

# Еталонен модел на мрежите

Характеристики на нивата.  
Модел ТСР/ІР.

# Какво ще научим

Защо е избрана слоеста архитектура. Какво печелим.

Понятие за услуги, интерфейси, прозрачност.

Модел на ISO и модел TCP/IP.

Слоестата архитектура е взаймствана от системната. (Интернет е един **глобален компютър**.)

Ще бъде ли TCP/IP изместен от мрежовото кодиране?

# Слоеста системна архитектура

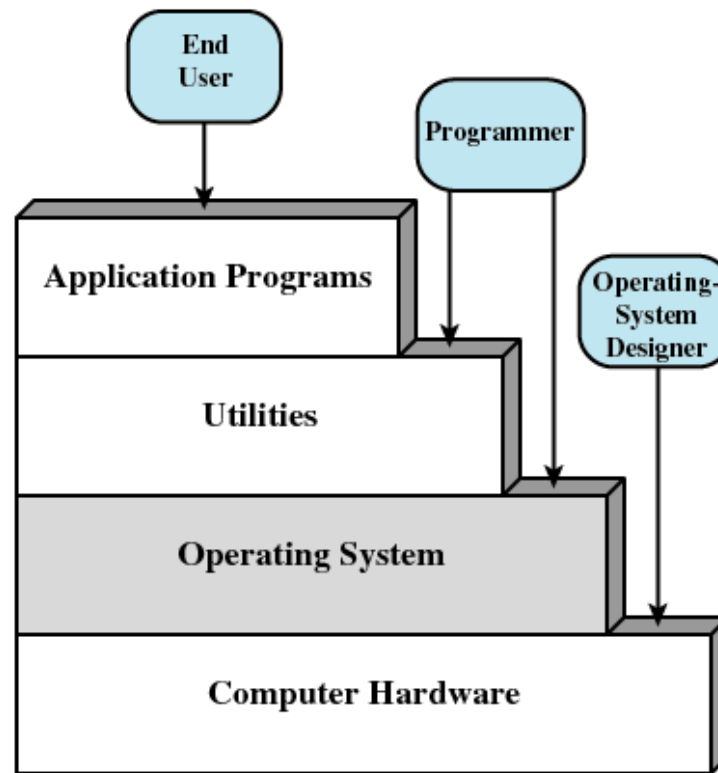


Figure 2.1 Layers and Views of a Computer System

# Архитектура на мрежите

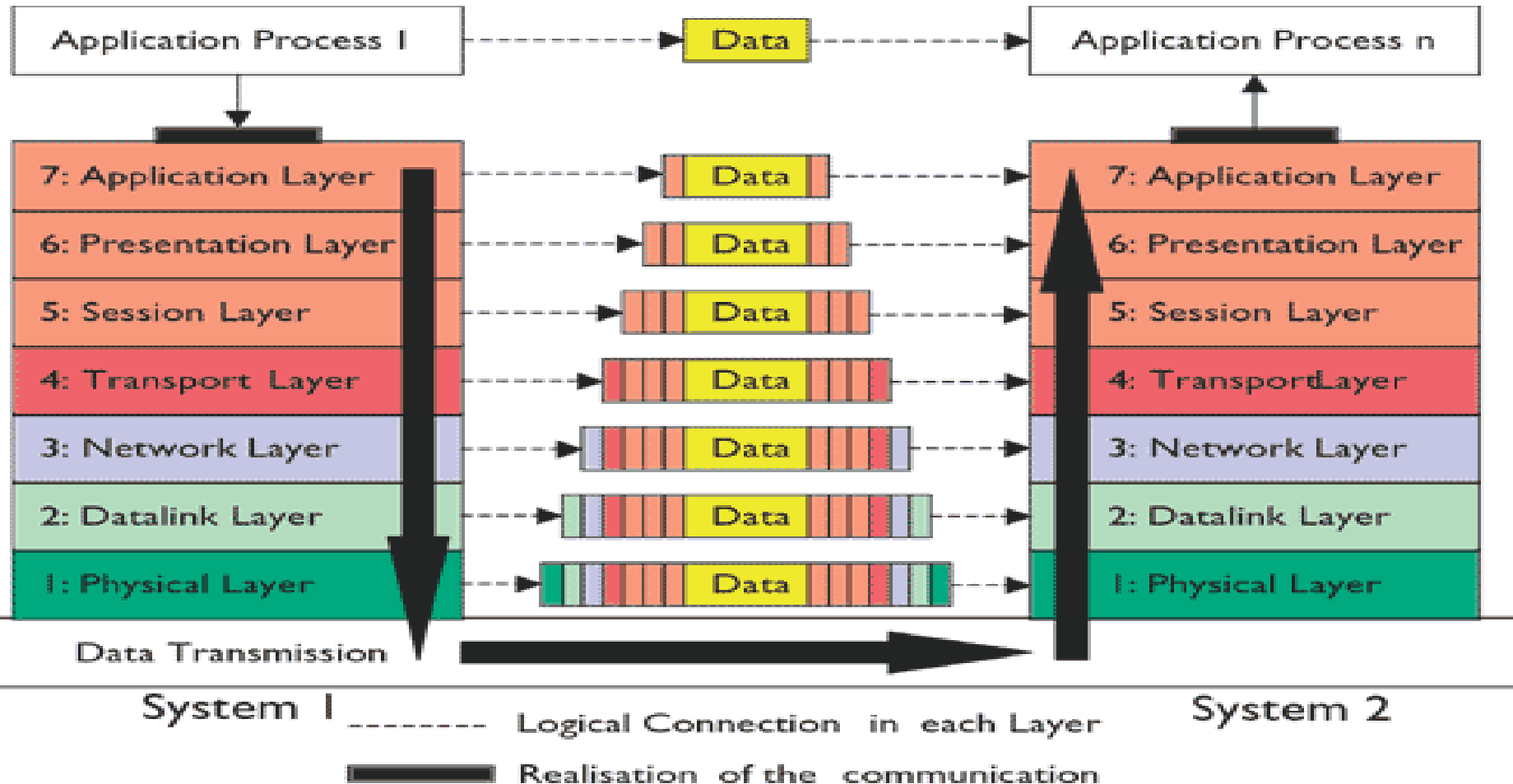
**Основен принцип** в съвременните мрежови архитектури е принципът за **разслояване на функциите** по управление на връзките, като всеки слой ползва услугите, предоставени от по-долните слоеве, без да знае как са реализирани тези услуги. Това е принципът на **прозрачност**.

