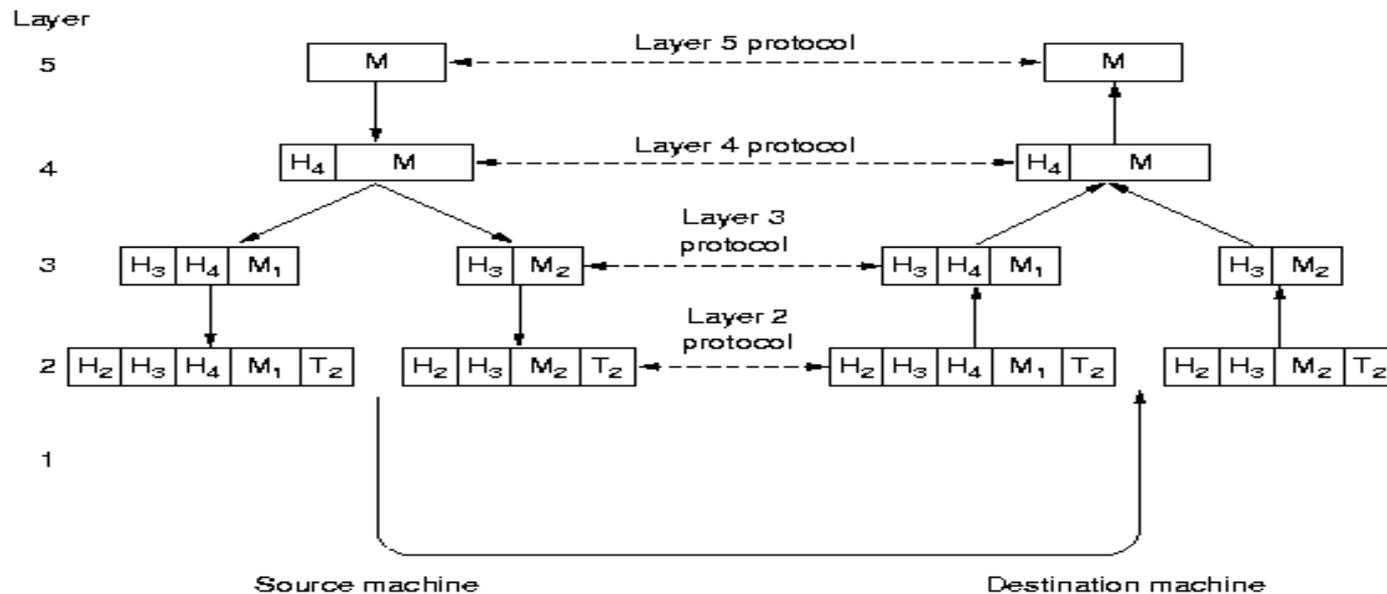


Взаимосвязь уровней и интерфейса

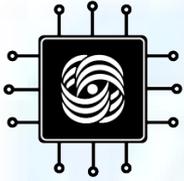


# Типы и классы сервиса

- **Сервис с соединением и без соединения.**
- **Сервис с соединением** сначала устанавливает соединение, и только потом доставляется сервис. Пример - телефонная сеть.
- **Сервис без соединения** действует подобно почтовой службе. Каждое сообщение маршрутизируется независимо от других сообщений.
- Любой сервис характеризуется качеством.
  - Надежность - дополнительные расходы на подтверждение
  - Поддержка структуры или поток байтов

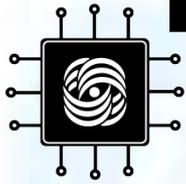


Пример организации потока информации при виртуальном взаимодействии уровней 5



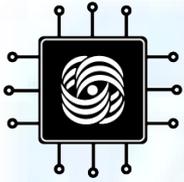
# Примитивы сервиса

- Сервис формально определяет набор **примитивных операций** (или **примитивов**), с помощью которых пользователь или какая-либо активность получает доступ к сервису.
- Сервис может быть надежный (с подтверждением) и не надежный (без подтверждения).



