



Katedra softwarového inženýrství,  
Matematicko-fyzikální fakulta,  
Univerzita Karlova, Praha

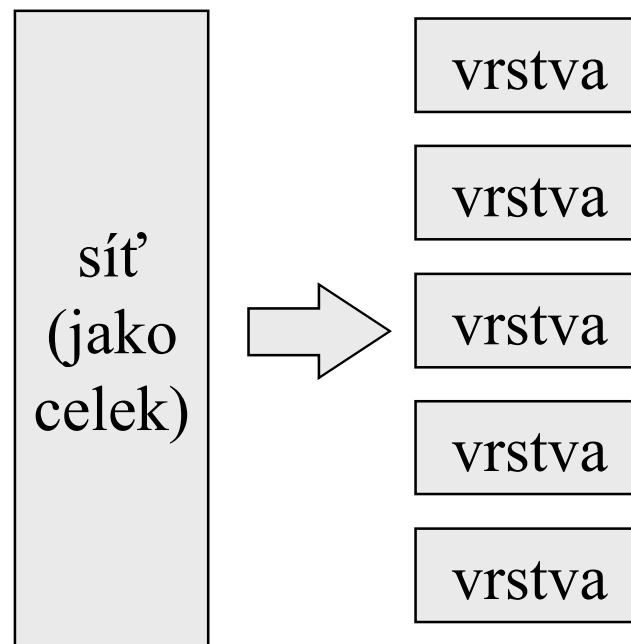


## Lekce 3: Síťové modely a architektury, RM ISO/OSI

*Jiří Peterka, 2006*

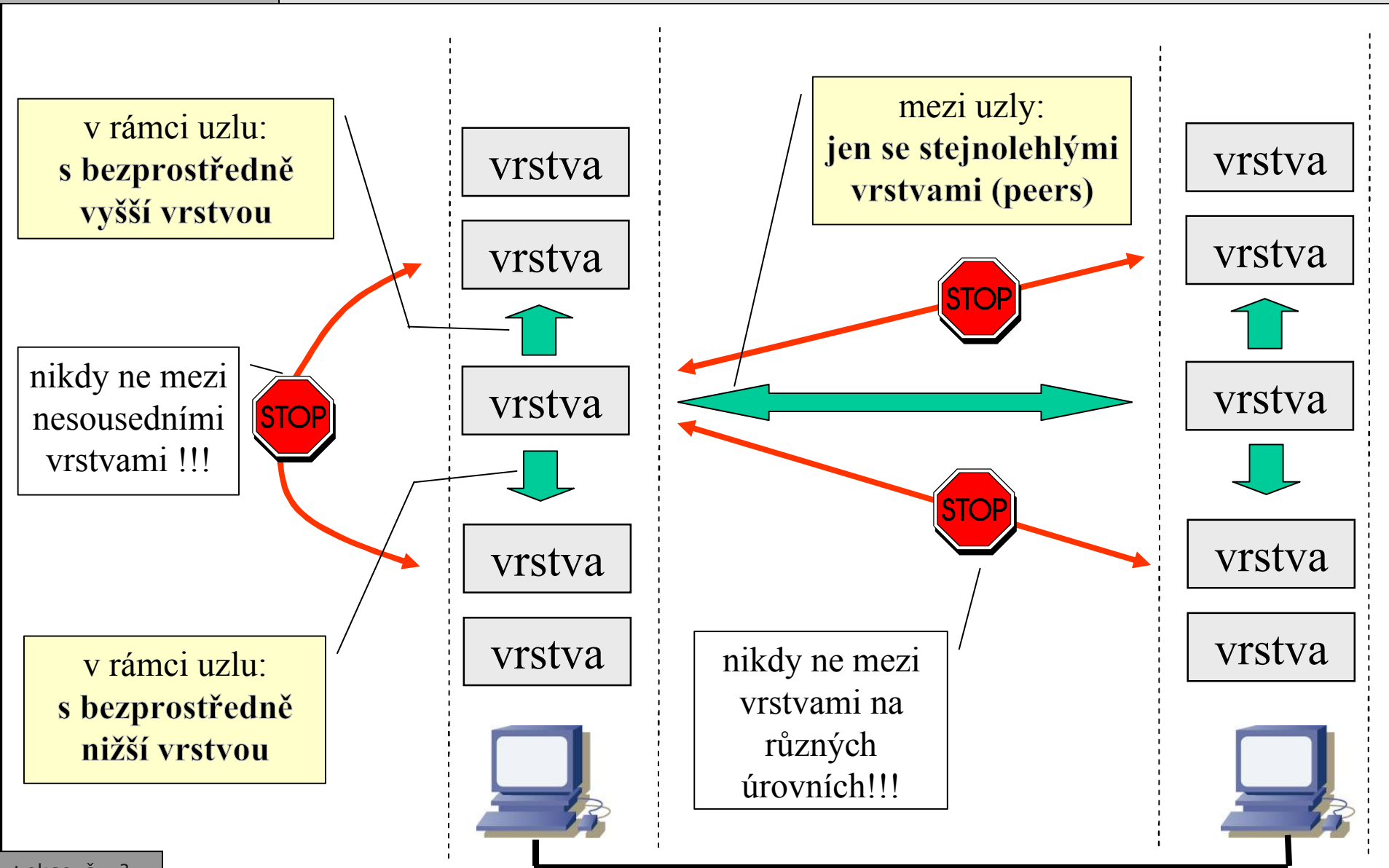
# „vrstevnatá filozofie“

- implementovat funkční síť je hodně složité a náročné
  - 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
- zde: dekompozice se provede po hierarchicky uspořádaných vrstvách
  - dobře to odpovídá povaze řešeného problému
  - přináší to i další výhody
    - možnost alternativních řešení na úrovni nižších vrstev
    - větší modulárnost
    - .....

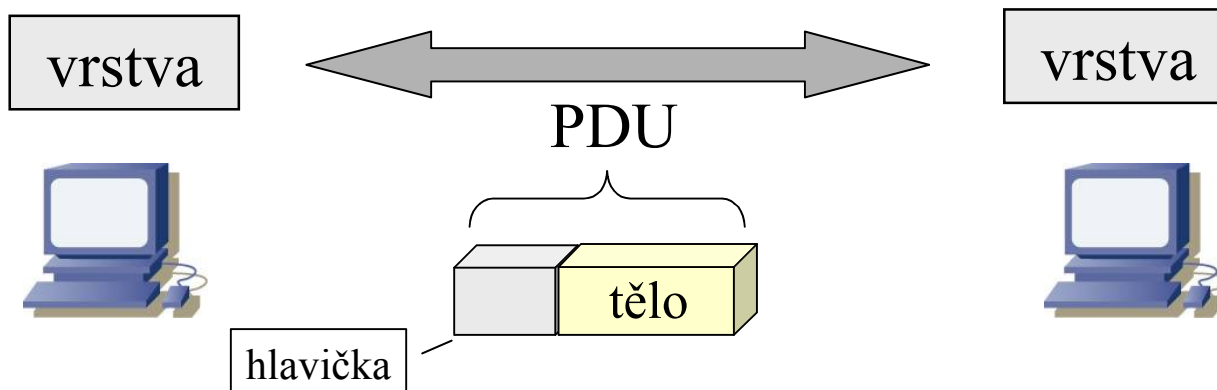


- musí se vyřešit otázky jako:
  - kolik má být vrstev
  - co má která vrstva dělat
  - jak mají vrstvy spolupracovat
    - vertikálně (v rámci uzlu)
    - horizontálně (mezi uzly)
  - .....

# způsob komunikace mezi vrstvami



# "horizontální" komunikace mezi vrstvami

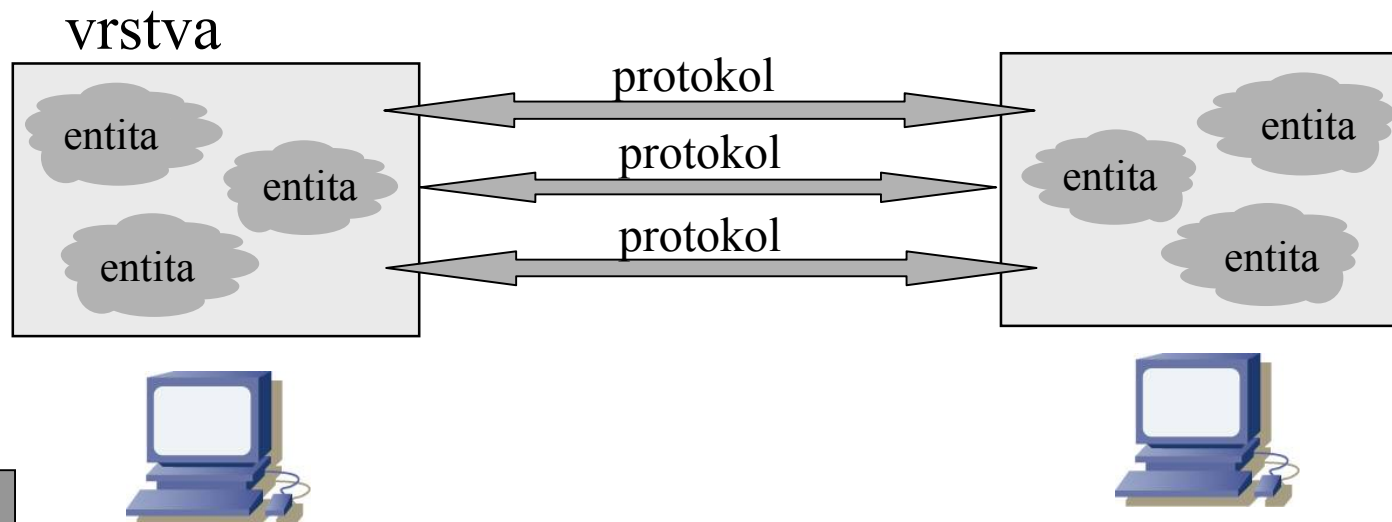


- vzájemná komunikace stejnohlých vrstev různých uzlů se musí řídit předem dohodnutými pravidly
- pravidla této vzájemné komunikace definuje tzv. **protokol**
- protokol definuje (mj.):
  - co si komunikující strany posílají, jaký to má formát a význam, kódování atd.
    - obecně: komunikující strany si předávají tzv. **PDU** (Protocol Data Unit)
    - každý PDU má dvě části: **hlavičku** (header) a **tělo** (náklad, payload)
  - jak komunikace probíhá, jak mají strany reagovat na různé situace atd.
    - ošetřuje nestandardní situace

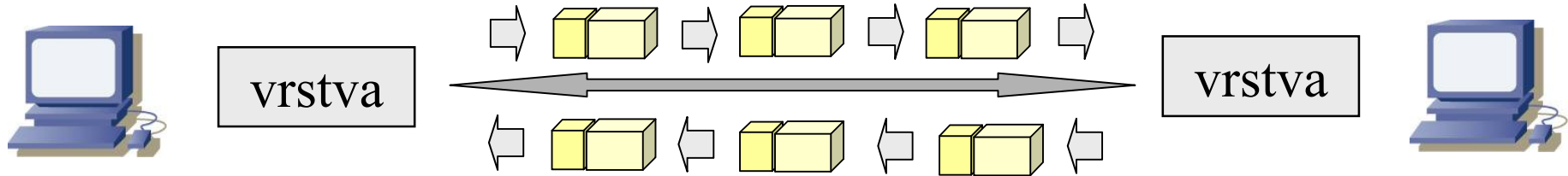
**PDU** se na úrovni různých vrstev nazývá různě:  
**rámec, paket, segment, ...**

