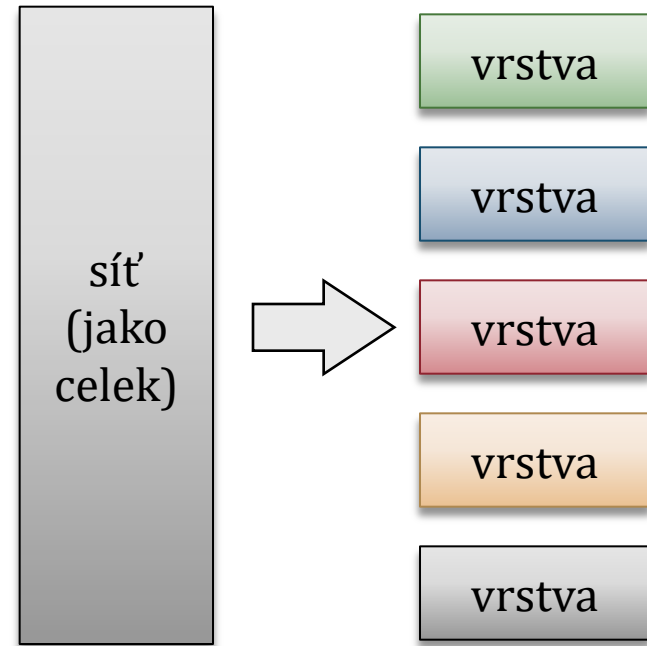


# Lekce 3: Vrstvy a vrstevové modely

*Jiří Peterka*

# proč vrstvy?

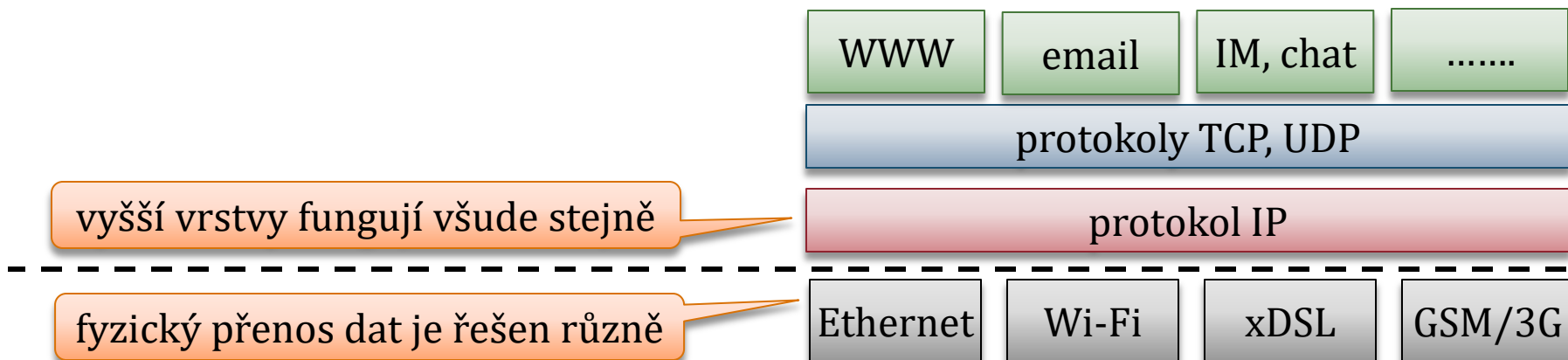
- **implementovat funkční síť je hodně složité a náročné**
  - je to stejná situace, jako při řešení velkých SW celků
- **jde o jeden velký problém, který se vyplatí dekomponovat**
  - rozdělit na menší části, které je možné řešit samostatně
- **podmínka:**
  - musí se přesně (a vhodně) specifikovat rozhraní mezi vrstvami
    - aby se každá vrstva dala řešit (navrhnout a implementovat) samostatně
      - jako samostatný úkol
    - aby řešení jedné vrstvy nebylo závislé na řešení jiné vrstvy
  - **výhoda, přínos:**
    - jednotlivé vrstvy se dají řešit alternativně
      - mohou existovat různé způsoby řešení téže vrstvy



- **musí se vyřešit otázky jako:**
  - kolik má být vrstev
  - co má která vrstva dělat
  - jak mají vrstvy spolupracovat
    - vertikálně (v rámci uzlu)
    - horizontálně (mezi uzly)
  - .....

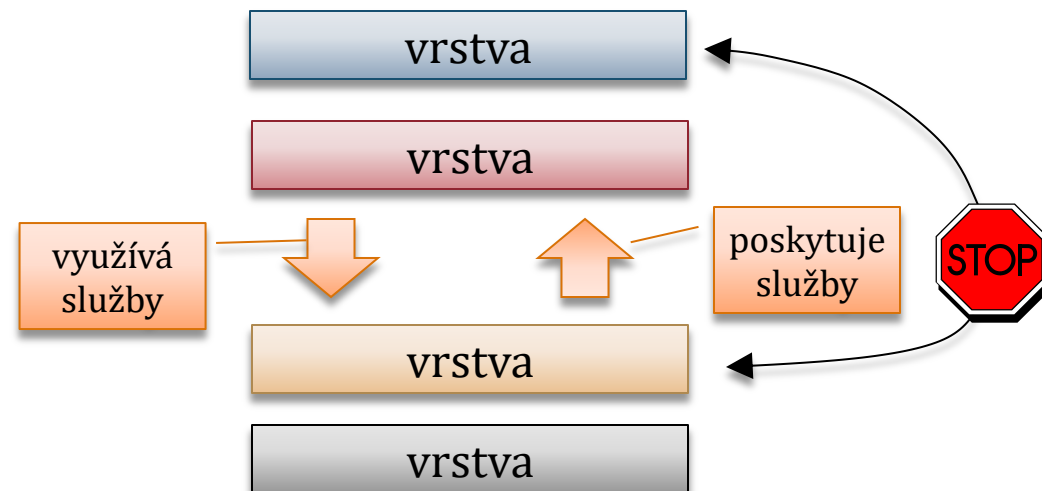
# proč vrstvy (u sítí)?

- **vyhovuje to povaze řešeného problému:**
    - je zapotřebí řešit dílčí úkoly, které odpovídají různým úrovním abstrakce
      - jsou různě detailní
        - např. přenos bitů vs. celých bloků
  - **princip:**
    - stejná úroveň abstrakce = stejná vrstva
      - neboli: úkoly na stejné úrovni abstrakce se řeší v rámci stejné vrstvy
  - **řešení:**
    - dekompozice se provede po hierarchicky uspořádaných vrstvách
- **je to modulární**
    - každou vrstvu lze řešit samostatně
      - odděleně od řešení jiné vrstvy
    - stejná vrstva může fungovat různými způsoby
      - např. dle přenosového média
  - **důsledek:**
    - vyšší vrstvy nemusí měnit své fungování v závislosti na tom, jak fungují nižší vrstvy
      - nižší vrstvy mohou v různých částech sítě fungovat různě, zatímco vyšší vrstvy mohou být všude stejné



# vertikální komunikace mezi vrstvami

- **každá vrstva poskytuje určité služby**
  - **například:**
    - **nejnižší (fyzická, L1) vrstva poskytuje služby charakteru: odešli bit a přijmi bit**
- **k realizaci svých služeb může každá vrstva využívat služby vrstvy bezprostředně nižší**
  - **příklad: druhá nejnižší (linková, L2) vrstva přenáší celé bloky dat (linkové rámce)**
    - **ale pro přenos jednotlivých bitů využívá služby „odešli bit“ a „přijmi bit“ vrstvy bezprostředně nižší (vrstvy fyzické, L1)**
      - **a sama zajišťuje „všechno ostatní ...“**
- **princip vertikální komunikace:**
  - **nižší vrstva poskytuje (nabízí) své služby vrstvě bezprostředně vyšší**
  - **vyšší vrstva využívá služby vrstvy bezprostředně nižší**
  - **dvě vrstvy, které spolu nesousedí, vzájemně nekomunikují**



