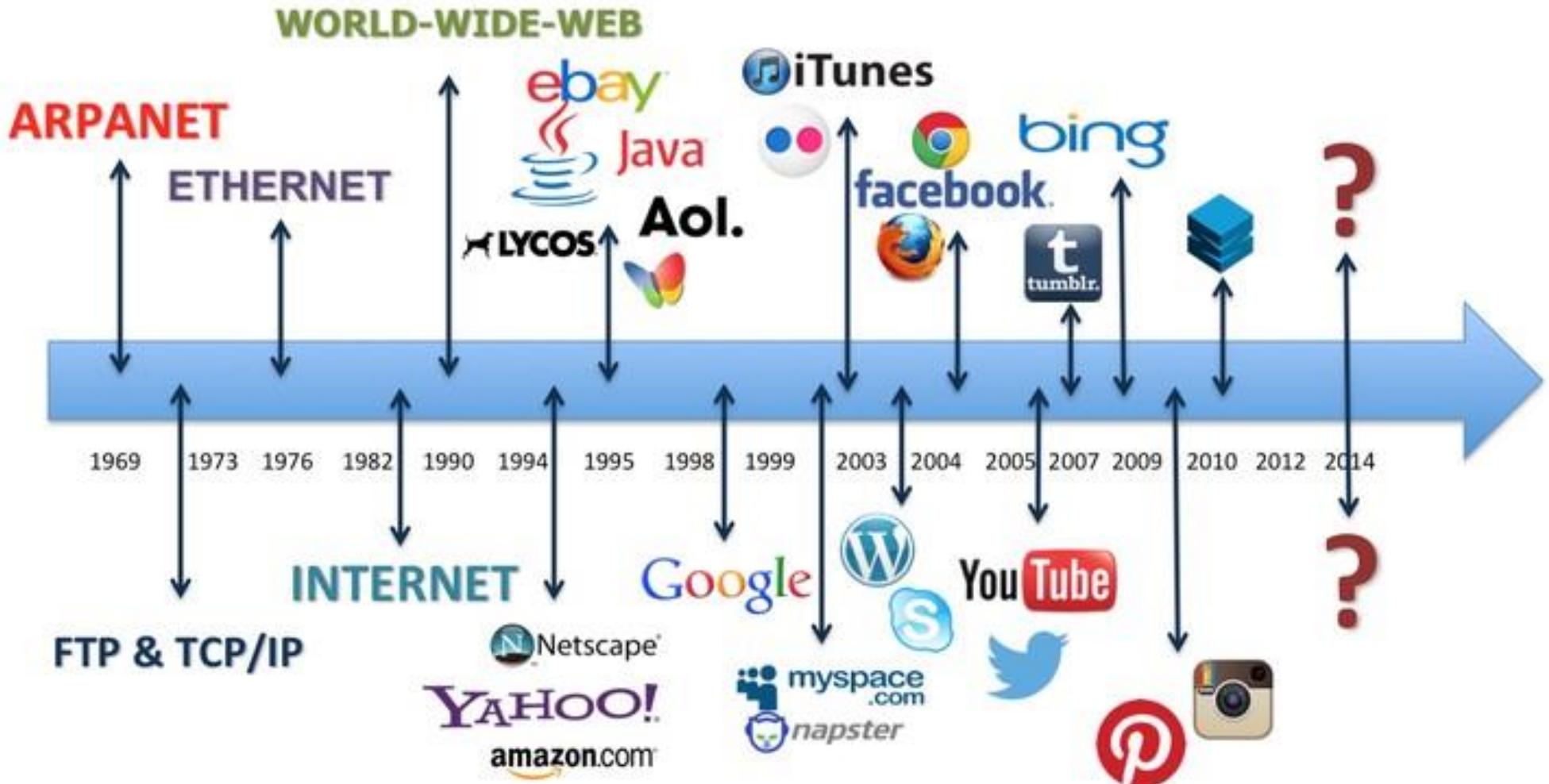


# Počítačové sítě

## Úvod do problematiky

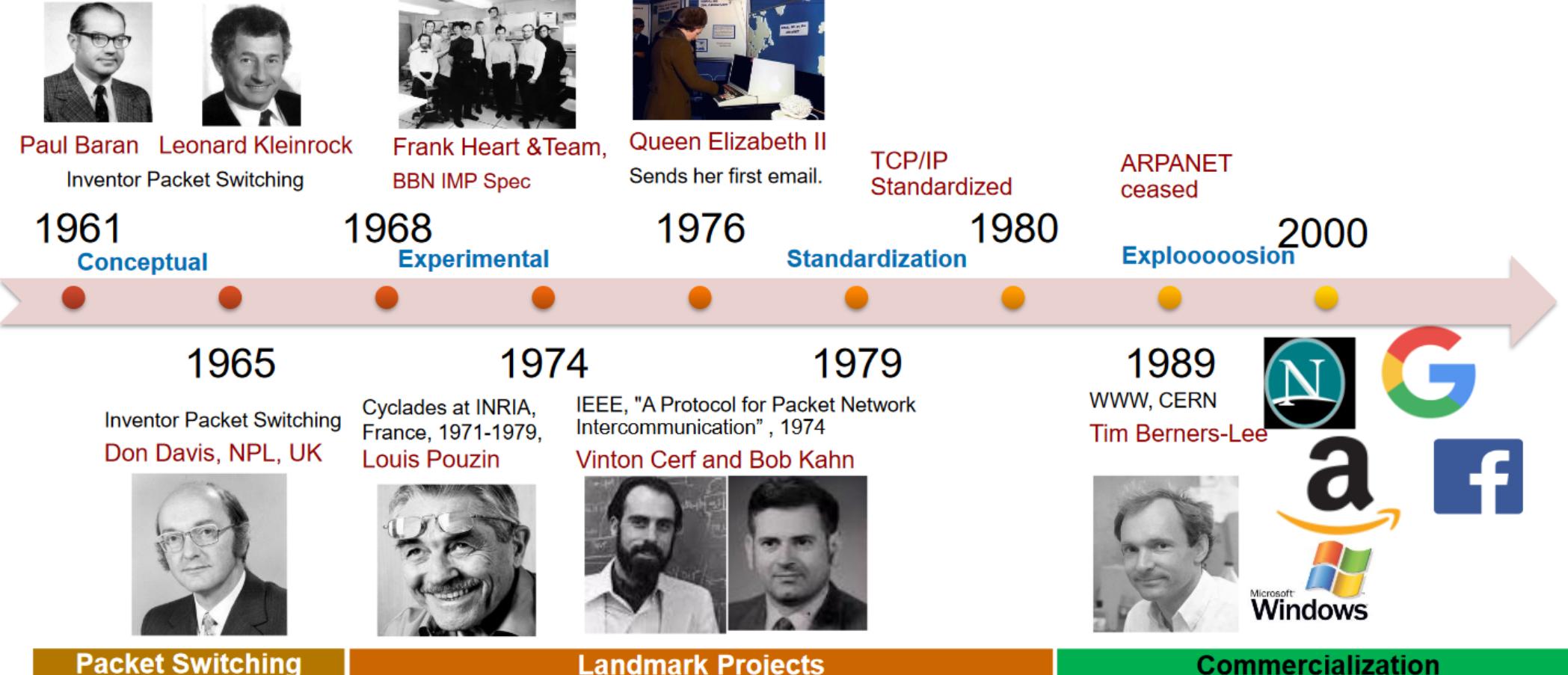


# Internet, sítě, proč ?



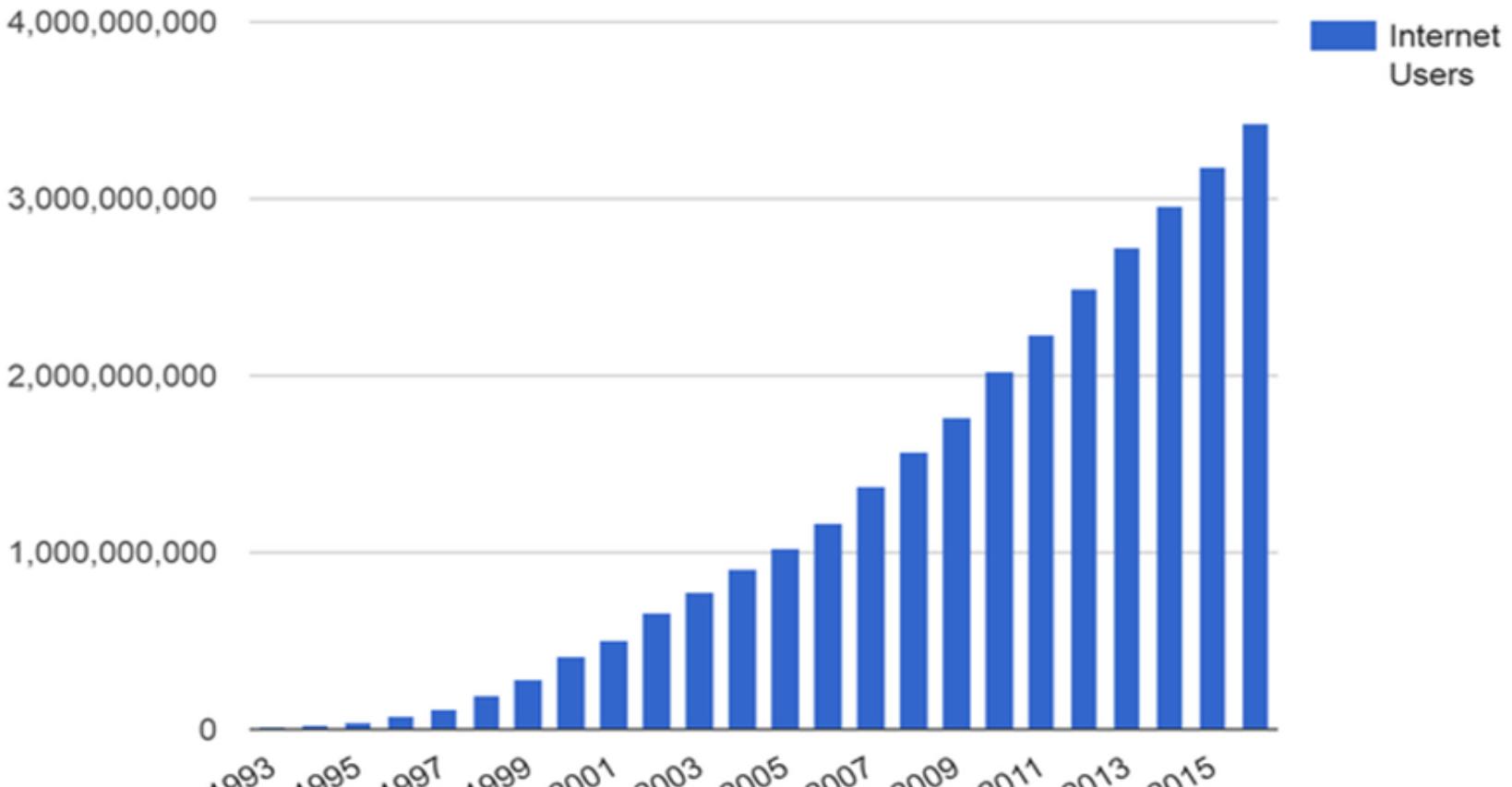
Zdroj: <https://socialoutside.blogspot.com/2017/10/timeline-of-social-media.html>

# Kdo byl "u toho" ?



Zdroj: <http://www.inmesol.com/blog/history-internet-told-creators-single-book>

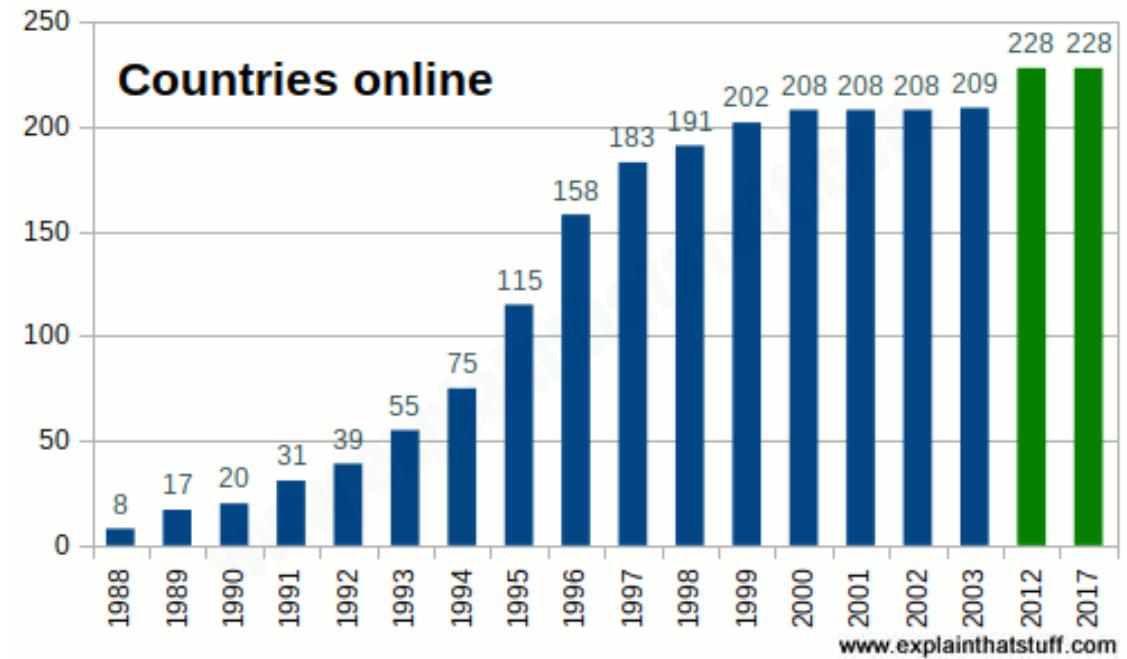
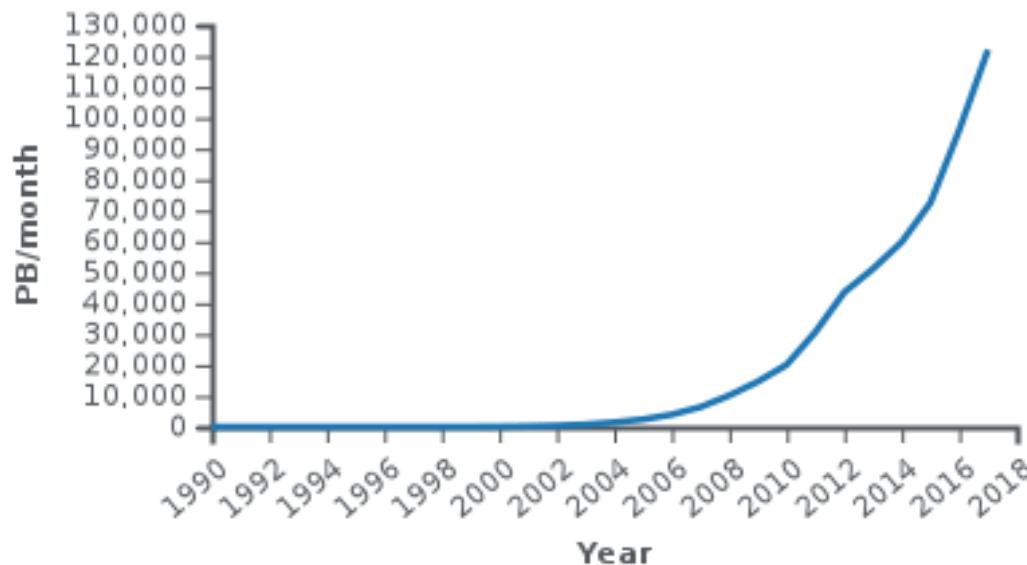
# Vývoj počtu uživatelů Internetu za posledních cca. 30 let



Zdroj: <https://www.websysanytime.com/Web3.php>

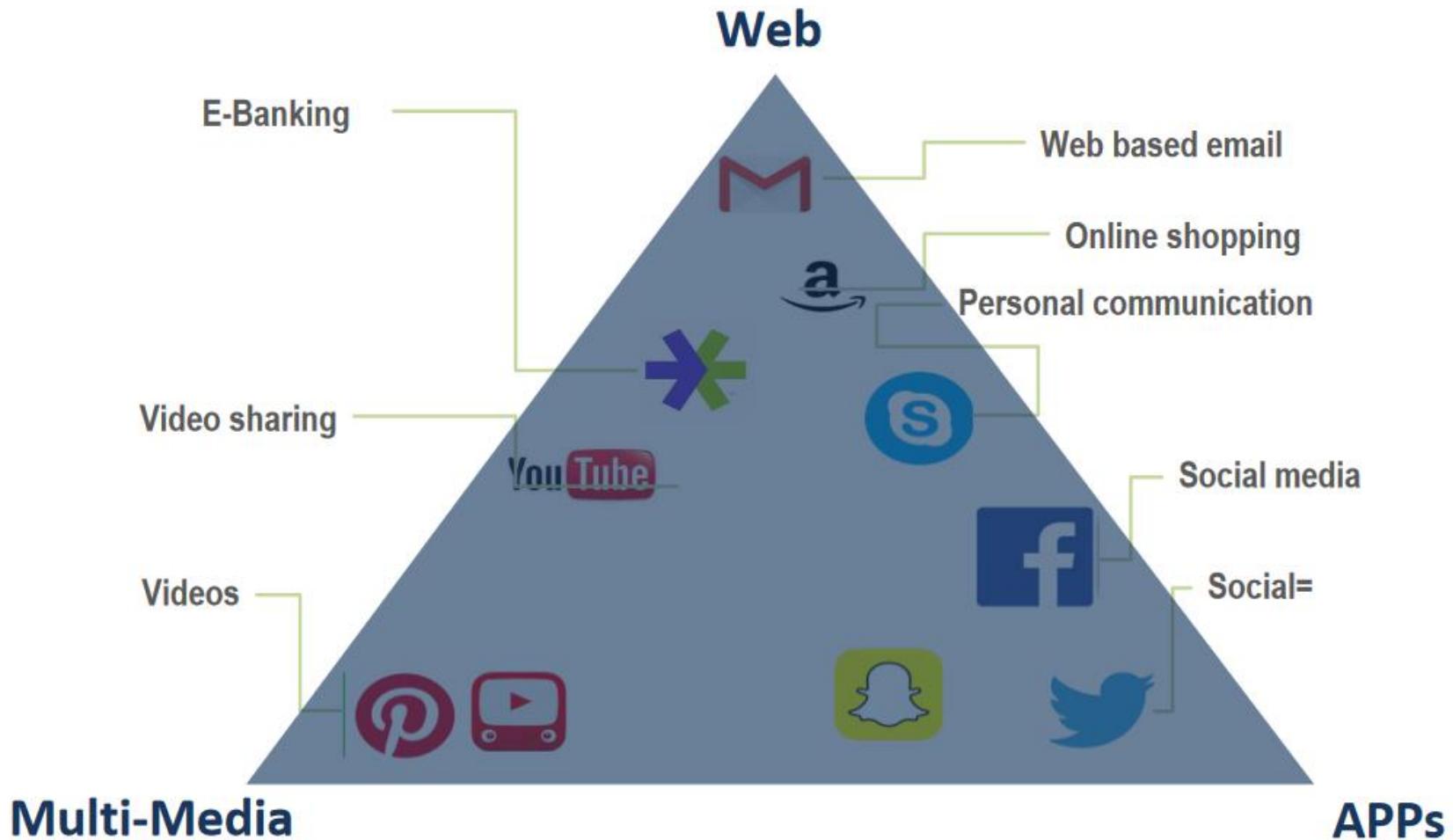


# Celkový průtok v internetu v posledních letech a připojené státy



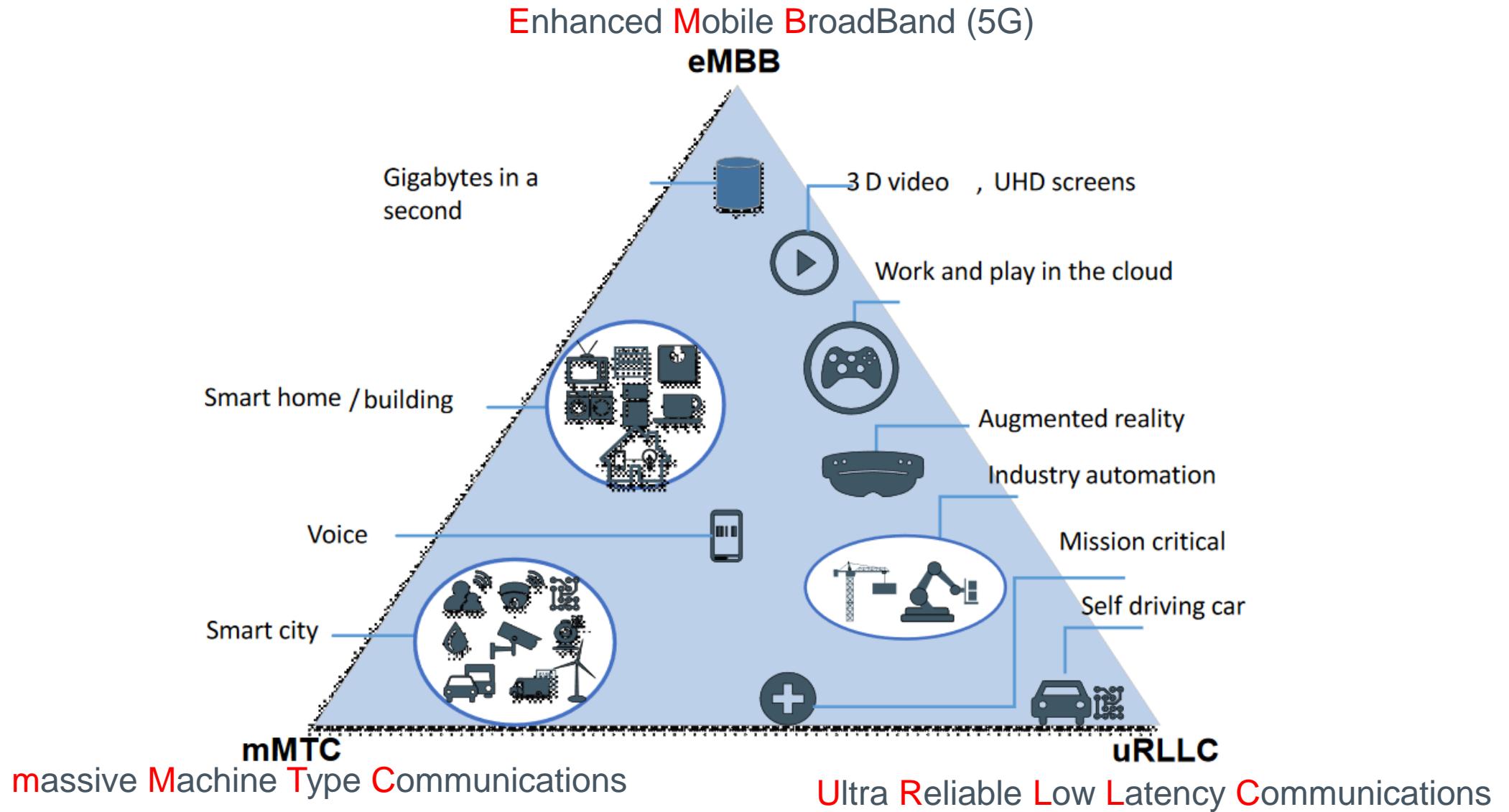
Zdroj: <https://www.websysanytime.com/Web3.php>

# Rozvoj služeb díky Internetu v posledních 20ti letech

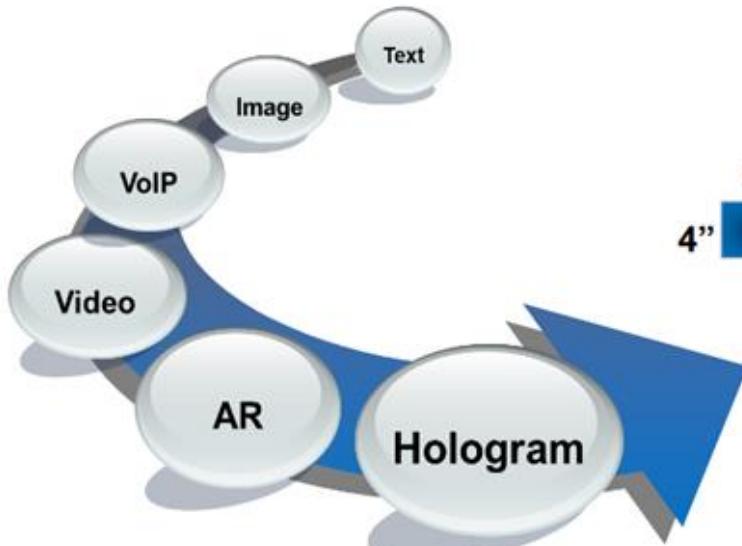


Zdroj: <https://steemit.com/technology/@brettu/internet-consumption-food-pyramid>

# Mírně vzdálená budoucnost ? Ani ne..



# Budoucnost ?



Dimensions	Bandwidth
Tile	4 x 4 inches
Human	72 x 20 inch

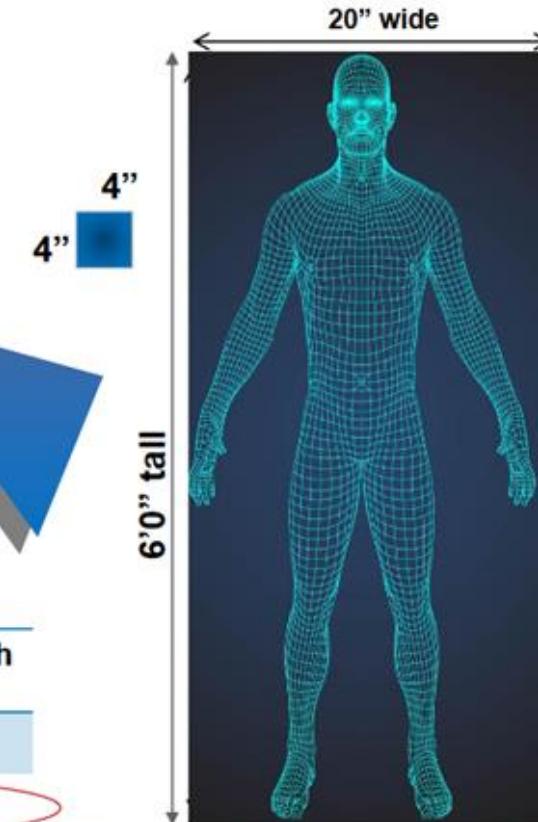
30 Gbps

4.32 Tbps

- Raw data; no optimization or compression.

- color, FP (full parallax), 30 fps

(reference: 3D Holographic Display and Its Data Transmission Requirement, 10.1109/IPOC.2011.6122872, derived from for 'Holographic three-dimensional telepresence'; N. Peyghambarian, University of Arizona)



Throughput goes up higher and higher



Latency falls down lower and lower



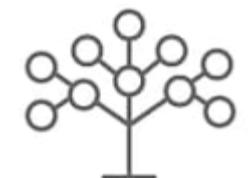
Synchronization of parallel streams



# Taxonomie sítí

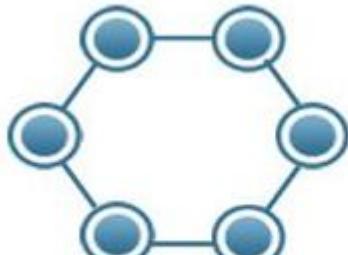


- **Přepojování** - jak se data šíří mezi dvěma stanicemi
  - **okruhů** (Connection-Oriented, circuit switching) - ATM, FrameRelay
    - Okruh = sestavená trasa, inspirace u analog. ústředen
  - **paketů** (Connectionless, packet switching) – Ethernet (dnes téměř všude)
    - Paket = balíček odesílaných dat
- **Architektura**
  - peer-to-peer (**P2P**)
  - client-server
- **Použití**
  - Veřejná, privátní, hybridní
- **Dle rozlehlosti**
  - Osobní - Personal Area Network (**PAN**) (~ 1 m)
  - Místní - Local Area Network (**LAN**) (~ 100 m)
  - Městské - Metropolitan Area Network (**MAN**) (~ 10 km)
  - Rozlehlé - Wide Area Network (**WAN**) (~ 1000 km)



# Základní síťové topologie

- **Topologie** je způsob uspořádání propojení (linek) mezi stanicemi (uzly) v síti
- Vhodná volba použité topologie souvisí s konkrétním použitím



Kruh



Hvězda



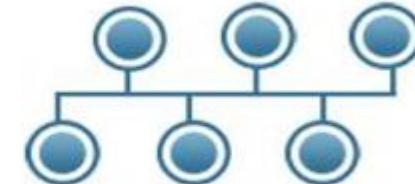
Plně propojená



Přímá



Strom

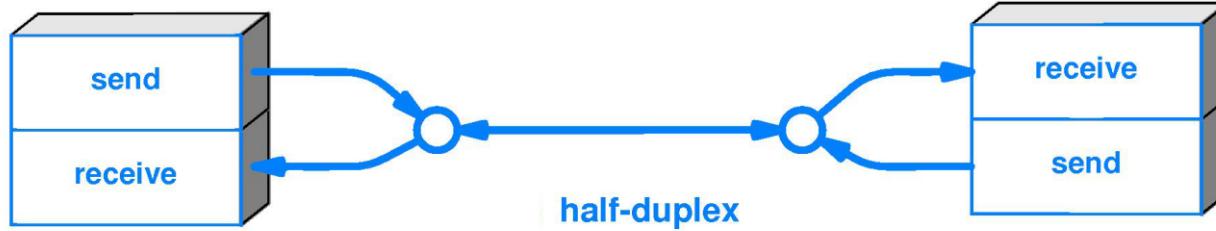
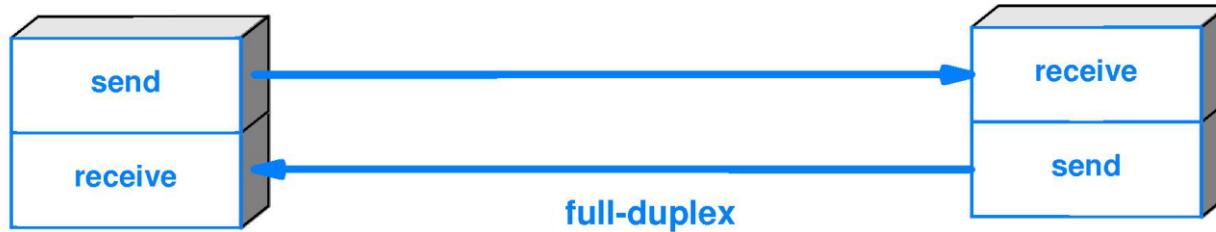
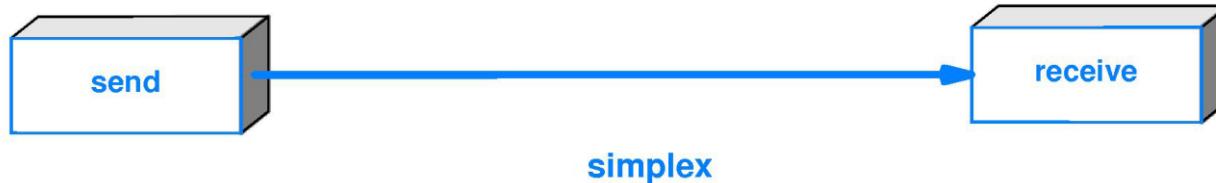


Sběrnice

Základní topologie lze (a dělá se to) kombinovat do sebe → **hybridní** topologie.

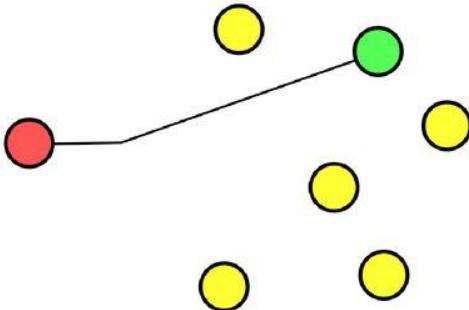


# Druhy linek v počítačových sítích dle možností vysílání a příjmu

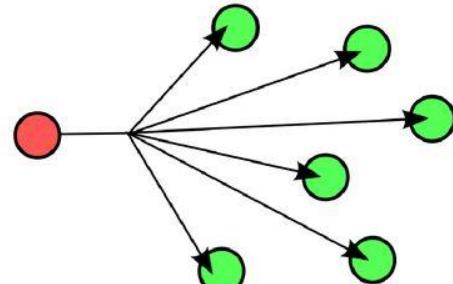


Zdroj: <https://www.black-box.de/en-de/page/25078/Resources/Technical-Resources/Black-Box-Explains/Fibre-Optic-Cable/simplex-vs-duplex-fiber-patch-cable>

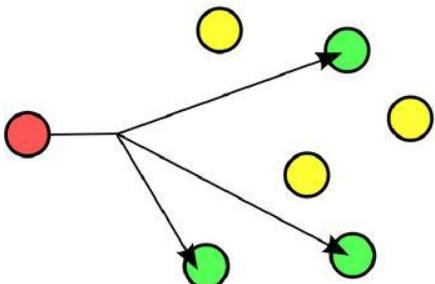
# Základní druhy komunikačních operací v počítačových sítích



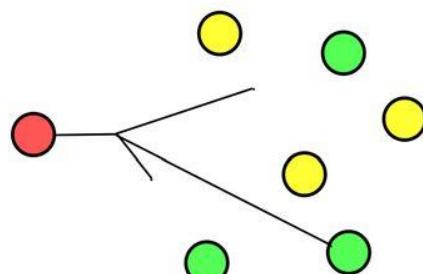
Unicast



Broadcast



Multicast



Anycast

- Vysílač
- Uzel (nezúčastněný)
- Přijímač

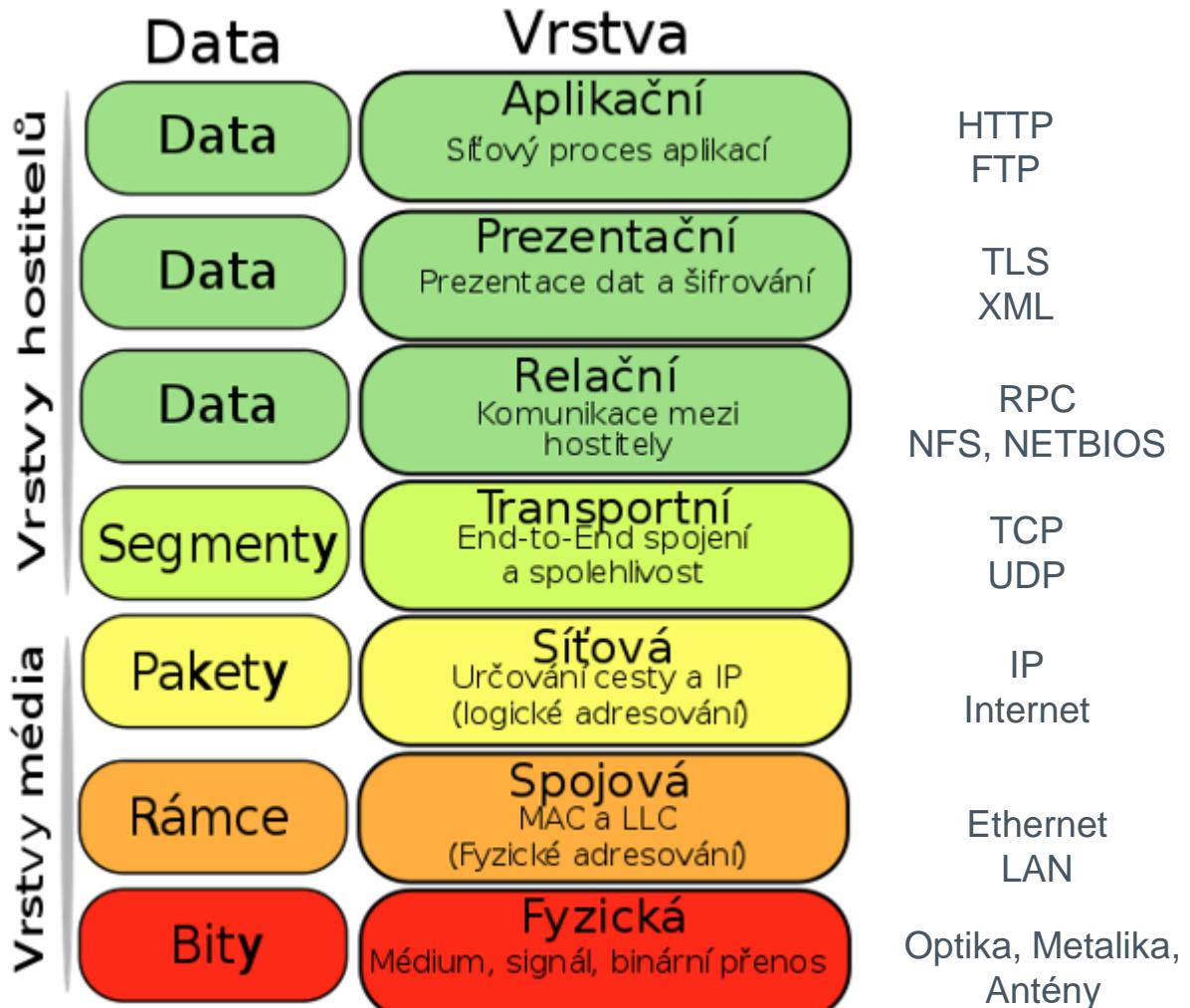
Způsob a rychlosť provedení dané operace závisí na konkrétní topologii.



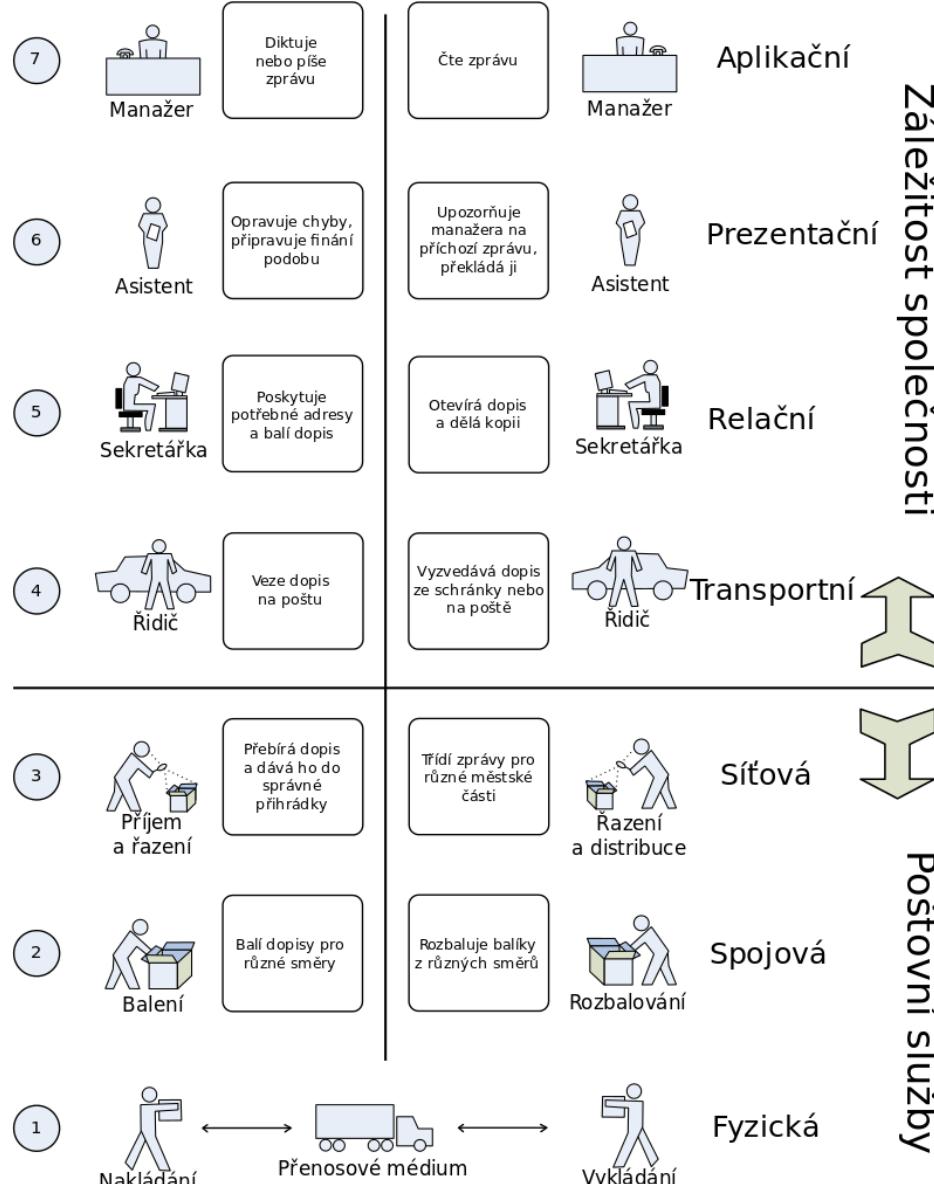
# Návrh konceptu počítačových sítí

- Idea pochází z poloviny 80. let minulého století.
- Návrh obsahuje **7. hierarchických funkčně disjunktních úrovní** (vrstev), které vzájemně spolupracují.
- Každá **vrstva** má svou **specifickou funkci**.
- Vrstvy jsou záměrně oddělené, toto umožňuje jejich nezávislý vývoj a implementaci (nicméně existují i “mezivrstevní” technologie jako např. MPLS).
- Konkrétní hardwarové technologie (Ethernet, Wifi, Bluetooth atd.) jsou vždy definovány pro více vrstev (rozhodně ale ne pro všechny).
- Pro dané vrstvy jsou definovány **komunikační protokoly** – scénáře komunikace mezi dvěma stranami, které se liší dle konkrétní vrstvy.
- Pro protokoly je důležitá standardizace (RFC, pokud není → problémy).

# Open System Interconnection (OSI) Model



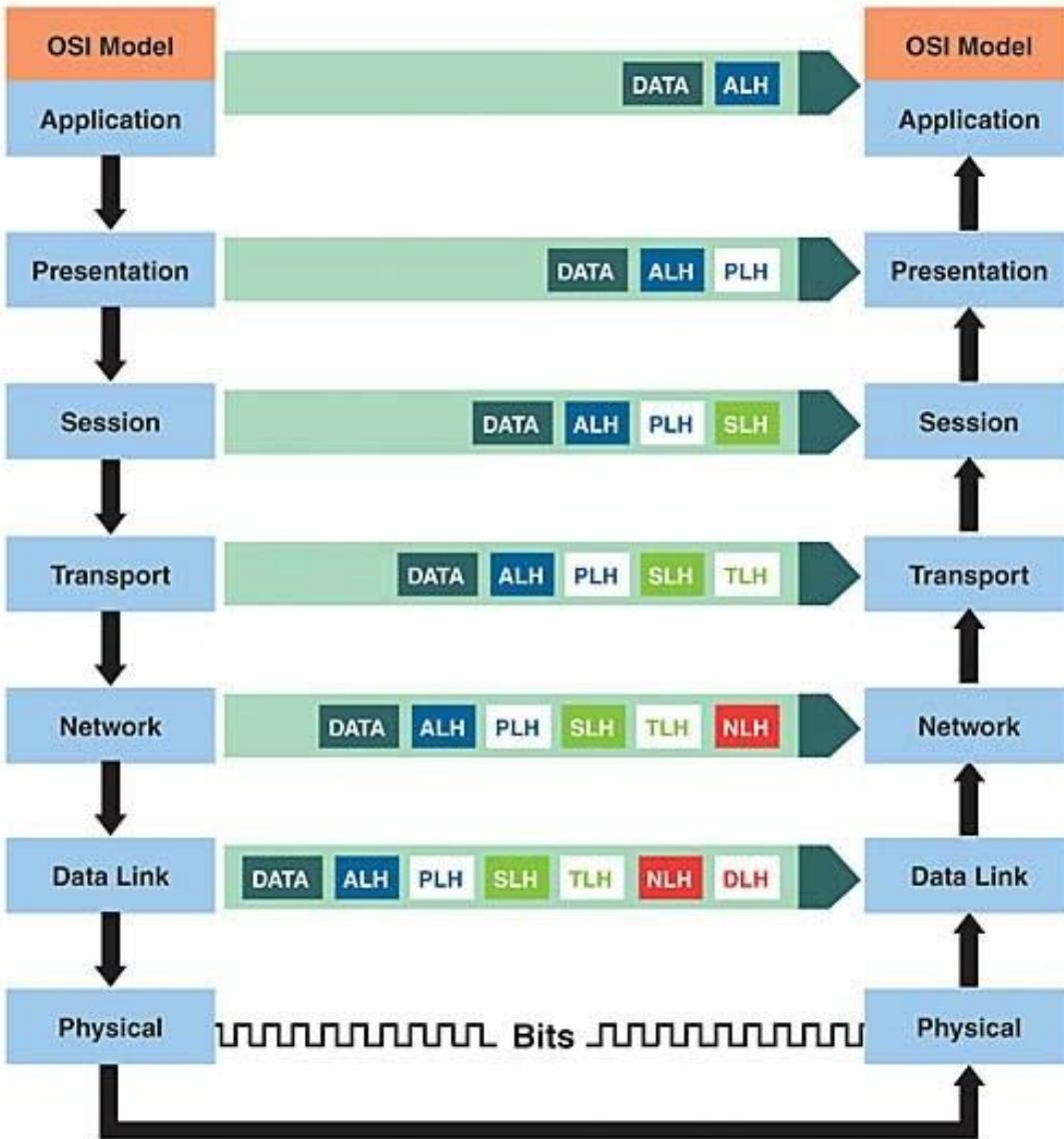
**Médium** – prostředí, ve kterém se přenášejí data (kabel, vzduch, atd.)



# Enkapsulace a Dekapsulace posílaných dat



A



Zdroj: <https://spyvision9.blogspot.com/search/label/decapsulation>

B

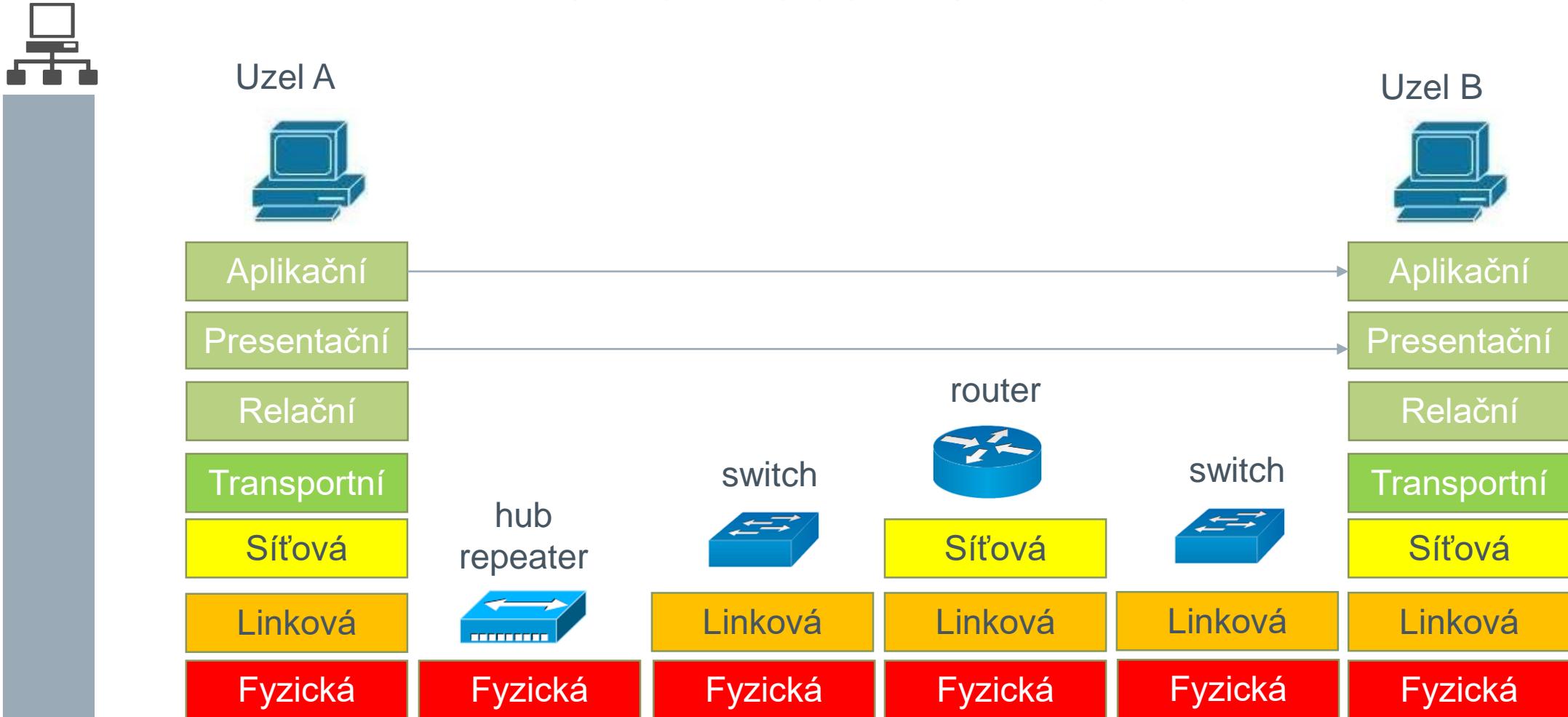
**A** – Application, **P** – Presentation,  
**S** – Session, **T** – Transport, **N** –  
Network, **D** – Data Link, **P** – Physical  
**LH** = Layer Header

**PDU** = Protocol Data Unit  
Protokolová datová jednotka

PDU = hlavička + obsah  
PDU7 = ALH+DATA  
PDU6 = PLH+PDU7 = PLH +ALH+DATA  
PDU5 = PLS +PDU6

Komunikace mezi stanicemi A a B  
probíhá vždy na úrovni stejných  
vrstev!!!

# Komunikace mezi vrstvami



**Router = směrovač, Switch = přepínač, Repeater = opakovač, Hub = rozbočovač**

Použité grafické symboly se běžně používají pro popis výše uvedených zařízení.

# Počítačové sítě

## TCP/IP model a adresace



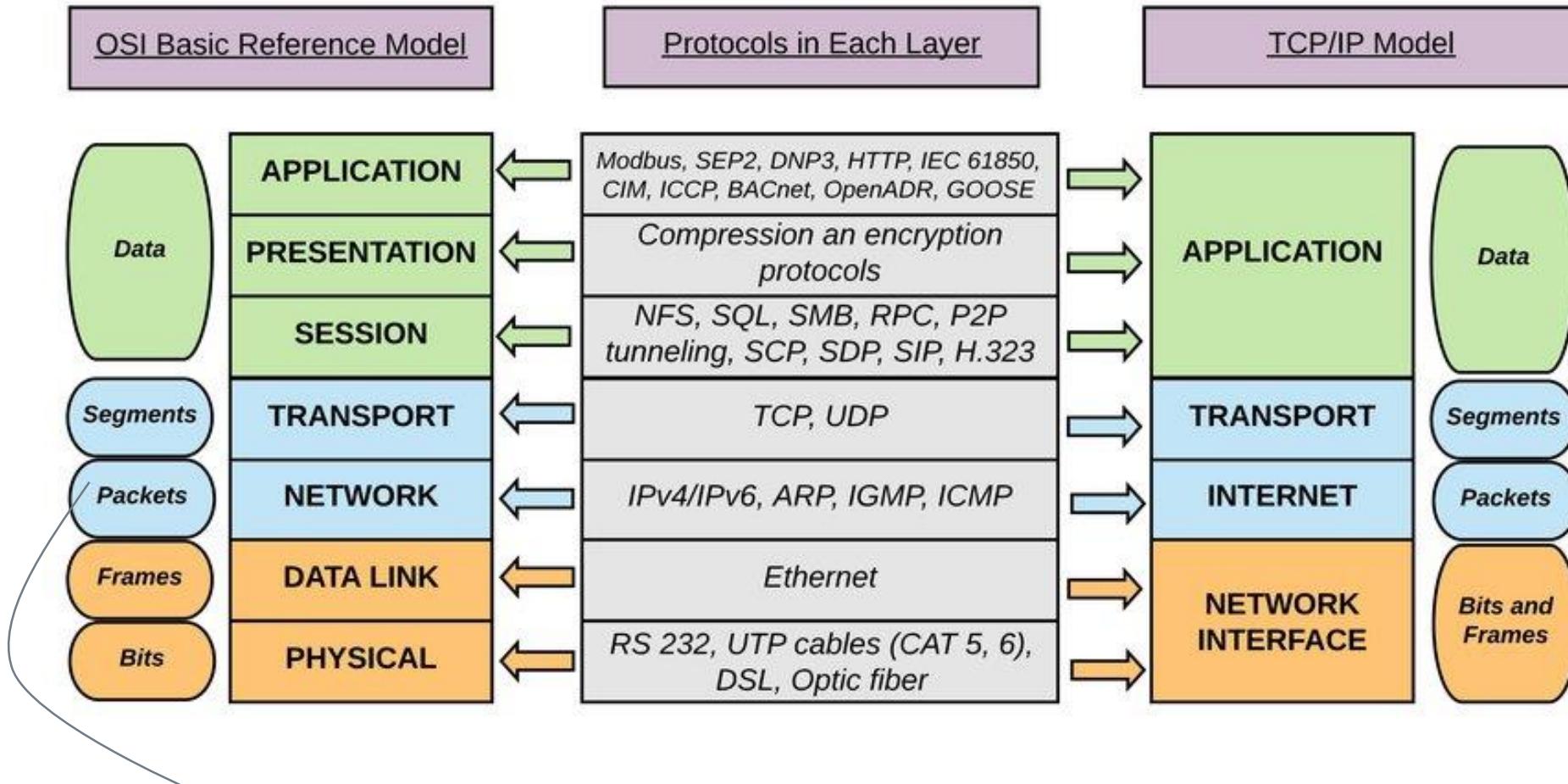


# K čemu je to dobré?

- Internet Protokol (**IP**) je základní komunikační protokol, který umožňuje doručování dat mezi stanicemi (uzly).
- IP protokol přiřazuje konkrétní stanici jednoznačný identifikátor (**IP adresu**).
- IP adresa náleží vždy konkrétnímu Network Interface Controller (NIC) (sítovému zařízení).
- IP je **hierarchický**, má logickou strukturu → umožňuje vytvářet rozsáhlé topologie, spojovat adresy ve větší celky.
- Pro IP protokol existují algoritmy díky kterým lze provádět optimální směrování mezi stanicemi.
- IP protokol je implementován na drtivé většině zařízení využívající sítové připojení (např. mobily, tablet či IoT prvky).

# TCP/IP model vs OSI model

Zdroj: [https://www.researchgate.net/figure/The-logical-mapping-between-OSI-basic-reference-model-and-the-TCP-IP-stack\\_fig2\\_327483011](https://www.researchgate.net/figure/The-logical-mapping-between-OSI-basic-reference-model-and-the-TCP-IP-stack_fig2_327483011)





# Princip IP adresace

- Hlavní motivací adresace je přiřazení “jmen” pro všechny počítače v síti dle nějakého jednoznačného schématu.
- Jednoduše by stačilo přiřadit každému unikátní číslo.
- Každé číslo lze zapsat ve dvojkové soustavě (pro počítače nutnost).
- Dle počtu bitů ( $N$ ), které použijeme na zakódování můžeme určit maximální počet ( $p$ ) přiřaditelných adres  $p = 2^N$ .
- Otázkou (v 80.letech) bylo, jaké  $N$  zvolit pro adresaci v Internetu.
- 32 se zdálo dost, odpovídá to cca.  $4 \times 10^9$  adresám → IPv4.
- Okolo roku 1995 se ukázalo, že to dost nebylo a nové  $N$  bylo zvoleno 128 ( $3 \times 10^{38}$ ) → IPv6.



# Adresace v IPv4

- Adrese s **32** bity (4 byty) se říká IPv4 adresa.
- Nicméně zapamatovat si číslo, které má  $32 \times (0 \text{ nebo } 1)$  za sebou je pro lidi složité.
- Používá se tudíž zápis po bytech (oktetech), každý byte může mít hodnotu 0-255 (zapsaných v desítkové soustavě).

XXX.XXX.XXX.XXX

123.123.123.123

01111011011110110111101101111011



# Síťování neboli segmentace

- Počítače, komunikující mezi sebou prostřednictvím IP adres je výhodné sdružovat do určitých skupin (sítí či segmentů).
- Celý adresní rozsah sítě se běžně označuje jako oblast **lokální sítě**.
- Pro určení velikosti skupiny se používá **síťová maska**, která má stejnou velikost (v bitech) jako **IP adresa**.
- **Maska říká to, jakou část IP adresy mají všichni členové dané skupiny (sítě) společnou.**
- Výhodná a intuitivní je prefixová notace -> viz dále.

# Prefixová notace u IPv4



- Tvar zápisu je obecně **IP/PREFIX**, kde prefix označuje počet bitů, které jsou společné pro všechny členy stejné sítě.
- Pro adresu 123.123.123.123 zvolíme např. prefix = 30.
- Pro síť bude platit, že 30 bitů adresy bude pro všechny stanice společný a zbytek ( $32-30=2$  byty) bude určovat konkrétní stanici.
- **2 bity** umožňují vytvořit  **$2^2$**  různých adres

011110110111101101111011011110|00 (120)

011110110111101101111011011110|01 (121)

011110110111101101111011011110|10 (122)

011110110111101101111011011110|11 (123)

Jako **adresa sítě** se uvádí vždy nejnižší možná adresa (bity za prefixem jsou nulové), pro tento případ tedy **123.123.123.120/30**.

# Prefix a maska sítě



- Maska sítě, která musí být rozměrově shodná s IP adresou, se z prefixu odvodí tak, že hodnota prefixu odpovídá počtu jedniček na začátku masky za sebou a zbytek jsou nuly.
- Prefix /30 odpovídá masce:

Prefix  
→  
**11111111 11111111 11111111 111111|00**  
**255.255.255.252**

Hodnota posledního oktetu masky lze vypočítat jako

$$256 - 2^{32-\text{délka prefixu}}$$

# Maska a adresa sítě

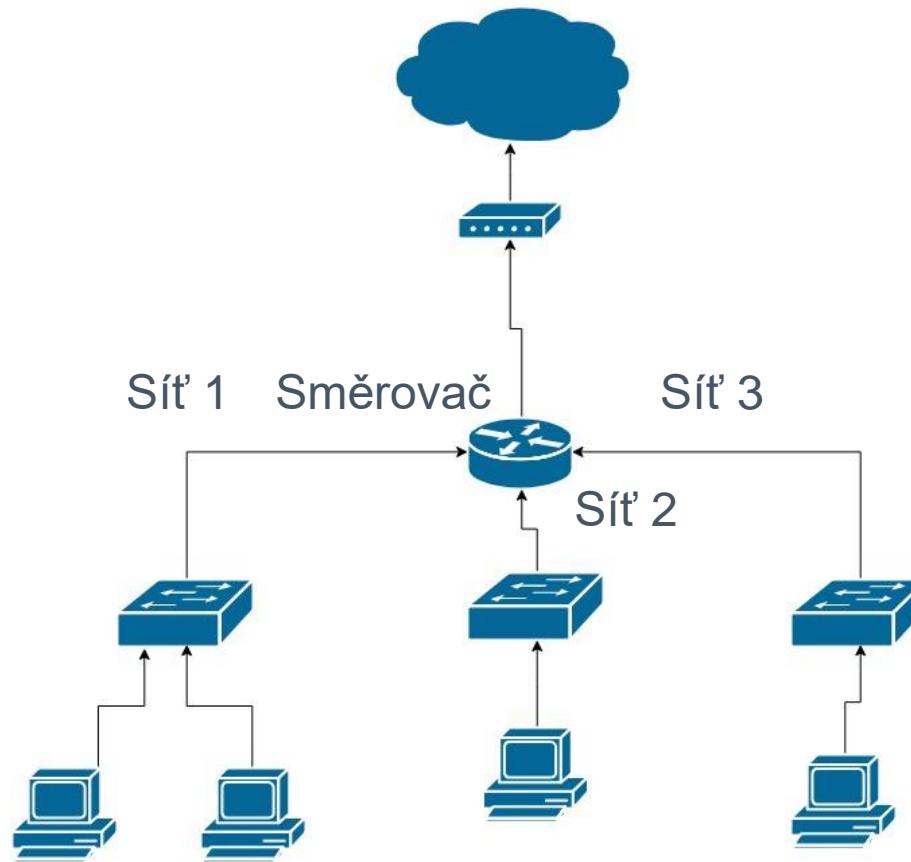


- Adresa sítě je označení všech IP adres, které patří do daného adresního rozsahu.
- Masku jedničkami označuje to, co stanice ve stejné síti mají v IP adrese mají společné a nulami v čem se liší.
- Pokud uděláme operaci **IP adresa AND Maska po bitech**, získáme adresu sítě (1. adresu skupiny), tato se běžně používá pro směrování.

AND      011101101110110111011011110|11 (123.123.123.123)  
          11111111111111111111111111111111|00 (255.255.255.252)  
          011101101110110111011011110|00 (123.123.123.120)

Jak poznáme, že dvě IP adresy pro zadaný prefix patří do stejné sítě ?

# Sít, brána, broadcast



**Směrovač paří do IP rozsahů všech připojených sítí.**

**Brána** je implicitní směrovač, přes který jde komunikace mimo lokální síť.

**Broadcast** je zpráva, kterou obdrží všechny stanice v síti mimo původce.



# Mapování broadcastu a brány do adres v IPv4, rozsah sítě

**Adresou sítě = nejnižší adresa** v daném rozsahu.

**Broadcast = nejvyšší adresa** v daném rozsahu.

**Brána** má typicky (nikoli povinně) **druhou nejvyšší či nejnižší adresu** v daném rozsahu.

Pro nás případ **prefix = 30**

123      123      123

011110110111101101111011011110|00 (120) Adresa sítě

011110110111101101111011011110|01 (121) Adresa pro stanici

011110110111101101111011011110|10 (122) Brána

011110110111101101111011011110|11 (123) Broadcast

Délka prefixu (L) sítě závisí na počtu stanic, které chceme do tohoto umístit.

Pro nalezení minimálního N, umožňujícího adresovat K adres musí platit  $K \leq 2^N$ ,

$L = 32-N$

Počet využitelných adres v síti je  $2^N-2$  (broadcast a adresa sítě dvě seberou).

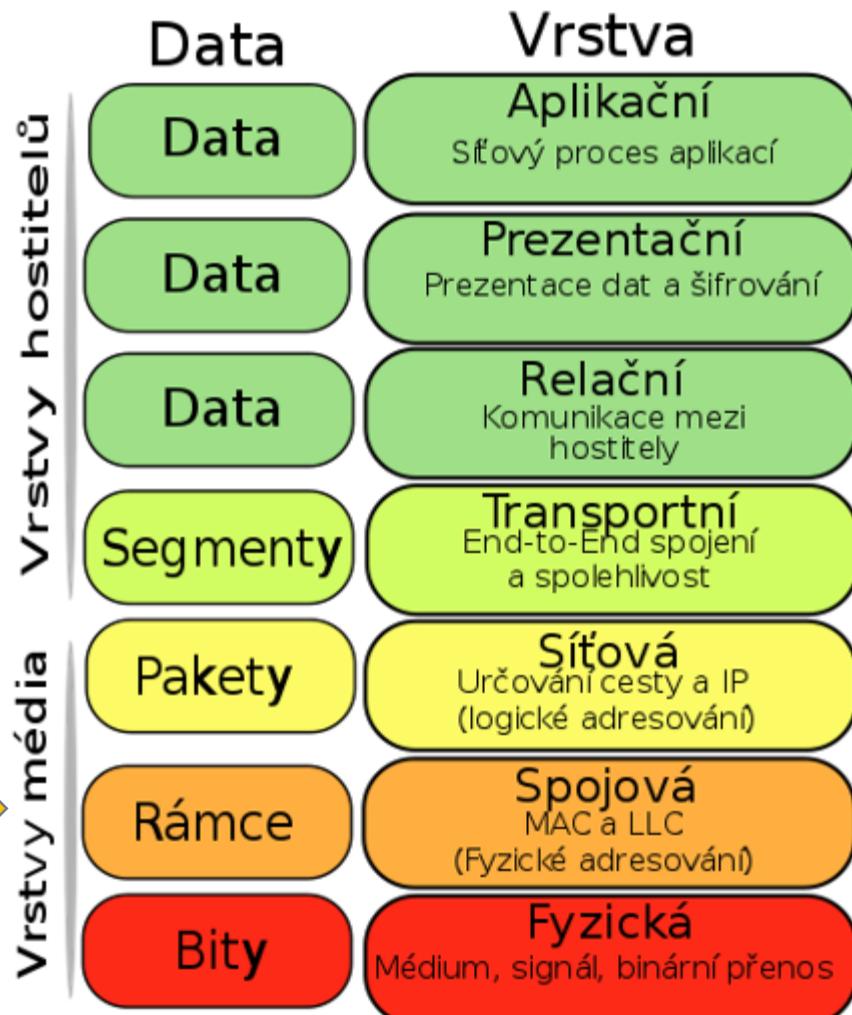
Počet využitelných adres v síti bez brány je  $2^N-3$  (broadcast, adresa sítě, brána).

# Počítačové sítě

## 2. Přednáška - linková vrstva



# Linková vrstva v OSI modelu a její účel



- Hlavní **účelem** linkové vrstvy je **přenášení dat v rámci lokální (LAN) sítě**.
- Základní přenášenou jednotkou je **rámec**.
- Linková vrstva se skládá z **dvou podvrstev** (MAC a LLC).
  - Podvrstva **Medium Access Control (MAC)**
    - Zajišťuje přístup k médiu, multiplex (viz dale).
    - Definuje **linkové adresy (MAC adresy)**.
    - **Je** hardwarově závislá.
  - Podvrstva **Logical Link Control (LLC)**
    - Kódování.
    - Zajišťuje logické řízení toku.
    - Definuje potvrzovací schémata pro spolehlivé doručování rámců.
    - **Není** hardwarově závislá.



# Medium Access Control (MAC) podvrstva

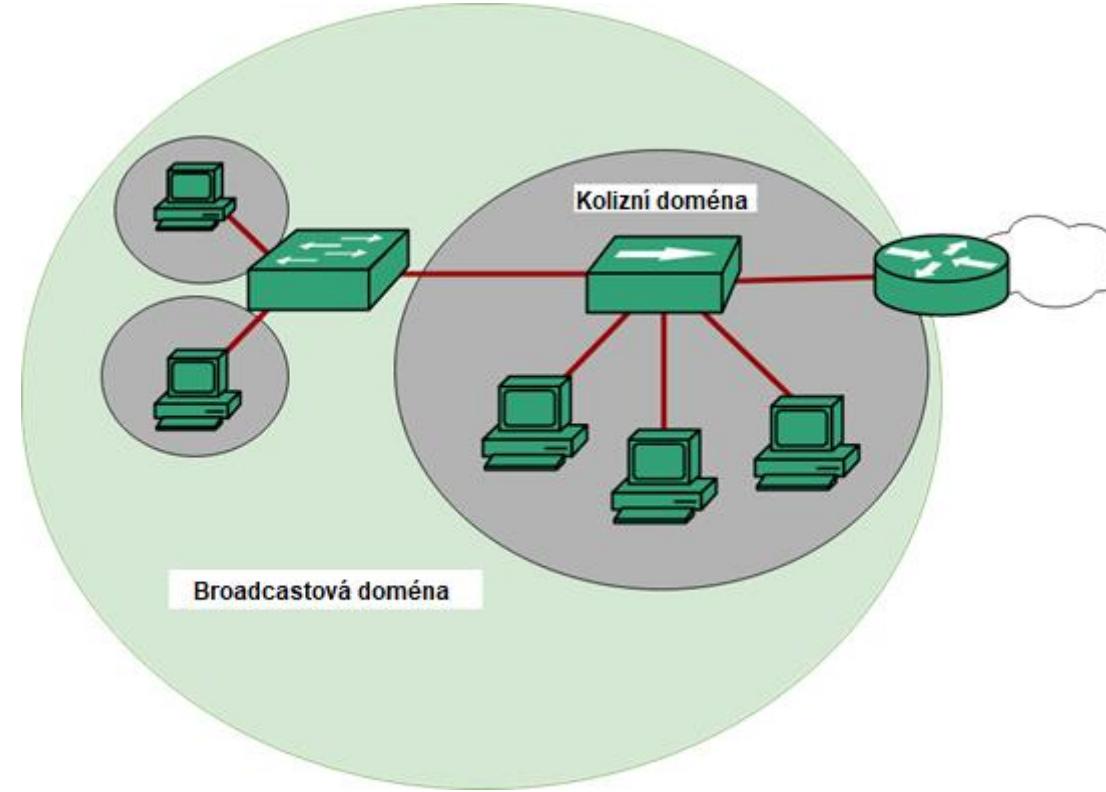
- Tato vrstva zajišťuje
  - Předávání dat fyzické vrstvě.
  - **Přístup k médiu** (může být nesdílený nebo sdílený).
  - **Sdílený přístup** k médiu se řeší pomocí **multiplexu** popř. **přístupových metod**.
  - Fyzickou adresaci prostřednictvím MAC adres.
  - Filtrování MAC adres (kvůli bezpečnosti).
  - Přepínání a doručování rámců (mezi zařízeními).
  - Ukládání rámců do front a plánování jejich odesílání.
  - Kvalitu služeb (Quality of Service).
  - Vytváření virtuálních sítí (VLAN).
- Její popis je vždy svázán vždy s konkrétní technologií, která je použita na úrovni **fyzické vrstvy** (metalika, optika, bezdrát).



# Multiplex, jeho význam a druhy

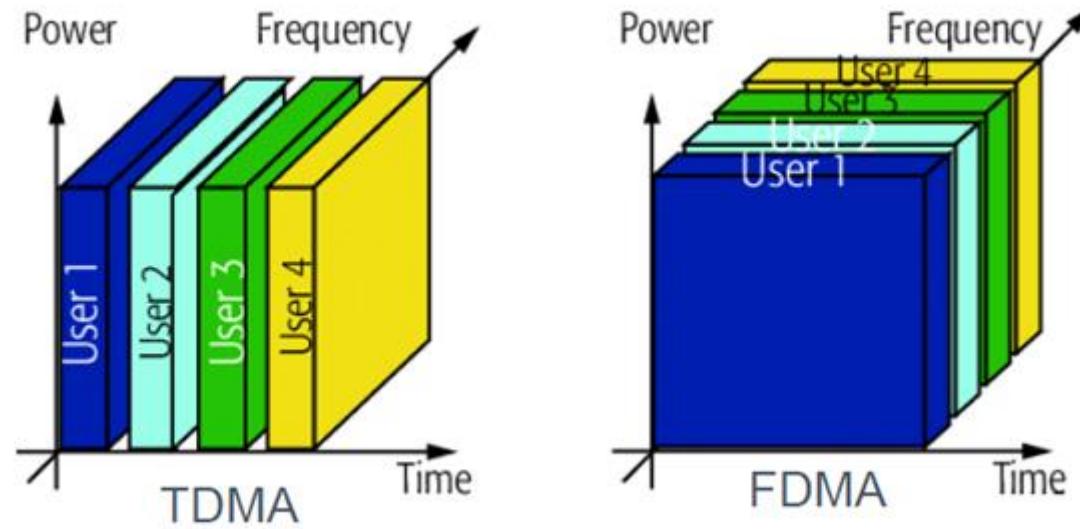
- **Multiplex** = technika umožňující současné použití stejného přenosového média více účastníky (stanicemi, počítači).
- Přístup více účastníků k jednomu přenosovému médiu se označuje jako **mnohonásobný přístup (Multiple Access)**.
- Pokud nastane **současné neoddělitelné použití** média více účastníky, dojde ke **kolizi**. Tato může vzniknou v celé **kolizní doméně** (viz dale).
- Možnosti, jak sdílet přenosové médium, je více (různé principy).
- Nejběžnější typy multiplexu jsou tyto:
  - Časový (**Time Division Multiple Access**)
  - Frekvenční (**Frequency Division Multiple Access**)
  - Kódový (**Code Division Multiple Access**)
  - Prostorový (**Space Division Multiple Access**)

# Kolizní doména



- **Kolizní doména (Collision Domain)** je množina stanic, které sdílí společné médium. Současné vysílání 2 anebo více stanic vede ke kolizi. Pojem broadcastová doména (Broadcast Domain) bude vysvětlen později.

# Princip časového a frekvenčního multiplexu



Zdroj: [https://www.researchgate.net/figure/Illustrative-example-of-different-multiple-access-schemes-a-TDMA-b-FDMA-c-OFDMA-d\\_fig4\\_323141497](https://www.researchgate.net/figure/Illustrative-example-of-different-multiple-access-schemes-a-TDMA-b-FDMA-c-OFDMA-d_fig4_323141497)

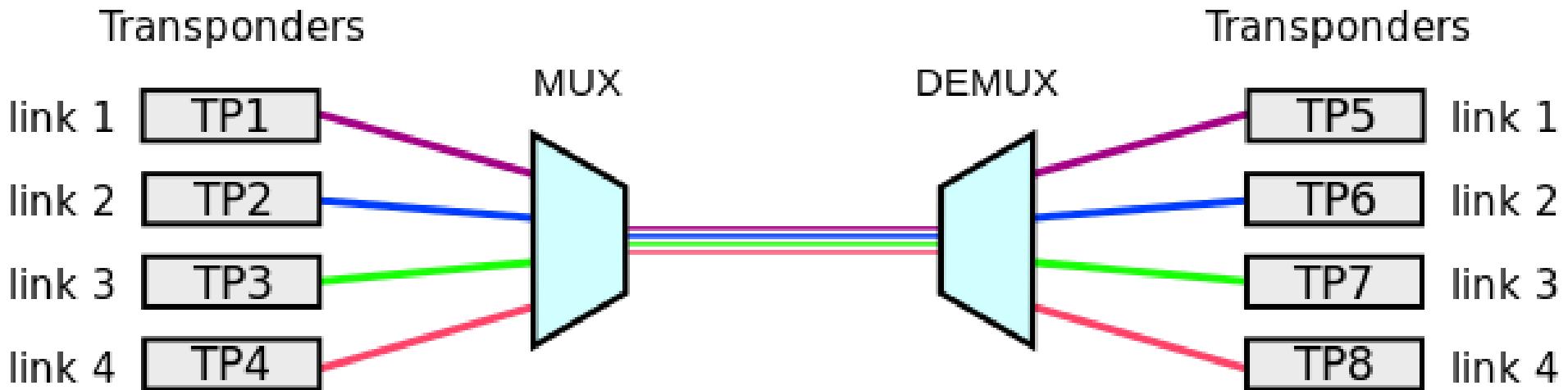
**Časový multiplex (TDMA)** znamená, že médium využívá v konkrétním časovém okamžiku pouze jeden účastník. Účastníci se při využití média střídají v definovaných časových intervalech.

**Frekvenční multiplex (FDMA)** znamená, že médium využívá současně několik různých účastníků, avšak každý z nich používá jiné frekvenční pásmo - díky čemuž se neovlivňují.



# Frekvenční multiplex používaný v optických sítích

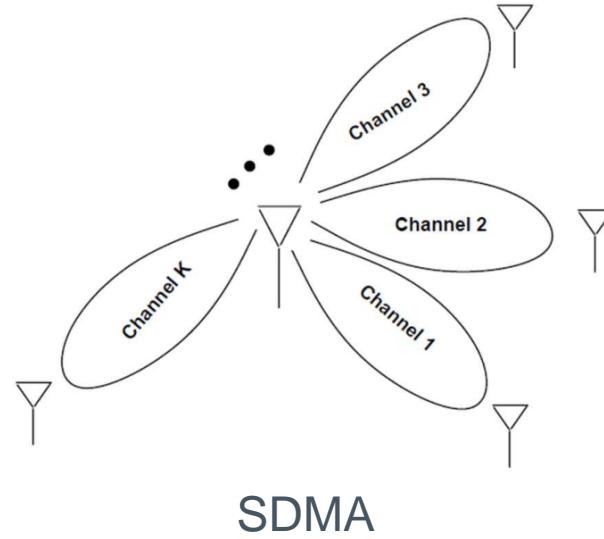
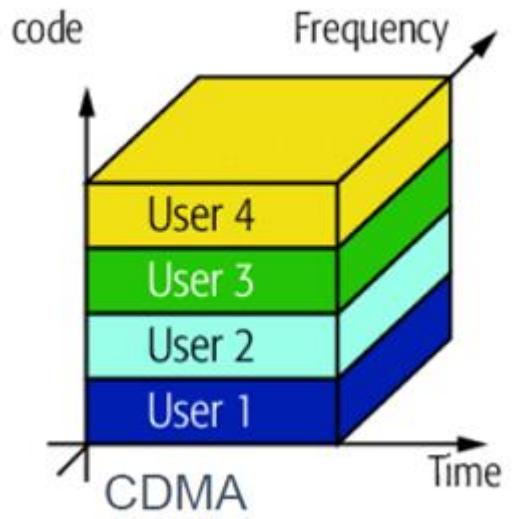
V optických sítích se pro přenos informací používá světelných paprsků. Světlo je elektromagnetické vlnění o určité frekvenci, která odpovídá vlnové délce (wavelength) a barvě.



Zdroj: <https://www.fiber-optic-solutions.com/what-is-wdm.html>

Princip multiplexu spočívá v tom, že do optického vlákna svítí více různobarevných zdrojů světla a každá barva nese samostatnou informaci. Na vstupu do média jsou jednotlivé barvy složeny pomocí multiplexoru (MUX) a na výstupu opět rozděleny pomocí demultiplexoru (DEMUX). Tento multiplex se označuje jako **WDM** (Wavelength Division Multiplex).

# Princip kódového a prostorového multiplexu



Zdroj: [https://www.researchgate.net/figure/Illustrative-example-of-different-multiple-access-schemes-a-TDMA-b-FDMA-c-OFDMA-d\\_fig4\\_323141497](https://www.researchgate.net/figure/Illustrative-example-of-different-multiple-access-schemes-a-TDMA-b-FDMA-c-OFDMA-d_fig4_323141497)

**Kódový multiplex (CDMA)** znamená, že médium využívá v konkrétním časovém okamžiku více účastníků. Účastníci, však zpracovávají pouze to, čemu rozumějí (svému kódu).

**Prostorový multiplex (SDMA)** znamená, že médium využívá současně několik různých účastníků, díky tomu, že konkrétní vysílání (může být i na stejně frekvenci) prochází různými vzájemně se nepřekrývajícími směry.



# Metody s příposlechu přenosového média

- Účel těchto metod je stejný jako v případě multiplexu - zabránit kolizím. Principy těchto metod jsou však oproti principům multiplexů složitější.
- Základem je naslouchání (Carrier Sense) stavu přenosového média.
- Prvotní inspirací byly staré metody Aloha (avšak ty byly bez příposlechu).
- Existují 2 základní varianty CSMA (M = Multiple, A = Access) metod.
  - CSMA/CD – Collision Detection (dokáže detektovat kolizi).
  - CSMA/CA – Collision Avoidance (dokáže se vyvarování se kolizi).

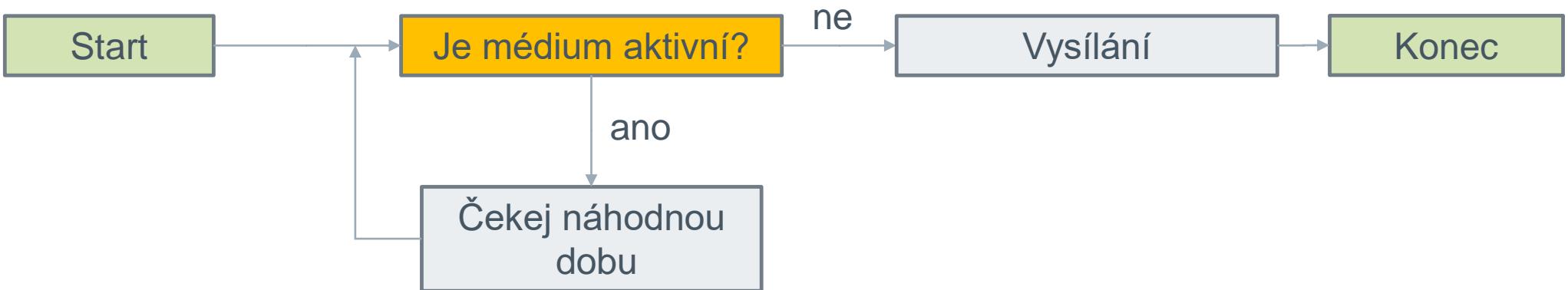
# Princip metody CSMA/CD



Jam signál má za úkol informovat všechny účastníky, že ke kolizi došlo. Náhodná doba čekání je kvůli tomu, aby se výrazně snížila pravděpodobnost toho, že doje k opětovnému přístupu dvou nebo více účastníků ve stejnou dobu. I u této metody **však může dojít ke kolizi**, její příčinou je konečná rychlosť šíření signálu po médiu, čímž vznikne tzv. **kolizní okénko (KO)**. KO je doba od začátku vysílání, během které se signál rozšíří po médiu.