Слоят  $n$  на една машина взаимодейства със слой  $n$  (на същото ниво) на друга машина. Правилата, по които се осъществява това взаимодействие, се определят от **протокола на  $n$ -то ниво**.

Най-общо под **протокол** се разбира съгласувани правила между комуникиращите страни за това как да протича комуникацията.

На практика при комуникацията между съответните слоеве на двете машини не се предават данни. Всеки слой  $n$  предава данни и контролна информация (**header+trailer**) на непосредствено по-долния слой  $n-1$ , докато се достигне най-долния слой  $1$ , където се осъществява реалната комуникация между машините през физическата среда. В приемника получените данни се разпространяват в обратна посока - от слой  $1$  нагоре, като всеки слой премахва контролната информация, която се отнася до него. **Опаковане и разопаковане (encapsulation – decapsulation)**.

# Архитектура на мрежите



Данните+Контр. Инф. на слой  $n$  се наричат протоколен блок от данни (PDU). За слой  $n-1$  PDU( $n$ ) са си обикновени данни. Чисто потребителските данни – **payload**.

# Архитектура на мрежите

Всеки **слой  $n$**  предоставя **интерфейс** на **слой  $n+1$**  - функциите и услугите, които слой  $n$  предоставя на слой  $n+1$ . Ясно да се знае какви функции изпълнява всеки слой.

**Разслояването** позволява да се промени изцяло реализацията на даден слой  $n$ , без да се променя реализацията на другите слоеве – достатъчно е да се запази множеството от услугите, които слой  $n$  осигурява на горния слой  $n+1$ . Прозрачност (**transparency**) и Гъвкавост (**flexibility**).