# horizontální komunikace mezi vrstvami

- **týká se:**

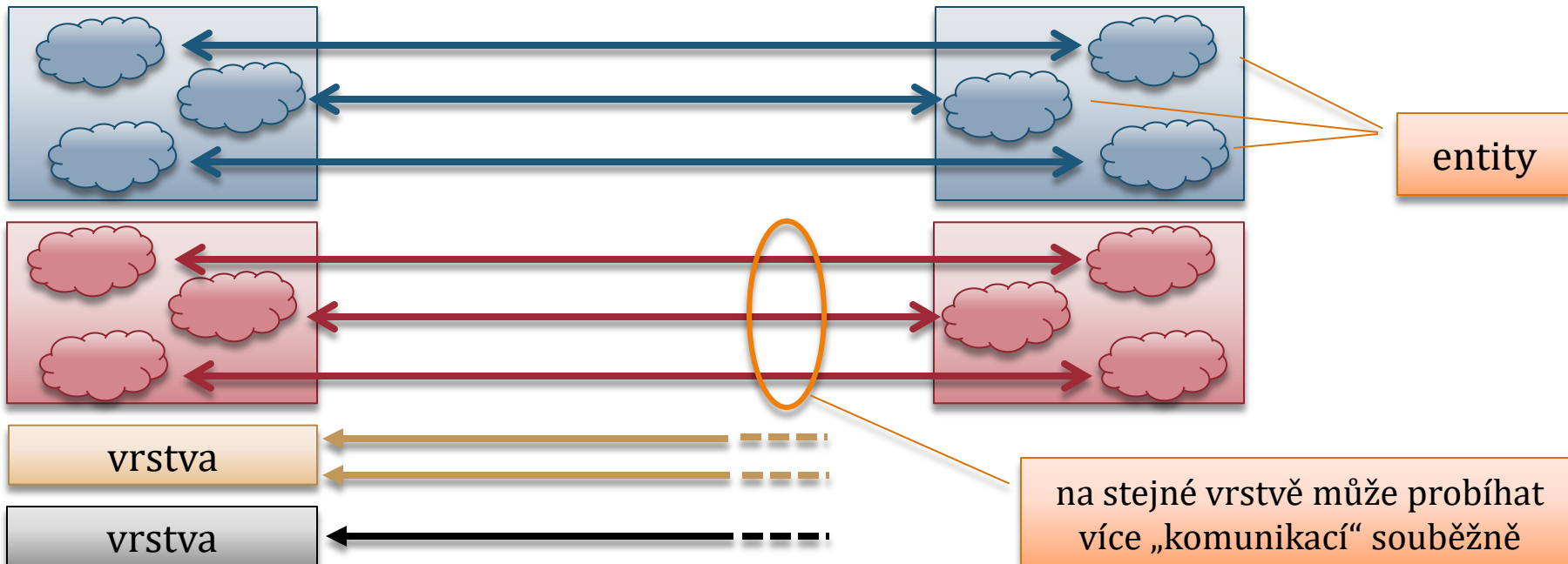
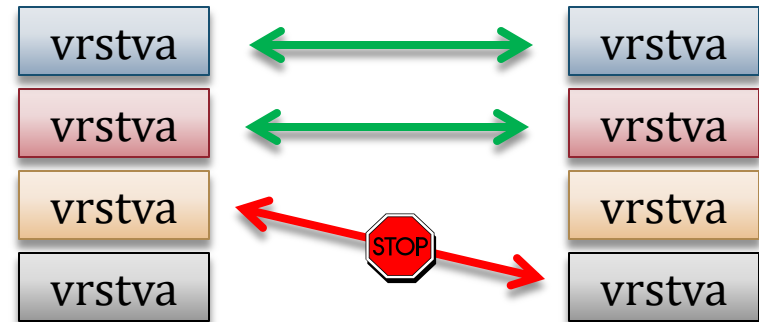
- komunikace mezi různými uzly v síti

- **princip:**

- vždy komunikují jen vrstvy na stejné úrovni
  - nikoli vrstvy na různých úrovních !!

- **upřesnění**

- nekomunikují spolu vrstvy jako takové, ale entity v rámci (stejnolehlých) vrstev
  - entity mohou být: procesy, úlohy, programy ....



- **co je nutné definovat?**
  - entity (procesy, úlohy, ...) na jednotlivých uzlech musí znát pravidla komunikace s jinými entitami na stejné vrstvě
    - musí je znát dopředu
    - tato pravidla musí být nezávislá na způsobu implementace entit
      - i na systémovém prostředí, ve kterém entity fungují (na operačním systému, ....)
- **takováto pravidla definují protokoly**
- **protokol**
  - vždy „patří“ do konkrétní vrstvy
    - nikdy nepokrývá více vrstev
  - do stejné vrstvy může „patřit“ více protokolů
    - které mohou být vzájemně alternativní či komplementární
      - mohou plnit stejné úkoly
      - mohou plnit různé úkoly
  - v rámci jedné vrstvy může být využíváno více protokolů současně
- **příklady (TCP/IP):**
  - zařazení do vrstev:
    - IP: „patří“ do síťové vrstvy
    - TCP, UDP: „patří“ do transportní vrstvy
    - SMTP, HTTP, FTP: „patří“ do aplikační vrstvy
  - více protokolů ve stejné vrstvě:
    - TCP a UDP plní stejné úkoly
      - ale jiným způsobem
    - SMTP, HTTP, FTP plní různé úkoly
      - využívají je jiné aplikace
    - uživatel využívá více aplikací a jejich služeb současně

# co definují protokoly?

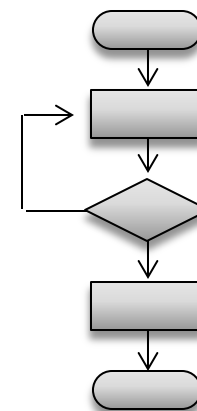
- **protokoly musí (jednoznačně) definovat:**

- **postupy vzájemné komunikace**

- co a jak má dělat každá ze stran v různých situacích, které mohou nastat
  - lze popsat např. stavovým diagramem, slovním popisem, ...

- **formáty dat, které si komunikující strany vzájemně předávají**

- jaký je formát bloků dat (rámců, paketů, buněk, ...)
- jaký je význam jednotlivých částí bloků
  - jaké údaje jsou v hlavičce bloků, jaké jsou adresy a co znamenají



- **obecný formát dat:**

- **blok dat (rámec, paket, buňka, ...) je rozdělen na:**

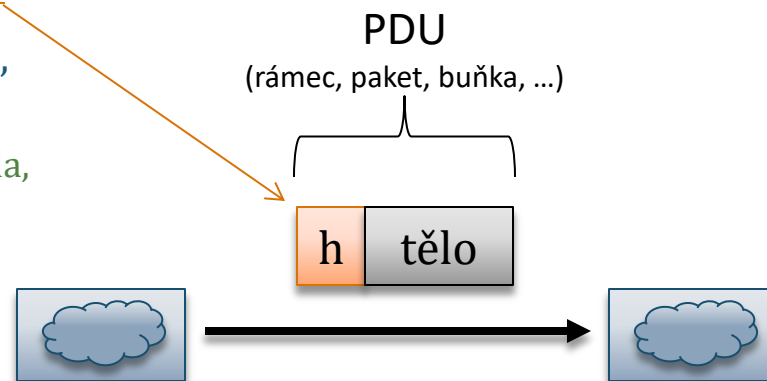
- **hlavičku (header)**

- obsahuje „režijní“ údaje: o odesilateli a příjemci, typu bloku, druhu nákladu, ....

- jsou to informace určené pro protistranu, aby věděla, co a jak má s přijatým blokem udělat

- **tělo (body, payload)**

- obsahuje „náklad“
  - „užitečná“ (přenášená) data



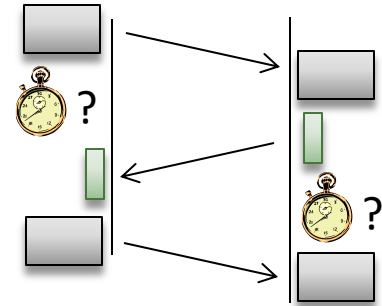
- **PDU (Protocolar Data Unit)**

- **obecné označení pro přenášený blok dat, doslova: blok dat „patřící protokolu“**

# jak fungují (síťové) protokoly?

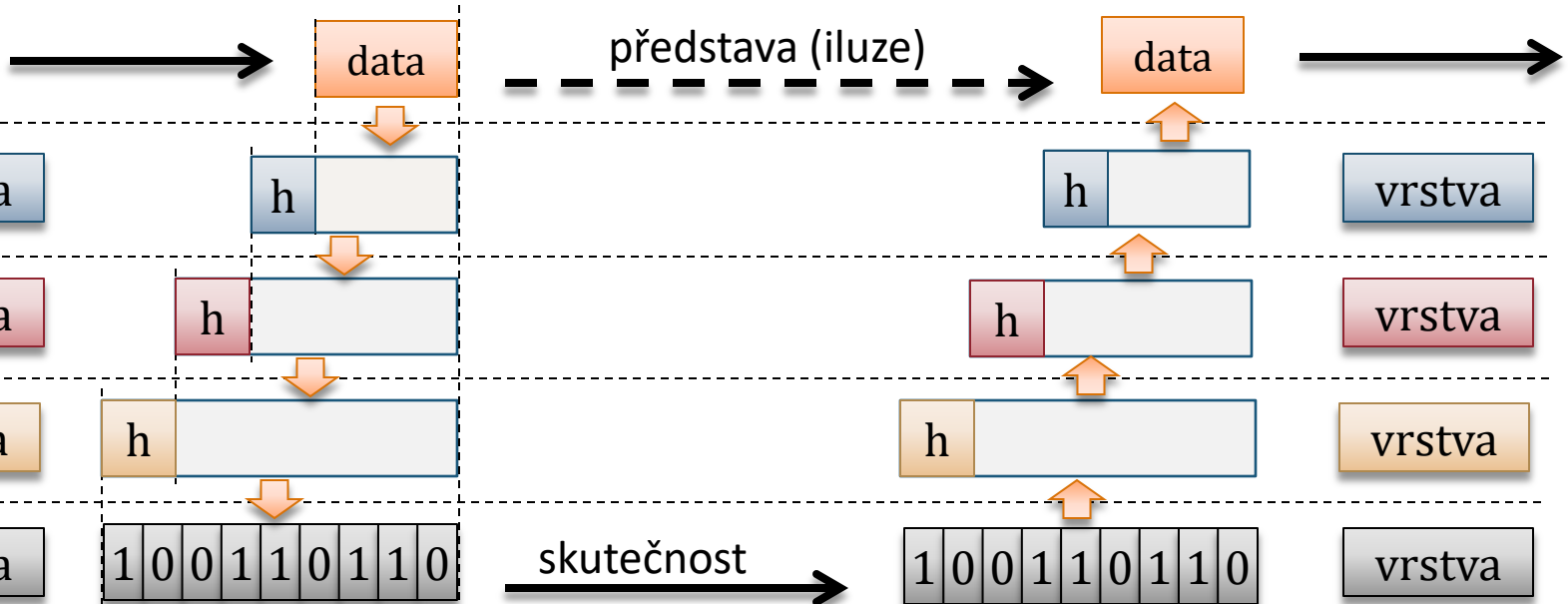
- **asynchronně**

- na principu: „pošli zprávu“ (zde: blok PDU) a čekej na odpověď
  - a pak, až odpověď přijde, pokračuj dál