# Princip metody CSMA/CA



Tato metoda se používá zejména u bezdrátových sítí. Je jednodušší než CSMA/CD a její úskalí spočívá v tom, že v případě kolize se někteří účastníci nemusí o jejím vzniku nic dozvědět (důvody budou detailně popsány v [11. přednášce](#)). V případě, že tedy skutečně kolize nastane, musí se o její eliminaci postarat vyšší vrstvy OSI modelu tím, že si znova vyžádají odeslání nedoručných či poškozených dat.



# Logical Link Control (LLC) vrstva

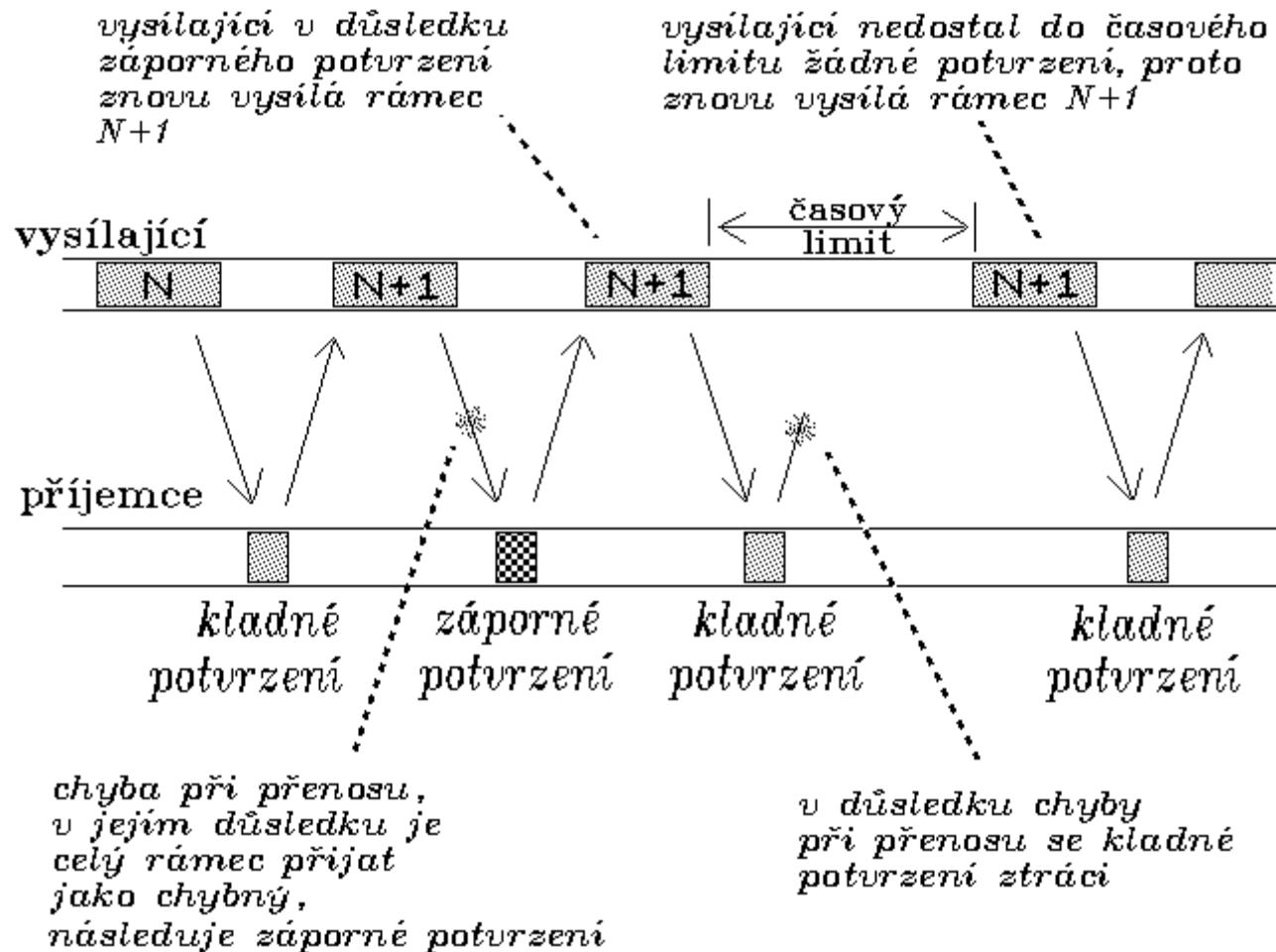
- Je specifikována v normě **IEEE 802.2**.
- Umožňuje existenci různých protokolů (IP, IPX, atd.) nad společnou MAC vrstvou.
- **Má za úkol řízení toku a kontrolu chyb** (k tomuto využívá detekční a samo-opravné kódy).
- Rozdělení toku dat z **vyšších vrstev** do rámců (**framing**)
  - **Dalším úkolem je:**
    - Určení **velikosti** rámce → optimální doba přenosu (u vysílání).
    - Určení **konce** rámce → mezera, doplňkový znak, doplňkový bit (u přijetí).



# Řízení toku a jeho mechanismy

- Obecně se používá pro zajištění doručování dat.
  - Zdroj nesmí odesílat více rámců, než je cíl schopný přijmout.
  - Níže popsané mechanismy jsou použitelné i pro jiné vrstvy.
- 
- **Potvrzovací schémata**
    - Jednotlivé potvrzování (Stop&Wait).
    - Kontinuální potvrzování se selektivním opakováním (Selective Repeat).
    - Kontinuální potvrzování s návratem (Go-Back-N).
    - Klouzavé okénko (Sliding Window).

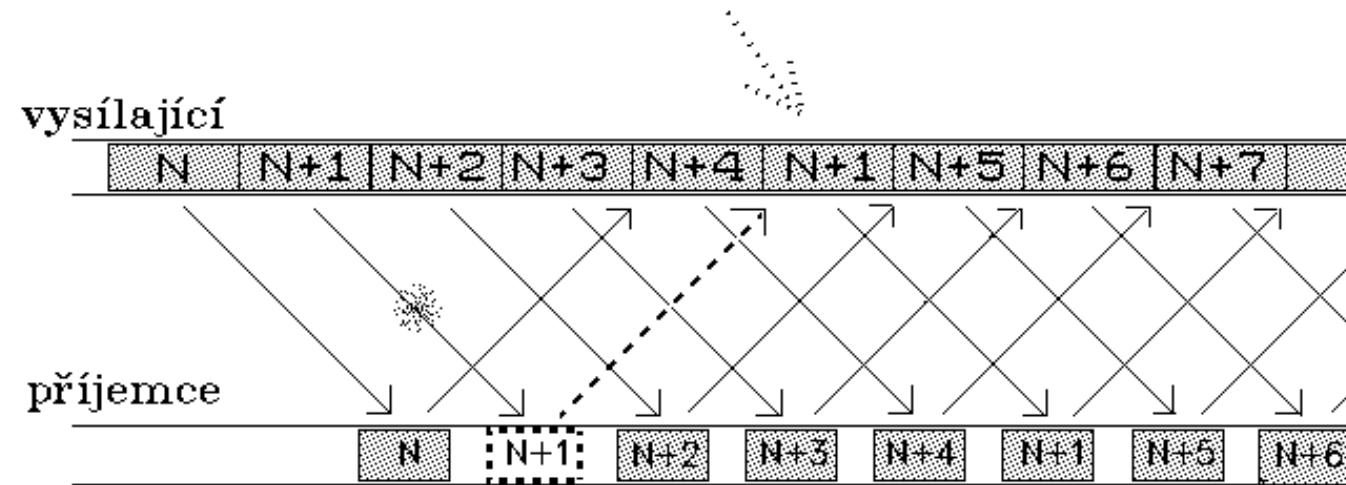
# Jednotlivé potvrzování



# Kontinuální potvrzování se selektivním opakováním



*vysílající v důsledku  
záporného potvrzení opakuje  
přenos rámce  $N+1$*

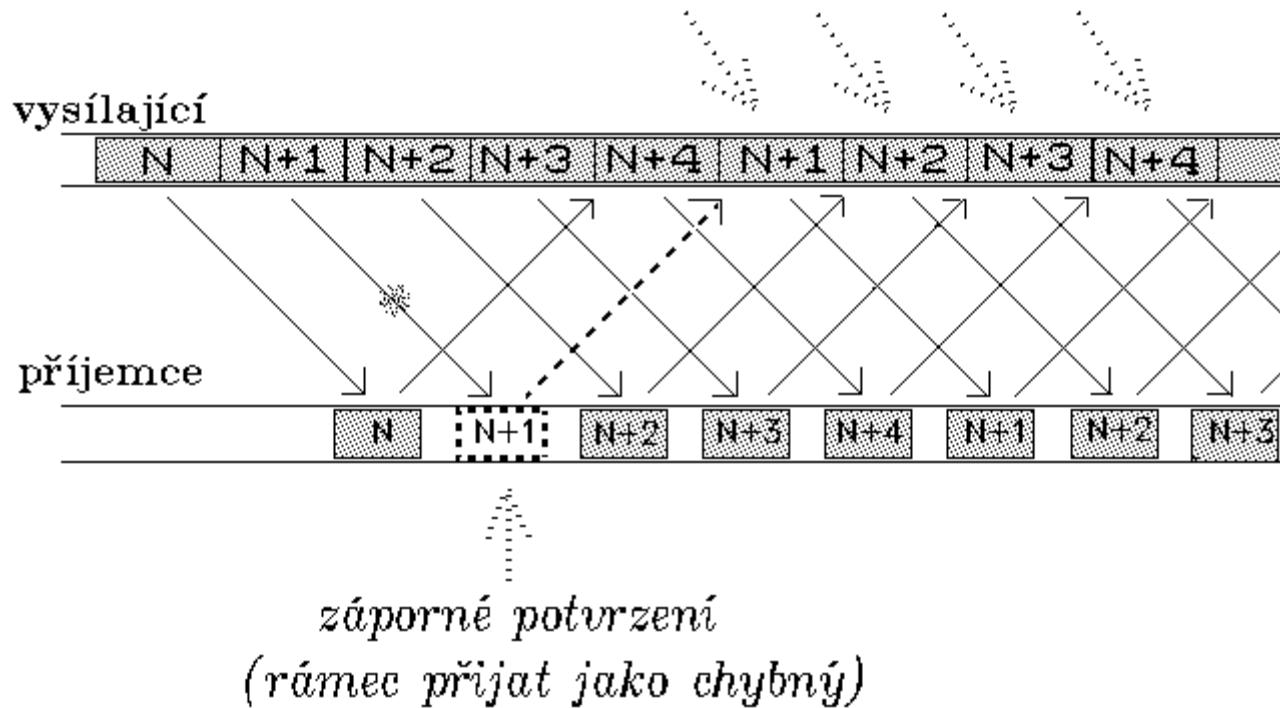


*záporné potvrzení  
(rámec přijat jako chybný)*

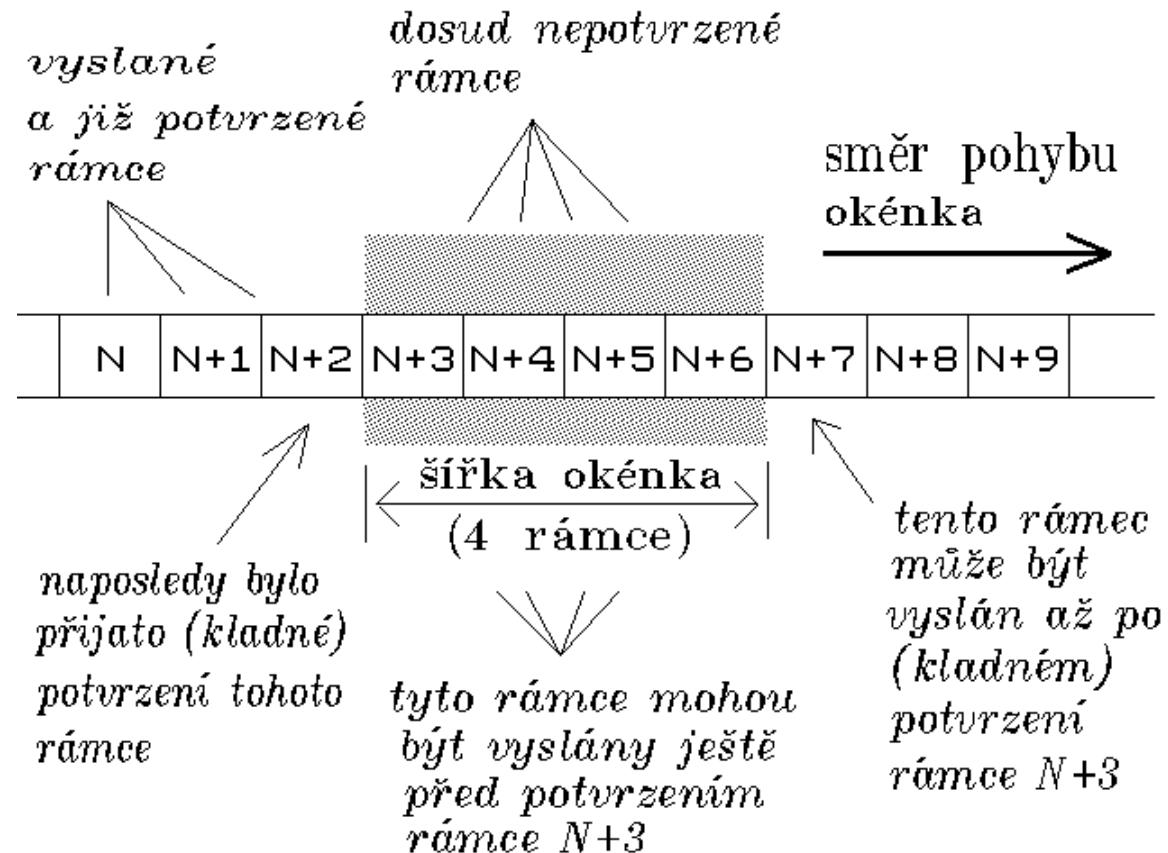


# Kontinuální potvrzování s návratem

*vysílající v důsledku  
záporného potvrzení rámce N+1  
opakuje přenos rámců N+1, N+2, N+3 ...*



# Klouzavé okénko

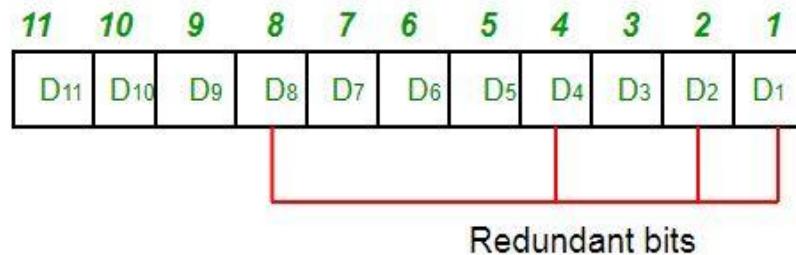


Základní myšlenka spočívá v tom, že přijímač vždy potvrzuje **nejvyšší číslo rámce, které tvoří souvislou ucelenou řadu od začátku**. Vysílač posílá najednou souvislou řadu rámců (počet udává velikost okénka), která začíná číslem o 1 vyšším, než bylo nejvyšší číslo rámce potvrzeného přijímačem.



# Kontrola, detekce a oprava chyb

- Pro tyto účely se používají bezpečnostní kódy (viz BI-SAP, 10. přednáška), běžné jsou např. **Hammingovy kódy**. **Hlavní myšlenka těchto kódů spočívá v tom, že** k datovým bitům nesoucím informaci se přidají redundantní bity, které umožnují chybu přenosu později detektovat resp. opravit.
- Např. **Hammingův kód (7,4)** – 7 bitů datových, 4 bity redundantní, celkem se přenáší **11** bitů místo původních **7** bitů. Kód může fungovat jako samoopravný (za přepodkladu 1 chyby) nebo detekční (předpoklad 2 chyb).



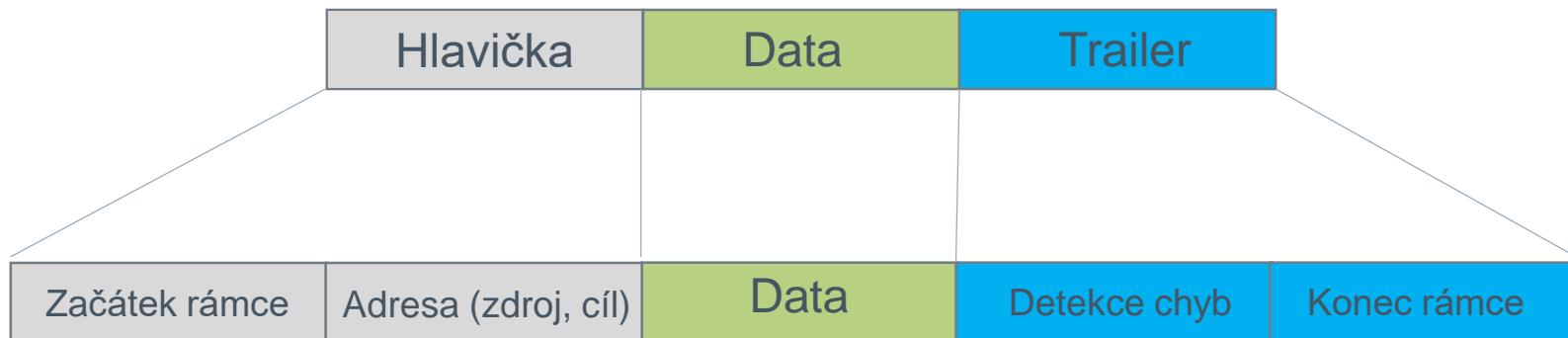
Pro kontrolní součty (Cyclic Redundancy Check) se používají např:

- **cyclic redundancy code (CRC), Parita** (sudá → dle počtu jedniček), **koktavý kód, atd.**

# Zařízení pracující na linkové vrstvě

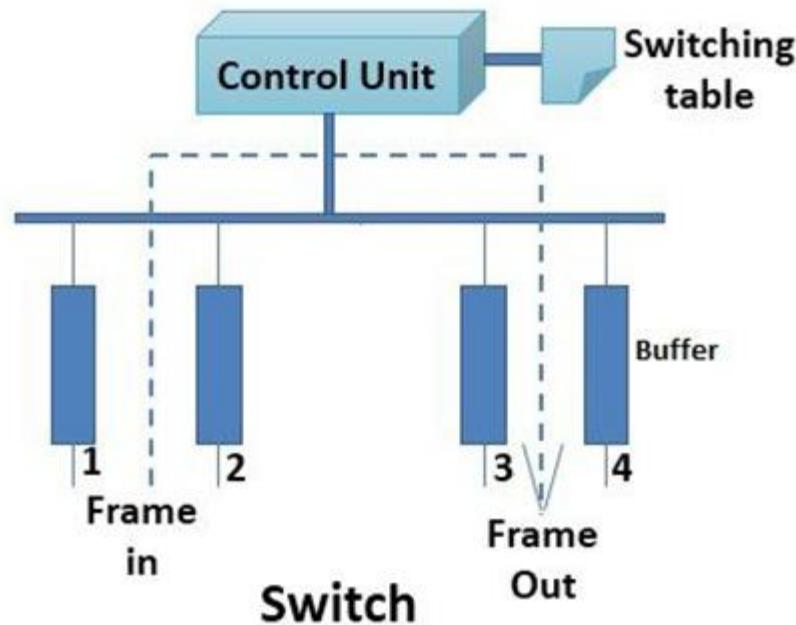


- Zařízení linkové vrstvy pracují s **rámci**.
- Obecný formát rámce obsahuje **hlavičku**, **data** a **konec rámce** (trailer/přívěsek).
- Hlavička má na začátku specifickou sekvenci bitů, podle které se rámec pozná (začátek rámce).
- Pomocí konce rámce (sekvence bitů) se zjistí, že rámec již přišel celý.
- V konci rámce je obsažena i informace pro detekci možných chyb (např. kontrolní součet).



- Zařízení, která pracují na linkové vrstvě se označují jako **přepínače (switches)** popřípadě **mosty (bridges)**. Princip funkčnosti obou typů je podobný (viz. dále).
- Maximální možná velikost přenášených dat se označuje jako **MTU (Maximum Transfer Unit)**. Její velikost je nastavena na konkrétním síťovém rozhraní, není uvedena v rámcích! Optimální velikost MTU záleží na typu a chybovosti použitého media (jinak na optice a bezdrátu).

# Přepínač a princip přepínání



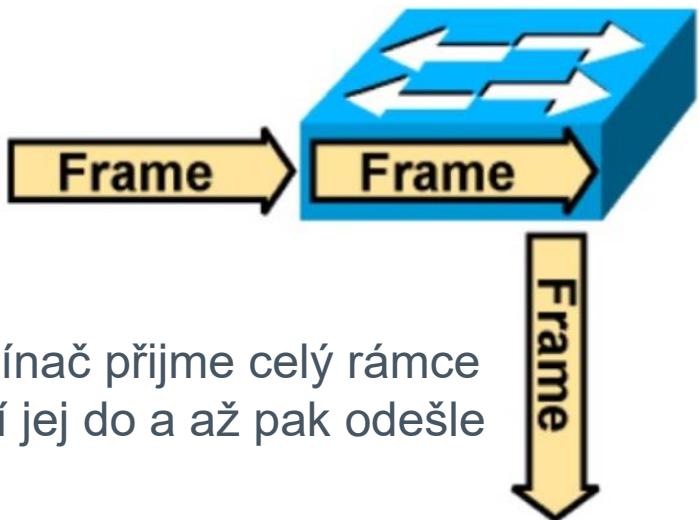
Zdroj: <https://www.computernetworkingnotes.com/ccna-study-guide/switching-methods-and-types-explained-in-computer-networks.html>

- Každý **přepínač (switch)** obsahuje vstupně/výstupní porty, pomocí kterých jsou připojeny komunikační linky.
- Porty obsahují vyrovnávací paměť (buffers).
- Funkcí těchto zařízení je **přepínání (switching)** rámců. Přepínání rámců znamená přijetí rámce (Frame in), zpracování tohoto rámce a odeslání výstupním portem ven směrem k příjemci (Frame out).
- **Přepínací tabulka** (Switching table) slouží k výběru výstupního portu dle MAC adresy příjemce.
- **Záznamy** ve směrovací tabulce **se aktualizují** u přepínačů **učením** (poprvé se rámec s neznámou adresou příjemce pošle na všechny porty s výjimkou odkud přišel a zaznamená se, z kterého portu přišla odpověď).
- Rámce lze zadržovat ve výstupních frontách (bufferech) a tím lze regulovat odchozí tok.

# Režimy práce přepínačů

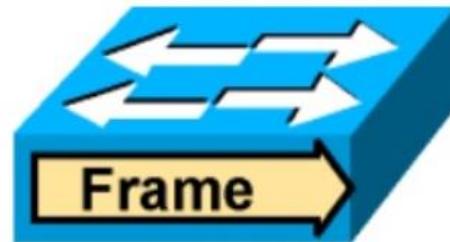


Store-And-Forward (SF)



- Přepínač přijme celý rámce
- Uloží jej do a až pak odešle dále.

Cut-Through (CT)

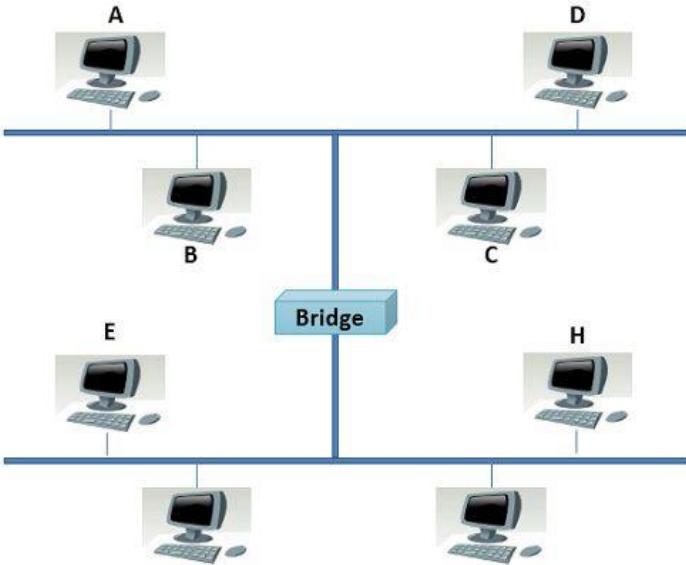


- Přepínač načte jen cílovou adresu (typicky 6 bajtů) z hlavičky rámce.
- Překontroluje ji a odesílá data okamžitě.

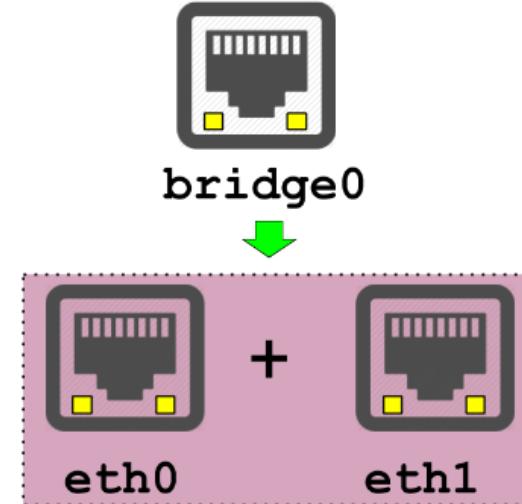
**Fragment-Free (FF):** Podobný princip jako u režimu Cut-Through nicméně s rozdílem, že z rámce je načteno a překontrolováno větší množství dat (celá hlavička, typicky 64 bajtů). Oproti CT dokáže FF odhalit větší množství chybných rámců.

**Poznámka.** V předmětu BI-PSI předpokládat, že přepínače pracují v režimu Store-And-Forward (nebude-li uvedeno jinak).

# Mosty a jejich funkce



**Hardwarový most (bridge)**

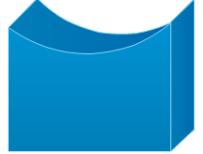


**Softwarový most (bridge)**

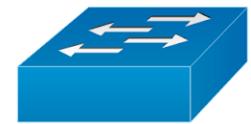
Zdroj: <http://etutorials.org/Networking/Lan+switching+fundamentals/Chapter+1.+LAN+Switching+Foundation+Technologies/Transparent+Bridging/>

Mosty **propojují** resp. **oddělují** provoz více částí sítě (obsahují taktéž přepínací tabulky). Mohou být realizované jako hardwarová či softwarová zařízení. Softwarové mosty jsou implementovány jako součást operačního systému.

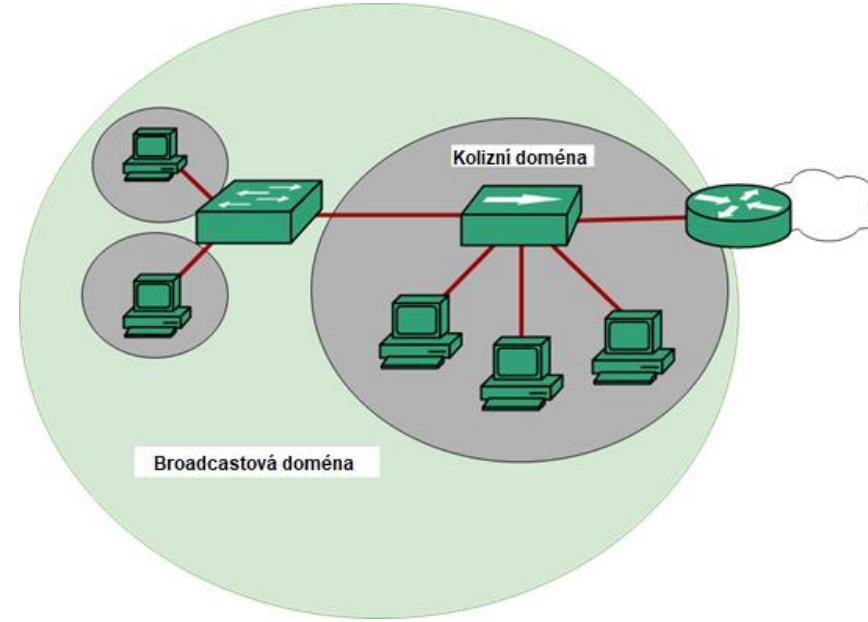
# Obecné srovnání mosty vs. přepínače



- **Mosty (bridges)**
  - jednodušší, málo portů (2 až 4).
  - Propojují **většinou jen dvě sítě**.
  - **Neobsahují** vyrovňávací paměť (**buffery**).
  - Označují se takto i softwarová řešení, která spojují několik sítí do jediné.
- **Přepínače (switches)**
  - Technicky složitější, hodně portů (klidně desítky až stovky).
  - **Hardwarová podpora** zpracování rámců na úrovni chipsetu u dražších zařízení (přepínání nezatěžuje procesor).
  - Obsahují **vyrovnávací paměť**, dovedou regulovat výstupní tok.
  - U dražších modelů je k dispozici např. webové rozhraní, možnost vzdáleného monitorování atd.



# Broadcastová doména



Přepínače a mosty propojují jednotlivé části sítě mezi sebou tak, že přeposílají mezi nimi rámce. Rámce, které obsahují v položce příjemce všesměrovou adresu, jsou doručovány všem stanicím (broadcast).

**Broadcastová doména** je tedy množina všech stanic dané sítě, kterým je doručen rámec s všesměrovou adresou. Jednotlivé broadcastové domény oddělují od sebe až zařízení pracující na síťové (3.) vrstvě OSI modelu (směrovače). Podrobněji bude uvedeno v další přednášce.

# Příklady některých protokolů linkové vrstvy



## Ethernet

- dnes nejběžnější technologie linkové vrstvy → detailněji dále.

## HDLC (High-Level Data-Link Control)

FYI

- Velmi rozsáhlá norma.
- Existují různé implementace s omezenou kompatibilitou.
- Bitově orientovaný protokol.
- Synchronní i asynchronní přenos.
- Využit v sériových linkách a ovlivnil řadu protokolů.

## SLIP (Serial Line Internet Protocol)

FYI

- Definuje pouze zapouzdření paketů na sériové lince.
- Nedefinuje: adresaci, typ paketů, detekci chyb, kompresi.

## PPP (Point-to-Point Protocol)

FYI

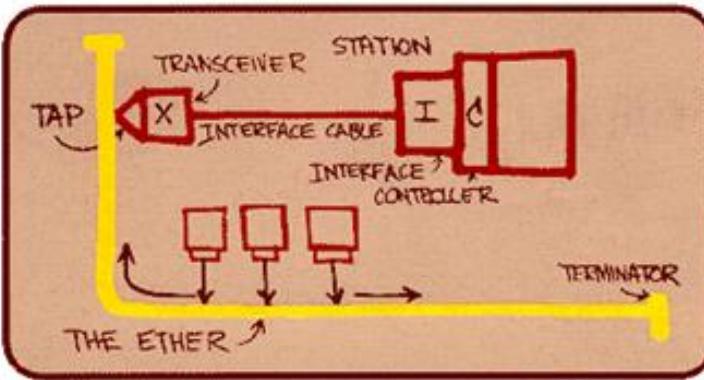
- Podmnožina HDLC.
- Asynchronní.
- Bitově i znakově synchronní.
- Umožňuje souběh více protokolů.

# Počítačové sítě

## 2. Přednáška – technologie Ethernet



# Trocha historie



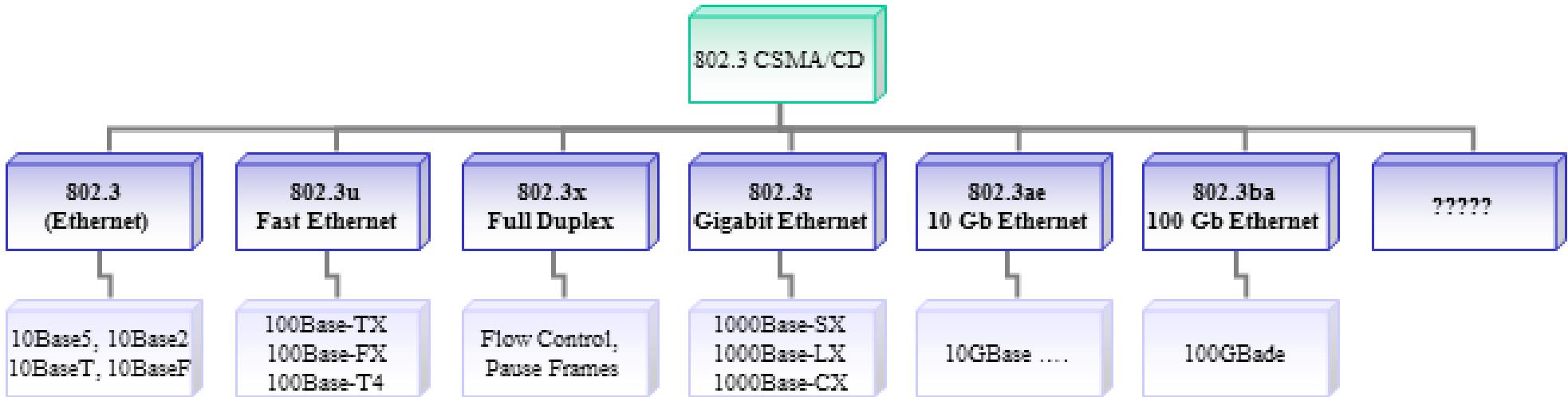
Zdroj: [https://www.ieee802.org/3/ethernet\\_diag.html](https://www.ieee802.org/3/ethernet_diag.html)

- Ethernet vznikl v 80. letech v laboratořích firmy **XEROX**.
- Autorem jsou **Robert Metcalfe** (založil 3COM) a **David Boggs**.
- **Ethernet** – souvisí se slovem éter (všeobjímající, souvislost s Maxwellem a elmag. vlnami).
- **První použitelná verze** – 10 Mbit/s byla vydána v roce 1980 (konsorcium firem DEC, Xerox, Intel, HP).



# Norma Ethernetu a jeho různé verze

FYI



Zdroj. [https://www.researchgate.net/figure/EEE-8023-Ethernet-Standards-overview-2-ETHERNET-COMMUNICATION-This-section-covers-OSI\\_fig5\\_267450031](https://www.researchgate.net/figure/EEE-8023-Ethernet-Standards-overview-2-ETHERNET-COMMUNICATION-This-section-covers-OSI_fig5_267450031)

Rodina protokolů Ethernetu je standardizována normami pod označením IEEE 802.3. Konkrétní verze se označuje písmenem uvedeným za označením 802.3.

# Původní model Ethernetu

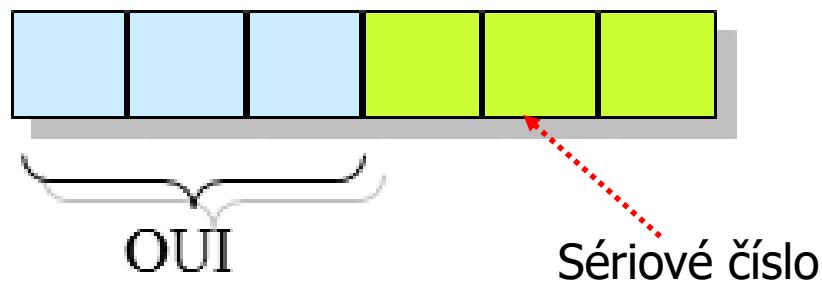


- Označován jako **DIX Ethernet**.
- Návrh byl předložen normalizační komisi IEEE (Institute of Electrical and Electronics Engineers).
- IEEE DIX Ethernet přijala za standard s určitými změnami v detailech jako **IEEE 802.3**.
- Původní DIX Ethernet už se dále nikdy nevyvíjel, pokračovalo se pouze ve vývoji standardu **IEEE 802.3**.
- **Původně byl Ethernet obchodním označením firmy Xerox, ta se jej ale nakonec vzdala.**



# Linkové (MAC) adresy v Ethernetu

- Jsou definovány v rámci MAC vrstvy, běžně se jím říká MAC adresy.
- Standardní velikost adres je **48 bitů** (6 bytů) (MAC 48).
- První tři byty adresy jsou unikátní, přiděleny konkrétnímu výrobci (**Organization Unique Identifier \***) a lze pomocí nich identifikovat konkrétní zařízení.
- Zbylé tři byty označují konkrétní **sériové číslo** daného zařízení.
- Je možné, že časem 48 bitů adresy stačit nebude, nicméně vyčerpání 48 bitových MAC adres se nepředpokládá až kolem roku 2100.

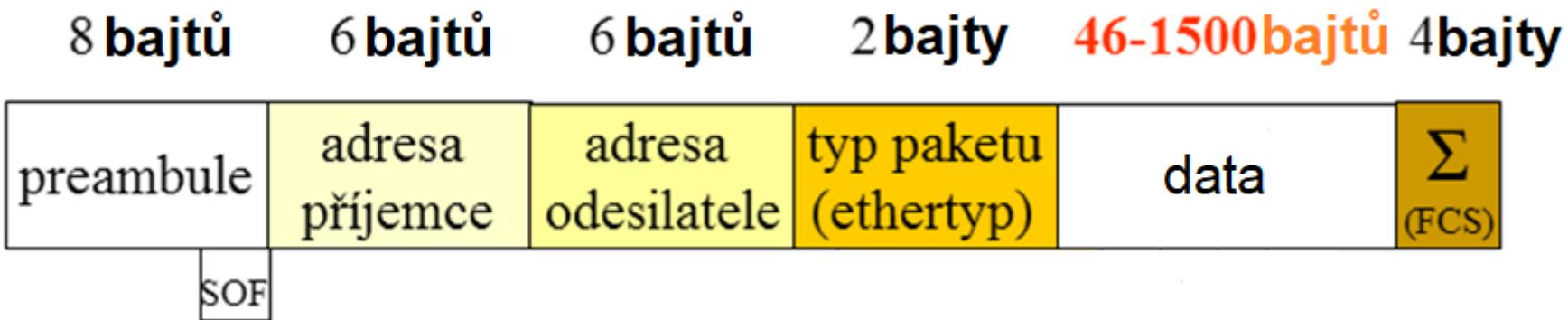


- **Všesměrová (broadcast)** MAC adresa = FFFF FFFF.

\* Na adrese <http://standards.ieee.org/regauth/oui/oui.txt> lze nalézt kompletní seznam výrobců.



# Obecný formát rámce Ethernetu



- preambule = úvodní hlavička pro identifikaci rámce () se liší dle verze použitého Ethernetu.
- SOF = start of frame
- ethertyp = 16 bitů, které specifikují detailněji typ použitého rámce (různé využití).
- FCS (Frame Check Sequence) = sekvence 32 bitů obsahující kontrolní součet rámce umožňující detekci poškozeného rámce.

Běžná velikost přenášených dat v Ethernetu je rovna maximální (1500 bytů). Tato velikost je nastavena v konfiguraci konkrétního síťového rozhraní a **není uvedena v rámci!**

# Označování druhů Ethernetu



Příklad:

**10 Base 5 - FD**

10 Mbit/s

Délka 5 m  
koaxiálního kabelu

Full duplex

Příklady:

**Rychlosť média:** 10,100,1000,10000

**Base** = základní pásmo.

**Specifikace média:** číslo (max. délka vedení [m]), T = kroucená **dvoulinky** (twisted pair), F = optika (fiber).

**Směrovost média:** Full Duplex (FD), Half Duplex (HD).

# Mechanismus řízení toku v Ethernetu



- Ethernet umí ovlivnit příchozí tok a zabránit **zahlcení sítě** (**specifikováno v normě IEEE 802.3x** ).
- Pokud určitý přepínač výkonově “nestíhá”, může požádat své sousedy o zpomalení vysílání prostřednictvím zaslání příkazu **PAUSE**.
- Příkaz PAUSE je možné zaslat konkrétnímu uzlu, všem (broadcastem) či multicastové skupině specifických přepínačů (pro tento účel je vyhrazena MAC adresa **01-80-C2-00-00-01**).
- Ethernet **však neobsahuje žádný mechanismus** (např. klouzavého okénka), který by garantoval spolehlivé doručení dat.



# Autonegociace

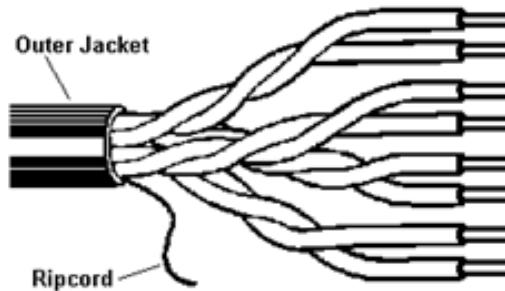
- **Autonegociace** = metoda, kterou si vysílač a přijímač **dohodnou parametry přenosu** (tyto souvisí s max. průtokem, konkrétně se jedná např. o modulaci či použité kódování).
- Pro dojednání parametrů se používají různé způsoby (signalizační pulzy či specifické rámce dle konkrétní verze použitého Ethernetu).
- Konkrétním dojednaným a použitým parametrem přenosu se říká **slinkování** (např. 100 Mbit FD).
- Autogenociace má význam pokud je médium zarušeno, nebo jsou-li fyzicky narušeny vodiče linek.
- Volba nižší propustnosti znamená použití robustnějšího způsobu přenosu dat (např. odolnější modulace).
- Díky autonegociaci je možná zpětná **kompatibilita zařízení** (novější zařízení se přizpůsobí).

# Ethernet 10 BASE T



Zdroj: <https://informatics.buzdo.com/p368-utp-stp-cable.htm>

**UTP Cable (4-pair)**

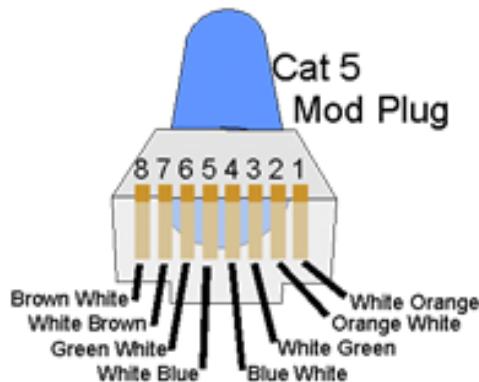


- Důvod, proč se použil tento typ kabelu, bylo to, že řada budov v USA v 90. letech obsahovala redundantní telefonní kabely.
- Použití **nestíněného kabelu Ushielded Twisted Pair (UTP)**.
- Firma, která v té době měla telekomunikační monopol, byla **AT&T**.
- Původní verze byla tzv. 1 BASE 5 (1987), s dosahem pouhých 5 metrů.
- U verze **10 Base T** je max. **dosah 100 metrů (běžné pro všechny novější verze)**.
- Na rozvětvení sítě se používaly **rozbočovače (huby)**.



# Zapojení vodičů v Ethernetu

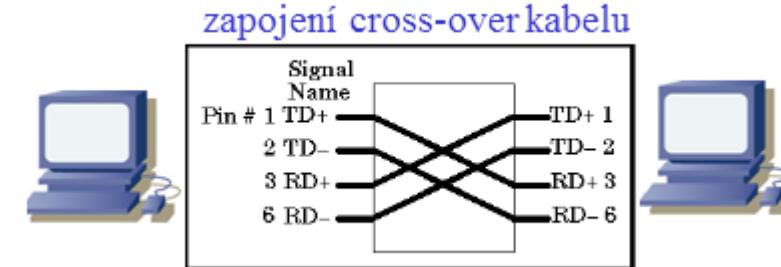
- pin č. 1: TransmitData+
- pin č. 2: TD-
- pin č. 3: Receive Data+
- pin č. 6: RD-
- ostatní: nevyužité



**Jeden pár vodičů se používá pro vysílání, druhý pro příjem.**

Žádný vodič není uzemněn, užitečný signál se bere jako rozdíl potenciálů obou vodičů.

RX = příjem, TX = vysílání.



Zdroj: <https://informatics.buzdo.com/p368-utp-stp-cable.htm>

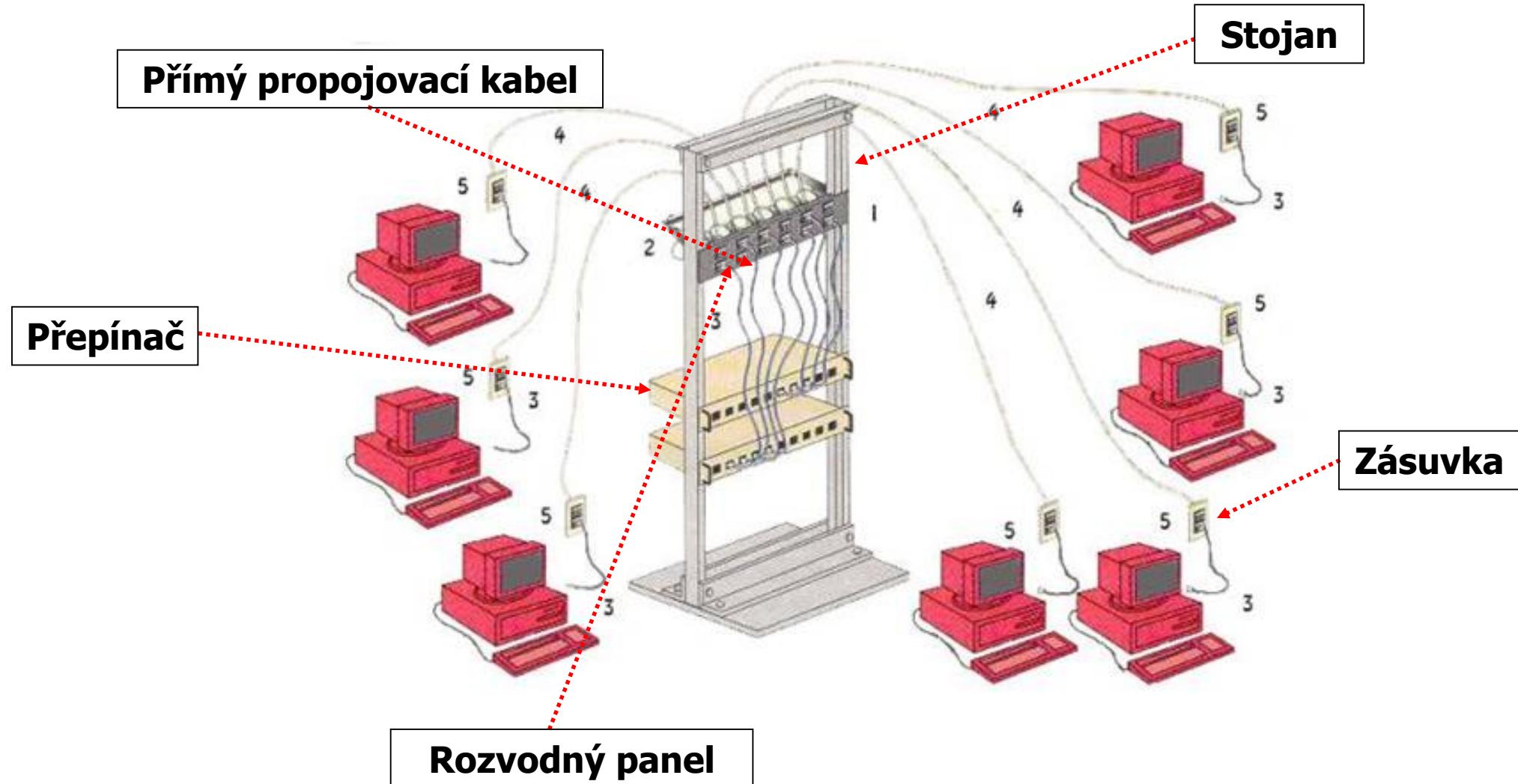
Vzhledem ke koncovým stranám není zapojení symetrické (RX-TX). Příjem jedné strany je pro opačnou stranu vysíláním a naopak.



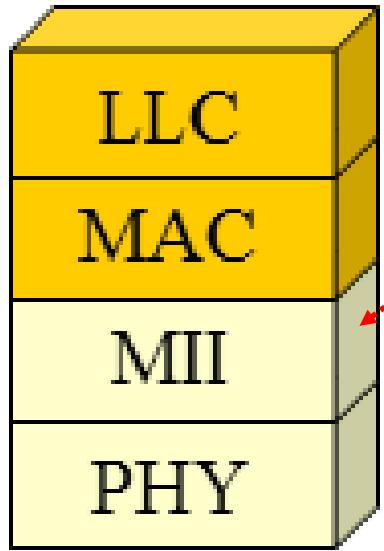
# Propojování zařízení v Ethernetu

- Počítače (stanice) se mezi sebou bez přepínačů propojují **kříženým kabelem** (crossover).
- Počítače se s přepínači propojují běžně **přímým kabelem** (patch cord).
- **Funkce auto-crossover** znamená schopnost zařízení automaticky detekovat použité zapojení kabelu (křížený nebo přímý).
- Strukturovaná kabeláž Ethernetu bývá běžně v budovách vyvedena do příslušné rozvodné skříně (racku) či stojanu.
- Jednotlivá vedení strukturované kabeláže končí na jedné straně v propojovacích zásuvkách umístěných v rozvodných (patch) panelech a na druhé straně v klientských zásuvkách.
- Z klientských zásuvek se připojují koncová zařízení (počítače, síťové tiskárny atd.).

# Ukázka strukturovaná kabeláže v Ethernetu



# Model 100 Mb Ethernetu



**MII = Media Independent Interface**

Zavádí mezi MAC a fyzickou vrstvou novou společnou vrstvu sjednocující různé druhy médií (MII).

Označení **100 Base X**



**100 Base T4 = zapojeny všechny 4 páry média.** **100 Base FX = optika (fiber),**  
**100 Base MX = metaliku (metal).**



# Gigabitový Ethernet



- Návrh 1995, standardizován v roce 1998 pod **IEEE 802.z**.
- Jako médium se používá optické vlákno, popřípadě kroucená dvoulinka.
- Přenosová kapacita tedy odpovídá **1000 Mbitům** (nikoli 1024).
- Verze **1000 Base SX** (multi-módová vlákna, max. 500m), **1000 Base LX** (single-módová vlákna, až 3km), **1000 Base T** (nutné 4 páry, až 100 m), **1000 base CX** (stíněný kabel (Shield Twisted Pair) – max. 25 m, ale jen 2 páry).
- Konkrétní specifikace optických vláken (single/multi módová) budou probírány na přednášce č. 12.



# 1000 Base T Ethernet

- Definován ve standardu **IEEE 802.3ab**.
- Nutnost zapojení všech 4 vodičů, každý pár 250 Mbit/s.
- Všechny vodiče se používají jak na vysílání, tak i k přijímání.
- Autonegociace je řešena **pomocí speciálních rámců**, řeší se takto i řízení toku a další parametry.
- Automatický výběr kříženého či přímého kabelu je definován již ve standardu.



FYI

# 10 Gbit Ethernet

- V roce 2002, standard **IEEE 802.3ae**.
- Provozovatelný jak na metalice tak i optice.
- Dosah na **optice až 40km**.
- Obvyklé je nasazení **WDM multiplexu**.
- **Možnost použití velkokapacitní rámců (říká se jim JUMBO, MTU je až 9000 bajtů)**.



# Různé verze 10G Ethernetu



Interconnect	AKA	Defined	Connector <sup>[7]</sup>	Medium	Media Type	Wavelength	Max range	Notes
10GBASE-SR	short reach	2003	XENPAK, X2, SFP+	fiber	serial multi-mode	850 nm	400 m	
10GBASE-LR	long reach	2003	XENPAK, X2, XFP, SFP+	fiber	serial single-mode	1310 nm	10 km	
10GBASE-ER	extended reach	2003	XENPAK, X2, XFP, SFP+	fiber	serial single-mode	1550 nm	40 km	
10GBASE-ZR		-	XENPAK, X2, XFP, SFP+	fiber	serial single-mode	1550 nm	80 km	Not covered by IEEE 802.3ae
10GBASE-LX4		2003	XENPAK, X2, SFP+	fiber	WDM multi-mode or single-mode	1310 nm	300 m (multi-mode), 10 km (single-mode)	Costly and complex, replaced by 10GBASE-LRM
10GBASE-LRM	long reach multi-mode	2006	XENPAK, X2, SFP+	fiber	serial multi-mode	1310 nm	220 m	
10GBASE-CX4		2004	XENPAK, X2	copper	InfiniBand 4X twinaxial 8-pair <sup>[8]</sup>	-	15 m	Four lanes, each at 2.5 Gbit/s. Larger form factor, bulkier cables and more expensive than SFP+ Direct Attach
SFP+ Direct Attach	DA	2006	SFP+	copper	twinaxial 2-pair	-	10 m	Cheap, low latency, low power
10GBASE-T		2006	RJ45	copper	category 5e, 6, 6a or 7 twisted pair	-	55 m (cat 5e or 6), 100 m (cat 6a or 7)	Can reuse existing cables, high port density, relatively high power
10GBASE-KX4	802.3ap	2007		copper	PCB backplane	-	1 m	
10GBASE-KR	802.3ap	2007		copper	PCB backplane	-	1 m	



FYI

# 40 a 100 Gbit Ethernet

- Nejrychlejší aktuálně **běžně dostupná komerční verze Ethernetu.**
- První studie rok 2007, 2010 oficiální standard **IEEE P802.3ba.**
- Specifikace existuje pro **optická vlákna** (40 km / single vs 125 / multi) i pro metaliku (neprodává se) a **backplane** (propojení modulů v rámci skříně).
- Pro tuto optiku se používá **WDM multiplex.**



# Druhy 40 Gb a 100 Gb Ethernetu

Physical layer	40 Gigabit Ethernet	100 Gigabit Ethernet
Backplane	40GBASE-KR4	
Copper cable	40GBASE-CR4	100GBASE-CR10
100 m over OM3 MMF		
125 m over OM4 MMF <sup>[11]</sup>	40GBASE-SR4	100GBASE-SR10
10 km over SMF	40GBASE-LR4	100GBASE-LR4
40 km over SMF		100GBASE-ER4
Serial SMF over 2 km	40GBASE-FR	



# Kam ještě dále?

FYI

- **IEEE 802.3s, 400Gbit/s** – základy 2013, nicméně technologicky je to ještě nedokončené (100,500, 2000 a 10000 m - verze).
- V roce 2010 byl již plánován **1Tb/s Ethernet**, nicméně v roce 2012 byl IEEE zastaven a pokračovalo se 400Gb Ethernetem.
- Oficiální zdůvodnění je to, že jeho další vývoj je prozatím **příliš nákladný** a současné technologie **jej neumožňují**.
- Detailní popis návrhu systému lze nalézt v knize:

**Theory and Design of Terabit Optical Fiber Transmission Systems od Stefana Bottacchiho.**

# Počítačové sítě

## 3. přednáška – Virtuální sítě (VLAN)



# Virtuální sítě (VLAN) a důvod použití



Vrstvy hostitelů	Data	Vrstva
Vrstvy média	Data	Applikační Síťový proces aplikací
	Data	Prezentační Prezentace dat a šifrování
	Data	Relační Komunikace mezi hostity
	Segmenty	Transportní End-to-End spojení a spolehlivost
	Pakety	Síťová Určování cesty a IP (logické adresování)
	Rámce	Spojová MAC a LLC (Fyzické adresování)
	Bity	Fyzická Médium, signál, binární přenos

**VLAN = Virtual Local Area Network**

Jsou určené k tomu, aby uměly oddělovat toky v sítích na linkové vrstvě.

**Důvody pro oddělování provozu:**

- Přehlednost (spojí se jen to, co patří k sobě)
- Bezpečnost (provoz jde jen tam, kam má)
- Efektivita (provoz sítí se nemíchá)
- Jednoduchost (snazší správa)

VLAN je více druhů.

**Ethernet** implementuje VLAN pomocí speciálních rámci, které se nazývají **značené** (tagované).

# Základní myšlenka VLAN

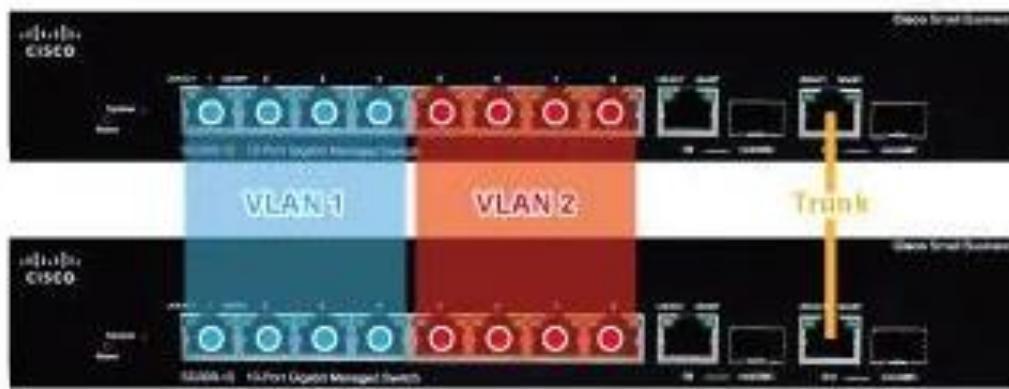


Určitá **organizace** (FIT ČVUT), která má **několik různých pracovišť** (např. **KPS** a **KIB**) a každé pracoviště pro svoji práci potřebuje vlastní přepínač. Z hlediska správy a poruchovosti je ale snazší nastavovat pouze jeden větší místo několika. Díky VLAN lze použít **pouze jeden přepínač**, který se nastaví tak, aby pracoval jako **dva nezávisle oddělené přepínače**.



# Nastavení portů u VLAN, trunk port

V případě, že existují v síti dva či více fyzických přepínačů, je možné virtuálně vytvářet oddělené skupiny i mezi nimi. Porty na každém přepínači se přiřadí do specifické skupiny (možností přiřazení je více).



Zdroj: <https://softagblog.wordpress.com/2016/08/02/vlan-configuration-on-hp-procurve-2810-switch/>

Část portů na každém přepínači je vyhrazena **VLAN1 (KPS)** i **VLAN2 (KIB)**. Aby mohly být stanice v rámci stejné skupiny propojeny napříč oběma přepínači, je nutné i tyto přepínače propojit. Provoz, který bude proudit mezi přepínači, bude však společný (VLAN1 i VLAN2 dohromady).

**Trunk port** = port, který odbavuje provoz více různých VLAN.



# Dělení VLAN sítí

**Dle portů** - konkrétní VLAN je vybrána dle toho, které konkrétní porty na přepínači jsou do ní přiřazeny (lze použít pro menší sítě).

**Dle adres** – konkrétní rámec je zařazen do konkrétní **VLAN** skupiny **dle linkové (MAC) adresy**.

**Dle protokolu** – rámec je zařazen do konkrétní **VLAN dle protokolu** vyšší vrstvy, který přenáší (např. přenos hlasu, videa, atd.).

**Dle značky** – připojením značky (tagu) do rámce (dnes nejběžnější), běžně se označuje jako **tagované vlan**.

**Vícenásobné** (VLAN uvnitř VLAN, Q-in-Q).

# Ethernet a VLAN



## Běžný rámec Ethernetu

Preambule	SFD	MAC cíle	MAC zdroje	Typ/délka	Data a výplň	CRC32	Mezera mezi rámci
7 bajtů 10101010	1 bajt 10101011	6 bajtů	6 bajtů	2 bajty	46-1500 bajtů	4 bajty	12 bajtů

## Ethernet obsahující značené (tagované) rámce (IEEE 802.1Q)

Preambule	SFD	MAC cíle	MAC zdroje	802.1Q Hlavička		Typ/délka	Data a výplň	CRC32	Mezera mezi rámci	
7 bajtů 10101010	1 bajt 10101011	6 bajtů	6 bajtů	2 bajty	VLAN protokol ID	2 bajty	PCP/CFI/VID	2 bajty	42-1500 bajtů 4 bajty	12 bajtů

Do typu rámce se vkládá za MAC adresy identifikátor VLAN. Vložení hlavičky (802.1Q) do rámce znamená, že běžná verze Ethernetu není se značenou kompatibilní.

Přepnutí rámce probíhá tak, že se nejprve dle značky (VID) vybere příslušná VLAN síť a pak se dle přepínací tabulky volí výstupní port.

**SFD** = Start Frame Delimiter (identifikace začátku rámce)

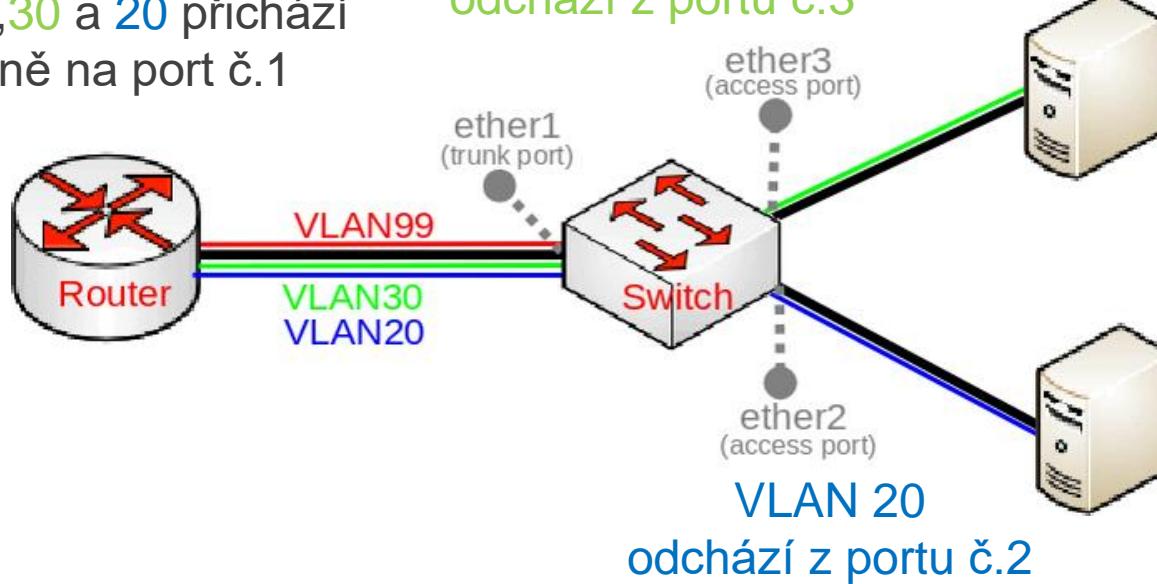
**VID** = VLAN Identifier (značka, tag), 12 bitů →  $2^{12} = 4096$  různých VLAN.

**PCP** = Priority Code Point (802.1p), 3 bity, lze využít k prioritaci provozu.

**CFI** = Canonical Format Indicator určuje v jakém tvaru je rámec přenášen (little vs. big endian), 1 bit.

# Ukázka hybridní VLAN sítě

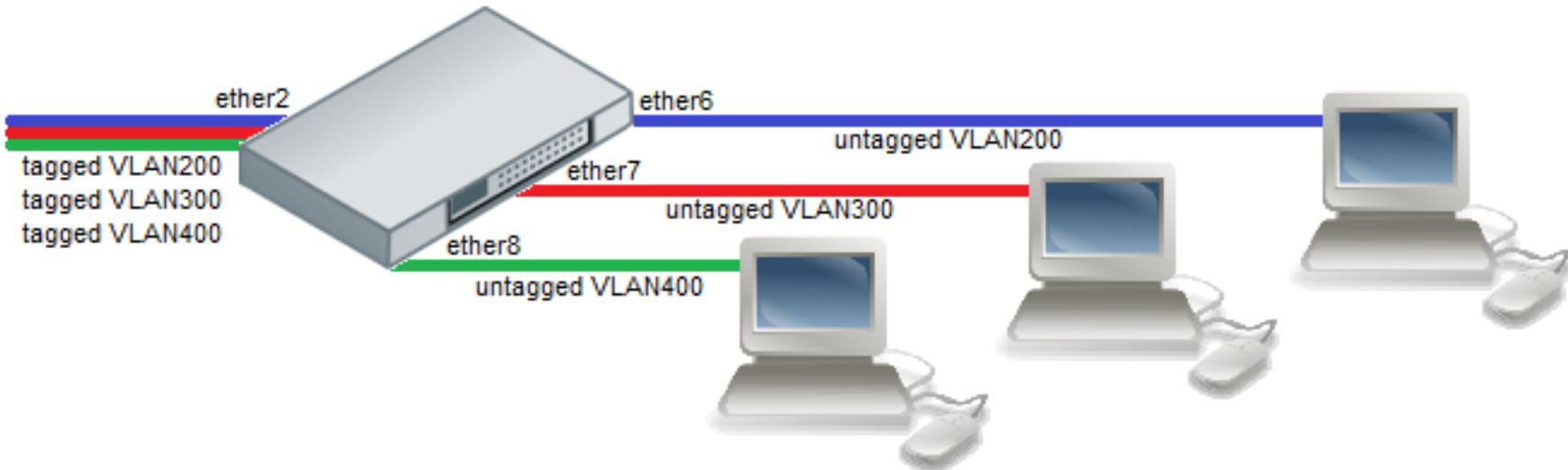
VLAN 99,30 a 20 přichází společně na port č.1



Zdroj: [https://wiki.mikrotik.com/wiki/Manual:Basic\\_VLAN\\_switching](https://wiki.mikrotik.com/wiki/Manual:Basic_VLAN_switching)

Ze směrovače (router) vychází VLAN 99, 30, 20 a obyčejná LAN (označená černě, používá neoznačené rámce). Na přepínači (switch) dojde k rozdělení provozu dle příslušné VLAN a portu. Porty 2 a 3 na přepínači jsou nastaveny tak, že patří jak do konkrétních VLAN tak i do sítě LAN. VLAN 99 končí v trunk portu a dále nepokračuje. Síť, která obsahuje značený i neznačený provoz (na trunk portu i na výstupních portech), se pak označuje jako **hybridní**.

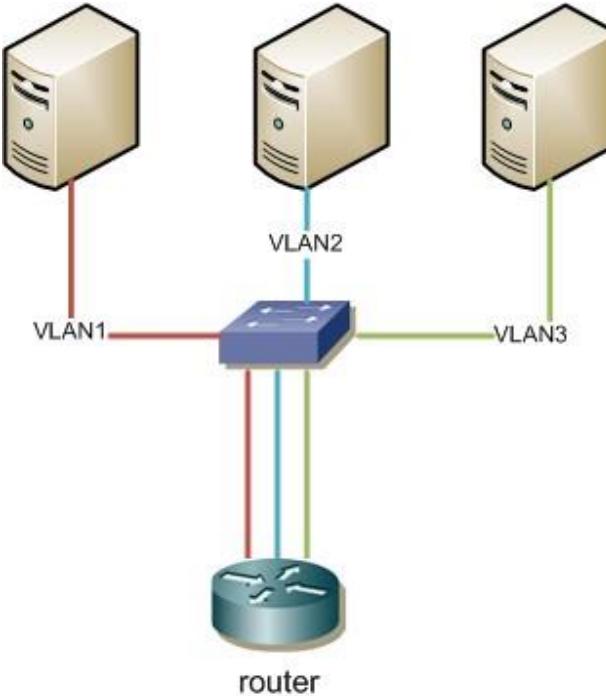
# Použití neznačených rámci v rámci VLAN



Zdroj: [https://wiki.mikrotik.com/wiki/Manual:Basic\\_VLAN\\_switching](https://wiki.mikrotik.com/wiki/Manual:Basic_VLAN_switching)

Přepínač používá jako **trunk port** rozhraní **ether2**, všechny rámce VLAN sítí směřující na toto rozhraní používají značení (tagged). Jakmile přepínač přijme rámec některé VLAN sítě, přepne jej dle přepínací tabulky na příslušný port. V tomto případě **jsou ale značky** (tags) při odchodu rámce z přepínače **odebrány** a výstupní rámec má běžný tvar (untagged). Při **příchodu** rámce na porty (ether 6-8) směrem od stanic, je však nezbytné příslušnou **značku** opět do rámce **navrátit** (dle toho, do které VLAN skupiny port patří), aby tento rámec mohl být vyslán zpátky přes port trunk port. Toto řešení se používá pro oddělení provozu např. u stanic, které značené VLAN nepodporují.

# Přeposílání provozu mezi VLAN sítěmi



Zdroj: [https://wiki.mikrotik.com/wiki/Manual:CRS1xx/2xx\\_series\\_switches\\_examples](https://wiki.mikrotik.com/wiki/Manual:CRS1xx/2xx_series_switches_examples)

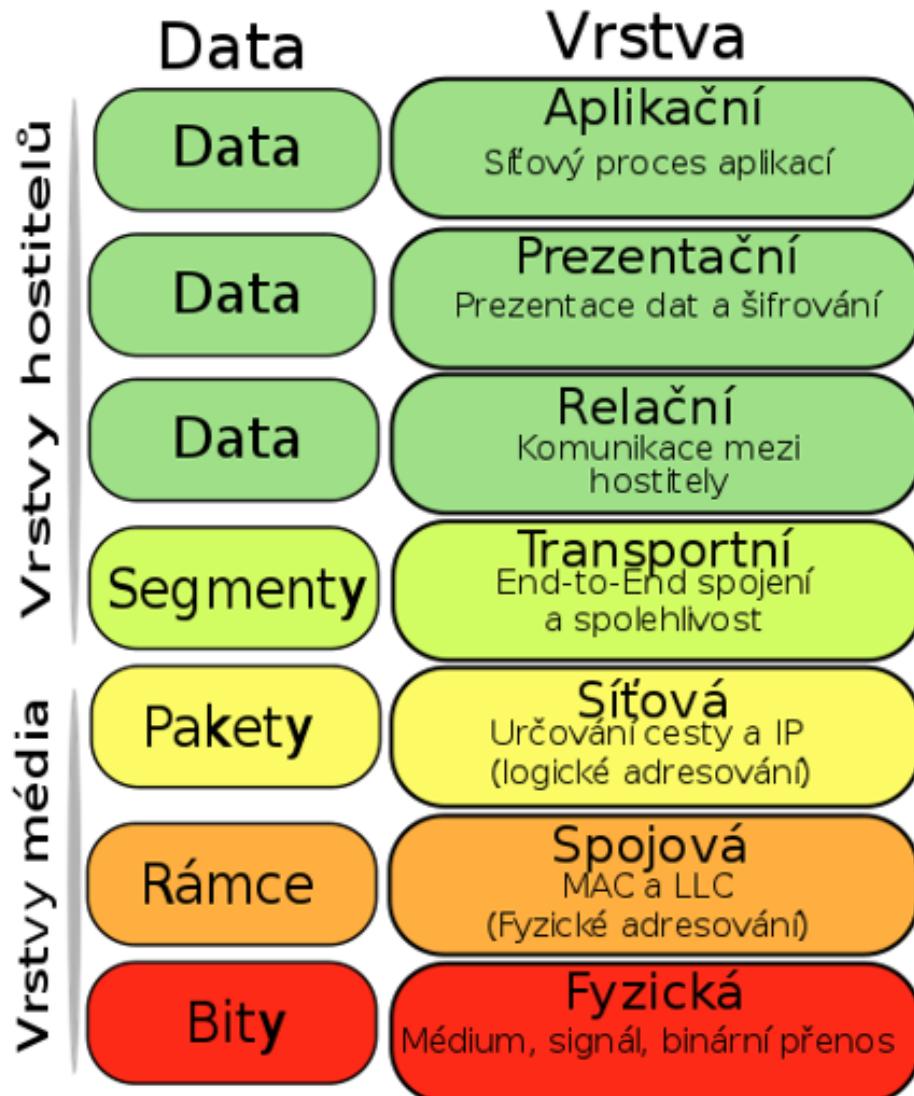
Různé VLAN sítě jsou přes **přepínač připojeny do směrovače** a jejich provoz je navzájem **striktně** oddělen. Přepínač nemůže provoz jedné VLAN sítě propouštět do druhé, jelikož jeho úkolem je provoz oddělovat. Pokud se mají přeposílat **data např. z VLAN1 do VLAN3**, musí tato data nejprve projít skrz přepínač ke směrovači. **Směrovač** již ale nepracuje na linkové vrstvě, ale **na síťové**. Provoz mezi VLAN není tedy přepínán ale směrován (viz dále).

# Počítačové sítě

3. přednáška - Sítová vrstva a IP protokol verze 4



# Začlenění síťové vrstvy a její úloha



Primárním úkolem je **doručování dat mezi stanicemi**, které se nachází v různých sítích.

**Definuje síťové (IP) adresy** (viz 1. přednáška).

Data jsou doručována po **paketech (až 64KB)**.

**IP = Internet Protocol, ICMP = Internet Control Message Protocol** (důležité protokoly síť. vrstvy).

Přeposílání paketů mezi různými sítěmi se dělá prostřednictvím **směrovačů (routerů)**.

Mechanismus hledání cesty mezi zdrojovou a cílovou stanicí se nazývá **směrování (routing)**.

**Adresace síťové vrstvy je hierarchická**, je vhodná pro směrování (**zásadní rozdíl oproti linkové**).



# Síťová adresace, IPv4

- IP = Internet Protocol. Adresa je dlouhá **32** bitů.
- Její **význam je stejný jako** v případě **linkové adresy** (rozlišení stanic - zdroj, cíl).
- IP adresy je možné uspořádávat hierarchicky, shlukovat do skupin (segmentů) → toto u **linkových adres možné není**.
- **Linkové adresy** jsou použitelné pro doručování dat jen v **lokální síti**. **Síťové adresy** jsou použitelné jak **v lokální síti tak i mezi různými sítěmi**.
- V adresaci jsou zohledněny druhy provozu
  - **Unicast**
  - **Broadcast**
  - **Multicast**
- Několik schémat **IP adresace**
  - **CIDR** (classless inter domain routing) – prefix. Notace = **IP/PREFIX** (viz. 1. přednáška, slajdy 20-27).
  - **Dle třídy** (classfull design) – již zastaralé, ale je dobré znát.



# Speciální adresní rozsahy u IPv4

- **Privátní adresy** (pouze v místních sítích)
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- **Link-local** (lokální, mohou se přiřadit automaticky)
  - 169.254.0.0/16
- **Loop-Back** (provoz sám sobě, testování síť. vrstvy)
  - 127.0.0.0/24
- **Multicast** (doručování dat skupině)
  - 224.0.0.0/4

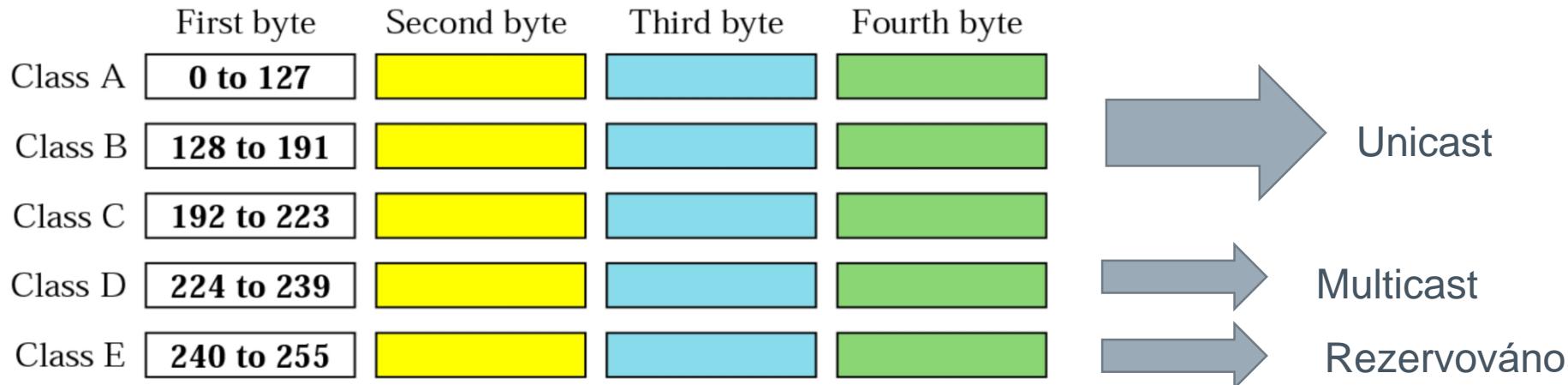
Zmíněné 4 adr. rozsahy lze používat **pouze v lokálních sítích**. Ostatní adr. rozsahy IPv4 se označují jako **veřejné** a je možné je **používat v Internetu**. Každý poskytovatel Internetu má přidělenou (organizaci spravující rozsahy dle příslušného kontinentu) určitou část těchto adr. rozsahů.

# Adresace podle třídy



Zdroj: <https://networkustad.com/tag/classful-subnetting/>

Class	HOB	NET ID Bits	Host ID Bits	No of Networks	Host Per Network	Start Address	End Address	Mask
Class A	0	8	24	$2^7=128$	$2^{24}=16,777,216$	0.0.0.0	127.255.255.255	255.0.0.0
Class B	10	16	16	$2^{14}=16,384$	$2^{16}=65,536$	128.0.0.0	191.255.255.255	255.255.0.0
Class C	110	24	8	$2^{21}=2,097,152$	$2^8=256$	192.0.0.0	223.255.255.255	255.255.255.0
Class D	1110	-	-	-	-	224.0.0.0	239.255.255.255	
Class E	1111	-	-	-	-	240.0.0.0	255.255.255.255	



Znalost adresace podle třídy se může hodit v okamžiku, kdy nezadáte např. v Linuxu **masku** či **prefix**. Systém příslušnou masku zvolí podle třídy, do které adresa patří.



# IPv4 paket a jeho popis

Bajty	0	1	2	3
Bajt 0 až 3	verze	IHL	typ služby	celková délka
Bajt 4 až 7		identifikace	příznaky (3 bity)	offset fragmentu (13 bitů)
Bajt 8 až 11	TTL	číslo protokolu		kontrolní součet hlavičky
Bajt 12 až 15			zdrojová adresa	
Bajt 16 až 19			cílová adresa	
Bajt 20 až ((IHL × 4) - 1)			rozšířená nepovinná nastavení	
...			data	

Paket je základní jednotkou přenosu dat přenášených na síťové vrstvě. Hlavička IPv4 paketu nemá na rozdíl od hlavičky ether. rámce fixní velikost. Pakety se taktéž označují jako **datagramy**. Pakety se mapují do ethernet. rámce jako data. (viz 2. přednáška, slajd 31). Nutností pro doručení eth. rámce je znalost **cílové MAC adresy** (její zjištění na základě znalosti **cílové IP adresy** umožňuje **Address Resolution Protokol** – viz dále).

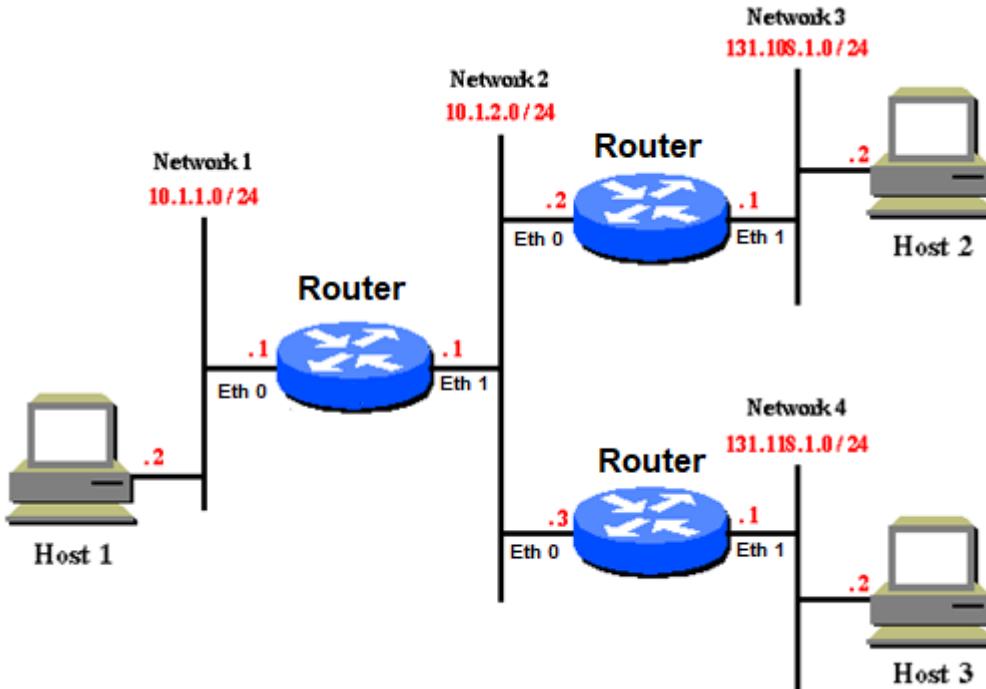
**Poznámka.** Detailní popis položek paketu je uvedený na další stránce.