# upřesnění: entity a protokoly

- vrstvy nejsou „jednotlivé“
- v každé vrstvě může existovat a fungovat několik relativně samostatných **entit**
  - entita může být např. proces, démon, úloha, ....
- entity ve stejné vrstvě mohou
  - plnit rozdílné funkce (nekonkurovat si)
  - plnit obdobné funkce (ale jiným způsobem, tj. konkurovat si)
- protokol definuje pravidla komunikace mezi entitami stejnohlých vrstev
  - každý protokol vždy „patří“ do určité konkrétní vrstvy
- protokol určuje způsob, jakým je realizována určitá služba
  - pro každou vrstvu může existovat několik alternativních protokolů
    - jeden např. pro spojovaný přenos, druhý pro nespojovaný
  - současné použití různých protokolů (v rámci téže vrstvy) se nemusí vylučovat



# "horizontální" komunikace mezi vrstvami



- komunikace mezi stejnohlými vrstvami (entitami) různých uzlů má asynchronní charakter

- *"pošlu své PDU a čekám na odpověď"*
- *odpověď přijde "kdykoli" (nezávisle na tom, co právě dělám)*

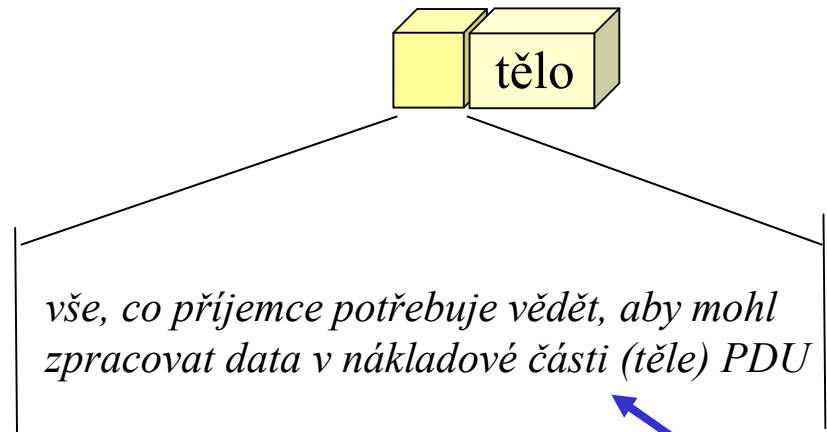
- lze přirovnat k zaslání zpráv

- pošlu zprávu a čekám na odpověď

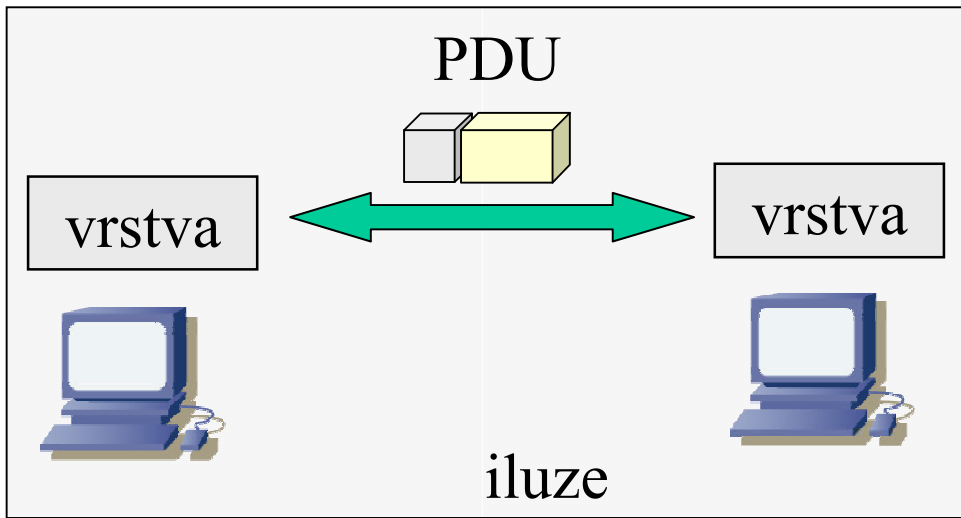
- veškeré "režijní" informace, určené "protistraně" (partnerské entitě na druhé straně) musí být obsaženy v jednotce PDU

- v její hlavičce

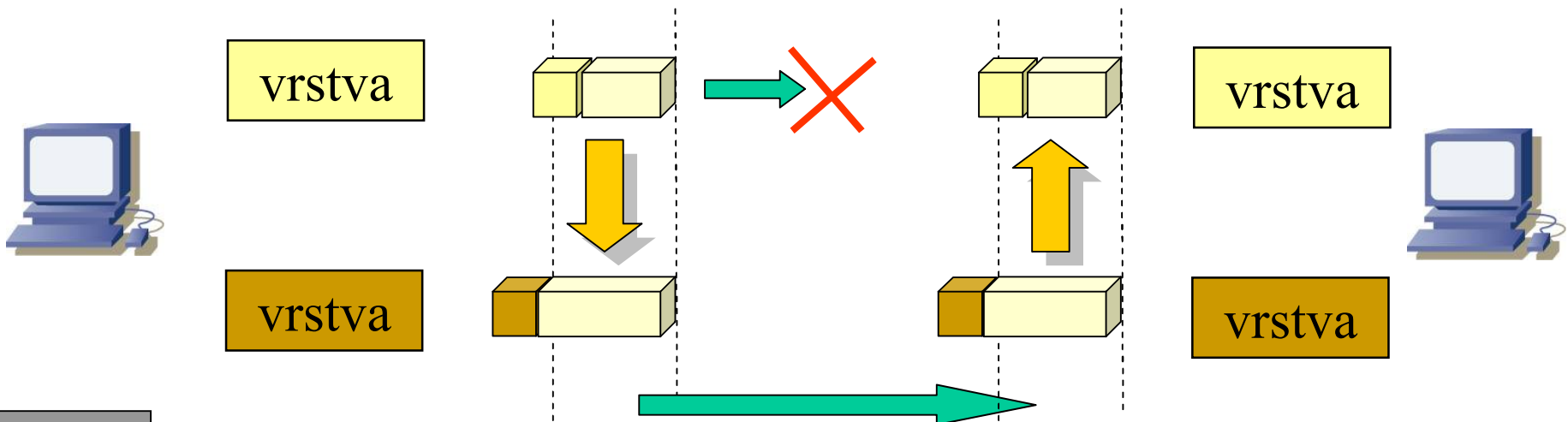
- např. druh zasílaných dat, pořadové číslo PDU, adresa odesílatele a příjemce, kontrolní součet, příznaky, .....



# "horizontální" komunikace mezi vrstvami



skutečnost



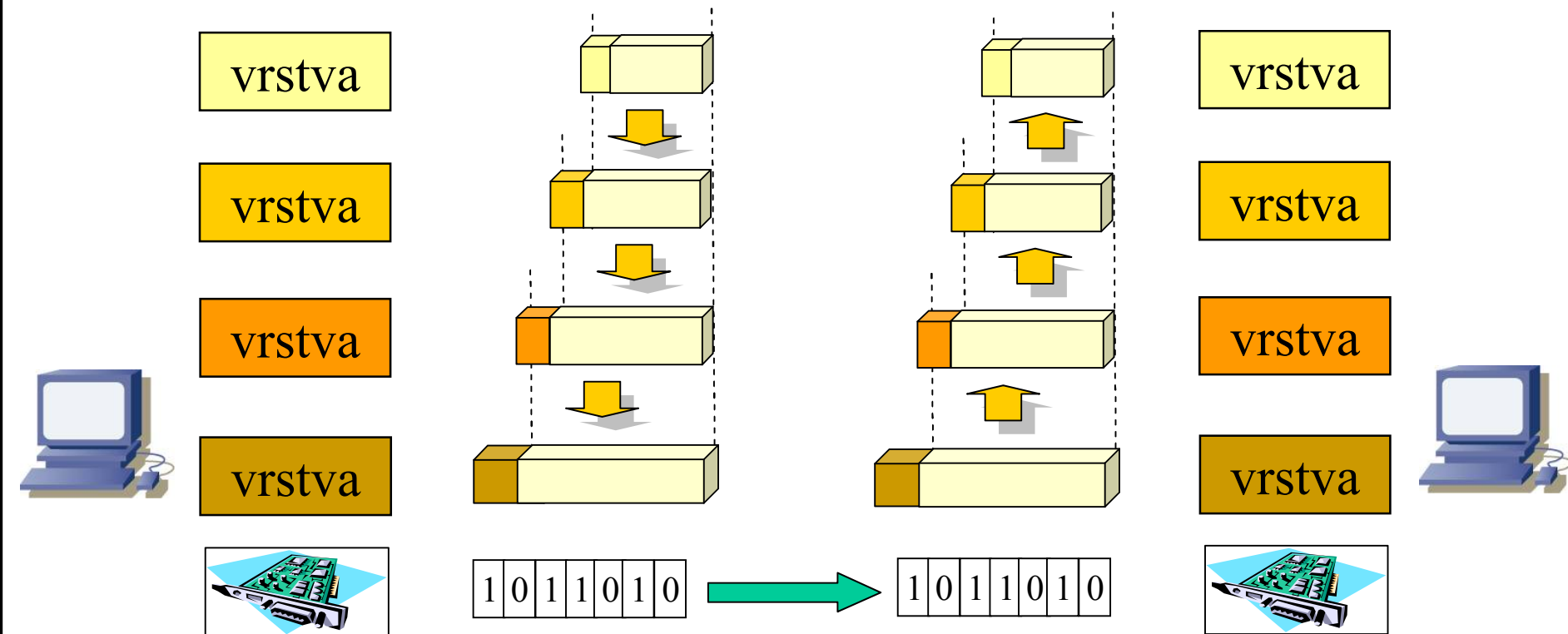
- představa, že si stejnohlé vrstvy skutečně předávají (přímo) mezi sebou jednotky PDU (*rámce, pakety, segmenty, ...*) je POUZE ILUZÍ

- ve skutečnosti je předávají bezprostředně nižší vrstvě k doručení druhé straně !!!

– nižší vrstva je vloží do svého PDU

# "horizontální" komunikace mezi vrstvami

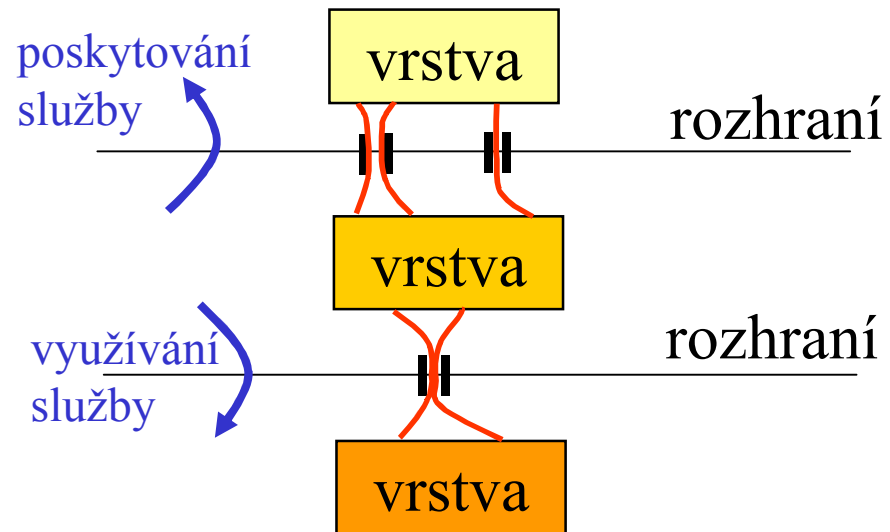
- předávání PDU (bezprostředně) nižší vrstvě k doručení pokračuje až k nejnižší vrstvě (fyzické vrstvě)
- pouze nejnižší vrstva (fyzická vrstva) skutečně přenáší nějaká data
  - po jednotlivých bitech !!!