- **virtuálně**

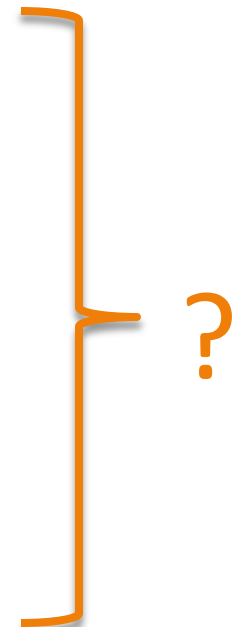
- pracují s představou (iluzí), že své PDU posílají své protistraně přímo
- ve skutečnosti:
  - odesílatel (entita dané vrstvy) připraví svůj blok dat (PDU) pro příjemce, ale předá jej k odeslání své bezprostředně nižší vrstvě
  - bezprostředně nižší vrstva jej vloží do svého PDU a předá k odeslání své nižší vrstvě
    - skutečně odesílá pouze nejnižší (fyzická) vrstva, která přenáší jednotlivé bity





# síťová architektura, síťový model

- **kolik by mělo být vrstev?**
  - na kolik hierarchických vrstev rozdělit (dekomponovat) celou síť?
- **co by jednotlivé vrstvy měly dělat?**
  - jaké úkoly by měly plnit?
    - které činnosti se mají vykonávat a na které vrstvě?
- **jaké protokoly používat?**
  - jakým způsobem by měly fungovat?
    - například: spojovaně/nespojovaně, spolehlivě/nespolehlivě, best effort/QoS?



- **„síťový světový názor“ (neformálně)**

- je celkový pohled na to, jak by měly být zodpovězeny výše uvedené otázky

- **síťová architektura**

- konkrétní představa o tom:

- kolik má být vrstev
- co má ta která vrstva dělat
- jaké protokoly má používat ta která vrstva

například TCP/IP

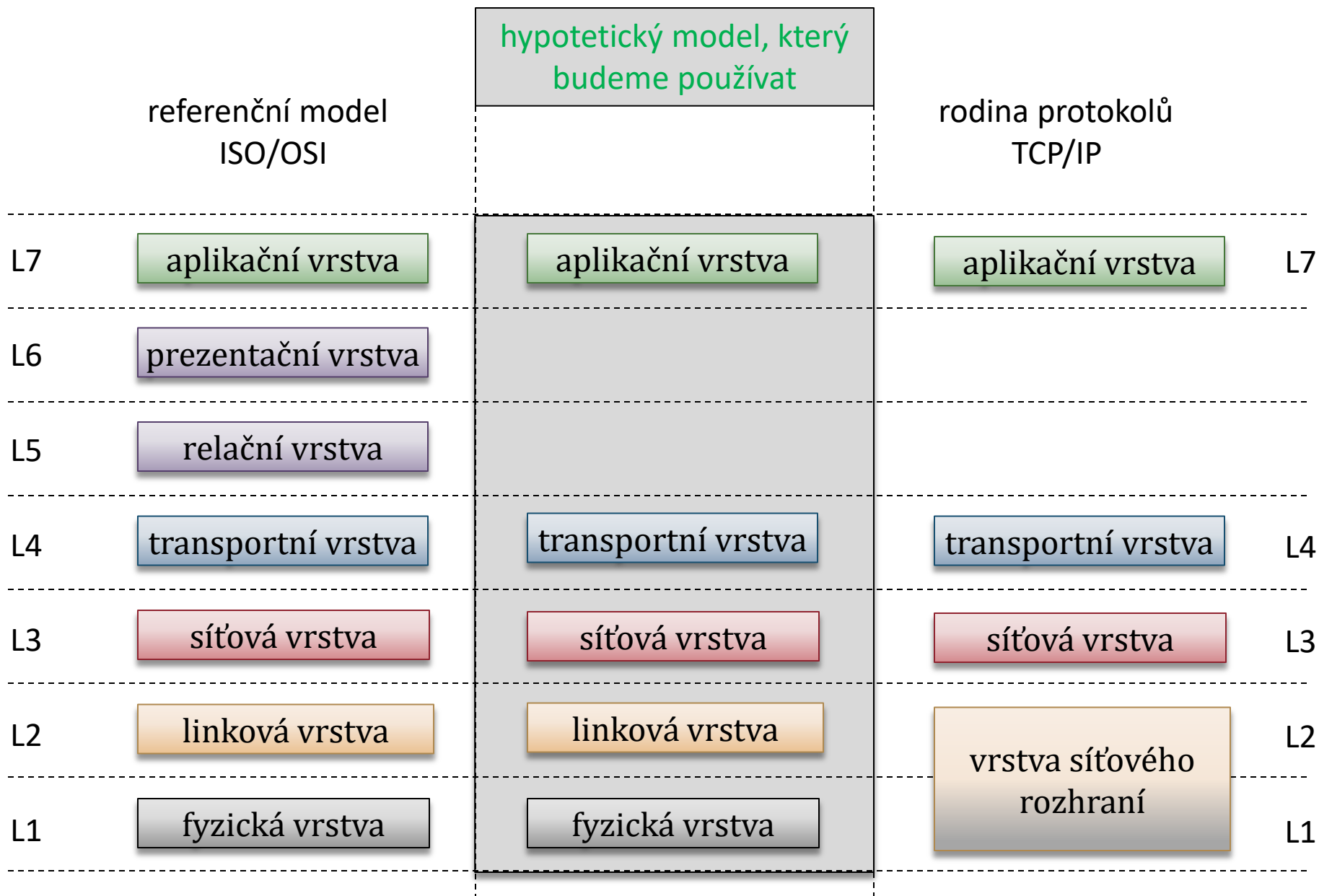
- **síťový model**

- pouze představa o vrstvách a jejich úkolech, bez protokolů

například RM  
ISO/OSI



# kolik má být vrstev?



# odkud pochází RM ISO/OSI?

- **pochází „ze světa spojů“**

- **od organizace ISO**

- International Standards Organization
  - správně: International Organization for Standardization
- v češtině: Mezinárodní organizace pro normalizaci



- **členy ISO jsou národní normalizační instituce**

- za ČR původně organizace ČSNÍ
- dnes: ÚNKM (Úřad pro technickou normalizaci, metrologii a státní zkušebnictví)



- **RM ISO/OSI reagoval na vznik proprietárních síťových architektur**

- jako je SNA (Systems Network Architecture od IBM)
- nebo DECNET (od firmy DEC/Digital)

- **RM ISO/OSI měl být „oficiálním řešením“**

- řešením, které prosazovaly "orgány státu" a chtěly jej nasadit do praxe
  - pokud budovaly nějaké sítě pro potřeby veřejné správy
- byl „megalomanským“ řešením, vznikajícím od zeleného stolu
  - chtěl být maximalistický: “umět všechno” – nakonec se v praxi nedal použít
- dnes je RM ISO/OSI pro praxi „odepsaný“, prohrál v souboji s TCP/IP
  - dostupné síťové technologie jsou založeny především na TCP/IP



# geneze RM ISO/OSI

## • geneze:

– prvotní záměr: definovat, jak mají vypadat otevřené systémy

- tj. definovat jejich chování jak „uvnitř“, tak i „mezi sebou“
- odsud: **Open Systems Architecture**

– ukázalo se jako příliš náročné, dochází k redukci ambic

– revidovaný záměr: definovat pouze vzájemné propojení otevřených systémů

- změna názvu: **Open Systems Interconnection Architecture**
- opět se ukázalo jako příliš náročné, nedalo se stihnout

– nakonec: nebude to obsahovat konkrétní protokoly

- ale jen představu o počtu vrstev a o tom, co má která vrstva dělat
- aby se to „vůbec stihlo“
- změna názvu: **Open Systems Interconnection** (OSI, ISO/OSI)

## • proč Referenční model (RM)?

– fakticky jde o síťový model

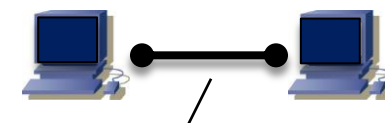
- pouze představa o vrstvách a jejich úkolech, bez konkrétních protokolů

– jednotlivé protokoly vznikaly dodatečně (a postupně)

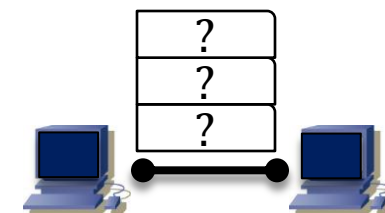
- ale moc úspěšné nebyly a v praxi se moc nepoužívaly
- nevznikly všechny, částečné využití našly jen protokoly:
  - X.400 (pro elektronickou poštu)
  - X.500 (pro adresářovou službu – základem pro protokol LDAP z TCP/IP)