# Význam položek IP paketu

- › **Verze:** verze protokolu (0x4).
- **IHL:** délka hlavičky jako počet 32bitových slov.
- **Typ služby (TOS, Type of Service):** nese značku pro mechanismy zajišťující služby s definovanou kvalitou ([QoS](#)).
- **Celková délka:** délka paketu v bajtech.
- **Identifikace:** Pokud byl paket při přepravě fragmentován, pozná se podle této položky, které fragmenty patří k sobě (mají stejný identifikátor).
- **Příznaky:** slouží pro řízení fragmentace.
- **Offset fragmentu:** udává, na jaké pozici v původním paketu začíná tento fragment.
- **TTL (Time To Live):** představuje ochranu proti zacyklení. Každý směrovač sníží tuto hodnotu o jedničku. 0 = zahození.
- **Protokol:** určuje, kterému protokolu vyšší vrstvy se mají data předat při doručení.
- **Kontrolní součet hlavičky:** slouží k ověření, zda nedošlo k poškození. **Počítá se pouze z hlavičky** a pokud nesouhlasí, paket bude zahozen.
- **Adresa odesílatele:** IP adresa síťového rozhraní, které paket vyslalo.
- **Adresa cíle:** IP adresa síťového rozhraní, kterému je paket určen.
- **Volby:** různé rozšiřující informace či požadavky.
- **Data:** obsahuje další zapouzdřené protokoly.

# Směrovače v síti a jejich význam



Zdroj: <https://www.orbit-computer-solutions.com/home/routing-protocols/>

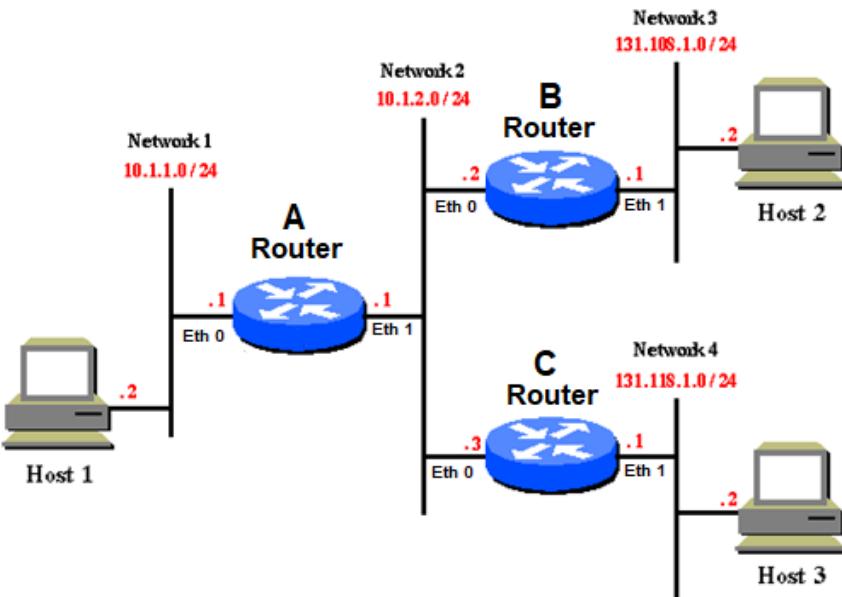
Na obrázku je vidět několik různých sítí, které mají přidělené navzájem disjunktní adresní rozsahy uvedné v prefixové notaci. O přeposílání provozu mezi sítěmi se starají **směrovače** (routery). Směrovače patří adresně vždy do všech sítí, které propojují. Doručování dat přímo mezi směrovači provádí linková vrstva. Při průchodu paketu směrovačem (**hop**) se snižuje hodnota TTL paketu.



# Princip směrování (routingu)

- Směrování = odeslání konkrétního paketu (P) prostřednictvím zvoleného odchozího síťového rozhraní na základě cílové adresy uvedeného v hlavičce paketu.
- **Princip** směrování paketů je velice **podobný** principu **přepínání** na linkové vrstvě.
- Směrování provádí **směrovač**, který paket zpracovává.
- Směrovač zvolí při **odchodu** paketu **odchozí síťové rozhraní**. Pomocí linkové vrstvy se paket P doručí sousednímu směrovači popřípadě cílové stanici.
- Zda se bude používat pro doručení další směrovač, či se paket doručí přímo cílové stanici, se pozná podle toho, zda cílová adresa P patří do některé ze sítí naležící směrovači, který o doručení rozhoduje.
- **Pro určení odchozího rozhraní resp. dalšího směrovače slouží směrovací tabulka (routing table), která obsahuje směrovací záznamy.**

# Směrovací tabulky a směrování



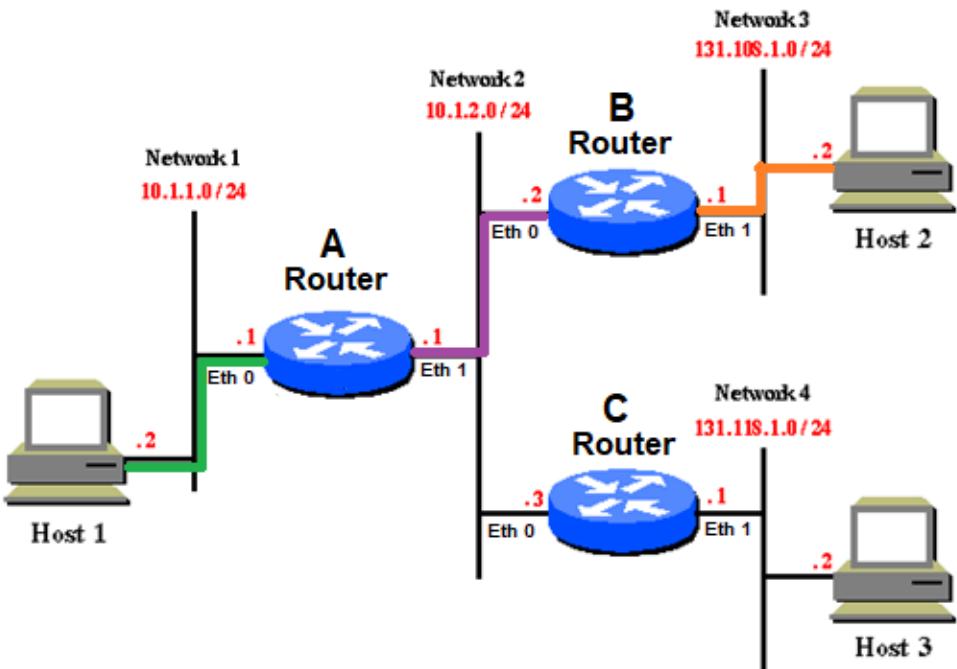
Směrovací tabulka A

Cíl	Směrovač	Metrika	Rozhraní
10.1.1.0/24	-	0	Eth 0
10.1.2.0/24	-	0	Eth 1
131.108.1.0/24	10.1.2.2	1	Eth 1
131.118.1.0/24	10.1.2.3	1	Eth 1

Zdroj: <https://www.orbit-computer-solutions.com/home/routing-protocols/>

**Destination** = cíl, **Nexthop** = směrovač k cíli, **Metrika** = vzdálenost ( cena ), **Interface** = rozhraní.  
Směrovací tabulku **obsahuje každý směrovač (resp. každý počítač s připojením k síti)**. Tabulka se může konfigurovat **manuálně** anebo **automaticky** prostřednictvím směrovacího protokolu (v přednáška 5).

# Proces směrování



Zdroj: <https://www.orbit-computer-solutions.com/home/routing-protocols/>

Host 1 (10.1.1.2/24) **chce komunikovat** s Hostem 2 (131.108.1.2/24).

Host 1 **nejprve zjišťuje, zda neleží ve stejné síti**, k tomuto využívá svoji IP, svoji masku (255.255.255.0) a IP Hosta 2.  
10.1.1.2 & 255.255.255.0 = 10.1.1.0

131.108.1.2 & 255.255.255.0 = 131.108.1.0

Jelikož **10.1.1.0 != 131.108.1.0** → nelze poslat lokální síť, musí se přes bránu (Router A).

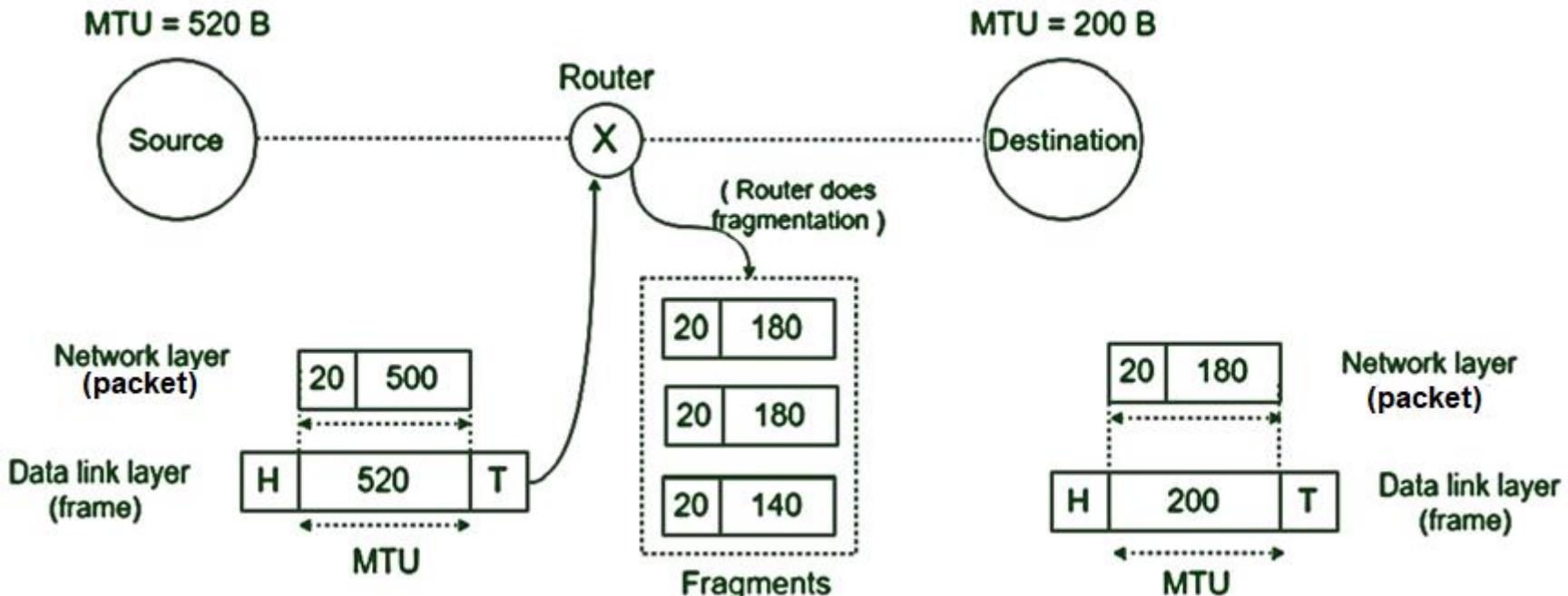
1. Host 1 pošle prostřednictvím link. vrstvy IP paket Routeru A, ten jej přijme (přes Eth 0).
2. A pomocí své směrovací tabulky a cílové IP zjistí, že pro doručení paketu k Hostu 2 jej musí doručit přes B (Eth 1). A při odchodu paketu sníží TTL o 1.
3. A doručí paket pomocí linkové vrstvy B pomocí linkové vrstvy.
4. B po přijetí paketu zjistí na základě cílové IP a své směrovací tabulky, že cílová adresa leží v jeho lokální síti. B sníží opět TTL o 1.
5. B doručí paket Hostu 2 prostřednictvím linkové vrstvy (přes Eth 1).

# Fragmentace IP paketů



- Pokud je velikost IP paketu větší, než maximální velikost části rámce linkové vrstvy určené pro data, musí se IP přenášet po částech (fragmentech).
- Fragmentaci **indikují příznaky v hlavičce IP paketu**, **offset fragmentu** udává jeho pořadí od začátku.
- Fragmentace může nastat a měnit se při průchodu přes libovolný směrovač během směrování.
- **Složení datagramu** nastává až **na cílové stanici!**
- Fragmentaci lze zabránit nastavením příznaku (**Do'nt Fragment**), nicméně pak může být paket nedoručitelný.
- Fragmentace způsobuje řadu problémů a je lepší se jí vyhýbat!

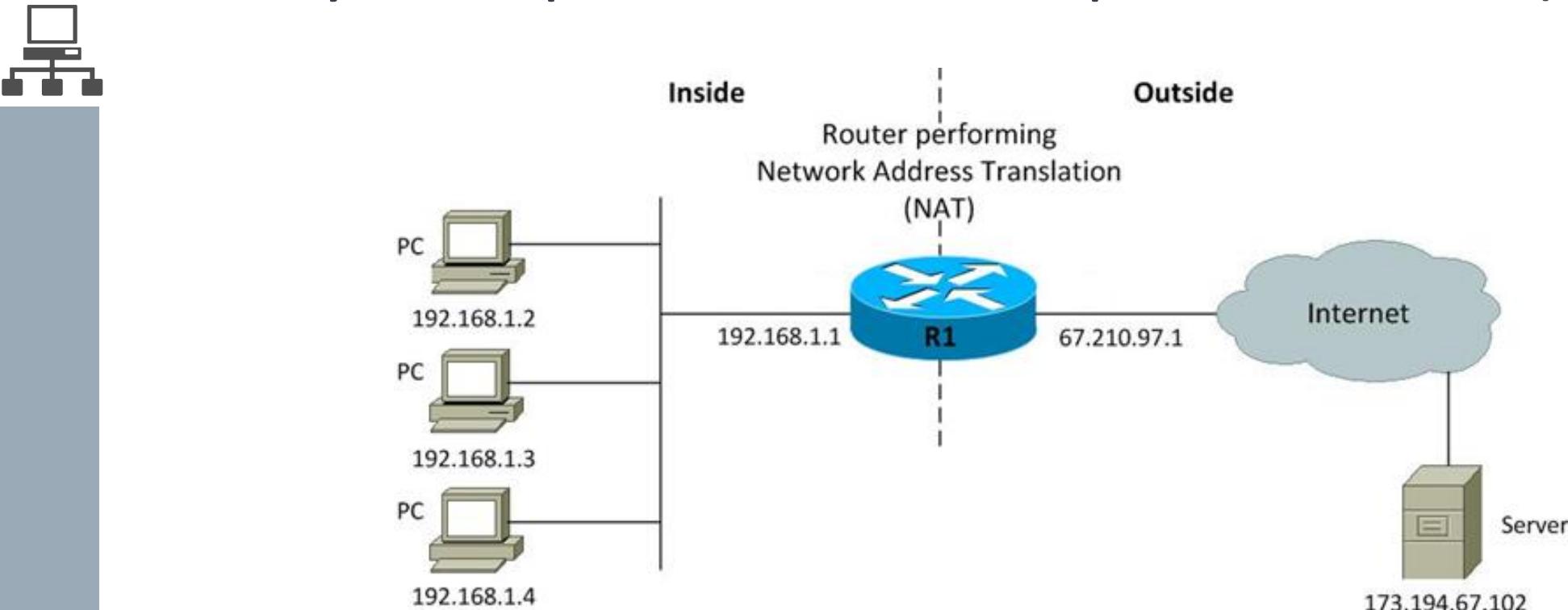
# Ukázka fragmentace IPv4 paketu



Zdroj: <http://www.networkscenarios.com/packet-fragmentation/>

Zdrojová stanice (Source) posílá data cílové stanici (Destination). Source je propojená se směrovačem (Routerem) rozhraním, na kterém je nastaveno, že rámec linkové vrstvy přenese najednou max. 520 bajtů dat (MTU=520). IP paket (hlavička + data) má dohromady  $500+20=520$  bytů a přenese se tedy k Routeru bez nutnosti fragmentace. Mezi Routerem a Destination je rozhraní, která dokáže najednou přenést max. 200 bajtů (MTU=200). Jelikož IP hlavička má 20 bajtů, lze přenést jen 180 bajtů dat. 500 bajtů se tedy rozdělí na  $2 \times 180 + 140$  bajtů, což znamená vyslání 3 rámců linkové vrstvy za sebou. Router označí rámce vzniklé fragmentací příznakem (identification) a konkrétním číslem (offset). Destination si z na základě offsetů a dat těchto 3 rámců složí původní IP paket po jejich doručení.

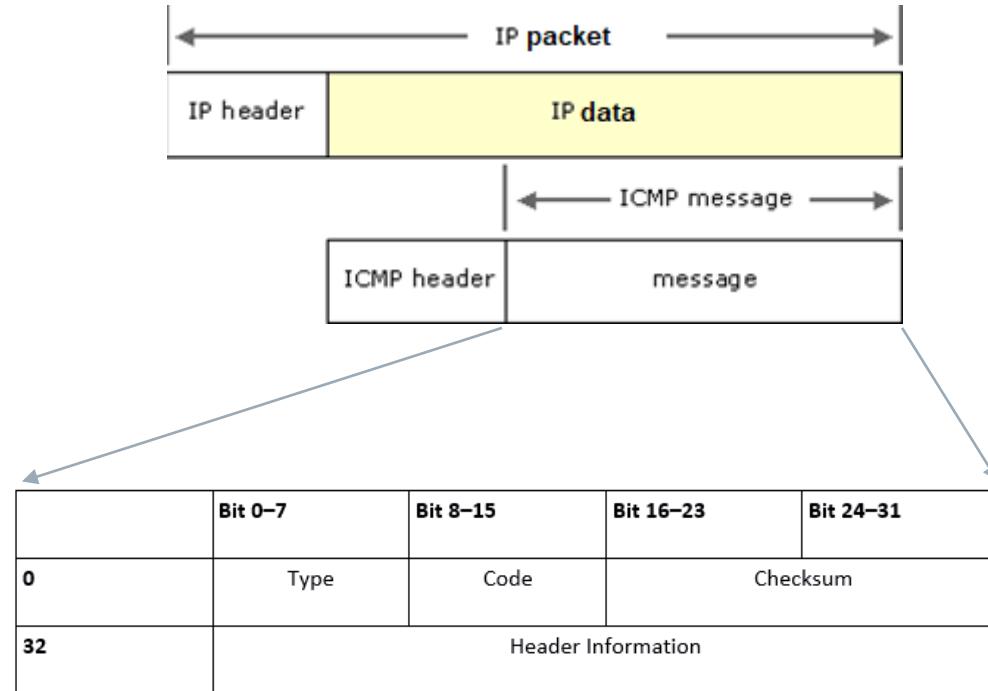
# Význam privátní IP adres, překlad adres (NAT)



Zdroj: <https://www.mustbegeek.com/configure-dynamic-nat-in-cisco-ios-router/>

Jelikož veřejných **IPv4 adres** není dostatek (počítačů je dnes více než těchto adres), nelze každému počítači v Internetu přiřadit **unikátní veřejnou adresu**. Řešení spočívá v tom, že **všechny privátní** (nesměrovatelné v Internetu) adresy vnitřní (inside) sítě (192.168.1.0/24) **se při průchodu** směrovačem do vnější (outside) sítě "vydávají" za jedinou veřejnou (směrovatelnou) adresu (67.210.97.1). Směrovač **vyměňuje při průchodu** paketu vnějším síťovým rozhraním **zdrojovou** adresu uvedenou v hlavičce paketu a **nahrazuje ji adresou svého vnějšího rozhraní**. Tomuto procesu se říká překlad adres (**Netwok Address Translation = NAT**).

# Internet Control Message Protocol (ICMP)



Zdroj: [https://www.researchgate.net/figure/ICMP-packet-structure\\_fig5\\_316727741](https://www.researchgate.net/figure/ICMP-packet-structure_fig5_316727741)

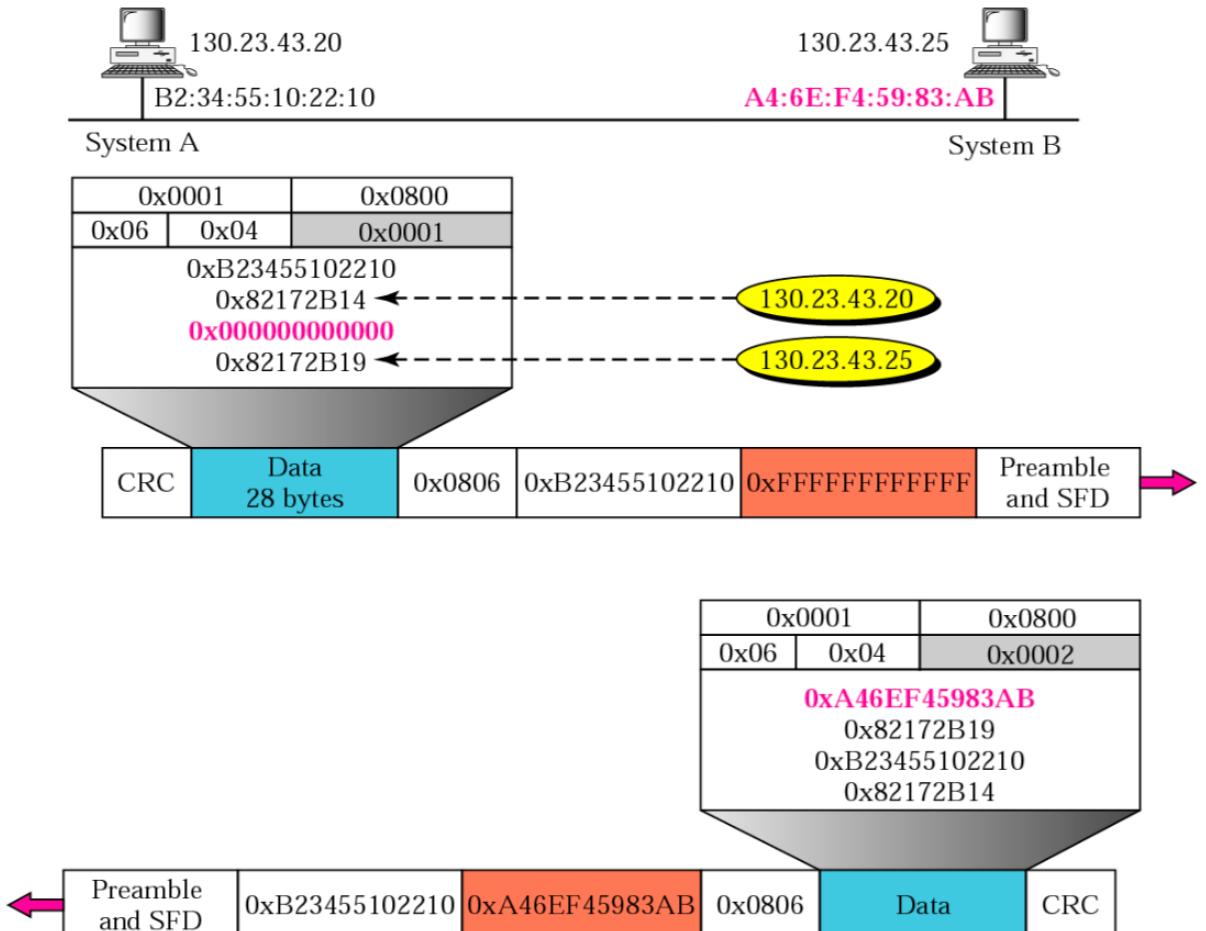
- Jednoduchý protokol sloužící k ověření funkčnosti vzdáleného zařízení.
- Posílají se zprávy
  - **ICMP REQUEST** (code 8), **ICMP REPLY** (code 0), **ICMP ERROR** (code 11).
- Utility **ping** (Packet Internet Groper) a **traceroute/tracert**.
- Traceroute **funguje opět např. přes ICMP REQUEST** s inkrementálně se zvětšujícím se TTL.



# Address Resolution Protocol (ARP)

- Je velmi důležitý pro **doručování IP paketů v lokální síti**.
- **V lokální síti probíhá doručování dat na základě linkových (MAC) adres (ty musí stanice znát).**
- Aby bylo možné doručit data mezi počítači v lokální síti jen na základě znalostí IP adres, je nutné umět namapovat tyto IP adresy na odpovídající MAC adresy síťových rozhraní.
- **ARP protokol slouží tedy k tomu, že dokáže najít pro konkrétní síťovou (IP) adresu její mapování na adresu linkovou (MAC).**
- Jak to dělá ? (viz dále).

# ARP dotaz → a odpověď ←



Zdroj: <https://ipcisco.com/lesson/address-resolution-protocol-arp/>

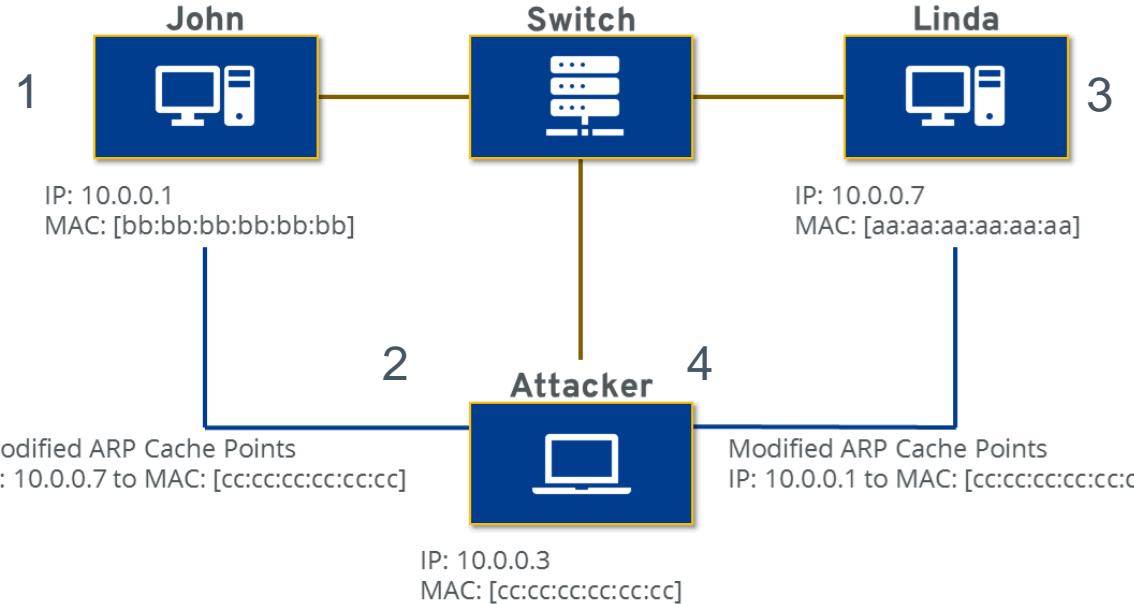
**Stanice A (130.23.43.20) chce komunikovat s B (130.23.43.20) v rámci lokální sítě, A zná pouze IP adresu B (stanice musí být ve stejné síti!).**

1. **A odešle žádost** žádost na zjištění MAC B přes ARP protokol, který se namapuje do eth. rámce s broadcastovou cílovou adresou (**0xFFFFFFFFFFFF**).
2. B žádost zachytí a vyplní do dat odpovědi svoji **MAC adresu**.
3. B odešle odpověď eth. rámcem A, ovšem již unicastově na adresu **0xB23455102210**.
4. A si MAC adresu přečte a zapíše si MAC B do své ARP tabulky.
5. A odešle data B unicastově přes eth. rámcem linkovou vrstvou.

# ARP spoofing



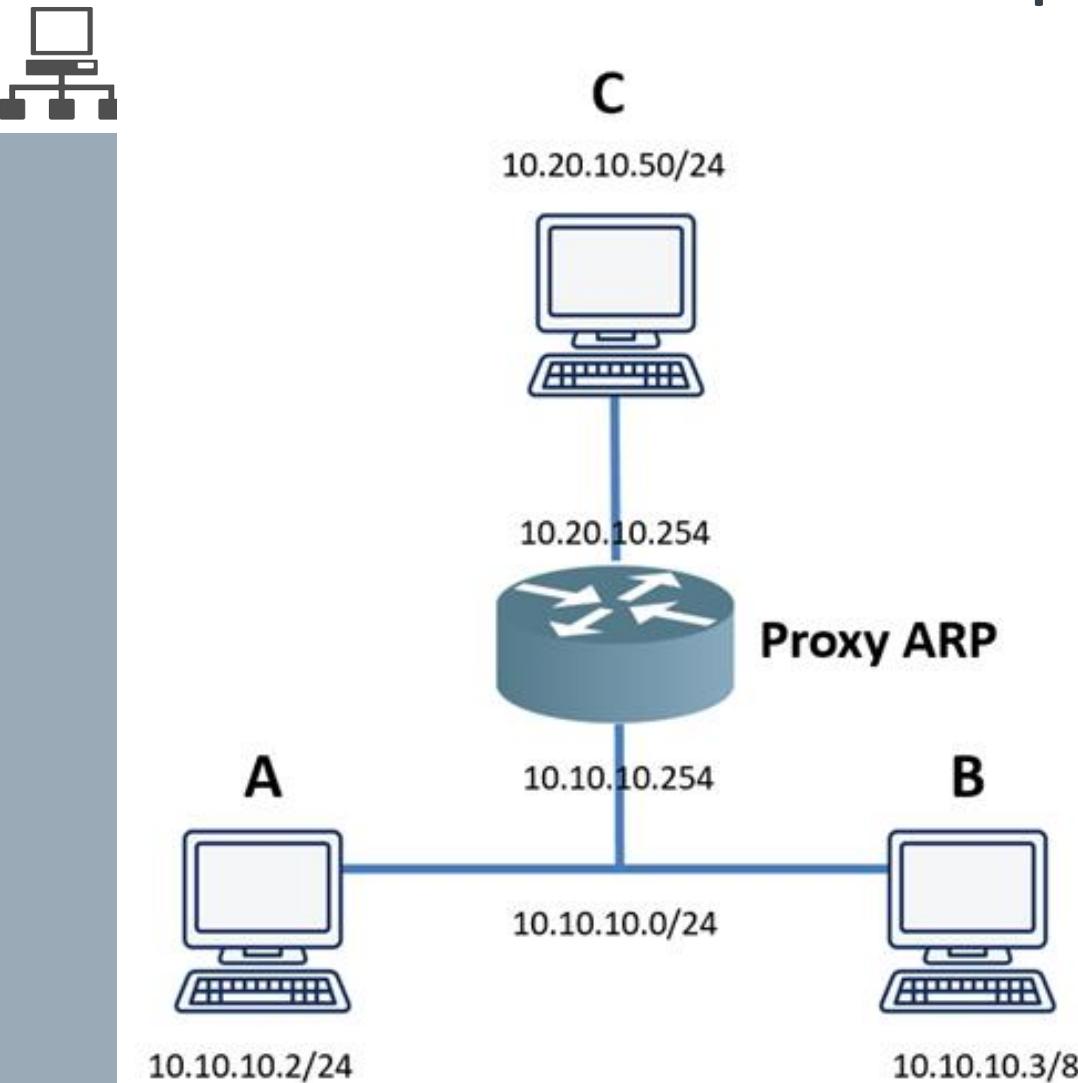
Zdroj: <https://www.ionos.com/digitalguide/server/security/arp-spoofing-attacks-from-the-internal-network/>



1. John chce komunikovat s Lindou (pošle ARP žádost).
2. Útočník (Attacker) Johnovi tvrdí (odpovídá na všechny ARP žádosti), že on je Linda (podvrhne ARP odpověď).
3. Linda chce komunikovat s Johnem pošle ARP žádost.
4. Útočník (Attacker) Lindě tvrdí, že on je John (podvrhne ARP odpověď).

Po těchto 4 krocích již **provoz mezi Johnem a Lindou může jít kompletně přes útočníka**, který jej může modifikovat, není-li tomu zamezeno např. na úrovni šifrování dat.

# Proxy ARP



Proxy ARP je mechanismus, díky kterému se směrovač pro zdrojovou stanici chová jako přepínač.

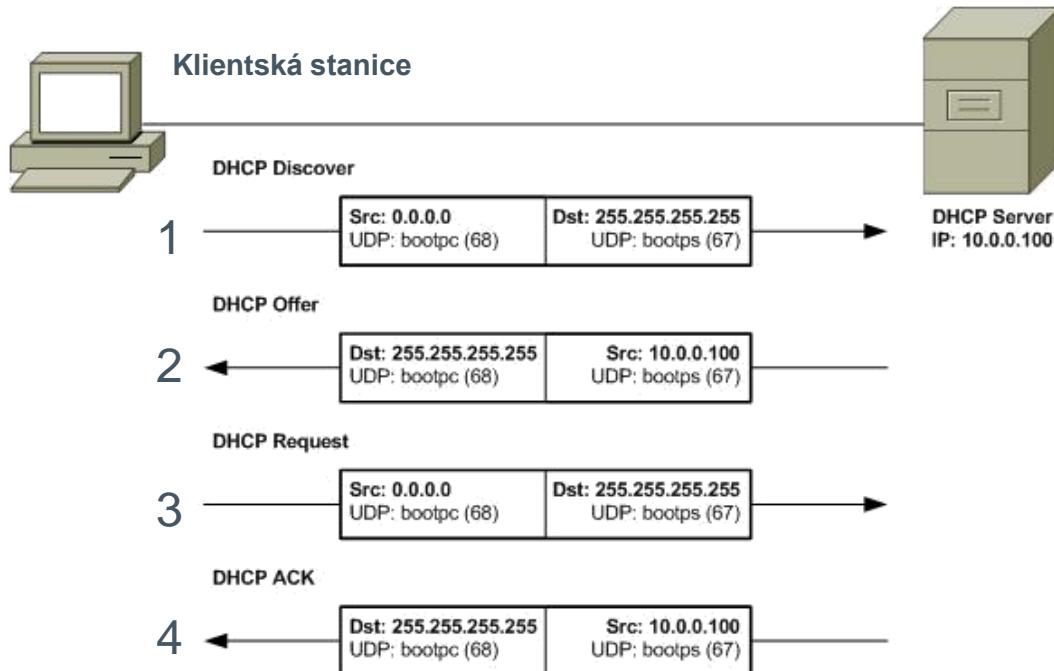
1. Stanice A, B a C jsou připojené k dvěma různým rozhraním směrovače.
2. Na směrovači je zapnutý **Proxy ARP**.
3. B chce komunikovat s C.
4. Jelikož C pro B leží ve stejné síti, bude B komunikovat napřímo (bez brány), posílá broadcastem ARP dotaz, aby zjistil MAC adresu B.
5. **Směrovač by normálně broadcast zastavil**, tentokrát ovšem (díky **Proxy ARP**) bude B tvrdit, že MAC adresa C je ta jeho (vrátí B ARP-REPLY se svoují MAC).
6. B pošle rámec směrovači.
7. Směrovač přepoše rámec C.

Zdroj: <https://networklessons.com/cisco/ccie-routing-switching/proxy-arp-explained>

# Dynamická konfigurace IPv4



- Účelem dynamické konfigurace je **automatické** nastavení **IP adresy na klientské stanici**.
- Nejznámějším protokolem umožňující dynamickou konfiguraci se nazývá **Dynamic Host Configuration Protocol (DHCP)**.
- DHCP protokol vychází ze staršího protokolu protokolu **BOOTP**.
- DHCP protokol je protokolem aplikační vrstvy (7. vrstva modelu OSI ), na **transportní vrstvě** (viz. 4. přednáška) využívá porty **67** (pro server) a **68** (pro klienta).



<http://www.netcontractor.pl/blog/wp-content/uploads/2010/05/DHCP-process.jpg>

Fáze přidělení adresy prostřednictvím DHCP protokolu

1. **DHPC Discover** = odeslání požavku pro nalezení DHCP serveru v síti klientskou stanicí.
2. **DHCP Offer** = odeslání volné IP adresy DHCP serverem klientovi.
3. **DHCP Request** = Zaslání požadavku pro přidělení dostupné IP adresy klientskou stanicí DHCP serveru.
4. **DHCP ACK** = Potvrzení rezervace DHCP serverem klientské stanici.

Veškerá komunikace je realizována prostřednictvím **broadcastu**, tudíž všechny klientské stanice mají přehled o přidělených adresách jiným klientským stanicím.

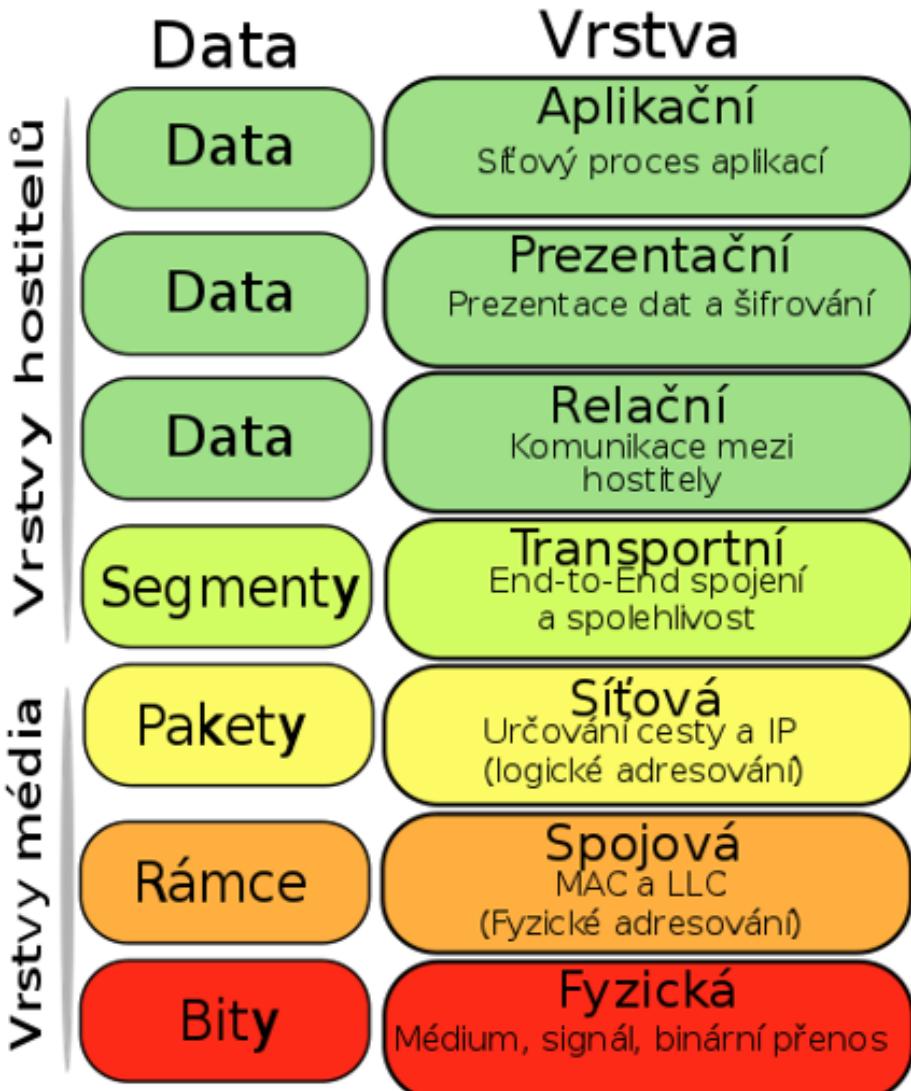
# Počítačové sítě

4. přednáška - IP protokol verze 6 (IPv6)





# Důvody zavedení a charakteristika IPv6



Dostupné veřejné adresy **IPv4** byly v roce **2011 rozebrány** a bylo potřeba tento fakt napravit.

Protokol **IPv6** zavádí **128 bitové adresy**, což výrazně zvyšuje velikost adresní prostor.

Protokol **IPv4** sám o sobě neobsahuje metodu automatické konfigurace, **IPv6** však ano.

Překlad IP adres (**NAT**) není již potřeba (adres je dostatek), nicméně lze použít kvůli bezpečnosti resp. oddělení síťového provozu vnější a vnitřní sítě.

Protokol **IPv6** definuje **mobilní** IP adresy, které jsou dosažitelné i mimo domácí síť.

Protokol **IPv6** obsahuje i metodu přeposílání provozu do sítí **IPv4** kvůli zpětné kompatibilitě.



# Osnova přednášky

- Zařízení a komunikační operace v IPv6
- Pakety v IPv6
  - Formáty, zápisy a konfigurace.
- Řízení provozu v IPv6
  - ICMPv6, objevování sousedů.
- Vybrané principy a technologie používané v IPv6
  - Směrování, inverzní objevování sousedů, DHCPv6, mobilita a TEREDO.

# Certifikace zařízení IPv6

- Zařízení podporující IPv6 se označují speciálním logem IPv6 Ready (stříbrným či zlatým).
- Barva loga odpovídá konkrétním funkcím, které daná certifikace podporuje.



Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

- **Stříbrná:** IPv6, ICMPv6, objevování sousedů, bezstavová konfigurace (viz dále).
- **Zlatá:** vše co stříbrná a dále: IPsec, mobilita, DHCP6, multicast.



# Podporované druhy adres u IPv6 dle druhu komunikačních operací

- Unicastové
- Multicastové
- Anycastové

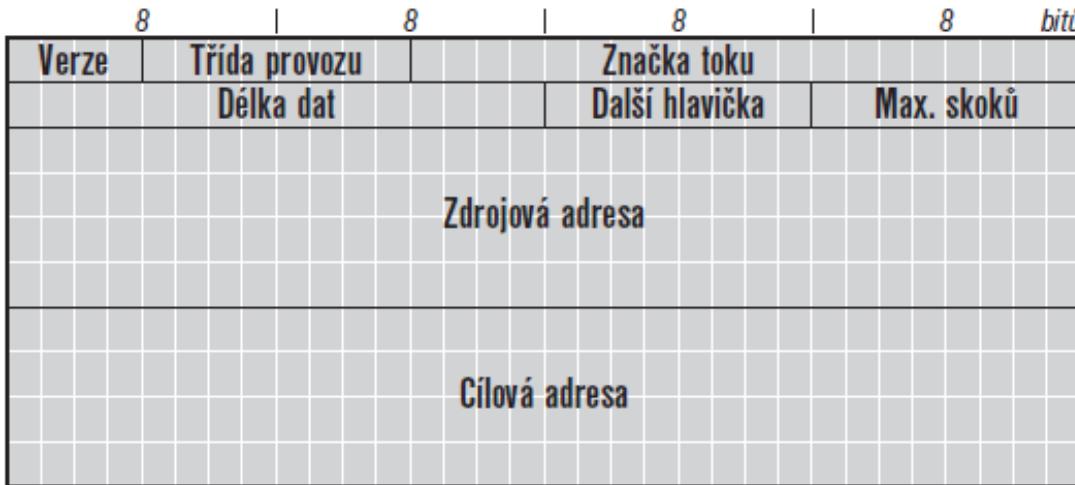
**Broadcastové adresy neexistují, jsou nahrazeny specifickými multicastovými!**

# Pakety v IPv6



- Paket v IPv6 = **Základní hlavička** + **(Rozšiřující hlavička)\*** + **Data**.
- **Základní hlavička** = struktura, která obsahuje informace, které souvisí se základními parametry nutnými pro doručení paketu (např. zdrojová + cílová adresa, TTL, velikost dat), je přítomna vždy.
- **Rozšiřující hlavička** = struktura, která obsahuje specifické informace sloužící pro doručení konkrétního paketu. Nemusí být přítomná vždy. Rozšiřující hlavičky se **spojují za sebe** (viz. dále).
- **Data** = hlavičky a data protokolů vyšších vrstev.
- Paket se může **doručovat celý** anebo **po částech** (fragmentace). Velikost fragmentu paketu na rozdíl od IPv4 **určuje zdrojová stanice** (u IPv4 směrovač) a po celou dobu doručování se tato velikost nemění (u IPv4 jí mohl měnit každý směrovač).
- Pro zopakování, u **IPv4** mohlo docházet k fragmentaci na **každém směrovači**, kterým paket prochází.

# Struktura základní hlavičky paketu IPv6



Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

**Verze** = Označení protokolu IPv6.

**Třída provozu** = priorita paketu, umožňuje garantovat kvalitu služeb.

**Značka toku** = informace pro směrovač, na základě které identifikuje posloupnost paketů jdoucích společně ke konkrétnímu cíli (nepoužívá se běžně, pomáhá efektivnějšímu zpracování).

**Délka dat** = celková velikost přenášených dat v bajtech bez hlavičky(ek), max 64KB.

**Další hlavička** = typ rozšiřující hlavičky, která následuje za hlavní hlavičkou nebo typ dat.

**Max. skoků** = obdoba TTL u IPv4 (viz 3. přednáška, slajd 14).

**Zdrojová adresa** = IPv6 adresa zdrojové stanice.

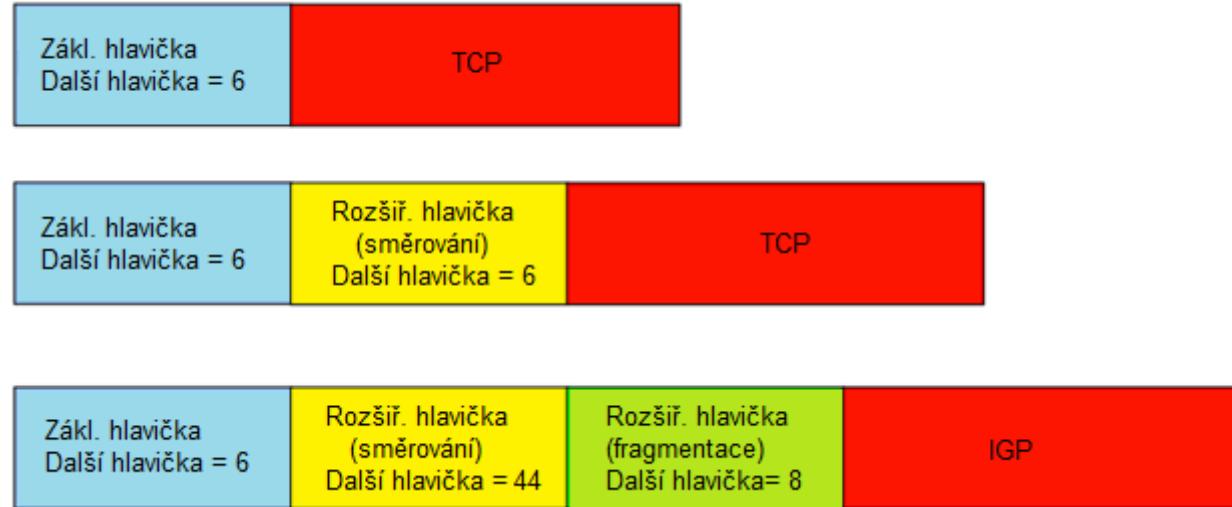
**Cílová adresa** = IPv6 adresa cílové stanice.

# Zřetězení hlaviček u IPv6



- Protokol **IPv4** definuje **pro paket pouze jednu hlavičku**, která obsahuje veškeré informace týkající se jeho doručování (např. fragmentace, offset atd.).
- Paket protokolu IPv6 **může obsahovat hlaviček více**, první hlavička se nazývá **základní** a další se označují jako **rozšiřující**.
- **Rozšiřující hlavičky** jsou umístěné za sebou **v definovaném pořadí**, každá má specifickou funkci (např. směrování, fragmentace, určení typu přenášených dat atd.), rozlišenou kódem.
- Pro zřetězení hlaviček se používá specifická položka v každé hlavičce - **další hlavička** (viz přechozí slajd).
- **Velká výhoda zřetězování hlaviček spočívá v tom**, že se přenášejí pouze hlavičky, které jsou skutečně potřeba a oproti hlavičce obsažené v IPv4 mohou být rozšiřující hlavičky podrobnější.
- **Počet rozšiřujících hlaviček může být pro každý paket individuální.**

# Zřetězení hlaviček - příklad



Pod označením další se rozumí označení následující rozšiřující hlavičky paketu.

**Označení rozšiřující hlavičky z hlediska funkce paketu:**

6 = identifikace protokolu, 43 = směrování, 44 = fragmentace, 50 = šifrování, 51 = autentizace, 59 = poslední hlavička (bez dat), 60 = volba cíle, 62 = mobilita.

**Označení rozšiřující hlavičky z hlediska typu dat:**

6 = TCP, 8 = IGP, 9 = EGP, 17 = UDP - tyto protokoly budou probírány v rámci 6. a 7. přednášky.

# Adresy v IPv6 a jejich zápis



- IPv6 adresa obsahuje 128 bitů = 8 x 16 bitů zapsáno hexadecimálně a odděleno :

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

1234:**0217**:2344:**0000**:2345:**0000**:**0000**:1111

V zápisu lze provést zkracování prostřednictvím vynechání nevýznamných nul (zleva)

1234:**217**:2344:**0**:2345:**0**:**0**:1111

nebo dokonce úplným vynecháním nul a nahrazením jejich posloupnosti ::

1234:**217**:2344:**0**:2345::1111

- adresu 0000:0000:0000:0000:0000:0000:0000:0000 tedy lze zkrátit na ::
- :: znamená vždy maximální počet 0 zase sebou, lze ovšem tuto notifikaci v zápisu použít maximálně jednou, protože se dopočítává do 128b.

# Druhy adres v IPv6

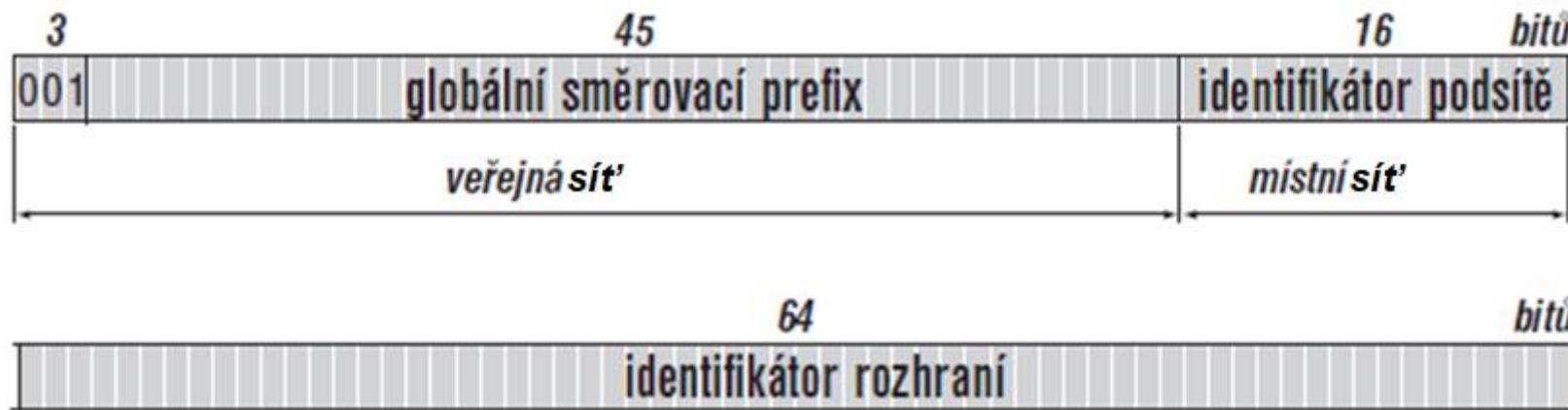


- **Lokální linkové (LL)** = adresy patřící konkrétním síťovým rozhraním, které se nastavují automaticky dle MAC adres těchto rozhraní (fe80::/10).
- **Skupinové (multicastové) adresy** (ff00::/8)
- **Unikátní lokální (UL)** = obdoba **neveřejných** u IPv4 (fc00::/7).
- **Smyčka/Loopback** (::1/128).
- Nedefinovaná adresa (::/128).
- **Přechodové** = slouží pro doručení dat do IPv4 sítí (64:ff9b::/96)
- **Individuální globální** = obdoba veřejných u IPv4 (všechny zbývající adresy).

# Individuální Globální Adresy (IGA)



- Jsou směrovatelné v Internetu, fungují tedy obdobně jako veřejné u IPv4.
- **IGA adresa (po bitech) = 001 + glob. směrovací prefix + identifikátor podsítě + identifikátor rozhraní.**
- **Globální směrovací prefix (GSP)** = sekvence prvních 45 bitů IPv6 adresy, která označuje konkrétní síť, která je dosažitelná v rámci Internetu (veřejná síť).
- **Identifikátor podsítě** = sekvence 16 bitů adresy, které se používají pro směrování do konkrétní podsítě (vnitřní síť) v rámci sítě určené konkrétním GSP.



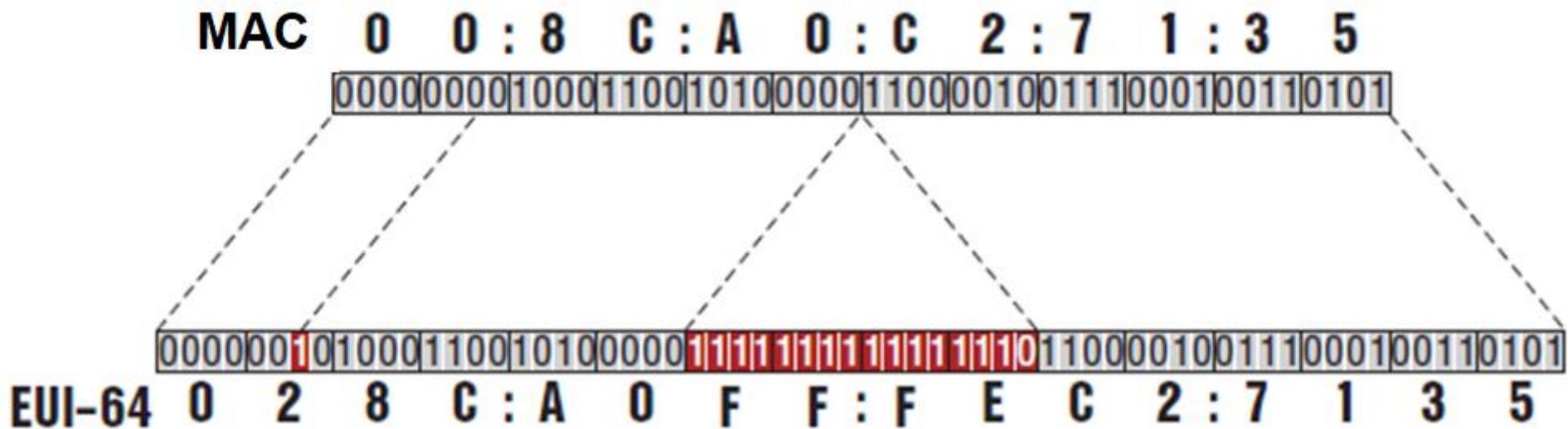
Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

- **Místní síť'** = síť jednoho poskytovatele (ISP).
- **Identifikátor rozhraní** = posloupnost 64bitů, která má za úkol určit konkrétní síťové rozhraní, označuje se též jako **EUI-64** a odvozuje se z **MAC** adresy daného rozhraní.

# Odvození EUI-64 z MAC adresy rozhraní



- Identifikátor rozhraní (EUI-64) se z MAC adresy (např. u Ethernetu) vytvoří tak, že se mezi 3. a 4. bajt MAC adresy se vloží dva bajty FF a FE a 7.bit zleva MAC adresy se invertuje .

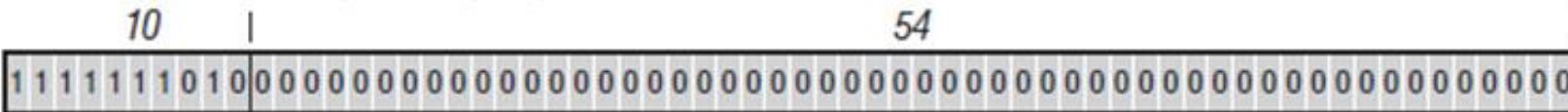


Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

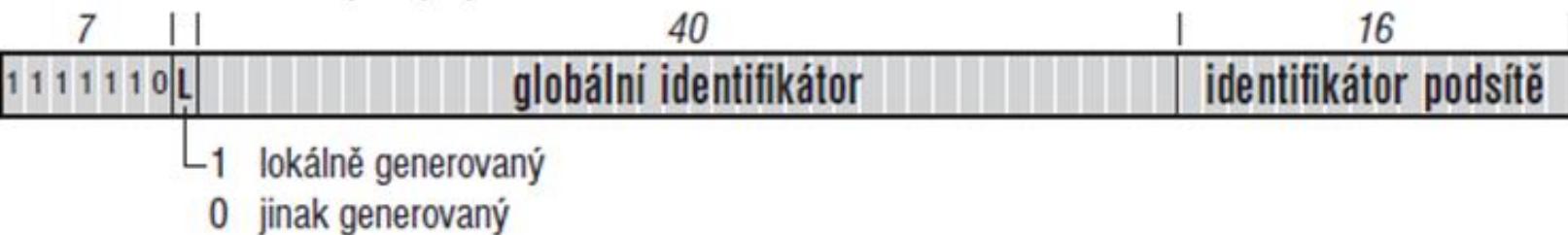
# Lokální Linkové (LL) a Unikátní Lokální (UL) adresy



## Lokální linkové (*fe80::/10*)



## Unikátní lokální (*fc::/7*)



Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

- LL adresa se používá pro **adresaci konkrétního síťového rozhraní v lokální síti**, nastaví ji automaticky operační systém dle MAC adresy daného síťového rozhraní.
- UL adresa se používá pro směrování ve stejné síti a musí se na rozdíl od LL nakonfigurovat.
- **Globální identifikátor** = posloupnost 40 bitů, která určuje konkrétní místní síť.
- **Identifikátor podsítě** = posloupnost 16 bitů, která určuje konkrétní podsíť v místní síti.
- LL i UL adresa jsou zakončeny **EUI-64 dle rozhraní**, stejně jako IGA.
- LL i UL adresy nejsou směrovatelné z Internetu.

# Řízení provozu v IPv6



- Protokol **IPv4** využívá několik dalších protokolů, které mu napomáhají v jeho činnosti (např. ICMP, ARP, DHCP atd.) – viz 3. přednáška, slajd 22.
- IPv6 dokáže zabezpečit svoji činnost pouze s využitím **jediného protokolu Internet Control Message Protocol for IPv6 (ICMPv6)**, který byl oproti původnímu ICMP zásadně přepracován.
- Paket ICMPv6 je přenášen **uvnitř IPv6 jako data**. Toto je jako obdobné u ICMP pro IPv4, viz 3. přednáška, slajd 17.
- ICMPv6 zabezpečuje celou sadu funkcí pro řízení provozu, která se označuje jako objevování sousedů (**Neighbor Discovery**). Funkce jsou realizovány prostřednictvím typů specifických zpráv → viz dále.

# ICMPv6 formát paketu



Příklad typů:

<i>chyby</i>	<i>objevování sousedů</i>	<i>inverzní objevování sousedů</i>
1    cíl je nedosažitelný	133    výzva směrovači	141    IND výzva
2    příliš velký paket	134    ohlášení směrovače	142    IND ohlášení
3    vypršela životnost paketu	135    výzva sousedovi	<i>mobilita</i>
4    problém s parametry	136    ohlášení souseda	144    žádost o adresy domácích agentů
<i>echo</i>	137    přesměrování	145    odpověď s adresami domácích agentů
128    požadavek na echo	148    žádost o certifikační cestu	146    žádost o mobilní prefix
129    odpověď na echo	149    ohlášení certifikační cesty	147    ohlášení mobilního prefixu
<i>MLD (skupinové adresování)</i>	<i>informace o uzlu</i>	154    rychlé předávání
130    dotaz na členství ve skupině	139    dotaz na informace	<i>objevování skupinových směrovačů</i>
131    ohlášení členství ve skupině	140    odpověď s informacemi	151    ohlášení skupinového směrovače
132    ukončení členství ve skupině		152    výzva skupinovému směrovači
143    ohlášení členství ve skupině (MLDv2)		153    ukončení skupinového směrovače

Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

Detailní specifikace všech zpráv je nad rámec tohoto předmětu. Dále budou probírány jen zprávy 133-136, popř. 141 a 142.

# Objevování sousedů (Neighbor Discovery)



- Komplexní způsob pro řízení provozu IPv6.
- Nabízí řadu funkcí jako např.
  - **Automatickou konfiguraci koncových stanic** (obdoba DHCP pro IPv4).
  - **Zjišťování linkových adres v lokální síti** (náhrada ARP pro IPv4).
  - **Rychlá aktualizace neplatných položek**
    - Detekuje nedosažitelné IPv6 adresy (např. kvůli vypnutí stanice).
  - **Hledání směrovačů**
    - Automaticky zjišťuje, přes které směrovače je možné přeposílat provoz určený pro stanice umístěné v jiných sítích.
  - **Přesměrování**
  - **Ověřování dostupnosti sousedů** (v místní síti, u známých sousedů) .
  - **Detekce duplicitních adres** (hledání a deaktivace chybných adres).

# Automatická konfigurace koncové stanice



- Automatická konfigurace = metoda, jejímž účelem je získání individuální globální adresy IPv6 a dalších parametrů, které umožní stanici komunikovat uvnitř místní sítě i Internetu.
- Automatická konfigurace může být **bezstavová** nebo **stavová**.
- **Bezstavová** – založená na myšlence, že v síti existují směrovače, které mají kompletní informace týkající se provozu a dokáží je předat dále.
- **Stavová** – v síti existuje specializovaný server, který zajišťuje přidělování adres, používá se protokol **DHCPv6**. V principu totožné s IPv4 (3. přednáška slajd 25).

# Bezstavová automatická konfigurace

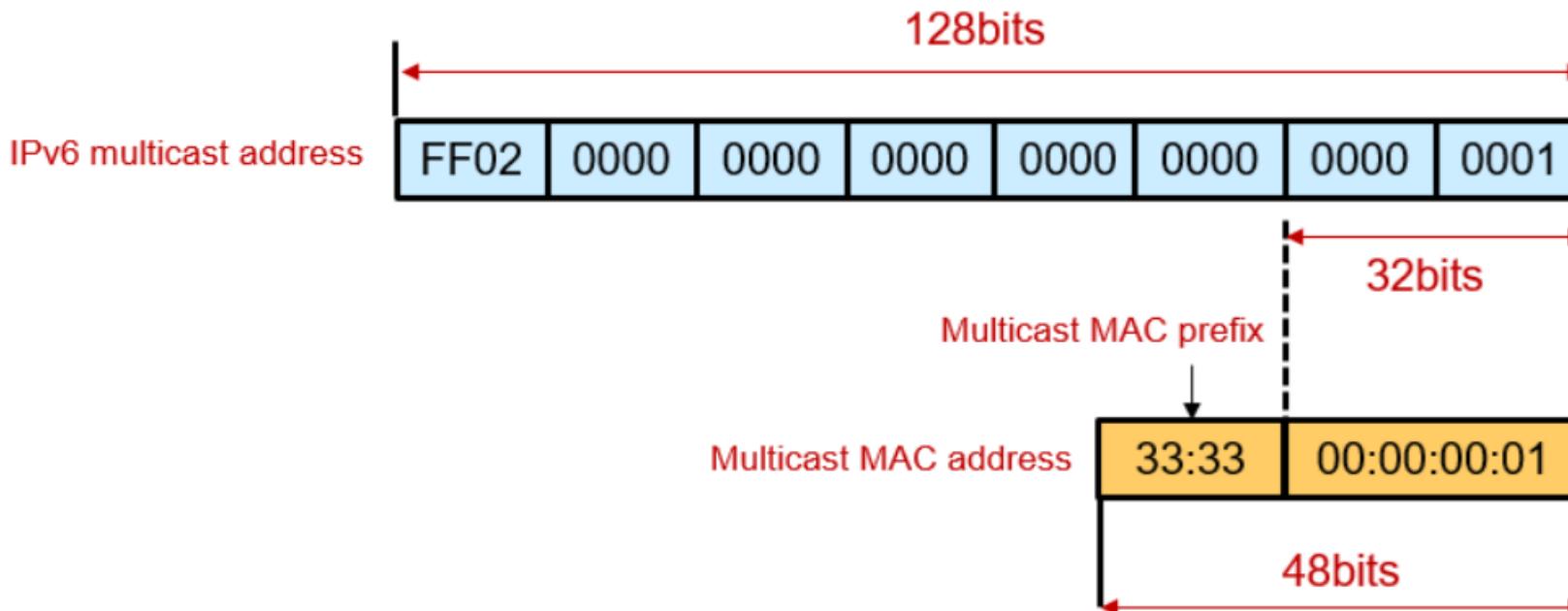


- Směrovač zasílá do sítě zprávu **Ohlášení směrovače (RA = Router Advertisement)**, která obsahuje **konfiguraci sítě**.
- Konfiguraci od směrovače využije stanice, která se chce účastnit komunikace, pro nastavení své IPv6 adresy.
- RA se posílá **periodicky** anebo **na výzvu** - stanice o konfiguraci požádá příslušný směrovač prostřednictvím zprávy **Výzva směrovači (RS = Router Solicitation)**.
- Zprávy RA a RS jsou doručovány pomocí protokolu ICMPv6.
- RS je zasílána skupinově na adresu **FF02::2**.
- RA se odesílá skupinově na adresu **FF02::1**.
- RA osahuje **64 bitový prefix**, který stanice zkombinuje s EUI-64 odvozeného od MAC adresy svého síťového rozhraní a získá tak použitelnou individuální globální IPv6 adresu. **Směrovač se stává pro stanici automaticky bránou**.
- RS i RA jsou zasílány prostřednictvím linkové vrstvy a je potřeba umět namapovat použité skupinové IPv6 adresy na MAC adresy (viz dále).

# Mapování skupinové IPv6 adresy na MAC adresu

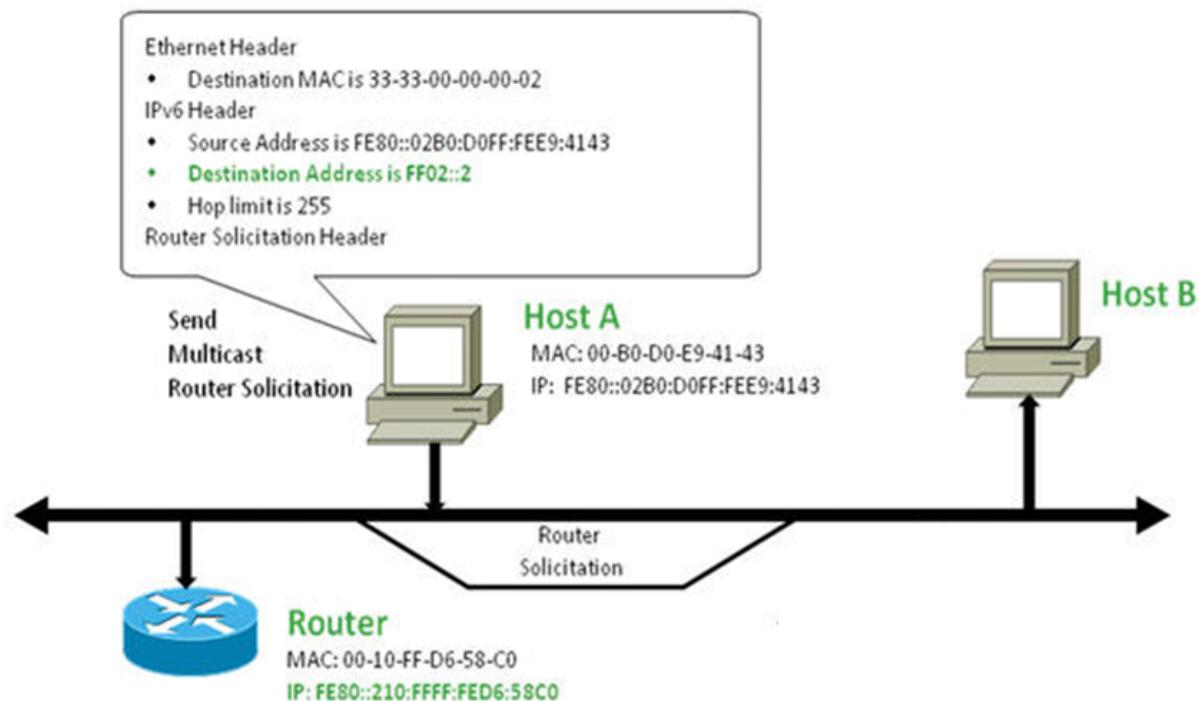


- MAC adresa se odvodí z posledních 32 bitů dané IPv6 adresy, před které se uvede specifický 16 bitový prefix 33:33 (v hexa). Přepínače pracující na linkové vrstvě musí toto mapování rovněž podporovat!



Zdroj: <https://forum.huawei.com/enterprise/en/a-peek-at-ipv6-05-understanding-ipv6-addresses-multicast-address-reprint/thread/523345-861>

# Výzva směrovači - příklad



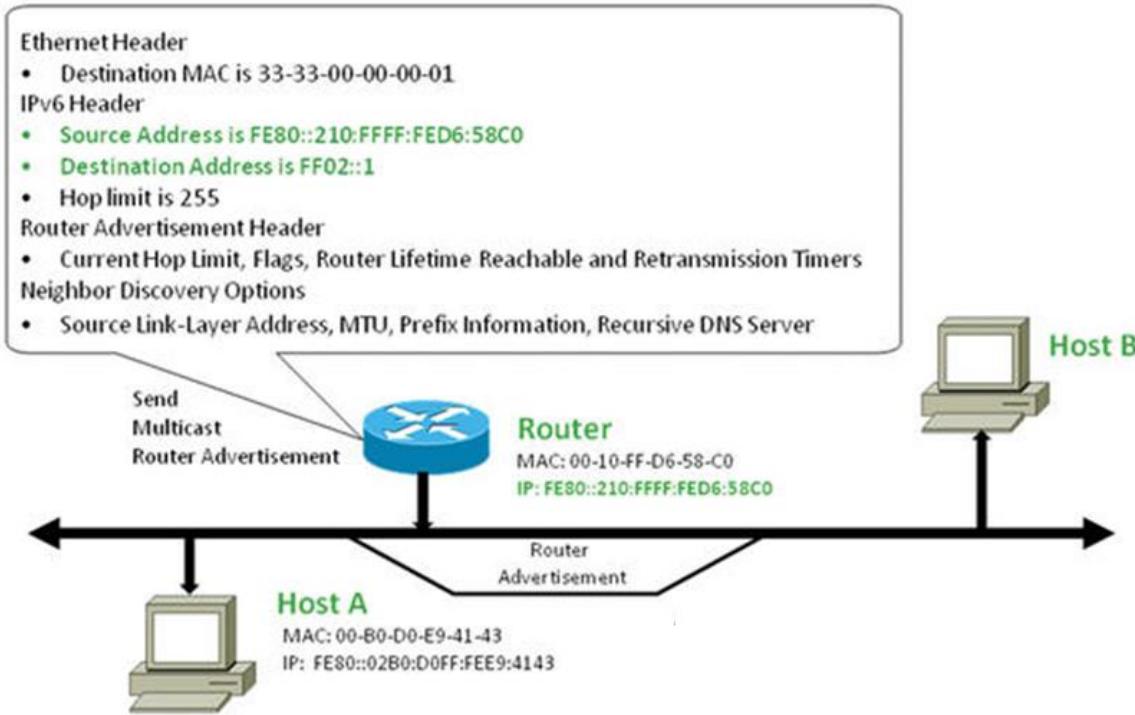
Zdroj: [https://www.sharetechnote.com/html/IP\\_Network\\_IPv6.html](https://www.sharetechnote.com/html/IP_Network_IPv6.html)

## Poznámka.

IP adresa FF02::2 se namapuje na MAC adresu 33-33-00-00-00-02.

z MAC adresy 00-B0-D0-E9-41-43 se vytvoří LL adresa FE80::02B0:D0:FF:FEE9:4143 (viz slajd 13).

# Ohlášení směrovače - příklad



Zdroj: [https://www.sharetechnote.com/html/IP\\_Network\\_IPv6.html](https://www.sharetechnote.com/html/IP_Network_IPv6.html)

**Poznámka.**

**IP adresa FF02::1** se namapuje na MAC adresu **33-33-00-00-00-01**.

Součástí ohlášení je **64 bitový prefix**, který Host A zkombinuje s EUI-64 svého síťového rozhraní a získá IGA adresu (viz slajd 12).

# Zjišťování linkových adres v lokální síti



- Pro komunikaci v lokální síti je nutné mimo IPv6 adres i znát odpovídající MAC adresy rozhraní okolních stanic, na kterých jsou tyto MAC adresy nastaveny.
- U sítích **IPv4** mapování IP adres na MAC adresy zajišťuje **protokol ARP** (3. přednáška slajd 26), pro sítě IPv6 již ARP však použít nelze (není definován), pro zjišťování linkových adres je zahrnuto do protokolu ICMPv6.
- **Algoritmus zjištění linkové adresy na základě známé IPv6 adresy:**
  - 1. **Sestavení adresy vyzývané cílové stanice** (SA = Solicitation Address) zdrojovou stanicí. Jedná se o skupinovou adresu, odvodí se ze známé IPv6 adresy (viz dále).
  - 2. **Poslání zprávy Výzva sousedovi** (NS = Neighbor Solicitation), skupinově na SA prostřednictvím linkové vrstvy.
  - 3. Přijetí zprávy NS cílovou stanicí a zaslání odpovědi prostřednictvím zprávy **ohlášení souseda** (NA = Neighbor Advertisement) .
  - 4. Načtením MAC adresy souseda zdrojovou stanicí a uložení této adresy do **tabulky sousedů**.

# Krok 1: Sestavení adresy vyzývané stanice

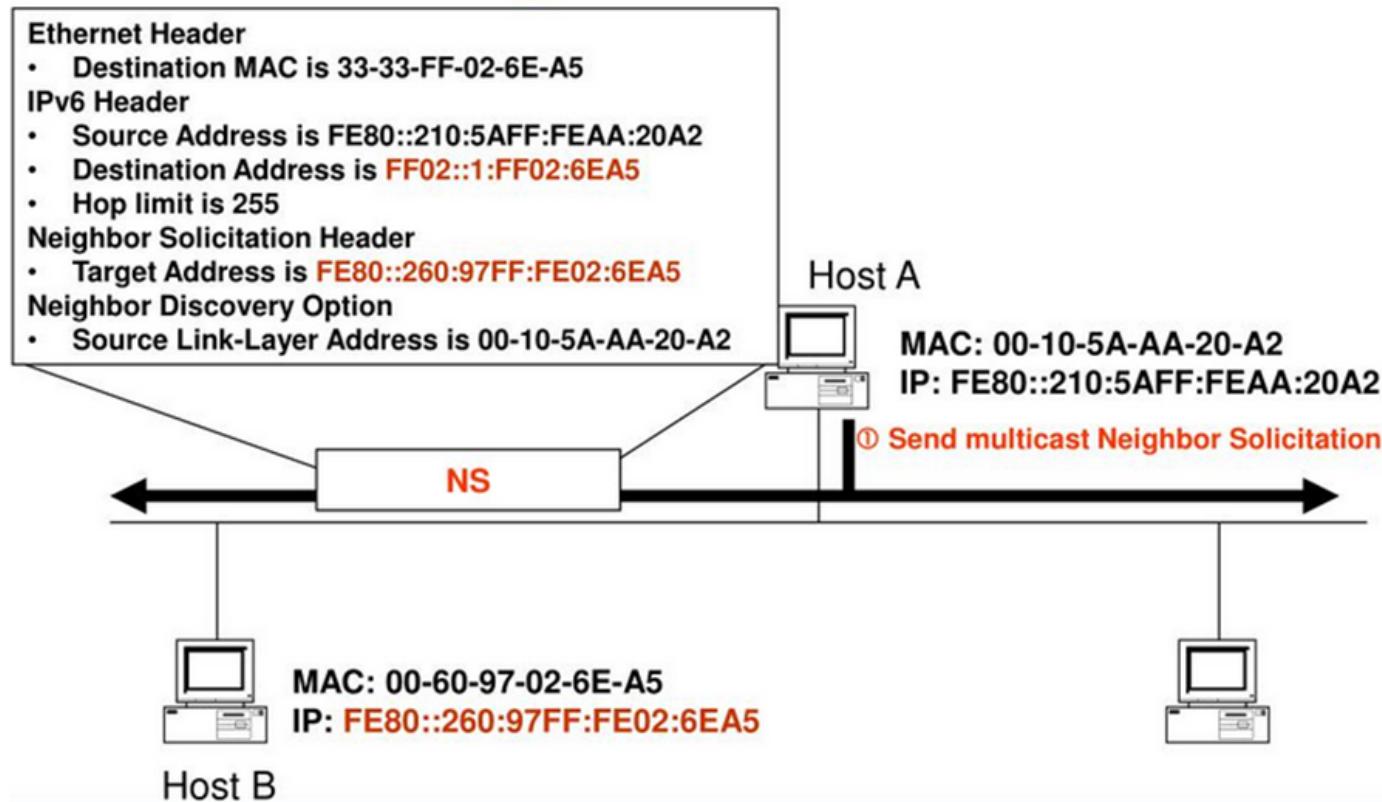


- Pro zjišťování linkových adres se používá specifický skupinový prefix
  - FF02::1:FF00:/104
- Zdrojová stanice vytvoří **IPv6** adresu cílové stanice pouze z posledních **24 bitů její IPv6 adresy** připojením za výše uvedený prefixu.
- Tato nová adresa se namapuje do rámce linkové vrstvy dle principu uvedeného na slajdu 20, tzn. vezme se posledních 32 bitů a přidá se prefix 33:33.
- Kompletní řešený příklad je na dalším slajdu.

## Krok 2: Výzva sousedovi – příklad



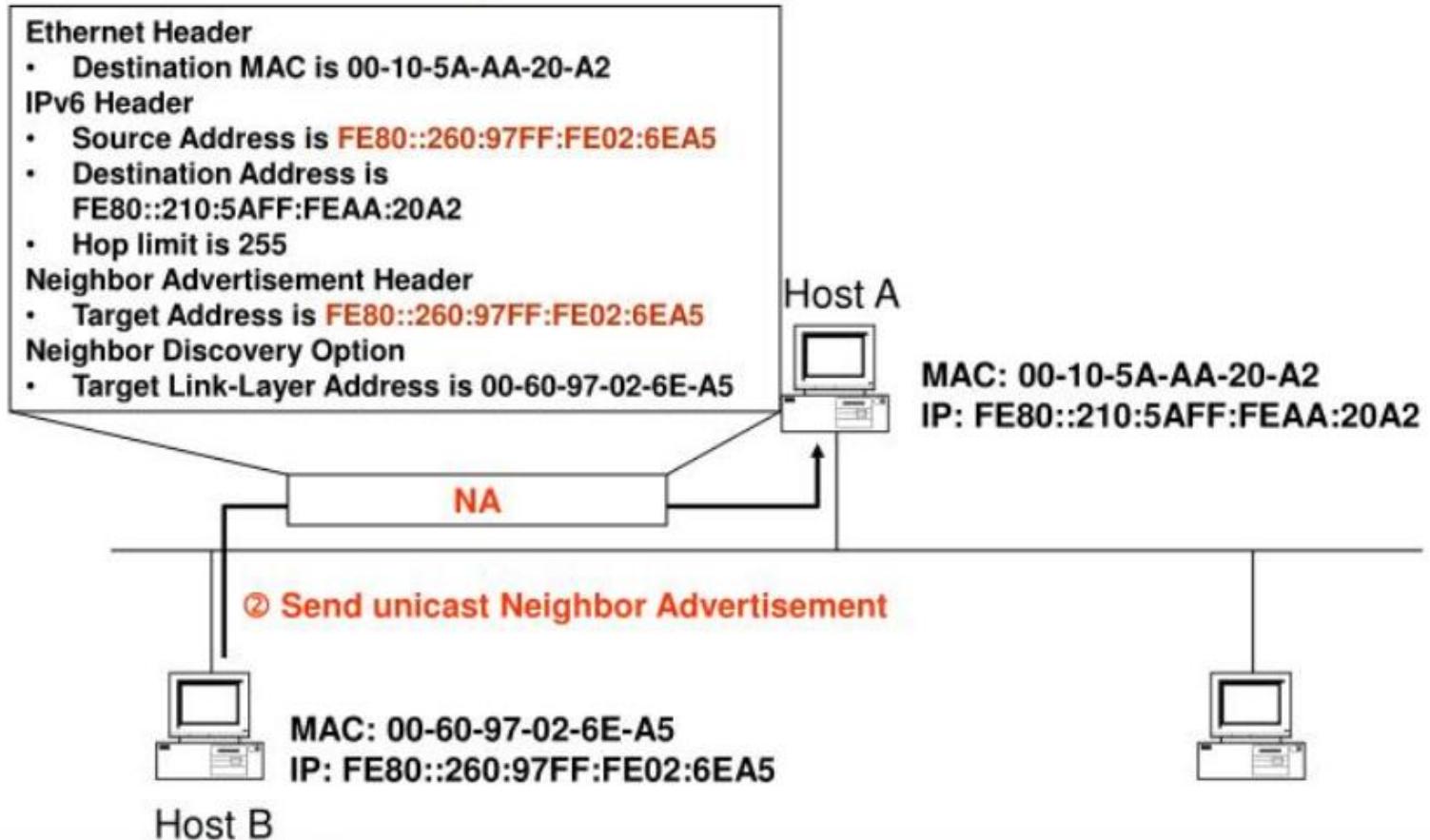
- Host A che zjistit MAC adresu rozhraní, které má nastavenu IPv6 adresu **FE80::260:97FF:FE02:6EA5** patřící stanici Host B.