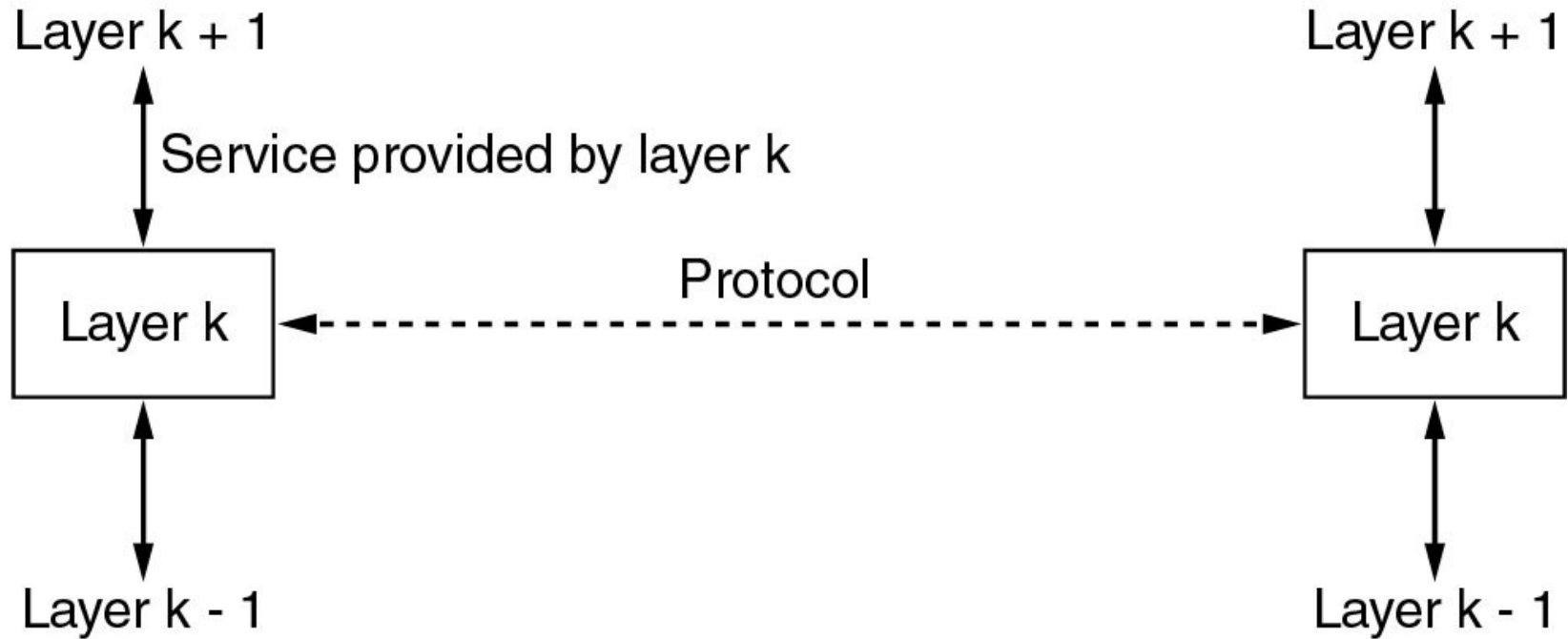
Една **мрежова архитектура** се определя от **множеството на слоевете**, услугите които те предоставят и протоколите, по които се осъществява взаимодействие между слоевете на едно и също ниво.

# Архитектура на мрежите

Реализацията на слоевете, както и интерфейсът между отделните слоеве не е задължително да са едни и същи на машините в една мрежа — достатъчно е всеки слой  $n$  да може да комуникира със съответния слой  $n$  по определения протокол и да предоставя съответните услуги на по-горния слой. Мащабируемост (*scalability*).

Списъкът от протоколи, използвани от една система, по един протокол за всеки слой се нарича **протоколен стек**.

# Протоколи и услуги



Протоколи и услуги на едно ниво:  $k$ .



# Моделът OSI

Съвременните мрежови архитектури следват принципите на **модела OSI (Open Systems Interconnection)**, създаден от Международната организация по стандартизация **ISO** (International Standards Organization) за връзка между отворени системи.

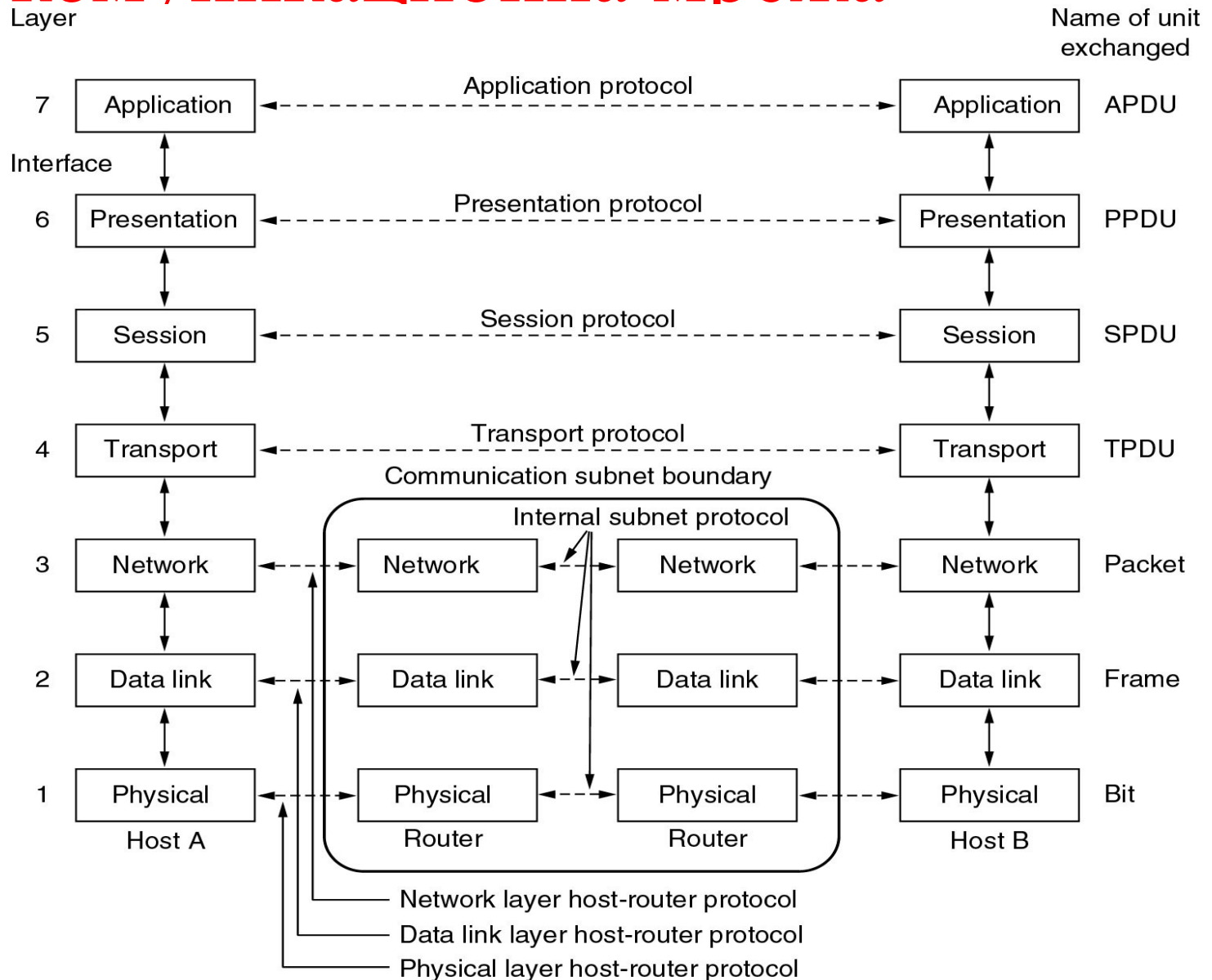
**Отворена система** е система, чиито ресурси могат да се използват от другите системи в мрежата.