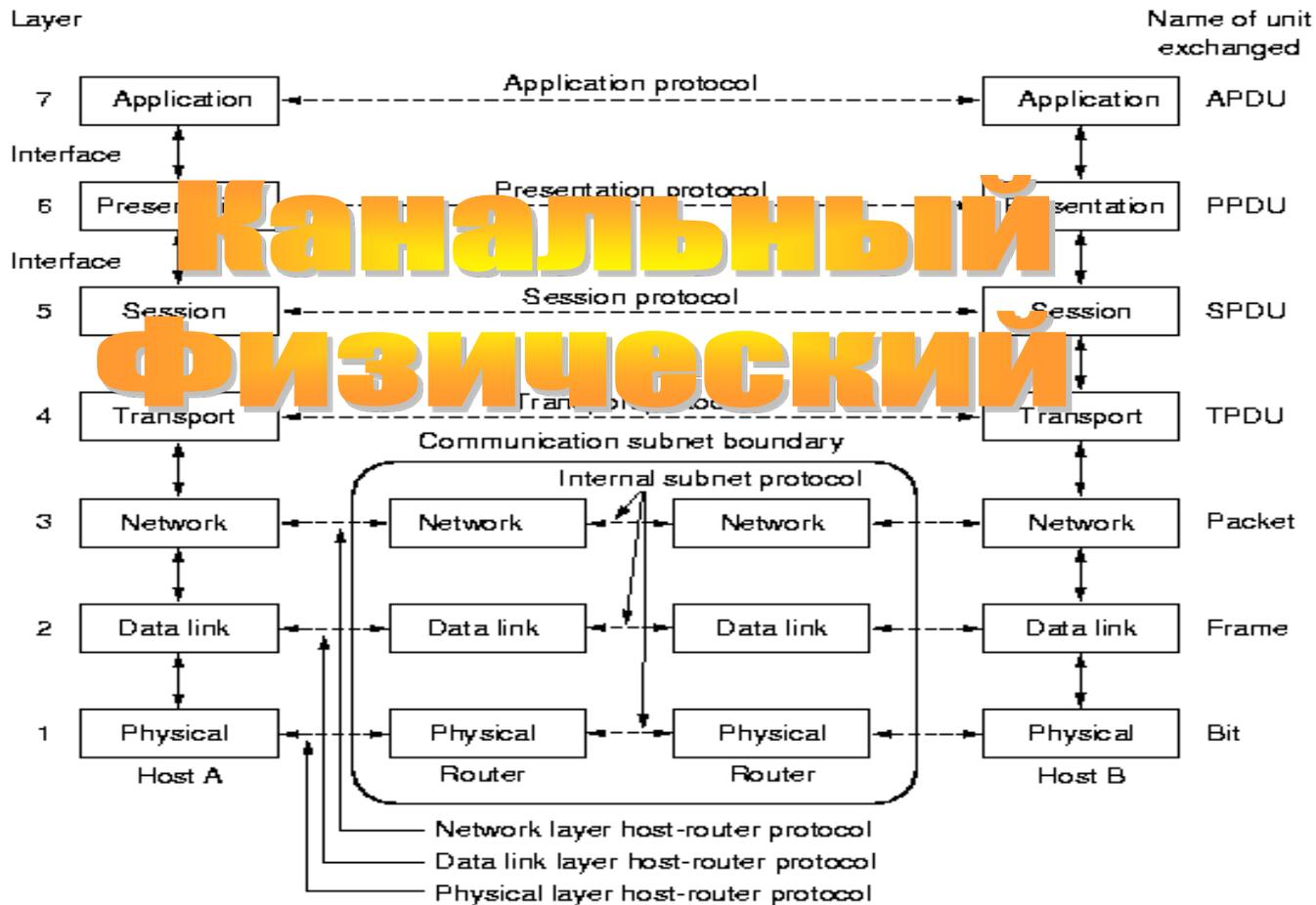
# Взаимосвязь сервиса и протоколов

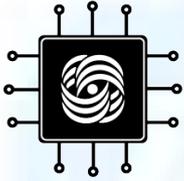
- Сервис определяет, какие операции данный уровень должен выполнить по поручению его пользователей, но он ничего не говорит о том, как эти операции реализованы.
- Сервис относится к интерфейсу между уровнями.
- Протокол - это набор правил, определяющих формат, назначение кадров, пакетов, сообщений, которыми обмениваются равнозначные активности
- Протокол и сервис не связанные понятия.



# Эталонные модели

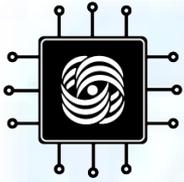
- Каждый уровень отражает надлежащий уровень абстракции.
- Каждый уровень имеет строго определенную функцию.
- Эта функция выбиралась прежде всего так, чтобы можно было определить международный стандарт.
- Границы выбирались так, чтобы минимизировать поток информации через интерфейсы.
- Число уровней должно быть достаточно большим, чтобы не объединять разные функции на одном уровне и оно должно быть достаточно малым, чтобы архитектура не была громоздкой.