Zdroj: [https://www.sharetechnote.com/html/IP\\_Network\\_IPv6.html](https://www.sharetechnote.com/html/IP_Network_IPv6.html)

- Host A určí odpovídající skupinovou IPv6 adresu (FF02::1:FF02:6EA5), vytvoří zprávu typu NS a odešle ji prostřednictvím linkové vrstvy na cílovou MAC adresu 33-33-FF-02-6E-A5.

# Krok 3: Ohlášení souseda – příklad

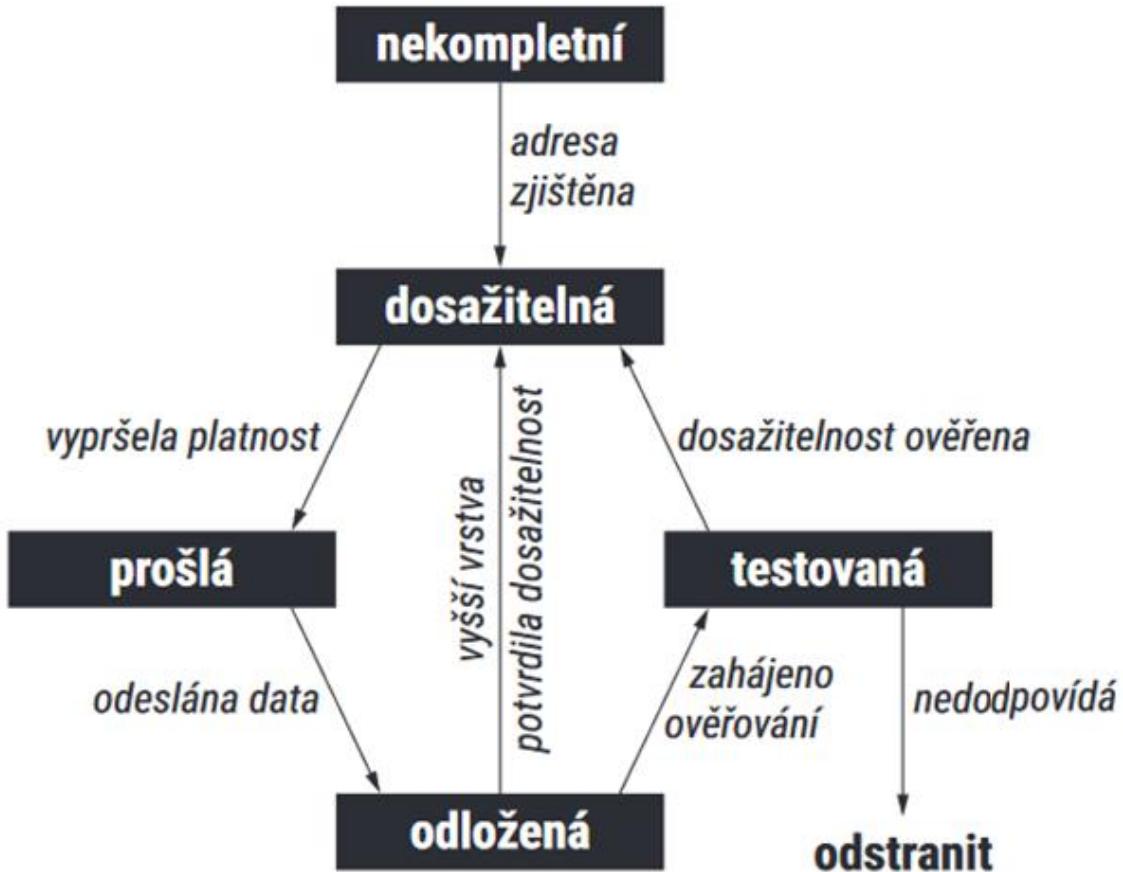


## Krok 4:

- Stanice Host B zprávu NS přijme a zasílá stanici Host A zprávu (unicastově na její MAC adresu) se svoují MAC adresou.

Zdroj: [https://www.sharetechnote.com/html/IP\\_Network\\_IPv6.html](https://www.sharetechnote.com/html/IP_Network_IPv6.html)

# Stavy adres v tabulce sousedů u IPv6



Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

Protože se v tabulce sousedů udržují jen nejčerstvější adresy, chová se tabulka jako cache.

# Vybrané principy a technologie u IPv6



- **Směrování dat**
  - Metoda, jak stanice s IPv6 adresou doručuje data cílové stanici.
- **DHCPv6**
  - Alternativa k bezestavové konfiguraci, obdoba DHCP pro IPv4.
- **Inverzní objevování sousedů**
  - Určení IPv6 adresy k odpovídající MAC adrese rozhraním, na kterém je IPv6 adresa nastavena.
- **Přechodový mechanismus IPv6 do IPv4**
  - Kvůli dosažitelnosti adres stanici v sítích IPv4 stanicemi v sítích IPv6.
- **TEREDO**
  - Určeno pro použití IPv6 v lokální síti, která IPv6 nepodporuje.
  - Implementace této technologie pod MS-Windows se nazývá MIREDO.
- **Mobilita u IPv6**
  - Způsob, jak zajistit dosažitelnost konkrétní stanice přes její IPv6 adresu, když se tato stanice nachází v jiné místní síti.

# Inverzní objevování sousedů



- Metoda dokáže pro známou linkovou adresu konkrétního rozhraní odvodit odpovídající adresu IPv6.
- Metoda posílá žádost prostřednictvím zprávy **Inverzní výzva sousedovi skupinově** na FF02::1 a přijímá odpověď unicastově prostřednictvím zprávy **Inverzní ohlášení souseda**.
- Vlastní způsob je až na informace určující typ dotazu v principu totožný s předchozí metodou, liší se v druhu zasílané informace.

# Směrování v sítích využívajících IPv6



- Směrování = způsob hledání cesty pro doručování dat mezi zdrojovou a cílovou stanicí (detailněji na příští přednášce).
- Stanice využívající IPv6 jsou schopné si zjistit dostupné směrovače automaticky pomocí ND.
- Stanice si udržují pro směrování směrovací tabulky (obdoba IPv4).
- Pro neznámé cílové stanice je známý implicitní směrovač, což je obdoba brány u IPv4.
- O tom, který směrovač je implicitní, se stanice dozví z ohlášení směrovače (Router Advertisement) či od DHCPv6 serveru (viz dále).

# DHCPv6



- DHCPv6 = **Dynamic Host Configuration Protocol for IPv6**.
- Alternativa k **bezestavové konfiguraci** (BK).
- V principu shodné s DHCP fungující pro konfiguraci IPv4, nicméně rozlišné z hlediska předaných informací - **DHCPv6 přiděluje pouze adresu** (ostatní zařídí metoda ND).
- Prefix adresy na rozdíl od BK, která striktně vyžaduje prefix=/64, může být libovolný např. /120.

Algoritmus konfigurace IPv6 adresy přes DHCPv6:

- 1. **Stanice** zasílá zprávu **DHCP discover** na objevení všech dostupných DHCP serverů (typicky jeden server, ale může jich být více).
- 2. Zprávu **DHCP offer** zasílá libovolný **DHCP server**, který od stanice obdržel zprávu DHCP discover. Součástí zprávy DHCP offer je konkrétní **konfigurační nastavení nabídnuté konkrétní stanici**.
- 3. Stanice si z došlých zpráv DHCP offer zvolí jeden DHCP server a prostřednictvím zaslání zprávy **DHCP request** požádá o rezervaci parametrů pro sebe.
- 4. Tento **DHCP server** odpoví stanici prostřednictvím **DHCP acknowledge** a zaručuje, že žádná další stanice v síti neobdrží od něj totožnou adresu.

# Přechodový mechanismus IPv6 pro přístup do IPv4

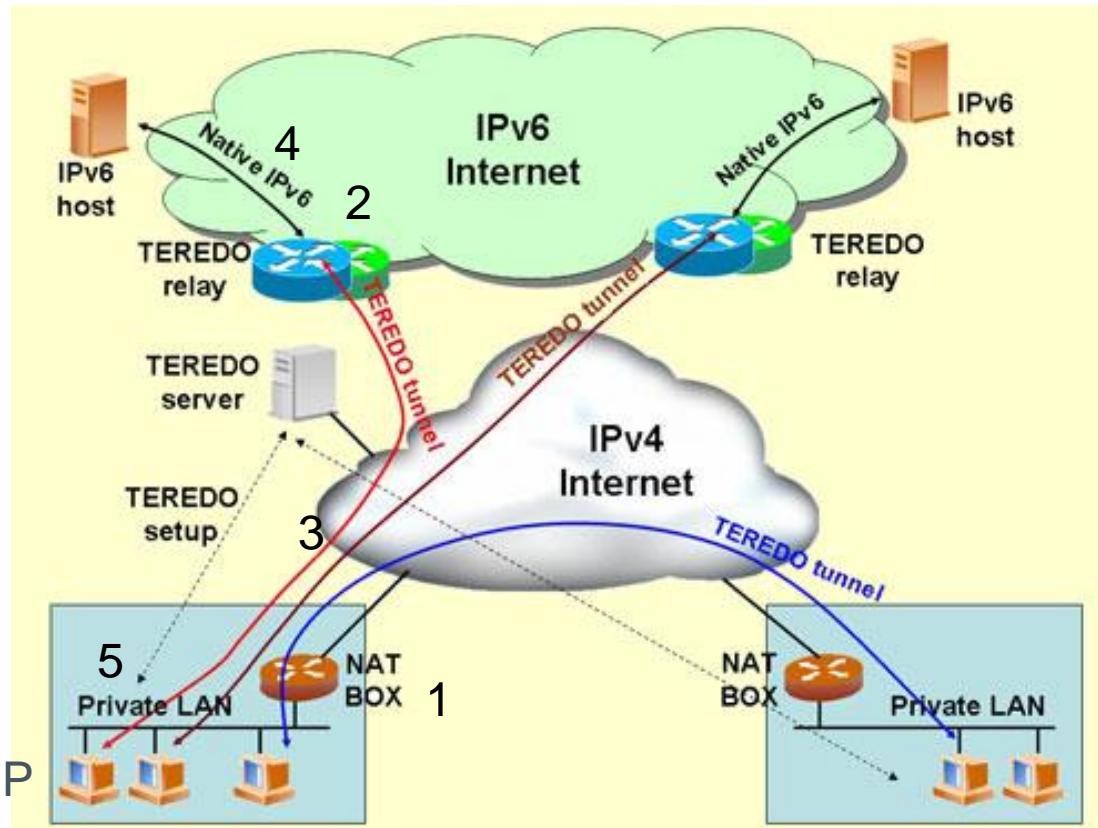


- Cílem je umožnit zařízení ze sítě s IPv6 dosáhnout zařízení v síti IPv4.
- Řešení se nazývá NAT64 (původní název NAT-PT).
- Podmínkou je, že v síti musí existovat směrovač, na kterém běží souběžně IPv4 i IPv6 protokol, který vstup zajistí.
- Pro mapování se používá prefix **64:FF9B::/96** a nebo správcem definovaný prefix.
- Obecně tedy se přechodová IPv6 adresa (IPP) pro dosažení IPv4 adresy sestaví jako  
$$\text{IPP} = \textcolor{red}{64:FF9B::/96} + 32 \text{ bitů IPv4 adresy}$$
- Tedy např. pro adresu 188.120.196.130 se vytvoří mapovaná adresa  
$$64:FF9B::\textcolor{red}{188}\textcolor{blue}{.120}\textcolor{orange}{.196}.130$$
, ve které se z původní adresy použije všech 32 bitů a tyto se stanou posledními bity adresy IPv6 → 64:FF9B:: **BC:78:C4:82**.

# Technologie TEREDO



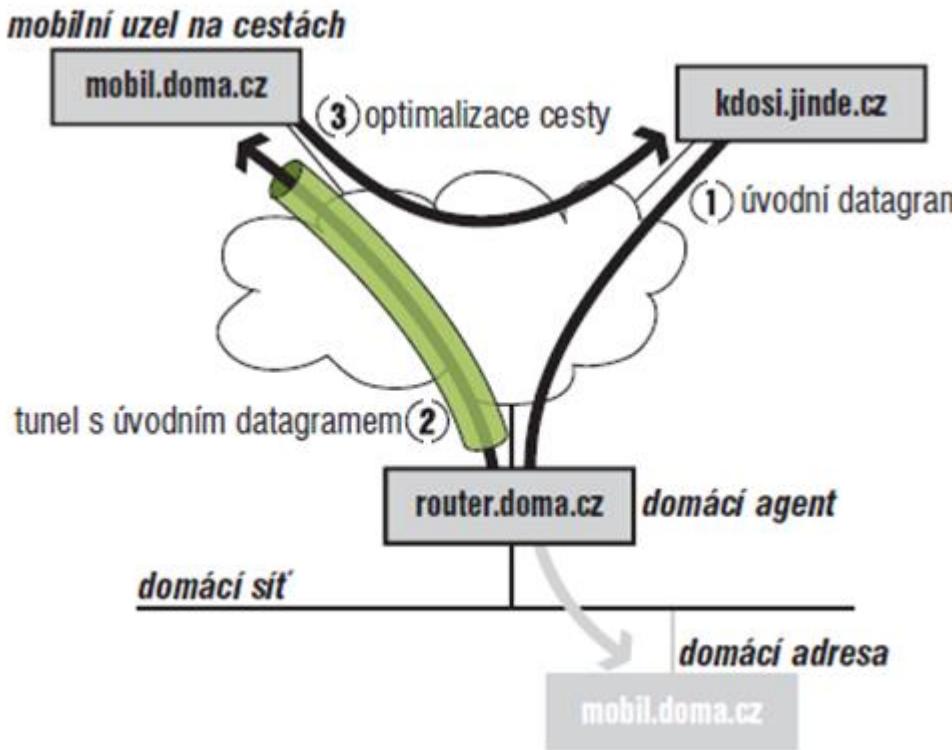
Technologie umožňuje komunikovat stanici přes IPv6, ačkoli se tato přímo nenachází v síti podporující IPv6.



1. Počítač **P** kontaktuje s využitím IPv4 TEREDO server, který mu sdělí adresu místa pro vytvoření tunelu (TR = TEREDO relay).
2. **Mezi P a TR se vytvoří spojení** přes IPv4 TEREDO tunel.
3. Veškeré pakety **IPv6 od P** se přenesou prostřednictvím tunelu k TR.
4. **TR zahájí směrování IPv6 paketů od P k cíli (IPv6 host)** prostřednictvím své připojené (native) IPv6 sítě.
5. **Jakmile přijde odpověď** od cíle k TR, je tato dále přeposlána k P opět prostřednictvím vytvořeného tunelu.

Zdroj: [https://www.researchgate.net/figure/Teredo-and-6to4-principles-x-Teredo-start-setup-y-Teredo-traffic-transiting-over-IPv4\\_fig12\\_271549193](https://www.researchgate.net/figure/Teredo-and-6to4-principles-x-Teredo-start-setup-y-Teredo-traffic-transiting-over-IPv4_fig12_271549193)

# Mobilita u IPv6



Zdroj: Pavel Satrapa, IPv6: Internetový protokol verze 6, CZ.NIC, 2019

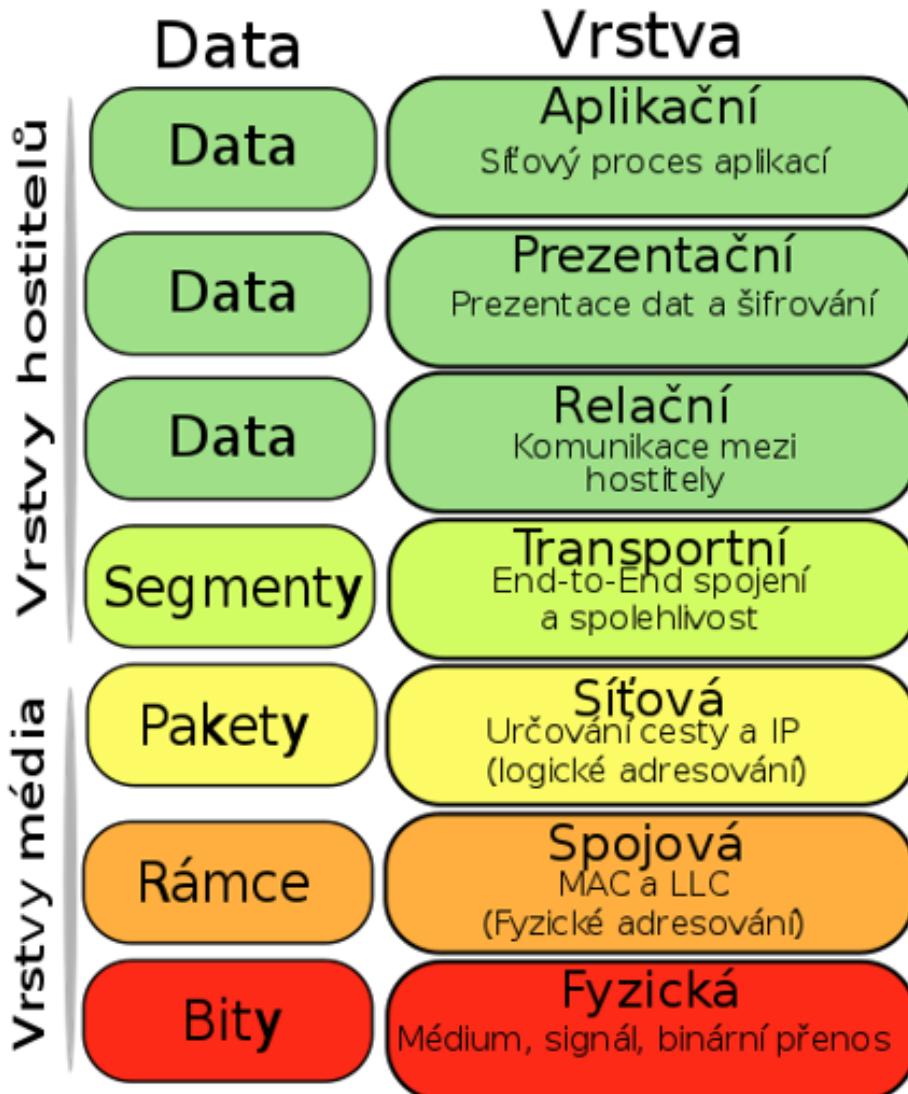
- A. Stanice (MD = mobil.doma.cz) má přidělenu svoji IPv6, kterou používá v domácí síti.
- B. Adresa (MD) je směrována přes místní směrovač (RD = router.doma.cz).
- C. MD se připojí v nějaké cizí síti pod jinou IPv6 adresou jako "mobilní uzel na cestách".
- D. Stanice kontaktuje svůj místní směrovač RD a ohlásí mu svoji aktuální adresu v cizí síti.
- E. Jakmile jiná stanice (KJ = kdosi.jinde.cz) chce kontaktovat MD (1), zařídí RD přeposlání požadavku na adresu, kterou mu MD z cizí sítě ohlásila (2).
- F. Poté, co MD obdrží žádost od KJ v cizí síti, navrhne MD KJ optimalizaci cesty (3), tj. posílání paketů přímo na IPv6 adresu v cizí síti.

# Počítačové sítě

5. přednáška - Směrování v počítačových sítích



# Důvody pro směrování, základní pojmy



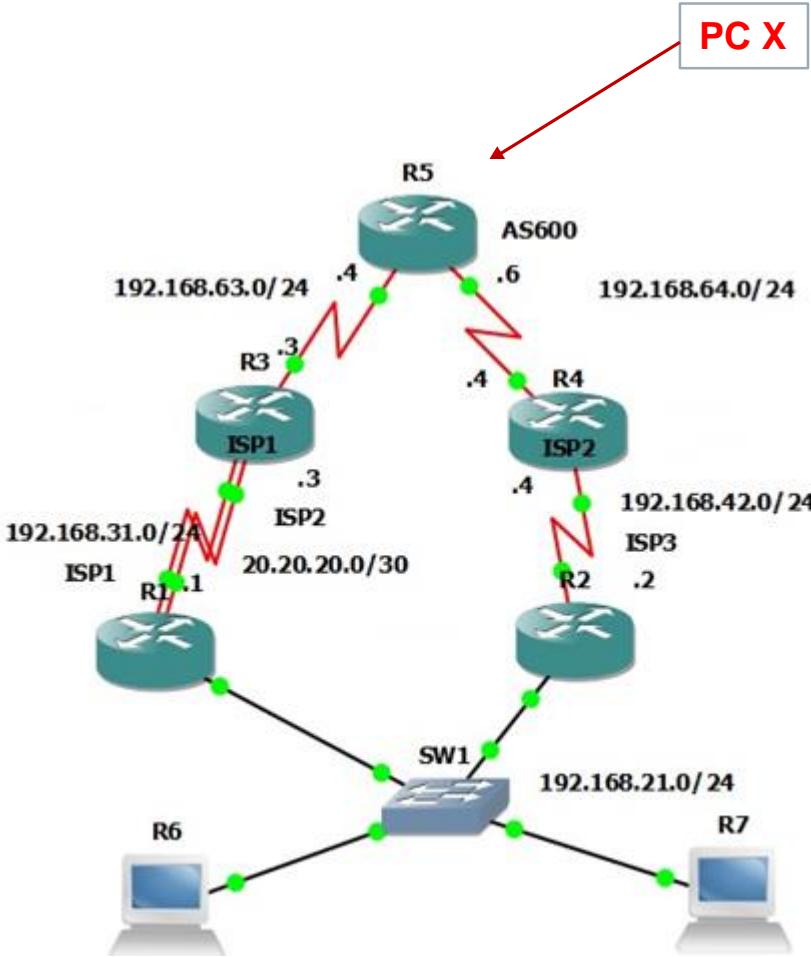
- Přepošílat data v lokální síti je snadné, vše zařídí přepínače.
- Mezi různými sítěmi oddělenými směrovači nelze data posílat přímo, musí se mezi nimi najít cesta.
- **Cesta** = posloupnost směrovačů, kterou musí projít paket mezi zdrojovou a cílovou stanicí.
- **Směrování** = hledání cesty mezi konkrétní zdrojovou a cílovou stanicí.
- Informace, na základě kterých probíhá směrování, jsou uloženy ve **směrovacích tabulkách**.
- Pokud se směrovací tabulky na směrovačích musí nastavovat manuálně, jedná se o **statické směrování**.
- Konfigurují-li směrovací tabulky autonomně směrovací protokoly během provozu, jedná se o **dynamické směrování**.

# Druhy směrování



- **Optimální** (z hlediska nějakého kritéria = metriky)
  - **nejkratší** (shortest path), typicky z hlediska počtu mezilehlých směrovačů.
  - nejlevnější (různé trasy stojí finančně různě, jsou-li na trase zpoplatněné celky).
- **Redundantní** (tzv. multi-path)
  - Provoz může být veden různými cestami. Hodí se pro zálohování cest popř. rozdělování zátěže.
- **Dle symetrie**
  - **symetrické** (dopředná i zpětná cesta jsou stejné).
  - **asymetrické** (dopředná i zpětná cesta nejsou stejné, např. satelitní systémy).
- **Dle způsobu hledání cesty**
  - **záplavové** (nejjjednodušší, ale používá je řada technologií a protokolů).
  - **proaktivní** (cesta je předpočítána a uložena ve směrovacích tabulkách).
  - **reaktivní** (cesta se vytváří až když je potřeba) → **adhoc** či mobilní sítě (proměnlivá topologie).
- **Dle oblasti výpočtu trasy**
  - **Vnitřní** (v rámci jedné sítě).
  - **Vnější** (mezi různými sítěmi).

# Redundantní (multipath) směrování

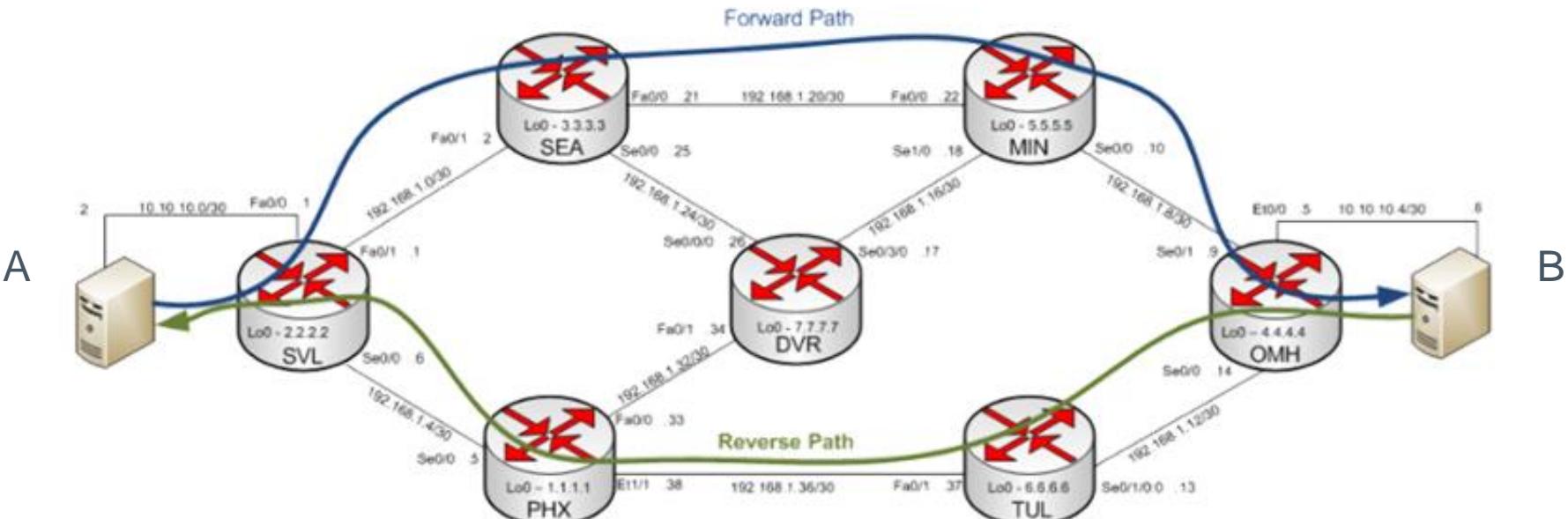


- **Redundantní směrování** = směrování k cílové stanici je možné prostřednictvím více navzájem disjunktních cest. Směrovač si zvolí jednu cestu na základě určitého kritéria(např. nejkratší vzdálenost).
- Redundantní směrování se běžně používá v případě, že je nutné garantovat **vysokou dostupnost** cílové stanice (resp. sítě, ve které se stanice nachází).
- **ISP** = Internet Service Provider, poskytovatel připojení s přidělenými specifickými IP rozsahy.

**Příklad:**

Od **PC X** k **R6** existuje cesta přes **R3** či **R4**, o výběr cesty se stará **R5**.

# Ukázka asymetrického směrování



Zdroj: <https://www.lumen.com/help/en-us/network/traceroute/ip-routing-symmetry-asymmetry.html>

Dopředná cesta (Forward Path) = cesta od A k B.

Zpětná cesta (Reverse Path) = cesta od B k A.

**Dopředná (forward) i zpětná (reverse) cesta nejsou stejné.** Na obrázku je vidět, že zdrojová stanice A používá pro dosažení cílové stanice B odlišnou cestu než stanice B pro dosažení A. Asymetrické směrování může zásadně ovlivňovat dobu doručení paketu a mít za následek nestejnou propustnost sítě v obou směrech.

# Záplavové směrování

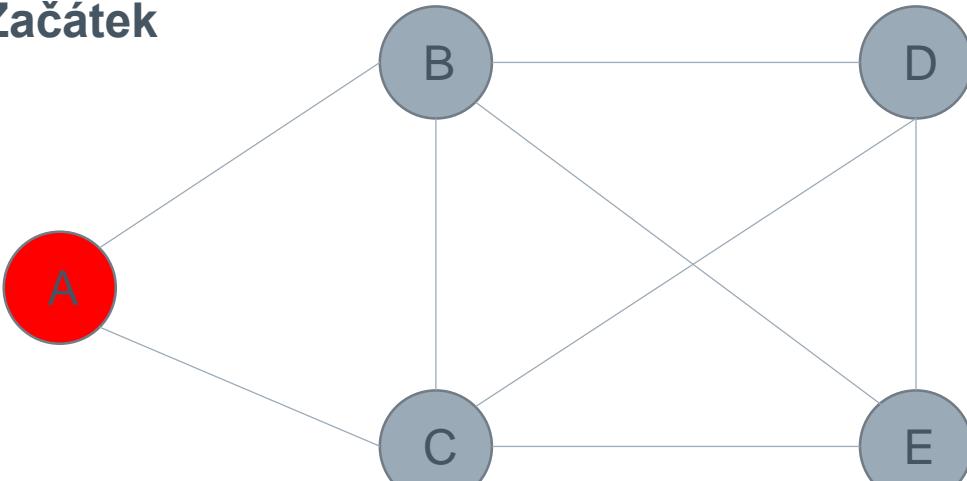


- Záplavové směrování se provádí za účelem kostrukce **kostry grafu sítě (Spanning Tree)**, viz BI-AG1 – 3. přednáška.
- Kostra grafu sítě je důležitá proto, že prostřednictvím ní je možné acyklicky doručit data od kořenového uzlu ke všem ostatním uzlům.
- Algoritmus záplavového směrování:
  - Záplavové směrování **začíná ve zvoleném počátečním uzlu**.
  - Počáteční uzel osloví všechny svoje sousední uzly s požadavkem na směrování. Požadavek je opatřen identifikátorem (ID), který má za úkol zabránit vytváření smyček během směrování. Důvodem použití ID je potřeba odlišit to, že záplava může být spuštěna z různých uzlů najednou.
  - Každý uzel, který obdržel požadavek na směrování nejprve zkontroluje, zda již dříve obdržel požadavek s tímto ID. Pokud ano, požadavek zahodí. V opačném případě provede přeposlání požadavku všem sousedům s výjimkou toho, od kterého požadavek přišel.
  - Přeposílání sousedům se opakuje až do doby, kdy jsou dosaženy všechny uzly grafu.
  - V teorii grafů se záplavový algoritmus nazývá **prohledávání do šířky (BFS)**.

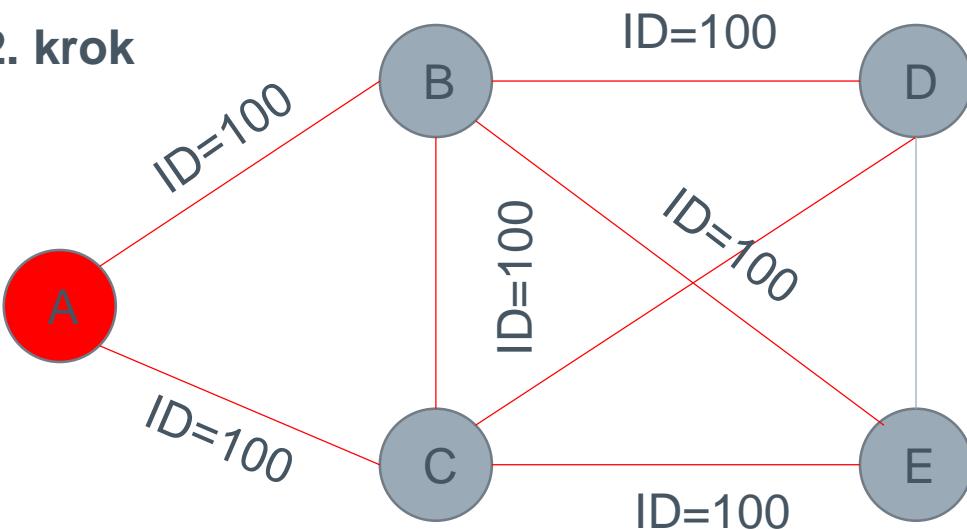
# Záplavové směrování – příklad



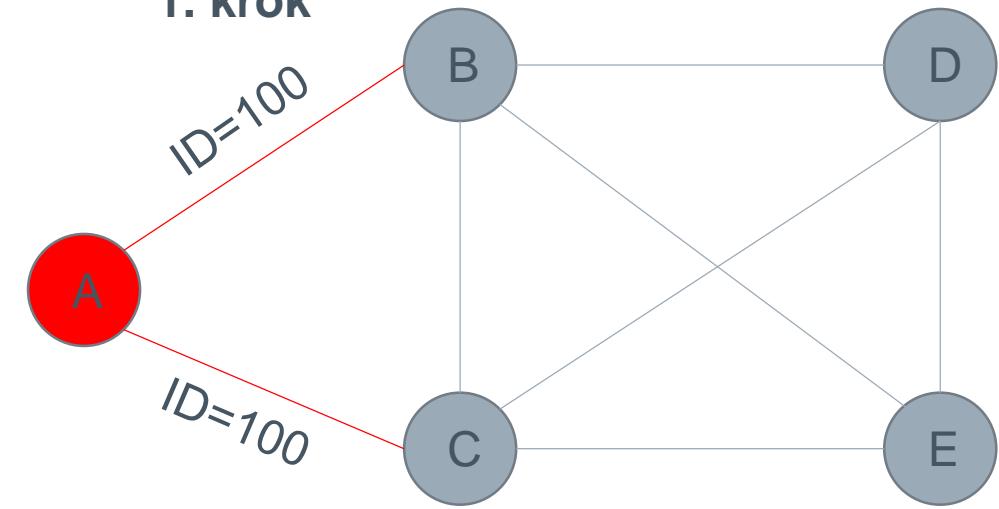
Začátek



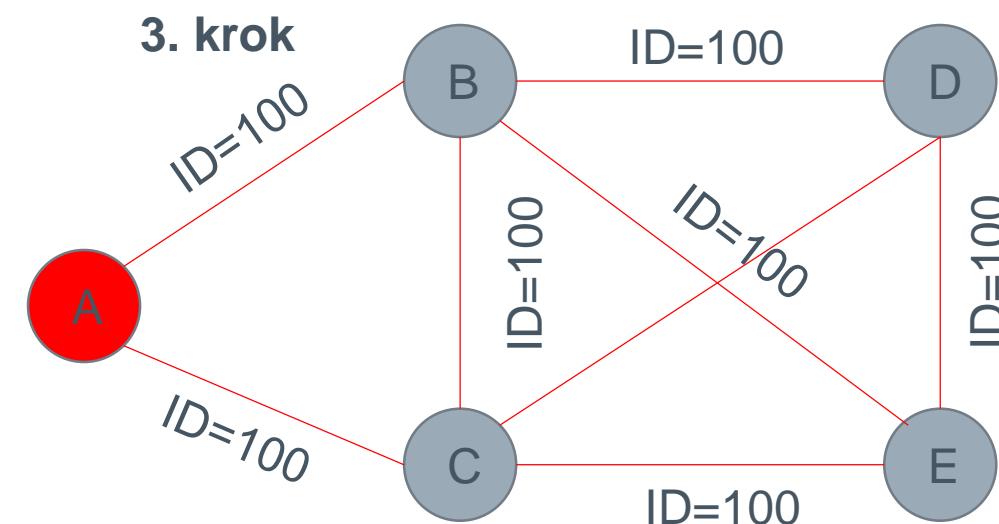
2. krok



1. krok

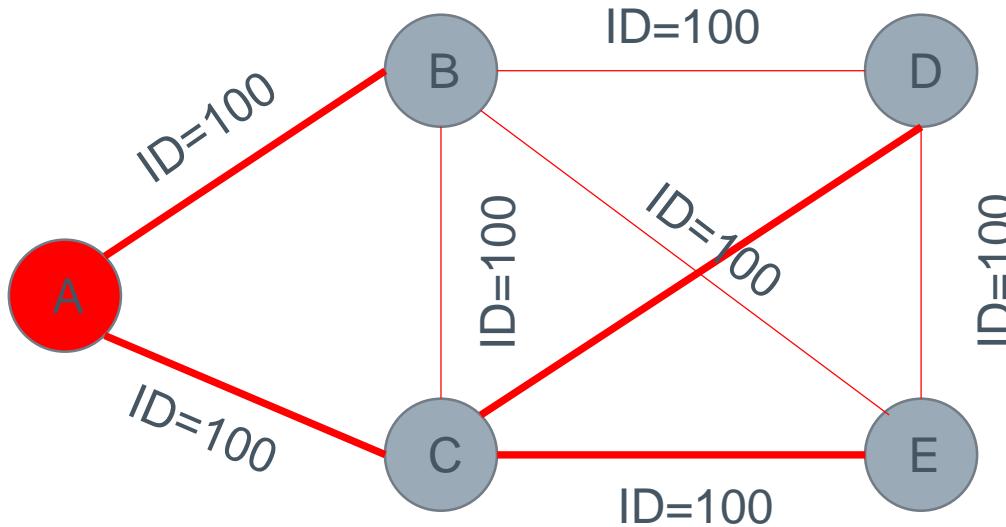


3. krok



Ve třetím kroku jsou všechny přepošlané požadavky již sousedními uzly zamítnuty, jelikož již dříve všechny uzly požadavek i ID=100 obdržely.

# Záplavové směrování – výsledná kostra grafu

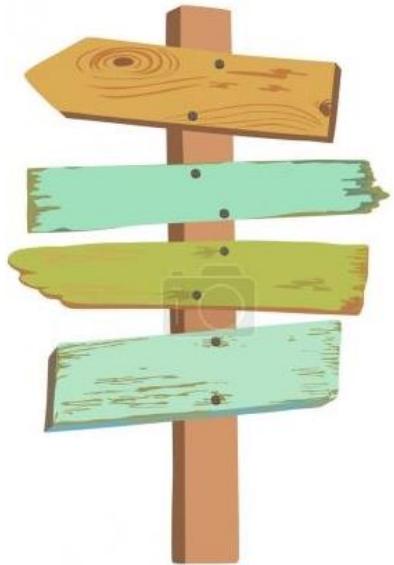


Výsledná kostra grafu je pouze **jedním z možných řešení**. Kritériem pro výběr hrany finální kostry byla v tomto případě nejkratší doba šíření záplavy k příslušnému uzlu a nikoli vzdálenost mezi uzly. Kostra grafu je základem řady algoritmů, které se využívají v počítačových sítích či distribuovaných systémech.

# Směrovací tabulka (pro zopakování)



- Tabulka funguje jako rozcestník, který umožňuje zjistit směr, kterým má směrovač přijaté pakety dále odeslat.
- Typické položky záznamu ve směrovací tabulce
  - **Destinace**: IP (analogie cílové město).
  - **Brána**: IP (analogie další rozcestník na trase).
  - **Maska**: dle prefixu (analogie oblast).
  - **Metrika**: cena (analogie vzdálenost v km).
  - **Rozhraní**: název (analogie vstup do silnice).



## Příklad reálné směrovací tabulky:

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
188.120.196.128	0.0.0.0	255.255.255.248	U	0	0	0	eth1.101
10.0.0.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.5.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.0.0.0	10.5.1.1	255.0.0.0	UG	0	0	0	eth2
0.0.0.0	188.120.196.129	0.0.0.0	UG	0	0	0	eth1.101

Brána **0.0.0.0** znamená, že se jedná o lokální síť, doručuje se přes linkovou vrstvu!

# Proaktivní vs. reaktivní směrování



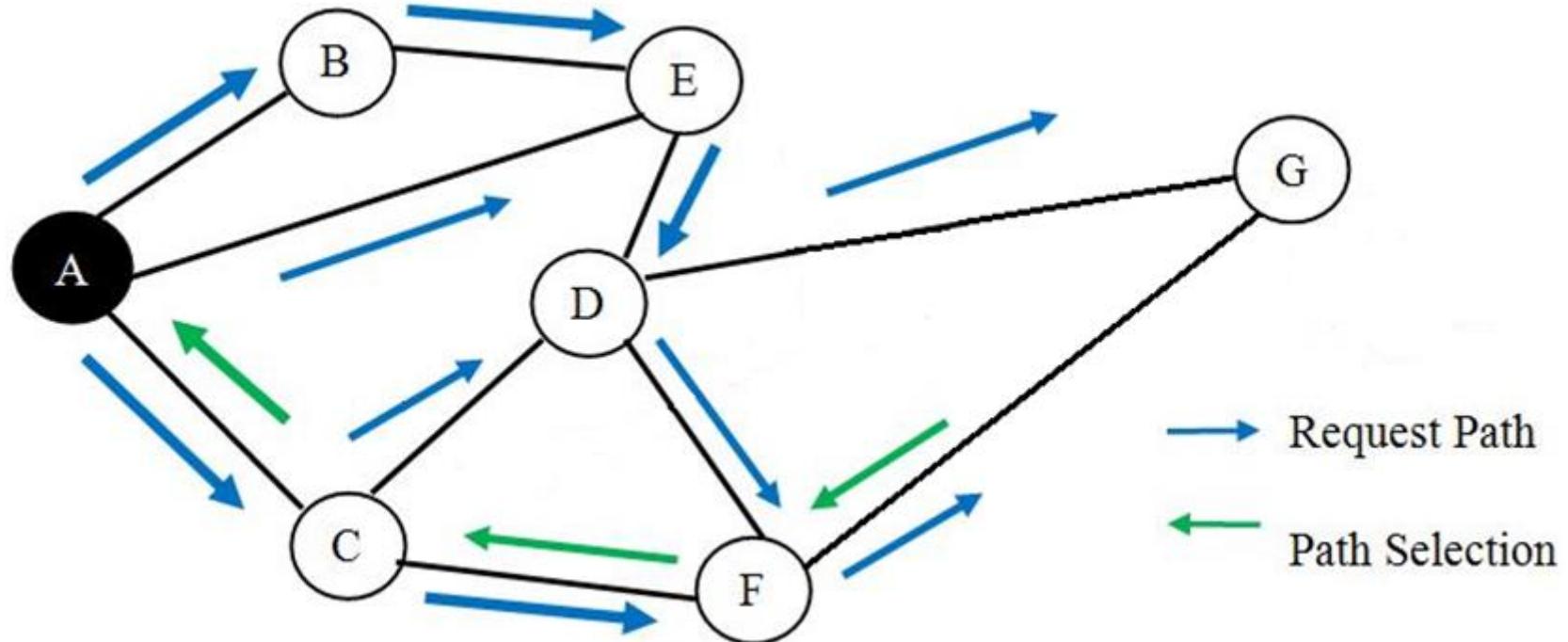
**Proaktivní směrování = směrování, které používá směrovací tabulky.**

- Proaktivní směrování se využívá v sítích, které se nemění anebo mění vždy za delší dobu.
- U proaktivního směrování před posíláním dat je cesta od zdrojové stanice k cílové již známá.
- Tento způsob **je nejběžnější** v počítačových sítích.

**Reaktivní směrování = směrování, které nepoužívá směrovací tabulky.**

- Směrovací **tabulky se nepoužívají**, jelikož jejich sestavování může být pro dané typy sítí příliš pracné či neúčelné.
- Využití toho druhu směrování má smysl např. v **mobilních sítích** popř. v **ad-hoc** sítích, protože se jejich uspořádání (topologie) často mění.
- Směrování má obvykle 2. fáze
  - 1. **nalezení cesty** mezi zdrojovou a cílovou stanicí (většinou zaplavovým směrováním).
  - 2. **posílání dat** přes nalezenou cestu a následné zapomenutí cesty.
    - Cesta může být uložena v hlavičce anebo dočasně v paměti mezilehlých směrovačů.

# Ukázka reaktivního směrování



Zdroj: [https://www.researchgate.net/figure/Example-of-reactive-routing-scheme\\_fig1\\_329510901](https://www.researchgate.net/figure/Example-of-reactive-routing-scheme_fig1_329510901).

Uzel A chce komunikovat s uzlem G, nejprve se zaplavovým směrováním **najde cesta**, následně se **posílají data** a nakonec se cesta zruší.

# Statické vs dynamické směrování



- V dalším výkladu pod pojmem směrování budeme chápát pouze **proaktivní směrování**.
- Základem tohoto směrování jsou **směrovací tabulky**, které se nacházejí **na každém směrovači**.
- Směrovací tabulky lze **konfigurovat manuálně (staticky) nebo dynamicky**. Dynamická konfigurace tabulek provádějí směrovače samy s využitím **směrovacích protokolů**.
- Směrovací protokoly mohou používat pro volbu optimální cesty různé metriky popř. jejich kombinace.
- Statické a dynamické směrování se mohou společně kombinovat a dělá se to takto běžně.

# Statická konfigurace směrovacích tabulek



- **Typické operace ve směrovací tabulce**
  - Přidání záznamu.
  - Odebrání záznamu.
  - Změna záznamu (např. sousední směrovač nebo metrika).
- **Příkazy používané v operačních systémech**
  - **route** (jak Unix tak Windows, nicméně odlišná syntaxe).
  - **ip route** (Linux, Cisco, Mikrotik).
- **Vlastnosti statického směrování**
  - Funguje okamžitě po zapnutí směrovače.
  - Není schopné reagovat na změny.
  - Vyžaduje manuální konfiguraci všech prvků → **pracné pro velké sítě**.
  - Hodí se hlavně pro menší sítě s jednoduchou strukturou.

# Dynamická konfigurace směrovacích tabulek



- Tabulky si nastavují směrovače **automaticky s využitím směrovacích protokolů**.
- Směrovače si informace týkající se směrování periodicky předávají zasíláním zpráv.
- Díky periodickému zasílání zpráv směrovače dokáží rychle reagovat na změny v síti (např. **připojení nových směrovačů, výpadky linek atd.**).
- Dynamické konfigurace směrovacích tabulek ve velkých sítích mohou trvat i desítky vteřin.
- Rozlišujeme 3 druhy dynamických směrovacích algoritmů (A), které jsou pojmenovány dle obsahu zasílaných zpráv a sice :
  - **Distance Vector Algoritmy (DVA)**
  - **Link State Algoritmy (LSA)**
  - **Path Vector Algoritmy (PVA)**

# Distance Vector Algoritmy (DVA)



- Směrovače si vyměňují navzájem záznamy svých směrovacích tabulek (vektory hodnot vzdáleností) prostřednictvím zpráv (M). Metrika je vzdálenost.
- Pro optimalizaci záznamů používají **Distribuovanou verzi Bellman-Fordova Algoritmu**(DBFA).
- DBFA není totožná s klasickou – viz. předmět BI-AG1, 12. přednáška.
- Základní optimalizační krok u DBFA , je relaxace jako u klasického BFA a sice:
  - Každý směrovač ( $X$ ) zná vzdálenost (metriku) ke všem svým sousedním směrovačům  $\{Y_i\}$ , kde  $1 \leq i \leq M$ , přičemž  $M$  je počet sousedních směrovačů daného  $X$ .
  - Směrovač  $X$  má ve své směrovací tabulce záznamy o všech směrovačích (cílech) v síti  $\{Z_j\}$ , kde  $1 \leq j \leq N$ , přičemž  $N$  je počet všech směrovačů v síti.
  - V případě, že směrovač  $X$  obdrží od souseda  $Y_i$  zprávu  $M$  týkající se  $Z_j$ , provede srovnání s aktuálním stavem ve směrovací tabulce a pokud platí, že:

$$\text{vzdálenost}(X, Z_j) > \text{vzdálenost}(X, Y_i) + \text{vzdálenost}(Y_i, Z_j)$$

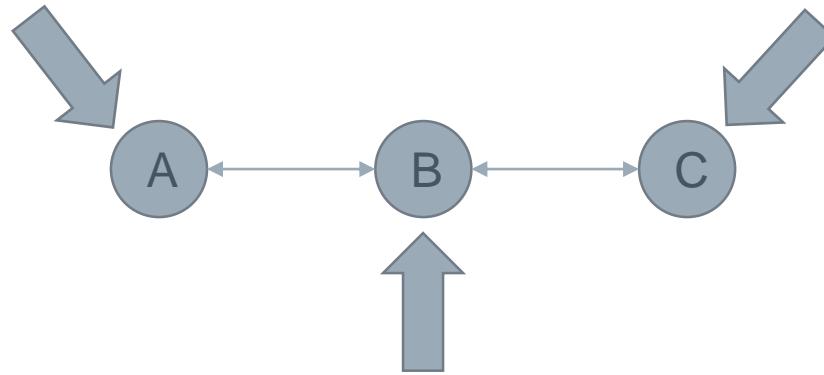
- Nahradí  $X$  ve své směrovací tabulce výchozí směrovač (bránu) pro  $Z_j$  směrovačem  $Y_i$ .
- Implementace DBFA je použita ve směrovacím protokolu, který se nazývá **RIP = Routing Information Protocol**.

# RIP – konstrukce směrovacích tabulek (Inicializace)



Cíl	Brána	Metrika
A	A	0

Cíl	Brána	Metrika
C	C	0



Cíl	Brána	Metrika
B	B	0

Metrika = vzálenost!

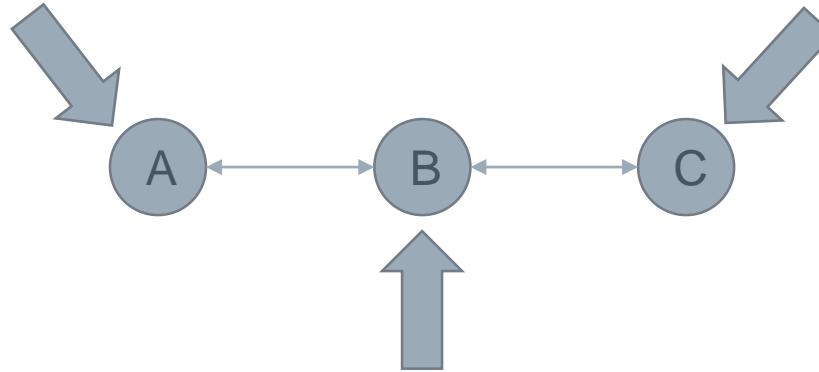
Směrovače A,B a C po zapnutí mají směrovací tabulky obsahující pouze jeden záznam týkající se jich samotných.

# RIP – konstrukce směrovacích tabulek (1. krok)



Cíl	Brána	Metrika
A	A	0
B	B	1

Cíl	Brána	Metrika
C	C	0
B	B	1



Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1

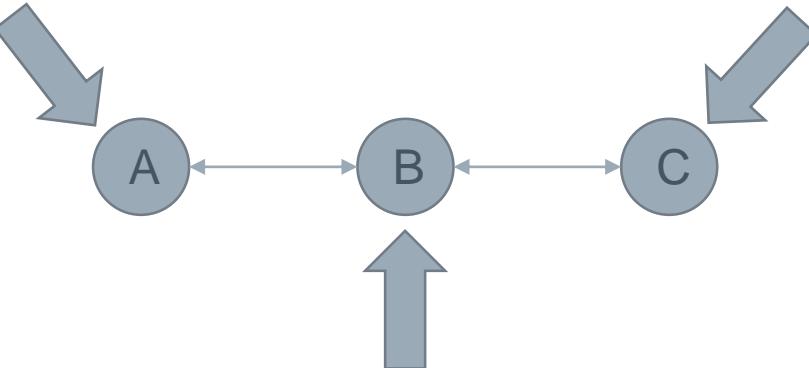
Sousední směrovače si poprvé vyměnily navzájem záznamy ve směrovacích tabulkách a provedly úpravy dle postupu popsaného na slajdu 17.

# RIP – konstrukce směrovacích tabulek (2. krok)



Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2



Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1

Sousední směrovače si opět vyměnily již aktualizované záznamy ve směrovacích tabulkách a provedly **úpravy**.

# RIP – přidání směrovače, inicializace

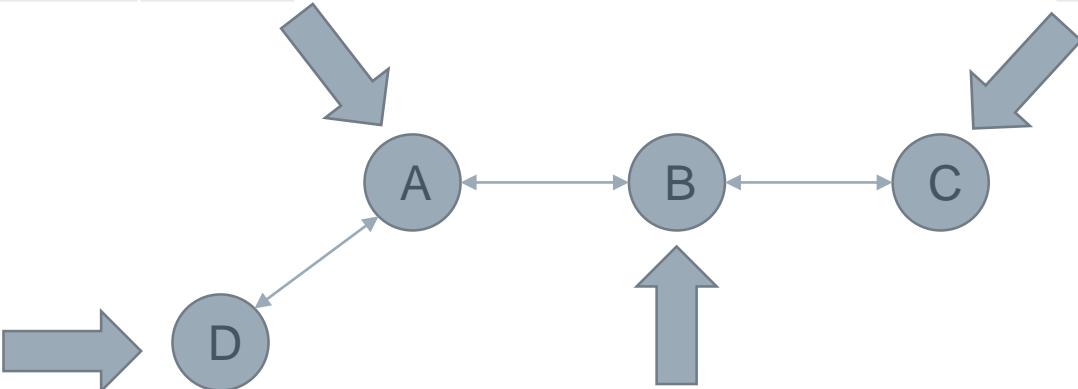


Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2

Směrovač D se chce připojit k síti a zašle sousednímu směrovači A zprávu.

Cíl	Brána	Metrika
D	D	0



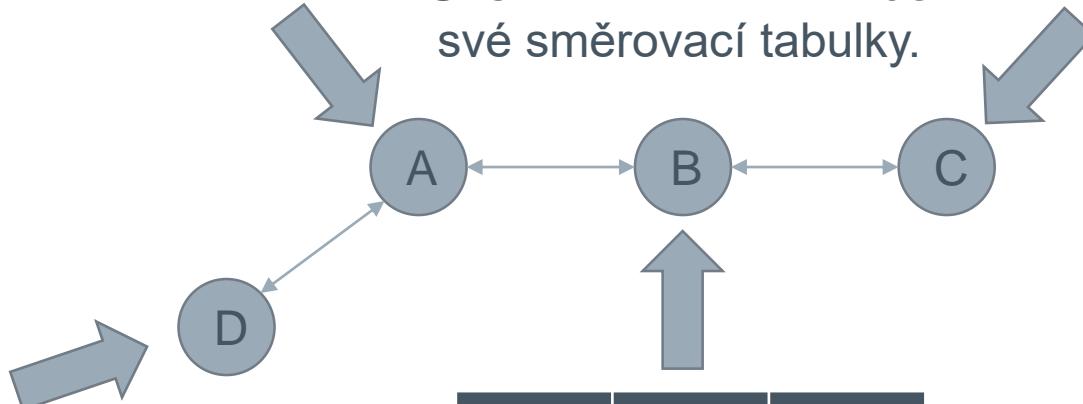
Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1

# RIP – přidání směrovače, 1. krok



Uzel D obdržel  
směrovací tabulku od  
uzlu A.

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3



Uzel A zařadil uzel D do  
své směrovací tabulky.

Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2

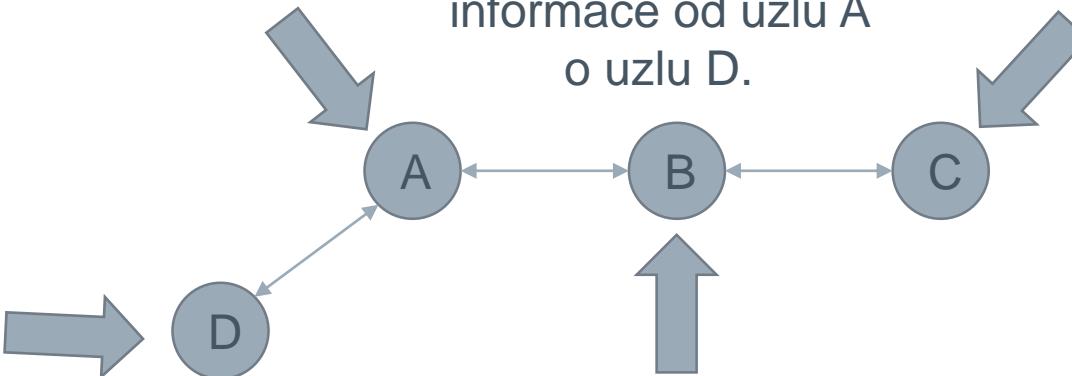
# RIP – přidání směrovače, 2. krok



Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2

Uzel B obdržel  
informace od uzlu A  
o uzlu D.



Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3

Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1
D	A	2

# RIP – přidání směrovače, 3. krok

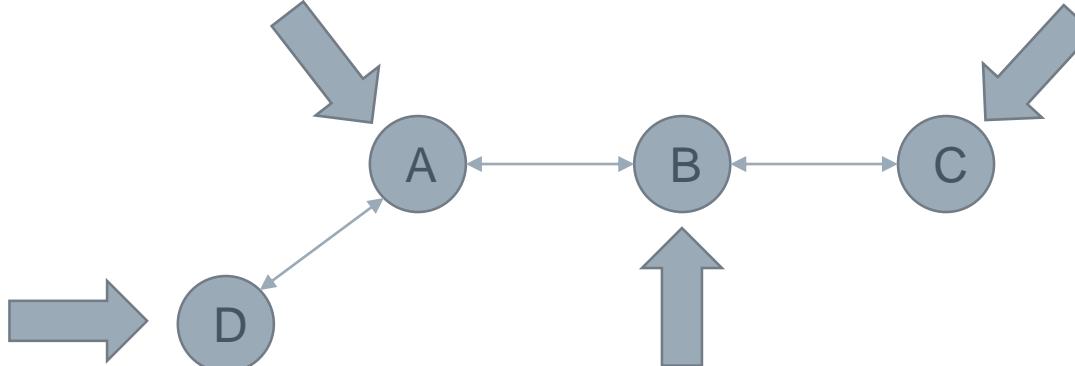


Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2
D	B	3

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3

Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1
D	A	2



Uzel C obdržel informace o směrovači D od B a přidání směrovače D je dokončeno v celé síti.

# RIP – porucha linky

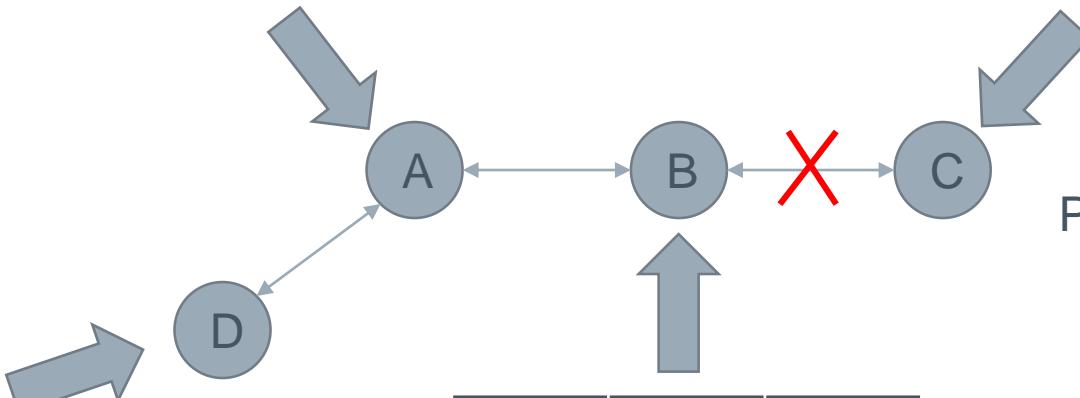


Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	1
A	B	2
D	B	3

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3

Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	1
D	A	2



Porucha nastane na  
lince mezi B a C.

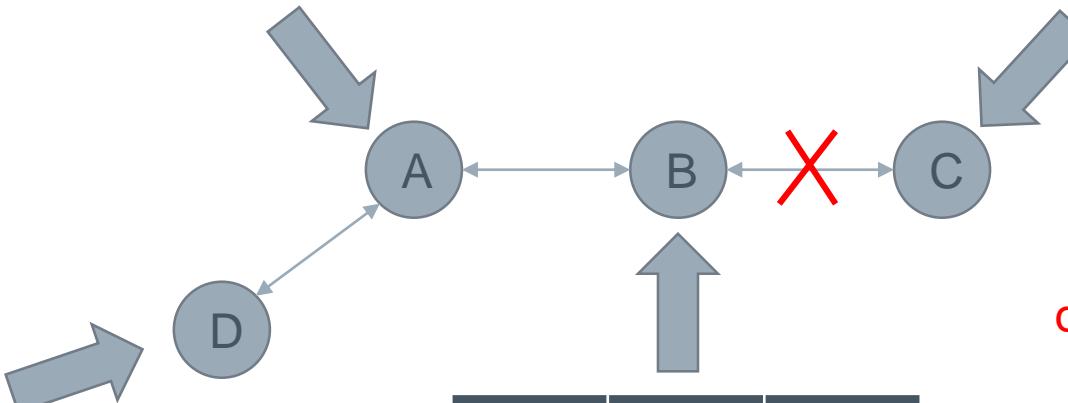
# RIP – oprava směrovacích tabulek, 1. krok



Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	2
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	$\infty$
A	B	$\infty$
D	B	$\infty$

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3



Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	$\infty$
D	A	2

Směrovače B a C detekují výpadek linky (nedostávají od sebe navzájem zprávy) a nastaví metriku pro všechny záznamy související s poruchou na  $\infty$ .

# RIP – oprava směrovacích tabulek, 2. krok

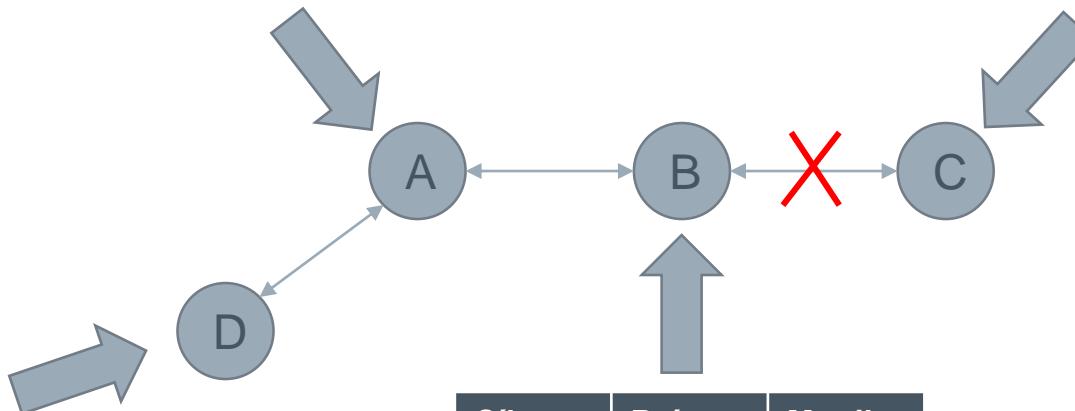


Uzel A obdrží od B informace o výpadku a nastaví metriku na  $\infty$ .

Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	$\infty$
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	$\infty$
A	B	$\infty$
D	B	$\infty$

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	3



Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	$\infty$
D	A	2

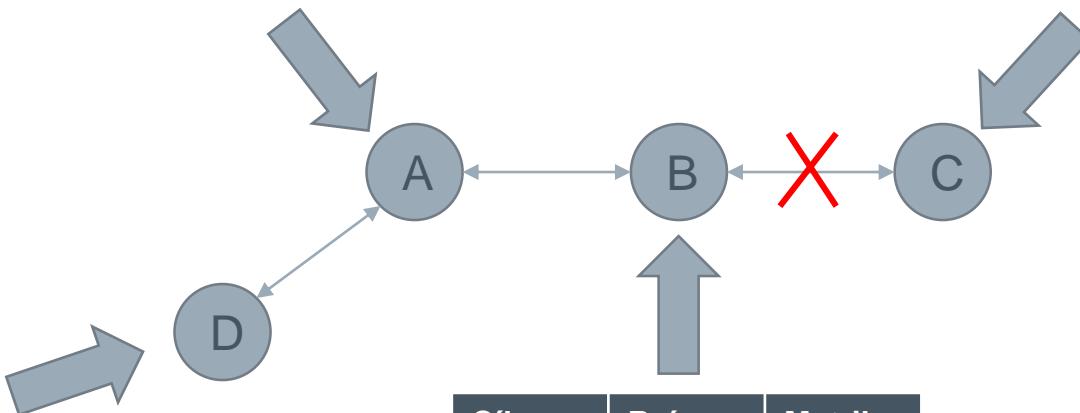
# RIP – oprava směrovacích tabulek, 3. krok



Cíl	Brána	Metrika
A	A	0
B	B	1
C	B	$\infty$
D	D	1

Cíl	Brána	Metrika
C	C	0
B	B	$\infty$
A	B	$\infty$
D	B	$\infty$

Cíl	Brána	Metrika
D	D	0
A	A	1
B	A	2
C	A	$\infty$



Uzel D obdrží od  
A informace o  
výpadku a  
nastaví metriku  
na  $\infty$ .

Cíl	Brána	Metrika
B	B	0
A	A	1
C	C	$\infty$
D	A	2

# Link-State (LS) algoritmy(A)



- Linka (L) = propojení dvou sousedních směrovačů (X a Y) s určitým ohodnocením (K).
- Spojením všech linek dohromady vznikne graf sítě (G).
- Směrovače si vyměňují navzájem stavy linek a z těchto si každý směrovač vytvoří vnitřní reprezentace ohodnoceného grafu sítě (G).
- Směrovač nad G hledá kostru, která obsahuje nejkratších cesty (KNC).
- Pro nalezení KNC se používá **Dijkstrův algoritmus (DA)** - viz předmět BI-AG1 (12. přednáška).
- Z nalezené kostry se procházením do hloubky (DFS) sestaví postupně směrovací tabulka.

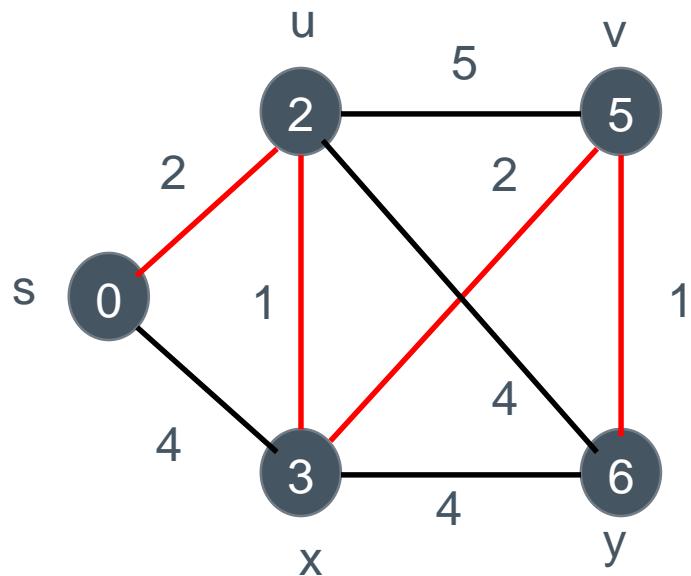
## Implementace LSA algoritmu:

- Známý směrovací protokol, který využívá ve své implementaci DA se nazývá **OSPF = Open Shortest Path First**. Tento se používá výhradně pro vnitřní směrování.
- Jiným známým LS protokolem, principiálně podobným OSPF je protokol **IS-IS** (Intermediate System to Intermediate System). IS-IS protokol se používá pro vnitřní i vnější směrování.
- LSA jsou v porovnání s DVA méně náročné na zatížení linky výměnou dat. U LSA se zasílají jen krátké zprávy se stavu linek, naopak u DVA se posílají delší zprávy s jednotlivými položkami řádků směrovacích tabulek.
- LSA jsou oproti DVA naopak náročnější na využití CPU, jelikož každý CPU řeší jednu instanci DA. DA se spouští při změně stavu libovolné linky a prohledává opět celý graf.

# Příklad: OSPF – konstrukce směrovací tabulky

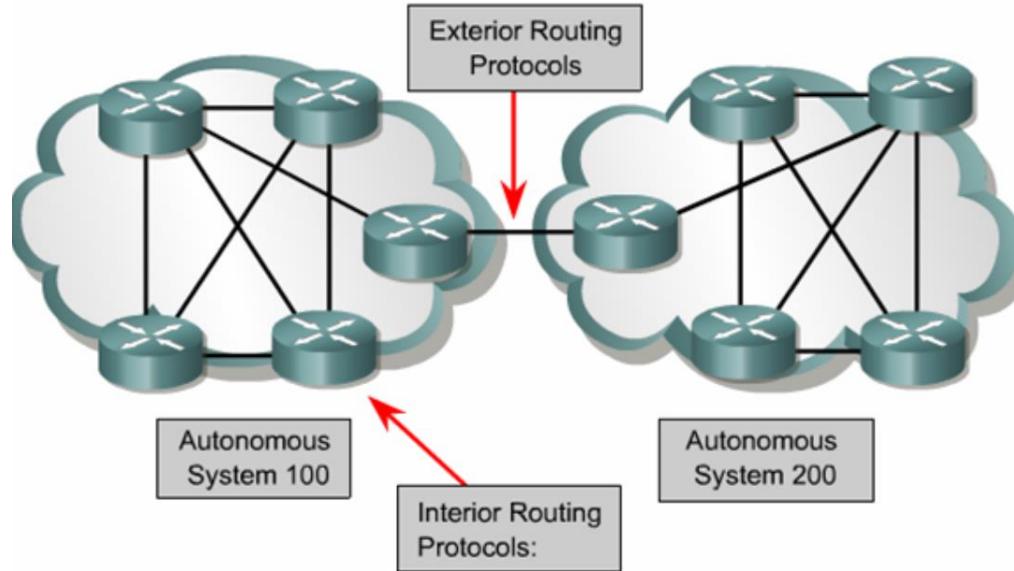


1. Směrovač **s** získal od sousedů množinu stavů linek (L) a sestavil si z ní graf (G).
2. Pomocí DA (startovní uzel **s**) vytvořena **kostra grafu**. Hodnota v šedém kolečku značí celkovou nejmenší vzdálenost uzlu od **s**.
3. Pro vytvořenou kostru byla procházením do hloubky sestavena postupně směrovací tabulka tak, že cíl určuje aktuálně procházený uzel (U), brána označuje uzel, který byl navštíven jako první po startovacím uzlu **s** a metrika je vzdálenost U od **s**.



Cíl	Brána	Metrika
u	u	2
x	u	3
v	u	5
y	u	6

# Vnitřní (Interior) vs vnější (Exterior) směrování



Zdroj: <http://msrblog.com/assign/science/eee/exterior-routing-protocols-multicasting.html>

**Autonomní Systém (AS)** = Skupina všech IP adresních rozsahů konkrétního ISP. Každý AS má svůj specifický identifikátor. Identifikátory se využívají u **vnějšího směrování (viz dále)**.

**Vnitřní směrování** = směrování uvnitř sítě patřící jednomu ISP. Pro toto směrování se používají vnitřní směrovací protokoly (např. OSPF, RIP).

**Vnější směrování** = směrování mezi sítěmi různých ISP. Pro toto směrování se používají vnější směrovací protokoly (např. BGP, viz dále).

# Path Vector Algoritmy (PVA)

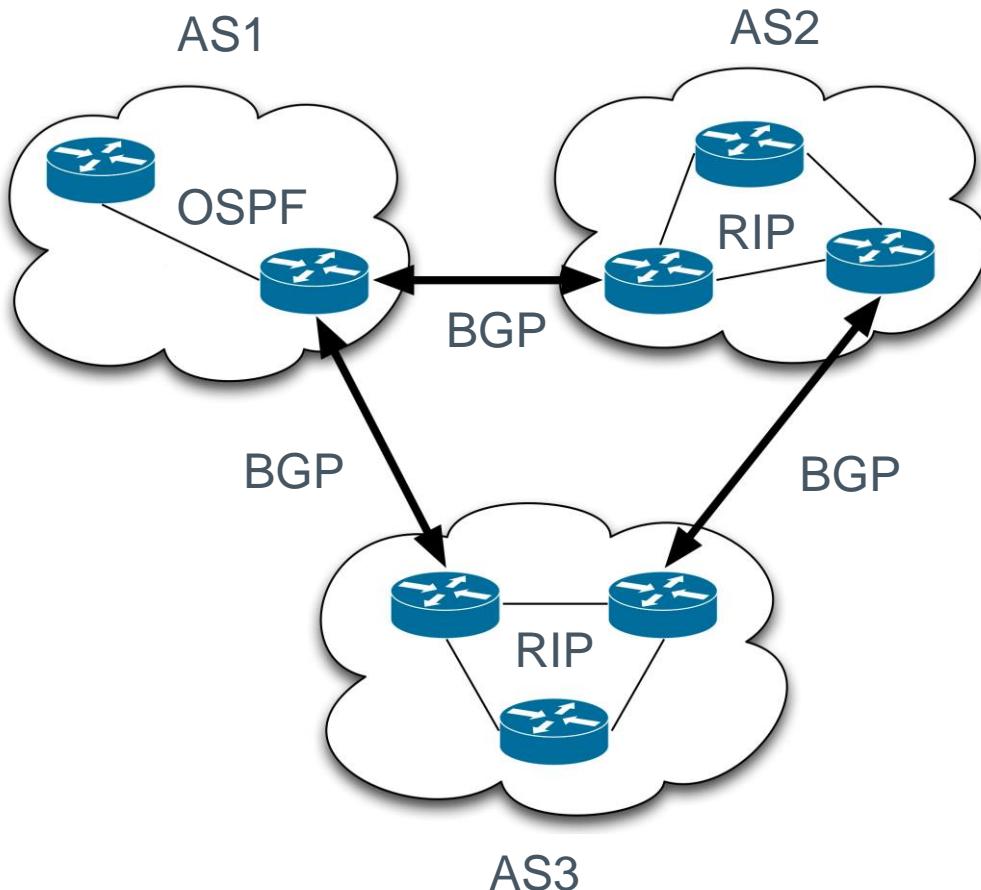


- Směrovače si vyměňují kompletní cesty ke konkrétním cílům.
- Cesta = posloupnost směrovačů od zdroje k cíli.

## Implementace PVA:

- Typickým představitelem dynamických směrovacích protokolů využívající PVA je **BGP** = **Border Gateway Protocol**.
- BGP však namísto směrovače předpokládá **autonomní systém (AS)**.
- Cesta u BGP = posloupnost směrovačů jednotlivých AS od zdroje k cíli.
- Každý AS v BGP má svůj identifikátor (ID) - 16ti bitové číslo, které je uvedeno v příslušné cestě.
- BGP se používá primárně pro vnější směrování.
- **Metrika konkrétní cesty u BGP** se určuje jako VH = **vektor hodnot**. Počet prvků VH záleží na konkrétním výrobci směrovače, typicky < 10. Hodnoty prvků VH se počítají navzájem různými způsoby a mají různé váhy při směrovacím rozhodnutí. Konkrétní význam všech hodnot je nad rámec tohoto předmětu a budeme proto dále uvažovat pouze jedinou - **celkovou délku cesty**.

# Vnější a vnitřní směrování aneb jak funguje Internet



Zdroj: <https://www.noction.com/blog/bgp-multi-homing-enough-for-network-performance>

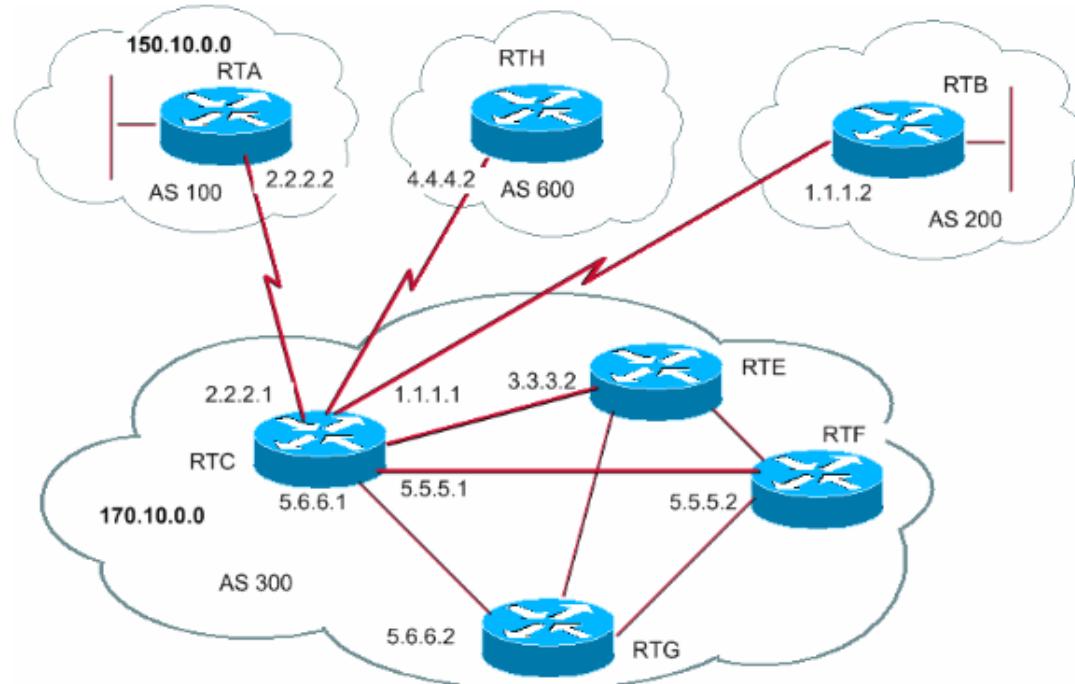
- AS se navzájem propojují prostřednictvím **hraničních směrovačů**.
- AS, přijímací provoz pouze pro své **vnitřní** stanice, se označuje jako **netransientní**.
- AS, přeposílající provoz určený stanici náležící jinému AS prostřednictvím své vnitřní sítě, se nazývá **transientní**.
- Hraniční směrovače** používají BGP protokol pro dynamickou konfiguraci směrovacích tabulek pro **adresní rozsahy náležící jiným AS**.
- Mezi sousedními hraničními směrovači jsou data doručována pomocí linkové vrstvy.
- Pro dynamickou konfiguraci směrovacích tabulek **uvnitř konkrétního AS**, používají směrovače vnitřní směrovací protokoly (RIP, OSPF) pouze pro **adresní rozsahy náležící danému AS**.

Internet = množina vzájemně propojených autonomních systémů.