OSI моделът е абстрактен модел на мрежова архитектура, който описва предназначението на слоевете, но не се обвързва с конкретен набор от протоколи. Поради това OSI моделът се нарича още **еталонен модел** и всъщност дава препоръки (**Reference Model**).

В еталонния модел има седем слоя – физически, канален, мрежов, транспортен, сесиен, представителен, приложен.

# OSI RM – хост машини и КОМУНИКАЦИОННА МРЕЖА

The OSI  
reference  
model.



# Физически слой

**Физическият слой** (physical layer) има за задача да реализира предаването на битове през физическата среда.

**Основна функция** на физическия слой е да управлява кодирането и декодирането на сигналите, представящи двоичните цифри 0 и 1. Той не се интересува от предназначението на битовете.

Физическият слой трябва да осигурява възможност на по-горния слой да активизира, поддържа и прекратява физическите съединения.

Обекти на този слой – хардуерни устройства, реализиращо предаването на 0-и и 1-ци през физическата среда – мрежови карти (NIC) и модули, модеми.

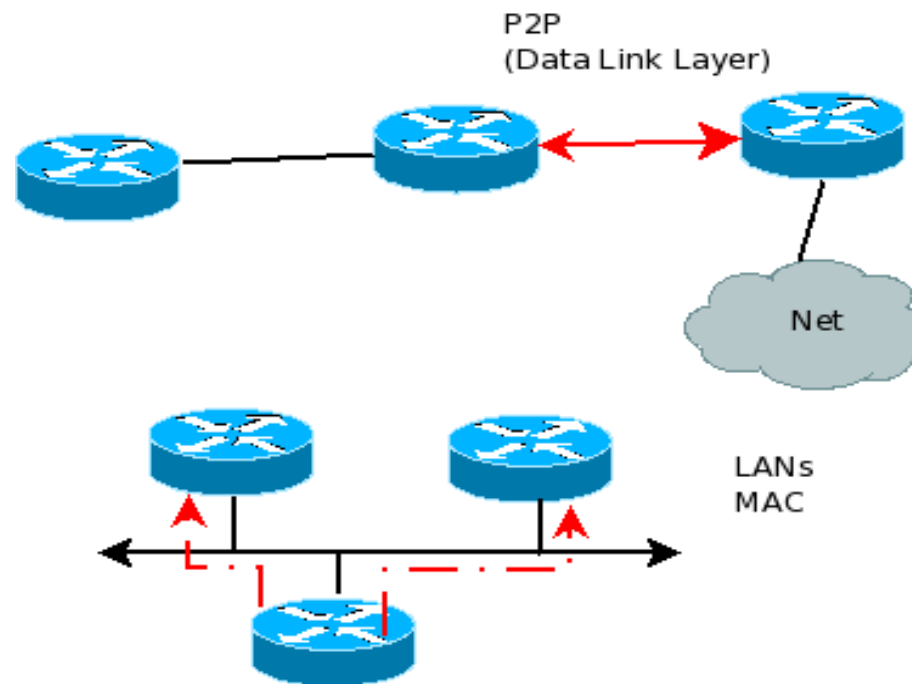
# Канален слой

Основна функция на **каналният слой (data-link layer)** е управлението на канала от един възел до друг (точка-точка) според класическия модел, “точка-много точки” (напр. Frame Relay) или достъп до преносната среда (MAC) в LAN.