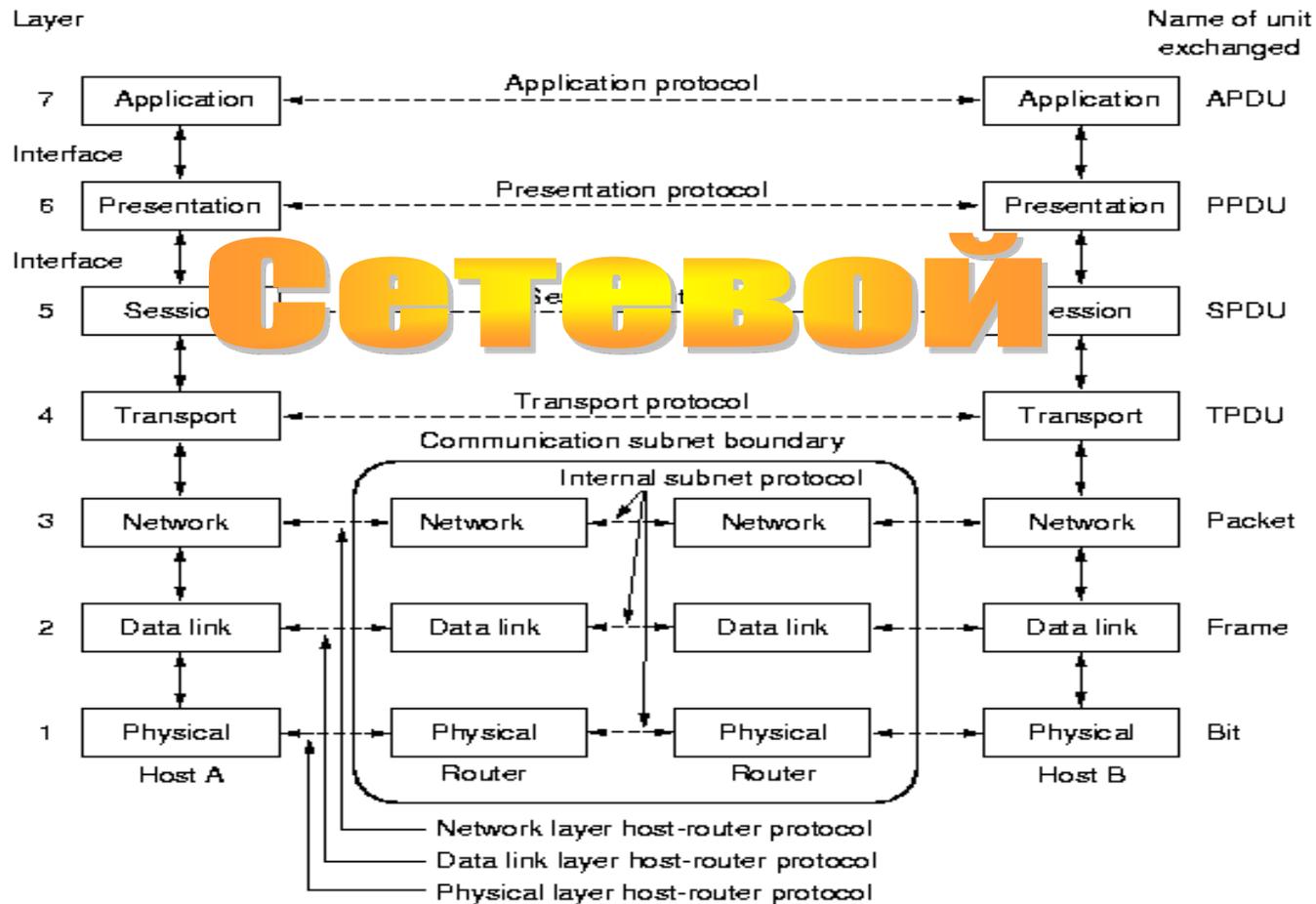
# Физический уровень

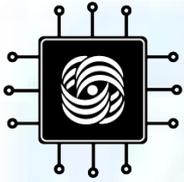
- *Физический уровень* отвечает за передачу последовательности битов через канал связи. Здесь в основном решаются вопросы механики, электрики



# Канальный уровень

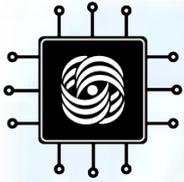
- Основной задачей *уровня канала данных* - превратить несовершенную среду передачи в надежный канал, свободный от ошибок передачи.
  - разбиение на фреймы
  - определение границ фреймов
  - исправление ошибок при передаче на физическом уровне
  - уничтожать дубликаты фреймов
  - минимизировать затраты на служебные фреймы (piggy backing)
  - регулировать доступ к каналу с множественным доступом (MAC подуровень)