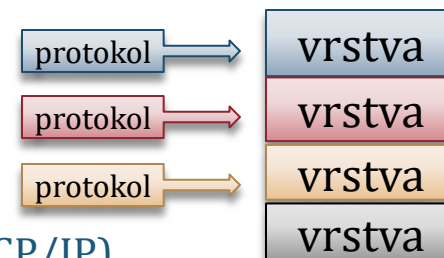
jak funguje propojení



jak funguje propojení



RM ISO/OSI

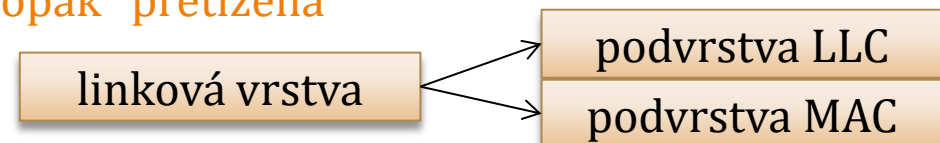
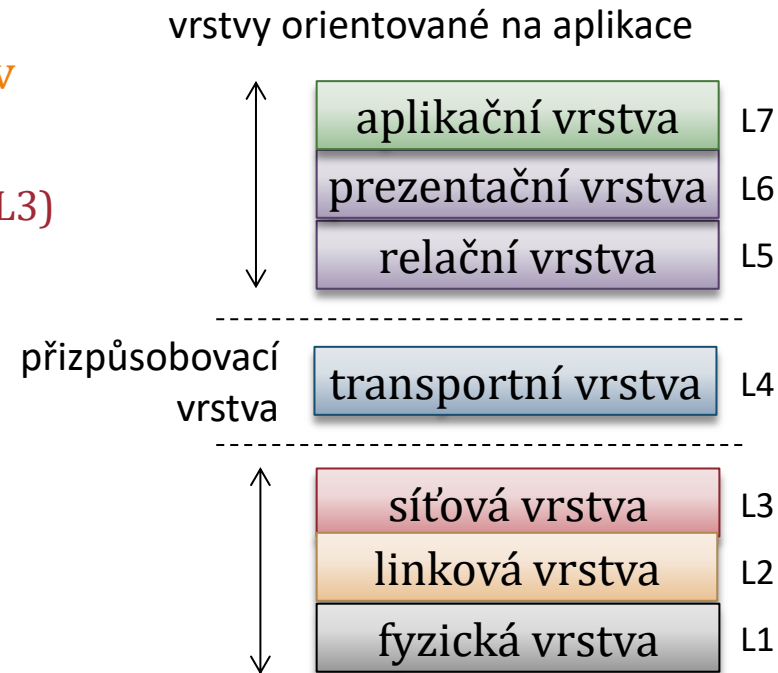


# filosofie RM ISO/OSI


- **vznikl ve světě spojů**
  - je příliš prodchnut „světonázorem“ lidí od spojů, ale měl sloužit „lidem od počítačů“
  - vzniká „od zeleného stolu“, nerespektuje požadavky a realitu běžné praxe
    - nejprve se „u zeleného stolu“ sepisuje, co všechno by měl nový standard zahrnovat
      - pak se standard vydá jako závazný
    - a teprve pak se řeší, zda je možné standard implementovat
  - je příliš složitý, těžkopádný a obtížně implementovatelný
    - je příliš maximalistický
    - chce nejprve všechno
      - a pak musí redukovat
        - hledat takovou podmnožinu, která by byla reálně implementovatelná
- **preference:**
  - jednoznačně upřednostňuje spolehlivé a spojované přenosové služby
    - motivováno „světonázorem“ lidí od spojů
      - „kdo by si kupoval nespolehlivou přenosovou službu?“
  - některé činnosti (funkce) zbytečně opakuje na každé vrstvě
    - například zajištění spolehlivosti, které chce na více vrstvách současně
  - počítá spíše s rozlehlými sítěmi než se sítěmi lokálními
    - autoři předpokládali propojení uzlů výhradně pomocí dvoubodových spojení
    - linková vrstva původně neřešila sdílený přístup ke sdílenému médiu
      - důsledek: došlo k rozdělení linkové vrstvy na dvě podvrstvy

# sedm vrstev ISO/OSI

- autoři ISO/OSI se dost dlouho přeli o počet vrstev
- kritéria pro volbu vrstev:
  - činnosti na stejném stupni abstrakce mají patřit do stejné vrstvy
  - odlišné funkce by měly patřit do odlišných vrstev
  - aby bylo možné převzít již existující standardy
    - zejména standard X.25 pro veřejné datové sítě (L1-L3)
  - aby datové toky mezi vrstvami byly co nejmenší
  - aby vrstvy byly rovnoměrně vytíženy
  - ....
- nakonec zvítězil návrh na 7 vrstev
- dnes se 7 vrstev zdá být zbytečně mnoho
  - např. rodina protokolů TCP/IP má jen 4 vrstvy
  - některé vrstvy ISO/OSI jsou "málo vytížené"
    - např. vrstva relační (L6) a prezentační (L7)
  - jedna vrstva ISO/OSI (linková, L2) je naopak "přetížená" a rozpadla se na dvě podvrstvy
    - podvrstvu LLC a podvrstvu MAC



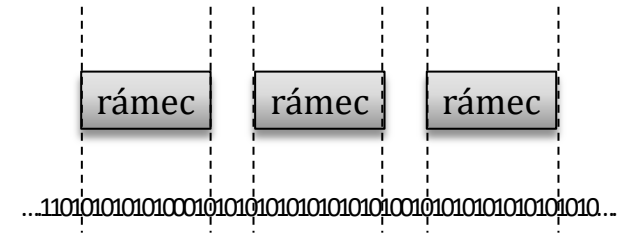
# fyzická vrstva (L1)

- **má za úkol přenášet jednotlivé bity**
    - poskytuje služby „odešli bit“ a „přijmi bit“
      - případně seskupuje bity do znaků
- 101010101 →  → 101010101
- **nijak neinterpretuje přenášené bity**
    - nerozumí tomu, co přenáší, ke všem bitům se chová stejně
  - **řeší aspekty jako je:**
    - kódování
    - modulace
    - časování
    - synchronizace
    - elektrické parametry signálů
    - konektory
    - řídicí signály rozhraní, ....
  - **pracuje s veličinami jako je:**
    - šířka pásma (bandwidth)
    - modulační rychlost (v Baudech)
    - přenosová rychlost (v bitech/s)
    - .....
  - **rozlišuje:**
    - paralelní a sériový přenos
    - synchronní, asynchronní a arytmičkový přenos
    - přenos v základním pásmu a v přeloženém pásmu
      - „modulovaný“ a „nemodulovaný“ přenos
  - **rozlišuje:**
    - bezdrátový přenos
      - v úzkém či rozprostřeném pásmu
      - v licenčním či bezlicenčním pásmu
    - „drátový“ přenos
      - pomocí optických přenosových cest
      - pomocí metalických (kovových) přenosových cest

# linková vrstva (L2)

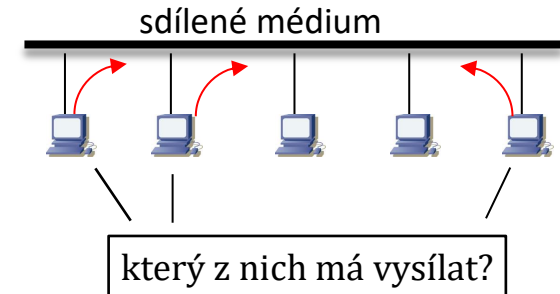
## • hlavní úkol: přenos celých bloků dat

- označovaných jako linkové **rámce** (anglicky: **frames**)
  - k tomu využívá služby fyzické vrstvy (odešli bit, přijmi bit)
  - to vyžaduje:
    - korektně rozpoznat začátek a konec rámce: tzv. **framing**
      - česky: synchronizace na úrovni rámců
      - lze řešit znakově, bitově či bytově orientovanými linkovými protokoly



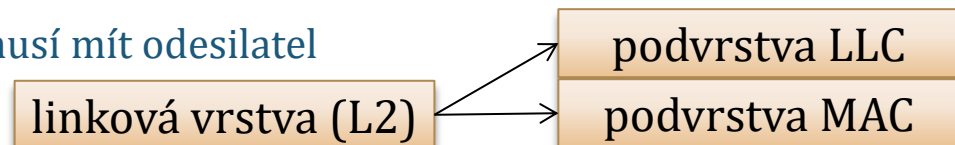
## • další úkol: řízení přístupu

- jen tam, kde je použito sdílené přenosové médium
  - více uzlů chce odesílat po jednom společném (sdíleném) médiu, ale skutečně činit tak může jen jeden
  - přístupová metoda: konkrétní způsob řízení přístupu
    - například na bázi soutěže
      - právo vysílat získá vítěz
- původní RM ISO/OSI na tento úkol nepamatoval
  - musí být řešen nad vrstvou fyzickou
    - protože sám využívá přenosu jednotlivých bitů
  - musí být vyřešen pod linkovou vrstvou
    - protože pro přenos celých bloků již musí mít odesílatel právo přístupu