Откриването и евентуалното коригиране на грешки при предаването на данните.

Данните на канално ниво се обменят на порции (PDU), наречени **кадри (frames)**, обикновено с дължина от няколко стотин до няколко хиляди байта в зависимост от скоростта на линията.

# Канален слой



# Канален слой

При надеждна комуникация приемникът трябва да уведомява изпращача за всеки успешно получен кадър като му изпраща обратно потвърждаващ кадър.

Форматът на кадрите се определя от избрания протокол на канално ниво. Функциите на каналния слой обикновено се реализират смесено - апаратно и програмно. Колкото повече функции са реализирани софтуерно (контролерът е реализиран на дънната платка), по-ниска е производителността.

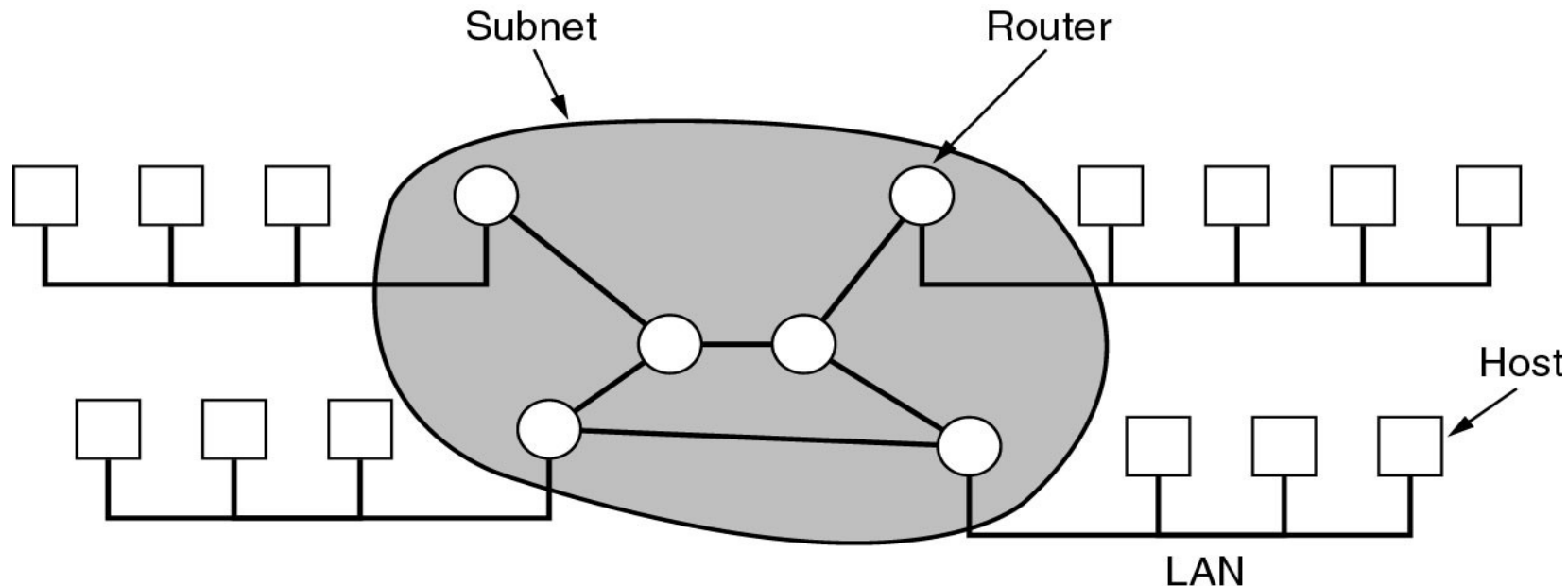
# Мрежов слой

**Мрежовият слой (network layer)** отговаря за функционирането на комуникационната подмрежа.

Приложните програми, които се изпълняват в двете крайни системи взаимодействат помежду си посредством **сегменти** от данни.

Пакетите са с фиксирана големина в рамките на една мрежа. Но при преминаване от една КМ в друга е възможно пакетът да се раздели на части – **фрагментира**, след което да се възстанови. Напр. Преход: LAN-WAN-LAN