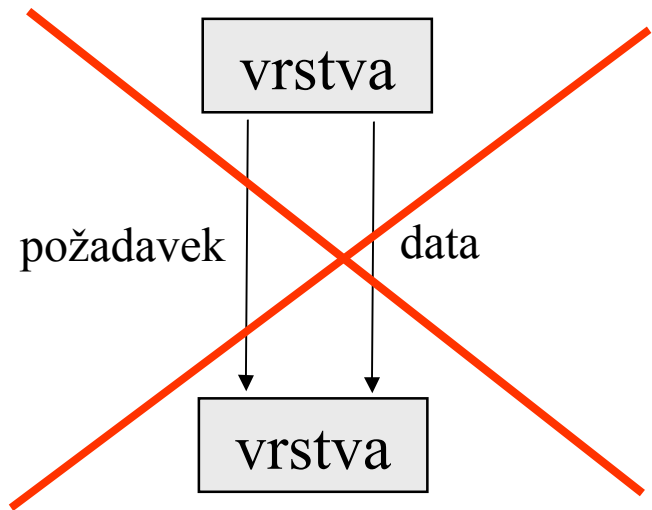
# "vertikální" komunikace mezi vrstvami

- má charakter:
  - poskytování služeb
    - vrstva poskytuje své služby bezprostředně vyšší vrstvě
  - využívání služeb
    - vrstva využívá služeb, poskytovaných bezprostředně nižší vrstvou
      - využívá je k plnění svých úkolů
    - příklad:
      - nejnižší (fyzická) vrstva přenáší jednotlivé bity
      - bezprostředně vyšší (linková) vrstva využívá přenosu jednotlivých bitů k tomu, aby přenášela celé bloky dat
        - » tzv. rámce (své PDU)



- mezi vrstvami je definované rozhraní
  - je implementačně závislé, není "vidět" mimo daný uzel !!!
- komunikace mezi vrstvami probíhá skrze "přechodové body" v tomto rozhraní
  - ISO/OSI: body **SAP** (Service Access Points)
  - TCP/IP: **porty**
- identifikace přechodových bodů je viditelná i "z vně" !!!
  - nezávisí na platformě – např. čísla portů

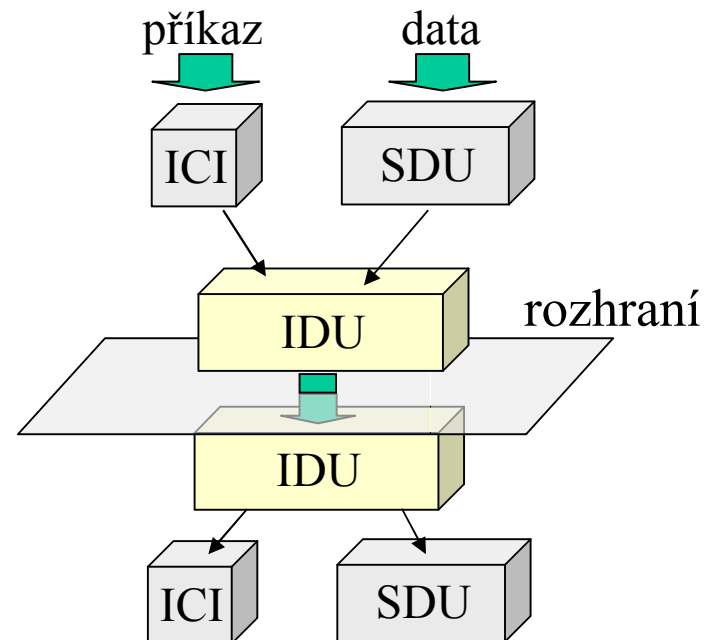
# "vertikální" komunikace mezi vrstvami



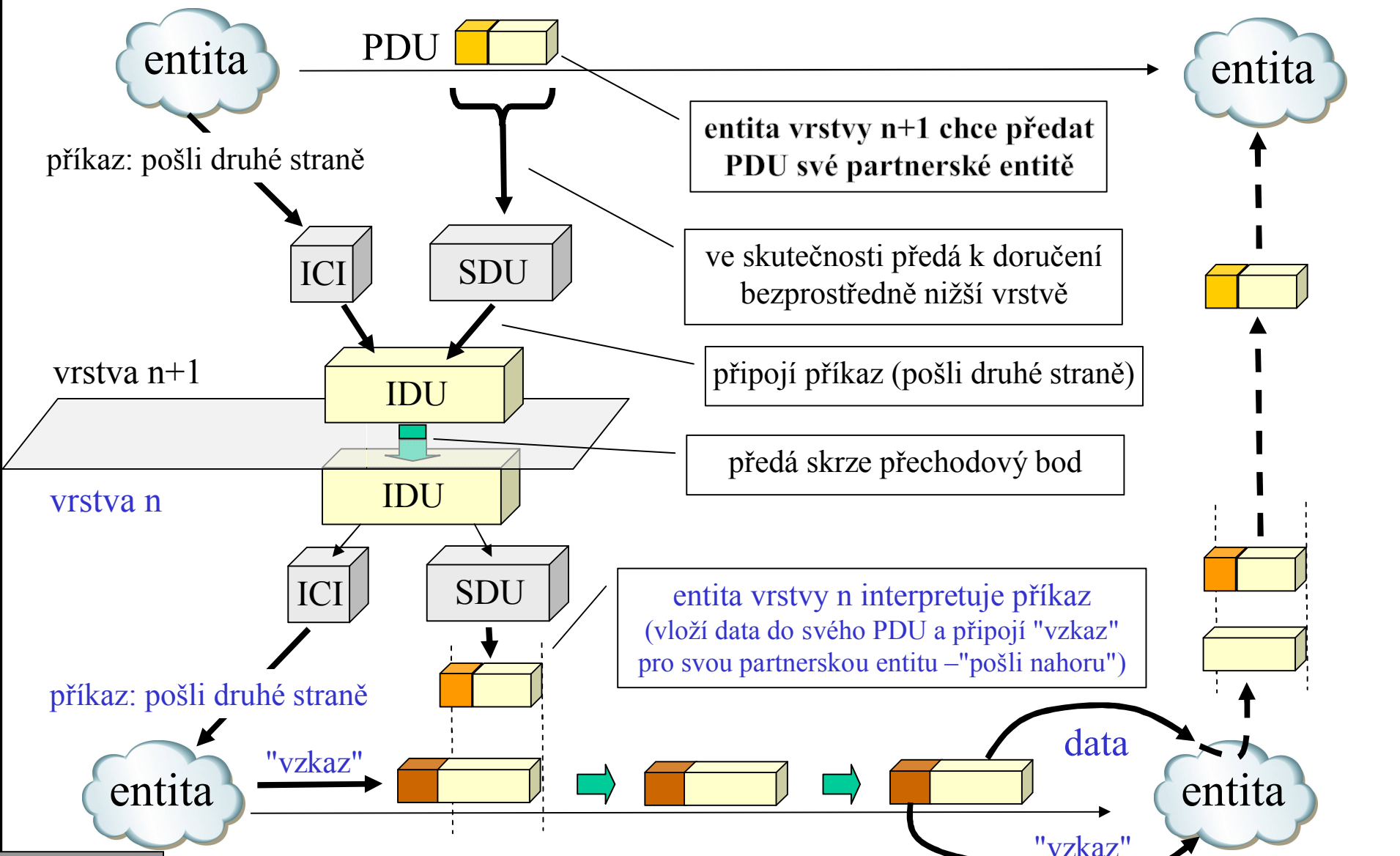
- neprobíhá po 2 samostatných kanálech
  - 1x pro požadavek (příkaz, operace)
  - 1x pro data
- důvod: byly by nutné 2 přechodové body
  - nebo nějaké zdvojení přechodových bodů

- místo toho:
  - požadavek (příkaz) se "zabalí" spolu s daty do jednoho "balíčku" (IDU) a ten se předá skrze přechodový bod
    - IDU (Interface Data Unit)
    - ICI (Interface Control Info)
    - SDU (Service Data Unit)

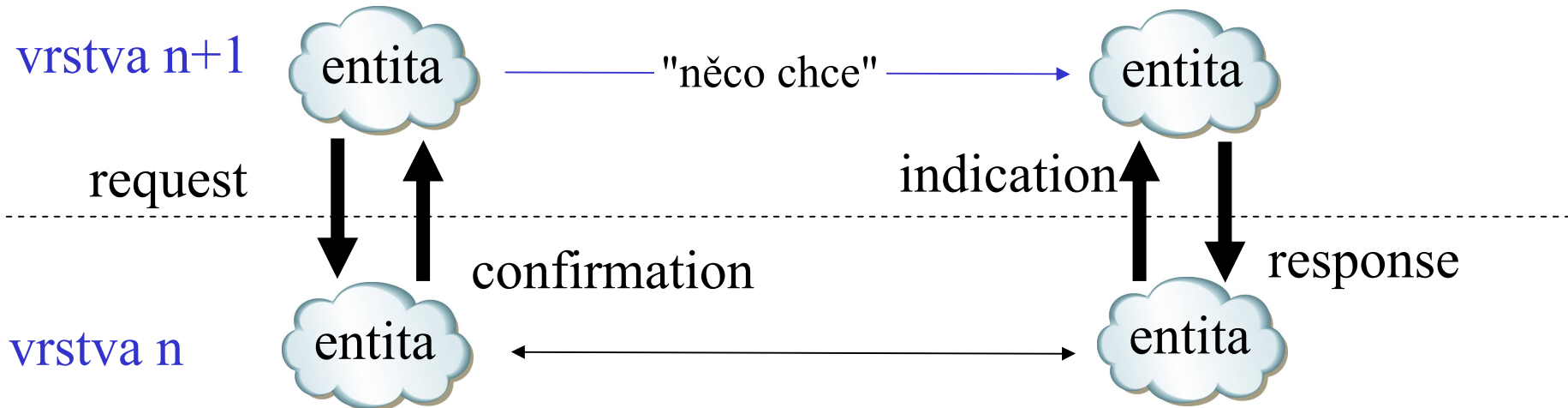
tzv. marshalling



# představa "vertikální" & "horizontální" komunikace



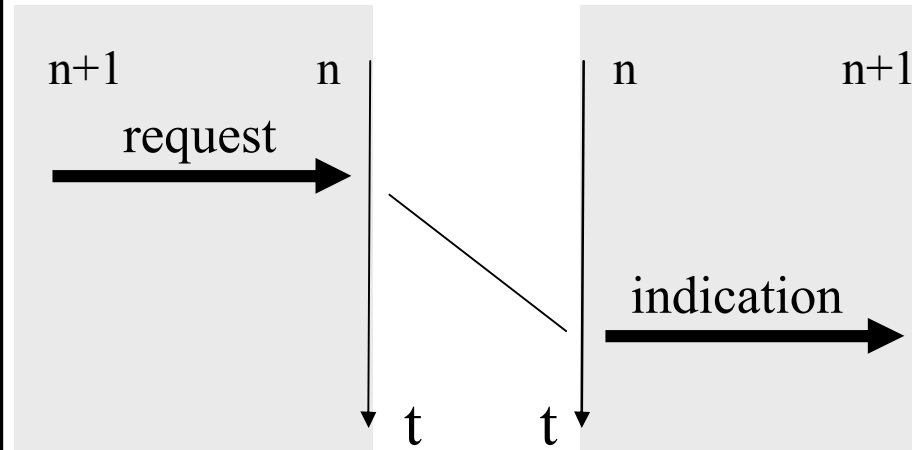
# "vertikální" komunikace – význam předávaných pokynů