# BGP a jeho spolupráce IP protokolem



Zdroj: <https://www.noction.com/blog/bgp-multi-homing-enough-for-network-performance>



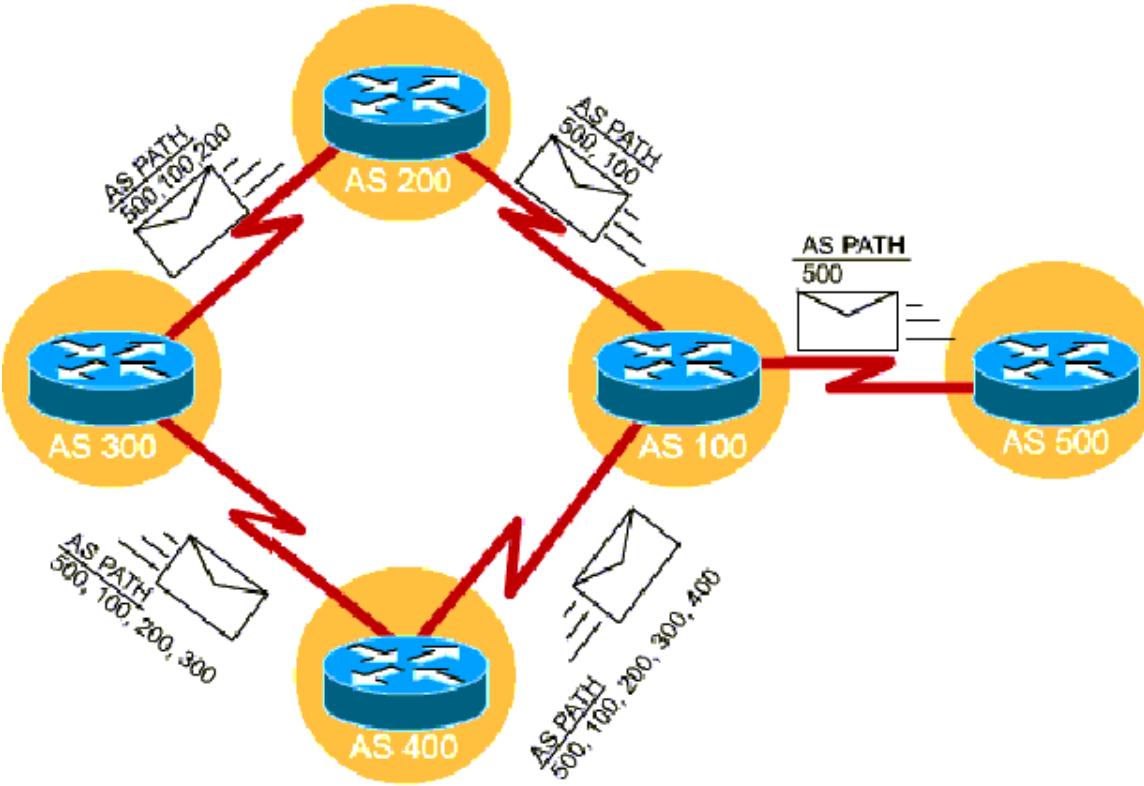
- Každému AS (100,600,200,300) náleží konkrétní síťové rozsahy pro jeho místní síť.
- Směrování BGP se uplatňuje vždy až na hraničním (edge) směrovači.
- **Hraniční směrovač** např. RTC spojuje AS 300, 100, 600 a 200).
- Pro **každou cílovou IP adresu** v hlavičce IP paketu, lze jednoznačně určit její příslušnost ke konkrétnímu cílovému AS.
- Např. tedy pro paket jdoucí z adresy 5.5.5.2 (AS 300) na cílovou adresu 150.10.0.0 proběhne na RTC jeho **namapování** na cílový AS 100.
- **NA RTC se zvolí optimální cesta** (např. dle celkového počtu směrovačů) a odešle se na hraniční směrovač sousedního AS, kterým zvolená cesta začíná.

**Poznámka.** Namapování IP adresy na konkrétní AS je v Internetu vázáno na databázi organizace, která IP rozsahy pro určitý kontinent (pro Evropu RIPE) spravuje.

# Postupné vytváření cesty u BGP



Zdroj: <http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html>



1. Směrovač (R) z konkrétního AS odešle zprávu obsahující jeho ID směrovači sousedního AS (S).
2. S si uloží cestu k R a přepoše dalšímu směrovači zprávu obsahující značky R,S (cestu) dalšímu směrovači z jiného AS.
3. Kroky 1 a 2 se neustále opakují, směrovače vždy dostávají zprávu o 1 ID delší než směrovač, od kterého zpráva přišla.
4. Pokud se stane, že zpráva dorazí směrovači, jehož ID již v cestě obsaženo je, detekuje se smyčka a dál se cesta nepřeposílá.

Směrovací tabulky BGP směrovače typicky obsahují pro různé cílové AS více cest a z těchto se volí ta nevhodnější.

# Ukázková směrovací tabulka BGP



Network	Next Hop	MED	LocPrf	Weight	AS_PATH
12.42.72.190/32	10.1.1.235	0	100	100	14207 3944 7777 i
12.43.144.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 16711 i
12.65.240.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.66.0.0/19	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.66.32.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.79.224.0/19	10.1.1.235	0	100	100	14207 3944 2914 7018 5074 i
13.13.0.0/17	10.1.1.235	0	100	100	14207 3944 2914 7018 22390 i
13.13.128.0/17	10.1.1.235	0	100	100	14207 3944 2914 4323 22390 i
13.16.0.0/16	10.1.1.235	0	100	100	14207 3944 2914 5511 5388 i
15.0.0.0/8	10.1.1.235	0	100	100	14207 3944 2914 209 71 i
15.130.192.0/20	10.1.1.235	0	100	100	14207 3944 2914 5400 1889 i
15.142.48.0/20	10.1.1.235	0	100	100	14207 3944 2914 3561 5551 1889 i
15.166.0.0/16	10.1.1.235	0	100	100	14207 3944 2914 209 71 i
15.195.176.0/20	10.1.1.235	0	100	100	14207 3944 2914 3561 1273 1889 i

Network = cílová síť.

AS\_PATH = cesta k cílové síti, L(AS\_PATH) = délka cesty.

MED, LocPrf, Weight - různé atributy vektoru hodnot metriky.

Při směrování se uplatňuje pravidlo

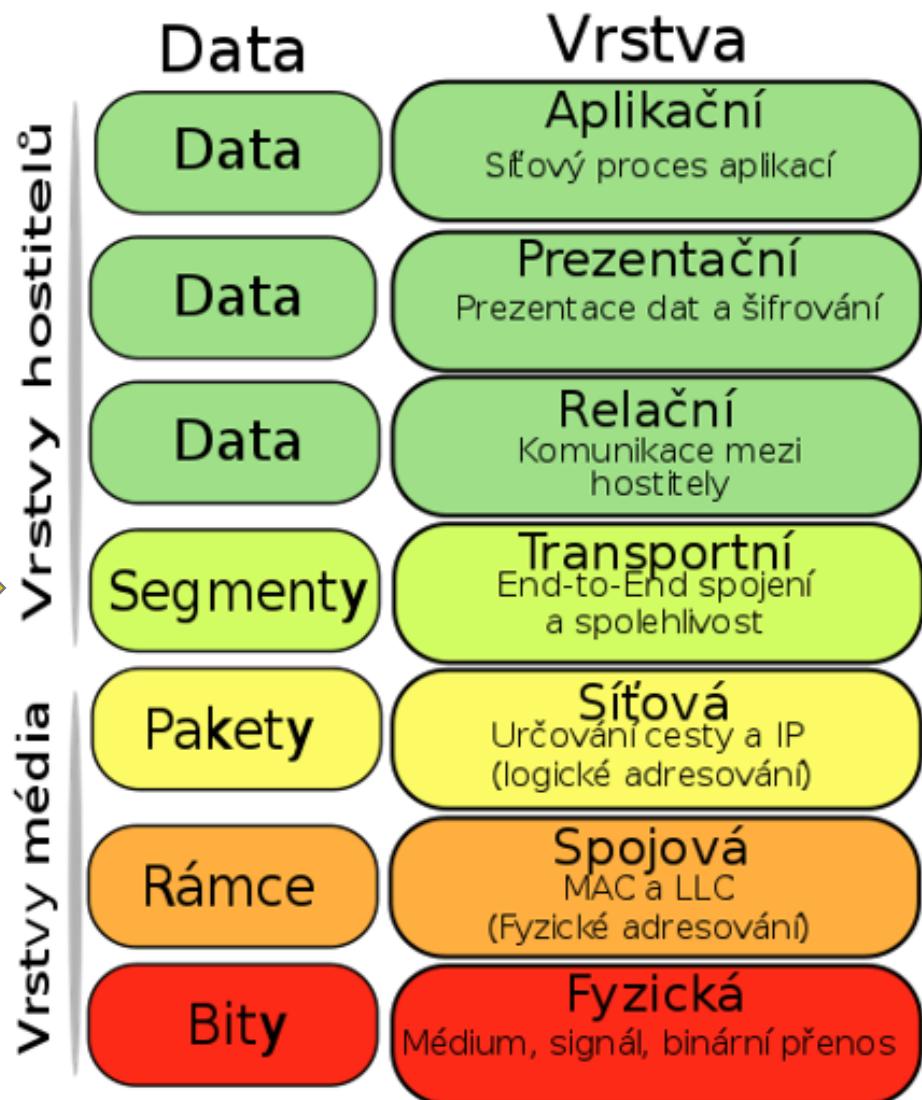
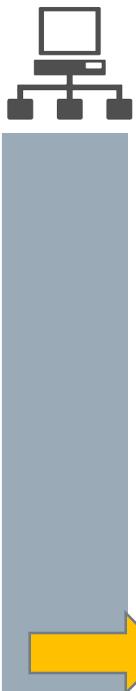
MED > LocPrf > Weight > AS\_PATH > L(AS\_PATH), tzn. pokud mají dvě cesty k jednomu cíli stejně např. MED, rozhoduje se dále podle LocPrf atd.

# Počítačové sítě

6. přednáška - Transportní vrstva, protokoly TCP a UDP



# Význam transportní vrstvy

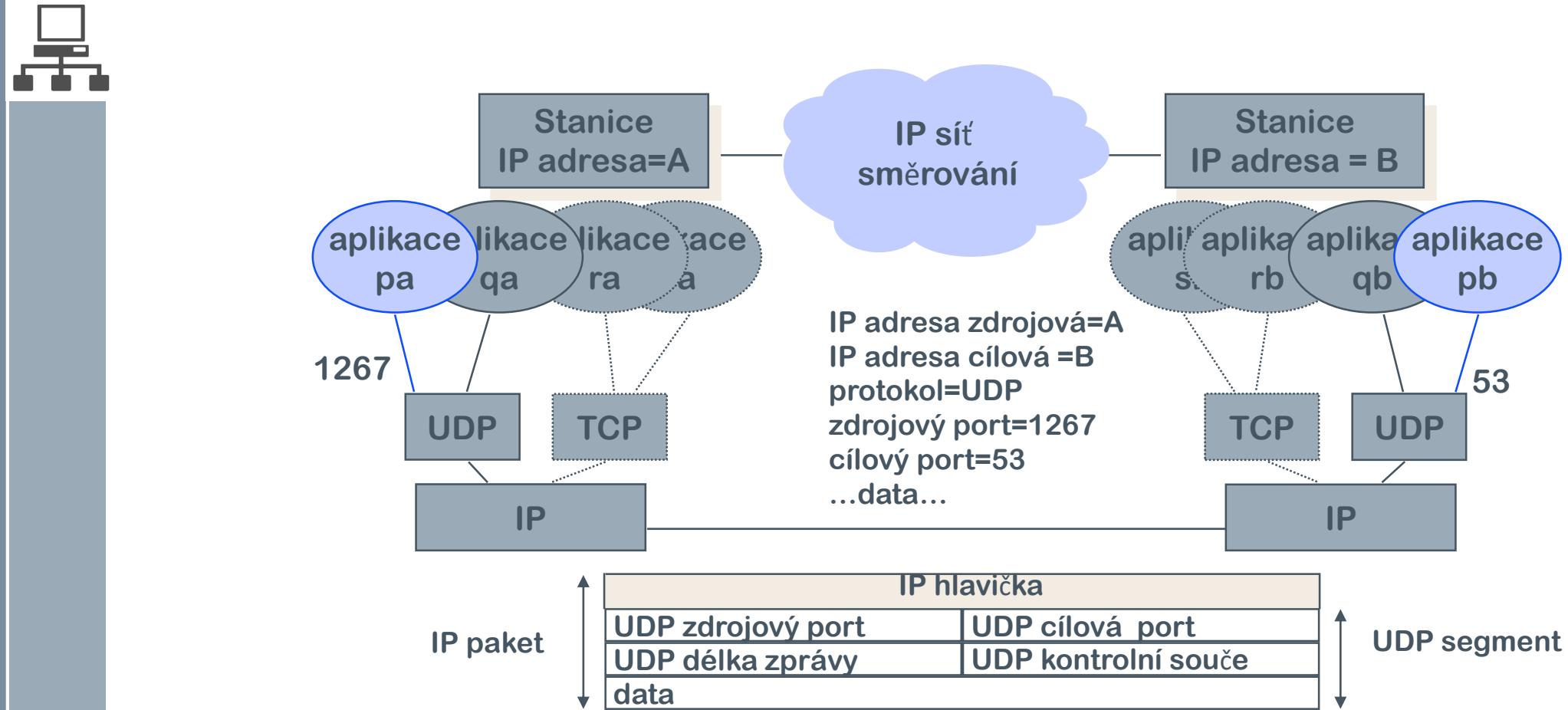


- Hlavním úkolem této vrstvy je doručení dat ke konkrétní službě/aplikaci (procesu) běžícím na stanici dosažitelné prostřednictvím konkrétní IP adresy.
- Transportní vrstva má za úkol zajišťovat kontrolu toku dat a opravy chyb (např. ztracených paketů).
- Rozlišení konkrétní zdrojové a cílové aplikace je založeno na použití **portů**.
- Port = **16 bitový identifikátor**, který se uplatňuje jako další adresační stupeň.
- Doručování dat může být **spolehlivé/potvrzované** anebo **nespolehlivé/nepotvrzované**.
- Minimální přenášená jednotka se označuje jako **segment**.
- Běžně používané protokoly této vrstvy jsou **TCP** = Transmission Control Protocol a **UDP** = User Datagram Protocol.



- **Principy doručování dat na transportní vrstvě**
  - Komunikace prostřednictvím IP adres a portů
  - Parametry přenosu paketů
  - Problémy při doručování paketů
  - Problém dvou armád (zahájení spojení)
- **Algoritmy pro doručování dat (spolehlivé doručování & řízení toku)**
  - Stop & Wait
  - Stop & Go
  - Klouzavé okno
- **Spolehlivé a nespolehlivé protokoly transportní vrstvy**
  - Protokol TCP (spolehlivý)
  - Protokol UDP (nespolehlivý)

# Komunikace prostřednictvím IP adres a portů



**Adresace:** [zdrojová IP adresa, zdrojový Port] → [cílová IP adresa, cílový Port].

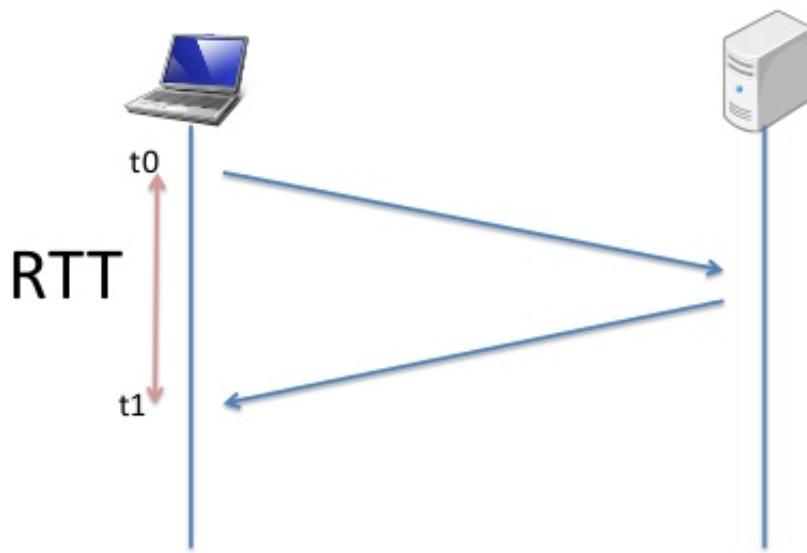
Cílové porty < 1024 se označují jako **privilegované** a alokují si je na svou žádost konkrétní aplikace. Ostatní porty se označují jako neprivilegované a přiděluje je jádro OS dynamicky.

# Parametry přenosu dat

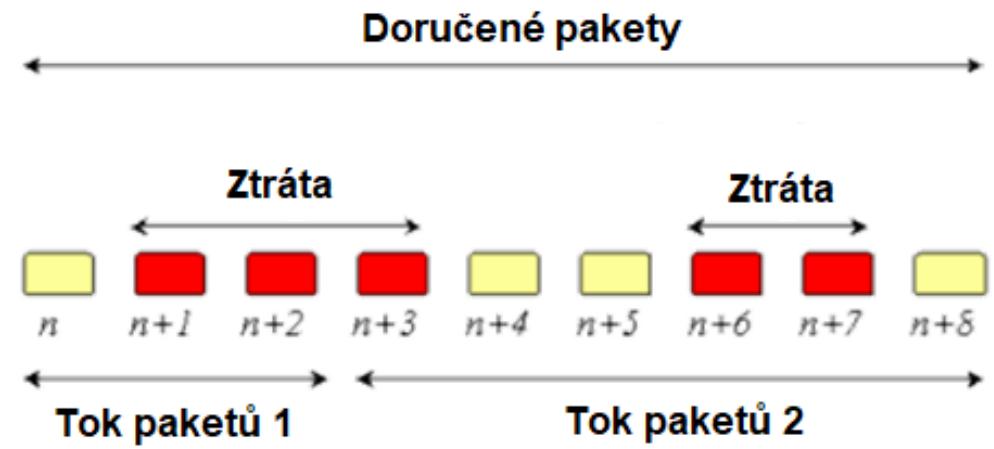


- Pro sledování kvality přenos dat lze monitorovat různé parametry, které pomáhají identifikovat příčiny potenciálních problémů. A jsou to zejména tyto:
- **Ztrátovost (packet loss) [%]** = množství paketů, která se ztratí během přenosu.
  - Řešení spočívá v opakovaném odesílání nedoručených paketů.
- **Zpoždění (delay) [ms]** = průměrná doba na doručení paketů mezi zdrojem a cílem.
  - Dlouhá doba zpoždění má vliv použitelnost služby přenášené daným protokolem (např. na přenos hlasu).
  - Hodnota zpoždění se počítá jako průměr hodnot zpoždění více paketů za sebou.
- **Obousměrné zpoždění (Round Trip Time = RTT) [ms]** = průměrná doba doručení paketu od zdroje k cíli a zpět (typicky dvojnásobek zpoždění).
  - Charakterizuje propustnost sítě v obou směrech.
  - Počítá se rovněž průměrem jako zpoždění.
- **Nestabilita, variabilní zpoždění (jitter)** = Kolísání zpoždění
  - Pakety stejného toku přicházejí v různých časových rozestupech po sobě, což má za následek jejich zpracování po různě dlouhých částech.

# Parametry přenosu názorněji

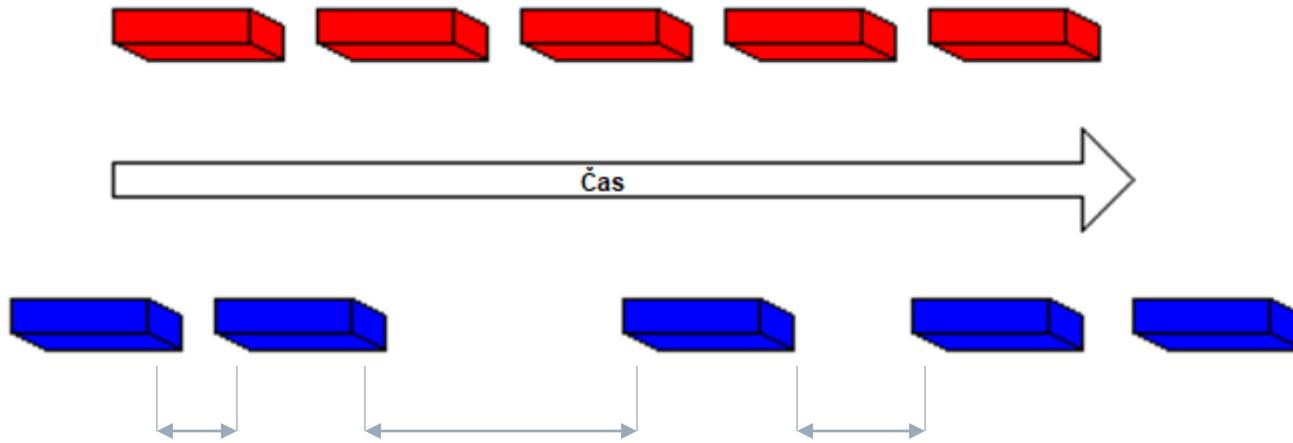


Obousměrné zpoždění



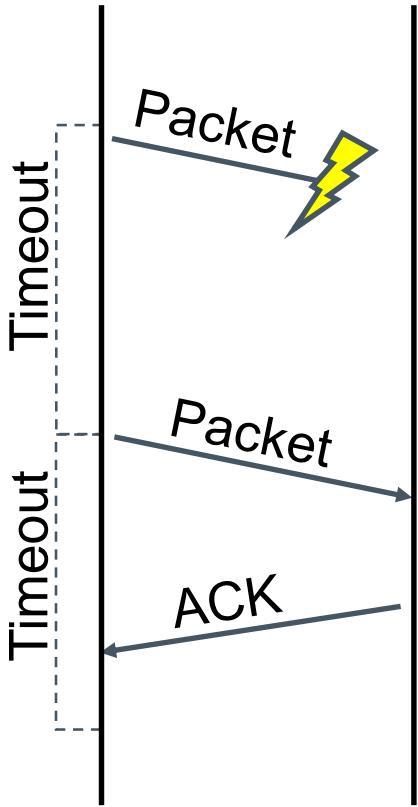
Ztrátovost

# Nestabilita (jitter)

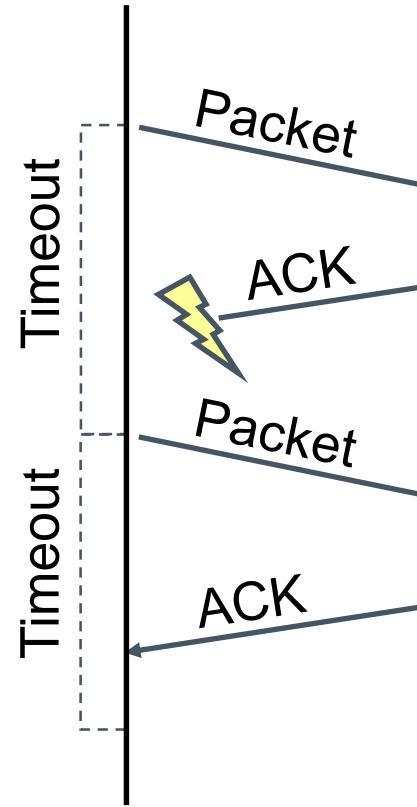


Pakety přicházejí v čase s proměnlivým zpožděním. Oproti ztrátovosti a zpoždění se tento parametr kompenzuje mnohem komplikovaněji, jelikož data chodí někdy často a jindy vůbec.

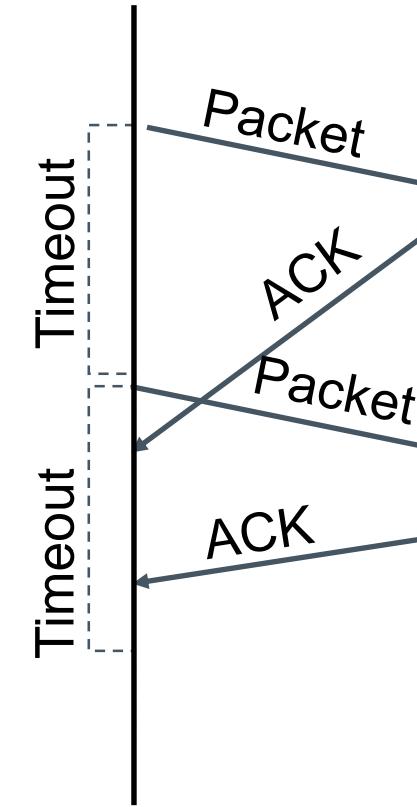
# Obecné problémy při doručování dat



Ztráta paketu



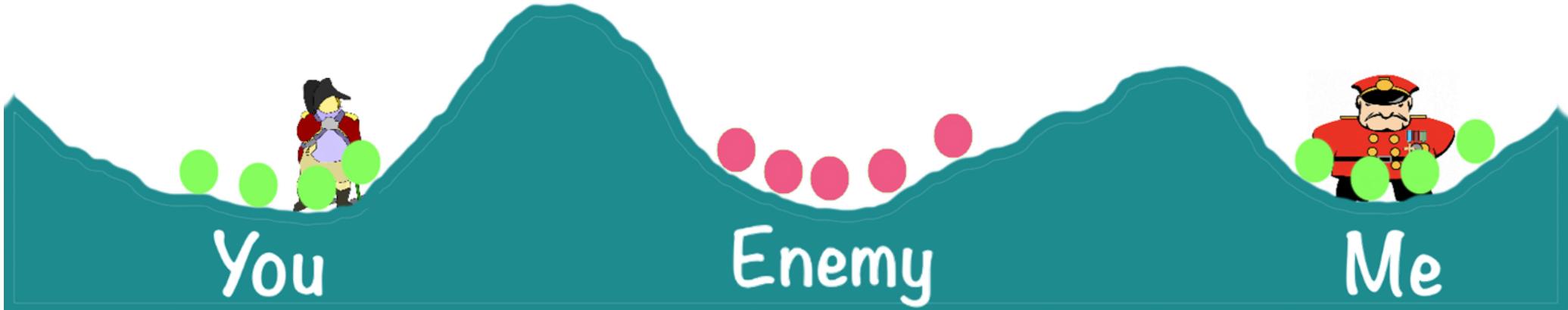
Ztráta potvrzení  
DUPLIKACE  
PAKETU



Opožděné potvrzení  
DUPLIKACE  
PAKETU

# Problém dvou armád (zahájení spojení)

Zdroj: <https://geeks.co.uk/2020/03/two-generals-problem/>



Spočívá v tom, že **zelená armáda** (ZA) vyhraje pouze v okamžiku, když zaútočí na **červenou armádu** ze dvou stran současně, jinak ZA prohraje. Na společném útoku se ovšem musí obě strany ZA domluvit a potvrdit si jej prostřednictvím poslů. Poslové avšak nemusí na druhou stranu dorazit a tím pádem není jasné, zda útok může skutečně začít. **Fáze zahájení útoku jsou následující:**

- 1. Poslat posla s informací o útoku (zpráva).**
- 2. Poslat posla zpět s odpovědí (odpověď).**
- 3. Poslat posla s potvrzením o přijetí odpovědi (potvrzení odpovědi).**

Těmto třem kroků se běžně říká **3-cestná výměna** (3-way handshake) a používá se na začátku a konci spojení dvou stran. Nicméně spolehlivé konečné řešení tohoto problému neexistuje, neboť vždy je potřeba další potvrzení.

# Algoritmy pro doručování dat



## Pro spolehlivé doručování dat:

- Stop & Wait (viz dále)
  - Jednoduchý, ale velmi málo efektivní.
  - Doručuje data přesně v tom pořadí, jak jdou za sebou.

## Pro řízení toku:

- Stop & Go (viz dále)
  - Dokáže ovlivnit množství paketů, které přichází od vysílače.
  - Nedokáže vyřešit problém ztráty paketů.

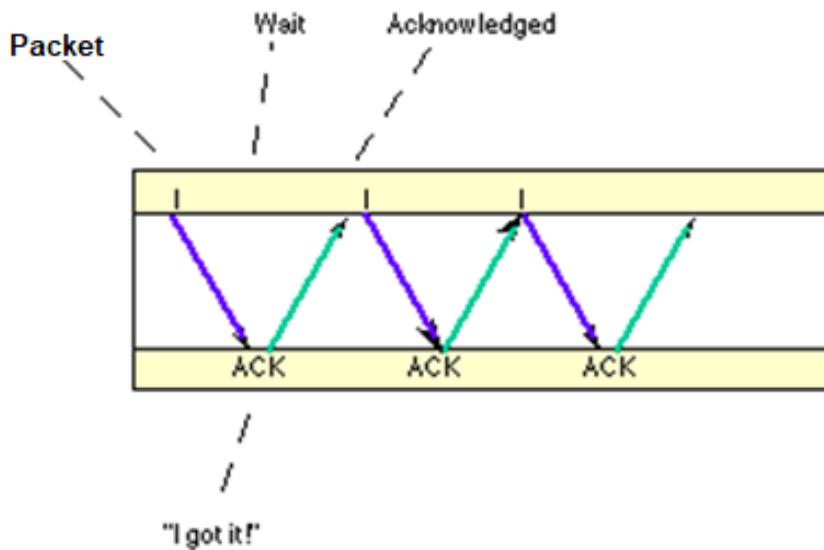
## Pro spolehlivé doručování dat i řízení toku:

- Klouzavé okno (Sliding Window )
  - Složitější, avšak efektivnější než Stop & Wait.
  - Více možných implementací (viz. 2. přednáška, slajd 17).
  - Dovoluje řízení toku úpravou velikosti okna.
  - Úprava velikosti okna se různými mechanismy optimalizuje na základě parametrů přenosu (delay, RTT, packet loss).

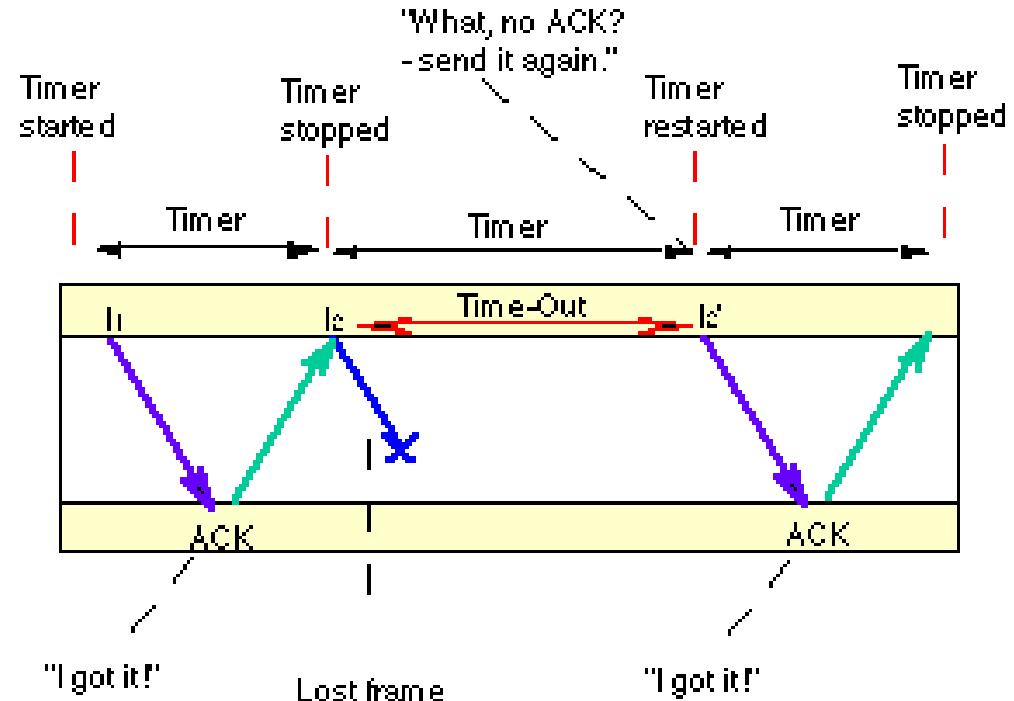
# Algoritmus Stop & Wait



Scénář: A) Normální stav bez chyb



Scénář: B) Nedoručení ACK

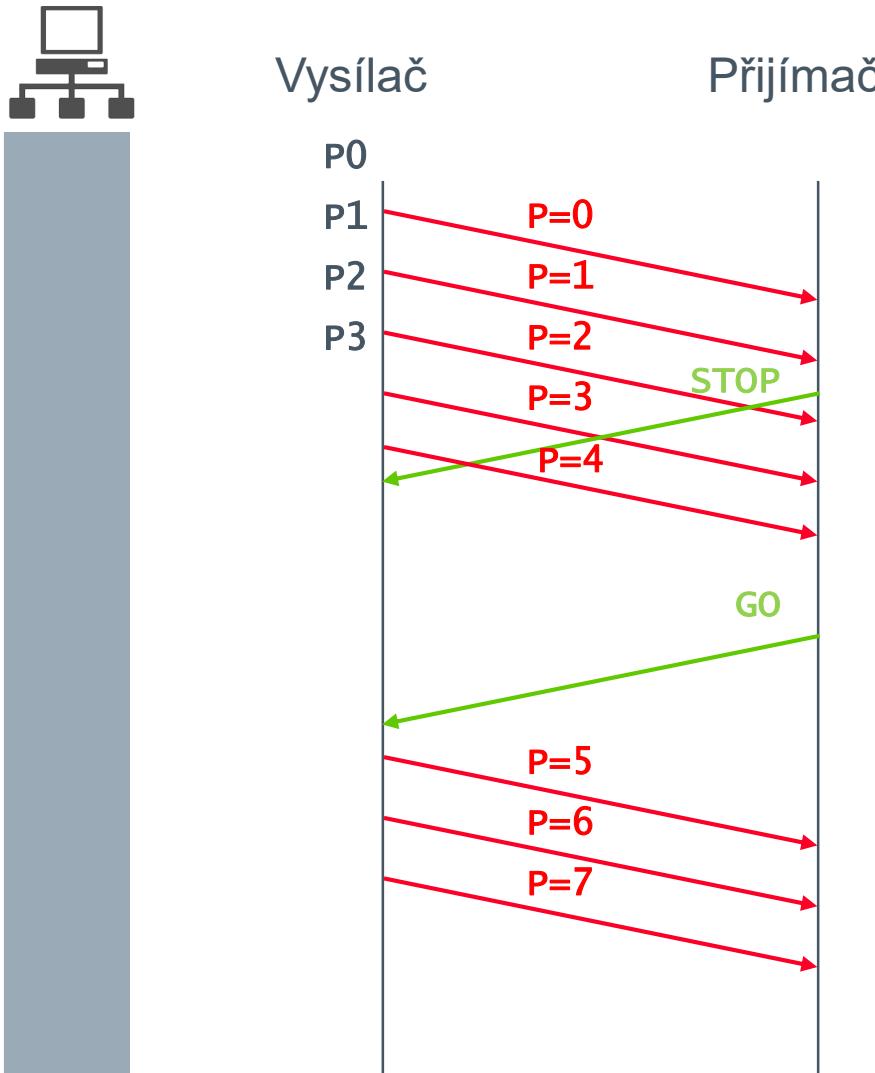


Zdroj: [https://www.isi.edu/nsnam/DIRECTED\\_RESEARCH/DR\\_HYUNAH/D-Research/stop-n-wait.html](https://www.isi.edu/nsnam/DIRECTED_RESEARCH/DR_HYUNAH/D-Research/stop-n-wait.html)

## Scénáře:

- A) Vysílač pošle paket a čeká do doby, než přijde potvrzení. Pak se pošle další paket.
- B) Vysílač pošle paket, na který nedostane potvrzení. Jakmile na vysílači vyprší timeout indikující chybějící potvrzení, pošle vysílač znovu tentýž paket.

# Algoritmus STOP & GO



1. Vysílač posílá Pakety (P=i) přijímači posloupnost paketů, jelikož ještě neobdržel od přijímače zprávu **STOP**.
  2. Jakmile vysílač zprávu STOP obdrží, vysílat přestane.
  3. Vysílač čeká do doby, dokud nedostane od přijímače zprávu **GO**.
  4. Jakmile vysílač obdrží zprávu Go pak opět začne vysílat.

# Algoritmus klouzavého okna



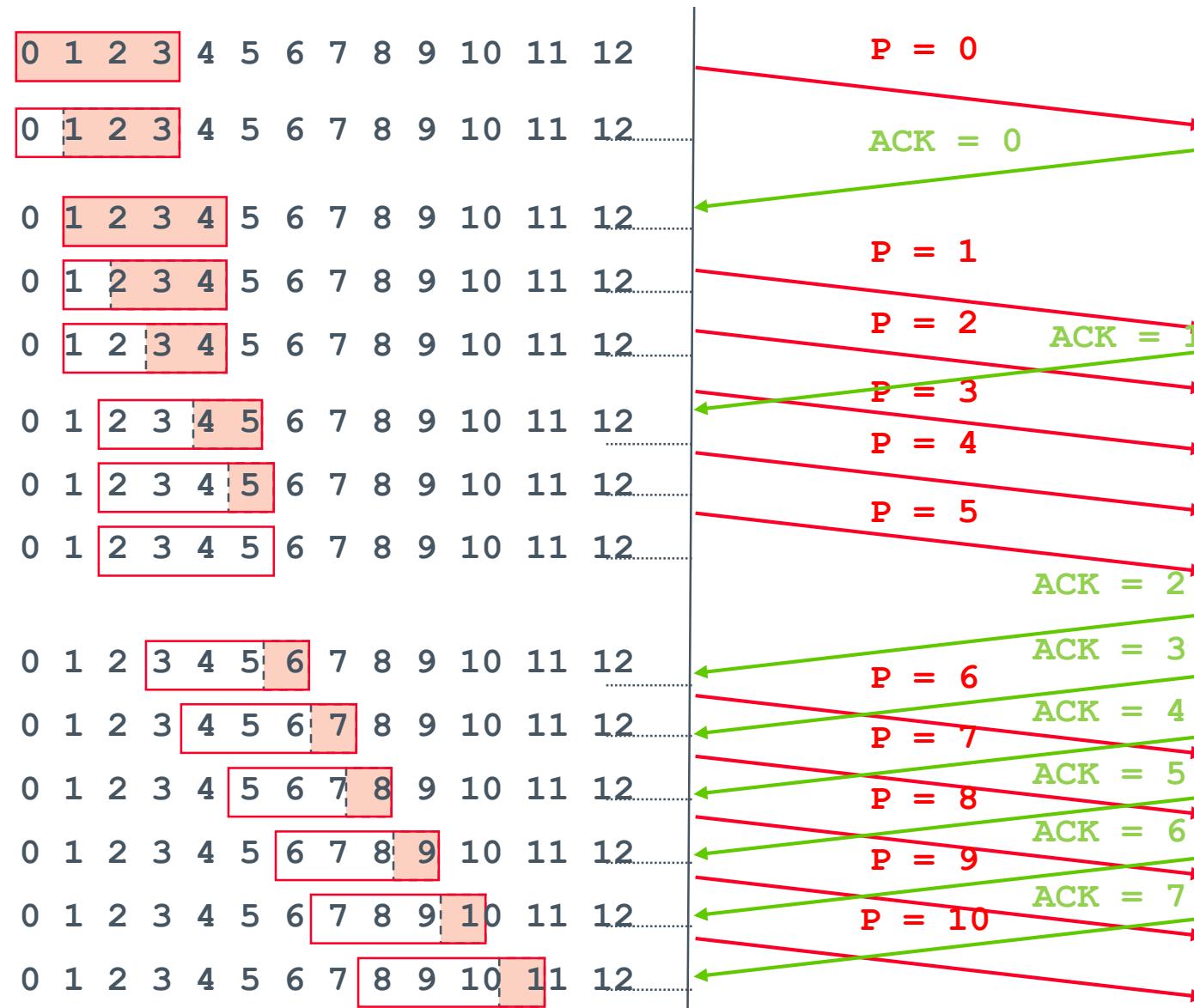
Legenda

Klouzavé okno

Data k odeslání

P = Paket

ACK = Potvrzení



# Dělení protokolů transportní vrstvy



- **Spolehlivé** = data jsou vždy doručena od zdroje k cíli, pokud je to možné.
  - **Spojově** orientované, mezi směrovači se sestavuje a drží virtuální spojení.
  - Návrh a implementace je složitější oproti nespolehlivým protokolům.
  - Musí obsahovat mechanismy detekce a opravy chyb.
  - Dokáží řídit tok, předcházet zahlcení.
  - Nejznámější a nejvíce používaným je protokol **TCP**.
- **Nespolehlivé** = data se doručují k cíli, ale mohou se během přenosu ztratit.
  - **Paketově** orientované, data se posílají po částech a spojení se nevytváří.
  - **Jednodušší, snažší implementace** oproti spolehlivým.
  - V případě spolehlivého řešení na nižší vrstvách OSI modelu jsou v porovnání se spolehlivými rychlejší, jelikož mají menší režii přenosu.
  - Nejznámější a nejvíce používaným je protokol **UDP**.

# Transmision Control Protocol (TCP)



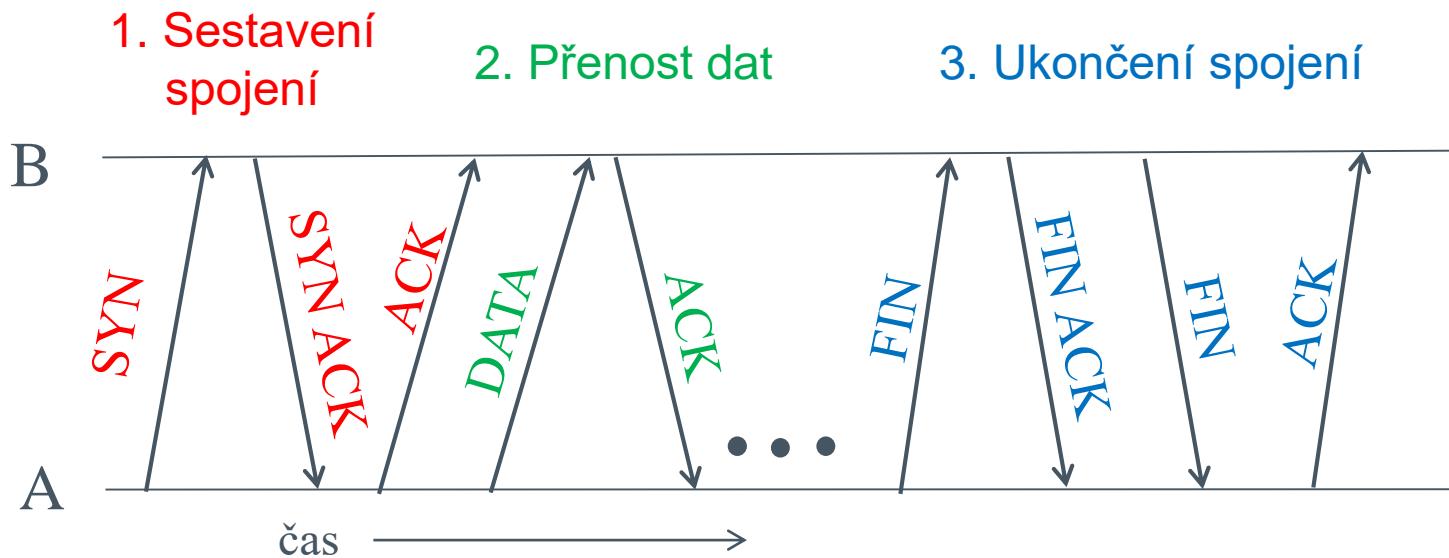
- TCP paket = položka data IP protokolu.
- Umožňuje **řízení toku** (klouzavé okno) i eliminaci zahlcení (CW = Congestion Window)
- Není šifrovaný.
- **Zabezpečený** oproti chybám, garantuje i pořadí doručených paketů.
- Detekuje duplicitní pakety.
- Je duplexní (odesílání a příjem z obou stran, každá strana otevírá vlastní spojení).
- Tok protokolu TCP je řízen prostřednictvím příznaků v hlavičce paketu (žlutá barva).



## Příznaky TCP:

U = Urgent, A = ACK (potvrzení), P = aplikační data, R = Reset, S = SYN, F = FIN (viz dále).

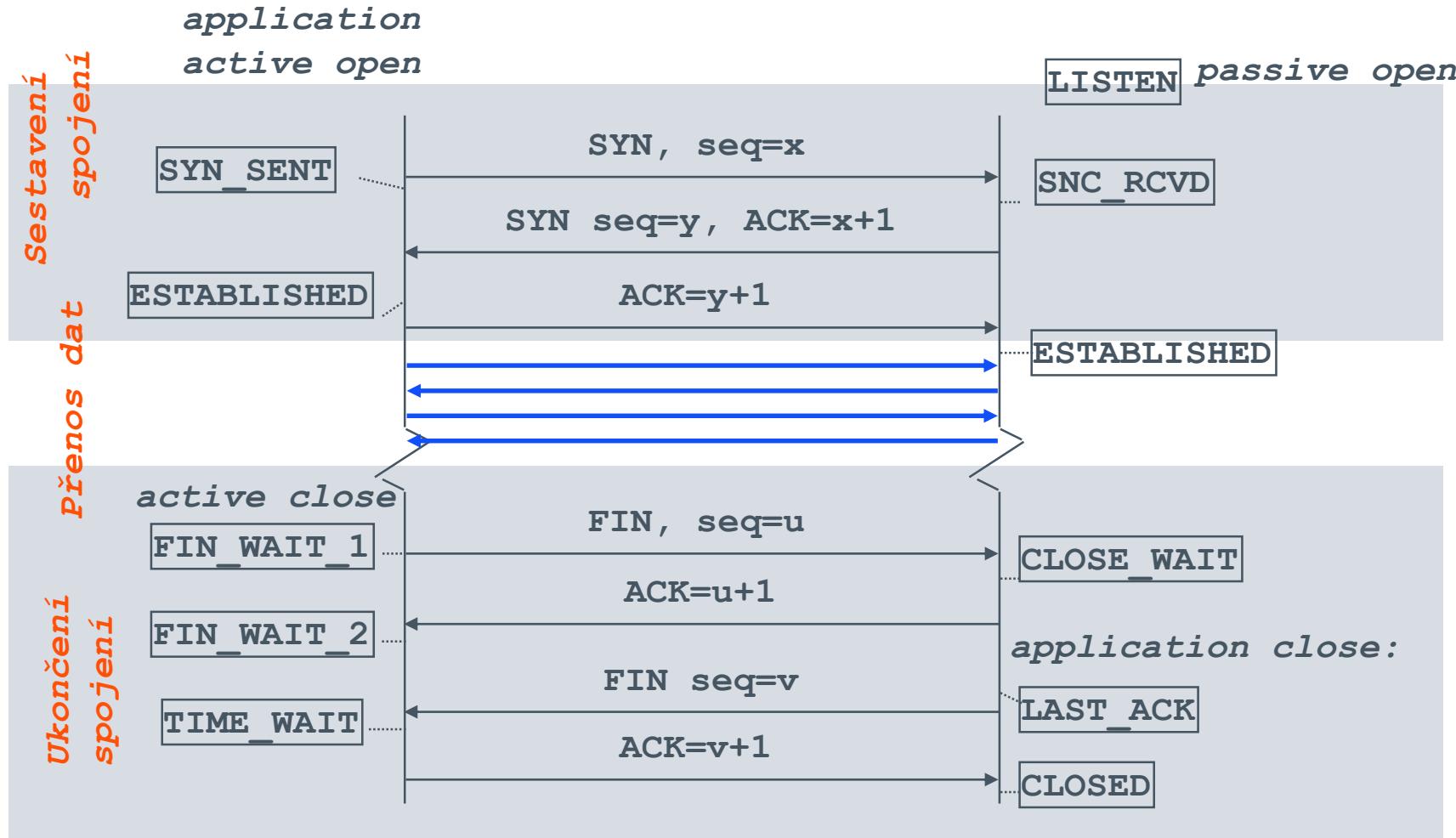
# Životní cyklus TCP spojení



TCP spojení je vždy duplexní.

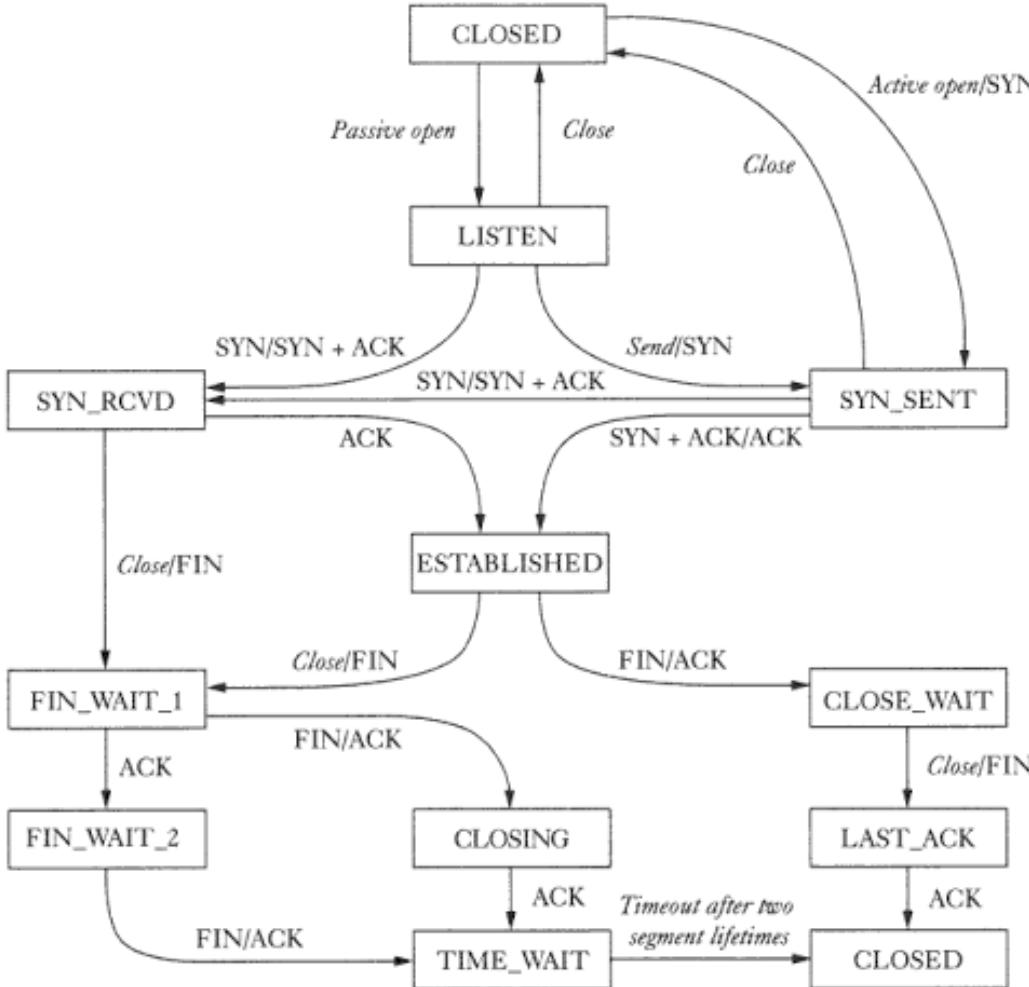
1. Pokud je v hlavičce TCP paketu nastaven příznak **SYN**, znamená to, že se spojení otvírá ze strany A. **SYN ACK** = otevření spojení ze strany B a zároveň B potvrzení doručení SYN od A. **ACK** = potvrzení SYN vyslaného B k A .
2. **DATA** = posílání dat od A, **ACK** = potvrzení doručených dat (od B).
3. **FIN** = uzavření spojení ze strany A, **FIN ACK** = akceptování uzavření spojení ze strany B, **FIN** = uzavření spojení ze strany B, **ACK** = akceptování uzavření spojení od A.

# Fáze TCP ve spojení s jeho stavy, sekvenční čísla



**Sekvenční číslo (seq)** = aktuální pořadové číslo paketu v rámci spojení. Počáteční hodnota je domluvena při zahájení spojení pro obě strany, typicky se nastavuje náhodně a dále inkrementálně zvyšuje.

# Stavový diagram TCP protokolu



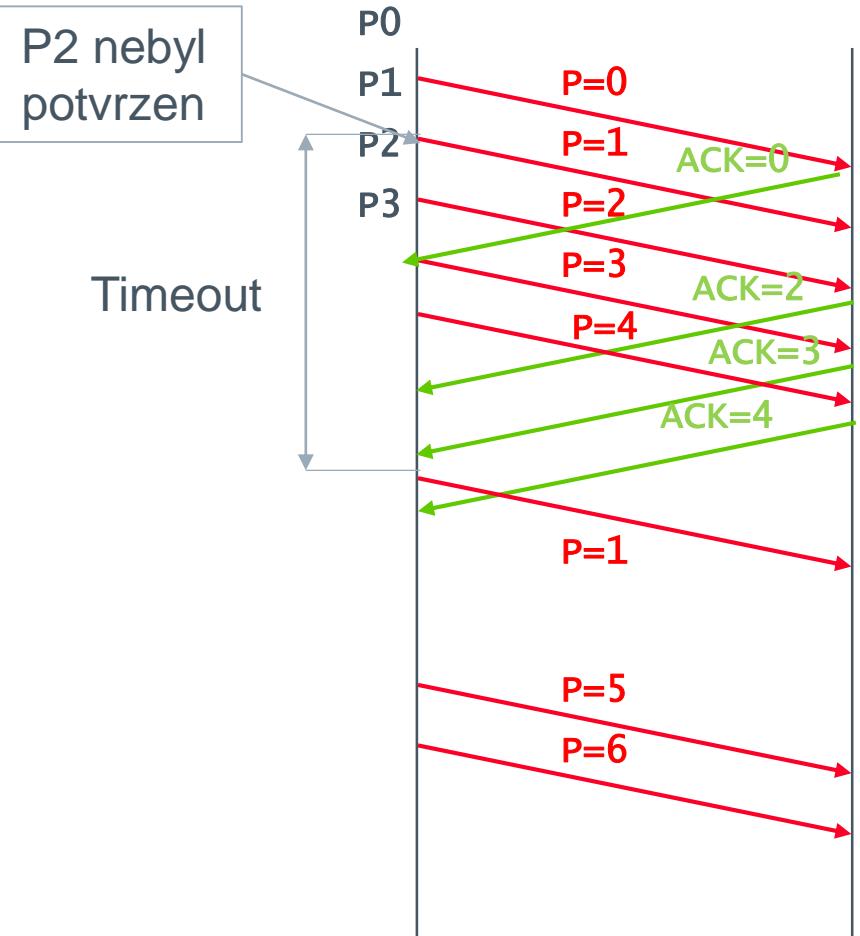
Zdroj: <https://www.chegg.com/homework-help/questions-and-answers/computer-networks-consider-following-tcp-state-transition-diagram-give-appropriate-timing--q67967265>

# Nastavení a úprava délky okna (DO)



- Pro řízení toku se používá mechanismus klouzavého okna.
- Aplikuje se pouze **ve směru od vysílače k přijímači**. Nicméně nastavení okna si **ovládá každá strana spojení samostatně**.
- DO = počet TCP paketů se posílá **najednou bez potvrzení**.
- Velikost DO **navrhoje vždy přijímač**. Pokud hodnotu přijímač nastaví na 0, vysílač přestane zcela posílat data.
- DO si upravuje dynamicky v závislosti na svých možnostech přijímač tak, aby nedošlo k jeho zahlcení.

# Fast Retransmit u TCP

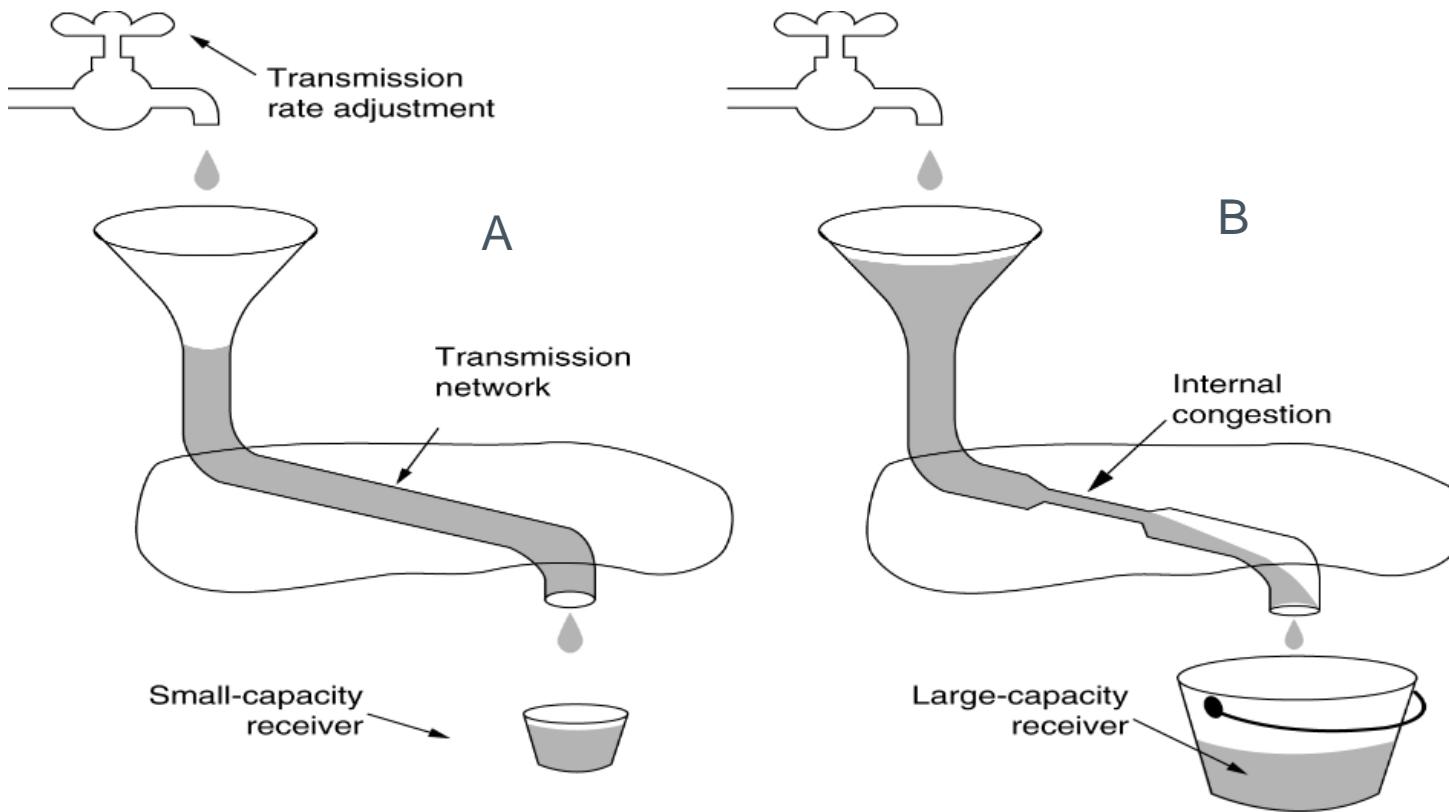


Komunikace běží podle algoritmu **klouzavého okna**.

1. Vysílač si navíc pamatuje dobu, po kterou mu určitý paket (P2) nebyl např. z důvodu jeho ztáty potvrzen.
2. Jakmile doba bez potvrzení je delší než Timeout, vysílač sám znovu vyšle nepotvrzený paket.

Důvodem tohoto řešení je **zvýšení efektivity přenosu** ve smyslu zamezení opakování vysílání více paketů, než je skutečně nutno.

# Zahlcení přijímače vs zahlcení (congestion) linky



Zdroj: [https://www.researchgate.net/figure/a-Flow-control-problem-b-Congestion-control-problem-17\\_fig2\\_324992523](https://www.researchgate.net/figure/a-Flow-control-problem-b-Congestion-control-problem-17_fig2_324992523)

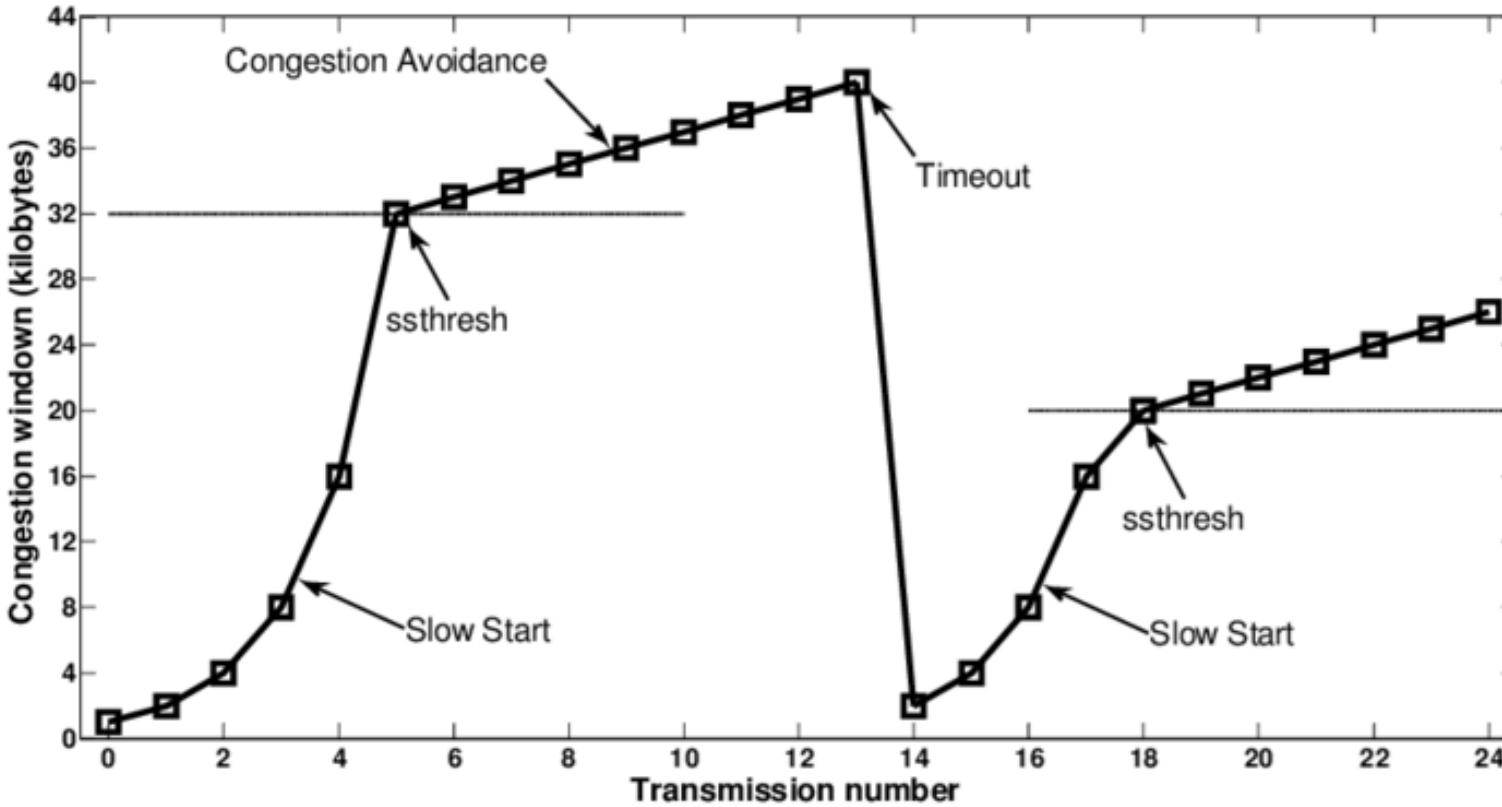
- Nastavením velikosti okénka si **přijímač** řídí množství odesílaných dat (transmission control) vysílačem dle svých možností (snadné). Tento algoritmus mají všechny varianty TCP stejný.
- Vysílač** musí kvůli možnému zahlcení linky na cestě regulovat množství odesílaných dat (složitější). Použitý algoritmus se liší dle konkrétní **verze TCP protokolu**.

# Nastavení DO pro vysílání



- DO pro vysílání se označuje jako CWL = Congestion Window Length.
- CWL = **maximální počet paketů, které může vysílač odeslat najednou, aby nedošlo k zahlcení linky.**
- **Hodnotu CWL si určuje vysílač** vnitřně pro sebe (způsobů je více), nicméně některé protokoly využívají i informace od některého okolního směrovače či přijímače.
- Pokud je délka klouzavého okna (tu nastavuje přijímač) **vyšší než CWL**, posílá se **maximálně CWL paketů**.
- Různé implementace nastavení CWL jsou podstatou **různých verzí TCP** protokolu (např. Reno, Tahoe, Vegas, New Reno, CUBIC, viz dále).
- Tyto algoritmy se stále ještě vyvíjejí, ale neexistuje zatím žádné řešení, které by bylo vždy nejlepší.

# Nastavení CWL a jeho fáze (varianta TCP Tahoe)

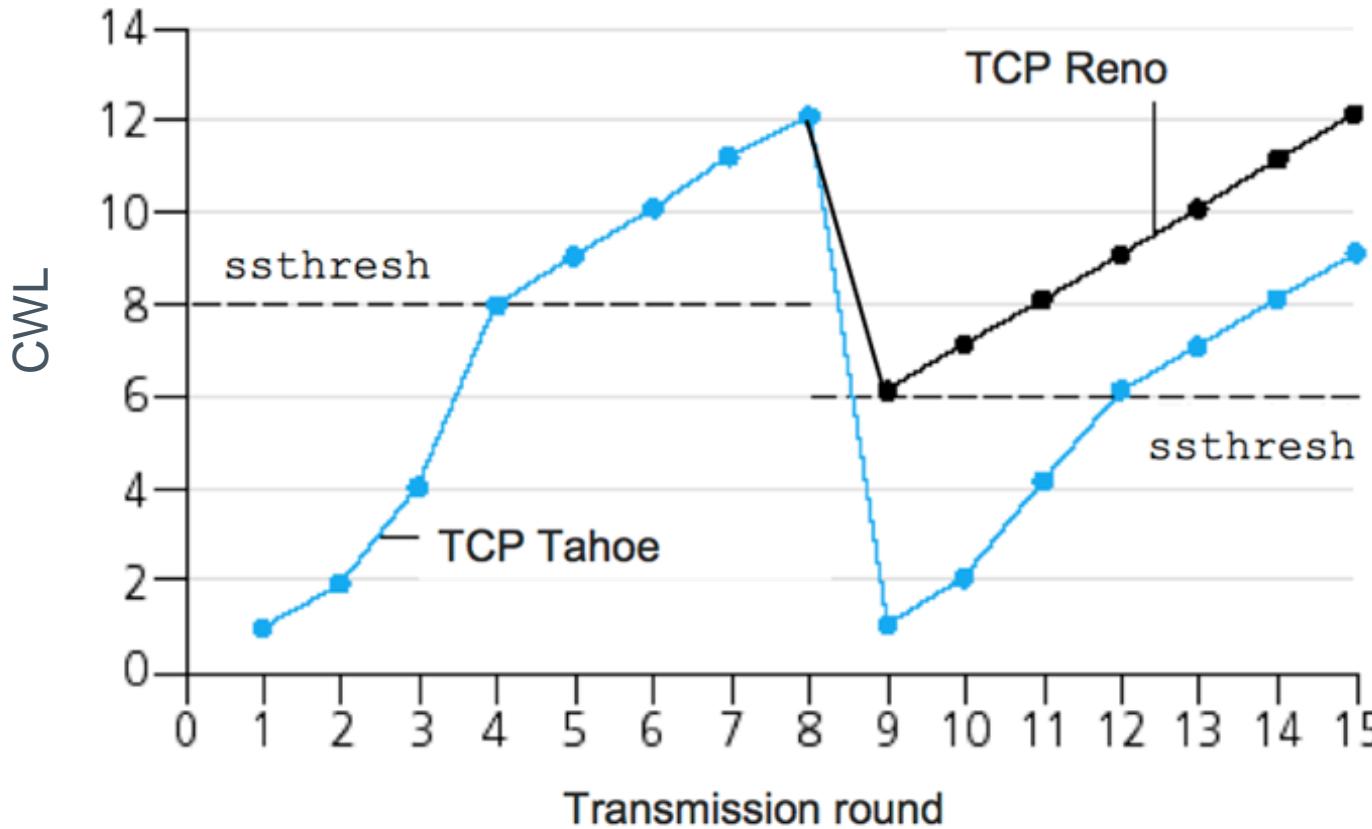


Zdroj: [https://www.researchgate.net/figure/TCP-congestion-control-example-with-slow-start-and-congestion-avoidance-phase\\_fig2\\_316898102](https://www.researchgate.net/figure/TCP-congestion-control-example-with-slow-start-and-congestion-avoidance-phase_fig2_316898102)

1. Na začátku **CWL = 1** a počet odesílaných paketů se zvyšuje při úspěšném doručení **vždy 2x**.
2. Jakmile počet odeslaných paketů dosáhne hodnoty **ssthresh** (na začátku 32KB), **CWL se zvyšuje lineárně** do doby, než dojde k výpadku potvrzení (Timeout).
3. Po detekci výpadku se **ssthresh sníží na polovinu** a pokračuje se bodem 1.

# Popis varianty TCP Reno a srovnání s TCP Tahoe

Zdroj: [https://www.researchgate.net/figure/TCP-congestion-control-example-with-slow-start-and-congestion-avoidance-phase\\_fig2\\_316898102](https://www.researchgate.net/figure/TCP-congestion-control-example-with-slow-start-and-congestion-avoidance-phase_fig2_316898102)

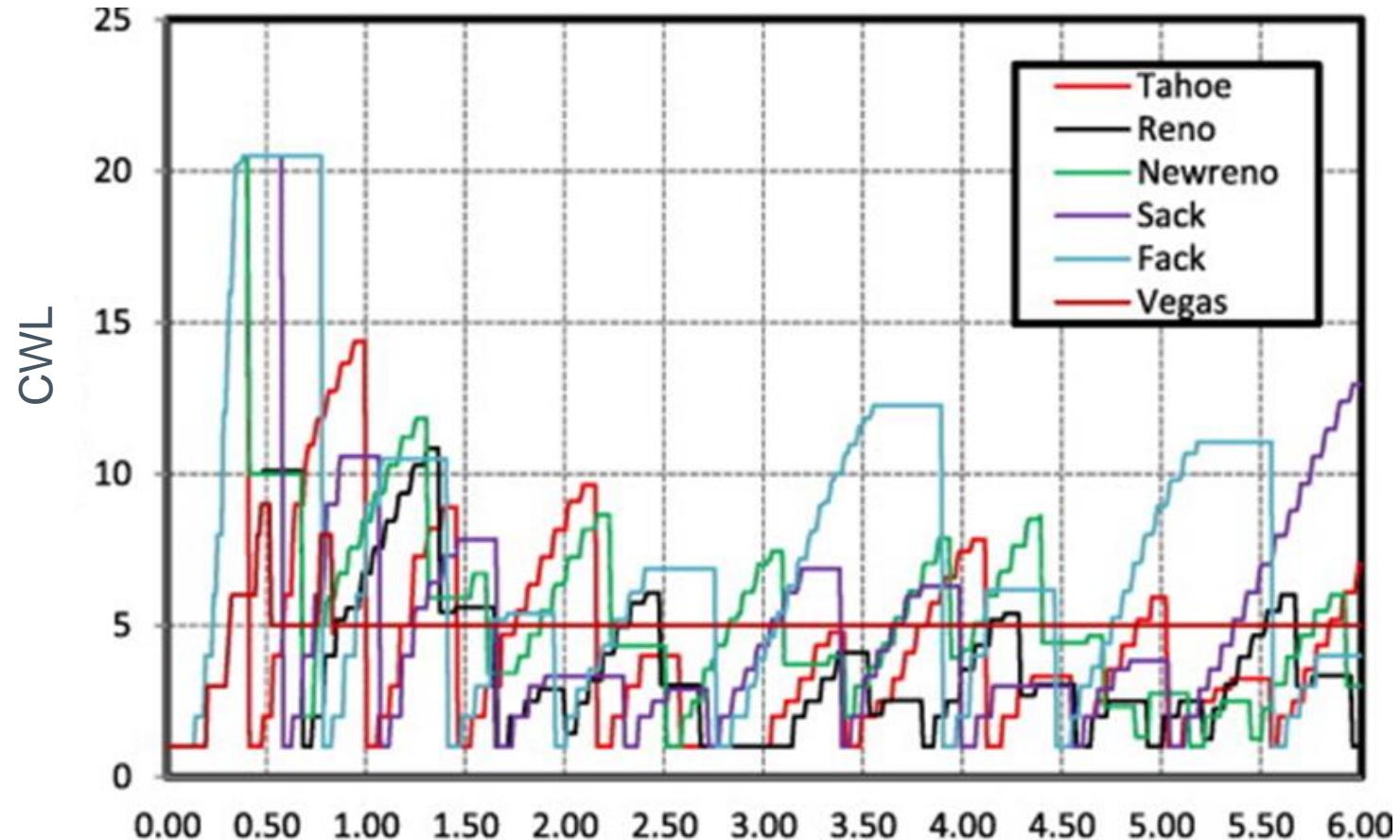


**TCP Reno** a **TCP Tahoe** mají stejný způsob nastavení CWL až do doby, kdy dojde k výpadku potvrzení (8). Tahoe nastaví CWL na 1. Reno, které je optimističtější, CWL nastaví na polovinu hodnoty před výpadkem. Reno sníží hodnota ssthresh o 25%, Tahoe je snižuje na polovinu. Pokud je linka poruchová pouze krátkodobě, Reno dokáže mnohem rychleji obnovit původní vysílání.

# Srovnání stanovení CWL u různých druhů TCP - FYI



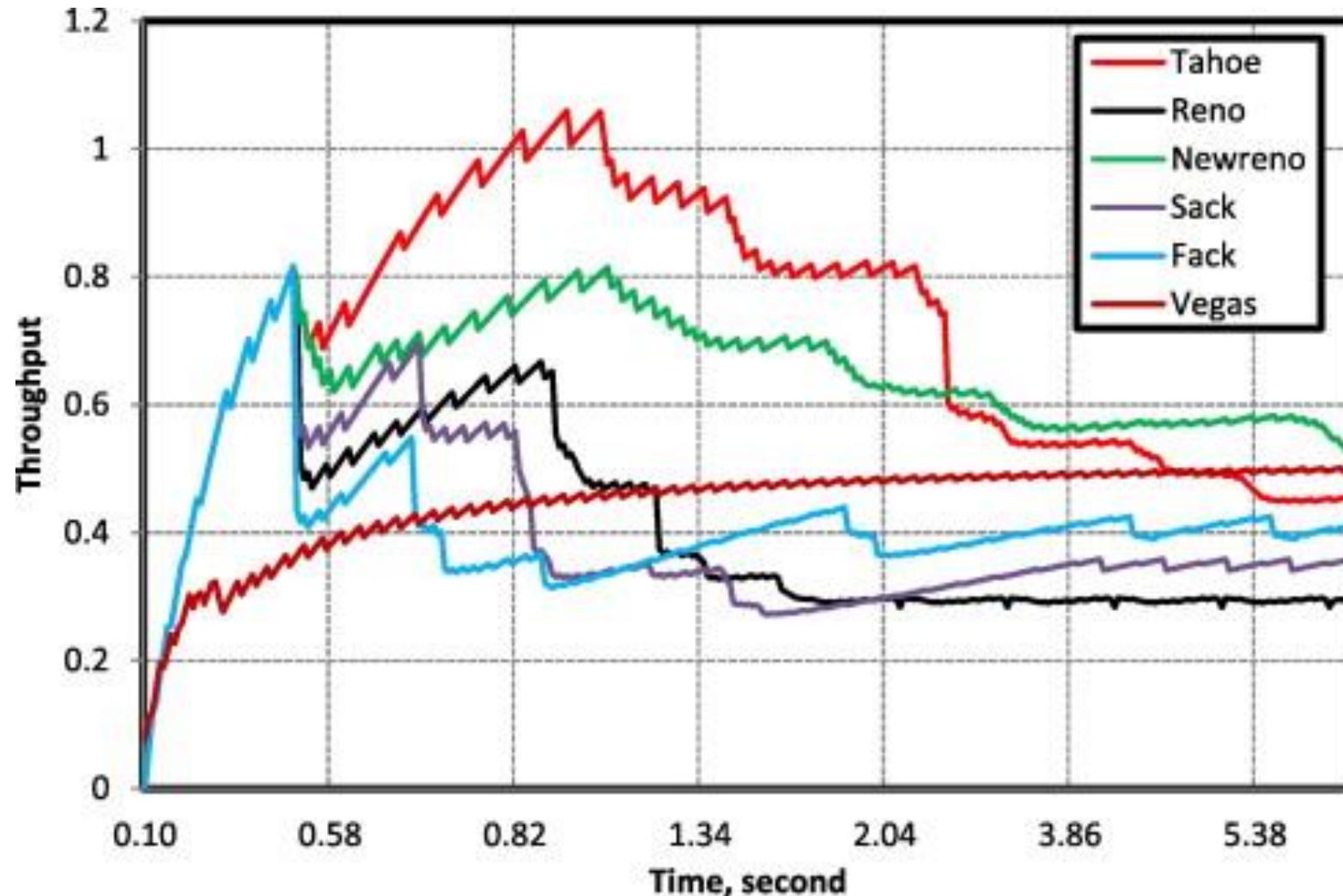
Zdroj: [https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants\\_fig1\\_241186797](https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants_fig1_241186797)



# Srovnání propustnosti různých verzí TCP v ideálních podmíkách - FYI



Zdroj: [https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants\\_fig1\\_241186797](https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants_fig1_241186797)

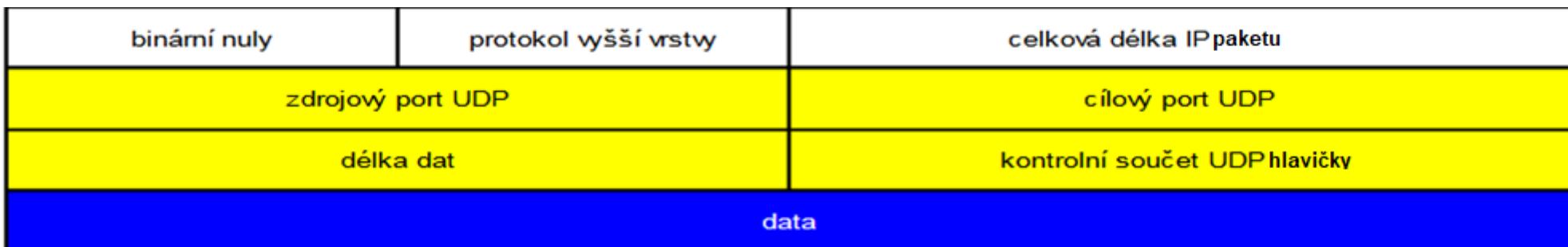


# User Datagram Protocol (UDP)



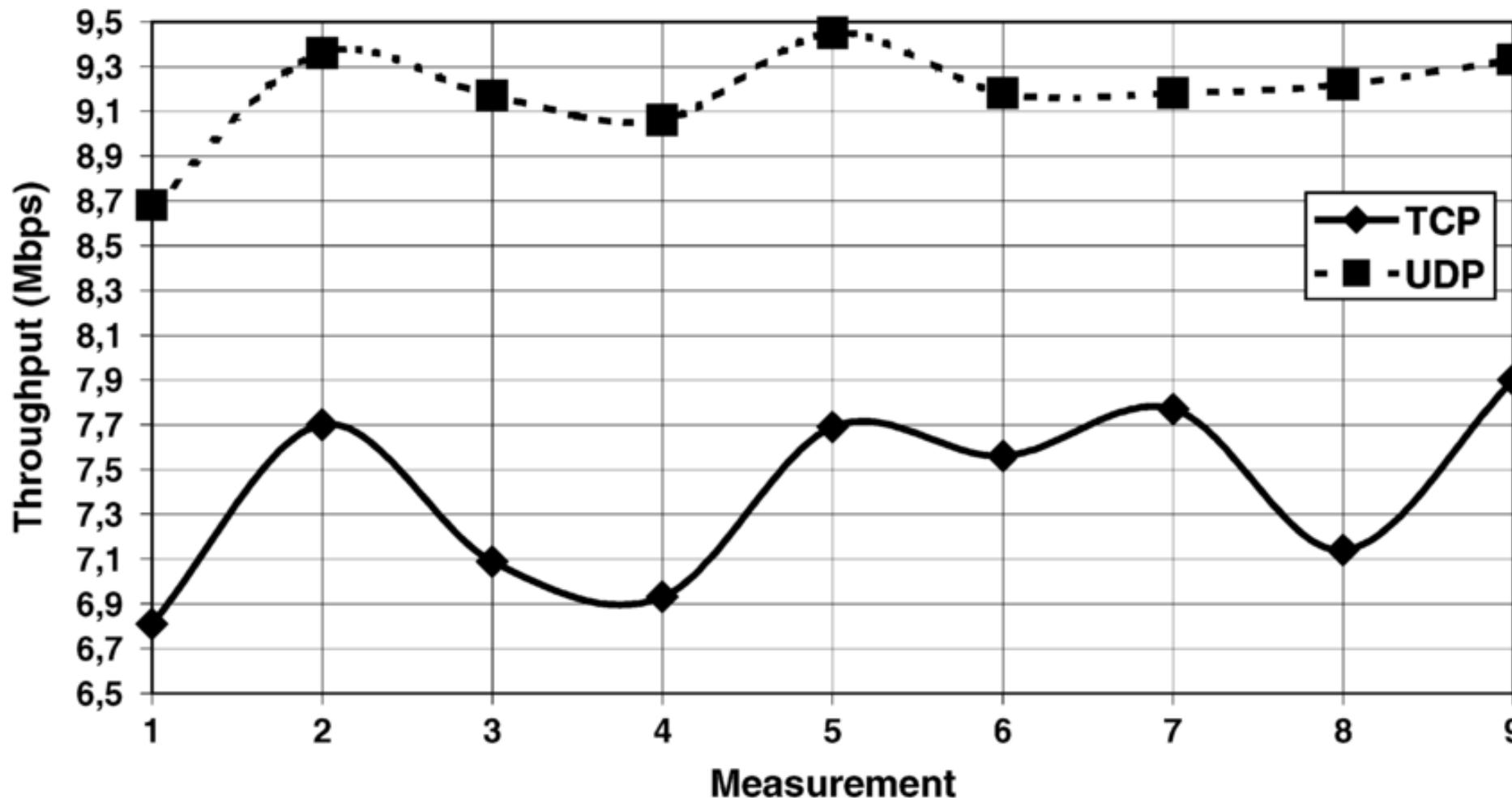
- Přenáší se v položce data paketu IP protokolu.
- Používá se pro jednoduchou výměnu dat.
- V ideálních podmírkách **má lepší propustnost než TCP** kvůli nižší režii.
- Používá se běžně např. pro přenos hlasu či videa (např. RTP = Real Time Protocol), kde občasná ztráta paketů tolik nevadí.
- **Nezabezpečený**
  - Negarantuje doručení paketů a ani pořadí jejich doručení.