# Мрежов слой

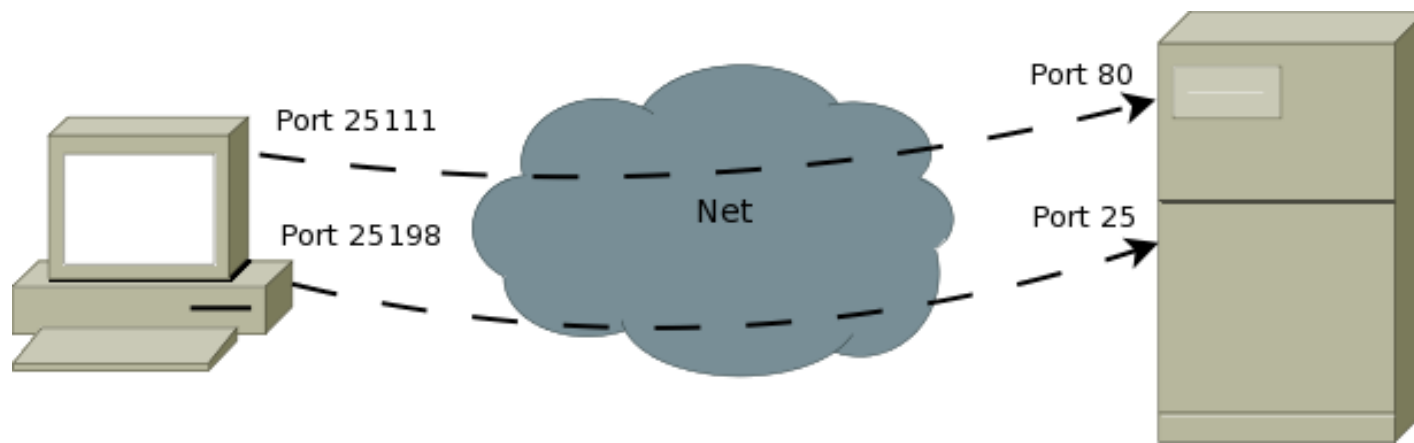


Основна задача на мрежовия слой е **маршрутизирането** на тези сегменти, опаковани като **пакети (PDU за мрежов слой)**.

За системите, реализиращи възлите на комуникационната подмрежа (**маршрутизатори - routers**) този слой е последен. Функциите на мрежовия слой, както и на по-горните слоеве се реализират програмно.



# Транспортен слой



**Транспортният слой** (transport layer) осигурява **транспортирането на съобщения от източника до получателя**. Той е най-ниският слой, който реализира връзка от тип “край-край” между комуникиращите системи.

Изгражда **програмен канал между портовете** на приложения, които си “говорят” през мрежата.

# Транспортен слой

В транспортния слой на изпращача съобщенията се разбиват на **сегменти** (PDU за тр. слой) и се подават на мрежовия слой, където се опаковат като пакети, а в транспортния слой на получателя разопакованите от мрежовия слой сегменти се реасемблират.

Транспортният слой освобождава по-горния сесиен слой от грижата за надеждното и ефективно транспортиране на данните между крайните системи.

Т.е транспортният слой отговаря за целостта на обменяните съобщения, което включва откриване на загубени сегменти и тяхното повторно предаване.

# Сесиен слой

**Сесийният слой** (**session layer**) е отговорен за диалога между две комуникиращи програми. Съобщения се обменят след като двата крайни абоната установят **сесия**.

Сесийният слой осигурява различни режими на диалог – двупосочен едновременно диалог (**full duplex - FD**), двупосочен алтернативен диалог (**half duplex - HD**), еднопосочен диалог (**simplex**).

Освен това той предоставя възможност за прекъсване на диалога и последващо възстановяване от мястото на прекъсването.

При липсата на сесиен слой всяко съобщение се предава независимо от другите съобщения.

# Представителен слой

**Представителният слой (presentation layer)** е най-ниският слой, който разглежда значението на предаваната информация.

**Първата функция** на този слой е да определи общ синтаксис за предаване на съобщенията.

**Втората функция** на слоя е да унифицира вътрешната структура на представените данни в съобщенията.

По този начин за по-горния приложен слой няма значение дали двете крайни системи използват различни представяния на данните.

**UTF-8** (8-bit UCS/Unicode Transformation Format) представя всеки символ в Unicode стандарта, но е и обратно съвместим с ASCII. По тези причини е предпочитан за e-mail, web страници и др.

Криптиране на данните, компресия.

# Приложен слой

**Приложният слой (application layer)** е най-горният слой, към който се свързват потребителските процеси в двата крайни абоната.

Някои потребителски процеси са интерактивни - взаимодействат си в голям период от време с кратки съобщения от тип заявка-отговор (request-reply).

Други потребителски процеси взаимодействат с малко на брой големи по обем порции от данни.

За двата вида процеси се предвиждат различни протоколи на приложния слой - например протокол **FTP** (file transfer protocol) за обмен на цели файлове, протокол **HTTP** (hyper text transfer protocol) за обмен на веб-страници и др.

# Модел ТСР/ІР

Когато започват да се изграждат реални мрежи, използвайки OSI-модела и съществуващите протоколи се вижда, че те не отговарят на изискваните спецификации за обслужване.

Въведен е за първи път през 1974 г. от V. Cerf и Kahn в ARPANET - първата компютърна мрежа, която прераства в Internet. Целта е била да позволи свързването на различни мрежи, да бъде жизнеспособна и гъвкава, да оцелее и в условията на ядрен апокалипсис.