- request (žádost)
  - generuje entita vrstvy n+1, která žádá o poskytnutí služby (žadatel)
- indication (indikace, oznámení)
  - generuje "protilehlá" entita vrstvy n, upozorňuje svou "nadřazenou" entitu vrstvy n+1 na požadavek
- response (odpověď)
  - generuje "protilehlá" entita vrstvy n+1, v odpověď na požadavek
- confirmation (potvrzení)
  - generuje entita vrstvy n, potvrzuje tím vyřízení původního požadavku entity vrstvy n+1

obecná  
primitiva, v  
terminologii  
"světa spojů"  
(umožňují  
přístup ke  
službám)

# časový průběh komunikace - obecně

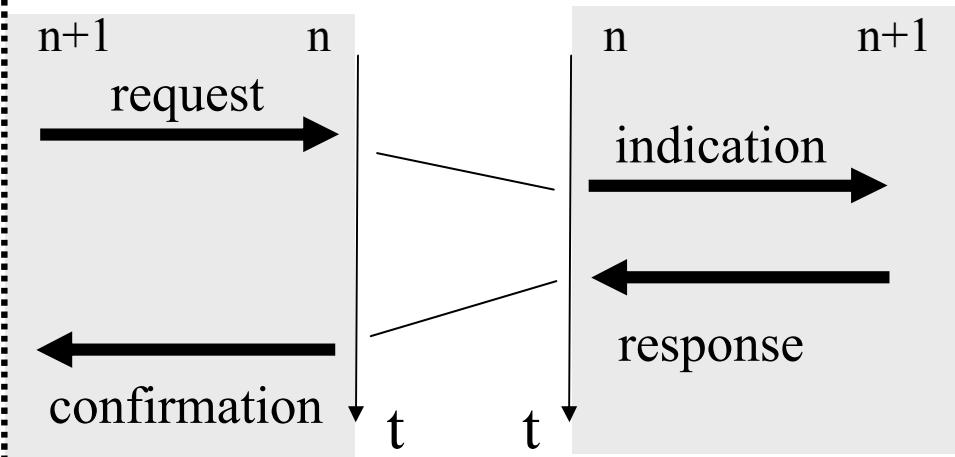


- **nespojovaný (nepotvrzovaný) způsob komunikace**

- "žadatel" vyšle žádost
  - žádost přijme entita "žadatele" na vrstvě n a předá své partnerské entitě u "poskytovatele"
- "poskytovatel" je upozorněn na žádost
  - entita vrstvy n u "poskytovatele" přijme požadavek a upozorní na něj nadřazenou entitu (na vrstvě n+1)

- **spojovaný (potvrzovaný) způsob komunikace**

- "žadatel" vyšle žádost
- "poskytovatel" je upozorněn na žádost
- "poskytovatel" vygeneruje odpověď
- "žadatel" obdrží potvrzení o vyřízení svého požadavku



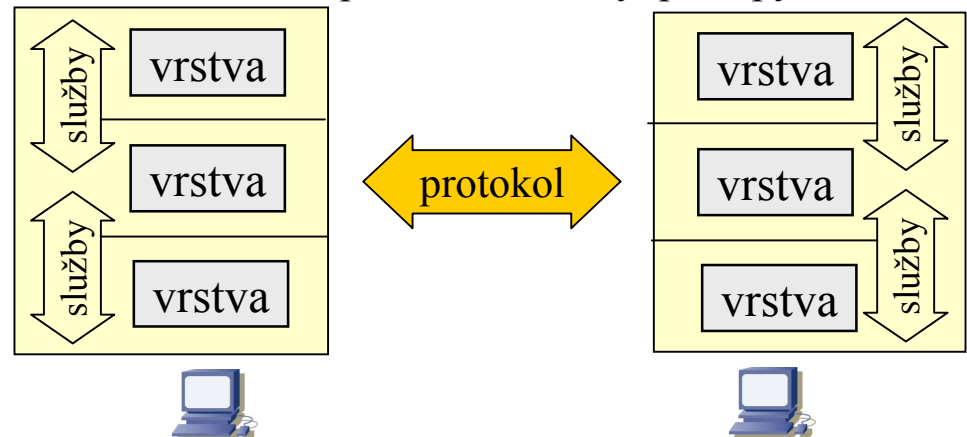
# shrnutí

## služby (angl.: services)

- týkají se vertikální komunikace
  - mezi vrstvami
  - jsou poskytovány "skrze" rozhraní mezi vrstvami
- *služby ani rozhraní nejsou "vidět" z vně daného uzlu*
  - jediná výjimka: identifikace přechodových bodů, slouží k identifikaci entit v rámci vrstvy
- rozhraní mezi vrstvami nemusí být standardizováno
  - může být (a bývá) na různých uzlech/platformách různé
- nemusí být (a nejsou) standardizovány ani služby
  - není potřeba, aby byly na všech uzlech stejné

## protokoly (angl.: protocols)

- týkají se "horizontální" komunikace
  - mezi stejnohlými vrstvami různých uzlů
- *protokoly jsou "vidět" z vně daného uzlu*
  - jejich implementace může zůstat skryta
- protokoly musí být standardizovány
  - musí být dopředu a všem známo, jaké jsou ...
    - pravidla, formáty, postupy atd.



# sít'ový model a sít'ová architektura

- **sít'ový model** je ucelená představa o tom, jak mají být sítě řešeny
  - zahrnuje:
    - představu o počtu vrstev
    - představu o tom, co má mít která vrstva na starosti
  - nezahrnuje:
    - konkrétní představu o tom, jak má která vrstva své úkoly plnit
      - tedy konkrétní protokoly
- příklad sít'ového modelu:
  - **referenční model ISO/OSI**
- **sít'ová architektura** obsahuje navíc také:
  - konkrétní protokoly
    - tj. obsahuje i konkrétní protokoly
- příklad sít'ové architektury:
  - **rodina protokolů TCP/IP**

konkrétní protokoly vznikaly samostatně a dodatečně

# referenční model ISO/OSI

- byl pokusem vytvořit univerzální síťovou architekturu,
  - skončil jako síťový model
    - bez protokolů, ty se dodělávaly postupně
- 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 organizace ČSNI
- byl „oficiálním řešením“
  - řešením, které prosazovaly "orgány státu" a chtěly jej nasadit do praxe
  - dnes je prakticky odepsaný, prohrál v souboji s TCP/IP





# geneze referenčního modelu ISO/OSI

- reagoval na vznik proprietárních a uzavřených sítí
  - IBM SNA, ....
- prvotní záměr: definovat, jak mají vypadat otevřené systémy
  - odsud: Open Systems Architecture
  - tj. i chování „uvnitř“, nejen „mezi sebou“
  - ukázalo se jako příliš náročné, dochází k redukci ambic
- druhé přiblížení: bude se týkat jen vzájemného propojení otevřených systémů
  - změna názvu: Open Systems Interconnection Architecture
  - opět se ukázalo jako příliš náročné
- třetí přiblížení: nebude to obsahovat konkrétní protokoly
  - ale jen představu o počtu vrstev a o tom, co má která vrstva dělat

## Co je „referenční model ISO/OSI“?

- poslední iterace názvu
  - Open Systems Interconnection
- když byly „odstraněny“ protokoly, zbyl jen síťový model
  - jde o obecný „rámec“, do kterého jsou „zasazována“ konkrétní řešení
- konkrétní protokoly pro RM ISO/OSI byly vyvíjeny samostatně
  - a teprve dodatečně zařazovány do rámce ISO/OSI

# filosofie RM ISO/OSI

- vznikal „od zeleného stolu“
  - a pak byl „nadiktován“ uživatelům
- vznikal maximalistickým způsobem
  - autoři se snažili zahrnout „vše, co by někdy někomu mohlo hodit“
- výsledek byl dosti odtažitý od reálné praxe
  - celá řešení se často ukázala jako nerealizovatelná, a hledala se implementovatelná podmnožina
  - vznikaly různé implementovatelné podmnožiny, které nebyly vzájemně kompatibilní
- mnohé výchozí předpoklady se ukázaly jako chybné



- profily GOSIP
  - když si státní (veřejné) instituce kupovaly síťové vybavení, trvaly na tom, aby vycházelo z ISO/OSI
    - ale RM ISO/OSI je příliš široký, a pro praktické použití je nutné jej „zúžit“ (vybrat implementovatelnou podmnožinu)
  - státní (veřejné) instituce musely takovouto podmnožinu přesně vyspecifikovat
    - tím vznikly profily GOSIP (Government OSI Profile)

pro srovnání:

- rodina protokolů TCP/IP vznikala postupným obohacováním
  - nejprve se navrhlo jednoduché řešení
  - teprve postupně se "obohacovalo"
    - až na základě skutečné potřeby
    - nové řešení muselo nejprve prokázat svou životaschopnost (možnost implementace, provozní zkušenosti)

# sedm vrstev ISO/OSI

- autoři ISO/OSI se dosti dlouho přeli o počtu 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
  - 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 a prezentační
- jedna vrstva ISO/OSI (linková) je naopak "přetížená" a rozpadla se na dvě podvrstvy
  - podvrstvu LLC
  - podvrstvu MAC

zde šlo hlavně o již existující standard X.25

# sedm vrstev ISO/OSI



↑  
vrstvy orientované  
na podporu aplikací  
↓

---

**přizpůsobovací vrstva**

---

↑  
vrstvy orientované  
na přenos dat  
(šito na míru filosofii  
veřejných datových sítí  
dle X.25)  
↓

# fyzická vrstva

- zabývá se výhradně přenosem bitů
  - a v rámci toho otázkami typu:
    - kódování,
    - modulace,
    - časování,
    - synchronizace,
    - el. parametry signálů, konektory,
    - řídicí signály rozhraní, ....
- nabízí služby typu
  - přijmi bit,
  - odešli bit
- nijak neinterpretuje to, co přenáší
  - jednotlivých bitům nepřisuzuje žádný specifický význam
  - každý bit přenáší stejně
    - datový, režijní, ....
- na úrovni fyzické vrstvy se rozlišuje:
  - paralelní a sériový přenos
  - synchronní, asynchronní a arytmičtý přenos
  - přenos v základním a přeloženém pásmu
- pro přenos bitů mohou být využívána různá přenosová média
  - drátová
  - bezdrátová
- na úrovni fyzické vrstvy se pracuje s veličinami jako je:
  - šířka pásma (bandwidth)
  - modulační rychlost
  - přenosová rychlost