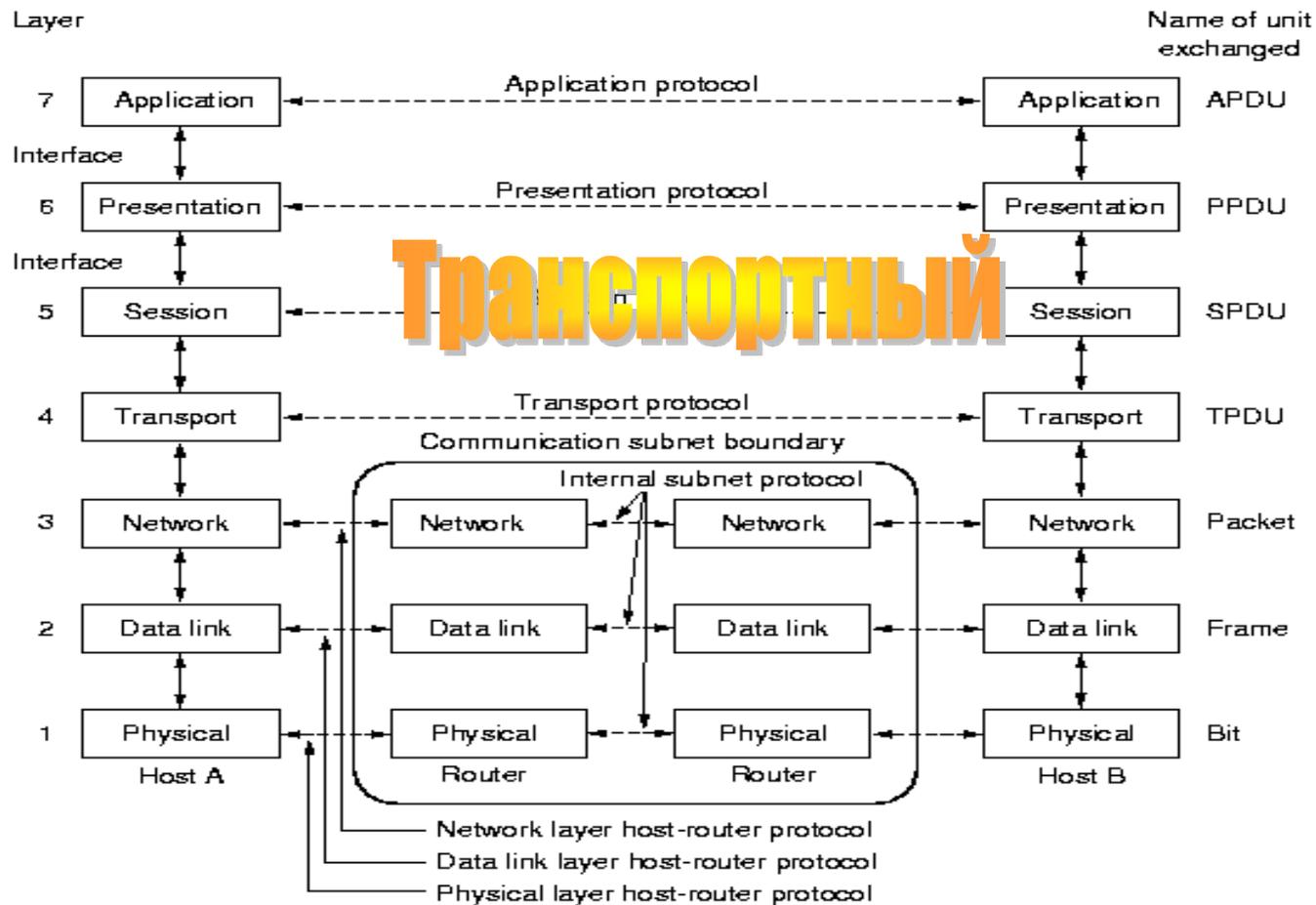
# Эталонная модель OSI

- *Сетевой уровень* отвечает за функционирование транспортной подсети
  - Основная задача маршрутизировать пакеты от отправителя к получателю
    - маршруты могут быть определены статически (заранее) и динамически в зависимости от текущей загрузки и состояния транспортной подсети
    - в сетях на основе СПД с каналом множественного доступа проблема маршрутизации практически отсутствует
  - Биллинг за услуги
  - Шлюзование