**Formát hlavičky (žlutá) + dat:**



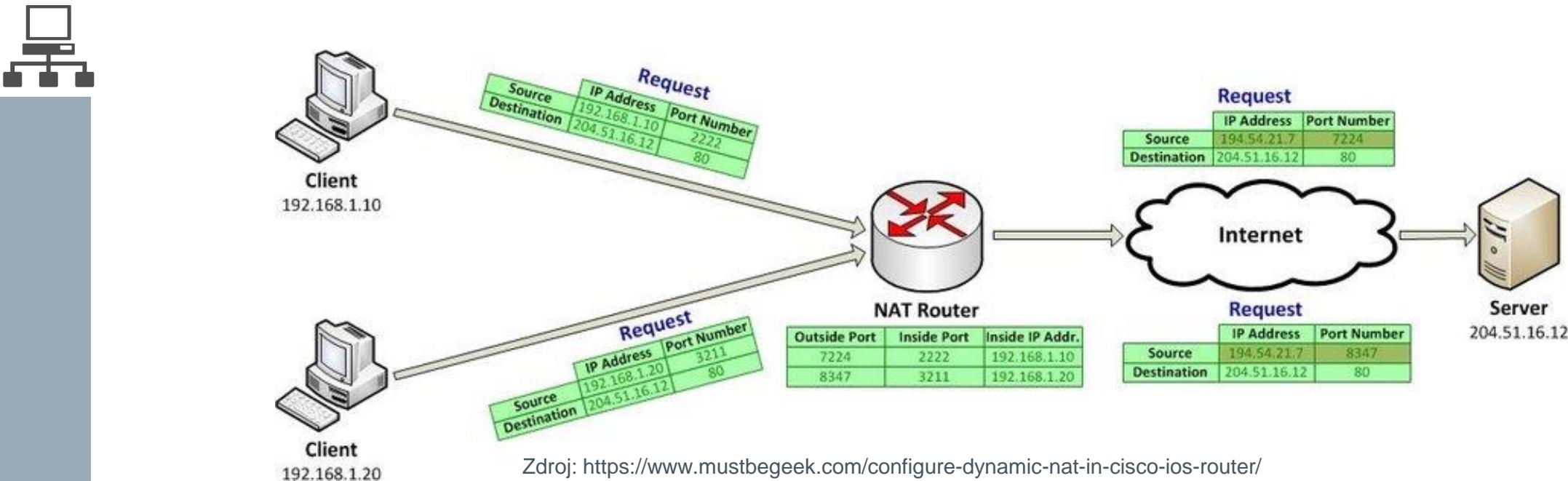
# Srovnání propustnosti TCP a UDP v ideálních podmírkách

Zdroj: [https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants\\_fig1\\_241186797](https://www.researchgate.net/figure/Comparison-Lost-Packets-versus-time-of-different-TCP-variants_fig1_241186797)



Propustnost se liší o **několik jednotek %**, toto platí i pro vyšší rychlosti, ovšem pro přenos jsou nutné ideální podmínky, tj. **bez zahlcení linky a hlavně beze ztrát při přenosu**.

# Transportní vrstva a překlad adres (NAT)



Díky portům je možné v případě NATu odlišit **různá současná spojení jdoucí od více stanic (client) ke stejnemu cíli přes stejnou veřejnou IP adresu**. Každé spojení je identifikováno na základě {zdrojová IP adresa<sub>vnitřní</sub>, zdrojový PORT<sub>vnitřní</sub>}, pro toto směrovač vyvoří mapování na {zdrojová IP adresa<sub>vnější</sub> zdrojový PORT<sub>vnější</sub>}. Jakmile dorazí paket od cíle zpět na {zdrojová IP adresa<sub>vnější</sub> zdrojový PORT<sub>vnější</sub>}, provede směrovač **opět překlad podle dříve vytvořeného namapování** na {zdrojová IP adresa<sub>vnitřní</sub>, zdrojový PORT<sub>vnitřní</sub>}.

**Poznámka.** Zdrojový port, pokud je to možné (např. konflikt s jiným spojením), se měnit nemusí.

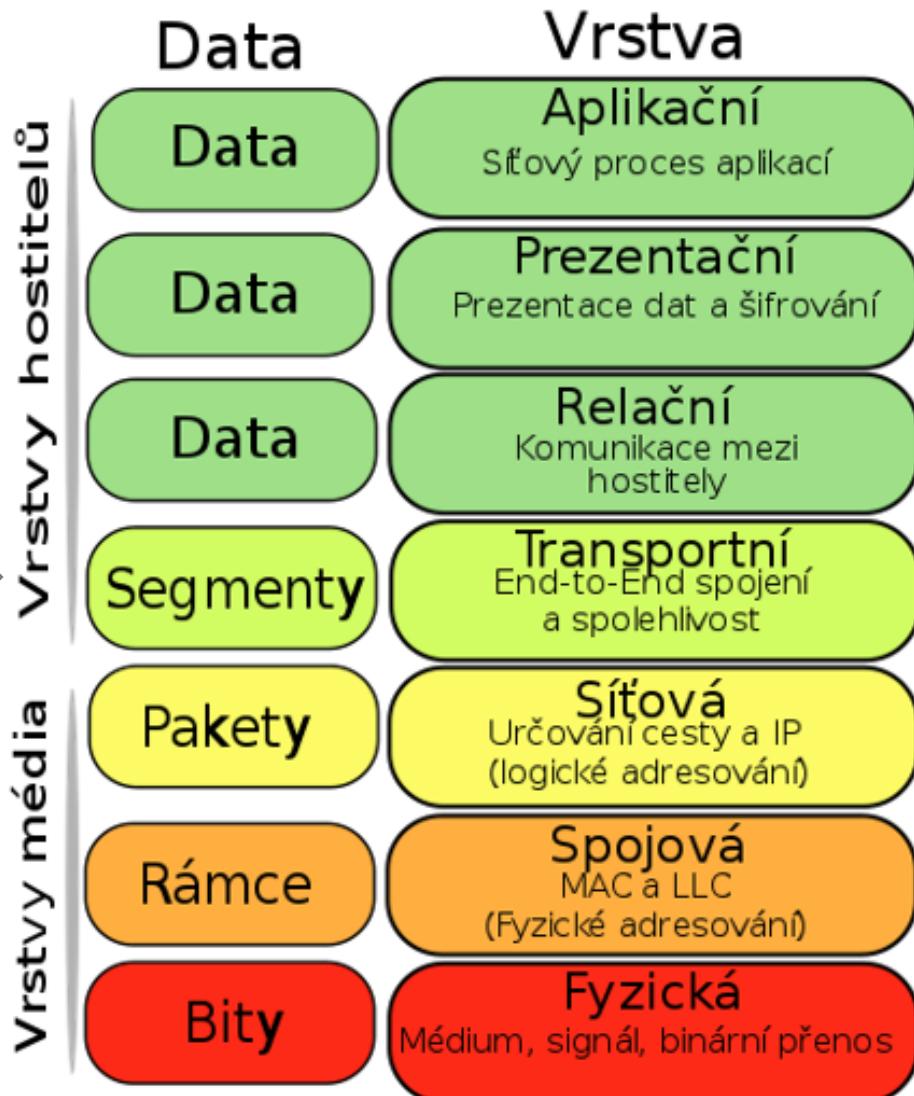
# Počítačové sítě

7. přednáška – Systém doménových jmen (DNS)





# Důvod zavední Domain Name System (DNS)



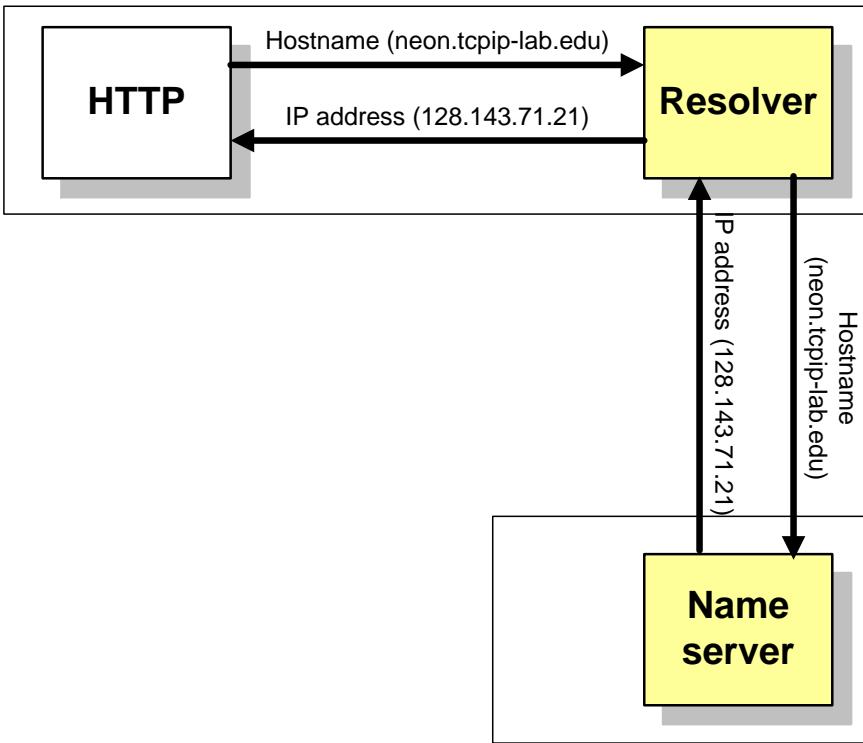
- IP adresy jsou pro lidi **špatně zapamatovatelné a neintuitivní**.
- Pro identifikaci konkétní stanice v síti je lepší zvolit její jméno reprezentované řetězcem - **doménovým jménem**, např. fit.cvut.cz.
- **Domain Name System (DNS)** = systém primárně určený pro překlad doménovýchjmén na IP adresy a naopak.
- DNS je hierarchický.
- DNS ve své původní verzi je **nezabezpečený**.
- **DNSSEC** = zabezpečené rozšíření DNS tak, aby bylo možné zabránit zfalšování odpovědi.
- DNS může kromě překladu doménových jmen provádět i další funkce (např. rozdělování zátěže či omezení přístupu).
- DNS je zásadní i pro další služby (např email, VoIP).
- **DNS běží standardně na portu 53 a většina jeho implementací využívá UDP protokol**.

# Osnova přednášky



- Role DNS v rámci OS a jeho konfigurace.
- Struktura DNS.
- Domény v DNS, doménové záznamy.
- DNS protokol a DNS dotazy.
- Princip překladu adres v DNS.
- Dynamické a zabezpečené varianty DNS.
- Bezpečnostní rizika DNS.

# Role DNS v rámci OS a jeho konfigurace



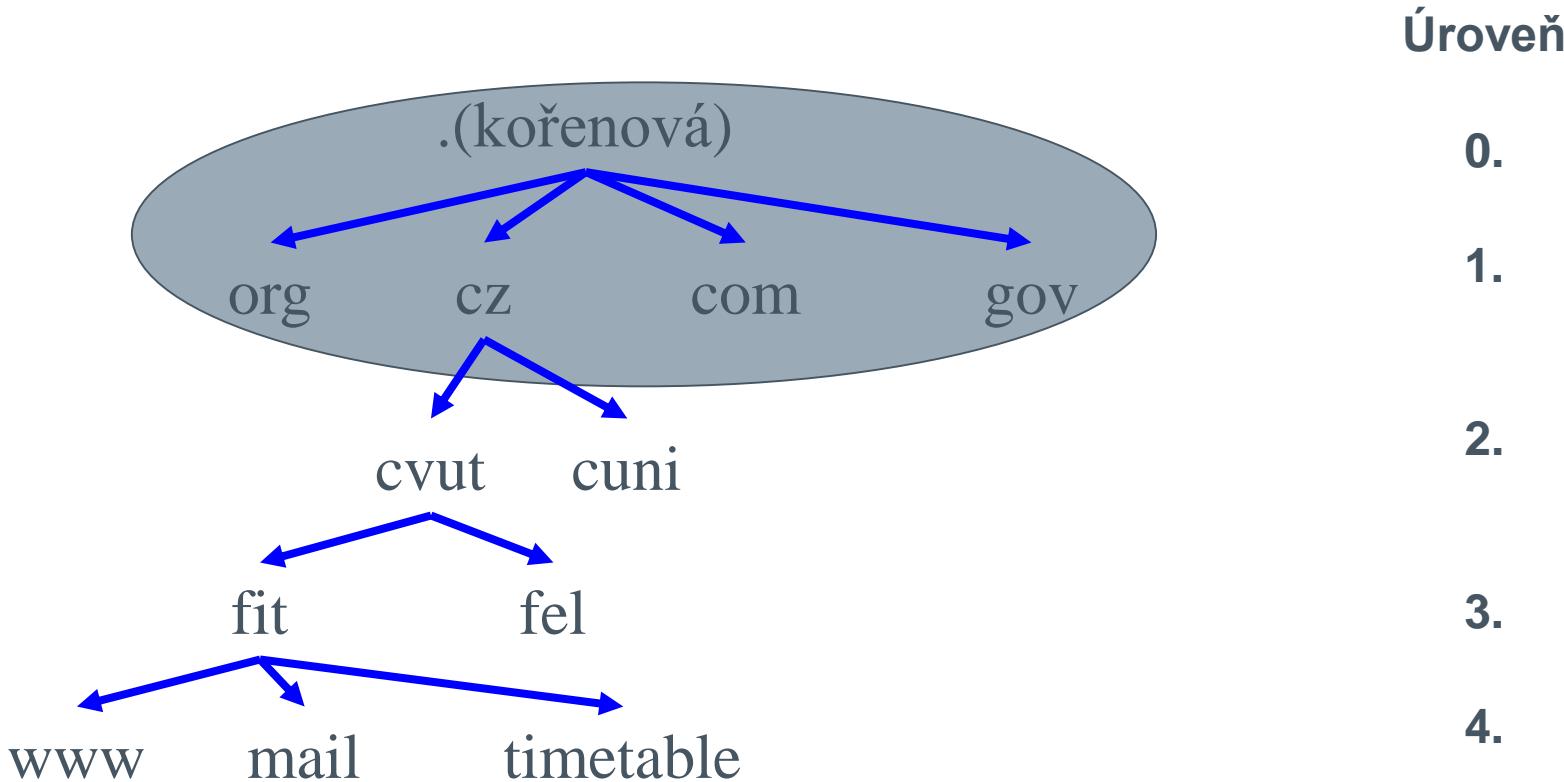
- Každý moderní operační systém obsahuje pro překlad doménových jmen **resolver**.
- **Resolver** = program, který komunikuje s **Name serverem** tak, že zasílá žádost o překlad doménového jména na odpovídající IP adresu.
- **Name Server (NS)** = počítač, který se stará o překlad doménových jmen na IP adresy.
- **DNS Server** = NS pracující v systému DNS.
- IP adresu DNS serveru lze nastavit **manuálně** či **dynamicky** - prostřednictvím **DHCP/DHCPv6**, **anebo bezstavové konfigurace** (viz. 4 přednáška, slajd 15).

# DNS architekura a jeho činnosti



- DNS = systém pro správu domén a doménových záznamů.
  - Doména = množina různých doménových záznamů sdružených pod společným doménovým jménem (např. cz, cvut.cz, fit.cvut.cz atd.).
  - Konkrétní typy záznamů pro danou doménu se nazývají **doménové zdrojové záznamy (DZZ)**.
  - Pro každou doménu je definovaný její hlavní DNS server, který její DZZ spravuje.
- DNS servery, které spravují DZZ pro konkrétní domény, se nazývají **autoritativní** DNS servery pro tyto domény.
- DNS provádí překlad doménových jmen :
  - Přímo = doménovému jménu se přiřadí odpovídající IP adresa, např. [www.fit.cvut.cz](http://www.fit.cvut.cz) → 147.32.232.212.
  - Reverzně = IP adrese se přiřadí odpovídající doménové jméno, např. 147.32.232.212 → pc-12.fit.cvut.cz.

# Hierarchie domén a doménových jmen v DNS



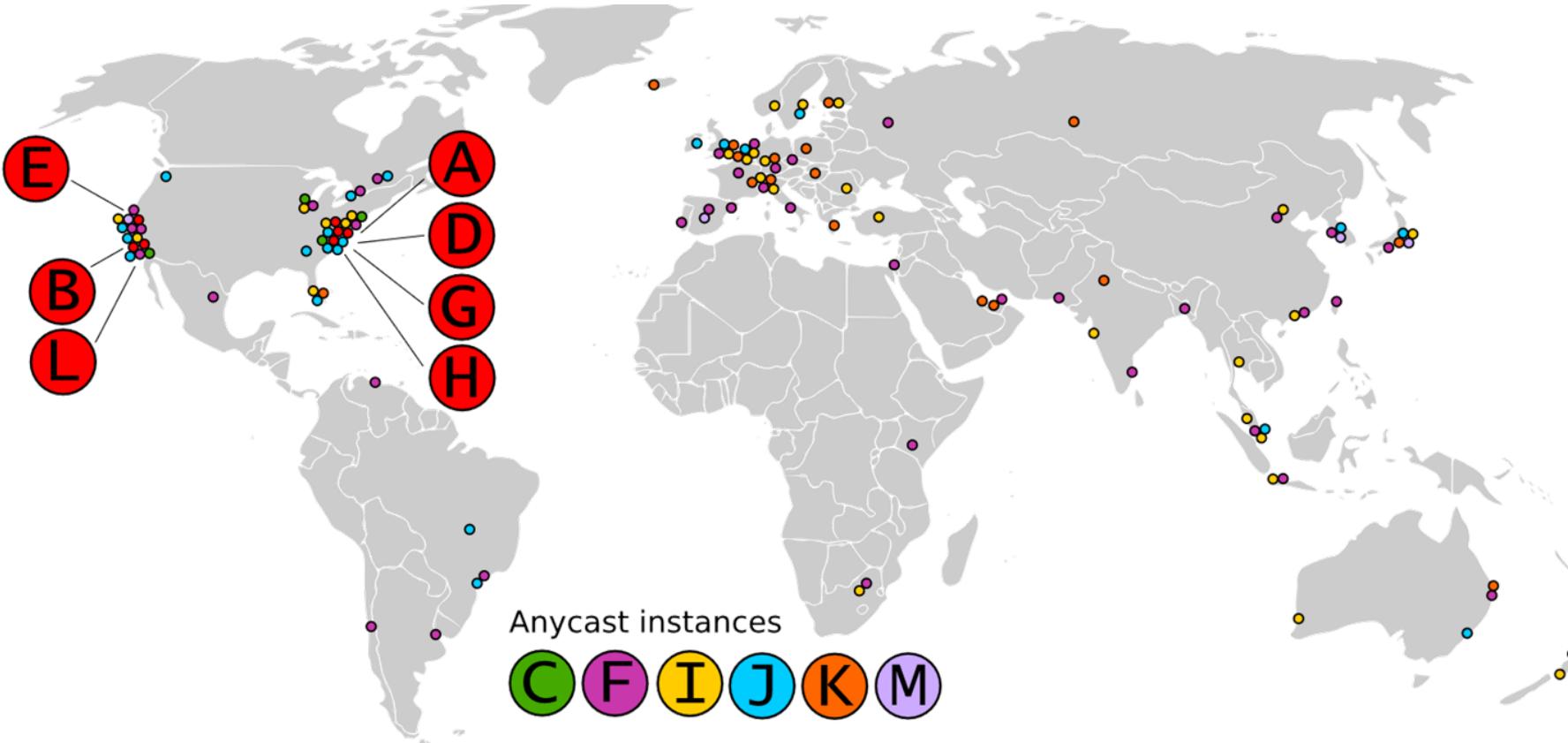
Systém doménových jmen vytváří stromovou strukturu.

Doménové jméno (Domain Name) = posloupnost jmen od listu ke kořenu. Např. **mail.fit.cvut.cz**.

Doména (Subdoména) = libovolný podstrom daného stromu. Např. **cz**, **cvut.cz**, **fit.cvut.cz**.

Hloubka zanoření ve stromu doménových jmen udává číselné označení úrovně domény.

# Kořenové DNS servery a jejich význam



Kořenové DNS servery = DNS servery 0. úrovně.

Virtuálně existuje 13 skupin (A..M) kořenových DNS serverů. DNS servery těchto skupin udržují informace o DNS serverech domén 1. úrovně. **13 skupin kořenových DNS serverů** je reálně tvořeno stovkami DNS serverů, které jsou přiřazeny konkrétním skupinám. Dotazy ke kořenovým DNS serverům se doručují pomocí **anycastu** - pro IPv6 nativně, pro IPv4 prostřednictvím protokolu **BGP**.

# Top-level domains (TLD) = domény 1. úrovně

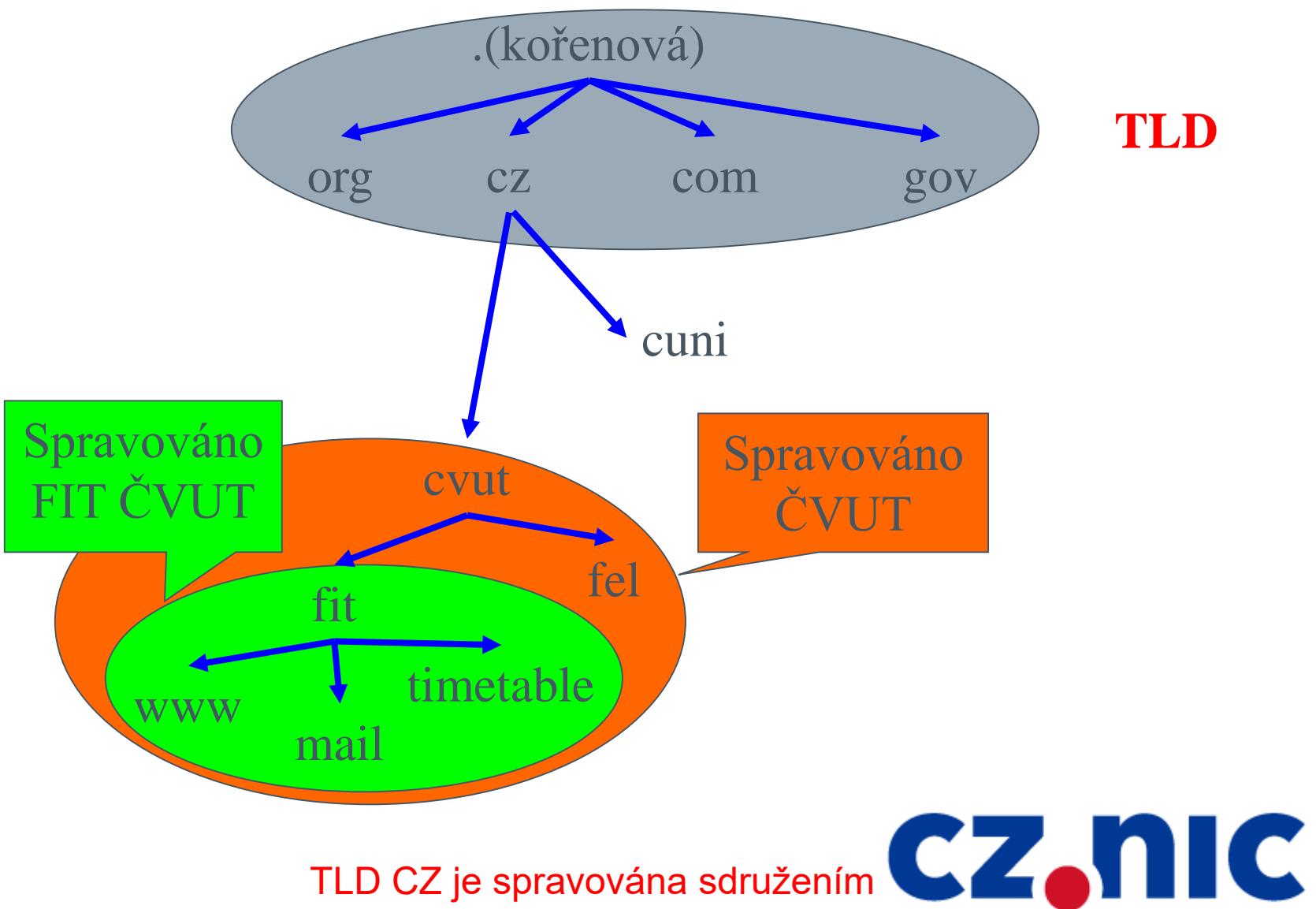


- Čtyři typy TLD domén:
  - **Generic Top Level Domains (gTLD)**
    - 3-znakový kód označuje funkci organizace používající tuto doménu.
    - Primárně použito v USA.
    - Např. gov, mil, edu, org, com, net.
  - **Country Code Top Level Domain (ccTLD)**
    - 2-znakový kód značí stát, ve kterém se typicky nachází servery uvedené v DZZ dané domény.
    - Např: cz, us, va, jp, de.
  - **New Generic Top Level Domains (ngTLD)**
    - Novinka od roku 2013.
    - Doménové jméno může být libovolný řetězec.
    - Označují takto např. města .paris, .london atd.
    - Jen společnost Google má ngTLD domén 101.
  - **Reverzní doména in-addr.arpa**
    - Určena pro převod IP adresy na doménové jméno (viz dále).

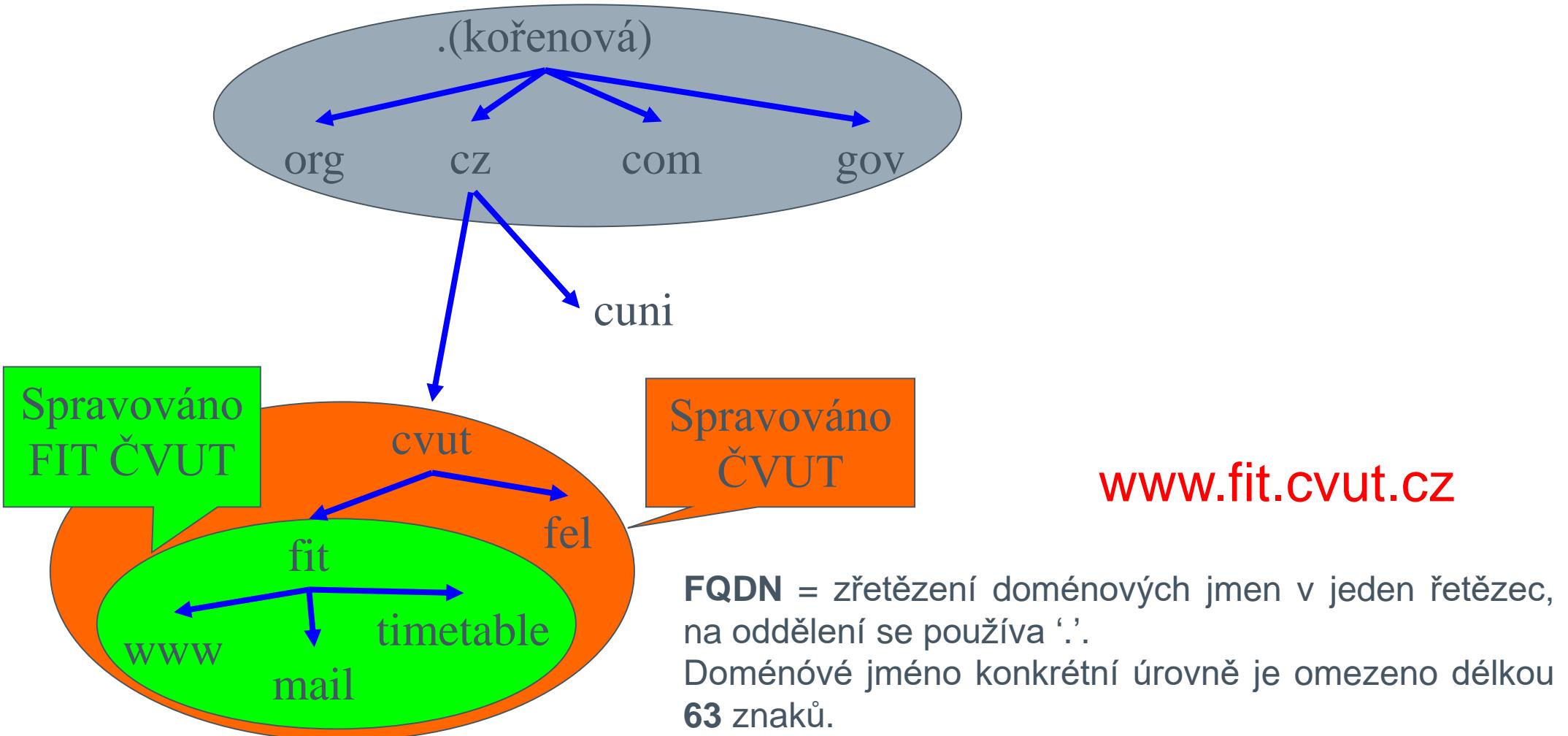
Aktuální statistiky říkají, že existuje cca. 1200 gTLD+ccTLD a zhruba 32 milionů ngTLD domén.

info převzato z <https://hostingtribunal.com/blog/tld-statistics/>

# Správa záznamů pro TLD CZ



# Plně kvalifikované doménové jméno Fully Qualified Domain Name (**FQDN**)



# Pojem doména a zóna v DNS



- Doména určité úrovně obsahuje **množinu doménových jmen** pro servery, které jsou pod jejím doménovým jménem dosažitelné.
- Např. pod doménou **fit.cvut.cz** existují servery **mail.fit.cvut.cz**, **www.fit.cvut.cz** atd.
- Množina jmen dané domény se dá rozdělit (ale nemusí se) na více částí (např. servery a\*-k\* a l\*-z\*).
- Zóna = konkrétní podmnožina množiny doménových jmen.
- DNS server, který se stará o překlad doménových jmen pro **konkrétní zónu**, se nazývá **autoritativní pro danou zónu**.
- Autoritativní DNS server může správu doménových jmen části zóny (**subzóny**) delegovat na jiný DNS server, např. kvůli snížení svéjí zátěže.

# Záznam zóny a jeho struktura



Úvodní záznam (SOA = Start of Authority)

Doménové zdrojové záznamy (DZZ)

- **Záznam zóny** = Úvodní záznam . {DZZ}\*
  - **Úvodní záznam** = záznam obsahující základní všeobecné informace o dané zóně (viz dále). Vyskytuje se pouze jeden.
  - **Doménový zdrojový záznam** = záznam obsahující konkrétní informace pro konkrétní doménové jméno (viz dále). Běžně se vyskytuje více vzájemně různých DZZ.

# Úvodní záznam (SOA) a jeho struktura - FYI



```
ZNAME IN SOA MNAME RNAME (  
SERIAL;  
REFRESH;  
RETRY;  
EXPIRE;  
MINIMUM;  
)  
;
```

**ZNAME** = doménové jméno zóny.

**MNAME** = doménové jméno autoritativního DNS serveru pro danou zónu.

**RNAME** = kontakt na správce zóny (@ se nahrazuje .).

**SERIAL** = sériové číslo zóny (SČZ), které udává aktuálnost záznamu (při změně se zvyšuje o 1).

**REFRESH** = počet vteřin, po jejichž uplynutí musí DNS servery, které SČZ načetly, načíst SČZ znovu.

**RETRY** = počet vteřin, po jejichž uplynutí se 1x provede opětovné načtení SČZ, pokud se předchozí načtení SČZ nepodařilo.

**EXPIRE** = počet vteřin, po jejichž uplynutí od načtení sériového čísla zóny lze libovolný DZZ považovat za neplatný.

**MINIMUM** = počet vteřin, který udává defaultní hodnotou životnosti DZZ, nemají-li ji implicitně v rámci záznamu uvedenu hodnotu TTL.

# Doménový zdrojový záznam v zóně



- Obecný tvar DZZ je (**Jméno**, **TTL**, **Třída**, **Typ** , **Hodnota**)
  - **Jméno** = doménové jméno daného serveru.
  - **TTL** = doba ve vteřinách, jak dlouho může být DZZ uložen v keši DNS serverů, po jejímž uplynutí musí být načten znova. Pokud TTL není v záznamu uvedena, bere se jako hodnota MINIMUM ze SOA záznamu.
  - **Třída** = rodina protokolů, k níž se záznam vztahuje (běžně IN, IN = Internet, jiné třídy sice existují, ale nepoužívají se).
  - **Typ** = určení druhu záznamu (viz dále).
  - **Hodnota** = IP adresa nebo doménové jméno.

# Vybrané typy DZZ



Typ	Význam
A	IPv4 adresa pro doménové jméno.
AAAA	IPv6 adresa pro doménové jméno.
NS	Autoritativní DNS server pro danou doménu
MX	Server, který zajišťuje doručování emailů pro doménu.
CNAME	Kanonické jméno, synonymum k doménovému jménu.
PTR	Reverzní záznam pro IP adresu (viz dále).
SRV	Server, který zajišťuje VoIP služby pro danou doménu.
TXT	Textový komentář.
CAA	Certifikační autority, které mohou vydávat certifikáty pro danou doménu
RRSIG	Veřejný klíč k ověření daného záznamu, tento používá záznam rozšíření DNSSEC (viz dále).

# Příklad záznamu domény mylab.com



```
mylab.com. IN SOA pc4.mylab.com.  
admin.mylab.com. (  
    1 ; SERIAL  
    28800 ; REFRESH  
    7200 ; RETRY  
    604800 ; EXPIRE  
    86400 ; MINIMUM  
)
```

```
mylab.com.      IN  NS      pc4.mylab.com.  
localhost.       IN  A       127.0.0.1  
pc4.mylab.com.  IN  A       10.0.1.41  
pc3.mylab.com.  IN  MX     10.0.1.31  
pc2.mylab.com.  IN  A       10.0.1.21  
pc1.mylab.com.  IN  CNAME   pc2.mylab.com.
```

pc4.mylab.com je **DNS** serverem pro doménu mylab.com.

pc3.mylab.com je **poštovním** serverem pro doménu mylab.com.

pc1.mylab.com je **alias** pro pc2.mylab.com.

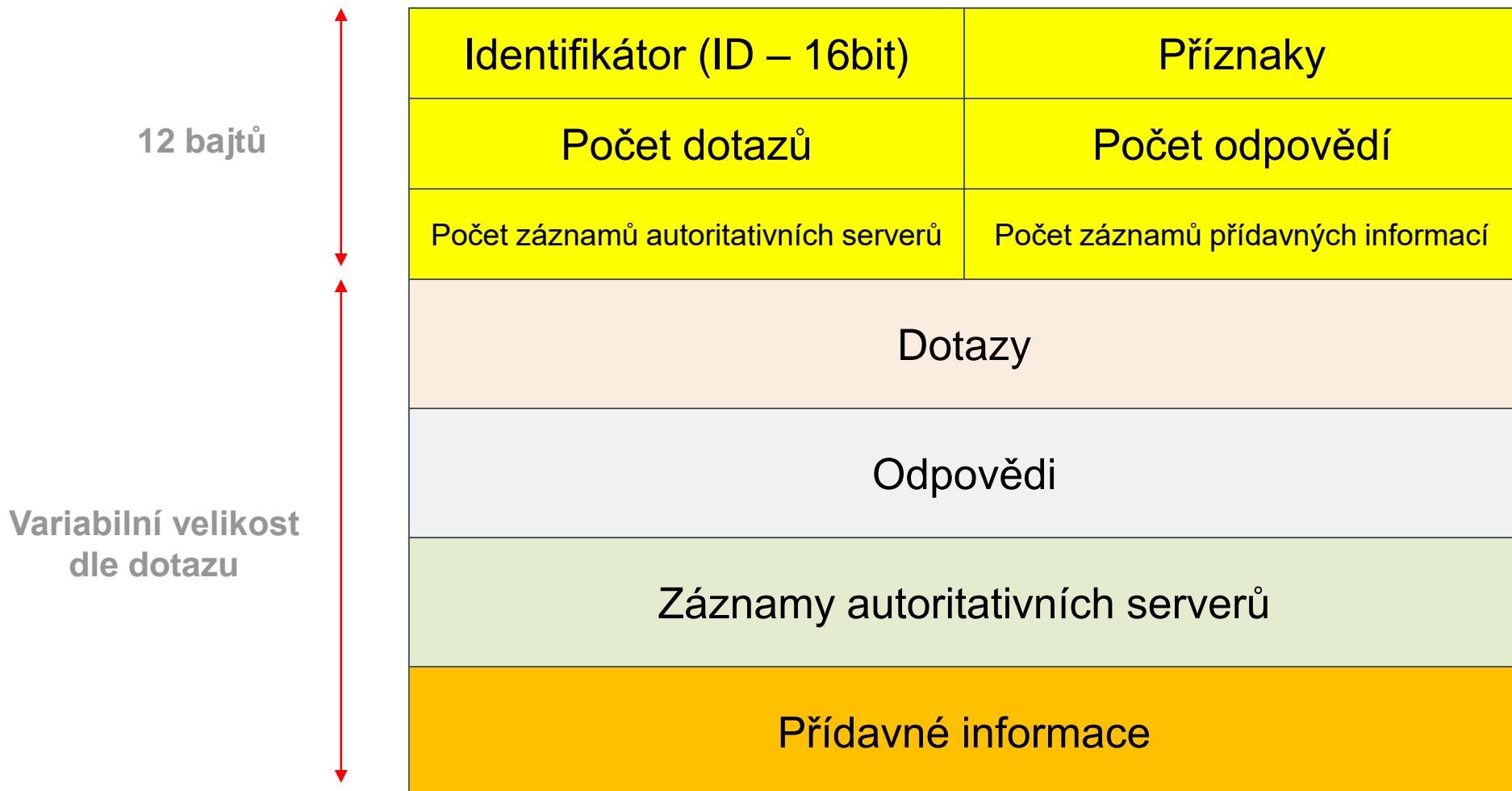
localhost, pc4.mylab.com a pc2.mylab.com jsou DZZ pro **konkrétní IP** adresy.

# DNS protokol



- DNS protokol = protokol, který slouží pro zasílání dotazů a odpovědí v systému DNS.
- Dotazy a odpovědi DNS protokolu je přenášeny v položce **DATA UDP** nebo **TCP paketu**. **Většina implementací** DNS protokolu používá protokol **UDP**.
- DNS protokol standardně používá port **53**.
- DNS protokol má pro dotazy/odpovědi **definovaný společný formát paketu** (viz dále), ve kterém se může současně přenášet jak více dotazů, tak i odpovědí.

# Obecný formát DNS dotazu/odpovědi



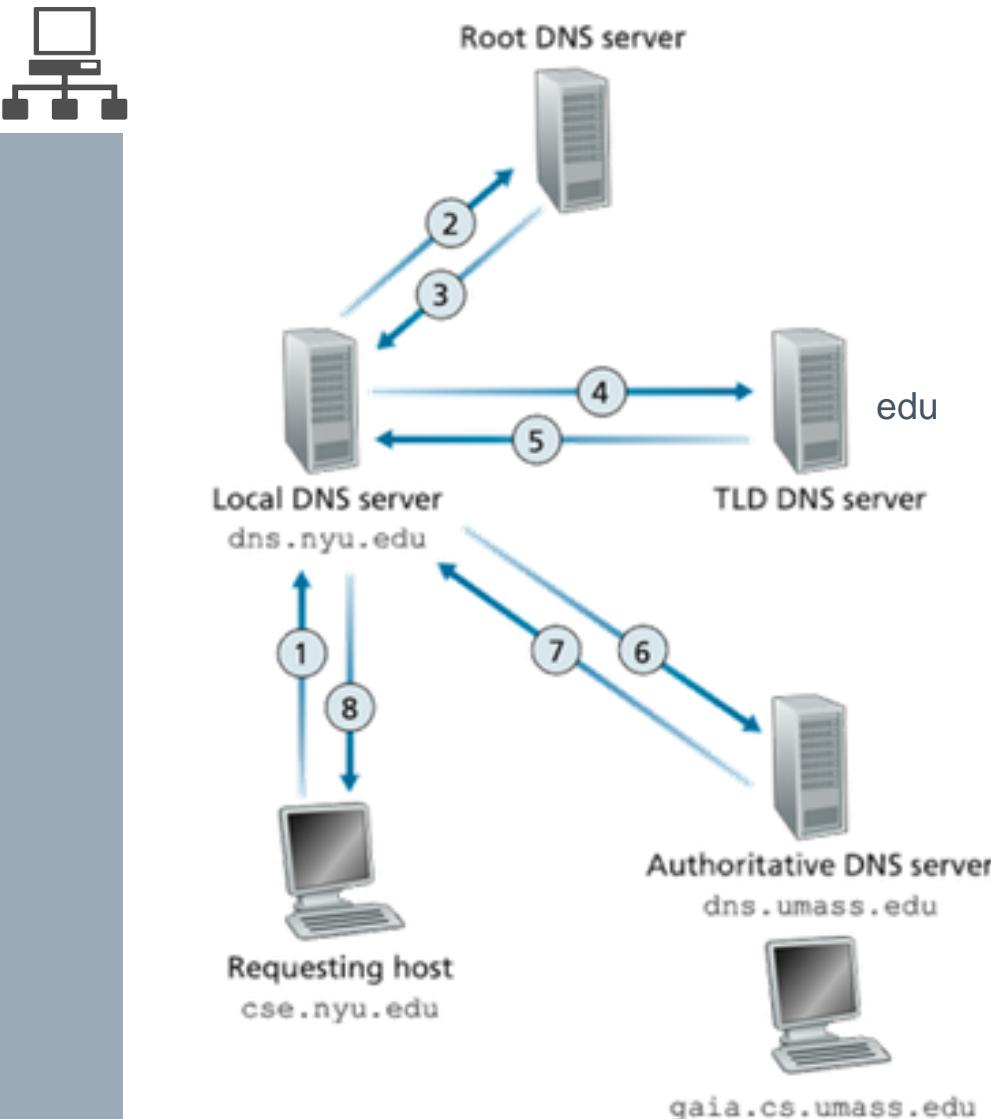
**Poznámka.** Detailní specifikace položek je nad rámec tohoto předmětu, zájemce odkazují na další informace uvedené v knize **Velký průvodce protokoly TCP/IP a systémem DNS od L. Dostálka**.

# Princip překladu jmen v DNS



- Původně existoval soubor **hosts.txt**, který obsahoval překlad všech doménových jmen ve formátu **{doménové jméno, IP}**. Tento soubor byl spravován centrální autoritou. Všechny DNS servery si tento soubor stahovaly a na základě jeho obsahu vracealy odpovědi.
- S **přibývajícím počtem záznamů** začal být soubor již **příliš velký**. Důsledkem tohoto bylo opuštění centralizovaného principu.
- Tento princip centrálního souboru s doménovými jmény je však podporován stále i moderními OS, např. v Linuxu v souboru: `/etc/hosts`.
- **Resolver** běžící v daném počítači používá pro překlad doménových jmen nejprve soubor s doménovými jmény. Pokud neuspěje, obrátí se na nadřazený DNS server.
- DNS server zasílá DNS dotazy distribuovaně, dvěma způsoby: **iterativně** anebo **rekurzivně**. Způsob zasílání dotazů, záleží na tom, jak je daný DNS server nakonfigurován.

# Iterativní způsob dotazování v DNS

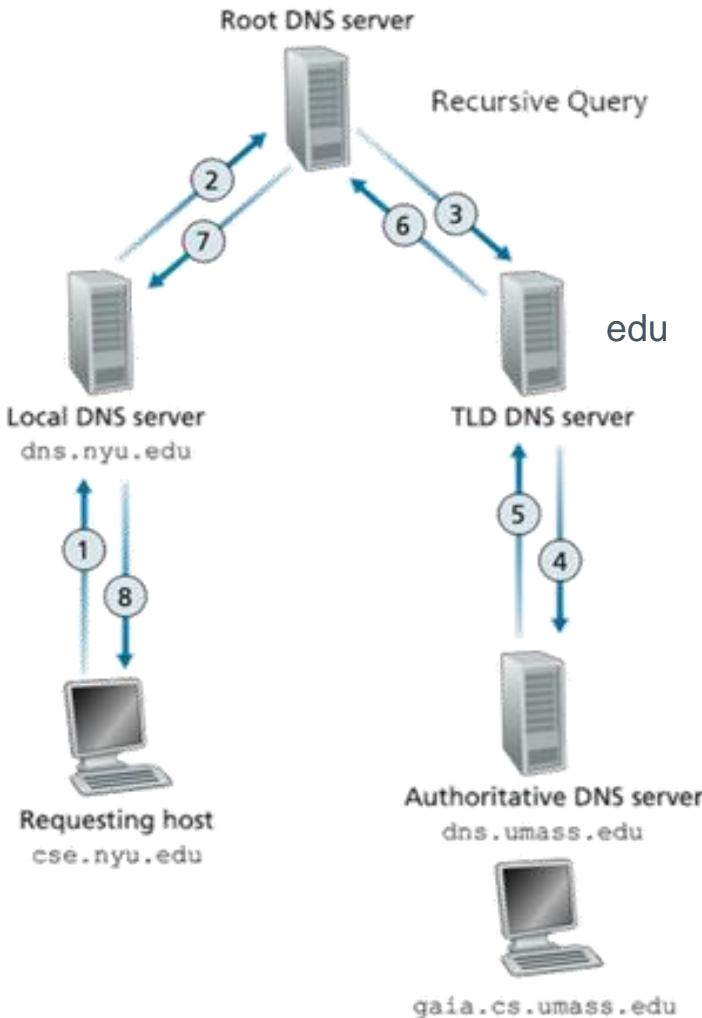


**Iterativní DNS server**, který překládá doménové jméno na IP adresu, se **ptá po krocích**. Začíná u kořenového DNS serveru a v každém kroku se posune o jednu úroveň níže.

- A. Resolver počítače, který potřebuje nalézt IP adresu pro doménové jméno vygeneruje dotaz nadřazenému DNS serveru X.
- B. X osloví kořenový DNS server (0. úroveň).
- C. Kořenový DNS server vrátí X informace o DNS serveru, který spravuje záznamy pro TLD (1.úroveň) dle hodnoty 1. řetězce doménového jména .
- D. Kroky B a C se opakují obecně tolikrát, dokud není dosažen záznam pro poslední řetězec zprava v doménovém jménu.
- E. X vrátí resolveru odpověď.

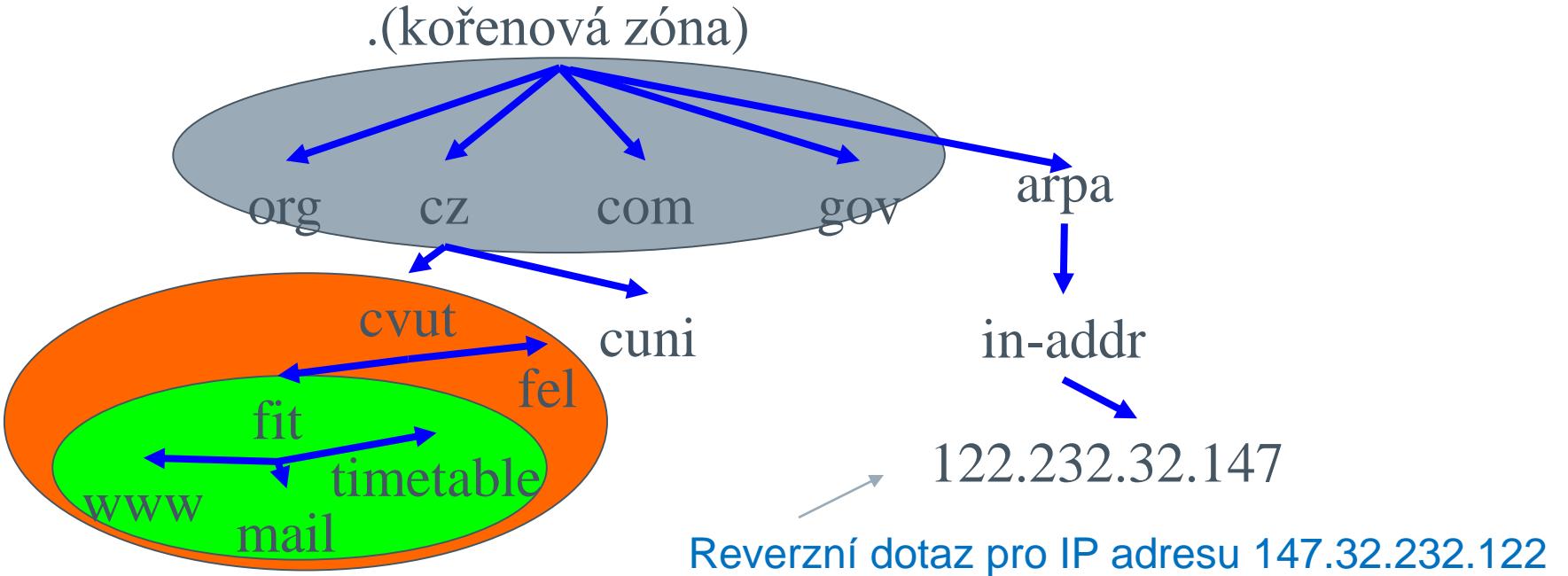
V příkladu na obrázku resolver počítače cse.nyu.edu požaduje překlad doménového jména 4. úrovně gaia.cs.umss.edu a obrátí se na svůj DNS server dns.nyu.edu.

# Rekurzivní způsob dotazování v DNS



- A. Resolver se ptá svého nadřazeného rekurzivního DNS serveru.
- B. Rekurzivní DNS (Y) se **ptá kořenového DNS serveru.**
- C. Kořenový DNS server dle 1. řetězce překládaného doménového jména přepošle dotaz DNS serveru 1 úroveň výše.
- D. Krok B se opakuje tolikrát, dokud není dosažen autoritativní DNS server úrovně, která odpovídá **počtu řetězců překládaného doménového jména.**
- E. Přeložená IP adresa související s doménovým jménem je vrácena postupně kořenovému DNS serveru.
- F. Kořenový DNS server vrátí Y přeloženou IP adresu.
- G. Y vrátí přeloženou IP adresu resloveru.

# Princip reverzních DNS dotazů

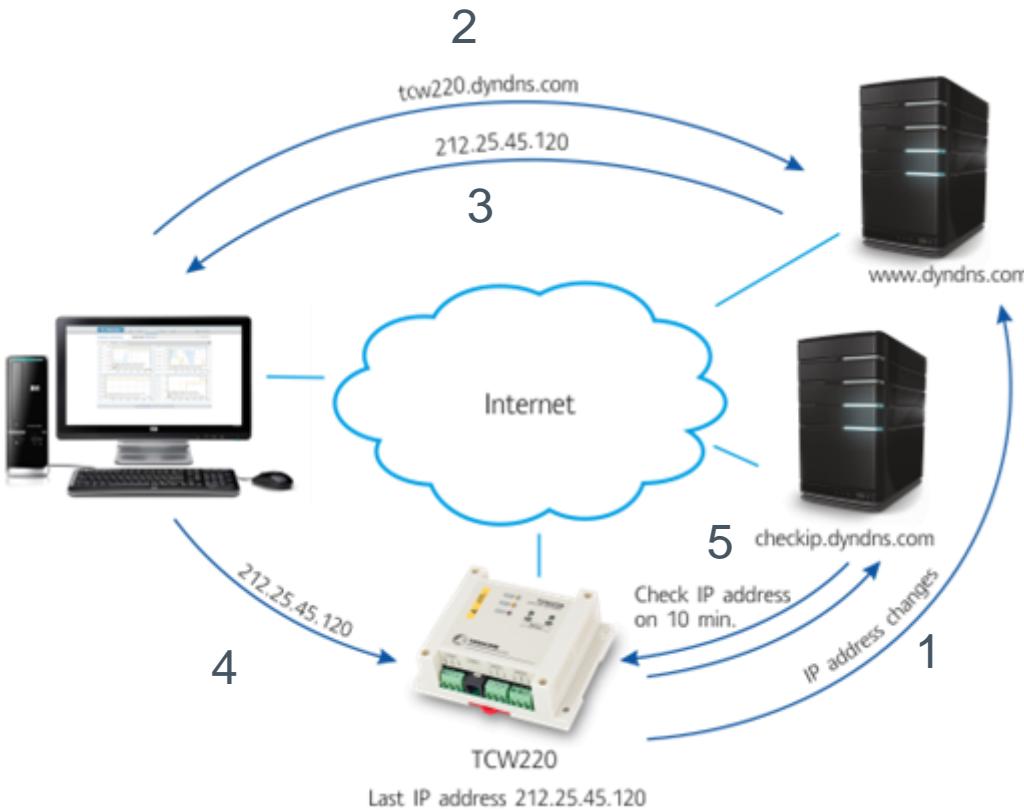


**Reverzní dotaz** = dotaz, který má za úkol pro danou IP adresu zjistit doménové jméno.

Reverzní dotaz pro IP **AA.BB.CC.DD** se zformuluje jako přímý dotaz pro doménové jméno **DD.CC.BB.AA.in-addr.arpa**.

Provedení dotazu probíhá **stejným způsobem jako u přímého dotazu**. Rozdíl je ale v tom, že finální reverzní překlad provede DNS server organizace (např. ČVUT), který má daný adresní rozsah na starosti. To, znamená že počet dotazů nemusí odpovídat počtu řetězců doménového jména, jak tomu bylo u přímého způsobu.

# Dynamické DNS (DynDNS)

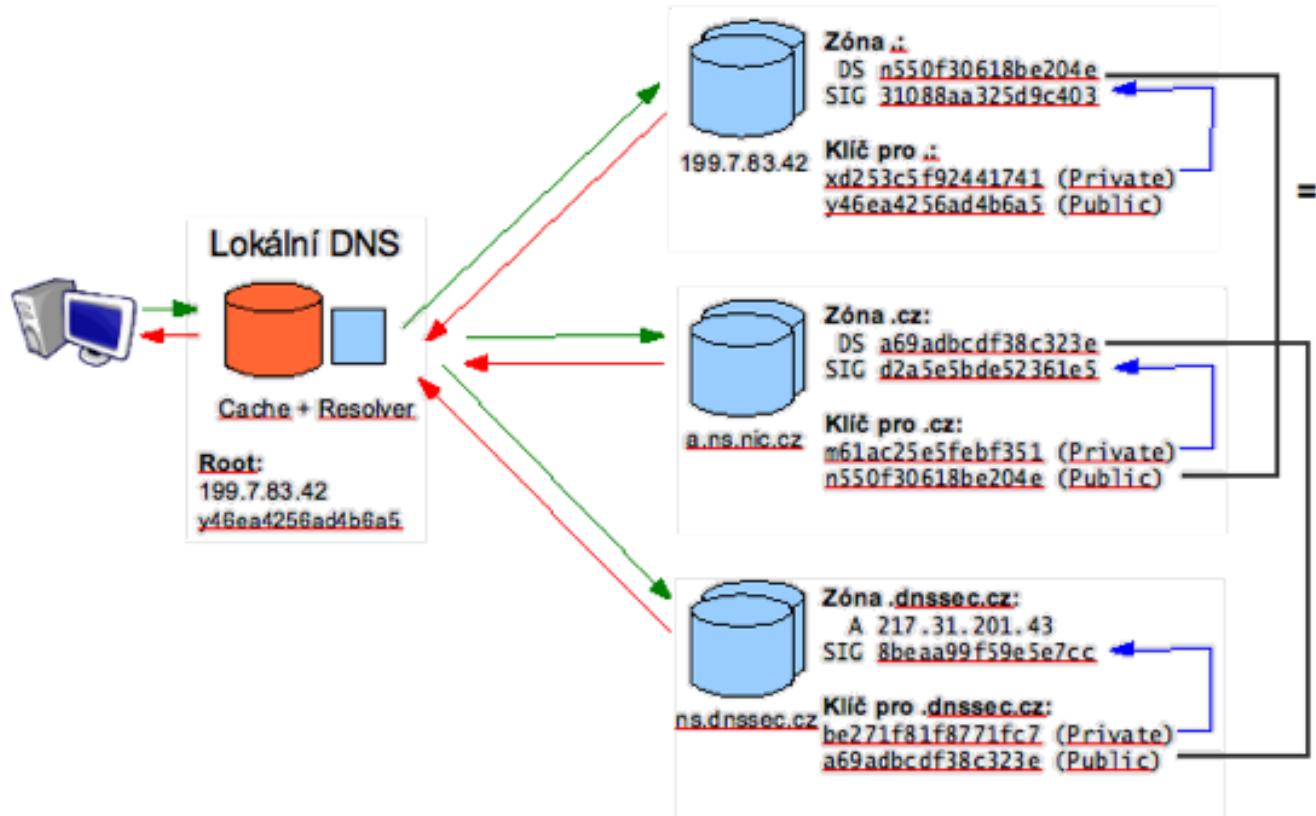


Zařízení, které často mění svoji IP adresu, nemůže klasické DNS použít, jelikož A záznamy může nastavovat pouze administrátor DNS serveru.

**DynDNS server** = DNS server, který dokáže A záznamy pro doménová jména měnit dle pokynů stanic za běhu.

1. Jakmile se zařízení (např. TCW220) připojí do sítě, nahlásí serveru (např. www.dyndns.com) spravující záznamy pro danou doménu (např. dyndns.com) svoje doménové jméno (např. tcw220.dyndns.com) a IP adresu (např. 212.25.45.120) a ten tyto informace uloží do příslušného A záznamu.
2. PC, která má nastaveno jako DynDNS server, (např. www.dyndns.com), vyšle dotaz na překlad doménové jména k tomuto serveru.
3. DynDNS server odpoví příslušnou nahlášenou IP adresou.
4. PC kontaktuje zařízení pod nahlášenou IP adresou.
5. Kontrola dostupnosti zařízení s danou IP adresou se pravidelně opakuje (např. každých 10 minut).

# DNSSEC – bezpečnější DNS



**DNSSEC = zabezpečené DNS, u kterého si pravost odpovědi lze ověřit .**

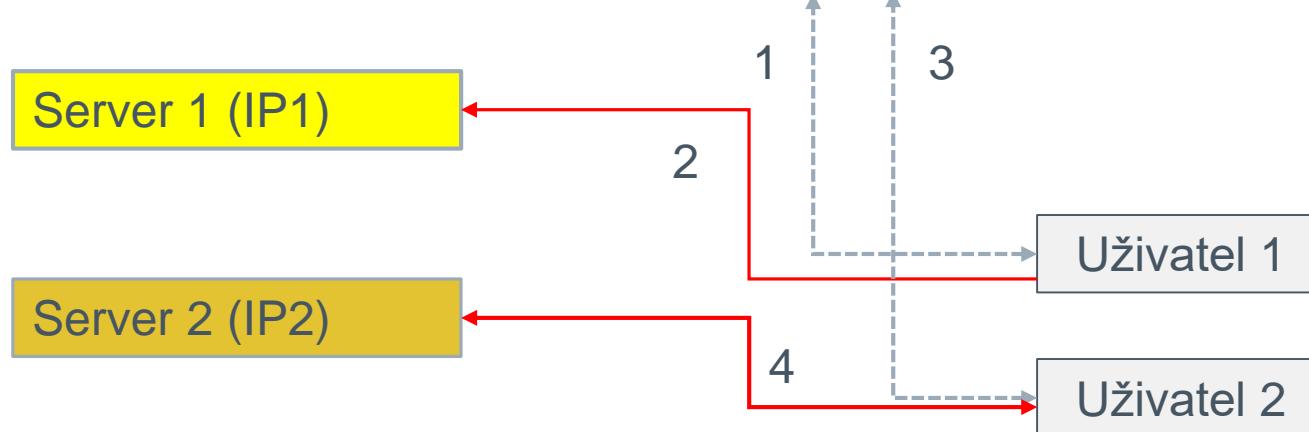
Založeno na **asymetrické kryptografii**. Každý DNS server podepisuje svoji odpověď svým soukromým klíčem. Pomocí veřejného klíče, který je uložený v RRSIG záznamu na DNS serveru nadřazené domény, si je možné pravost odpovědi ověřit.

# DNS a rozdělování zátěže (load balancing)



Servery provozující  
**www.mojesluzba.cz**

Autoritativní DNS  
pro doménu  
**mojesluzba.cz**

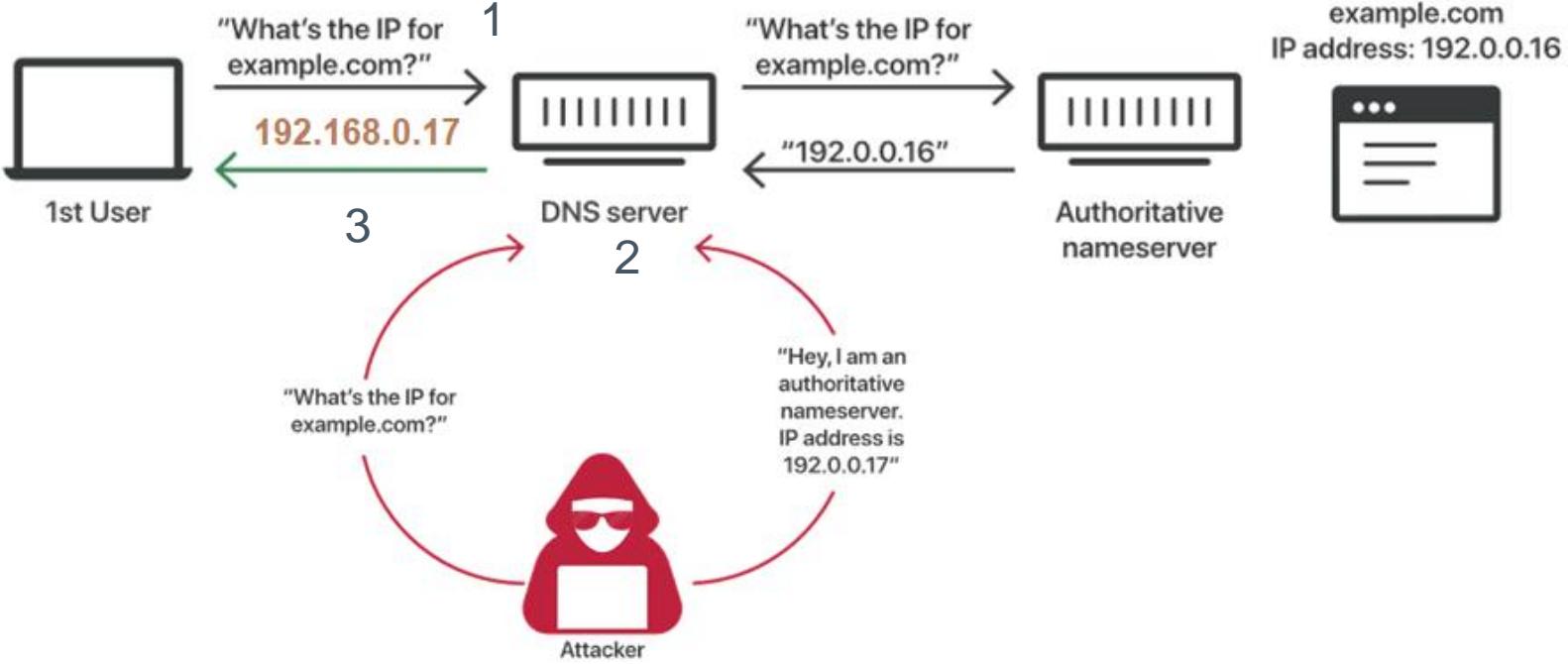


Rozdělování zátěže lze použít např. pro webové služby.

Princip spočívá v tom, že při opakovaném dotazu na stejné doménové jméno vrací autoritativní DNS server pokaždé jinou IP adresu.

1. Uživatel 1 se prostřednictvím svého DNS serveru dotáže na IP adresu, která odpovídá A záznamu doménového jména [www.mojesluzba.cz](#), v odpovědi je uvedena IP1.
2. Uživatel 1 sestaví spojení na adresu IP1.
3. Uživatel 2 se prostřednictvím svého DNS serveru dotáže na IP adresu, která odpovídá A záznamu doménového jména [www.mojesluzba.cz](#), v odpovědi je uvedena IP2.
4. Uživatel 2 sestaví spojení na adresu IP2.

# Útok na DNS – Cache Poisoning



DNS používá UDP protokol. Každý DNS dotaz je identifikovaný konkrétním 16ti bitovým identifikátorem (ID).

1. Uživatel se dotáže DNS serveru na překlad doménového jména, DNS server zjišťuje překlad pro doménové jméno distribuovaným způsobem.
2. V tuto samou chvíli, kdy DNS server zjišťuje překlad, útočník vygeneruje odpověď, která obsahuje totožné ID odpovědi a podvrženou IP adresu.
3. Jelikož odpověď od útočníka přijde dříve, než od autoritativního DNS serveru, DNS server útočníkovu odpověď přepošle uživateli a uloží si ji do své keš paměti.

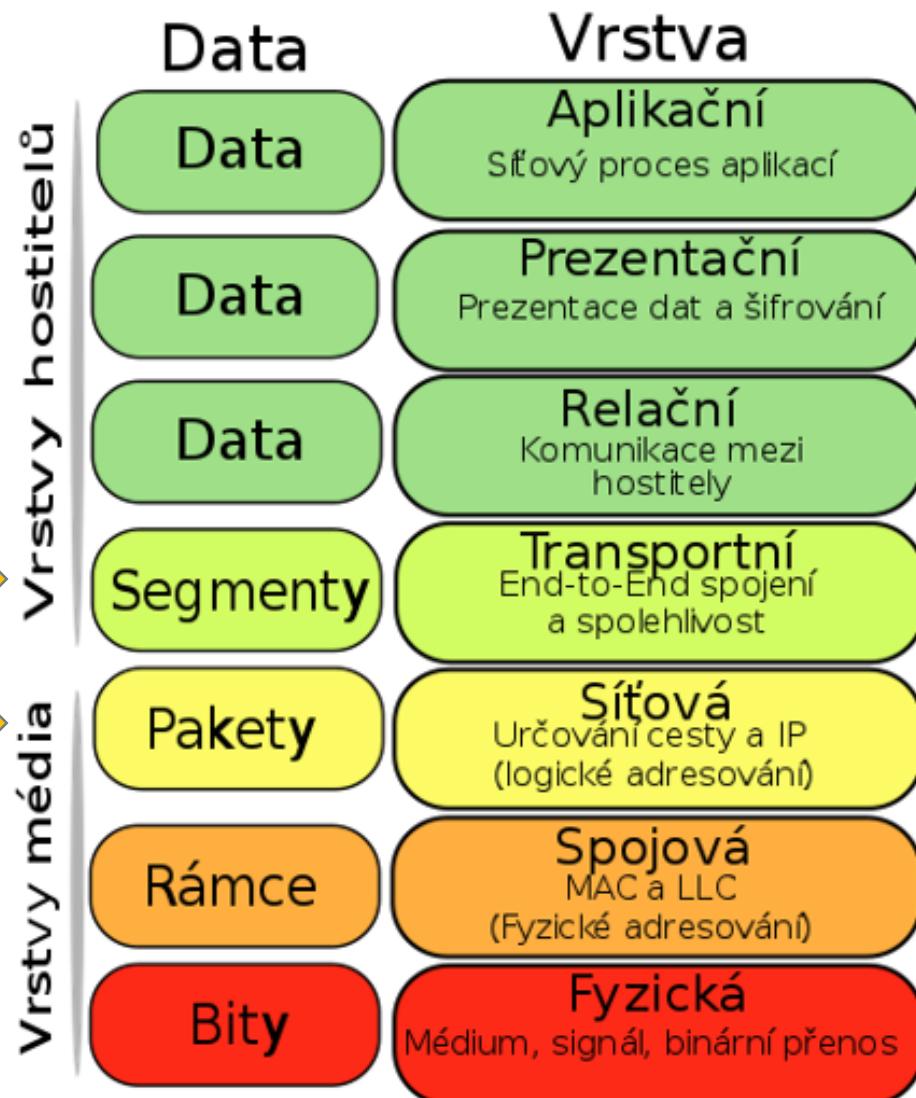
**Obranou je použití DNSSEC.**

# Počítačové sítě

Přednáška č. 8 - Bezpečnost v počítačových sítích



# Bezpečnost v Internetu - motivace



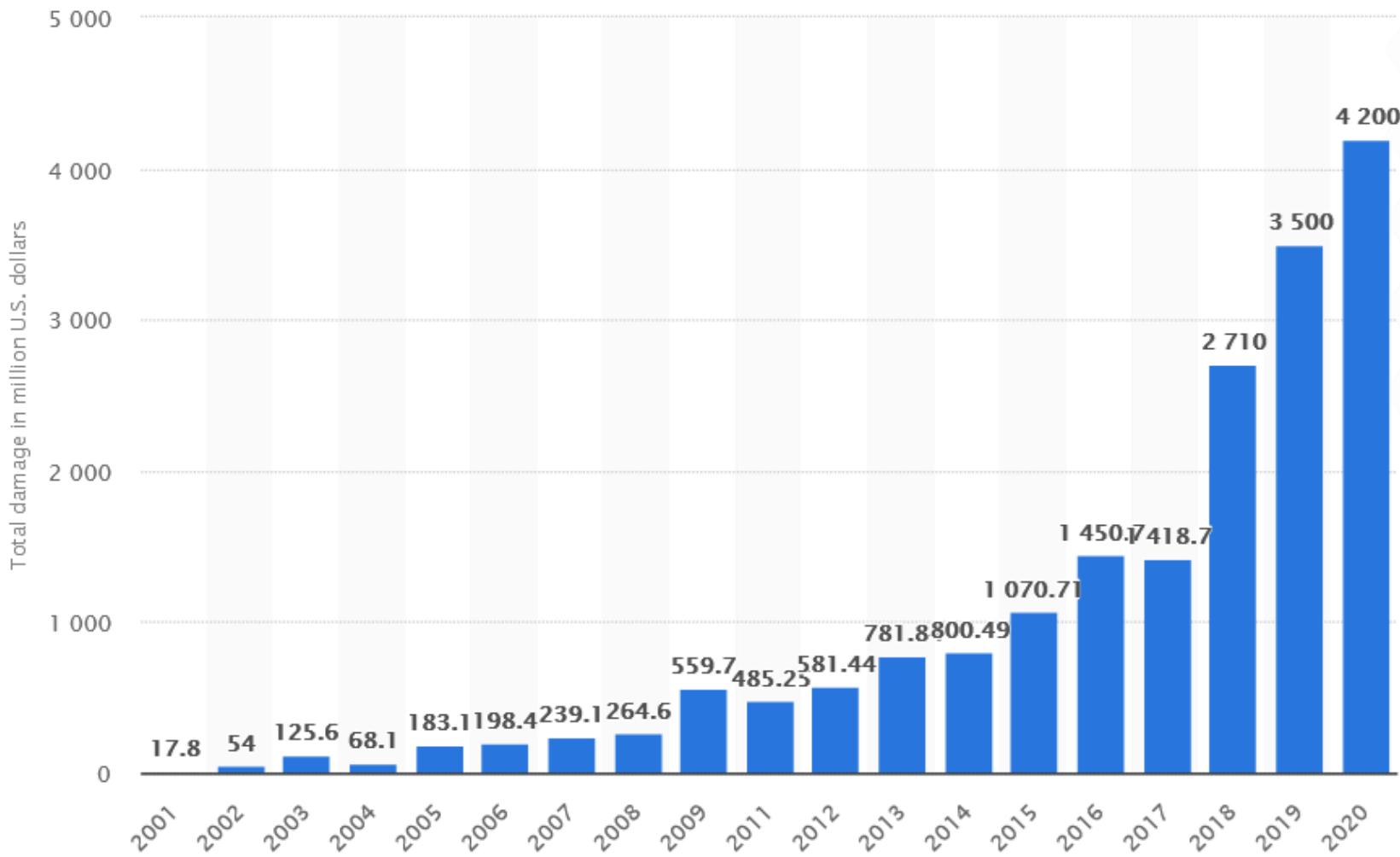
- Internet dnes zpřístupňuje obrovské množství služeb, které **využívá obrovské množství lidí**.
- Velké množství služeb a uživatelů souvisí s počtem útoků, které jsou denně prováděny útočníky (hackery).
- Útočník** = uživatel Internetu, který sám nebo prostřednictvím speciálních programů či hardwaru provádí útoky vůči jiným uživatelům (např. proniknutí do systému, zcizení dat, podvržení identity atd.)
- Firewall** = zařízení, které má za úkol chránit uživatele Internetu před útočníky.
- IPS/IDS** = zařízení, které má za úkol odhalovat resp. eliminovat síťové bezpečnostní hrozby (viz dále).
- Honeypoty** = nástraha na útočníka, která zjišťuje informace o hackery prováděných útocích. Nashromažděné informace jsou dále použity pro eliminaci budoucích útoků.
- S rozvojem IoT, IPv6 a autonomních vozidel atd. je situace stále složitější.**

# Osnova přednášky



- **Obecné pojmy z oblasti bezpečnosti počítačových sítí**
- **Technologie pro zabezpečení počítačových sítí**
  - Firewall
  - IPS/IDS systém
  - Honeypot
  - Pračky síťového provozu
- **Vybrané útoky útoky v počítačových sítích**

# Přehled škod způsobených kyberterorismem

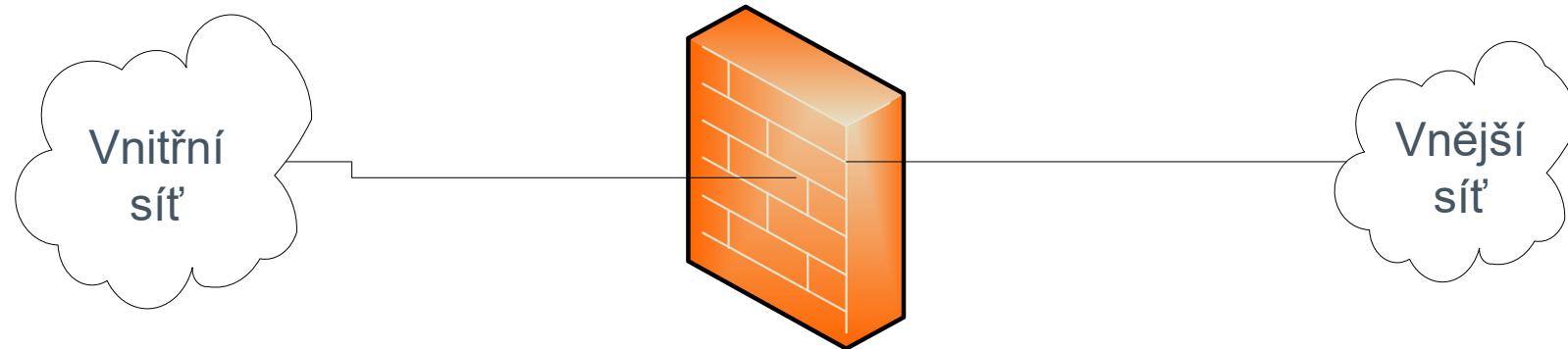


Zdroj: Převzato z <https://www.statista.com/statistics/267132/total-damage-caused-by-cyber-crime-in-the-us/>.

# Firewall

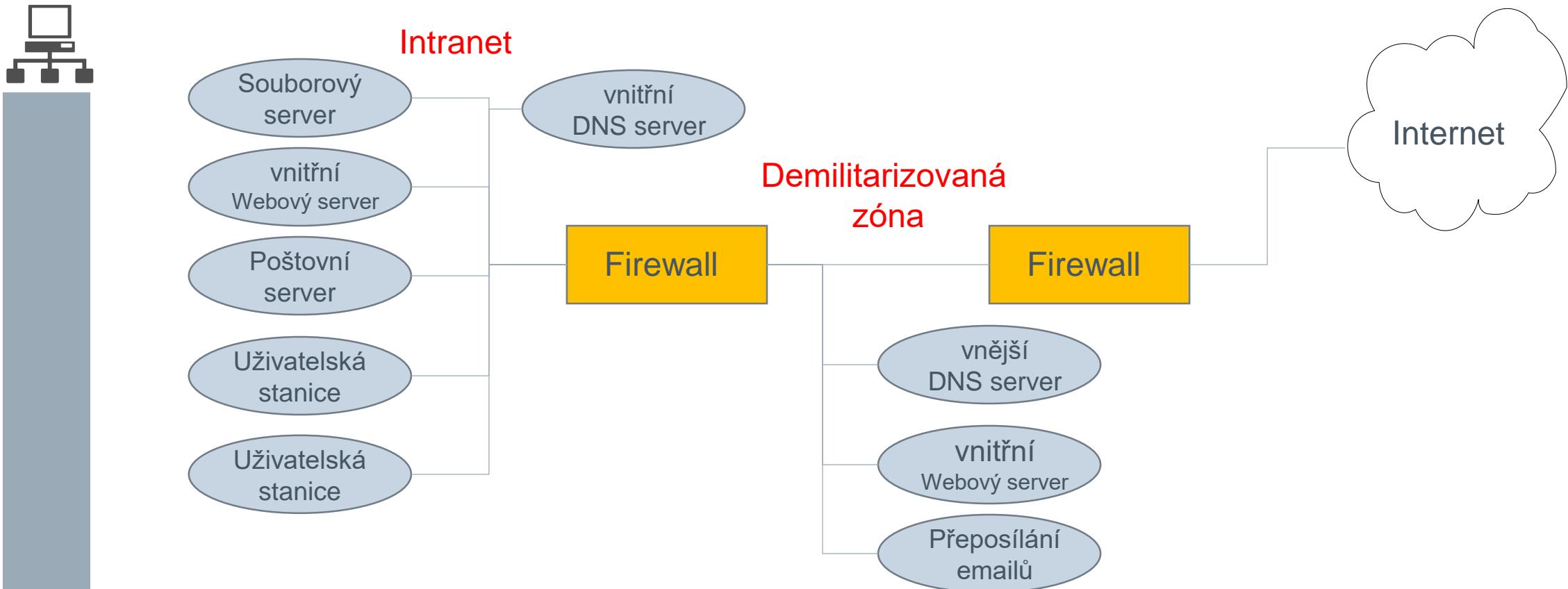


**Firewall** = zařízení, které kontroluje, propouští, filtruje či upravuje síťový provoz, který přichází na vstupní síťové rozhraní. Konkrétní činnost firewallu pro specifický druh síťového provozu je definována pomocí pravidel firewallu.



Firewall funguje jako bariéra, která odděluje důvěryhodný síťový provoz vnitřní sítě od nedůvěryhodného síťového provozu pocházejícího z vnější sítě resp. Internetu.

# Běžná architektura firemní sítě



**Demilitarizovaná zóna (DMZ)** = část počítačové sítě, která je částečně nebo plně přístupná z vnější sítě (Internetu). Umíšťují se do ní servery, které zpřístupňují určitou službu libovolným uživatelům, např. webové stránky či přeposílání pošty.

**Intranet** = vnitřní část sítě, která je přístupná pouze oprávněným uživatelům, např. server pro sdílení souborů, neveřejné webové stránky atd.