11001001

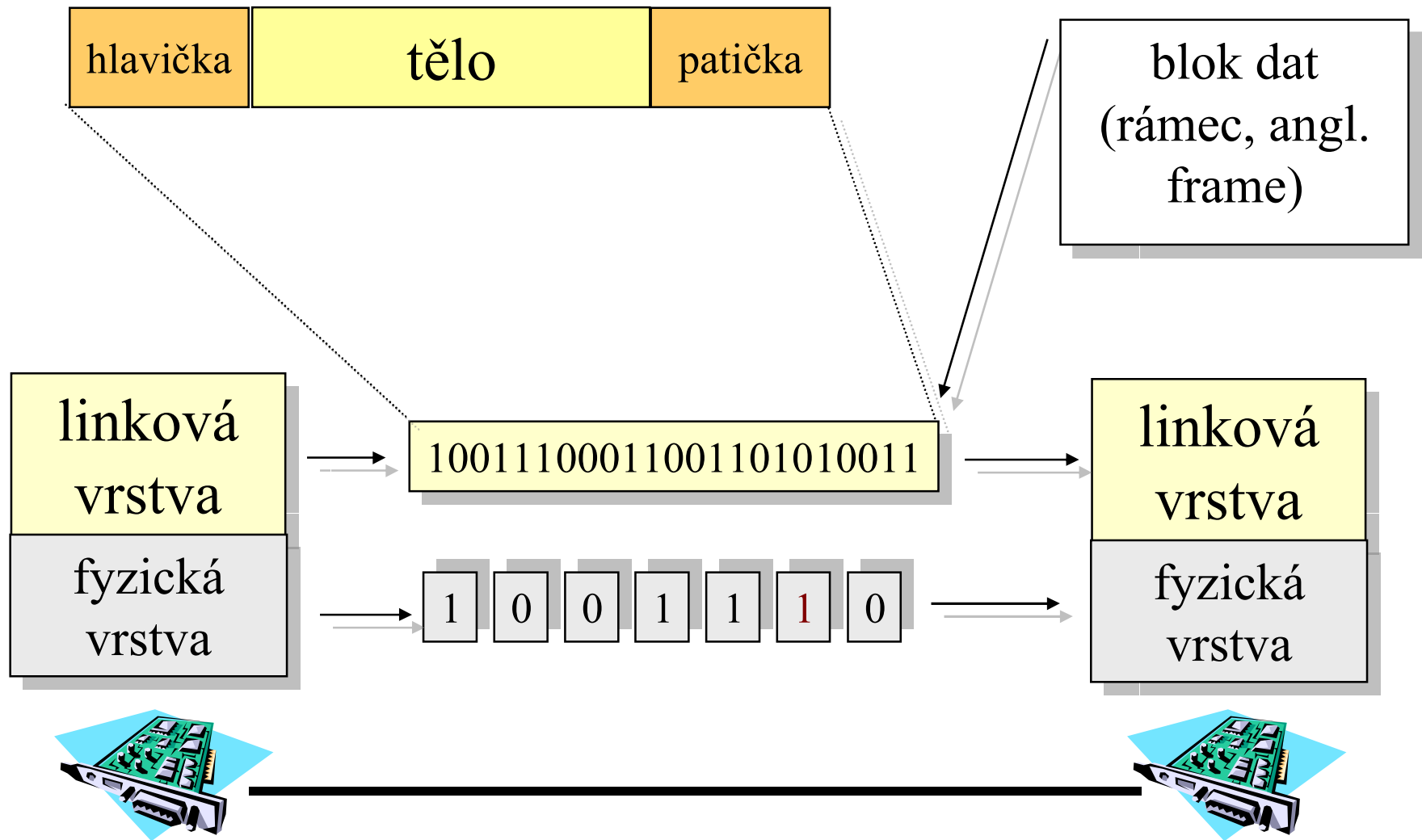
11001001010101001010101010010

# linková vrstva (spojová vrstva)

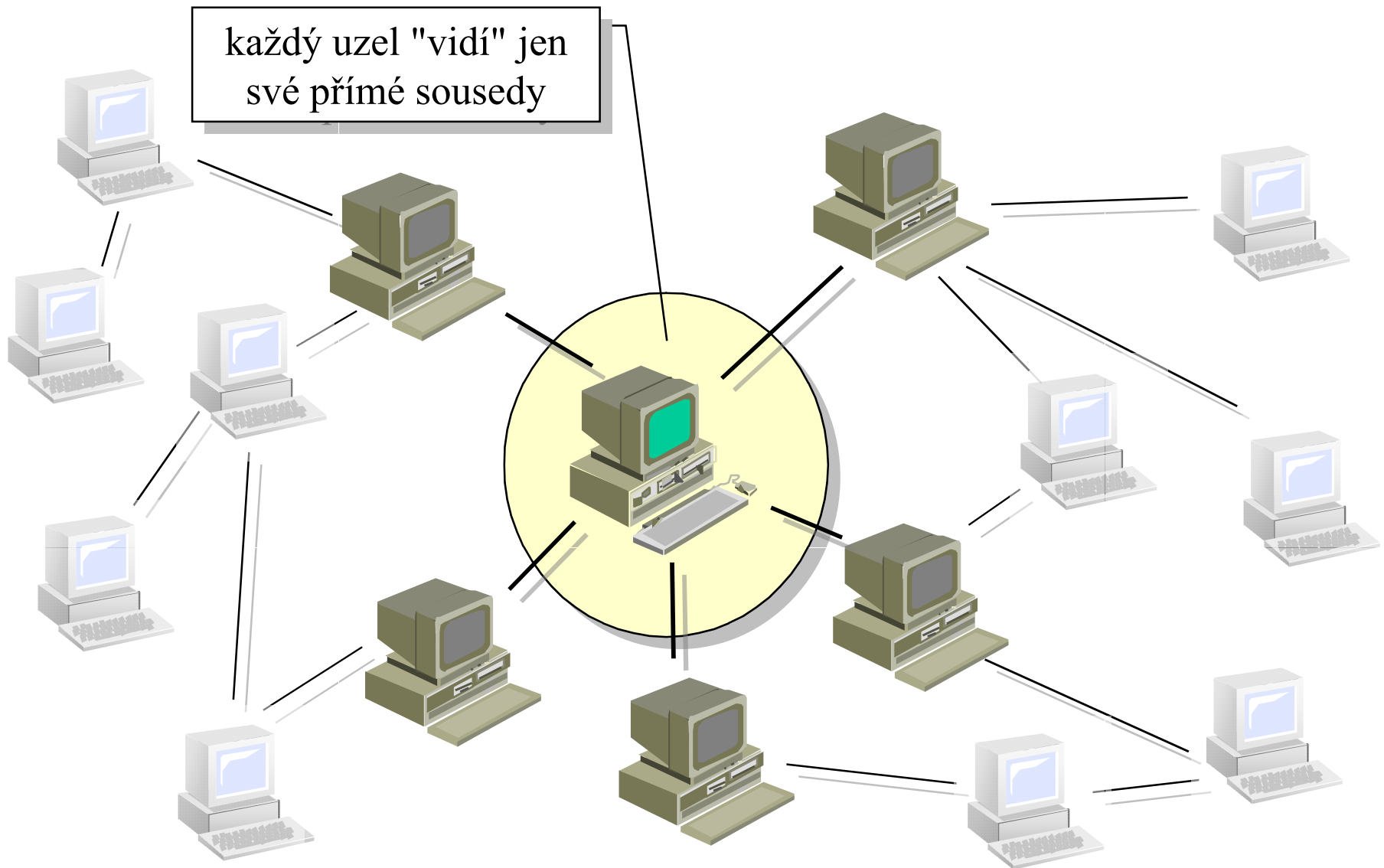
- přenáší celé bloky dat
  - tzv. **rámce** (frames)
  - využívá k tomu služby fyzické vrstvy
    - přijmi/odešli bit
- přenos rámců zajišťuje pouze k přímým sousedům
  - pouze v dosahu přímého spojení, bez "přestupu" přes mezilehlé uzly
- může fungovat
  - spolehlivě či nespolehlivě
  - spojovaně či nespojovaně
  - best effort / QoS
- může využívat různé přenosové technologie
  - na úrovni fyzické vrstvy
- úkoly linkové vrstvy:
  - synchronizace na úrovni rámců
    - správné rozpoznání začátku a konce rámce, i všech jeho částí
  - zajištění spolehlivosti (pokud je požadováno)
    - detekce chyb a jejich náprava
  - řízení toku
    - zajištění toho, aby vysílající nezahltil příjemce
  - přístup ke sdílenému médiu
    - řeší konflikty při vícenásobném přístupu ke sdílenému médiu
    - tento úkol nebyl mezi původně uvažovanými – následně způsobil rozpad linkové vrstvy na dvě podvrstvy