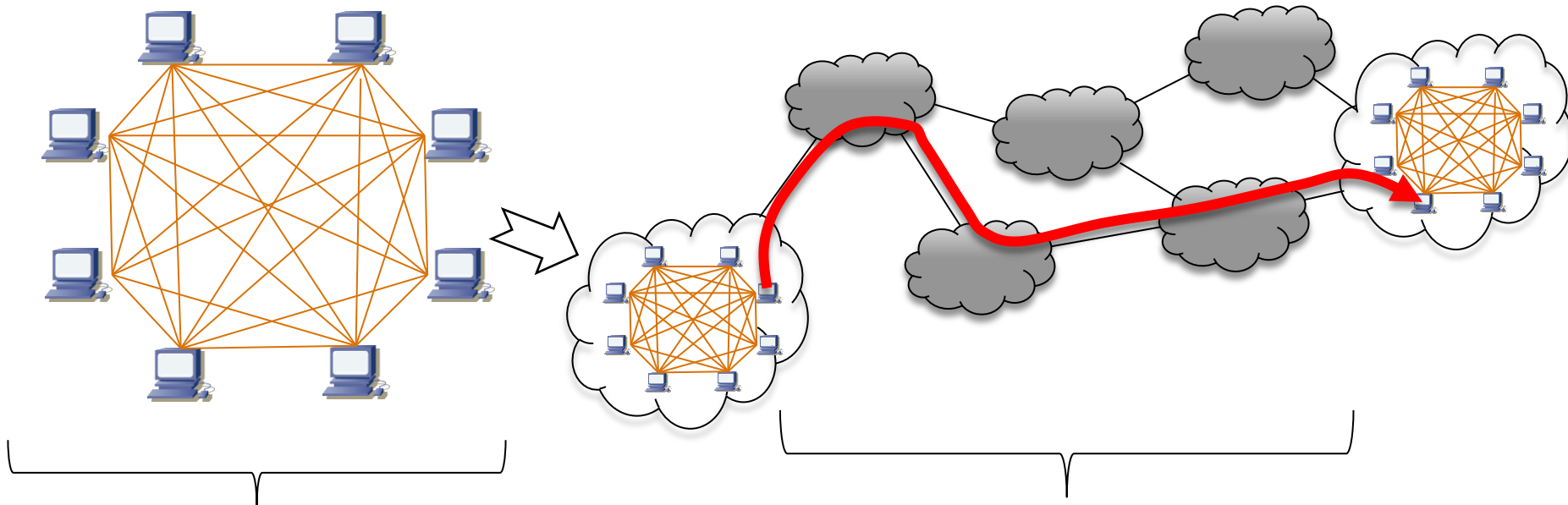
## – možné řešení:

- ~~přidat další (8.) vrstvu~~
- rozdělit linkovou vrstvu na 2 podvrstvy





# linková (L2) vs. síťová (L3) vrstva



- **představa linkové (L2) vrstvy:**

- všechny koncové uzly (v dané síti) jsou propojeny stylem „každý s každým“
- „vidí“ vždy jen danou síť
  - a nevidí její okolí / další sítě

- **proto:**

- linková vrstva neřeší hledání cesty
  - přenáší data mezi dvěma uzly, které mají mezi sebou přímé spojení

- **představa síťové (L3) vrstvy:**

- uvědomuje si existenci více sítí i způsob jejich propojení
  - zná topologii internetu
    - (jakékoli) soustavy vzájemně propojených sítí
  - neřeší topologii uvnitř jednotlivých sítí
- hledá (nejvhodnější) cestu z výchozí sítě do cílové sítě
  - tzv. **směrování (routing)**

# síťová vrstva (L3)

- **hlavní úkol síťové vrstvy:**

- dopravovat bloky dat (pakety) od jejich zdroje až k jejich cíli

- i přes mezilehlé uzly / celé sítě

- zahrnuje:

- **směrování (routing)**

- v užším slova smyslu: rozhodování o cestě / směru dalšího přenosu

- **cílené předávání (forwarding)**

- samotná manipulace s jednotlivými pakety („předání dál“ ve zvoleném směru)

- obě tyto funkce mohou být realizovány společně, v zařízení zvaném **směrovač (router)**

- ale mohou být také oddělené

- směrování (rozhodování) může být řešeno centrálně, distribuovaný je pouze forwarding



- **algoritmy směrování mohou být:**

- **statické nebo dynamické**

- podle toho, zda reagují na změny v soustavě vzájemně propojených sítí

- **izolované**

- pokud se každý směrovač rozhoduje jen podle sebe, nespolupracuje s ostatními směrovači

- **distribuované**

- pokud jednotlivé směrovače vzájemně spolupracují na hledání nejlepších cest

- **hierarchické**

- pokud je celá soustava sítí rozdělena na oblasti, které si směrování řeší samostatně

# představa fungování L2 a L3

- **odesílající uzel:**

- vloží data do síťového paketu, ten vloží do linkového rámce a odešle

například: IP paket

IP paket

Ethernet

- **směrovač:**

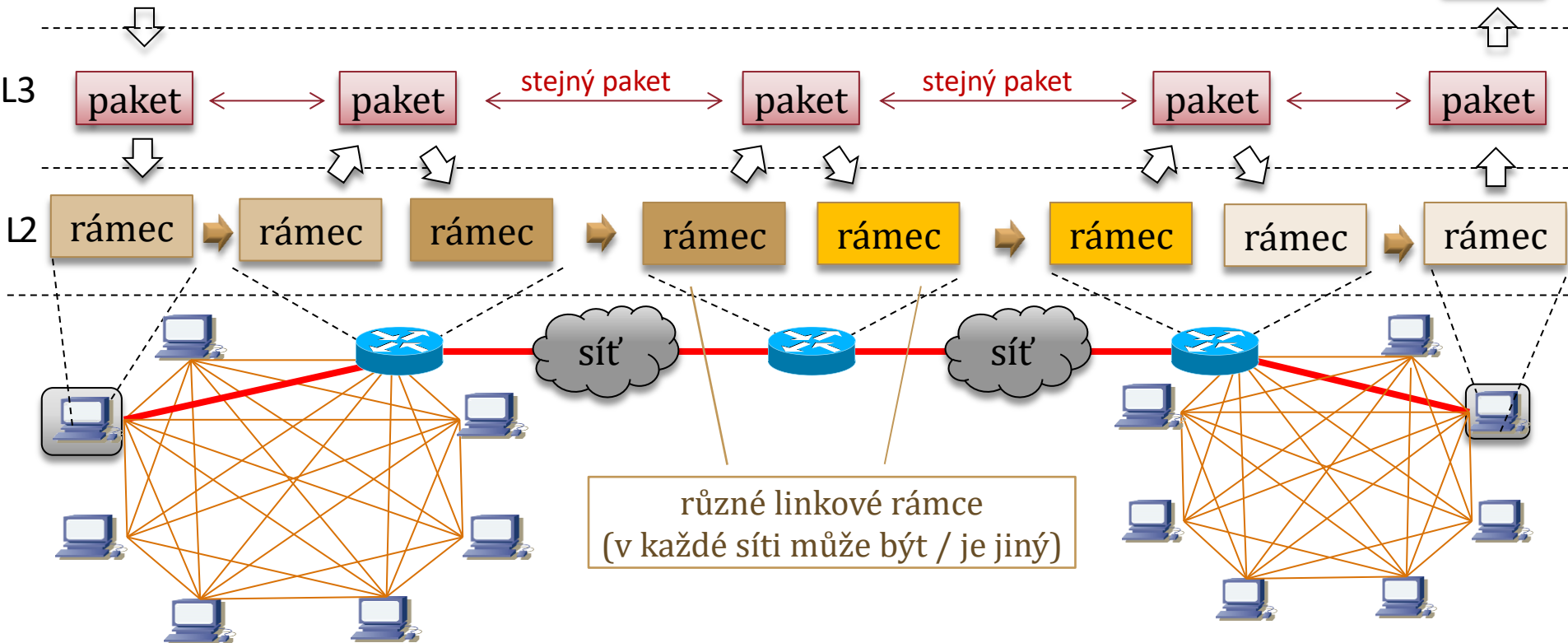
například: Ethernetového rámce

- přijme linkový rámec, „vybalí“ síťový paket, rozhodne o dalším směru přenosu paketu, vloží paket do (nového) linkového rámce a odešle jej zvoleným směrem

data

například: rámce Token Ring, PPP, SLIP, ...