# Osnova přednášky



- Obecné pojmy z oblasti bezpečnosti počítačových sítí
- Tehnologie pro zabezpečení počítačových sítí
  - Firewall
  - IPS/IDS
  - Honeypot
- Vybrané útoky v počítačových sítích



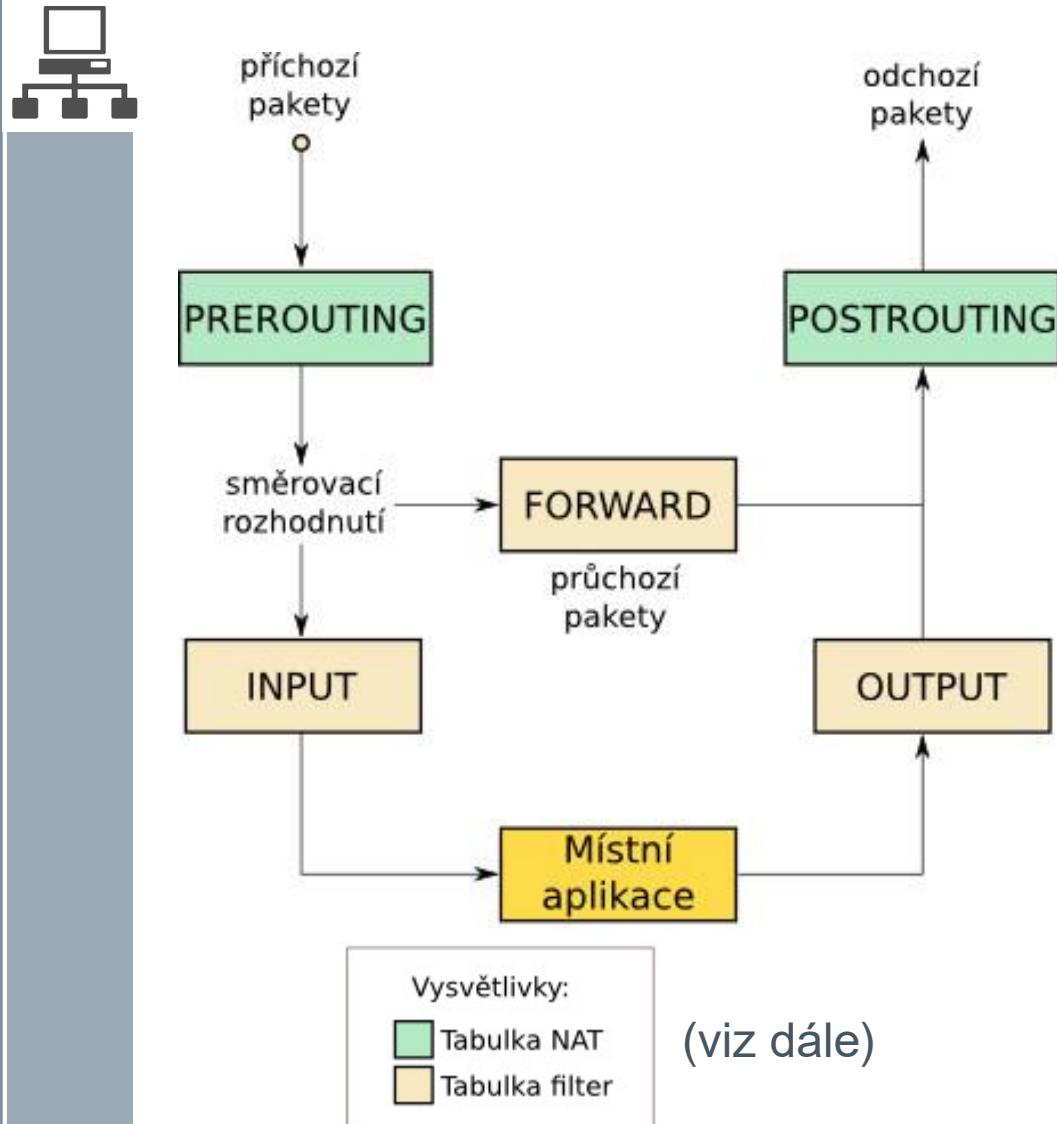
- **Paketové**
  - Pracují s informacemi **sítové** vrstvy.
  - Vyhodnocují sítový provoz na základě zdrojové a cílové IP adresy paketu.
  - Pokročilejší firewally provádějí i detailní hloubkovou analýzu hlaviček a dat jednotlivých paketů, tzv. **Deep Packet Inspection (DPI)**.
    - DPI není použitelné u **kryptovaných** protokolů, např. HTTPS.
    - Nešifrované protokoly lze identifikovat dle specifických řetězců, viz slajd 12.
- **Stavové**
  - Pracují s informacemi **transportní** vrstvy.
  - Jejich konfigurace je složitější, vyžaduje znalost konkrétních protokolů transportní vrstvy.
  - Propustnosti jsou srovnatelné s paketovými firewally.
  - V pokročilejších bezpečnostních systémech se používají v kombinaci s **Intrusion Prevention Systémy (IPS)**, viz slajd 15.
- **Aplikační = proxy**
  - Zařízení či programy, které kontrolují a přeposílají provoz mezi zdrojovou stanicí a cílovým serverem.
  - Veškerá komunikace probíhá prostřednictvím protokolů **aplikaci** vrstvy OSI modelu, např. HTTP, FTP (viz přednáška 10).

# Činnost paketového firewallu



- Paketový firewall analyzuje veškeré pakety, které přicházejí na konkrétní síťové rozhraní či odcházejí z některého síťové rozhraní daného počítače.
- Firewally provádějí s pakety na základě předem daných pravidel specifické akce, jako např. **povolení**, **zakázání**, **úprava položek v hlavičce paketu** či **logování**.
- Firewally v různých operačních systémech fungují na podobných principech, avšak způsob jejich konfigurace se zásadně liší, např. vytvářením pravidel pro práci s pakety.
- **Obecné pravidlo pro správce firewallů: „Povol to nejnужnější a zbytek zakáz!“**
- Známý linuxový paketový firewall se nazývá **iptables** (novější verze se označuje jako **nftables**).
- iptables rozlišují několik **úrovní (chains)** práce s pakety podle cílové IP adresy paketu (viz dále).
- Pro každou úroveň je možné v **tabulkách pravidel (tables of rules)** definovat záznamy, které přikazují firewallu provést s konkrétním paketem specifickou akci.

# iptables – úrovňě pravidel práce s pakety



**PREROUTING** = úroveň týkající se všech paketů před provedením směrovacího rozhodnutí na základě cílové IP adresy paketu.

**POSTROUTING** = úroveň týkající se všech paketů po provedení směrovacího rozhodnutí na základě cílové IP adresy paketu.

**FORWARD** = úroveň týkající se pouze paketů, které jsou přeposílány dále.

**INPUT** = úroveň týkající se pouze paketů, které je určeny místním aplikacím.

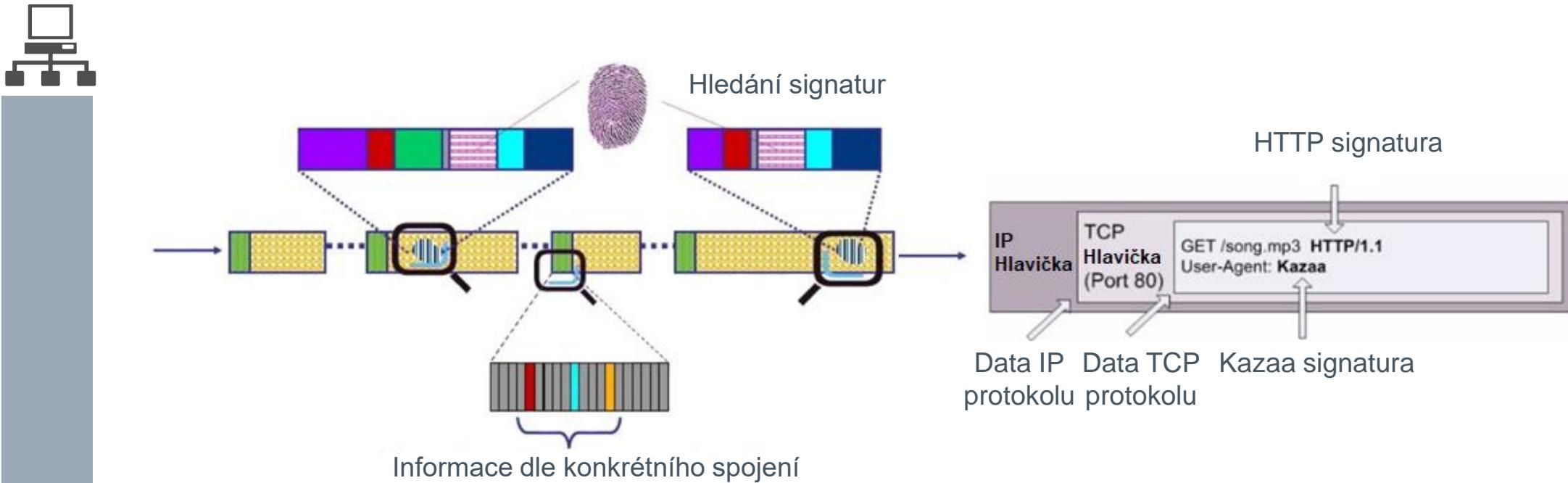
**OUTPUT** = úroveň týkající se pouze paketů, které jsou generovány místními aplikacemi.

# iptables – tabulky pravidel



- Každá tabulka pravidel je definována pro konkrétní úrovně práce (jednu nebo více) s daným paketem.
- iptables rozlišují **tři základní tabulky** pravidel:
  - **filter** = povolení či zakázání doručení paketu.
  - **nat** = překlad IP adres či portů v hlavičce paketu.
  - **mangle** = úpravy položek v hlavičce paketu pro další zpracování.
- iptables je možné rozšiřovat o přídavné **moduly**, které dokáží provádět složitější specifické akce, jako např. označování paketů (mark), logování paketů (log) či identifikaci použitého nešifrovaného protokolu aplikacní vrstvy (l7filter).
- Pro každý modul jsou definovány úroveň a tabulka pravidel, v rámci kterých je možné daný modul používat.

# DEEP Packet Inspection (DPI)

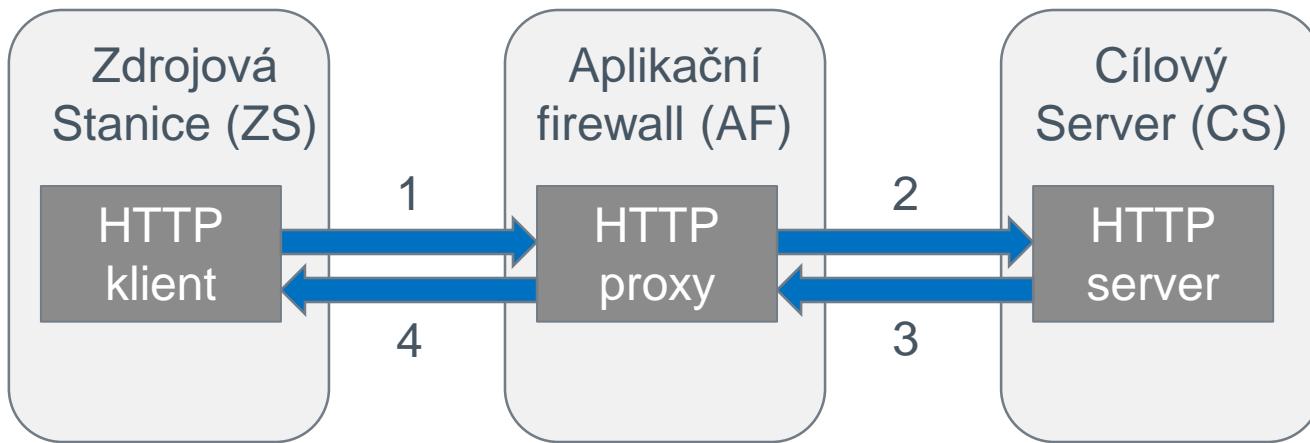


- 1) Řadu protokolů je možné identifikovat na základě specifického formátu hlavičky a obsahu dat.
- 2) Firewally provádějící DPI mají pro vybrané protokoly definované tzv. **signatury**, na základě kterých protokoly rozpoznávají.
- 3) **Signatury** = specifické **jednoznačné řetězce (otisky)** znaků umožňující s vysokou pravděpodobností identifikovat hlavičky konkrétních protokolů.
- 4) Rozpoznání protokolu umožňuje následné provedení dalších akcí.

# Aplikační firewall (proxy)



**Důvod použití:** skrytí zdrojové IP adresy, snížení množství odchozího provozu (ukládáním odpovědí od cílových serverů do keš paměti).

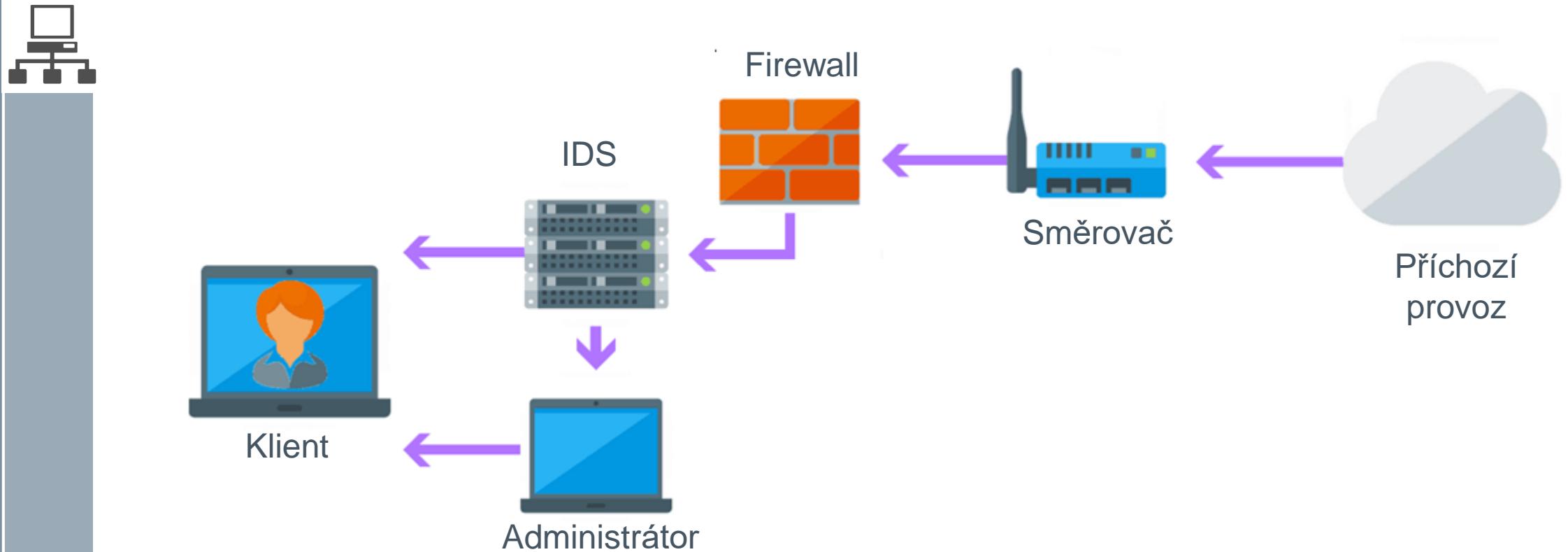


**Aplikační firewall** = firewall komunikující prostřednictvím specifického aplikačního protokolu.

**Fáze činnosti:**

1. Zdrojová stanice (ZS) se obrátí prostřednictvím aplikačního protokolu na aplikační firewall (AF).
2. AF zkopíruje požadavek, jako původce požadavku nastaví sebe a kontaktuje cílový server (CS).
3. CS vrátí odpověď AF.
4. AF přepošle odpověď ZS.

# Intrusion Detection System (IDS)

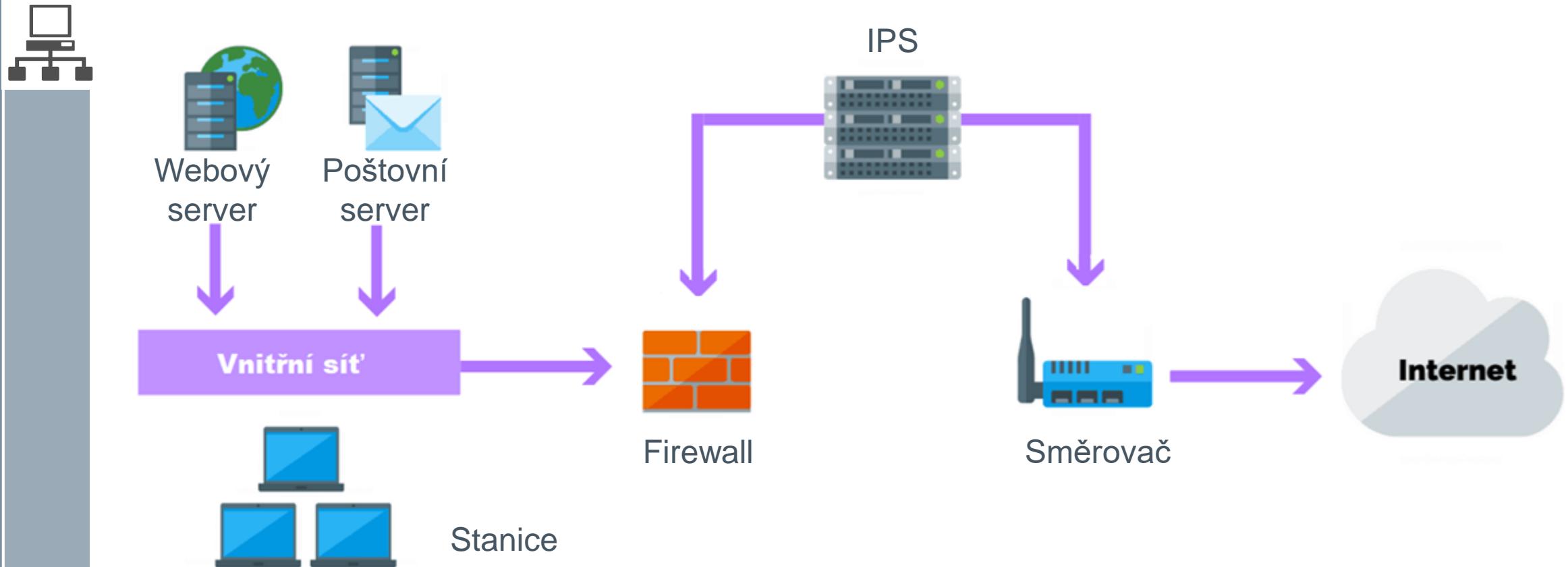


**IDS** = systém pro detekci hrozeb.

**Fáze činnosti:**

1. IDS Systém provádí analýzu **příchozího provozu** a detekci hrozeb (**intrusion = narušení**).
2. Pokud je detekována hrozba, informuje administrátora, který provede opatření pro eliminaci hrozby.

# Intrusion Prevention System (IPS)

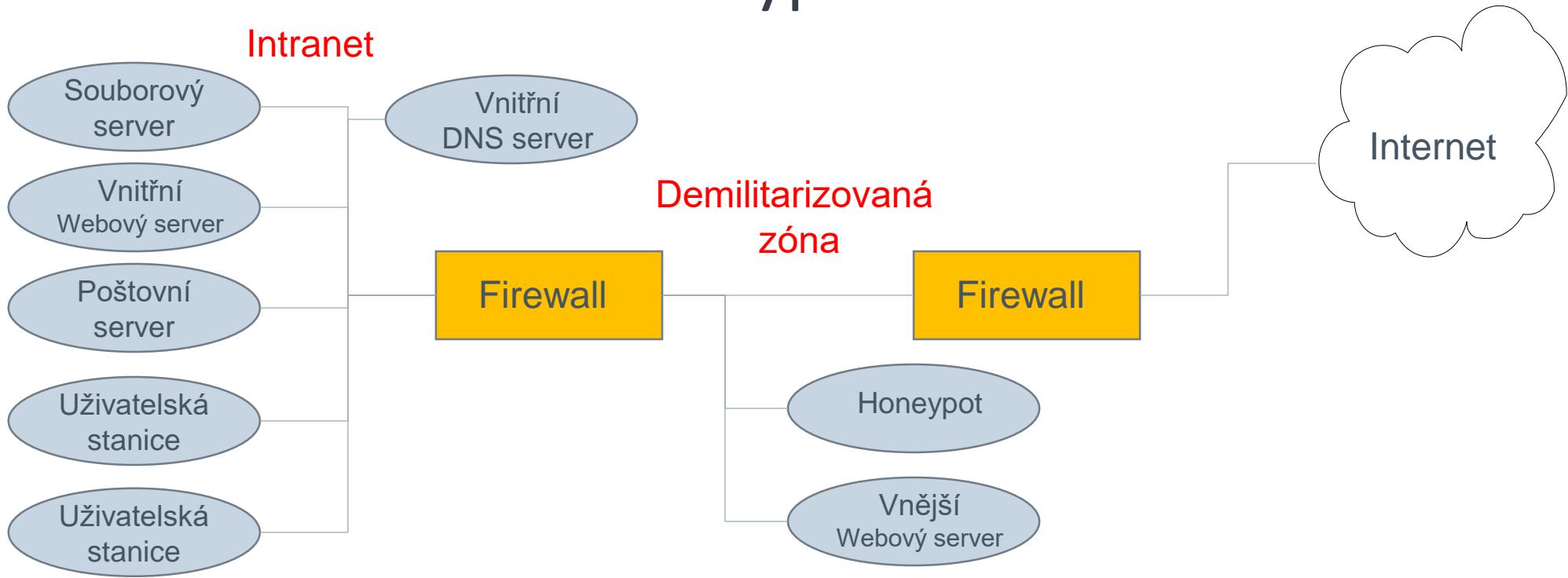


**IPS** = systém pro detekci a eliminaci hrozeb.

Fáze činnosti:

1. IPS Systém provádí analýzu **příchozího i odchozího** provozu a detekci hrozob.
2. Pokud je zjištěna hrozba, IPS zašle pokyn **Firewallu s požadavkem na aktivaci či vygenerování pravidla**, kterým Firewall provede např. filtraci všech paketů podezřelého provozu.

# Honeypot



**Honeypot** = počítač, který slouží jako návnada na útočníka za účelem získání informací o něm.

**Fáze činnosti:**

1. Do demilitarizované zóny se umístí honeypot. Honeypot vytváří dojem, že provozuje nějakou specifickou síťovou službu, jako např. emails či web.
2. Útočník, který honeypot objeví, se jej snaží napadnout.
3. Honeypot o útočníkovi během útoku **zaznamenává informace** (např. zdrojovou IP adresu).
4. Zaznamenané informace od honeypotu **využije následně firewall pro vytvoření pravidel, která zabrání budoucím útokům.**

# Osnova přednášky



- **Obecné pojmy z oblasti bezpečnosti počítačových sítí**
- **Zařízení pro zabezpečení počítačových sítí**
  - Firewall
  - IPS/IDS
  - Honeypot
- **Vybrané útoky v počítačových sítích**

# Některé typické útoky v počítačových sítích I



- 1) **Skenování portů** = zjišťování služeb, které běží na daném počítači (výhodné pro zahájení dalšího konkrétně cíleného útoku).
  - Skenování portů je realizováno prostřednictvím navazováním spojení (TCP) či zasíláním skupin paketů (UDP) na vybraný cílový port stanice s příslušnou IP adresou.
  - V případě, že stanice na požadavky zasílané na daný cílový port odpovídá, hovoříme o **otevřeném portu**, v opačném případě o **uzavřeném portu**.
  - **Typy skenování:**
  - **Horizontální** = útočník skenuje konkrétní port na počítačích s různými IP adresami.
  - **Vertikální** = útočník skenuje více portů na počítači s jednou IP adresami.
  - Účinnou obranou jsou **firewally (znemožnění)**, **IDS** a **IPS (nahlášení či znemožnění)**.

## 2) Prolamování hesel

- Útok probíhá prostřednictvím terminálových služeb (SSH, RDP, TELNET, viz přednáška 10).
  - **Hrubá síla (brute force attack)** = útočník zkouší všechny možné kombinace znaků nad danou abecedou.
  - **Slovníkový útok (dictionary attack)** = útočník používá předem definovaný seznam hesel, která uživatelé používají často.
- Účinnou obranou jsou **firewally (Ize se připojit pouze z povolených IP rozsahů)** a **IPS (znemožnění útoku)**.

# Některé typické útoky v počítačových sítích II



**3) Denial of Service (DOS)** = útok realizovaný za účelem zabránění přístupu ke službě.

- **Typy realizace:**
  - Zasíláním velkého množství dotazů ICMP echo.
  - Otevřáním velkého množství spojení TCP-SYN (SYN FLOOD).
- **Distributed DOS (DDOS)**
  - Verze DOS vedená současně z různých infikovaných počítačů, viz dále.
- **Reflected/Spoofed DOS (RDOS)**
  - IP Adresa zdroje v hlavičce IP paketu je podvržena a lze tudíž těžko identifikovat útočníka, viz dále.
- Účinnou obranou jsou **IPS, viz slajd 24**.

**4) Útoky využívající specifické bezpečnostní chyby v konkrétním systému**

- Těchto útoků je mnoho a řeší je bezpečnostní záplaty systému.
- Škodlivé aplikace provádějící tento typ útoku se označují jako **červi (worms)**.
- Např. útok Buffer overfill = přepsání konkrétní oblasti hlavní paměti vyvolá specifickou akci (např. restart počítače, vyčtení hesel atd.).
- Účinnou obranou jsou pravidelné **bezpečnostní záplaty systémů**.

# Některé typické útoky v počítačových sítích III



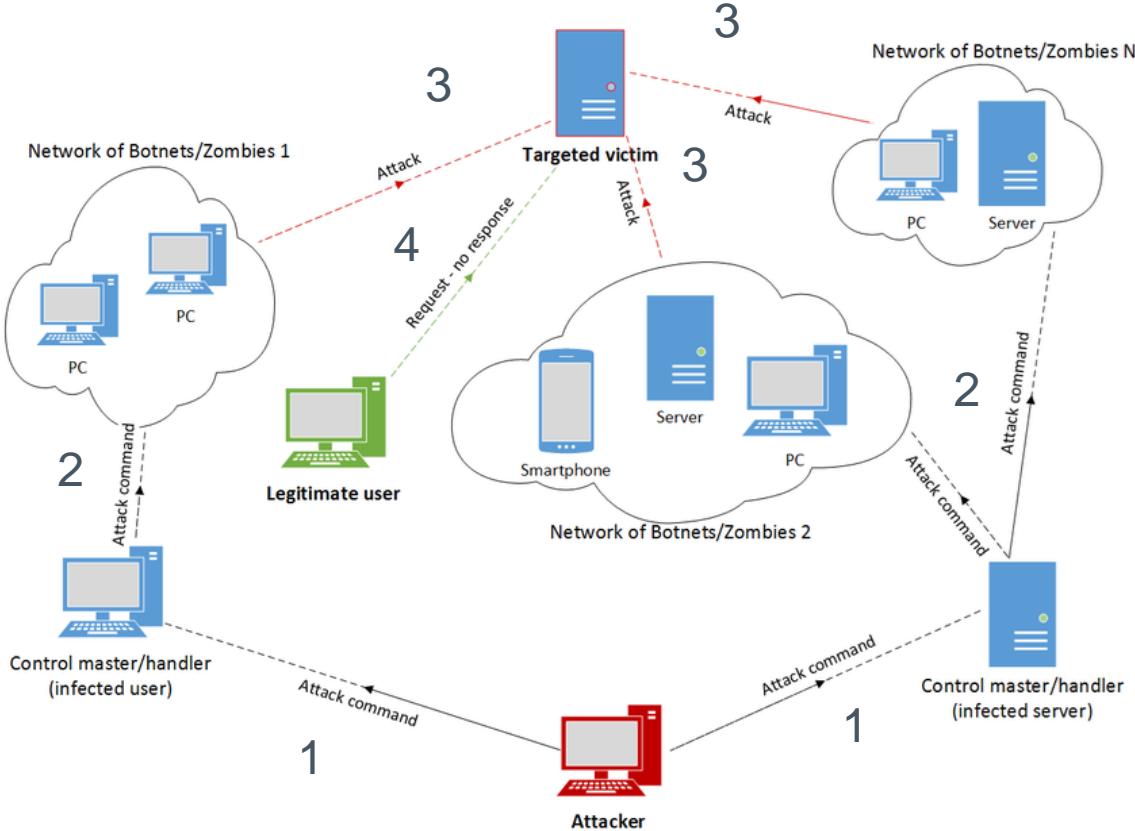
## 5) Odesílání spamu

- Uživatel obdrží od útočníka email, který obsahuje **závadný obsah**.
- Nejčastější typy útoků dle typu obsahu jsou tyto:
  - **Phishing** = útok na vylákání identity (jméno a heslo) uživatele, např. otevřením odkazu vedoucím na falešnou přihlašovací stránku.

## 6) Malware

- **Malware** = aplikace, které si uživatel stáhne z Internetu popř. dostane emailem a spustí. Aplikace následně provádí specifické akce již bez vědomí uživatele.
- **Trojský kůň** = malware, který se většinu času chová jako běžná legitimní aplikace, avšak obsahuje i nepravidelně spouštěný škodlivý kód.
- **Ransomware** = malware, který provádí šifrování obsahu pevného disku.
- **Spyware** = malware, který se snaží vyčíst na základě nastavení OS různé informace o uživateli (přihlašovací údaje, bankovní informace atd.).
- **Adware** = malware, který zobrazuje reklamní nabídky.
- Účinnou obranou jsou **antivirové programy** a **emailové blacklisty**.
- Infikování uživatelských stanic malwarem je efektivním mechanismem pro možnost ovládnutí stanic za účelem **realizace DDOS útoku**, viz dále.

# Distributed Denial of Service (DDoS) útok



Zdroj: [https://www.researchgate.net/figure/The-principle-of-a-DDoS-attack\\_fig1\\_273918445](https://www.researchgate.net/figure/The-principle-of-a-DDoS-attack_fig1_273918445).

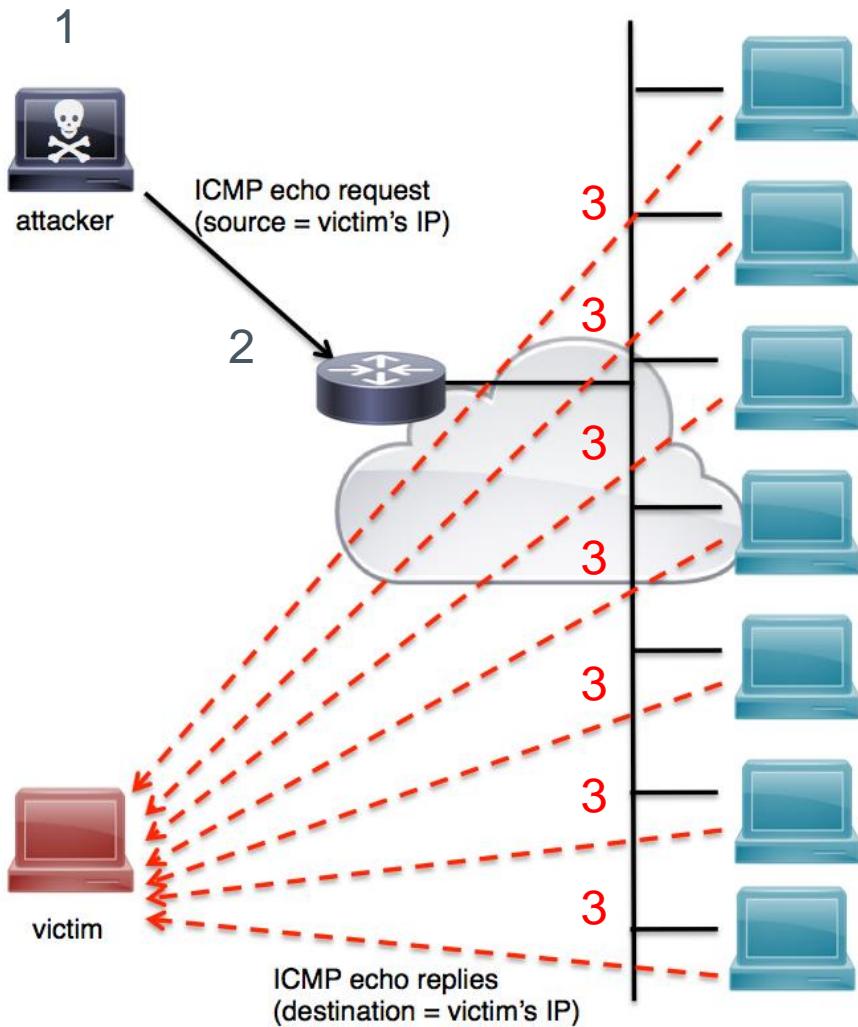
Původní útočník (Attacker) útok přímo neprovádí.

## Fáze útoku:

1. Útočník **nejprve napadne a ovládne několik počítačů** (CM = Control master/handler).
2. Pomocí CM **infikuje a ovládne další skupiny počítačů** (botnets/zombies).
3. Útočník prostřednictvím botnets/zombies **aktivně útočí (attack command)** na konkrétní **cílovou stanici** (targeted victim).
4. Legitimní uživatel nedostane od služby, běžící na cílové stanici odpověď kvůli jejímu síťovému přetížení.

Jelikož je útok veden víceúrovňově, je **velmi složité odhalit** původního útočníka.

# Reflected/Spoofed DDoS útok



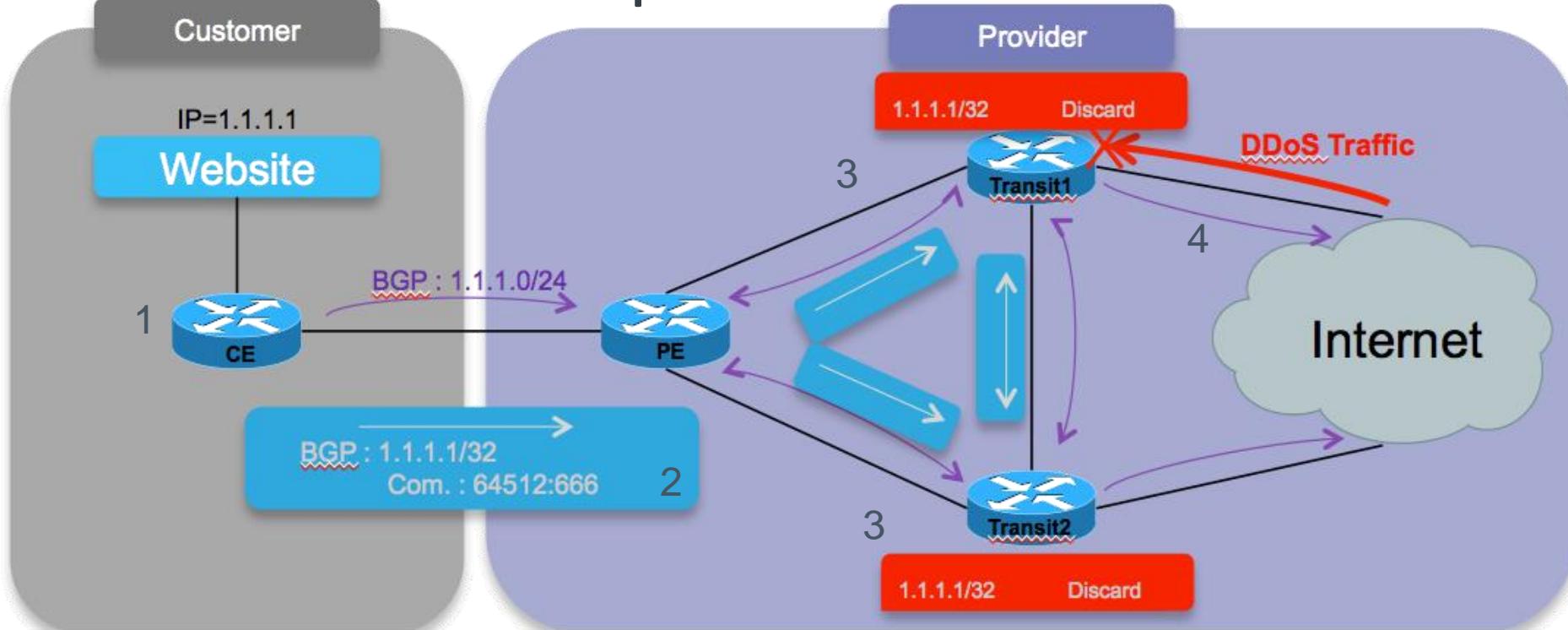
## Fáze útoku:

1. Útočník (attacker) nastaví v hlavičce IP paketu se zprávou (např. **ICMP request**) jako **zdrojovou adresu oběti** (victim).
2. Tímto paketem útočník obešle unicastově či **broadcastově** (tím se útok ještě zesílí = amplified attack) jeden či více počítačů v lokální síti.
3. Všechny počítače, které zprávu dostanou, odpoví (např. zasláním **ICMP echo**) dle podvržené zdrojové IP adresy oběti, což způsobí přetížení na jejím síťovém rozhraní.

Primární motivací tohoto útoku je zamezení možnosti vystopování útočníka.

Zdroj: <https://blog.cloudflare.com/reflections-on-reflections/>.

# Řešení proti DDoS útokům



Zdroj: z <https://community.cisco.com/>

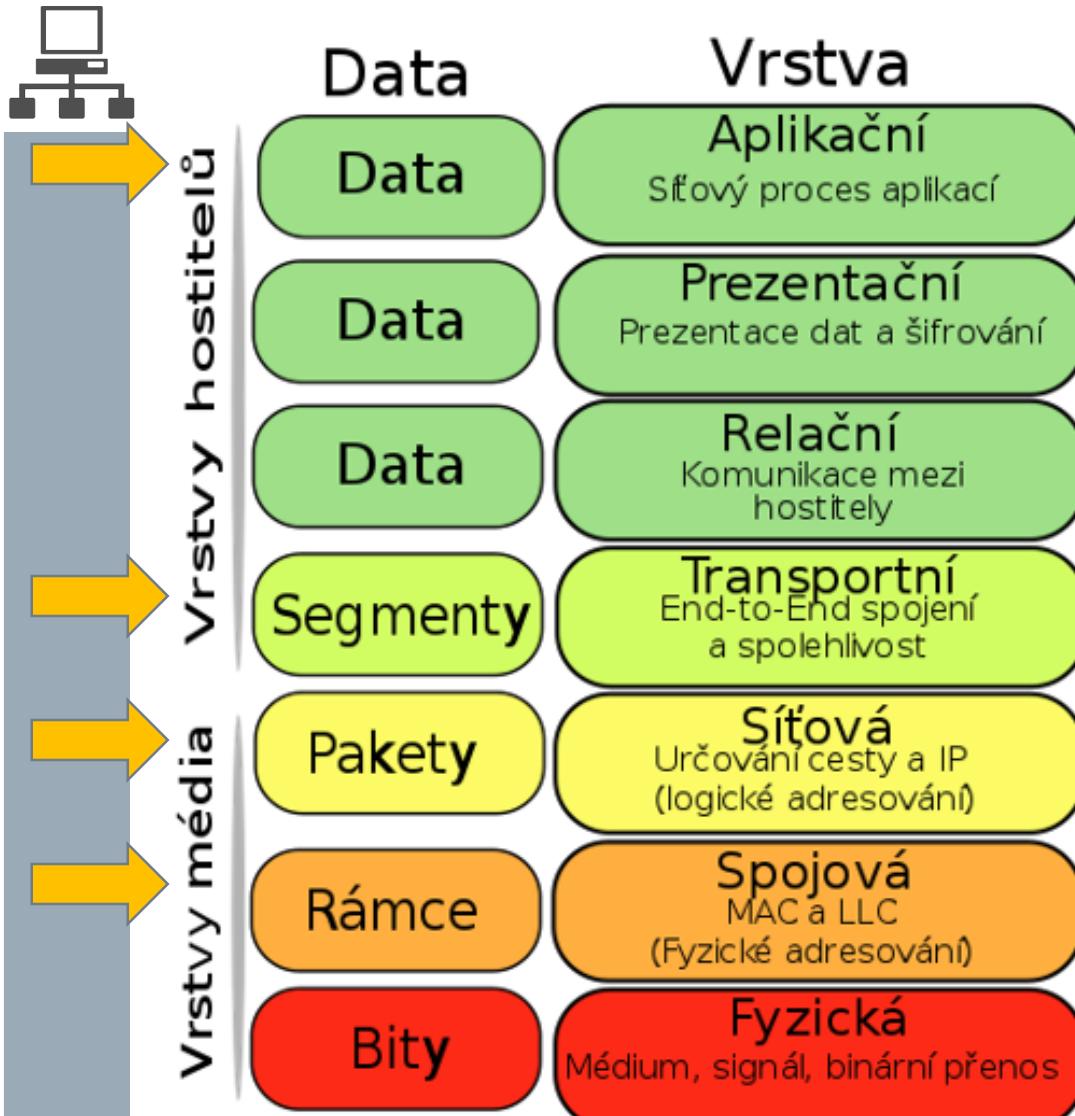
1. V síti (Customer) je díky IDS spolupracující se směrovačem (CE) zjištěn DDoS útok vedený na stanici s určitou IP adresou (např. 1.1.1.1).
2. Pomocí rozšíření FlowSpec protokolu BGP, směrovač, který útok detekoval, odešle zprávu směrovači (PE) v síti poskytovatele (Provider), přes který je atakovaná IP adresa dosažitelná.
3. PE obešle všechny hraniční (transit) směrovače, které propojují síť poskytovatele do Internetu
4. Hraniční směrovače provedou zablokování provozu směřujícího na atakovanou IP adresu a tím cílovou stanici ochrání před přetížením.

# Počítačové sítě

9. Přednáška - Virtuální privátní sítě (VPN)



# Základní pojmy a motivace pro využití VPN



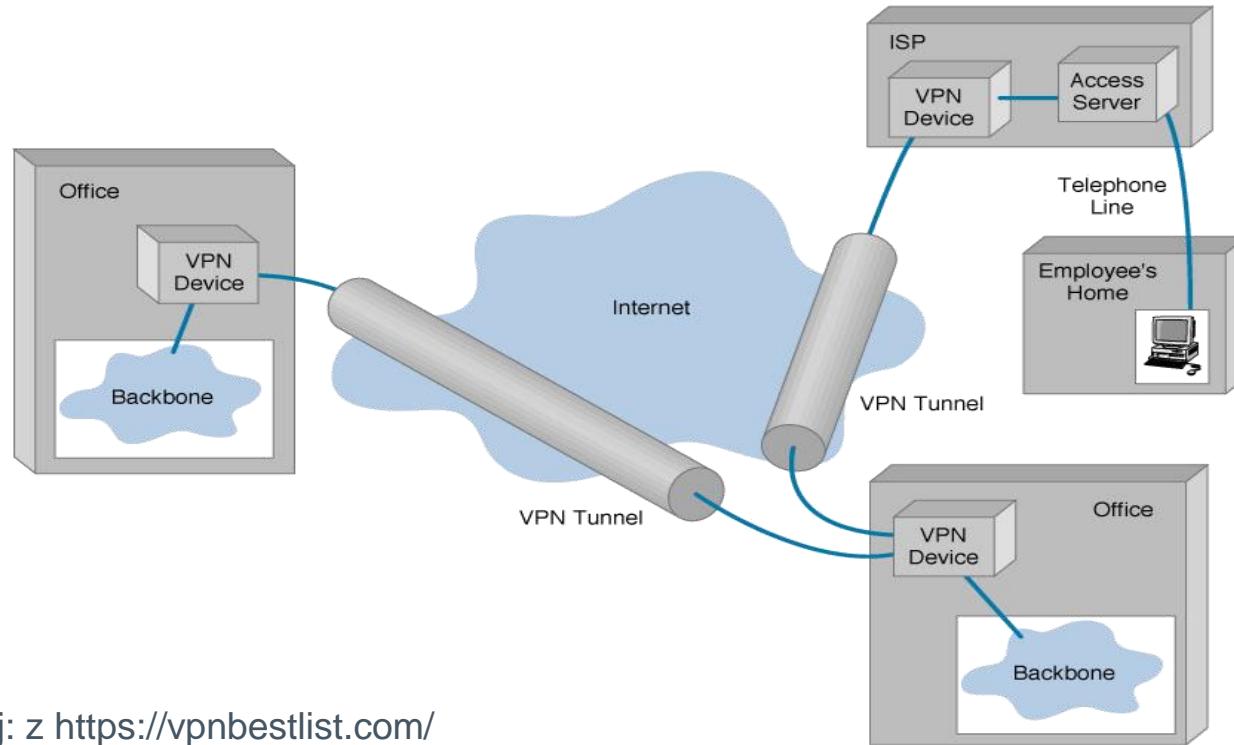
- VPN = **Virtual Private Networks**.
- Úkolem VPN je zpřístupnění místní sítě či její části uživateli, který je **síťově resp. geograficky** od ní oddělený.
- Důvodem pro využití VPN bývá často anonymizace původu konkrétního uživatele.
- VPN se využívají na odstranění omezení vyplývající z geografických důvodů, např. určité webové stránky jsou přístupné pouze v některém státě.
- VPN existují v **šifrované i nešifrované** formě.
- Vstupním místem VPN do místní sítě je VPN zařízení (koncentrátor).
- Existují více druhů VPN. Liší se tím, že pracují na **různých vrstvách** OSI modelu.
- Existují jak v hardwarových tak i softwarových implementacích.
- V praxi se využívají velmi často.
- VPN se v operačních systémech implementují nejčastěji pomocí virtuálních síťových adaptérů.

# Druhy VPN – dle místa použití



- **Intranet** – v rámci stejné sítě jedné organizace, např. propojení dvou oddělení v rámci jedné firmy.
- **Extranet** – v rámci jedné organizace, nicméně prostřednictvím externího poskytovatele Internetu (ISP), např. dvě pobočky stejné firmy v rámci jednoho města či republiky.
- **Vzdálený přístup (Remote Access)** – připojení určitého zaměstnance do firmy, např. možnost práce z domova v případě home office.

# Zařízení a části VPN



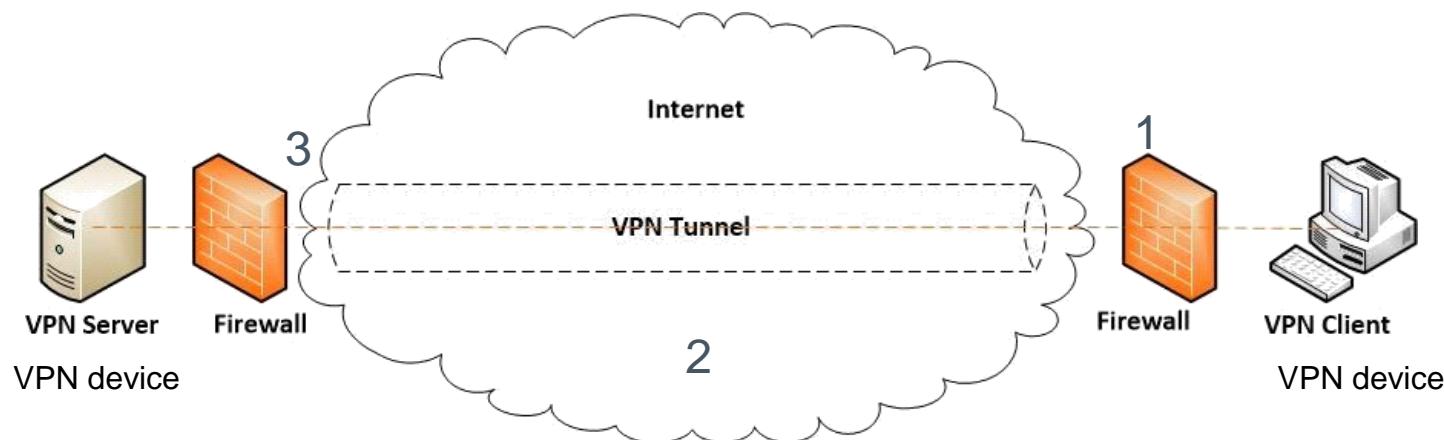
Zdroj: z <https://vpnbestlist.com/>

- **Přístupový (Access)/VPN Server** = přístupový server, který zprostředkovává vzdálený přístup uživateli připojenému prostřednictvím datové či telefonní linky.
- **VPN zařízení (VPN Device)** = zařízení, které slouží k propojení místní sítě s okolními sítěmi prostřednictvím VPN. Pro komunikaci mezi zařízeními se využívají VPN protokoly. VPN zařízení může být realizováno **softwarově** či **hardwarem**.
- **VPN tunel** = virtuální spojení realizované mezi VPN zařízeními.

# Tunelování



- **Tunelování** = posíláním dat přes vytvořený tunel.
- Vytvoření tunelu musí být **povoleno ve firewallech**, které se nacházejí mezi zařízením na straně klienta a přístupovým serverem.
- Velmi často se tunelování používá v případě, že existuje síťová služba, která **nemá nativní podporu šifrování**. Tento nedostatek vyžaduje vytvoření tunelu prostřednictvím šifrovaného VPN protokolu.



1. Před vstupem do tunelu jsou **data zapouzdřena** do příslušného VPN protokolu.
2. Zapouzdřená data se přenesou prostřednictvím VPN tunelu.
3. Doručená zapouzdřená data jsou opět zbavena zapouzdření a doručena k cílové stanici.

# Zapouzdření dat ve VPN tunelu



- **Nosný protokol (NP)** = protokol nižší vrstvy, který zajišťuje doručení dat paketů VPN protokolu.
- **VPN Protokol** = protokol, který se používá pro vytvoření tunelu.
- **Zapouzdřená data** = pakety či rámce původního protokolu která jsou přenášeny prostřednictvím tunelu. Zapouzdření může být i vícenásobné.

**Příklad zapouzdření dat:**

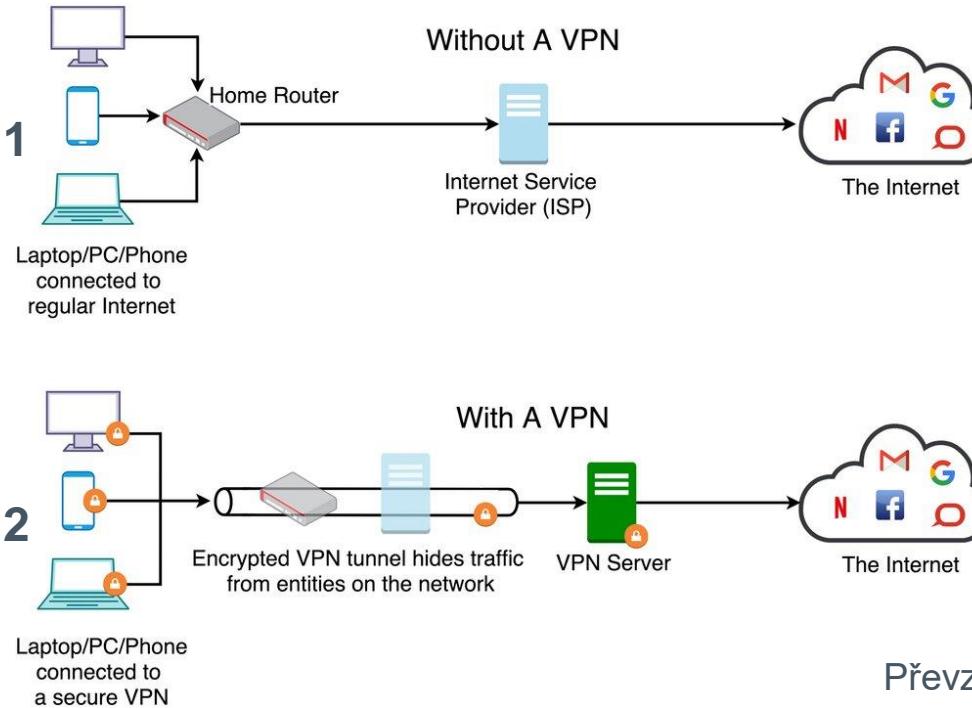


Formát paketu VPN, která pracuje  
na síťové vrstvě

# Směrování provozu bez využití resp. s využitím VPN



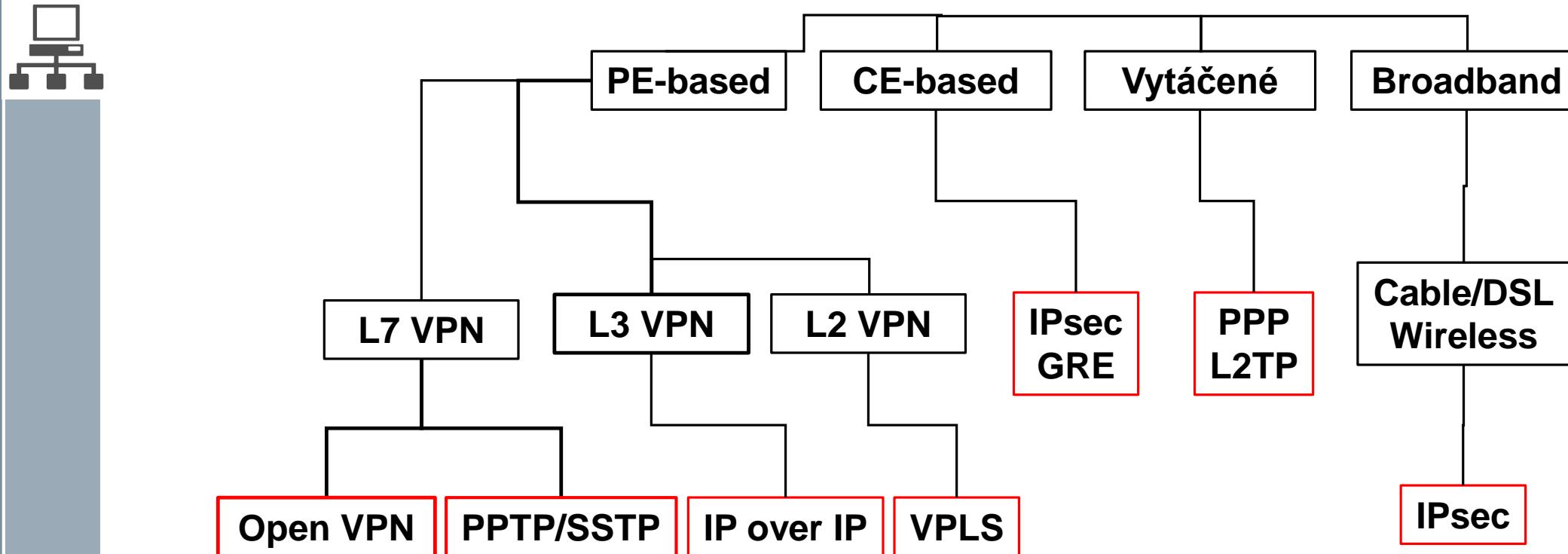
Častým důvodem pro využití VPN je **anonymizace** IP adresy uživatele.



Převzato z <https://vpnbestlist.com/>

1. Provoz uživatele prochází **přes domácí směrovač**. Servery v Internetu komunikují s uživatelem prostřednictvím IP protokolu. Z pohledu serverů v Internetu, je v hlavičkách paketů IP protokolu uvedena **jako zdrojová/cílová IP adresa** domácího směrovače **uživatele**.
2. Provoz uživatele prochází přes domácí směrovač, který přeposílá veškerý provoz tunelem na příslušný VPN server. Z pohledu serverů v Internetu je v hlavičkách paketů IP protokolu uvedena **jako zdrojová/cílová IP adresa** **VPN serveru**. Díky tomuto je možné uživatele anonymizovat.

# Důležité VPN protokoly a jejich dělení



- **CE** = Customer Edge, VPN zařízení je na straně zákazníka.
- **PE** = Provider Edge, VPN zařízení je na straně poskytovatele.
- **LX VPN** = Layer X VPN, VPN pracuje na vrstvě X modelu OSI.
- **DSL** = Digital Subscriber Line, VPN je realizována prostřednictvím digitální linky. Typicky se používá pro telefonní linky či kabelovou televizi.
- VPN protokoly uvedené v **červeném rámečku** budou probírány v další části výkladu.

# Point-to-Point protokol(PPP)



- Jednoduchý protokol, který se používá pro **vzdálený přístup** (původně navržen pro využití u telefonních linek či DSL).
- Pracuje na **linkové vrstvě**.
- Nepoužívá pro adresaci dvojice adres, jelikož vždy propojuje **právě 2 strany**.
- Podporuje **kompresi i šifrování** dat či možnost autentizace.
- Dokáže přenášet rámce linkové vrstvy i pakety síťové vrstvy.

## Formát rámce PPP



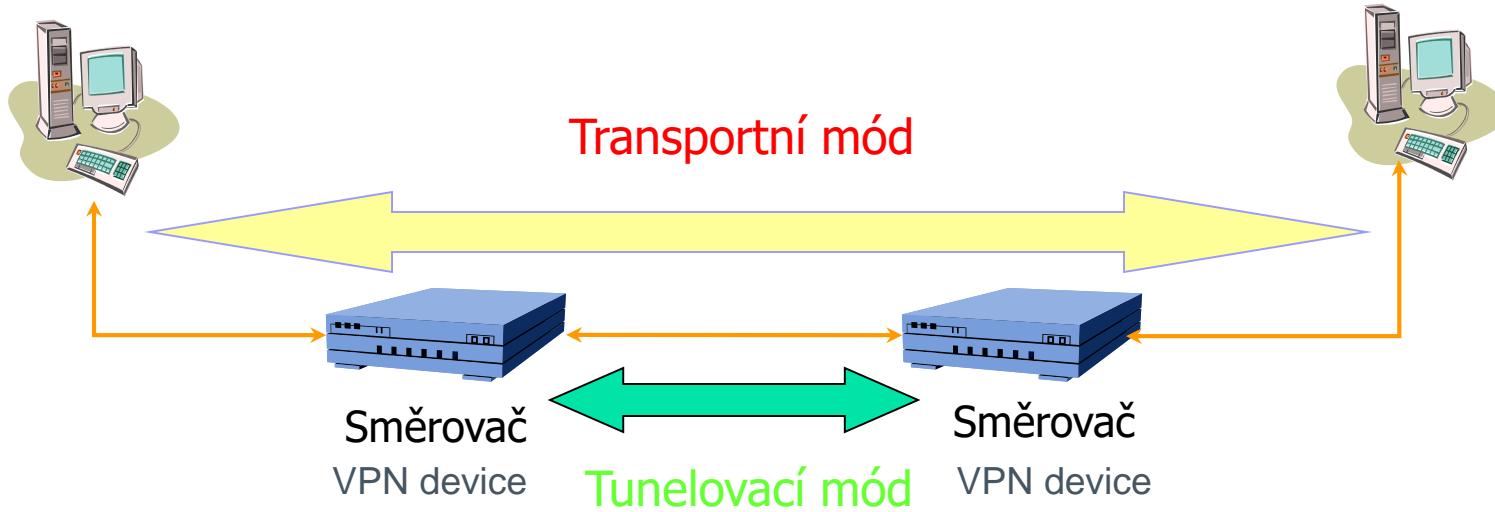
- **TAG** (1 bajt) = označení začátku a konce rámce.
- **Adresa** (1 bajt) = identifikace cíle, má vždy hodnotu 255 (broadcast).
- **Řídicí pole** (1 bajt) = určení typu prováděné operace, např. autentizace, přenášení dat.
- **Protokol** (2 bajty) = identifikace protokolu, jehož data jsou přenášena v položce Data (např. IP).
- **Data** (max. 1500 bajtů) = konkrétní přenášené informace.
- **Kontrolní součet** (1 bajt) = ochrana vůči chybám při přenosu.

# IPSec protokol



- **IPSec = IP Secured.**
- Pracuje na **síťové vrstvě**.
- Jeho primárním úkolem je **ověření identity účastníků a šifrování síťového provozu**.
- Pro zabezpečení IP sítí je IPsec nutnost, neboť žádná verze IP protokolu v návrhu neobsahuje vlastní bezpečnostní mechanismy.
- IPv6 již od dob návrhu počítá s možným využitím IPsecu.
- IPSec propojuje vždy právě dvě strany.
- IPSec je posloupnost tří **důležitých fází**:
  - **Internet Key Exchange (IKE)** = protokol, který má za úkol dohodnutí šifrovacích klíčů, které dále využívají **AH či ESP**.
  - **Authentication Header (AH)** = protokol, který slouží pro zajištění vzájemné autentizace mezi stranami.
  - **Encapsulating Security Payload (ESP)** = protokol, který zajišťuje symetrické šifrování přenášených dat.
- Aby byl IPsec **kompatibilní** se zařízeními, které jej nepodporují, resp. podporují jen IP, je šifrována v IP paketu pouze část DATA.
- IPsec pracuje ve dvou módech – **transportním a tunelovacím (viz dále)**.

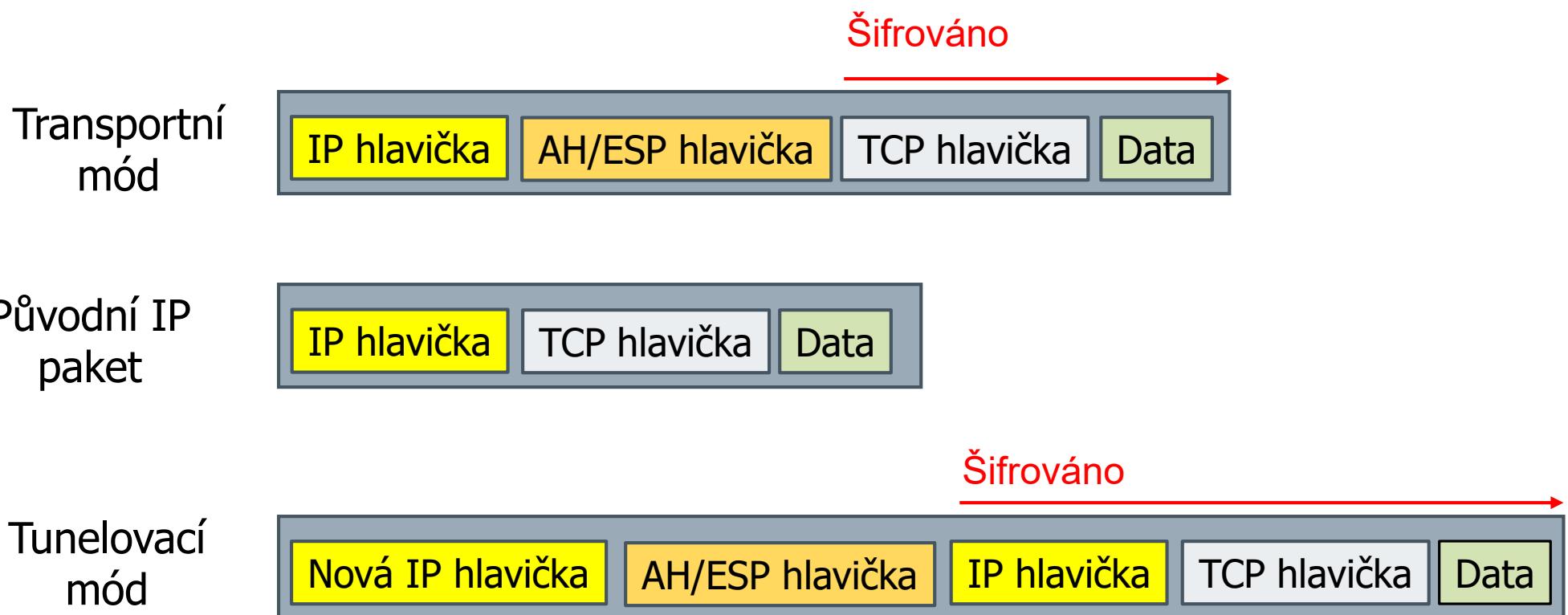
# IPsec – transportní a tunelovací mód



**Transportní mód** = původní IP hlavička je zachována a šifruje se pouze datová část paketu. Výhodou transportního módu jsou nižší nároky na kapacitu přenosové linky.

**Tunelovací mód** = původní IP paket je zabalen a chráněn v nově vytvořeném IP paketu. Nový paket obsahuje IP adresy směrovačů, mezi kterými je tunel sestaven. Tento mód je náročnější na kapacitu přenosové linky, avšak dokáže skrýt IP adresu zdrojové a cílové stanice.

# Hlavičky v IPsec protokolu dle módu



AH/ESP hlavička: ve fázi autentizace resp. přenosu

# Protokol IP over IP

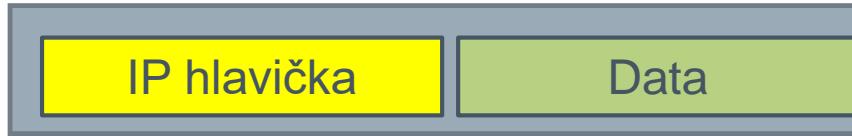


- Důvodem pro použití tohoto protokolu je propojování sítí, které jsou umístěné za směrovači, které provádějí překlad adres (NAT).
- Protokol pracuje na **sítové vrstvě**.
- Principem je to, že původní IP pakety **jsou zabalené do nových IP paketů**. Původní pakety se přenáší v položce **DATA** nových IP paketů.
- Přeposílají se pouze **celé IP pakety**, fragmentace není povolena.
- Protokol je velmi jednoduchý a nenáročný, má nízkou komunikační režii.
- **Neobsahuje** šifrování ani bezpečnostní mechanismy.
- Existuje jako standard a je podporován všemi předními výrobci sítových technologií.

# IP over IP - zapouzdření dat



Původní IP paket



IP over IP paket



# Generic Routing Encapsulation (GRE)



- Jeden z prvních VPN protokolů, vyvinutý firmou **Cisco**.
- **Pracuje na transportní vrstvě**.
- **GRE zapouzdření je využíváno dalšími VPN protokoly, např. PPTP (viz dále).**
- Hlavička protokolu GRE neobsahuje porty na rozdíl od protokolů TCP/UDP a tudíž je špatně průchozí pro překlad adres (**NAT**).

**Hlavička GRE protokolu (detailně viz dále):**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31										
C	R	K	S	s	Rekurze		Příznaky		Verze		Protokol																														
Kontrolní součet (volitelné)																Offset (volitelné)																									
Klíč (volitelné)																																									
Sekvenční číslo (volitelné)																																									

# Položky v hlavičce GRE protokolu

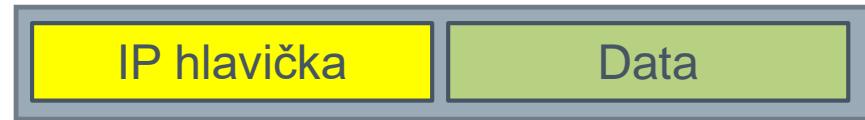


- **C, R, K, S** = odpovídající příznaky, které udávají informace o tom, zda se v hlavičce nachází dané volitelné položky.
- **Rekurze** = počet dalších zapouzdření je povoleno.
- **Příznaky** = rezervované byty, musí být nastaveny na 0.
- **Verze** = verze GRE hlavičky: 1, pokud se jedná o protokol PPTP, jinak 0.
- **Protokol** = označení protokolu, jehož data jsou zapouzdřena.
- **Kontrolní součet (volitelný)** = vypočítaný kontrolní součet z GRE hlavičky a dat.
- **Klíč (volitelný)** = číslo, které bylo vloženo společně se zapouzdřením. Přijímací strana je používá k ověření odesílatele paketu.
- **Sekvenční číslo** = číslo, které bylo vloženo společně se zapouzdřením. Přijímací strana určuje pořadí, v jakém byly pakety odeslány.

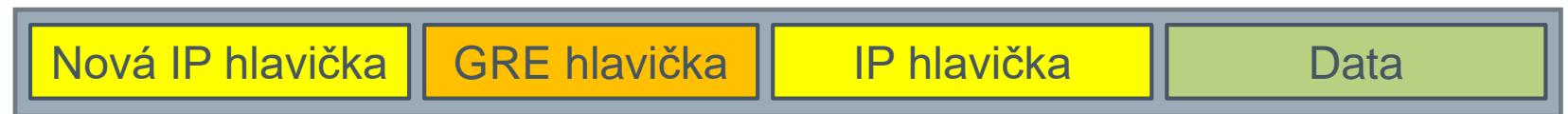
# GRE - zapouzdření dat



Původní IP paket



GRE paket



# Point to Point Tunneling Protokol (PPTP)



- Vyvinutý společností **Microsoft** → integrován nativně ve Windows.
- Využívá pro svou činnost protokoly **GRE** a **PPP**.
- Je postaven na modelu **klient-server**.
- **Spojení, které vznikne, je obousměrné.**

**Hlavička PPTP protokolu:**

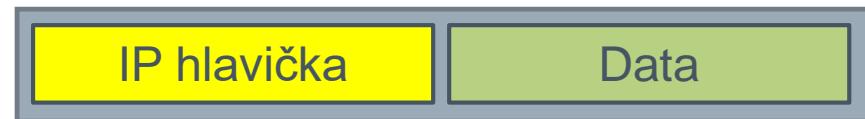
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Délka																Typ zprávy															
Magické cookie																															

- **Délka** = velikost PPTP zprávy v bytech.
- **Typ zprávy** = identifikátor nabývajíc hodnot 1 nebo 2. Zprávy typu 2 nicméně nejsou definovány.
- **Magické cookie** = hodnota tohoto pole je vždy nastavena na 0x1A2B3C4D.

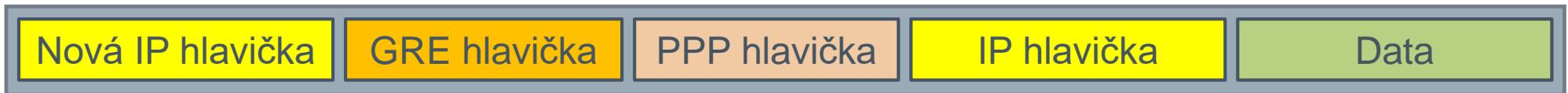
# PPTP - zapouzdření dat



Původní IP paket



PPTP paket



# Layer 2 Tunneling Protokol (L2TP)



- L2TP je výsledkem spolupráce členů PPTP fóra, Cisca a organizace IETF (Internet Engineering Task Force).
- Účelem L2TP je tunelování rámců **linkové vrstvy**.
- L2TP sám o sobě je ale nešifrovaný, nicméně může používat pro zaouzdroření Ipsec. Tato kombinace se nazývá L2TP/IPsec.

Hlavička L2TP protokolu (detailně viz dále):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
T	L	x	S	x	O	P		x																								
ID Tunelu																Délka (volitelné)																
Ns (volitelné)																ID relace																
Velikost offsetu (volitelné)																Vyplnění offsetu (volitelné)																
Data																																

# Položky v hlavičce L2TP protokolu



- **T, L, x, S, x, O, P** = příznaky a rezervované bity.
- **Verze** = verze protokolu.
- **Délka** = velikost zprávy v bajtech.
- **ID tunelu** = jednoznačný identifikátor tunelu.
- **ID relace** = jednoznačný identifikátor relace v rámci daného tunelu.
- **Ns** = pořadové číslo pro data nebo řídící zprávu.
- **Nr** = pořadové číslo, které je očekávané v příští řídící zprávě, která má být přijata.
- **Velikost offsetu** = určuje počet bajtů za L2TP hlavičkou, kde se očekává začátek dat.

# L2TP - zapouzdření dat



Původní Ethernet rámec



L2TP paket v ethernetovském rámci



# Secure Socket Tunneling Protokol (SSTP)

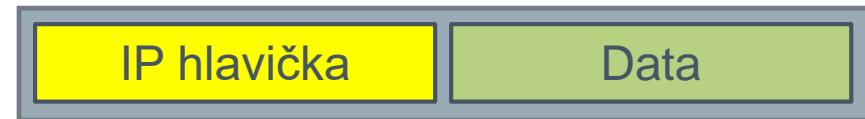


- SSTP pracuje na **aplikační vrstvě**.
- Autorem je **Microsoft**.
- **Integrován pod Windows, ale implementovaný i v Linuxu.**
- **Je přenášen v protokolem HTTPS.**
- Nahrazuje postupně PPTP protokol.
- Pro zapouzdření dat používá **PPP protokol**.
- Je velmi dobře připustný pro firewally, jelikož se tváří jako HTTP provoz.
- **Fáze vytváření SSTP spojení:**
  - 1. Sestavení TCP spojení
  - 2. Spuštění šifrování prostřednictvím SSL
  - 3. Přenos přes HTTPS

# SSTP - zapouzdření dat

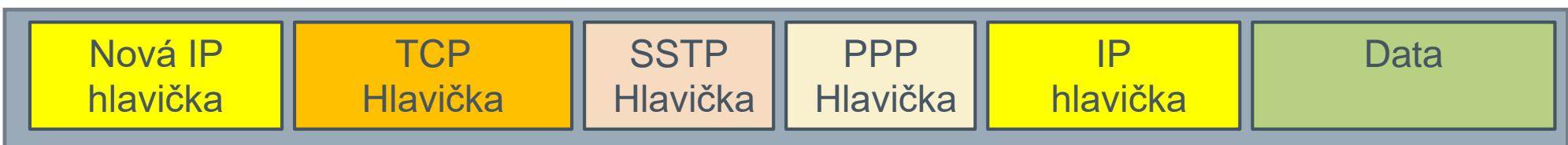


Původní IP paket



SSTP paket

šifrováno

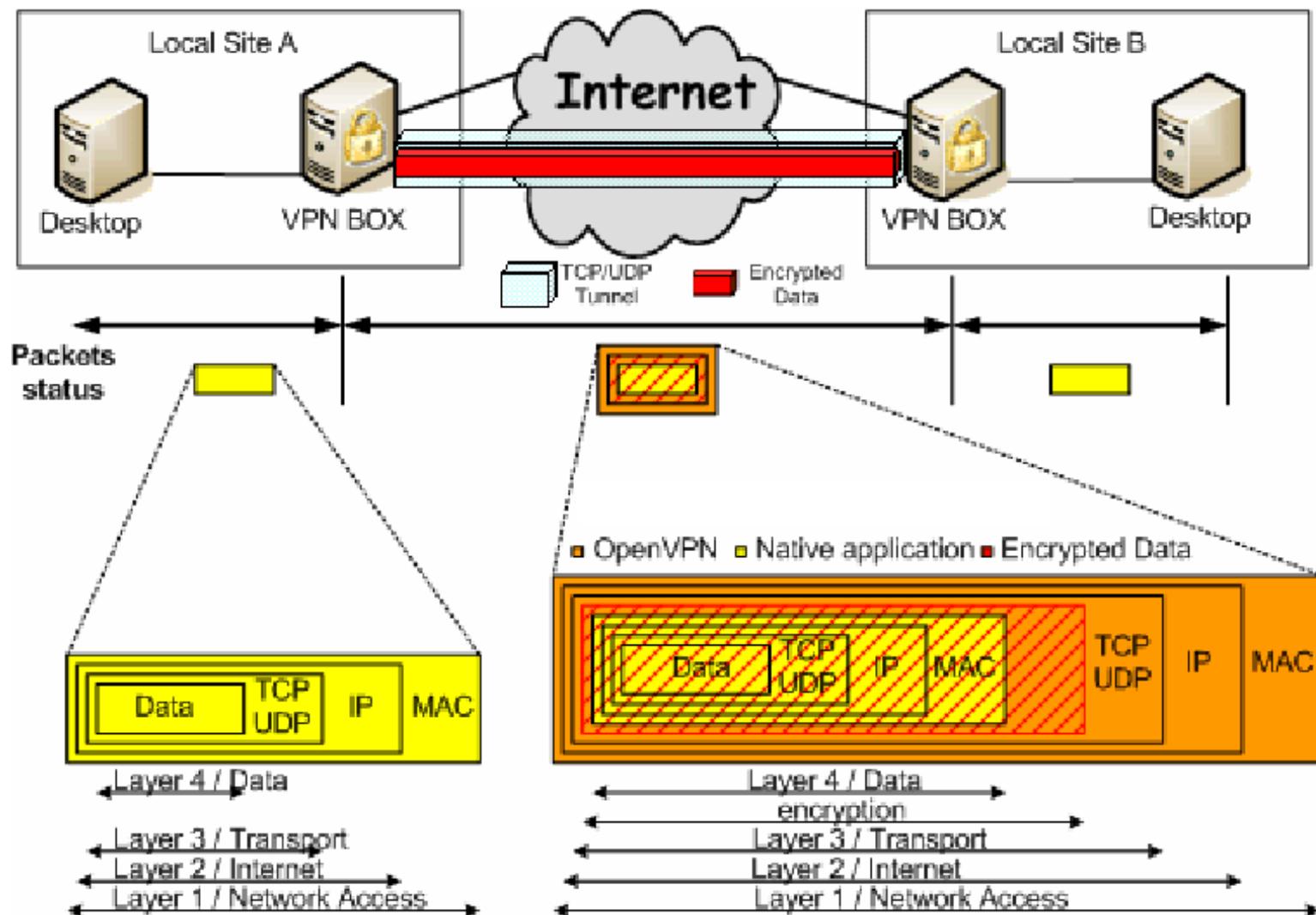


# Protokol Open VPN



- Jedná se o protokol **aplikační vrstvy**, který dokáže přenášet rámce linkové či pakety síťové vrstvy.
- VPN je v operačním systému implementovaná prostřednictvím virtuálního síťového rozhraní.
- Virtuální síťová rozhraní se v OS označují jako **TAP** (pro linkovou vrstvu) nebo **TUN** (pro síťovou vrstvu)
- Pakety OpenVPN jsou přenášeny jako data **protokolu UDP nebo TCP** dle nastavení administrátora.
- OpenVPN může fungovat v režimu point-to-point (2 strany) anebo point-to-multi-point (více stran).
- OpenVPN obsahuje implementace pro téměř všechny běžně dostupné operační systémy a je populární.

# OpenVPN – zapouzdření dat



Převzato z <https://openmaniak.com/cz/openvpn.php>

# Počítačové sítě

10. Přednáška - Anonymita v Internetu



# Internet není anonymní



- Prohlížeč uživatele na Internetu zanechá specifický otisk.
- **Otisk prohlížeče (OP) = množina specifických atributů zasílaných v rámci HTTP spojení webovým prohlížečem. Tyto atributy lze využít pro pravděpodobnou identifikaci uživatele.**
- Řada služeb využívá cíleně OP pro přizpůsobení obsahu poskytnutého uživateli (např. reklamy).
- Částečnou ochranou před zneužitím OP je využití **VPN** a **PROXY** serverů, obojí však vymění pouze IP, **OP zůstává stejný**.
- Řešením může být VPN + vhodný anonymizér či použití **anonymního módu v prohlížeči** (to má však další omezení, např. znova načítání cookies).
- Ověřit si anonymitu prohlížeče lze na <https://panopticlick.eff.org/>
- Pro různé prohlížeče existuje řada anonymizérů, známým řešením je Privacy Badger.