# představa fungování linkové vrstvy



# představa fungování linkové vrstvy

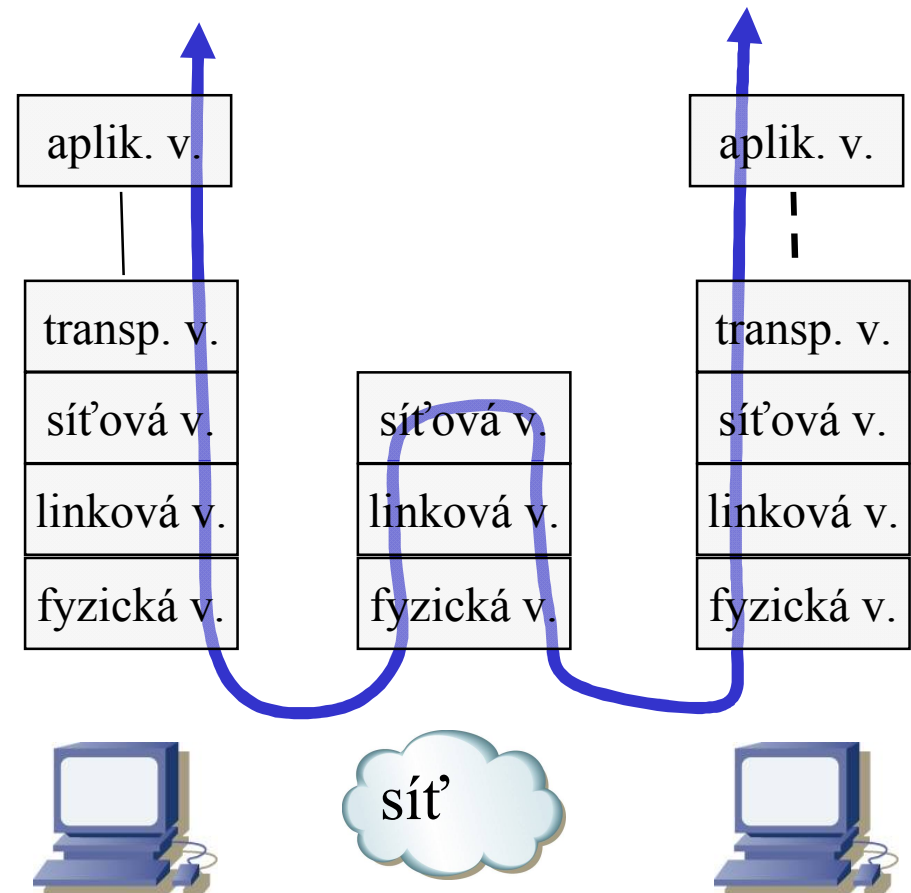




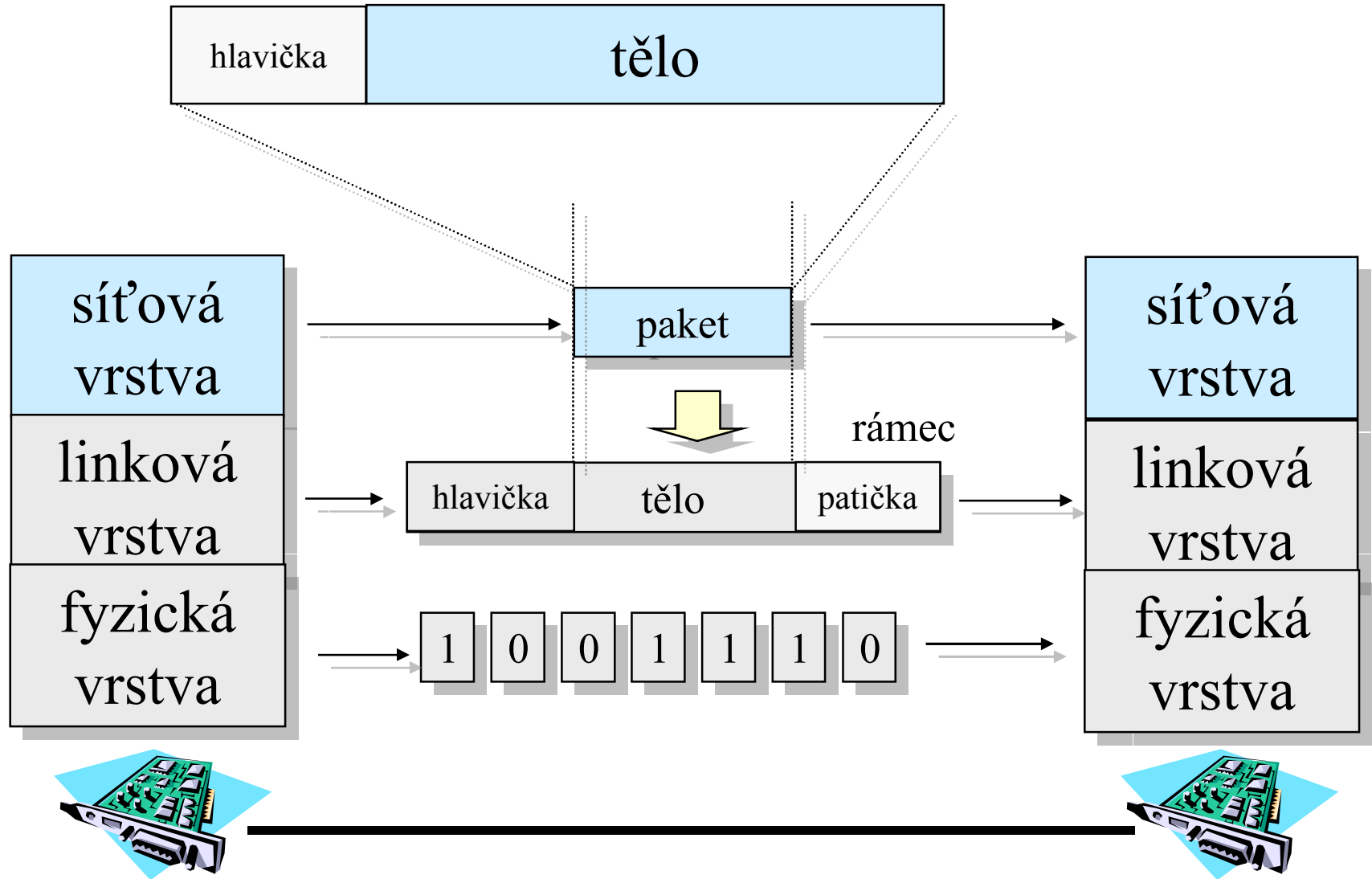
# síťová vrstva

- přenáší bloky dat označované jako **pakety** (packets)
  - fakticky: vkládá je do linkových rámců
    - předává linkové vrstvě k doručení
- zajišťuje doručení paketů až ke konečnému adresátovi
  - tj. přes různé mezilehlé uzly
    - tzv. směrovače
  - hledá vhodnou cestu až k cíli
    - zajišťuje tzv. **směrování** (routing)
- může používat různé algoritmy směrování:
  - adaptivní, neadaptivní
  - izolované, distribuované, centralizované, ..

- je poslední vrstvou, kterou musí mít i "přestupní" uzly
  - tzv. směrovače (zajišťující směrování)

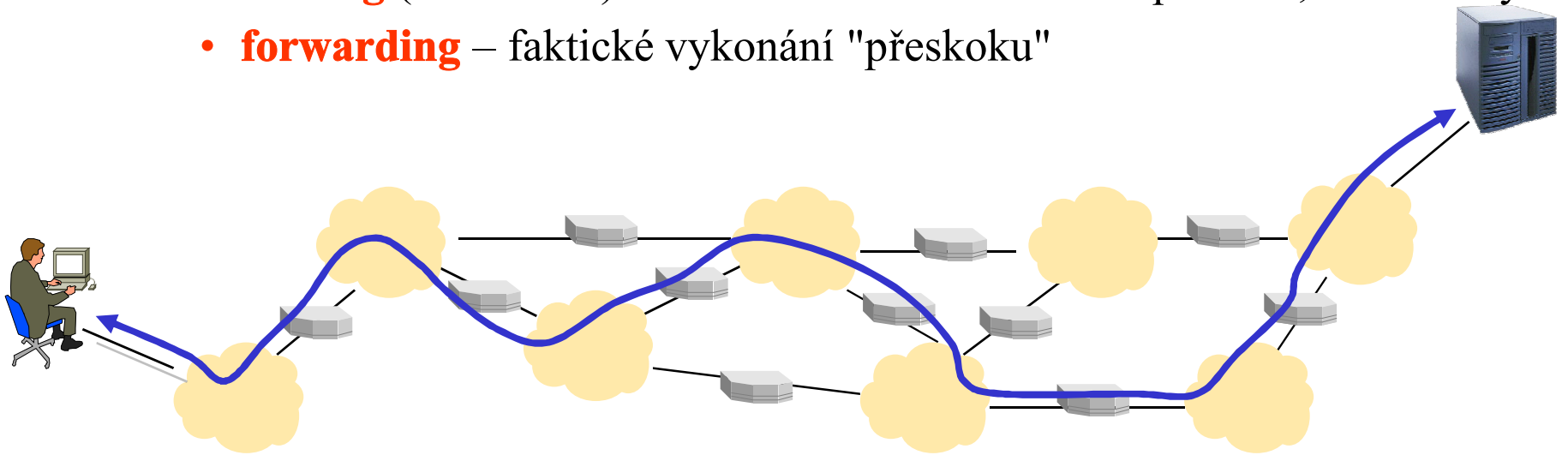


# představa fungování síťové vrstvy

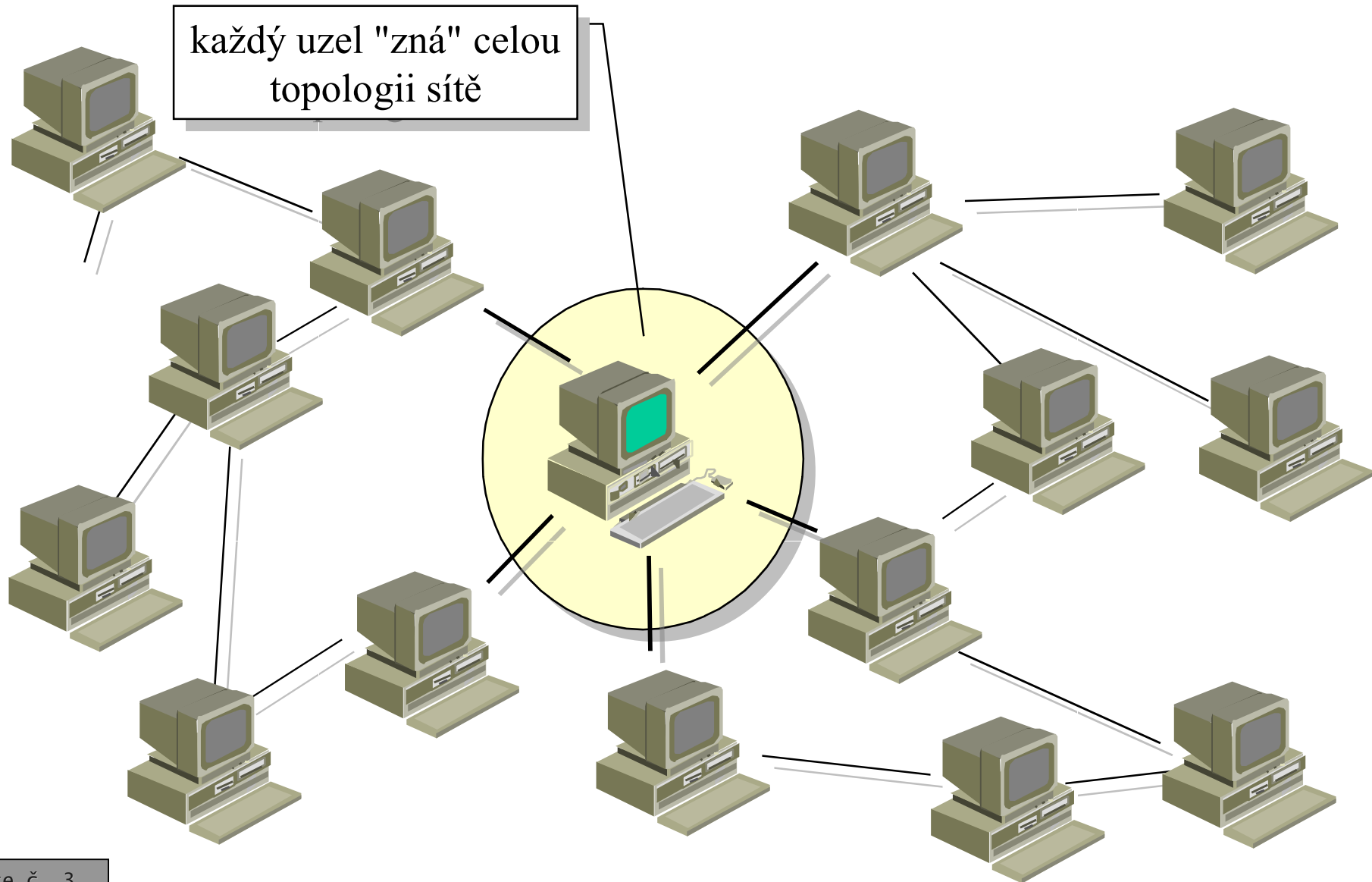


# hlavní úkol síťové vrstvy

- doručovat data od jejich zdroje až k jejich koncovým adresátům
  - což může obnášet „přeskok“ přes různé mezilehlé uzly
    - linková vrstva se stará jen o doručování k přímým sousedům (v dosahu přímého spojení) a nezabývá se přeskoky
  - „přeskok“ vyžaduje:
    - **routing** (směrování) - rozhodnutí o dalším směru přenosu, volba trasy
    - **forwarding** – faktické vykonání "přeskoku"