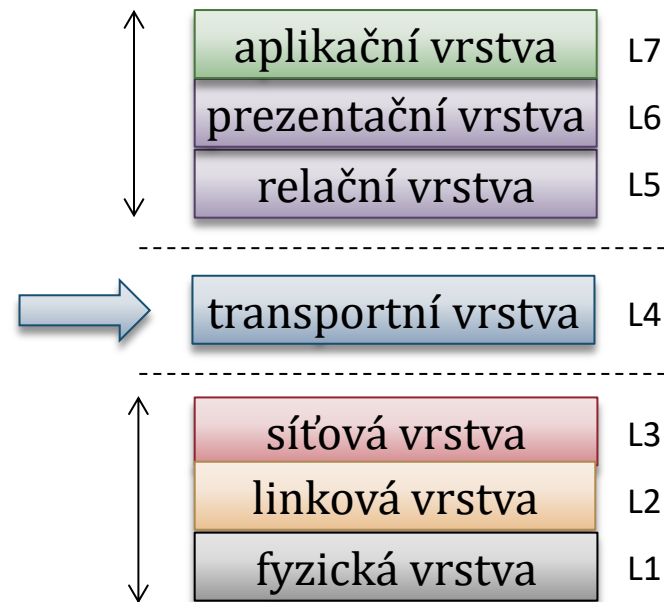
data



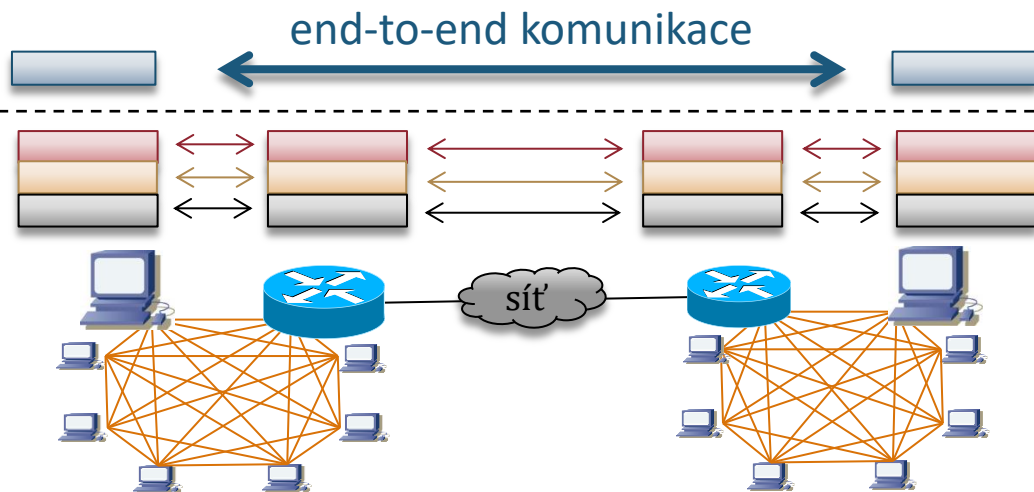
# transportní vrstva (L4)

- je to „přizpůsobovací“ vrstva:
  - přizpůsobuje představy vyšších vrstev možnostem nižších vrstev
    - například pokud jde o:
      - spojovaný či nespojovaný způsob přenosu
      - spolehlivý či nespolehlivý způsob přenosu
      - styl best effort nebo podpora QoS
      - .....
- vyskytuje se pouze v koncových uzlech
  - nikoli v mezilehlých (vnitřních) uzlech sítě
    - tedy: ve směrovačích

vrstvy orientované na aplikace



vrstvy orientované na přenos

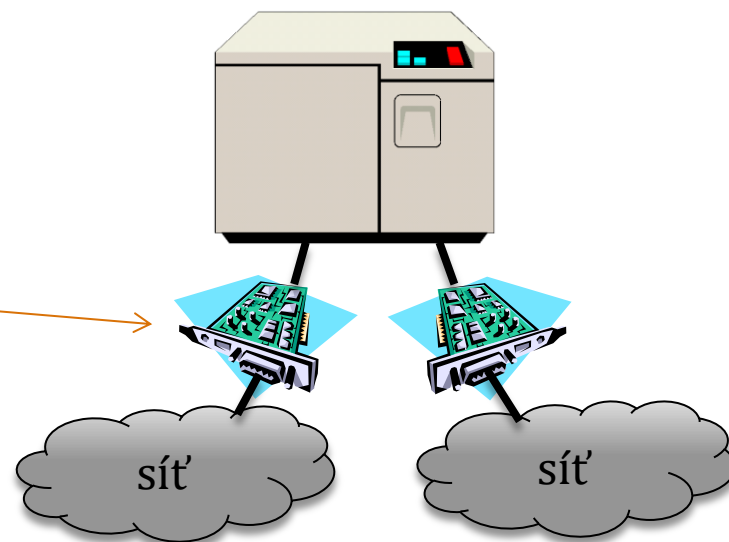


- zajišťuje vzájemnou komunikaci koncových uzlů
  - tzv. end-to-end komunikaci
    - kterou směrovače „nevidí“
      - protože nemají transportní vrstvu

# další úkol transportní vrstvy (L4)

## • nižší vrstvy (fyzická, linková, síťová):

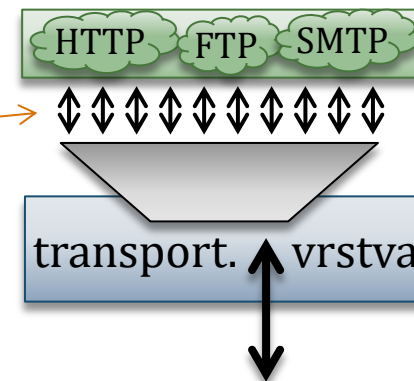
- **chápu uzly sítě vždy jako celek**
  - nijak ho nedělí
- **jejich adresy identifikují uzel jako celek**
  - přesněji: identifikují rozhraní k uzlu
    - jde o adresu síťové karty ...
    - příklad:
      - ethernetová adresa (L2), IP adresa (L3)
  - neumožňují rozlišit různé entity v rámci uzlu
    - rozlišují pouze různá rozhraní



## • transportní vrstva (L4):

- **již rozlišuje jednotlivé entity v rámci uzlu**
  - jak v roli odesílatelů dat, tak i v roli příjemců dat
- **její adresy (transportní adresy) jsou relativní**
  - mají smysl jen v rámci daného uzlu
  - v TCP/IP: tzv. **čísla portů**
- **transportní adresy musí být „všude stejné“**
  - nezávislé na implementaci konkrétního uzlu
- **transportní adresy musí být „apriorní“**
  - nesmí být závislé na „momentálním vývoji“, např. na pořadí spuštění jednotlivých entit

← tzv. multiplex / demultiplex



# relační vrstva (L5)

- **původní představa:**

- má zajišťovat „vedení relací“

- navazování, vedení a ukončování relací mezi aplikacemi (aplikačními entitami)

L5

L4

L5

L4

L5

L4

L5

L4

- vedení jedné relace pomocí více transportních spojení

- bonding, bundling: „sloučení“ více transportních spojení, s cílem dosažení vyšší přenosové kapacity,

- zajištění kontinuity: pokračování relace po výpadku transportního spojení, navázáním nového transportního spojení

- vedení více relací pomocí jednoho transportního spojení

- více po sobě jdoucích relací je vedeno po jednom transportním spojení

- snaha minimalizovat počet navázaných transportních spojení - ve veřejných datových sítích se za každé navázání spojení platí

- multiplexing: více různých relací vedeno souběžně po jednom transportním spojení

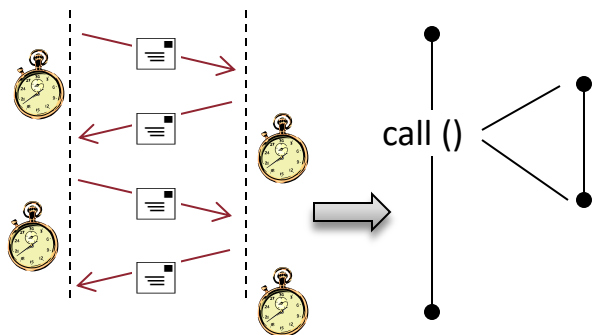
- přenos jednotlivých částí (dat) v rámci probíhající relace

- například ve smyslu toho, jak dnes v TCP/IP funguje RPC (Remote Procedure Call)

- komunikace je ve skutečnosti asynchronní, má charakter odesílání zpráv a čekání na odpovědi

- RPC fakticky mění komunikaci na synchronní: má charakter volání procedury

- bez asynchronního čekání, s transformací parametrů atd.

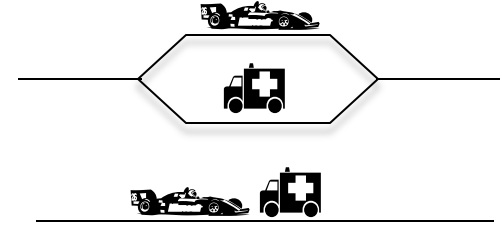


# relační vrstva (L5)

- **původní představa:**

- relační vrstva bude zajišťovat i další úkoly, například:

- synchronizaci komunikace
  - zajištění poloduplexní či plně duplexní komunikace
  - ochrana před zablokováním (deadlock, ...)
- podporu přenosů
  - checkpointing
    - nastavování „zarážek“ (bodů obnovy), od kterých je možné pokračovat v dříve přerušném přenosu
- podporu transakcí
  - např. 2fázový commit
- zabezpečení
  - zajištění identifikace a autentizace komunikujících stran
  - zajištění důvěrnosti přenášených dat (šifrováním, ...)
  - autorizace (řízení/kontrola oprávnění ke komunikaci)
- „nalezení protistrany“
  - vyhledání aktuální polohy konkrétního uzlu
    - například v tom smyslu, jak jej dnes řeší protokol SIP (Session Initiation Protocol)



- **realita:**

- relační vrstva nedělá skoro nic, často není ani implementována

- je kritizována jako nejméně „vytížená“ vrstva RM ISO/OSI



# prezentační vrstva (L6)

- nižší vrstvy (L1 až L5) přenáší data „tak jak jsou“, beze změny
  - usilují o to, aby nedošlo ke změně ani v jediném bitu

- to ale nemusí být správné a žádoucí !