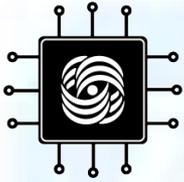


# Эталонная модель OSI

- *Сетевой уровень* определяет какой тип сервиса предоставить вышележащим уровням
  - канал точка-точка без ошибок, обеспечивающий доставку сообщений или байтов в той последовательности, в какой они были отправлены.
  - доставка отдельных сообщений без гарантии сохранения их последовательности, рассылка одного сообщения многим в режиме вещания.
- Класс сервиса определяется при установлении транспортного соединения

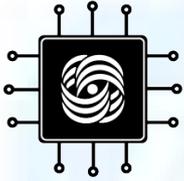


# Транспортный



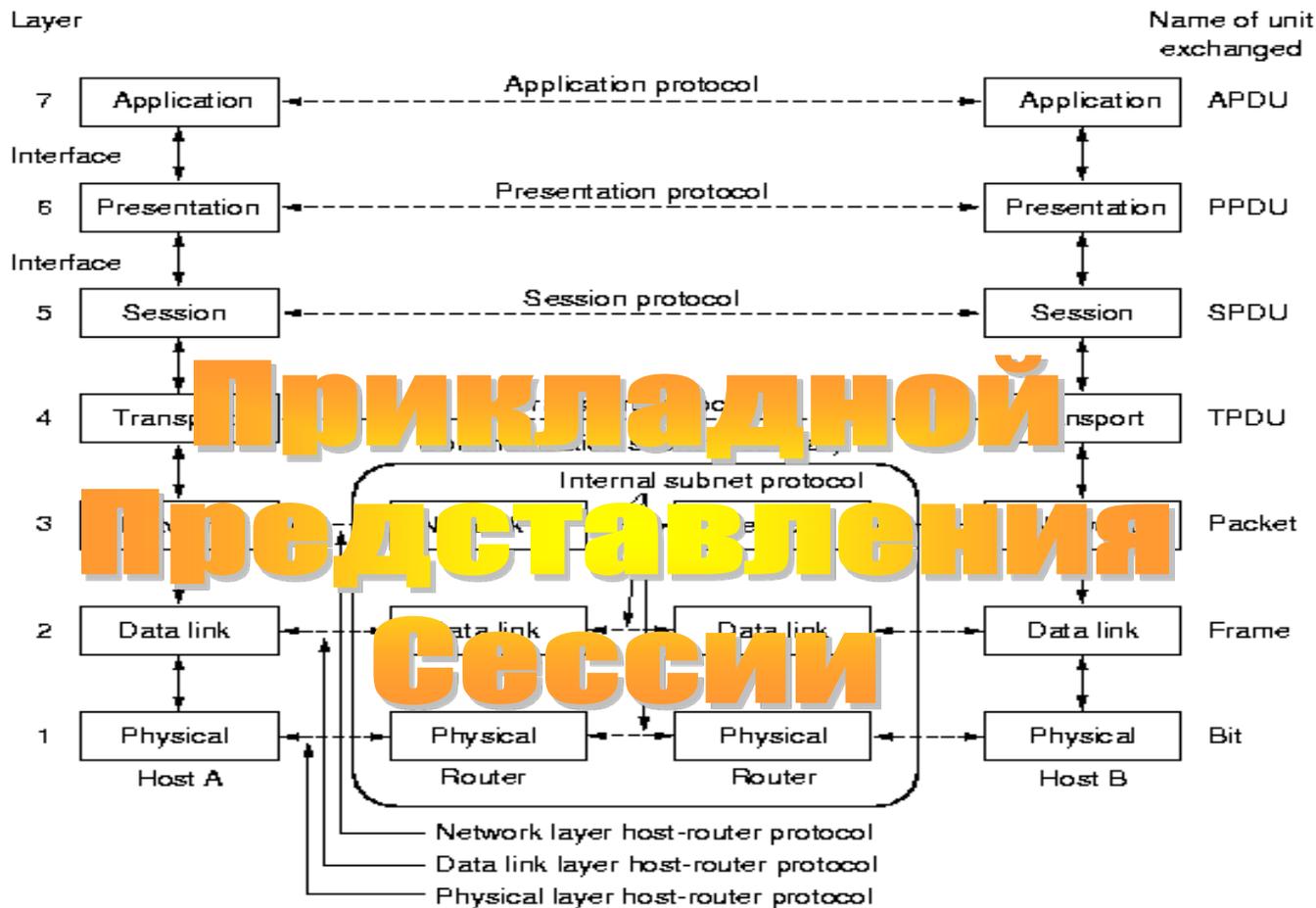
# Эталонная модель OSI

- Основная функция *транспортного уровня* это: принять данные с уровня сессии, разделить, если надо, на более мелкие единицы, передать на сетевой уровень и позаботиться, чтобы все они дошли в целостности до адресата.
- транспортный уровень должен создать транспортное соединение по запросу уровня сессии
  - одно транспортное соединение может использовать несколько сетевых
  - нескольких транспортных соединений - одно сетевое соединение.