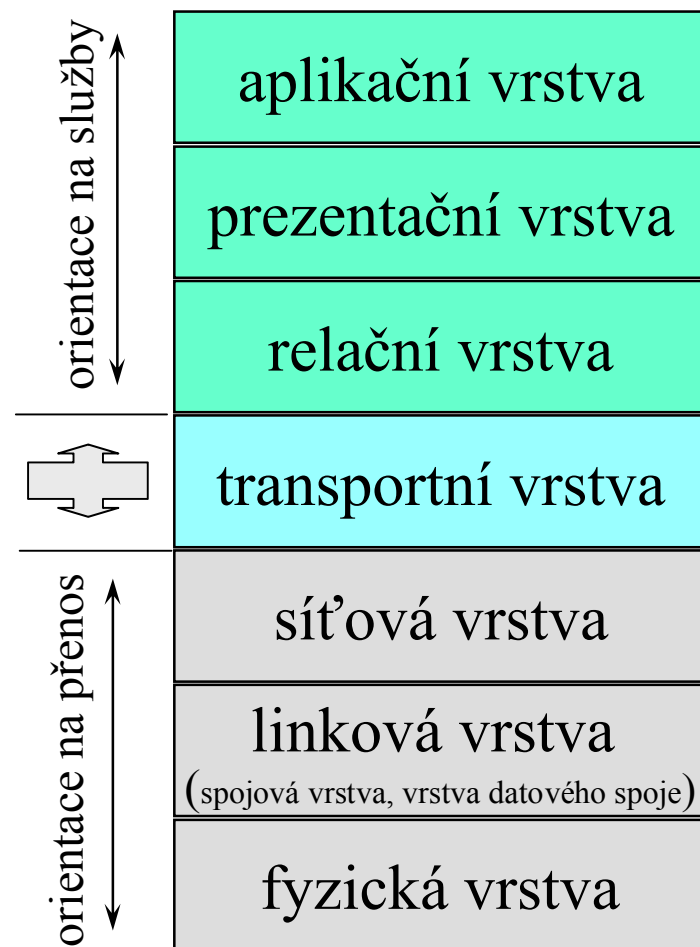


# představa fungování síťové vrstvy

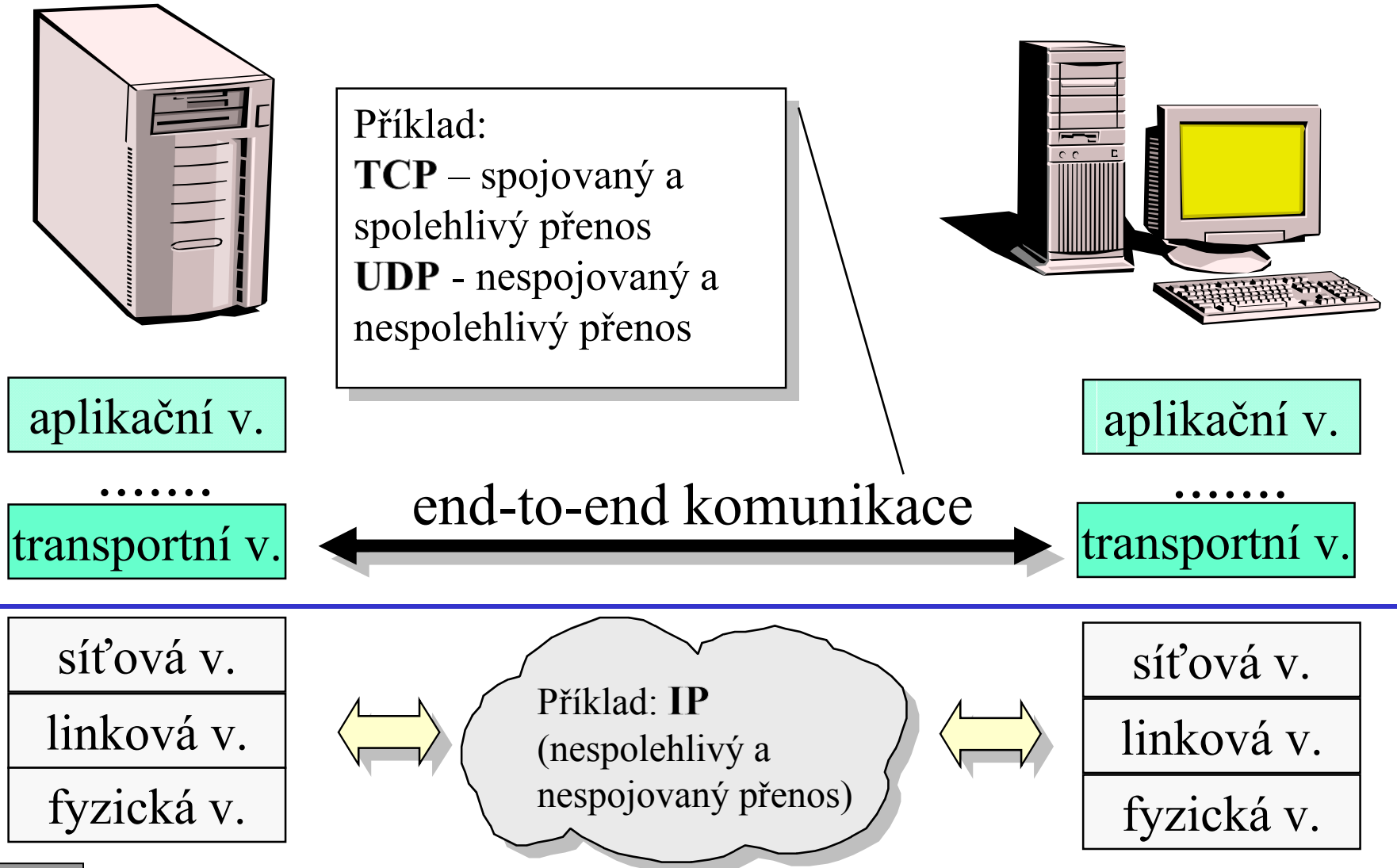


# transportní vrstva

- teze:
  - nelze „hýbat“ s vlastnostmi a funkcemi nižších vrstev
    - třeba proto že patří někomu jinému
  - vyšší vrstvy mohou chtít něco jiného, než co nabízí nižší vrstvy
- je úkolem transportní vrstvy zajistit potřebné přizpůsobení!
- zajišťuje:
  - komunikaci mezi koncovými účastníky (end-to-end komunikaci)
- může měnit
  - nespolehlivý charakter přenosu na spolehlivý
  - méně spolehlivý přenos na více spolehlivý
  - nespojovaný přenos na spojovaný

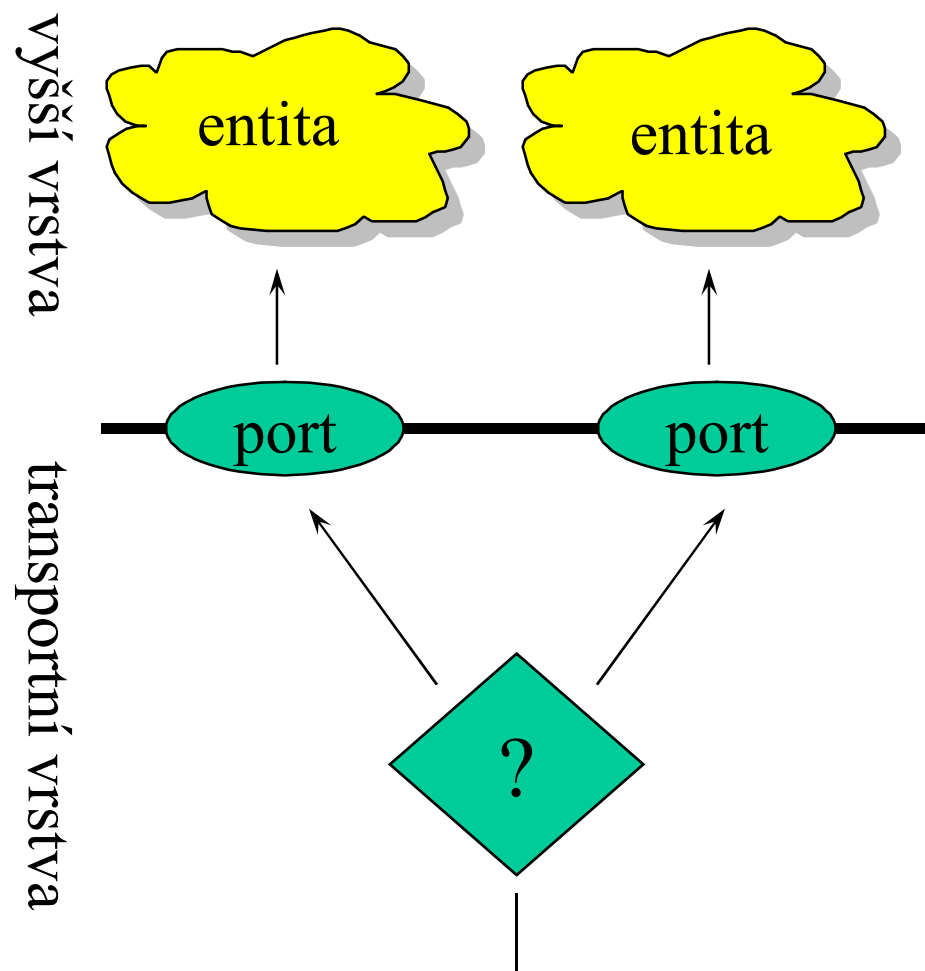


# představa transportní vrstvy



# další úkol transportní vrstvy

- do vrstvy síťové (včetně) se uzly chápou jako nedělitelné celky
  - síťové adresy reprezentují celé uzly
  - například: IP adresy v TCP/IP
- transportní vrstva již rozlišuje konkrétní entity v rámci každého uzlu
  - jednotlivé procesy, démony, úlohy ....
    - rozlišuje je obvykle nepřímo, skrze přechodové body (**body SAP, porty**) ke kterým jsou tyto entity asociovány
    - např. čísla portů v TCP/IP



# relační vrstva

- zajišťuje vedení relací
  - a co jsou relace?
- může zajišťovat:
  - synchronizaci
  - šifrování
  - podporu transakcí
  - .....