- různé strany mohou „rozumět“ stejným datům různě !
  - na různých platformách mohou stejná data „vypadat“ různě
    - mohou se lišit svou hodnotou, strukturou .....



o řešení se má postarat právě prezentační vrstva, prostřednictvím **konverzí**

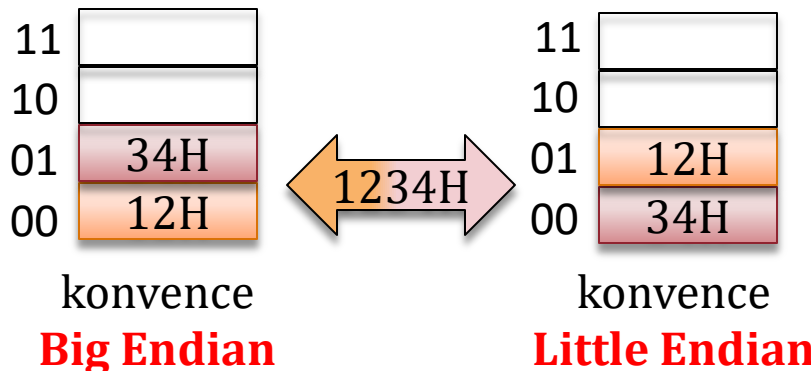
- příklady odlišností:

- kódování textu

- jednotlivé znaky lze kódovat např. v ASCII, EBCDIC, UTF-8, UTF-16 atd.

- pořadí bytů

- ve vícebytových položkách může být pořadí jednotlivých bytů různé



- formáty čísel

- v pevné i pohyblivé řádové čárce
  - různé velikosti mantisy a exponentů, různé základy, ...

- datové struktury

- pole, záznamy, fronty atd. ....
  - dají se konvertovat
- struktury provázané pointery
  - pointery nelze konvertovat
    - adresové prostory příjemce a odesilatele mohou být různé

- .....



# prezentační vrstva (L6)

- **další úkol prezentační vrstvy:**

- převést data do takové podoby, aby se dala přenést

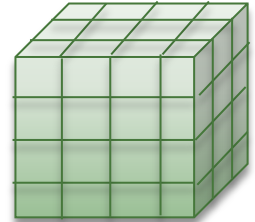
- **v čem je problém?**

- data, určená k přenosu (datové struktury) mohou být vícerozměrné

- například vícerozměrná pole, struktury provázané pointery .....

- přenosový kanál je pouze jednorozměrný

- přenáší pouze lineární data (lineární posloupnosti bitů)



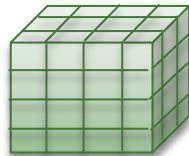
11010010101010...



- **důsledek**

- data, určená k přenosu, musí být převedena do tvaru, který je vhodný pro přenos

- **možnosti řešení:**



tzv. serializace .....



- „specifické“

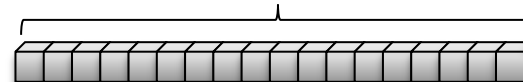
- řešení v rámci jedné aplikace (protokolu), která si vše vyřeší tak, jak považuje za vhodné
  - nelze jej ale použít univerzálně, pro jiné aplikace/protokoly

- **univerzální:**

jakási průvodka, podle ní příjemce pozná, co data znamenají

- pomocí vhodného jazyka (např. ASN.1) je popsána obecná struktura dat – jejich význam
  - abstraktní syntaxe (abstract syntax)
- pomocí vhodného kódování (např. BER, Basic Encoding Rules) je vytvořen přenosový (serializovaný) tvar dat, určených k přenosu
  - přenosová syntaxe (transfer syntax)

kódování BER



průvodka  
v ASN.1



# aplikační vrstva (L7)

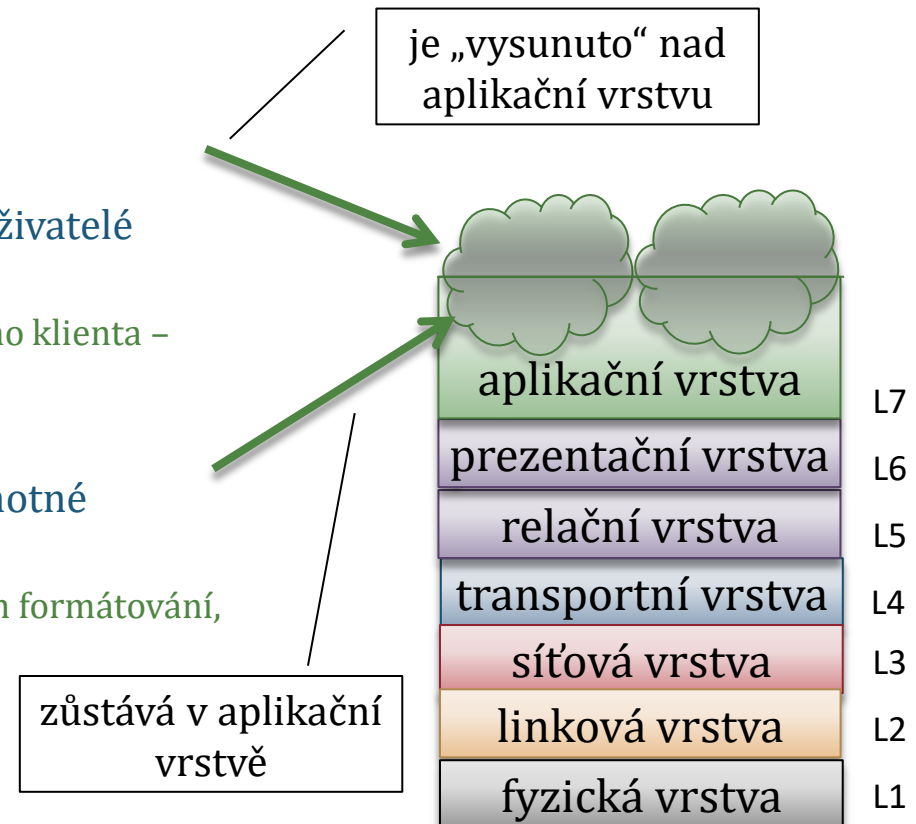
- **původní představa:**

- v aplikační vrstvě budou provozovány jednotlivé aplikace
- to se ukázalo jako nevhodné !
  - důvod: tyto aplikace by musely být standardizovány
    - nejenom jejich fungování, ale také jejich vzhled, chování pro uživatele
      - což by bylo příliš omezující pro autory aplikací, ale i pro uživatele

všechny aplikace by musely vypadat a ovládat se stejně ....

- **skutečnost:**

- aplikace jsou rozděleny na dvě části:
  - část, která nemusí být (a nemá být) stejná
    - zejména: uživatelské rozhraní – různí uživatelé mohou chtít různé rozhraní ....
      - například: uživatelské rozhraní poštovního klienta – umožňuje čtení zpráv, jejich psaní atd.
  - část, která musí být všude stejná
    - zejména: „to, co zajišťuje fungování samotné aplikace/slужby“
      - například: přenos emailových zpráv, jejich formátování, adresace, .....



# srovnání RM ISO/OSI a TCP/IP

## • RM ISO/OSI

- není síťovou architekturou
  - neobsahuje (všechny) protokoly
- vzniká ve světě spojů
  - koncepce ovlivněna filosofií a „logikou“ světa spojů
- vytvářeli jej lidé, zvyklí na to, že služby jsou někomu prodávány
  - poskytovány za úplatu
  - proto: důraz spíše na bohatší a komplexnější služby
    - viz „*chytrá síť, hloupé uzly*“
    - preference spojovaného a spolehlivého způsobu přenosu
    - spíše podpora QoS
- při vzniku menší vazba na praxi
  - více „teoretický“ přístup
    - nejdříve se vymyslí standard, pak se řeší možnost implementace
  - i proto: více vrstev (7)

## • TCP/IP

- je síťovou architekturou
  - vznikalo „obráceně“: nejprve protokoly, pak představa o vrstvách
- vzniká ve světě počítačů
  - v akademické sféře, za grantové peníze
- vytvářeli jej lidé, kteří nepotřebovali nikomu nic prodávat
  - ani poskytovat za úplatu
  - proto: důraz spíše na jednoduchost a efektivnost
    - viz „*hloupá síť, chytré uzly*“
    - preference nespojovaného a nespolehlivého přenosu
    - princip best effort
- při vzniku větší vazba na praxi
  - více „praktický“ přístup
    - nejprve se ověří možnost implementace, pak standardizace
  - i proto: méně vrstev (4)

# filosofie TCP/IP

## • obecně:

- preference nespojovaného způsobu přenosu
  - protože ten se lépe vyrovnává s výpadky sítě či změnami v topologii
- preference nespolehlivého přenosu
  - protože ne každá aplikace/služba potřebuje spolehlivost
    - například pro multimediální aplikace je důležitější pravidelnost a včasnost doručování jejich dat
      - zajišťování spolehlivosti zdržuje a zavádí nepravidelnost (když se přenos opakuje)
  - protože ne každé aplikaci vyhovuje stejná míra spolehlivosti
    - spolehlivost není absolutní (ano/ne), ale vždy jen relativní (některé chyby jsou odhaleny a napraveny, jiné nikoli)
      - některé aplikace mohou mít vyšší požadavky na míru spolehlivosti, než je ta společná
  - protože spolehlivost se snáze (levněji) zajistí v koncových uzlech, než v přenosové síti
    - zabudovat potřebnou inteligenci do koncových uzlů je jednodušší a levnější
      - jsou to „univerzální“ počítače, mají levnější HW i levnější tvorbu/úpravy SW
    - různé koncové uzly i různé aplikace si mohou zajistit různou míru spolehlivosti

dodnes se ukazuje jako správné a účelné

už tolik ne ..