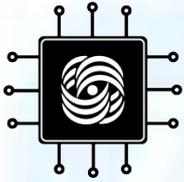


# Эталонная модель OSI

- транспортный уровень и ниже обеспечивают соединение типа точка-точка
- абонентские-машины - мультипрограммные, поэтому транспортный уровень для одной такой машины должен поддерживать несколько транспортных соединений.
- установление и разрыв транспортного соединения в сети
- транспортный уровень обеспечивает механизм *управления потоком*.
- управление потоком между хостами отличен от управления потоком между маршрутизаторами.

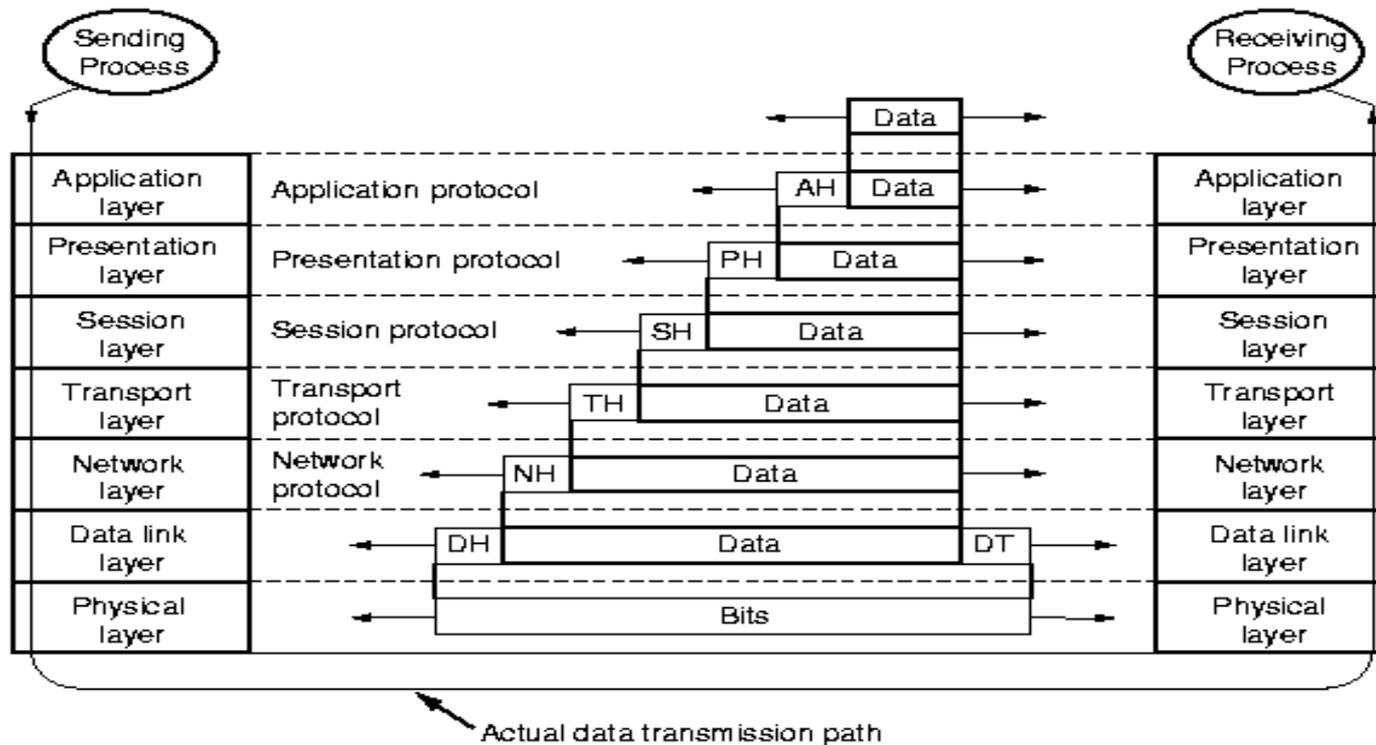


# Прикладной Представления Сессии

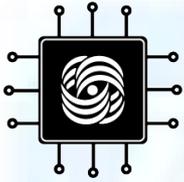


# Эталонная модель OSI

- *уровень сессии* обеспечивает сложный сервис, полезный в некоторых приложениях.
  - вход в удаленную систему, передать файл между двумя приложениями.
  - управление направлением передачи.
  - **управление маркером**. (операцию может выполнять тот кто владеет маркером).
  - **синхронизация** (расстановка контрольных точек).
- *уровень представления* решает проблемы семантики и **синтаксиса** передаваемой информации. (кодировка и структура данных)
- *уровень приложений* обеспечивает нужные часто используемые протоколы (TELNET, FTP)