..... má toho poměrně málo na práci

- existence relační vrstvy v RM ISO/OSI je nejvíce kritizována
  - v TCP/IP relační vrstva zcela chybí

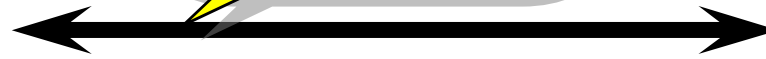


# představa relací

relační  
vrstva



jeden dialog  
(relace)



transportní vrstva

po výpadku je spojení vytočeno znovu



v rámci jednoho spojení probíhají  
(postupně) dva dialogy



# prezentační vrstva

- nižší vrstvy se snaží doručit každý bit přesně tak, jak byl odeslán
- stejná posloupnost bitů může mít pro příjemce jiný význam než pro odesilatele, např. kvůli
  - kódování znaků (ASCII, EBCDIC,...)
  - formátu čísel
  - formátu struktur, polí
  - ukazatelům (pointerům)
- prezentační vrstva má na starosti potřebné konverze, tak aby obě strany interpretovaly přenášená data stejně
  - např. u reálných čísel, textů
- prezentační vrstva má na starosti převod přenášených dat z/do takového tvaru, v jakém je možné je přenášet
  - např. je nutné "zlinearizovat" vícerozměrná pole, vícerozměrné datové struktury
  - týká se to např. i způsobu ukládání vícebytových položek do paměti
    - která je posloupností bytů

přenosový kanál je 1-rozměrný

# příklad: konvence Little Endian vs. Big Endian

týká se ukládání  
vícebytových  
položek do paměti



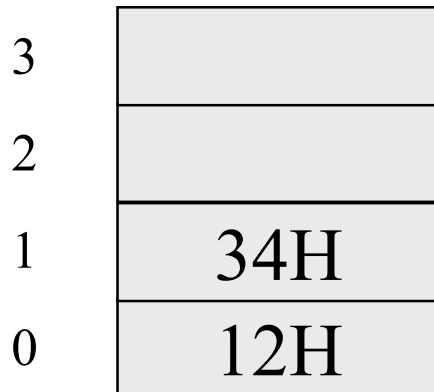
**podobný problém:**  
US: mm/dd/yy  
EU: dd/mm/yy  
Jap.: yy/mm/dd

## Big Endian

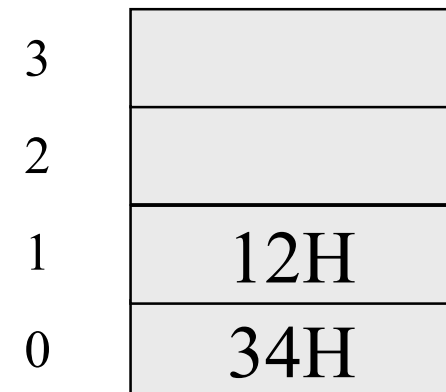
(vyšší byte je na  
nižší adrese)

## Little Endian

(nižší byte je na  
nižší adrese)



← 1234H →



např. mikroprocesory  
Motorola, SUNSparc, PowerPC,  
mainframy IBM, Internet



např. mikroprocesory  
Intel, AMD, Ethernet

# proč Little a Big Endian?

- 1. apríla 1980 napsal Dave Cohen dokument IEN 137

- IEN (Internet Experiment Notes) byly typem dokumentů, které posléze splynuly s RFC

- s názvem:

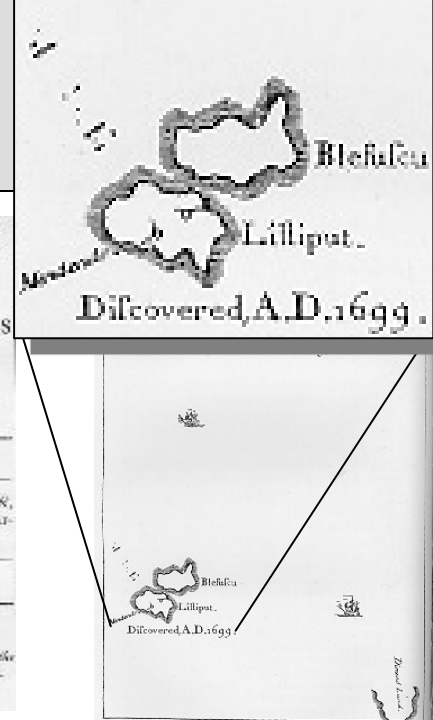
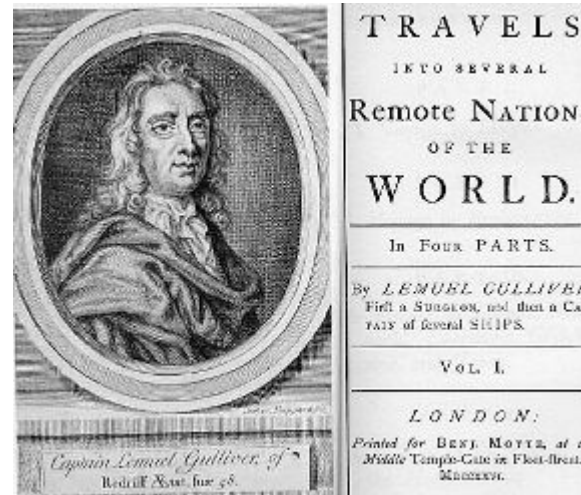
- **ON HOLY WARS AND A PLEA FOR PEACE**

- This is an attempt to stop a war. I hope it is not too late and that somehow, magically perhaps, peace will prevail again.
    - The latecomers into the arena believe that the issue is: "What is the proper byte order in messages?".

- přirovnává to k pasáži z Gulliverových cest

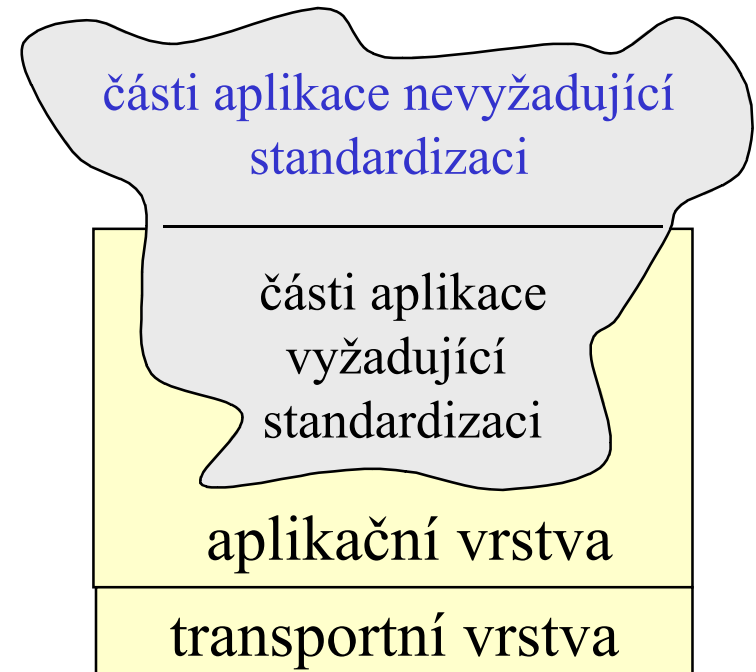
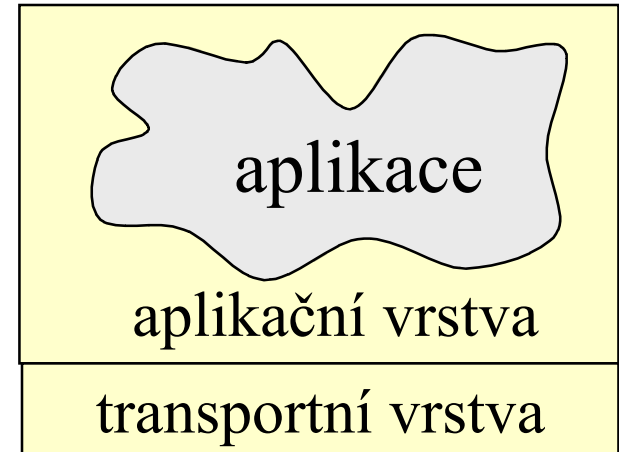
- kde Jonathan Swift parodoval náboženské války mezi Anglií a Francií, mezi protestanty a katolíky

- *Gulliver se dostal do země Lilliput, která vedla nesmiřitelnou válku s říší Blefuscu*
    - *bojovalo se o to, zda rozbít vajíčka "na malém konci" ("at the **L**ittle End", jak to praktikovali v říši **L**illiput), nebo "na velkém konci" ("at the **B**ig End", v říši **B**lefuscus)*



# aplikační vrstva

- původní představa:
  - bude obsahovat aplikace
  - problém: aplikací je moc, musely by být všechny standardizovány
    - to nejde stihnout
    - nemělo by to ani smysl
- později:
  - aplikační vrstva bude obsahovat pouze „jádro“ aplikací, které má smysl standardizovat
    - například přenosové mechanismy el. pošty
    - ostatní části aplikací (typicky: uživatelská rozhraní) byly vysunuty nad aplikační vrstvu



# kritika RM ISO/OSI

- je příliš prodchnut „světonázorem“ lidí od spojů, ale měl sloužit „lidem od počítačů“
- je příliš složitý, těžkopádný a obtížně implementovatelný
- je příliš maximalistický
  - chce nejprve všechno, a pak se musí uskrovnovat
- vzniká „od zeleného stolu“, nerespektuje požadavky a realitu běžné praxe
- počítal spíše s rozlehlými sítěmi než se sítěmi lokálními
- některé činnosti (funkce) zbytečně opakuje na každé vrstvě
- jednoznačně upřednostňuje spolehlivé a spojované přenosové služby
- některé předpoklady, učiněné při vzniku RM ISO/OSI, se nenaplnily
- například:
  - autoři předpokládali, že v každé zemi bude jedna velká síť vlastněná a provozovaná státem
    - důsledek: nebyl žádný důraz na tzv. internetworking, neboli na vzájemné propojování sítí
    - v TCP/IP byl od začátku velký důraz právě na internetworking
  - autoři předpokládali propojení uzlů výhradně pomocí dvoubodových spojení
    - linková vrstva původně neřešila sdílený přístup (když jde o sdílený spoj)
    - důsledek: došlo k rozdvojení linkové vrstvy na dvě podvrstvy

# RM ISO/OSI, spolehlivost a spojovanost

- spolehlivost:
  - RM ISO/OSI počítá primárně se spolehlivými přenosovými službami
- představa lidí od spojů:
  - nespolehlivou přenosovou službu (která může ztrácet data) by si nikdo nekoupil!!
  - proto i přenosová infrastruktura (do síťové vrstvy) musí fungovat spolehlivě
- představa lidí od počítačů:
  - zajištění spolehlivosti je spojeno s velkou reží
  - spolehlivost si efektivněji zajistí koncové uzly
- nespolehlivé přenosové služby se do RM ISO/OSI dostaly až dodatečně
  - na nátlak lidí od počítačů
- RM ISO/OSI počítal jen se spojovanými přenosovými službami
  - odpovídá to způsobu fungování telekomunikačních sítí, např. telefonní sítě
  - pro menší a méně pravidelné přenosy může být výhodnější nespojovaný charakter přenosu
- možnost nespojovaného přenosu se do RM ISO/OSI dostala až dodatečně

# RM ISO/OSI dnes

- RM ISO/OSI byl dlouhou dobu "oficiálním řešením"
  - státní správa chtěla řešit všechny sítě ve své kompetenci na RM ISO/OSI
    - nutné profily GOSIP a "zúžení"
    - nakonec se ukázalo jako neschůdné
- dnes je RM ISO/OSI "na hlavu poražen" rodinou protokolů TCP/IP
  - lze jej považovat za "odepsaný"
- dodnes je dobrým "učebnicovým příkladem"
  - toho jak se věci mají dělat
    - rozdělení do vrstev, rozdíl mezi službami, protokoly a rozhraními
  - toho, jak se věci nemají dělat
    - návrhy od zeleného stolu, velikášství, nepraktičnost, ....
- některé protokoly, vyvinuté v rámci RM ISO/OSI, byly různě převzaty a používají se i dnes
- příklady:
  - X.400 (řešení el. pošty)
    - velmi univerzální, bylo základem MS Exchange (uvnitř, nikoli navenek)
  - X.500 (adresářové služby)
    - "odlehčením" (zjednodušením) vznikl úspěšný protokol LDAP (Lightweight Directory Access Protocol)