- preference principu best effort (oproti podpoře QoS)
  - protože to vede na podstatně jednodušší, efektivnější (i levnější) implementaci
    - se všemi přenášenými daty je nakládáno stejně, nerozlišují se různá data

není to optimální pro multimediální služby (např. přenos hlasu a obrazu)

# vrstvy (L1 a L2) TCP/IP

## • filosofie TCP/IP:

– nevymýšlet znovu to, co už bylo vymyšleno

– konkrétně:

- pokud již existuje nějaká přenosová technologie (na úrovni fyzické a linkové vrstvy), pak ji rovnou použijme

– například: Ethernet

- a soustředíme se na to, jak ji využít co nejlépe

– například: jak co nejlépe „balit“ IP pakety do ethernetových rámců

– důsledek:

- TCP/IP nepokrývá linkovou (L2) ani fyzickou (L1) vrstvu

– v tom smyslu, že by definoval vlastní protokoly pro tyto vrstvy

- výjimka z pravidla: protokoly SLIP a PPP

- jako řešení pro dvoubodové spoje, kde i Ethernet je „overkill“

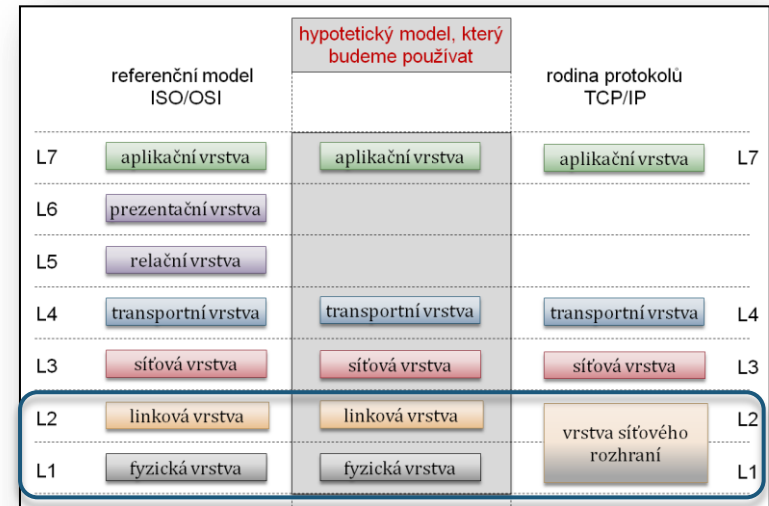
- TCP/IP nerozlišuje mezi linkovou a fyzickou vrstvou

– místo toho je „srdružuje“ do jedné vrstvy, které říká **vrstva síťového rozhraní**

- anglicky: **network interface layer**

- někdy též **network access layer**, nebo jen **link layer**

– zde se používají ona „cizí“ řešení (např. Ethernet) která nejsou součástí TCP/IP



## absence vrstev L5 a L6 v TCP/IP

- **filosofie TCP/IP:**

- nevnucovat nikomu něco, co nutně nepotřebuje

- **konkrétně:**

- služby relační a prezentační vrstvy nemusí potřebovat každá aplikace

- úvaha autorů TCP/IP:

- potřebovat je bude jen menšina aplikací

- ti, kteří je potřebují, většina si je raději (a lépe) zajistí samy, podle svých představ

- proto: nedělejme samostatnou relační a prezentační vrstvu

- protože jejich samotná existence zvyšuje režii na celkové fungování

- **důsledek:**

- TCP/IP nemá ani relační (L5), ani prezentační (L6) vrstvu

- **srovnání:**

- autoři RM ISO/OSI ve stejné situaci dospěli k závěru, že relační a prezentační služby bude potřebovat většina aplikací

- a to ve stejné podobě/provedení

- v praxi se to nepotvrdilo !

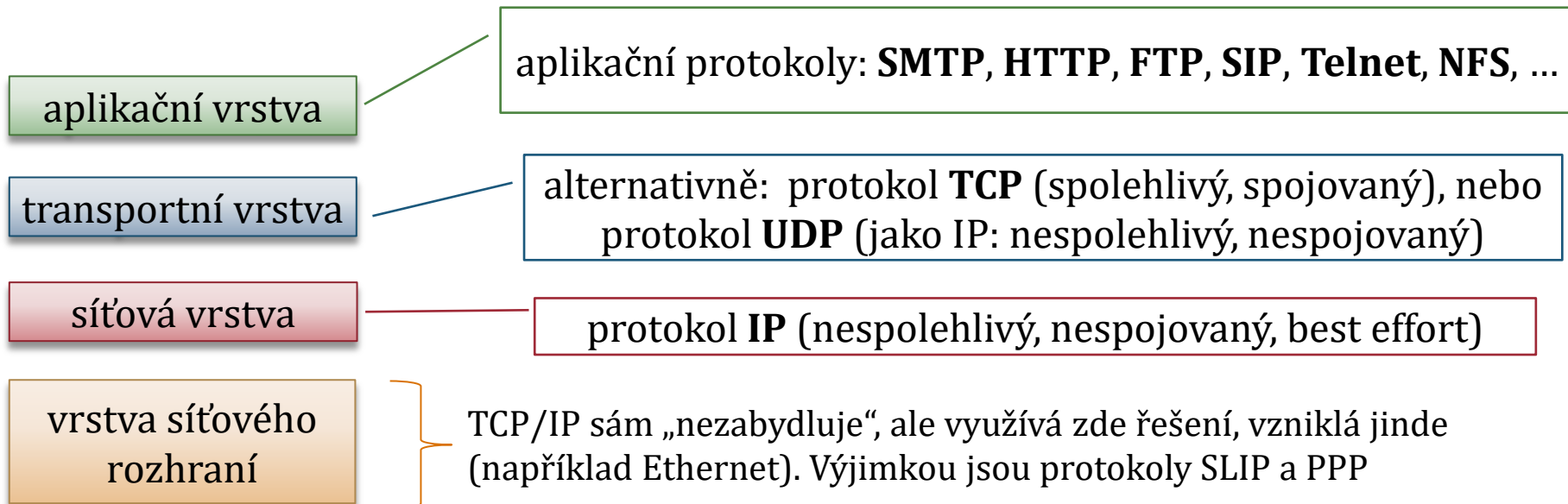
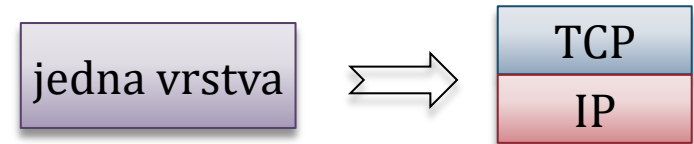
- proto zavedli samostatnou relační a prezentační vrstvu

- které jsou dnes kritizovány jako zbytečné

	referenční model ISO/OSI	hypotetický model, který budeme používat	rodina protokolů TCP/IP	
L7	aplikační vrstva	aplikační vrstva	aplikační vrstva	L7
L6	prezentační vrstva			
L5	relační vrstva			
L4	transportní vrstva	transportní vrstva	transportní vrstva	L4
L3	síťová vrstva	síťová vrstva	síťová vrstva	L3
L2	linková vrstva	linková vrstva	vrstva síťového rozhraní	L2
L1	fyzická vrstva	fyzická vrstva		L1

# vrstvy TCP/IP

- **TCP/IP má jen 4 vrstvy**
  - z nichž jednu vůbec „nezabývá“
    - vrstvu síťového rozhraní
  - a také mu to stačí !
- **role vrstev TCP/IP odpovídá jejich rolím v RM ISO/OSI**
  - liší se ale představa o tom, jak by daná vrstva měla svou roli plnit
- **proč se TCP/IP jmenuje právě TCP/IP?**
  - protože původně měl mít ještě méně vrstev
    - transportní a síťová vrstva byly spojené
  - pak došlo k rozdělení na transportní vrstvu (TCP) a síťovou vrstvu (IP)
    - ke zdůraznění tohoto rozdělení se zavedlo označení TCP/IP



# TCP/IP je více než síťová architektura

- **TCP/IP je síťovou architekturou**
  - protože zahrnuje představu o vrstvách a také o konkrétních protokolech
- **ale k TCP/IP „patří“ i další věci**
  - **standardizační proces**
    - vše kolem vzniku technických řešení, která se stávají standardy
      - orgány IETF, IAB, IANA, ICANN, .....  
které se standardizací zabývají
  - **publikační mechanismus**
    - dokumenty RFC (Request for Comment),
    - ale také: STD, FYI, BCP, Internet drafts
  - **správa jmenného prostoru**
    - koncept DNS
    - systém domén nejvyšší úrovně (TLD)
      - zřizování nových TLD
      - správa a provoz kořenových NS
    - pravidla pro správu a fungování domén nižších úrovní
  - **správa adresového prostoru**
    - koncepce IP adres
      - IPv4 i IPv6
    - pravidla „používání“ IP adres (přidělování IP adres koncovým uzlům, ...)
      - ruční konfigurace, DHCP, autokonfigurace (v IPv6), .....
    - pravidla přidělování IP adres
      - pravidla a systém distribuce IP adres
        - IANA (Internet Assigned Numbers Authority)
        - RIR (Regional Internet Registries)
        - LIR (Local Internet Registries)
    - koordinace číselných parametrů
      - tzv. dobře známé porty
        - spravuje IANA