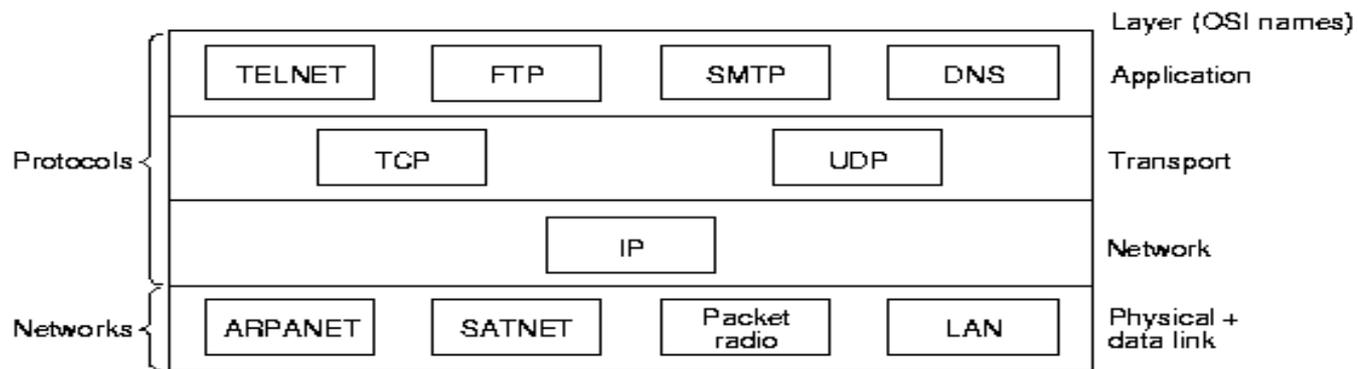


Пример передачи данных в модели OSI

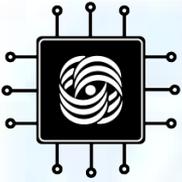


# Эталонная модель ТСР/ІР

- два транспортных протокола.
  - **ТСР** (Transmission Control Protocol) : надежный протокол с соединением. Он получает поток байт, фрагментирует его на отдельные пакеты и передает их на межсетевой уровень.
  - **UDP** (User Datagram Protocol): ненадежный протокол без соединения для тех приложений, которые используют свои механизмы фрагментации, управления потоком

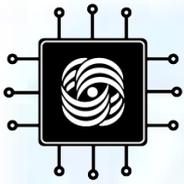


## Протоколы и сети, учитываемые моделью TCP/IP



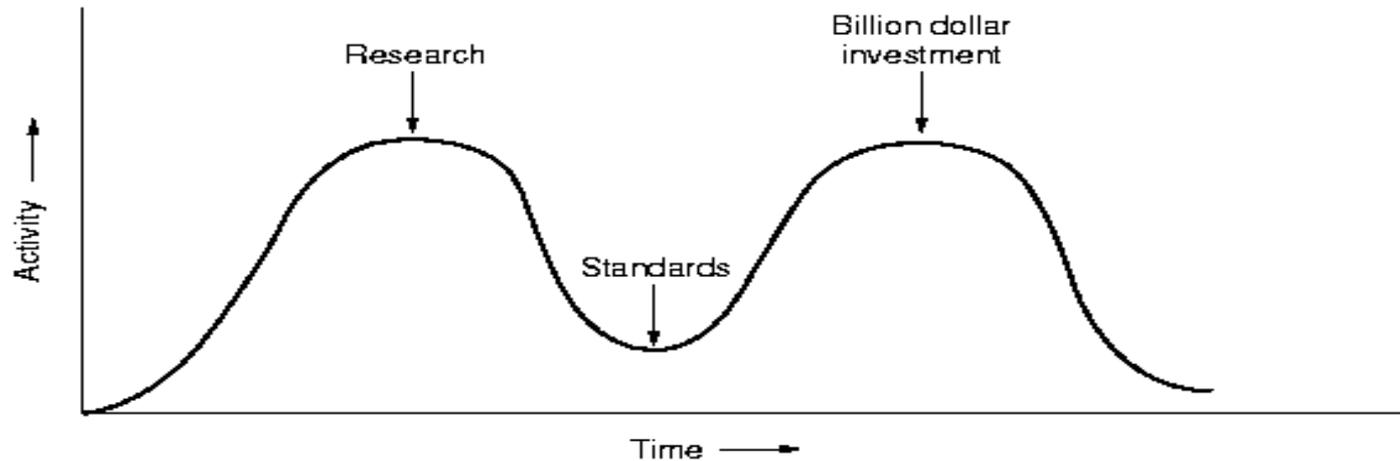
# Сравнение моделей ISO и TCP/IP

- В модели OSI центральными являются три понятия:
  - *сервис* определяет что делает уровень, но ничего не говорит как.
  - *интерфейс* определяет для вышележащего уровня доступа к сервису
  - *протокол*. определяет реализацию сервиса