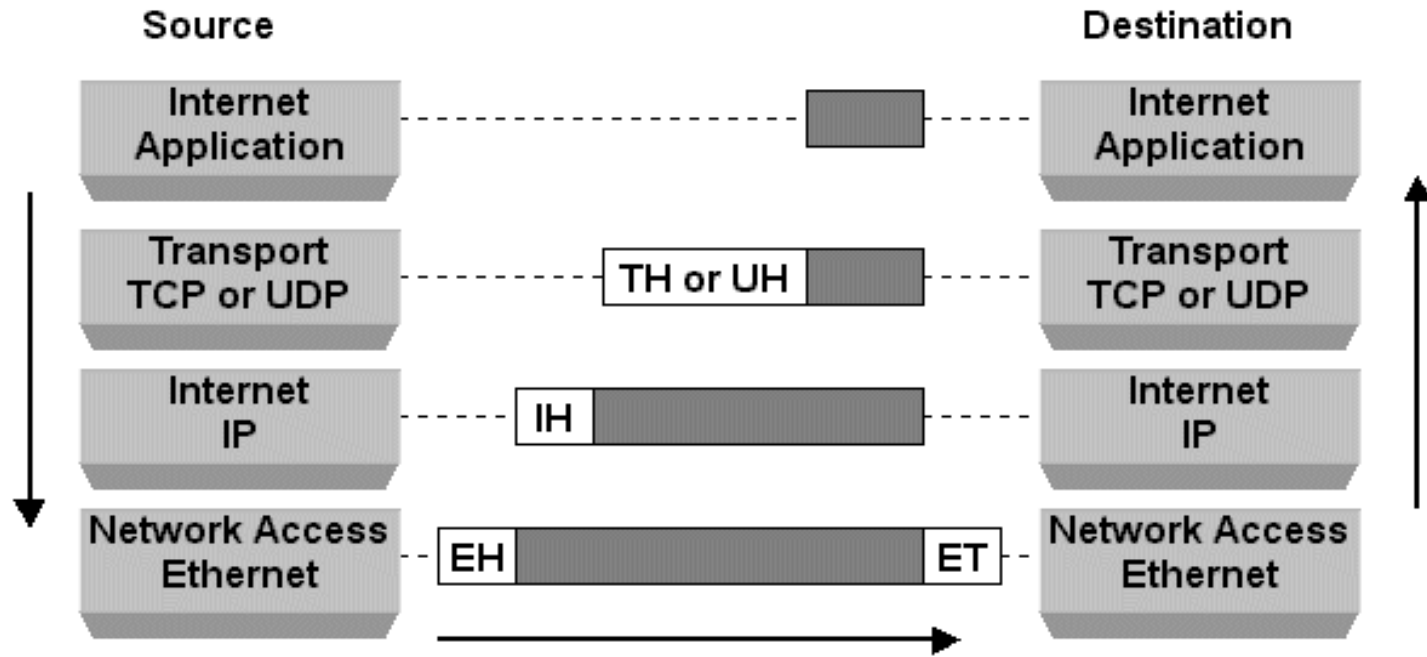
Мрежа с комутация на пакети, базирана на обслужване с неустановена връзка (connectionless - без предварително уговаряне на параметрите на връзката между източник и приемник).

Това е мрежовото ниво – Интернет, където имаме “best effort delivery”. ІР протокол, ІР пакети.

# OSI vs. TCP/IP

OSI Model	TCP/IP Model (DoD Model)	TCP/IP – Internet Protocol Suite
Application	Application	Telnet, SMTP, POP3, FTP, NNTP, HTTP, SNMP, DNS, SSH, ...
Presentation		
Session		
Transport	Transport	TCP, UDP
Network	Internet	IP, ICMP, ARP, DHCP
Data Link	Network Access	Ethernet, PPP, ADSL
Physical		

# TCP/IP – мрежи, протоколи и услуги



Както Интернет, така и транспортният слой е подобен на OSI.

**TCP** (Transmission Control Protocol) е connection-oriented. Потокът от байтове да бъде доставен без грешка. Съобщението се разбива на сегменти.

**UDP** (User Datagram Protocol) е connectionless за обмен на звук, къси съобщения: NTP, TFTP, SNMP.



# Сравнение на OSI и TCP/IP

Общи свойства: единен стек от независими протоколи, подобни функции.

Три основни свойства на OSI:

- Дефиниране на услуги
- Дефиниране на интерфейси
- Дефиниране на протоколи

Основно предимство на OSI: прави разграничение между тези три свойства.

TCP/IP няма точно разграничение между трите.

Протоколите в OSI са по-добре обособени, отколкото в TCP/IP.  
Могат да бъдат заменяни по-лесно.

# Сравнение на OSI и TCP/IP

**OSI** – преди да е създадена концепцията за протоколите –  
достатъчно общ.

Липса на опит с конкретни обекти – недостатъчна функционалност

Канален слой – за връзки “точка-точка”. С поява на LAN – broadcast мрежите – нов подслой.