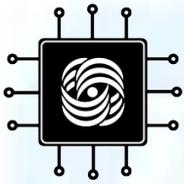
# Сравнение моделей ISO и TCP/IP (2)

- В TCP/IP модели нет столь же четкого выделенных понятий (post factum)
- Модели имеют разное число уровней
  - OSI модель поддерживает на сетевом уровне как сервис с соединением, так и без соединения. На транспортном уровне этой модели поддерживается сервис только с соединением.
  - TCP/IP : сетевой уровень обеспечивает сервис без соединения, но транспортный - как с соединением, так и без.
  - Разный взгляд на приоритеты и распределение инвестиций



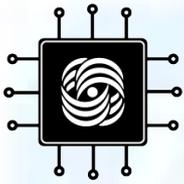
**Принцип двух слонов**

**Не вовремя**



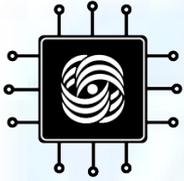
# Недостатки модели и протоколов МОС

- функциональность между семью уровнями распределена не равномерно
- ISO поспешило за IBM SNA
- описание модели и ее протоколов очень сложно
- некоторые функции такие как управление потоком, исправление ошибок, адресация - повторяются на каждом уровне, для некоторых функций не ясно на какой уровень их поместить (виртуальный терминал)
- шифрование и защита отсутствуют в модели
- хорошо проработан сервис с соединениями и слабо сервис без соединений
- влияние МКТТ (не отражена взаимосвязь между вычислениями и связью (indication vs receive)).



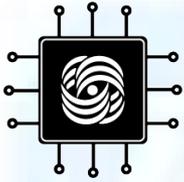
# Недостатки эталонной модели ТСП/IP

- В модели нет четкого разграничения понятий сервис, интерфейс, протокол;
- Эта модель годится только для описания стека ТСП/IP;
- Уровень хост-сеть по существу уровнем не является, это больше интерфейс;
- В этой модели не разделяются физическая среда передачи и уровень канала данных;
- Многие протоколы прикладного уровня проработаны плохо (TELNET)
- Не учтены вопросы безопасности
- Структура IP пакета оказалась слишком жесткой и ограниченной



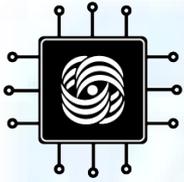
# Примеры сетей

- **NetBIOS** - Net Basic Input Output system. Это сетевое расширение стандарта I/O операций фирмы IBM для IBM PC.
- **SNA, DNA** - Сетевые архитектуры фирм IBM и DEC. Работают под их операционными системами. В настоящее время не имеют широкого распространения.



# Internet

- NSFNet (1984) - высокоскоростной приемник ARPANet.
  - 6 суперкомпьютерных центров
  - 20 региональных подсетей в США
  - ANS (Advanced Network and Service) - некоммерческая корпорация (MCI, IBM, MERIT): СПД увеличилась с 1.5 до 45 Mbps
  - NREN (1991) - 3Gbps инфраструктура для informational highway.
  - В Европе создаются IP сети ENET, EBONE
  - 1990 - подключение EARN, NER, SPAN, BITNet
  - С 1995 во всем мире идет развитие и модернизация сетей на Mbps и Gbps



# Internet

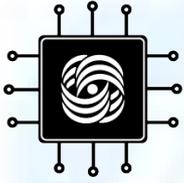
- На сегодня более 400 млн. абонентских машин в 170 странах
- Проблемы
  - ограниченность адресного пространства
  - переполнение таблиц маршрутизации и ограниченность пропускной способности каналов опорной сети
  - ограниченные возможности по интеграции услуг и потоков данных (голос, аудио, видео, данные)
  - эффективные механизмы обеспечивающие гарантированное качество услуг
  - защита информации





Москва

-  Действующая ЗС
-  Планируемая ЗС
-  Газопроводы
-  Газовые месторождения



**Спасибо за внимание!**