Подслоевите да бъдат изменяни в зависимост от различията в конкретните мрежи.

OSI създателите – всяка страна по една OSI мрежа под управлението на правителството. Не е мислено за междумрежово свързване.

**TCP/IP** – първо се разработват протоколите. Моделът – реално описание на вече съществуващи протоколи. Т.е. пасват перфектно, без да е необходима да са напасвани към модела, както при OSI.

# Сравнение на OSI и TCP/IP

TCP/IP не е приложим за описание на мрежи, които не поддържат TCP/IP. Но днес всички производители го поддържат. Такива със собствени протоколни стекове. Novell се отказа от SPX/IPX, Apple – от AppleTalk, Microsoft – от NetBIOS и др.

Т.е TCP/IP стана световен мрежов стандарт.

Други разлики:

- На мрежово ниво - TCP/IP само connectionless; OSI – и connection oriented.
- На транспортно ниво - OSI – само connection oriented; TCP/IP – и двете (TCP и UDP).

# Идва ли краят на ТСП/ІР?

Internet може да стане по-бърза и по-сигурна, като се изостави концепцията за пакети и корекцията на грешките, която забавя трафика заради повторните предавания.

Това предполагат учени от Aalborg University, Дания, в сътрудничество с MIT и Caltech (California Institute of Technology).

Те искат да заменят сегашния модел със система от линейни уравнения.

# Методиката

Методиката се базира на мрежовото кодиране и декодиране, по-точно RLNC (**Random Linear Network Coding** – Случайно линейно мрежово кодиране)

Все едно, колите навлизат в кръстовище от всички посоки, без да се налага да се изчакват помежду си или да чакат да им светне “зелено”.

Естествено, без да стават катастрофи :)

4-минутно видео се “сваля” 5 пъти по-бързо отклкото по традиционната технология.

# Как работи

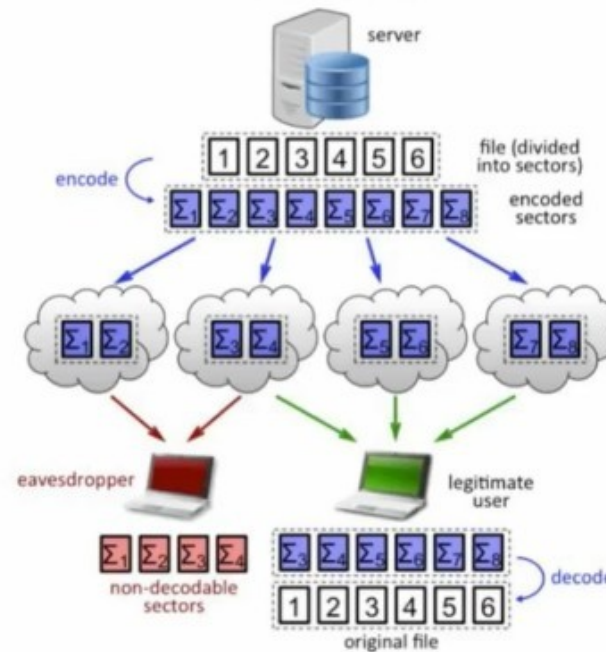
На съдържанието на пакета се гледа като на отделно число. Всеки възел в мрежта създава система от линейни уравнения от числата, извлечени от съдържанието на пакетите и множество от случайно генерирани коефициенти.

Ако си спомняте от гимназиалния курс по математика, необходими са ви  $N$  линейни уравнения, за да намерите стойностите на  $N$  неизвестни.

Всеки кодиран пакет съдържа по едно уравнение. Т.е на получателят му трябва  $N$  пакета (с различни коефициенти), за да може да декодира данните.

По-сигурна система. На “подслушвачите” ще им трябва да прихванат всички пакети, за да декодират информацията

## Coding as a Measure Content of Protection



# Противници

На нас ни предстои да се занимаваме с модела ТСР/IP. Рано е да се каже, че е за изхвърляне. Трябва да отбележим, че:

Пакетите не е задължително да бъдат подредени. Протоколът ТСР няма такова изискване. Сегментите, на които се разделя дадено съобщение, се номерират.

Благодарение на прозоречни механизми и др. техники, повторните предавания (ако се наложи) не забавят скоростта.

Нека имаме предвид и високото бързодействие на днешните интегрални схеми.

Все пак предлаганата технология сигурно ще намери приложение в **5G** мобилни мрежи, сателитни комуникации и **Internet of Things**.



# Вече в Силиконовата долина

RLNC технологията е патентована и “опакована” в C++ софтуер от фирмата Steinwurf с марката Kodo. Steinwurf планира да я продава на хардуерни производители.

Steinwurf е основана от професор Frank Fitzek от Aalborg University и двама негови бивши студенти заедно с американските им колеги.

Компанията вече има офис и в Силиконовата долина, но управлението ѝ е все още в Aalborg.