# Atributy sloužící pro vytvoření otisku prohlížeče



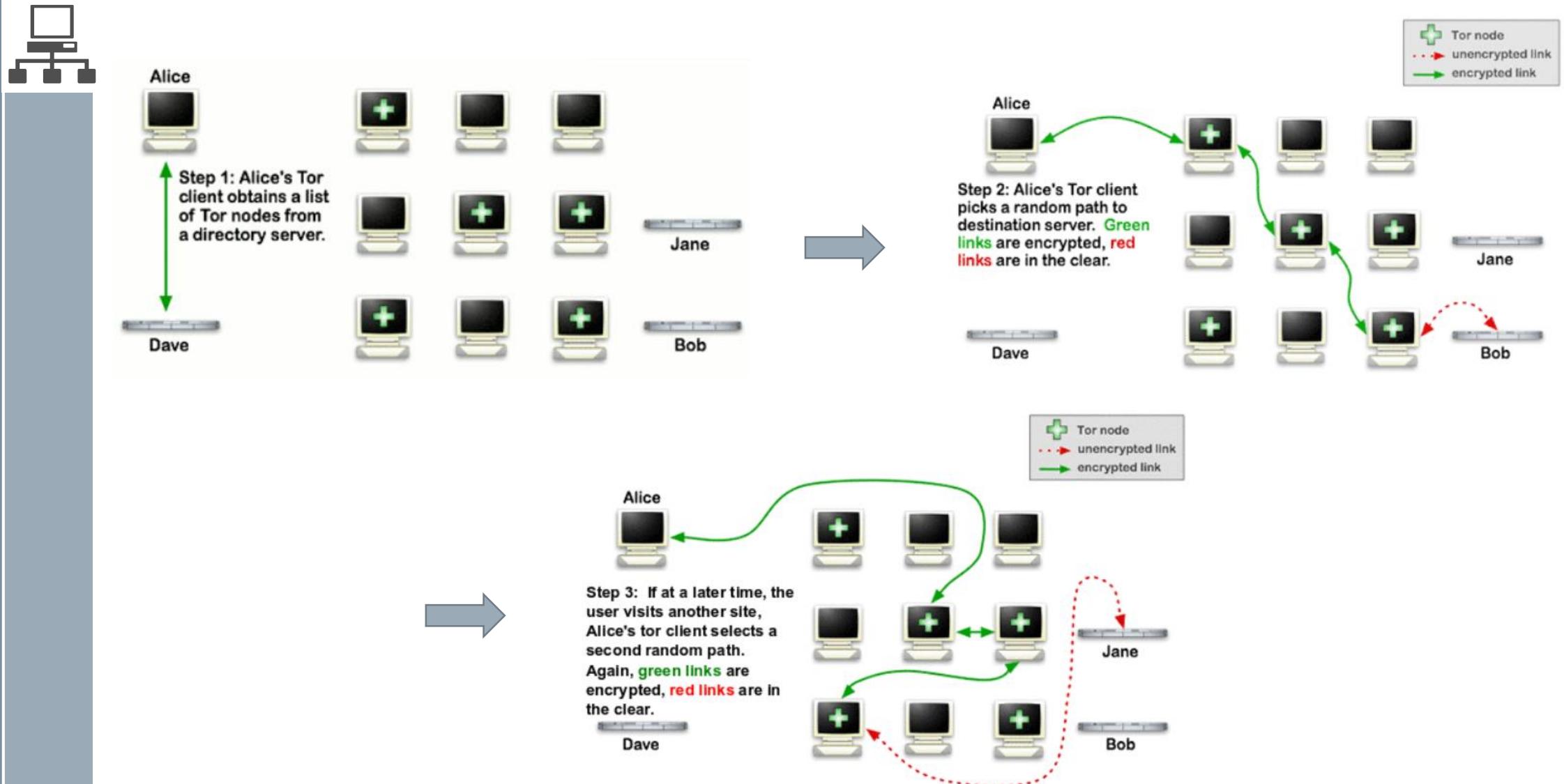
Attribute	Panopticlick (2010)		AmIUnique (2016)		Hiding (2018)	
	Entropy	Normalized entropy	Entropy	Normalized entropy	Entropy	Normalized entropy
User agent	10.000	0.531	9.779	0.580	7.150	0.341
Accept	-	-	1.383	0.082	0.729	0.035
Content encoding	-	-	1.534	0.091	0.382	0.018
Content language	-	-	5.918	0.351	2.716	0.129
List of plugins	15.400	0.817	11.060	0.656	9.485	0.452
Cookies enabled	0.353	0.019	0.253	0.015	0.000	0.000
Use of local/session storage	-	-	0.405	0.024	0.043	0.002
Timezone	3.040	0.161	3.338	0.198	0.164	0.008
Screen resolution and color depth	4.830	0.256	4.889	0.290	4.847	0.231
List of fonts	13.900	0.738	8.379	0.497	6.904	0.329
List of HTTP headers	-	-	4.198	0.249	1.783	0.085
Platform	-	-	2.310	0.137	1.200	0.057
Do Not Track	-	-	0.944	0.056	1.919	0.091
Canvas	-	-	8.278	0.491	8.546	0.407
WebGL Vendor	-	-	2.141	0.127	2.282	0.109
WebGL Renderer	-	-	3.406	0.202	5.541	0.264
Use of an ad blocker	-	-	0.995	0.059	0.045	0.002
$H_M$ (worst scenario)	18.843		16.860		20.980	
Number of FPs	470,161		118,934		2,067,942	

# Tor a DarkNet



- Změna zdrojové IP adresy (jako při použití VPN) nemusí být z hlediska anonymity dostatečná. Lze odhadnout na základě hodnoty odezvy např. síťová vzdálenost uživatele.
- **Tor** = síť, která si klade za úkol znemožnit sledování uživatelů.
- Tor umožňuje anonymní (cibulové) směrování. Toto směrování mezi uživateli je pokaždé jiné (nedá se pak odhadnout ani vzdálenost zdroje).
- **Cibulové směrování zaručuje anonymitu klienta i serveru.**
- Tor poskytuje službu **Onion** service = anonymní obdoba DNS, založená na **odlišném principu**, která má za úkol zpřístupnění určité služby.
- **Darknet** = anonymní neregulovaná verze WWW.
- Pro přístup do sítě Tor se používá specifický prohlížeč (Tor Client), který zmíněné služby podporuje.
- Více lze nalézt na <https://www.torproject.org/>.

# Tor – Cibulové směrování

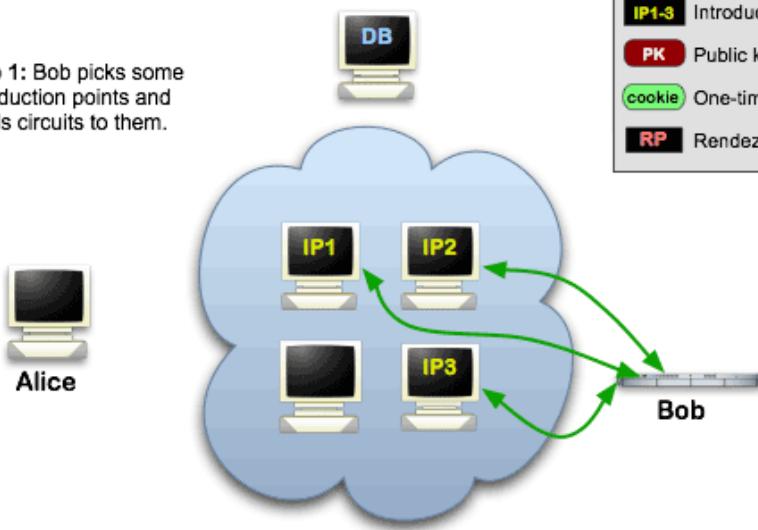


Převzato z <https://2019.www.torproject.org/about/overview.html.en>

# Tor – Onion Service

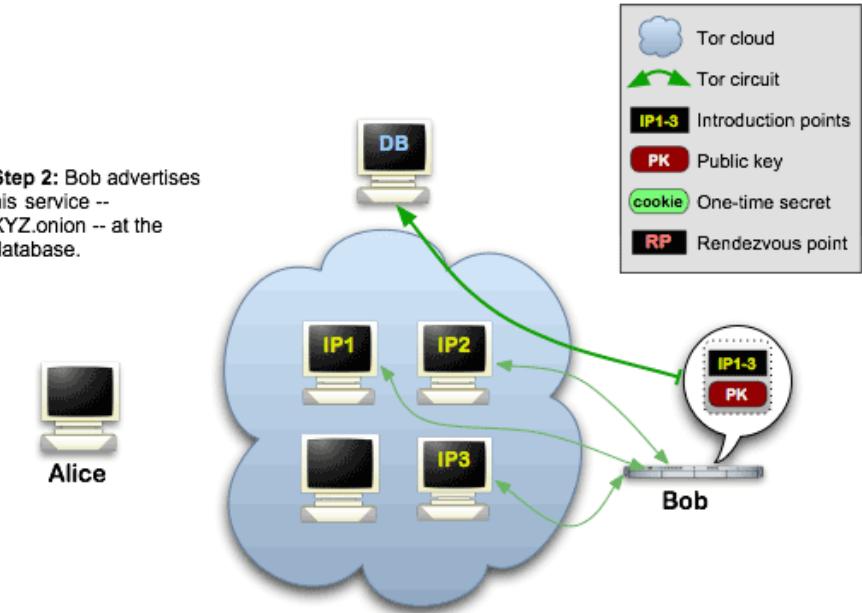


Step 1: Bob picks some introduction points and builds circuits to them.



Bob vytvoří spojení k několika vstupním Tor uzlům.

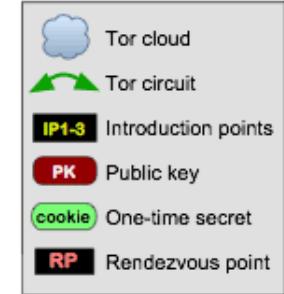
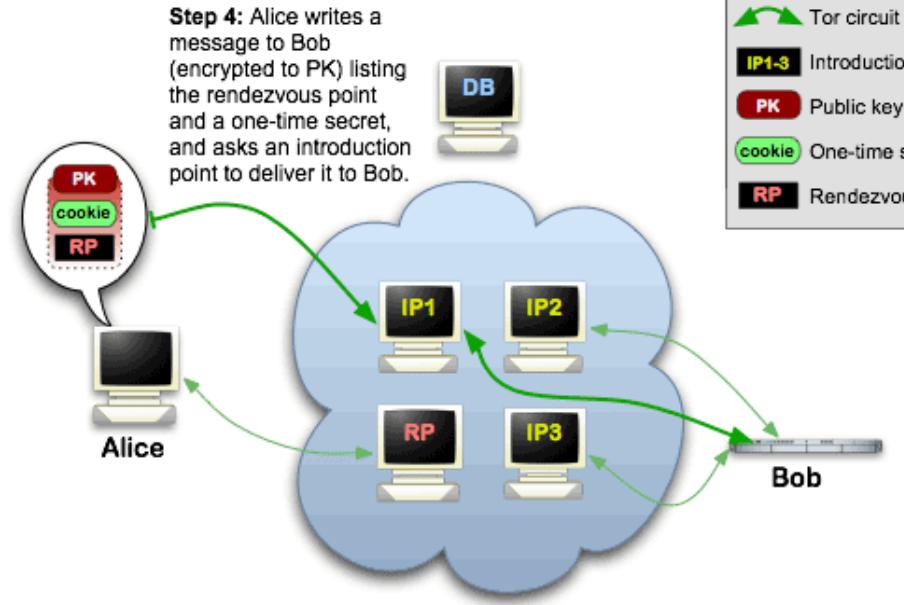
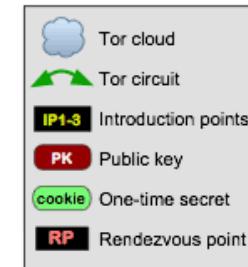
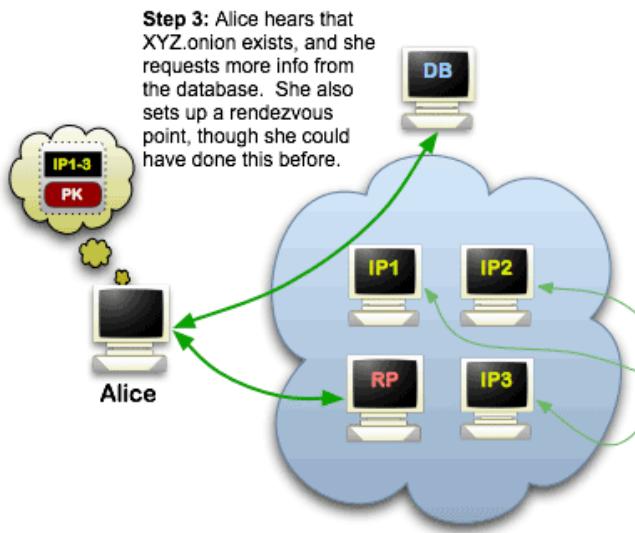
Step 2: Bob advertises his service -- XYZ.onion -- at the database.



Bob nabídne svoji službu XYZ.onion prostřednictvím centrální databáze (DB). V DB uveďe svůj PK a vstupní Tor uzly.

Převzato z <http://tor.void.gr/docs/onion-services.html.en>

# Tor – Onion Service



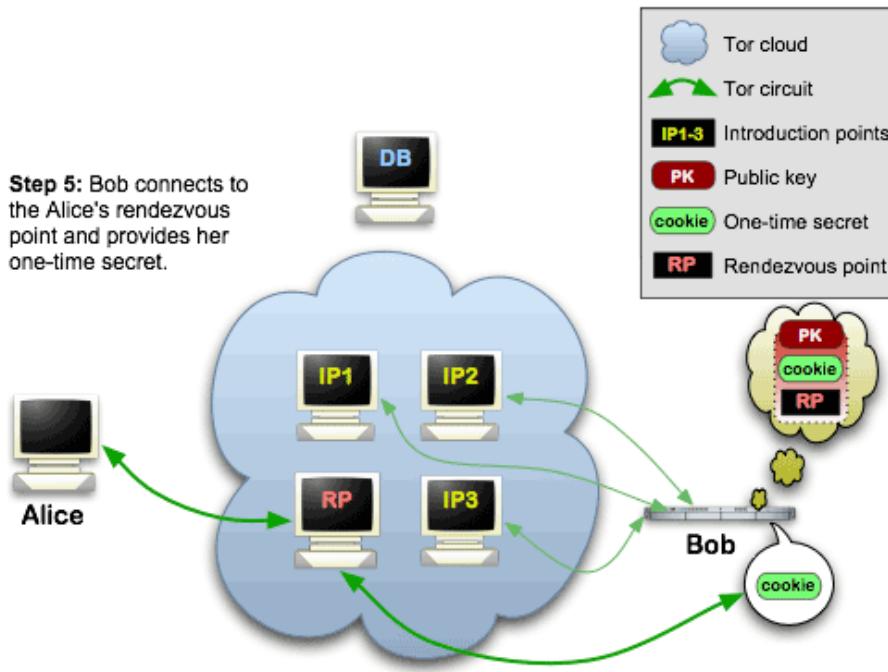
Alice se dozví o existenci služby XYZ.onion a přečte si o ní informace z centrální DB.

Jeden z uzelů Tor sítě Alice označí jako Rendezvous Point (RP = místo setkání).

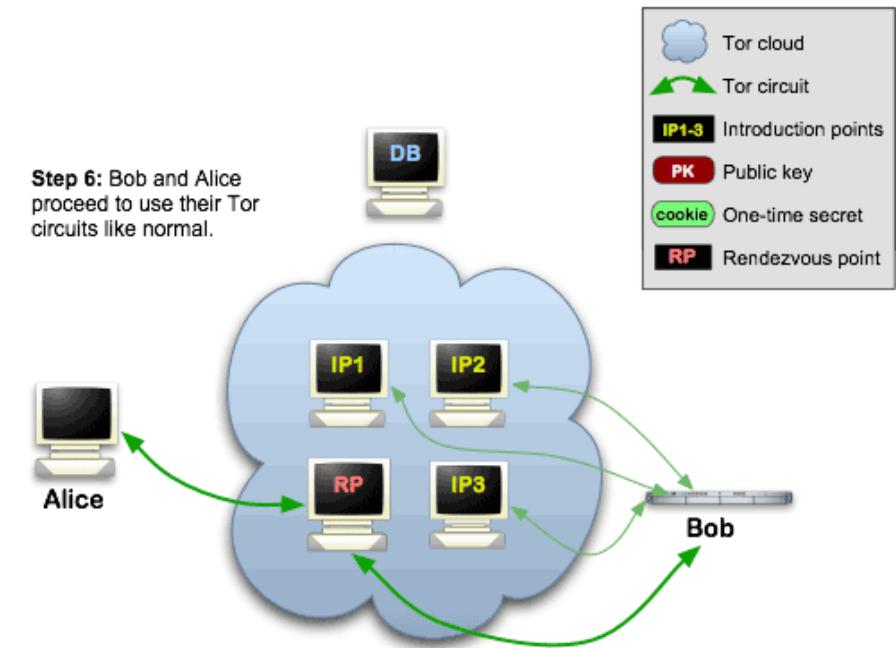
Alice si vybere náhodně vstupní bod, napíše Bobovi zprávu, kterou zakóduje jeho jeho veřejným klíčem. Zpráva obsahuje RP a jednorázový šifrovací klíč (cookie), který si Alice vygenerovala pro konkrétní relaci.

Převzato z <http://tor.void.gr/docs/onion-services.html.en>

# Tor – Onion Service



Bob se připojí k RP a poskytne  
Alici rovněž jednorázový  
šifrovací klíč.



Bob a Alice zahájí mezi sebou  
komunikaci.

Převzato z <http://tor.void.gr/docs/onion-services.html.en>

# Počítačové sítě

10. Přednáška - Aplikační vrstva, protokoly a služby



# Aplikační vrstva a její význam



- Úkolem aplikační vrstvy je **interakce uživatele a vzdálených síťových služeb**.
- Uživatelé využívají síťové služby prostřednictvím aplikačních protokolů.
- Implementace protokolu aplikační vrstvy je realizovaná v konkrétní aplikaci. Aplikace komunikuje s uživatelem.
- Protokoly aplikační vrstvy spolupracují s protokoly prezentační vrstvy, které přinášejí další služby, např. šifrují přenášená data.
- Aplikačních protokolů je velké množství a stále se vyvíjejí nové.
- Typickými službami, které zprostředkovávají aplikační protokoly, jsou např. tyto
  - Terminálové služby (vzdálená správa)
  - Přenos souborů
  - Elektronická pošta (emaily)
  - Webové stránky
  - Zasílání zpráv (messaging)
  - Komunikace prostřednictvím VoIP

# Přehled vybraných služeb a aplikačních protokolů



Druh služby	Protokol	Aplikace (aktuálně používané)
Vzdálená správa	SSH, Telnet, RDP, VNC	Putty, Remote Desktop, Teamviewer, Aeradmin
Přenos souborů	FTP, TFTP, SCP/SFTP	ProFTPD
Email	SMTP, POP3(s), IMAP	PostFix, Dovecot
Webové služby	HTTP(s), QUICK	Mozilla, Safari, Chrome, MS Edge
Časová synchronizace	NTP	ntpd
Vzdálené bootování a konfigurace	BOOTP, DHCP	
P2P sdílení	BitTorrent, Donkey, Gnutella	mTorrent
Monitorování sítě	SNMP	Nagios, Cacti, Zabbix
VoIP	SIP, RTP, Skype	Skype
IPTV	RTP	VLC
IoT	MQTT	

Vybrané služby a protokoly označené červeně budou probírány v další části výkladu.

# Protokoly vzdálené správy – protokoly SSH a TELNET



- Protokoly pro vzdálenou správu stanic či serverů prostřednictvím terminálu.
- Terminál emuluje obrazovku a klávesnici.
- Zprávy obou protokolů se přenáší v položce data **TCP** paketu.
- Oba protokoly jsou založeny na modelu **klient-server** = aplikace běžící na klientské stanici se připojuje k aplikaci, která běží na serveru.
- Tyto služby jsou poměrně často používány pro nabourání do systému, např. pro **útok hrubou silou** či **pomocí slovníku** (viz 8. přednáška, slajd 15).
- Pro Linux existuje utilita **fail2ban**, která prohledává **autentizační log** (soubor se záznamy přístupů). Prostřednictvím **firewallu** (iptables) zakazuje přístup k terminálovým službám klientským stanicím s IP adresami, které překročí předem nastavený maximální počet neúspěšných přihlášení.

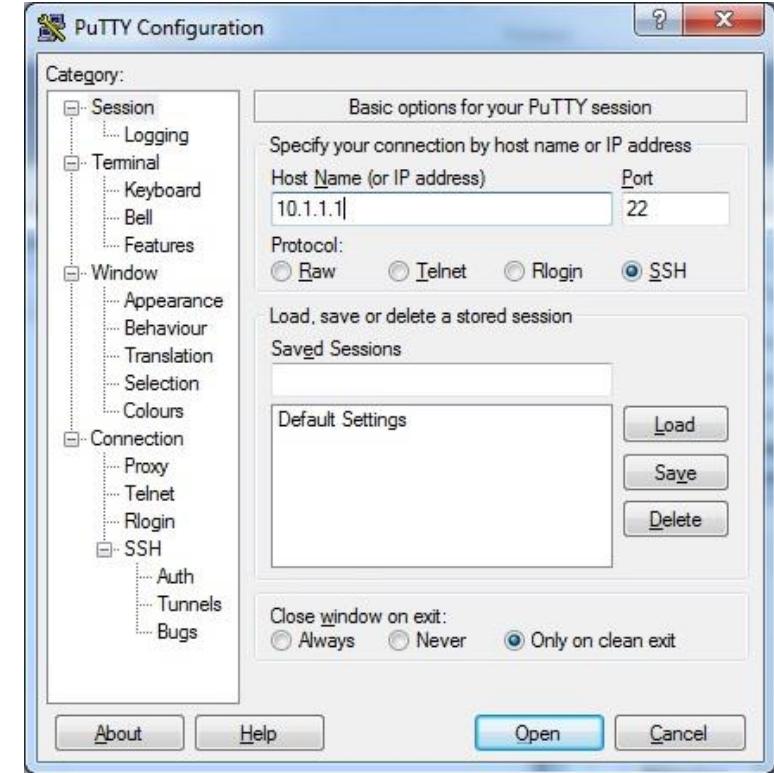
SSH = Secure Shell	Telnet
PORT 22	PORT 23
Používá šifrování	Bez zabezpečení
Autentizace heslem, sdíleným klíčem	Autentizace heslem
Podpora na všech populárních operačních systémech	Podpora v Linuxu a ve Windows
Náročnější na využití CPU	Nenáročné na využití CPU

# Ukázka vzdálené správy - protokol SSH

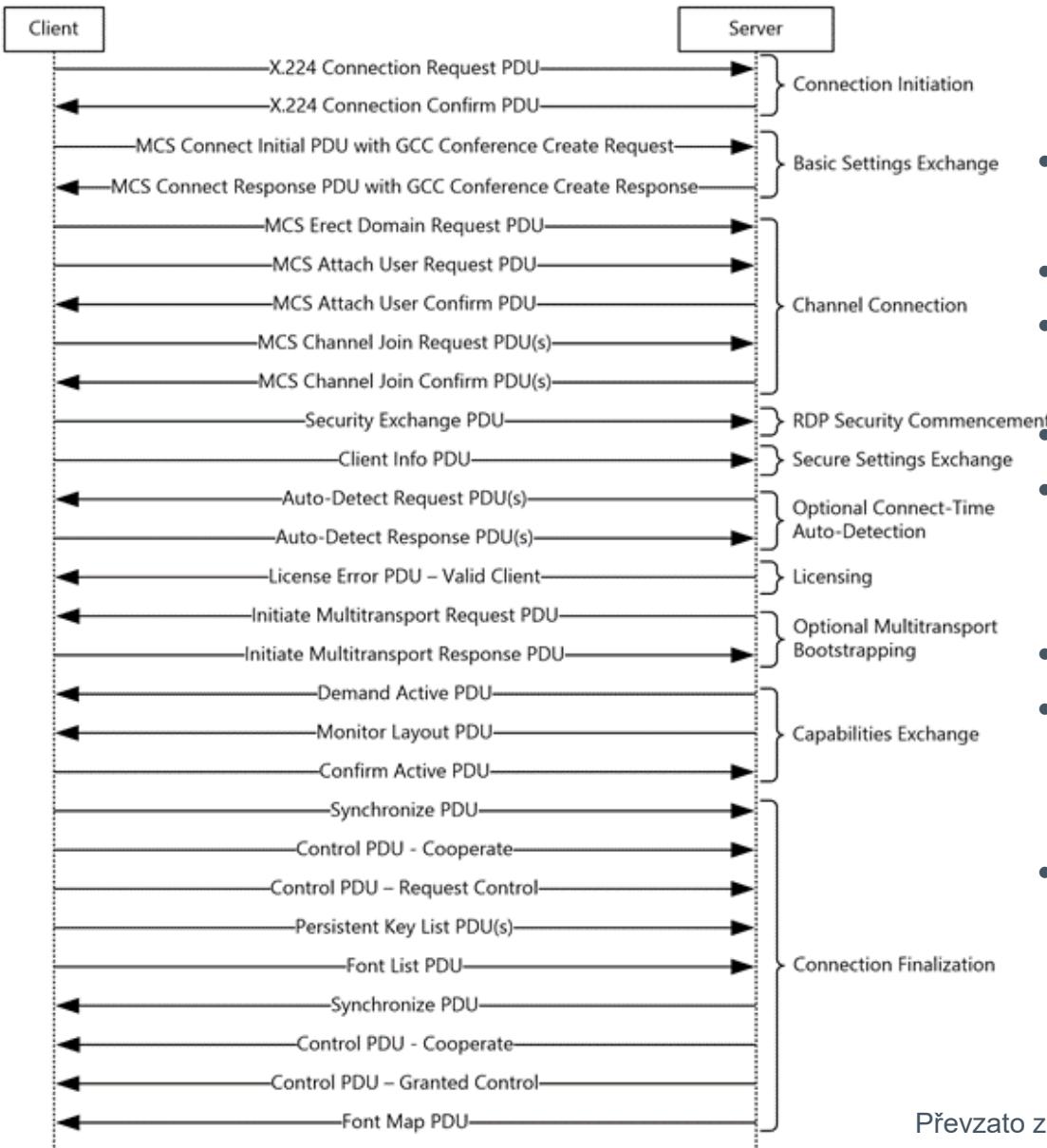


```
login as: root
root@10.5.1.1's password:
Linux AZRAEL-1 4.14.0LuckyNet-edition #1 SMP Sun Nov 26 21:07:48 CET 2017 x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 23 16:31:32 2020 from 10.0.255.89
root@AZRAEL-1:~#
```



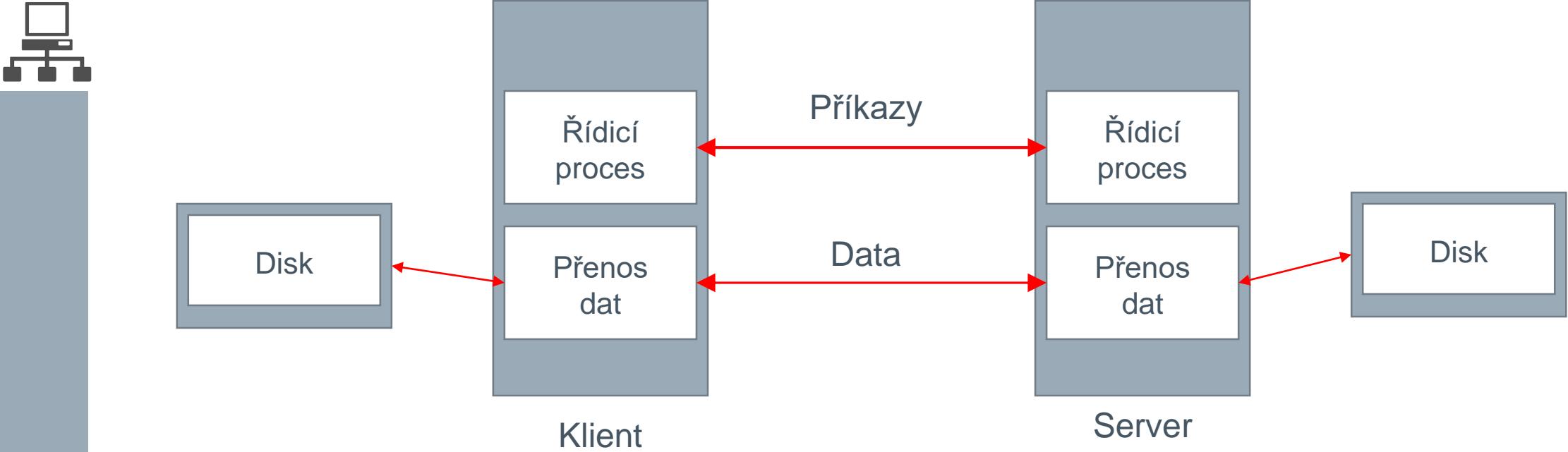
# Protokoly vzdálené správy – Remote Desktop Protocol (RDP)



- Emuluje přístup ke grafickému rozhraní počítače.
- Běžně se používá port **3389**.
- Zprávy RDP jsou přenášeny v položce data v paketu **TCP protokolu**.
- Primárně vyvinut pro **MS Windows**.
- Současné verze Linuxu** již ovšem emulaci napojenou na X Window systém **zvládají** rovněž.
- Protokol RDP je **vývoji** a stále **se vylepšuje**.
- Protokol je napsaný efektivně, přenáší se pouze nutné informace a inkrementální změny.
- I tento protokol je poměrně často zneužíván pro nabourání do systému, princip je obdobný jako u protokolu SSH či Telnet.

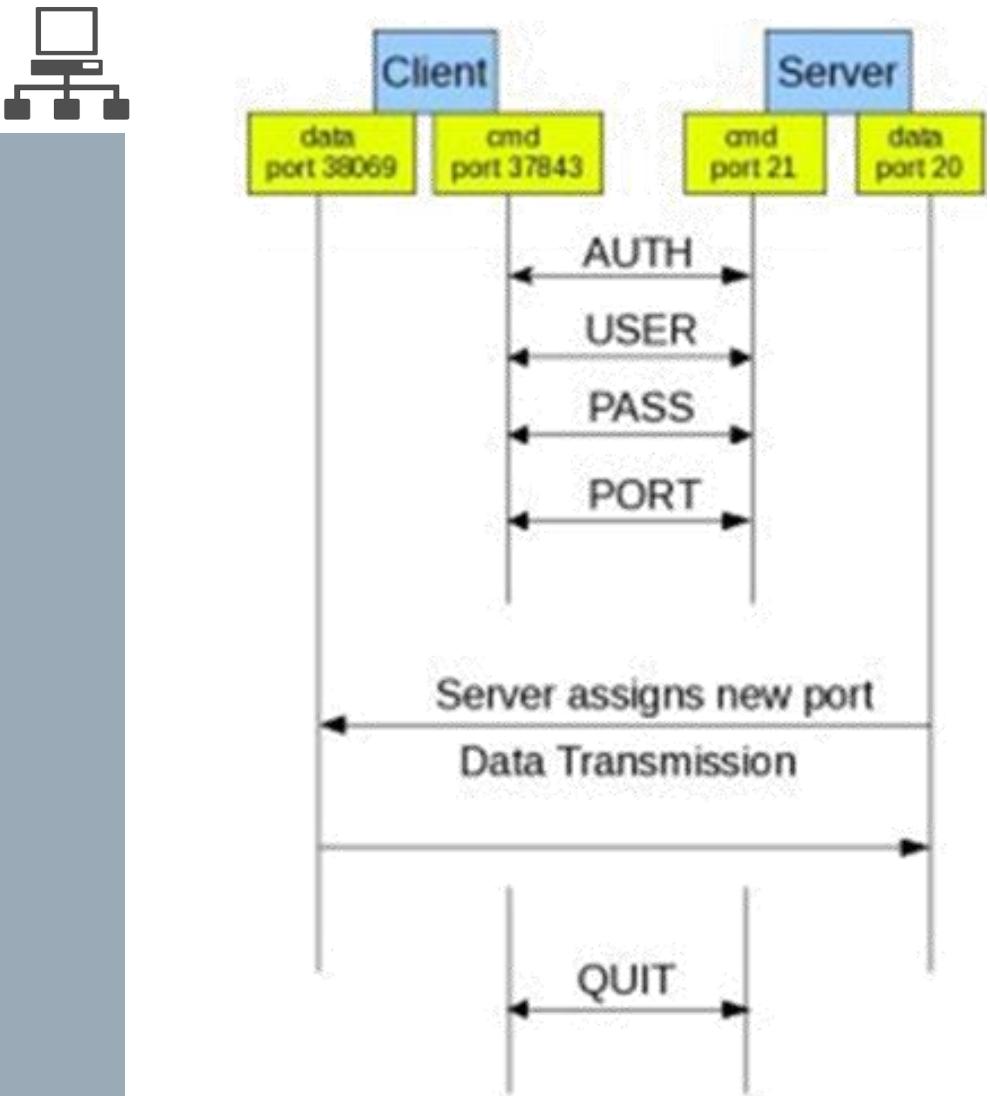
Převzato z [https://docs.microsoft.com/en-us/openspecs/windows\\_protocols/ms-rdpbcgr/](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpbcgr/)

# Protokoly přenosu souborů - FTP



- Zprávy FTP se přenáší jako data v paketu protokolu TCP.
- FTP vytváří dvě různá spojení (pro příkazy a data), která se liší cílovými porty.
- FTP funguje v pasivním nebo v aktivním módu, viz dále.
- **Běžný protokol FTP není šifrovaný, šifrovaná varianta se nazývá Secured FTP (SFTP).**
- FTP obsahuje množství příkazů pro práci se souborovým systémem, jako např. načtení adresářové struktury, otevření/uzavření souboru, vytvoření/smazání souboru atd.
- Pro přenos souborů se používá často, má menší režii než HTTP protokol, viz dále.
- FTP explicitně otevří/zavírá spojení mezi klientem a serverem, které se udržuje neměnné (virtuální kanál).

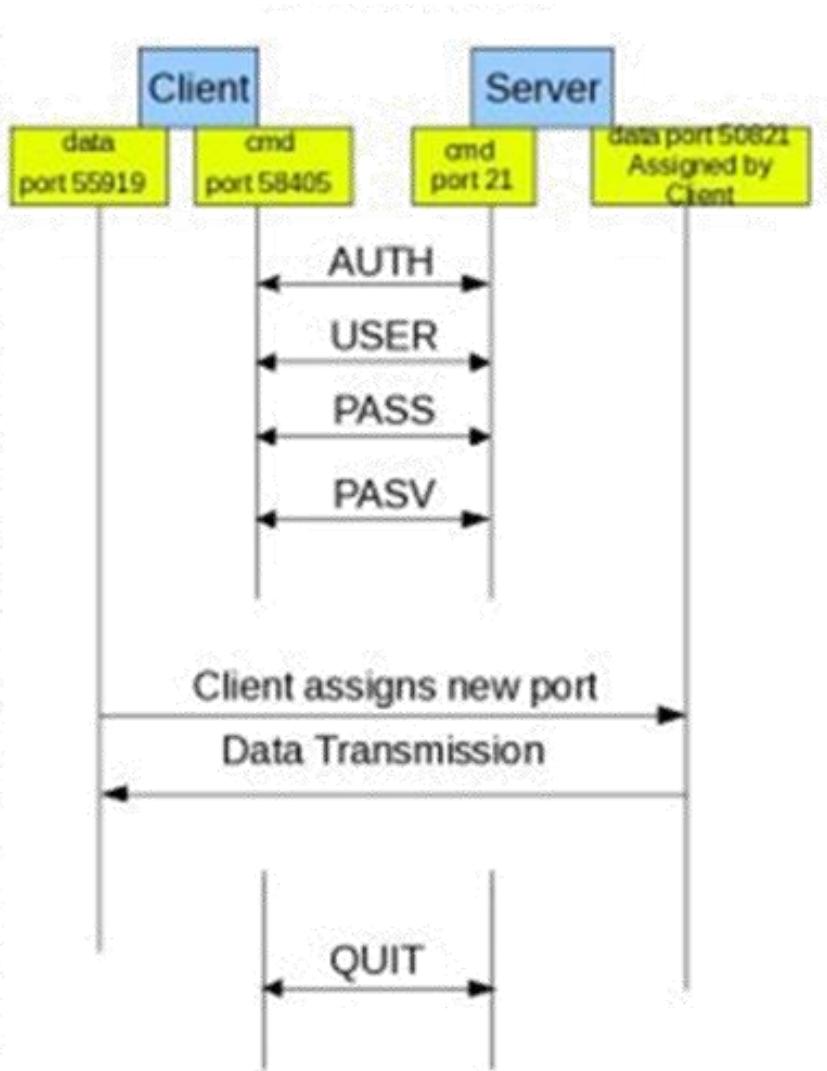
# FTP – aktivní mód



1. Klient zahájí se serverem autentizační proces (příkaz **AUTH**) tak, že otevře spojení na port 21. Přes vytvořené spojení se zasílají příkazy.
2. Klient zašle své jméno (příkaz **USER**) a heslo (příkaz **PASS**).
3. Server jméno a heslo ověří, v případě neúspěchu zašle klientovi informaci o chybných autentizačních údajích a klient se musí pokusit o autentizaci znovu.
4. **V případě úspěchu, klient navrhne serveru datový port** pro spojení, které bude přenášet data (příkaz **PORT**).
5. Server otevře spojení na navržený datový port klienta.
6. Dojde k přenosu dat.
7. Klient ukončí spojení se serverem (příkaz **QUIT**).

Převzato z <https://www.jspa.com/blog/bid/80512/active-v-s-passive-ftp-simplified>.

# FTP – pasivní mód



1. Klient zahájí se serverem autentizační proces (příkaz **AUTH**) tak, že otevře spojení na port 21. Přes vytvořené spojení se zasílají příkazy.
2. Klient zašle své jméno (příkaz **USER**) a heslo (příkaz **PASS**).
3. Server jméno a heslo ověří, v případě neúspěchu zašle klientovi informaci o chybných autentizačních údajích a klient se musí pokus o autentizaci znovu.
4. Klient požádá server o pasivní mód (příkaz **PASV**).
5. **Server navrhne klientovi svůj datový port** pro spojení, které bude přenášet data.
6. Klient si zvolí svůj datový port a otevře spojení na navržený datový port.
7. Dojde k přenosu dat.
8. Klient ukončí spojení se serverem (příkaz **QUIT**).

Důvodem pro zavedení pasivního módu bylo to, že klient se často nachází za směrovačem, který provádí překlad adres (**NAT**). V tomto případě není zaručeno, že server, může realizovat otevření spojení směrem ke klientovi.

Převzato z <https://www.jscape.com/blog/bid/80512/active-v-s-passive-ftp-simplified>.

# Elektronická pošta, používané protokoly



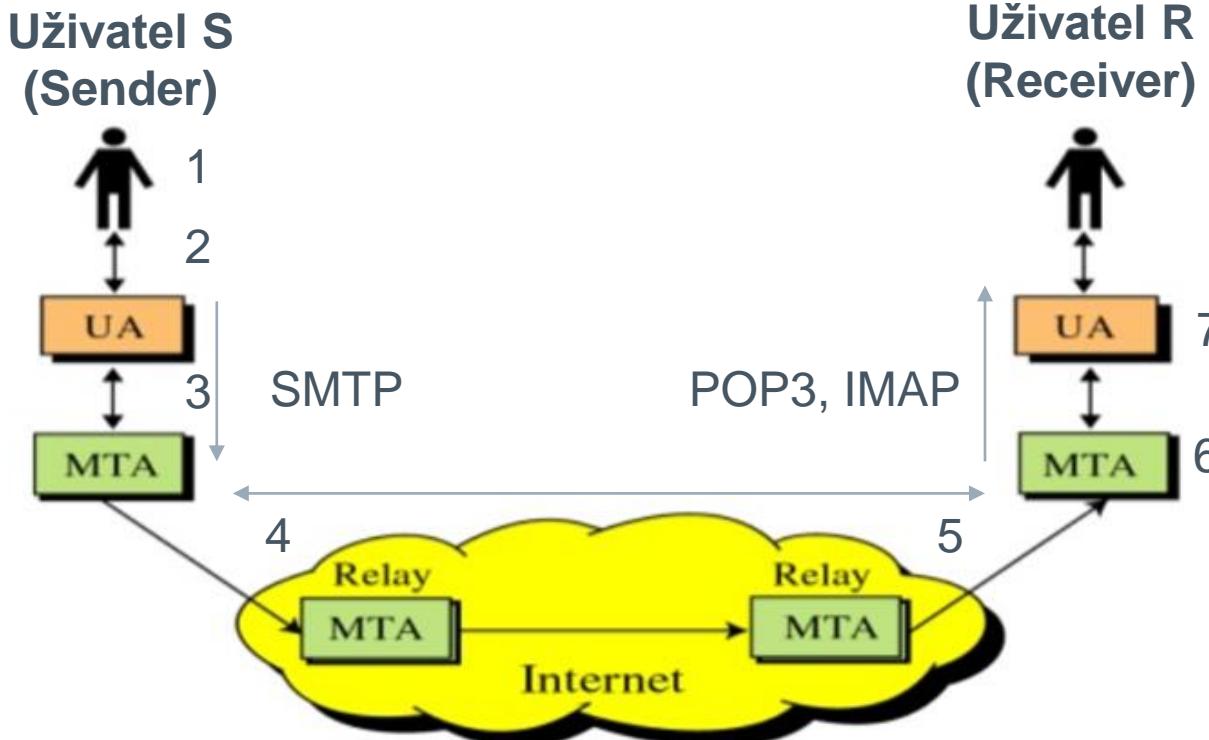
## Základní pojmy

- Emailová adresa = jméno\_uživatele@doménové\_jméno (fesljan@fit.cvut.cz)
- Mail Transfer Agent (MTA) = program běžící na poštovním serveru, který odesílá či přijímá elektronickou poštu (např. Postfix, Dovecot, MS Exchange).
- MTA relay (forward) = server přístupný z Internetu, který plní funkci MTA. Používá se na ochranu MTA před napadením útočníky (nachází se v demilitarizované zóně).
- User Agent (UA) = program, běžící v zařízení uživatele, který načítá poštu z mailboxu (poštovní schránky = souboru s obsahem pošty) na poštovním serveru.

## Protokoly používané pro doručování el. pošty:

- Simple Mail Transfer Protocol (SMTP) = protokol určený pro přenos emailů mezi poštovními servery, popřípadě odesílání emailů mezi UA a MTA, používá se cílový port 25.
- POP3 (port 110), POP3s (port 995), Internet Mail Access Protocol (IMAP, port 143), IMAPs (port 993) = protokoly určené pro čtení pošty prostřednictvím UA a MTA.

# UA, MTA a MTA Relay

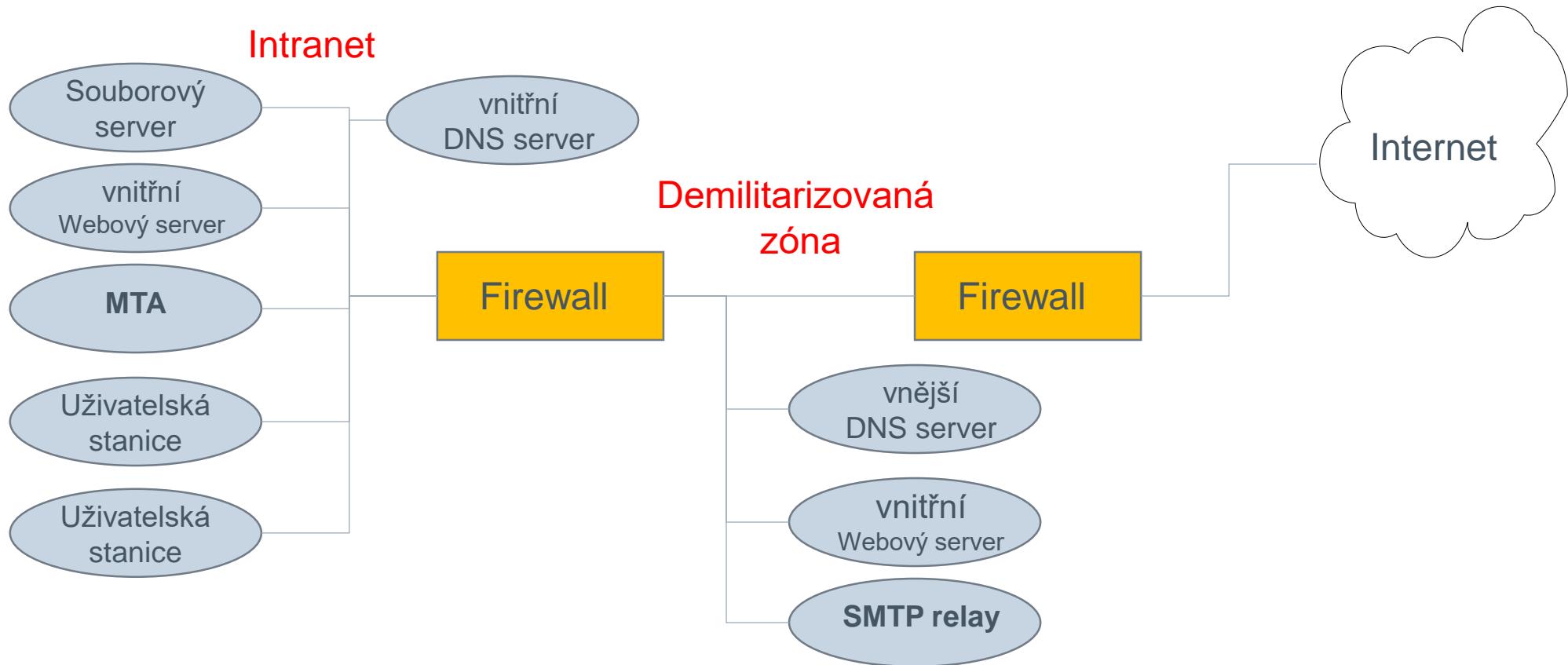


1. Uživatel S chce odeslat email uživateli R.
2. A napíše email prostřednictvím svého UA, který doručí email MTA.
3. MTA odešle email uživateli R prostřednictvím svého MTA Relay.
4. MTA Relay, který zajišťuje přeposílání pošty pro uživatele S, email přijme.
5. MTA Relay doručí poštu příslušnému MTA, který uchovává emaily uživatele R.
6. MTA uloží email do mailboxu.
7. UA uživatele S kontaktuje své MTA a načte obsah mailboxu.

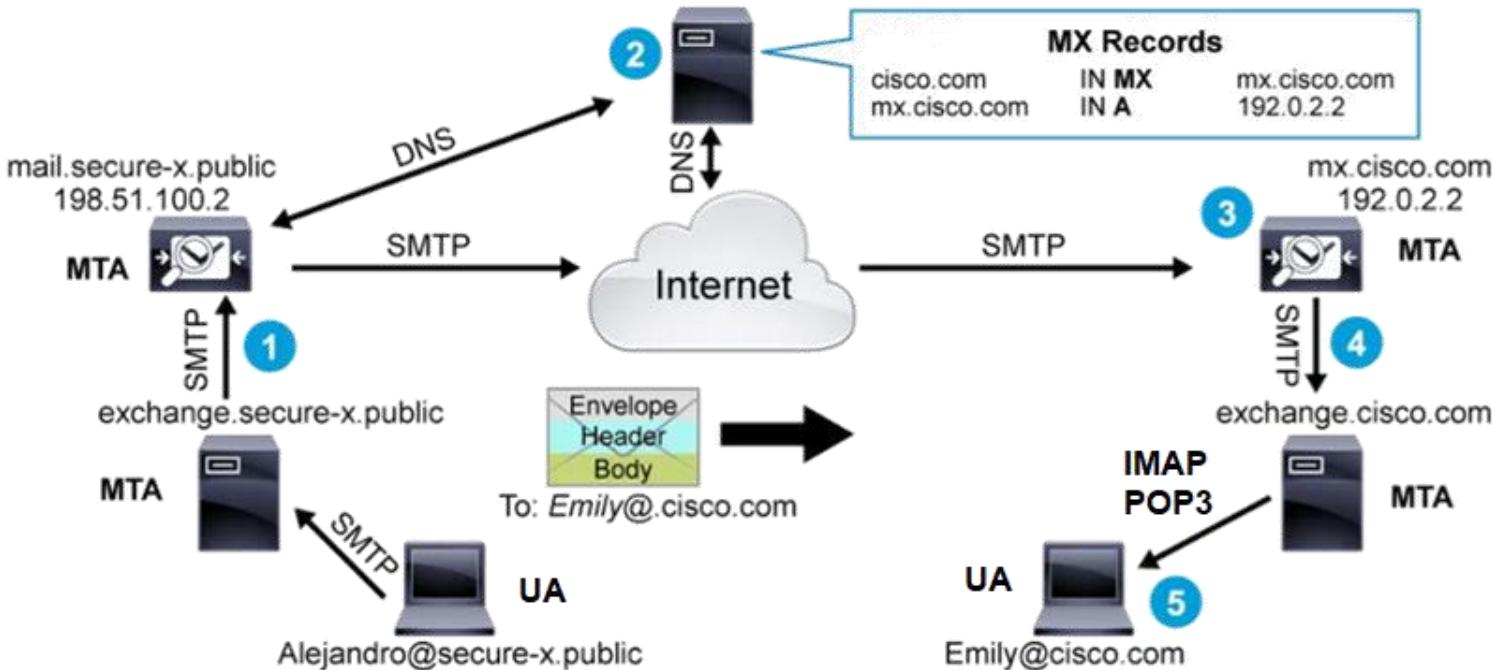
Převzato z <https://www.javatpoint.com/simple-mail-transfer-protocol>.

MTA Relay se nutně používat **nemusí**, MTA mezi sebou mohou komunikovat i přímo. Použití MTA Relay je ale výhodné zejména z bezpečnostních důvodů.

# MTA (SMTP) Relay – umístění v síti



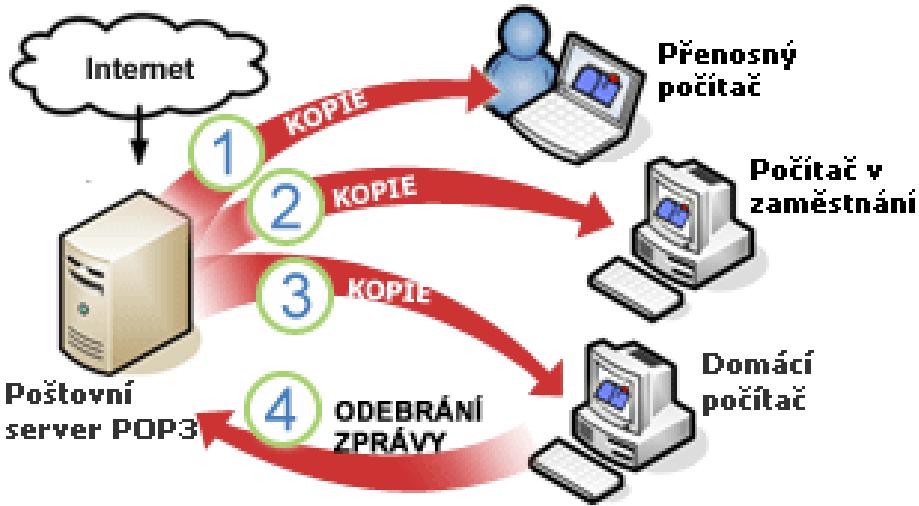
# Doručování emailů mezi poštovními servery



Převzato z [https://cjs6891.github.io/el7\\_blog/texts/cisco-ccna-cyber-ops-secfnd-6/](https://cjs6891.github.io/el7_blog/texts/cisco-ccna-cyber-ops-secfnd-6/).

1. MTA obdrží od UA prostřednictvím SMTP email, který má doručit na **specifickou emailovou adresu**.
2. Emailová adresa obsahuje v názvu doménové jméno. Pro dané doménové jméno se zjistí pomocí DNS protokolu **MX záznam** = IP adresa či doménové jméno poštovního serveru (PS), který se stará o doručování el. pošty pro danou doménu.
3. MTA doručí prostřednictvím protokolu email PS, který funguje jako MTA Relay.
4. P přepošle email pomocí SMTP MTA. MTA uloží email do svého mailboxu.
5. UA uživatele načte obsah mailboxu (viz další slajd).

# Čtení el. pošty, protokoly POP3 a IMAP – srovnání



Prostřednictvím protokolu **POP3** kopírují UA na klientských počítačích obsah stejného mailboxu. Emaily existují ve více kopiích. Obsah mailboxů uložených na různých PC, může být odlišný, mazání zprávy lokálně či na poštovním serveru neovlivní dříve načtené kopie mailboxů .



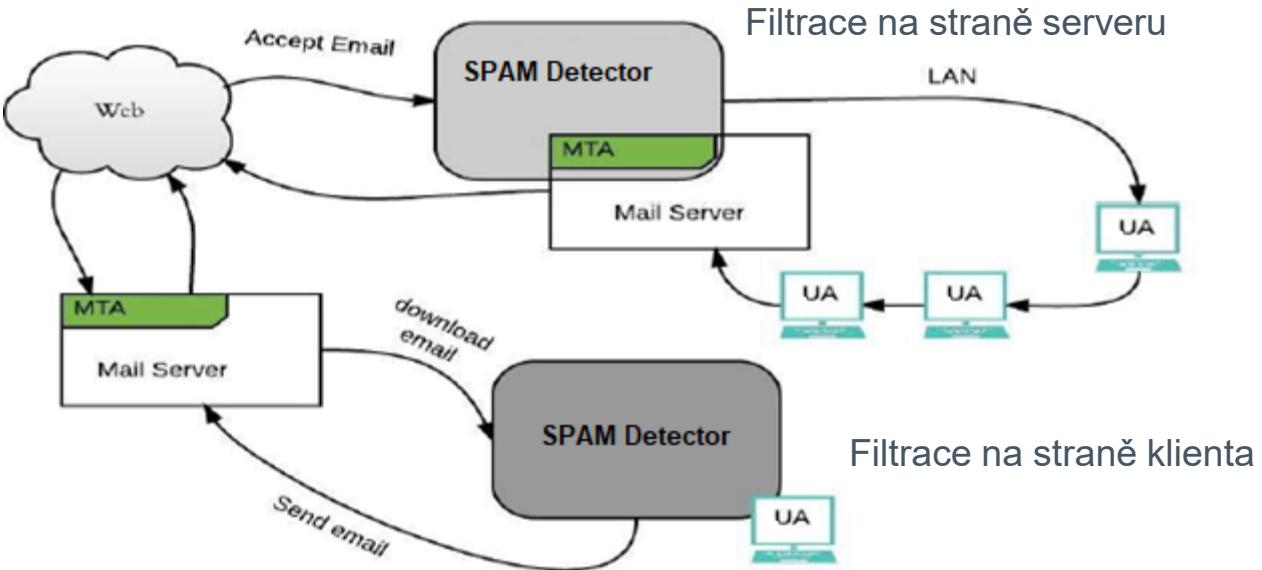
Převzato z <https://www.mailtrim.com/blog/pop3-vs-imap/>.

**IMAP** funguje hlavně jako “**projekce mailboxu**”, obsah mailboxu je trvale uložen na poštovním serveru. Stav na všech počítačích obsluhující stejný mailbox, je totožný. V případě modifikace obsahu na jednom počítači se změna se projeví shodně i na ostatních počítačích.

# Nevyžádaná elektronická pošta (SPAM)



**SPAM** = elektronická pošta, která je zasílána do mailboxu uživatele bez toho, že by vyjádřil zájem o její odběr. Součástí SPAMU může být i malware (viz přednáška 8, slajd 17).



Převzato z [https://www.researchgate.net/figure/Client-Side-and-Enterprise-level-Email-spam-filtering-system\\_fig3\\_332865507](https://www.researchgate.net/figure/Client-Side-and-Enterprise-level-Email-spam-filtering-system_fig3_332865507)

Pro **filtrování SPAMU** lze použít dvě řešení:

- Filtrování na straně serveru** : email, dříve než je uložen do mailboxu uživatele na serveru, je analyzován SPAM detektorem (modul MTA). Pokud je detekován jako SPAM je označen buď označen a uložen do mailboxu nebo smazán.
- Filtrování na straně klienta** : principiálně stejně jako na straně serveru, s tím rozdílem, že filtrace probíhá až v UA na počítači klienta.

# Hyper-Text Transfer Protocol (HTTP)



- Používá se pro přenos obsahu **webových** stránek.
- **Zprávy HTTP protokolu se přenáší se jako data protokolu transportní vrstvy.**
- **Používá se port 80.**
- Aktuálně existují 3 verze, každá má svá specifika.
  - **HTTP v1** je textový protokol, používá TCP protokol. Pro každou entitu na webové stránce, např. obrázek, video atd., se používá samostatné TCP spojení.
  - **HTTP v2** je binární protokol, který dokáže více spojení přenášet přes jedno TCP spojení.
  - **HTTP v3** je binární protokol, pro přenos dat se používá nepotvrzovaný protokol transportní vrstvy QUIC (autorem je spol. Google). Tento protokol je stále ve vývoji, je vyvíjen pro dosažení min. latence a maximální propustnosti.
- Operace HTTP: **GET, HEAD, POST, PUT, DELETE.**
- Funguje na principu **žádost → odpověď**
- **Je bezestavový, tzn. nic si nepamatuje.** Každá relace je chápána jako nová. Proto musí existovat metody, jak osvěžit předchozí informace → použití **cookies**, viz slajd 17.
- HTTP protokol se používá prostřednictvím webových prohlížečů.
- **Šifrovaná verze je HTTPS, port 443.**

# Žádosti a odpovědi v HTTP



## Žádosti:

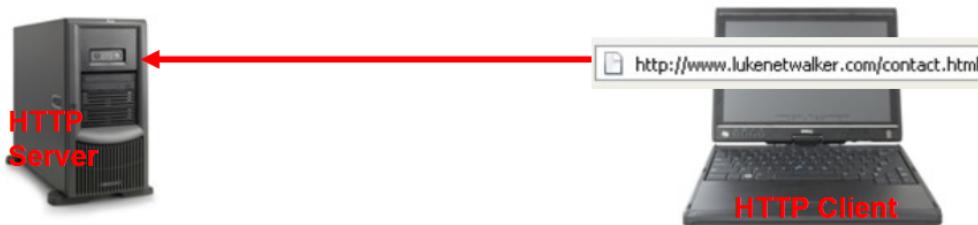
**GET** = načtení stránky ze serveru, **HEAD** = načtení jen hlavičky ze serveru, **POST/PUT** posílání dat (např. jméno a heslo) na serveru, **DELETE** – smazání informací na serveru.

## Odpovědi:

**200** – OK, **301** – permanentně přesunuto, **404** – chyba ve zprávě, **505** – vnitřní chyba serveru.

## Příklad žádosti:

```
GET /content.html / HTTP/1.1
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET
    CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.04506; InfoPath.1)
Host: www.lukenetwalker.com
Connection: Keep-Alive
```



## Příklad odpovědi:

```
HTTP/1.1 200 OK
Date: Fri, 22 Feb 2008 16:34:18 GMT
Server: Apache/2.0.52 (Red Hat)
Last-Modified: Thu, 15 Nov 2007 19:33:12 GMT
Content-Length: 15137
Connection: close
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```



Převzato z <https://networkencyclopedia.com/http-cookie/>.

# Princip cookies v HTTP



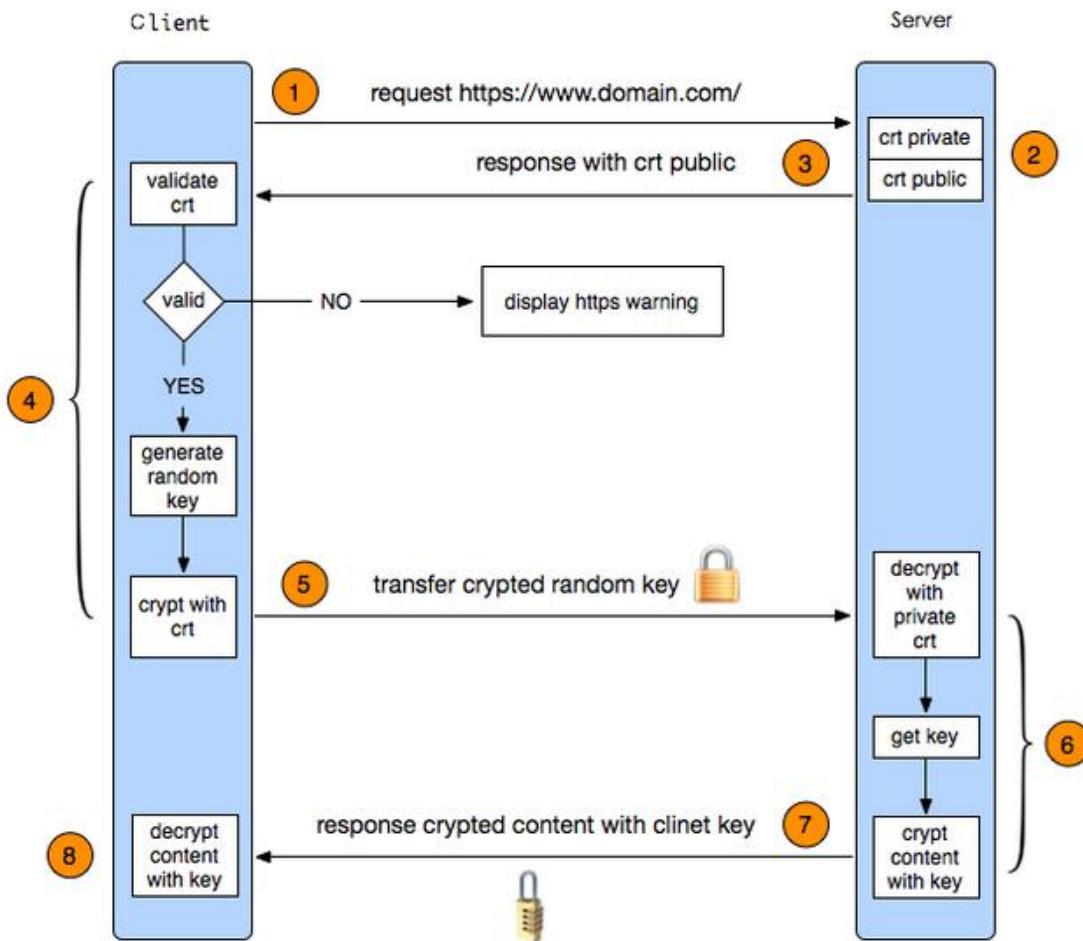
```
GET /jpeg/cap81/cam0.36705623.rgb888.enc HTTP/1.1
<information omitted>
Cookie: SLSPOTNAME5=Cowells; SLSPOTNAME4=Waimea%20Bay;
SLSPOTNAME3=Pipeline; SLSPOTNAME2=38th%20Ave%2E; SLSPOTNAME1=Cowells;
SLSPOTID5=4189; SLSPOTID4=4755; SLSPOTID3=4750; SLSPOTID2=4191;
SLSPOTID1=4189; OAX=R8bfwEbcU08ABCu; USER_ID=5551212 <not my actual user-id>; <rest of information omitted for brevity>
```



```
HTTP/1.1 200 OK
Date: Fri, 22 Feb 2008 19:00:15 GMT
Server: Apache/1.3.34 (Unix)
Last-Modified: Fri, 22 Feb 2008 18:51:47 GMT
ETag: "760a31-18ce-47bf19c3"
Accept-Ranges: bytes
Content-Length: 6350
Keep-Alive: timeout=15, max=257
Connection: Keep-Alive
Content-Type: text/plain <information omitted>
```

1. Webový server v případě prvního přístupu klienta vygeneruje klientovi identifikátor (ID)
2. Server uloží u klienta **cookie**. Cookie = soubor s nastavením parametrů konkrétního klienta pro daný web, např. ID klienta.
3. Při další žádosti od klienta, která obsahuje informace z cookie, webový server upraví obsah odpovědi dle konkrétního ID klienta. Využívá se pro personalizaci zasílaných dat (reklama, obsah či nabídky).

# Protokol HTTPS, princip zabezpečného spojení



1. Klient pošle žádost o načtení stránky.
2. Server zašle klientovi svůj certifikát.
3. Klient certifikát ověří, pokud nalezne problém vypíše varování.
4. Klient vygeneruje náhodný šifrovací klíč (NŠK).
5. Klient odešle serveru zprávu obsahující NŠK, kterou zašifruje veřejným klíčem serveru.
6. Server dešifruje zprávu svým privátním klíčem a získá NŠK.
7. Server pomocí NŠK zašifruje obsah webové stránky (data) a odešle je klientovi.
8. Klient dešifruje data a získá obsah webové stránky.

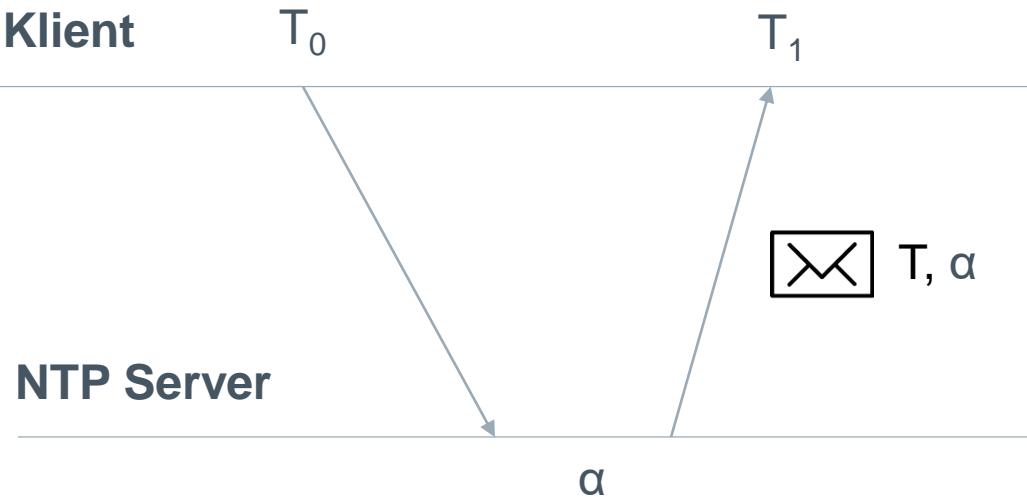
Převzato z <https://programmer.group/https-principle-and-okhttp-support-for-https.html>

# Network Time Protocol (NTP)



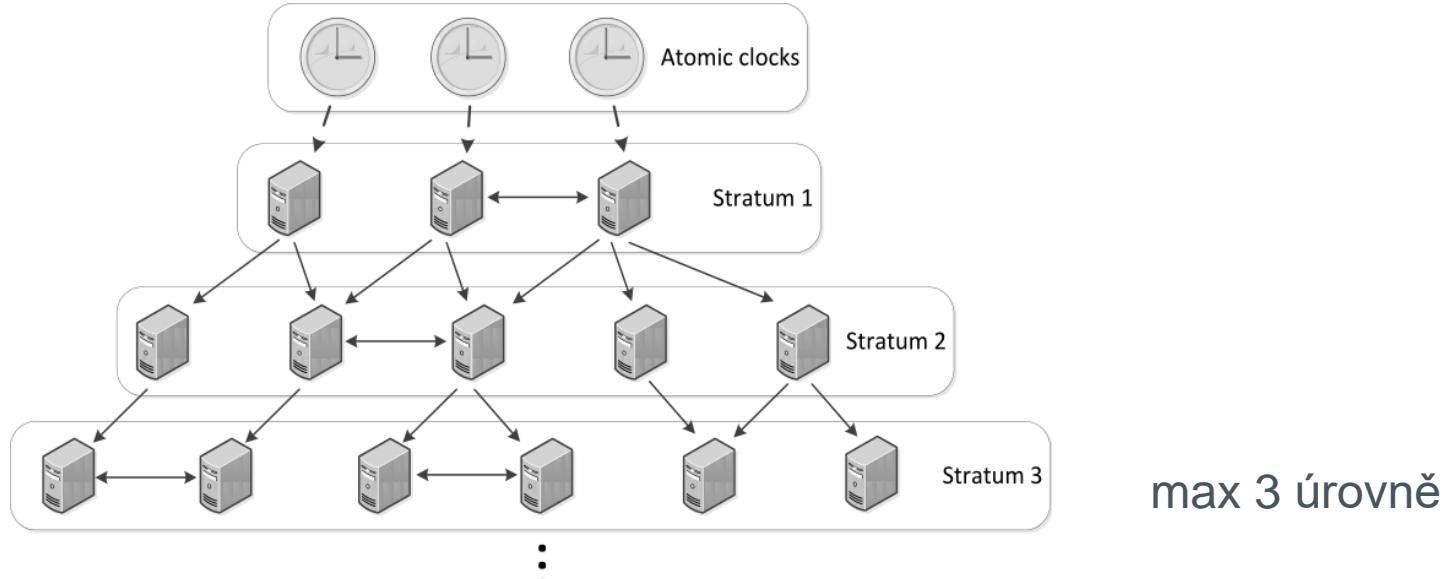
- NTP protokol slouží pro synchronizaci času, nastavení systémových hodin.
- Zprávy NTP protokol se přenáší jako data protokolu **UDP**.
- Běžně se pro posílání zpráv používá na straně serveru port 123.

## Christianův algoritmus



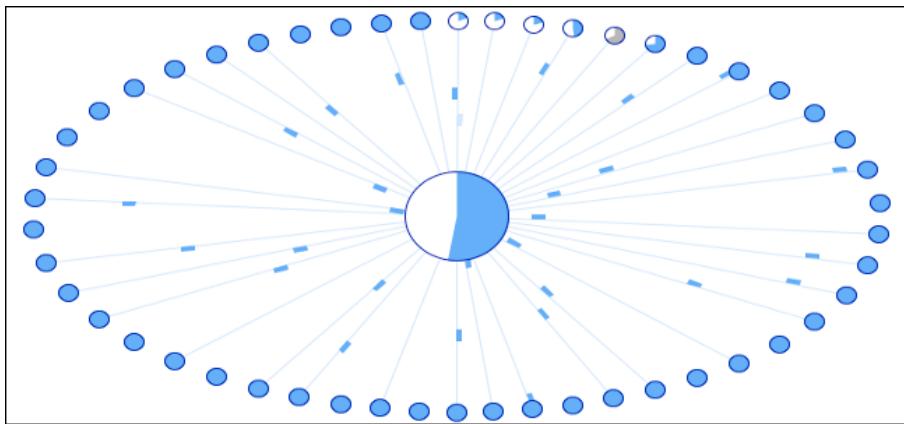
1. Klient v čase  $T_0$  vyšle požadavek na zjištění času.
2. Server přijme požadavek.
3. Server zpracuje požadavek za čas  $\alpha$  a zašle klientovi zprávu s časem  $T$  a dobou zpracování  $\alpha$ .
4. V čase  $T_1$  klient od serveru obdrží zaslanou zprávu.
5. Klient nastaví nový čas jako  $T_C = T + (T_1 - T_0 - \alpha) / 2$ .

# Network Time Protocol (NTP) – Hierarchie Serverů



- Nejpřesnější údaje o času lze získat z **atomových hodin**. Servery, které komunikují přímo s atomovými hodinami, se označují jako **Stratum 1**.
- Stratum 1 mají informace o času zkreslené jen minimálně, jelikož jsou přímo připojené k atomovým hodinám (řádově mikrosekundy).
- Servery, které komunikují přímo se servery Stratum 1, se označují jako Stratum 2.
- Servery Stratum 2 oproti serverům Stratum 1 mají **informace o času zkreslené již více**, jelikož při přenosu času reálnými linkami dochází ke zpoždění. Zpoždění se následně projeví **nepřesností v určení času**.
- Každá další úroveň Stratum serverů zanáší do určení času další nepřesnosti kvůli zpoždění. Dokud je nepřesnost v řádu desítek milisekund, nemá na většinu síťových služeb vliv.

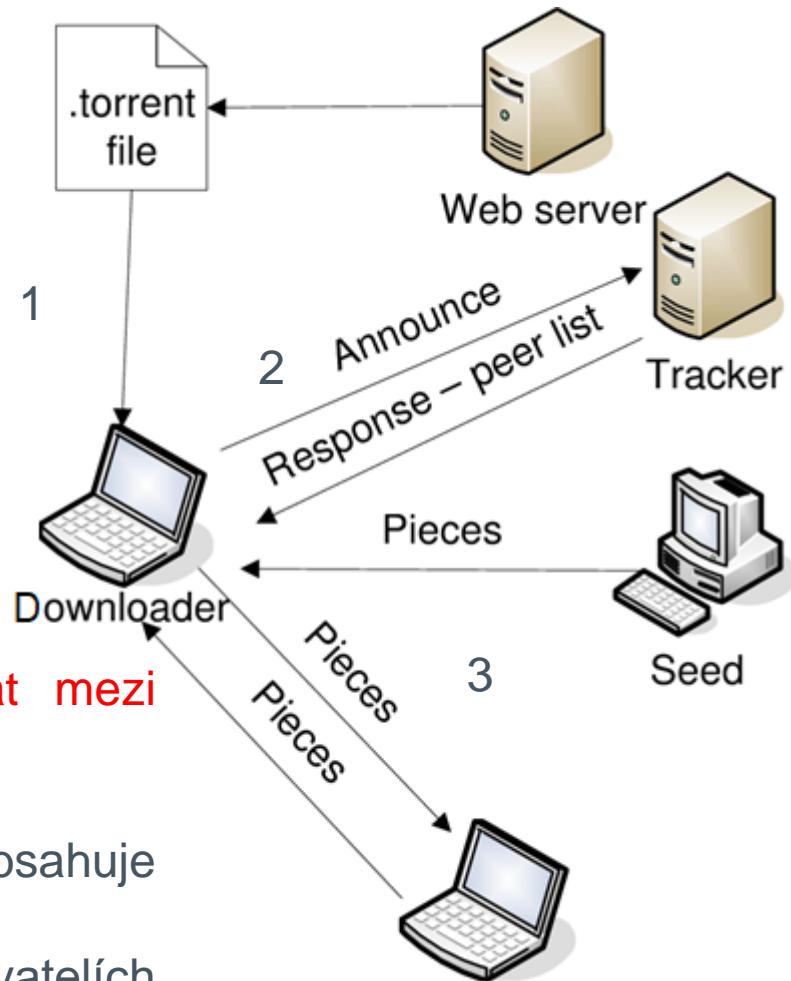
# BitTorrent protokol



Převzato z [https://www.researchgate.net/figure/BitTorrent-network\\_fig4\\_301204987](https://www.researchgate.net/figure/BitTorrent-network_fig4_301204987)

Důvodem vzniku tohotu protokolu je realizace sdílení dat mezi uživateli napřímo (P2P = peer to peer).

1. Uživatel X si “odněkud” stáhne “.torrent soubor”, který obsahuje informace o jednotlivých částí nějakého většího souboru.
2. X kontaktuje (Announce) **tracker**, který má informace o uživatelích (Seed), kteří dané části souboru mají a tento seznam vrátí X.
3. Uživatel X naváže spojení s těmito uživateli a začne paralelně stahovat části souboru (Pieces) z různých zdrojů.



# VoIP prostřednictvím SIP protokolu



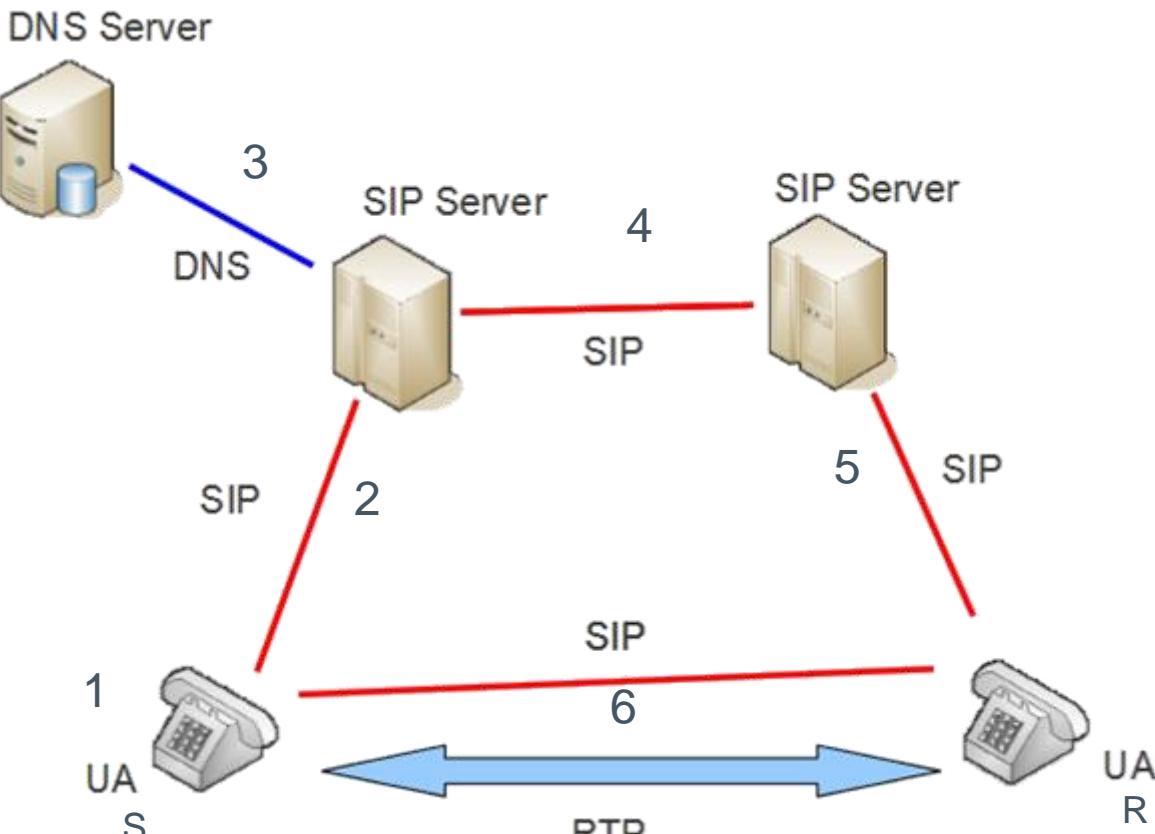
- **SIP = Session Initialization Protocol.**
- Jeho **úkolem je navázání spojení mezi stanicemi za účelem realizace hlasového hovoru.**
- SIP je jednoduchý otevřený protokol.
- SIP obsahuje pouze signalizační příkazy, pro zasílání dat obsahující hlas je nutné použít další protokol - RTP.
- **RTP = Real Time Protocol**, jednoduchý nepotvrzovaný protokol pro přenos hlasu.
- Pro RTP je zásadní aby obě strany komunikace byly dostupné mezi sebou pomocí směrování, tj. bez NATu.
- V případě NATu je nutné použít RTP proxy.
- **SIP má přímou vazbu na protokol DNS**, tzv. **SRV** zážnam v doméně.
- Hledání SIP serveru pro danou doménu je v principu totožné s hledáním serveru pro příjem el. pošty (MX zážnam v DNS).
- Komunikace v protokolu SIP není šifrována, ověření je bázi uživatelského jména a hesla.
- Při ověření údajů se však neposílá heslo, nýbrž jen jeho MD5 hash.

# Zařízení používaná v SIP architektuře



- **User Agent (UA)** = klientský program/zařízení umožňující VoIP služby.
- **SIP Server** = server, který se zprostředkovává SIP komunikaci pro skupinu UA patřící pod stejné doménové jméno.
- **SIP Proxy** = uzel, který vede komunikaci se SIP serverem, pokud není tento dosažitelný přímo.
- **SIP Registrátor (registrar)** – uzel, který odpovídá na žádosti přihlášení k SIP serveru.
- **Přesměrovací server (redirect)** – server, který přesměrovává žádosti pro registraci na jiný server (typický kód 3XX).
- **SIP Gateway** – místo, kde dochází k napojení na jiné operátory (např. veřejnou telekomunikační síť).

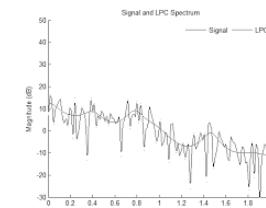
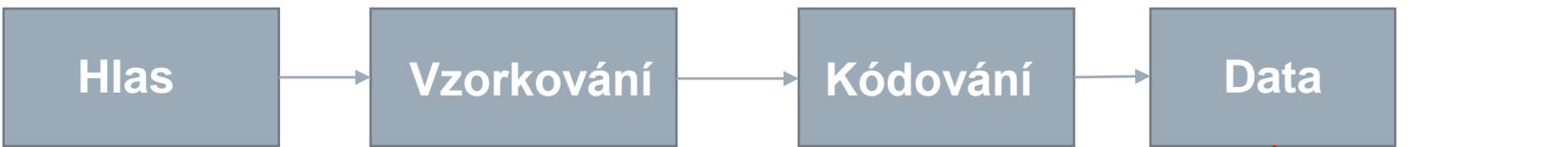
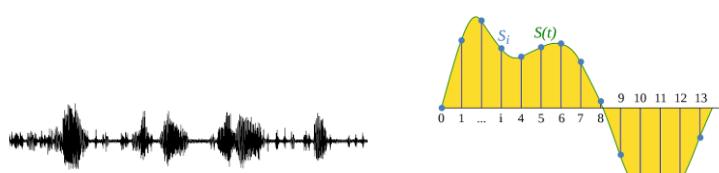
# SIP architektura a sestavení hovoru



1. UA S chce komunikovat s UA R **prostřednictvím VoIP**.
2. UA S se vyšle žádost na svůj SIP server.
3. SIP server od DNS serveru v položce **SRV záznamu** pro doménové jméno UA B zjistí doménové jméno a IP adresu SIP serveru, který zprostředkovává komunikaci s UA R.
4. SIP server UA S se obrátí s žádostí na SIP server UA R.
5. SIP server UA R se obrátí s žádostí na zahájení hovoru na UA R.
6. UA S a UA R mezi sebou spustí RTP protokol a začne přenos hlasu.

Převzato z [https://www.oreilly.com/library/view/the-ims-ip/9780470019061/9780470019061\\_sip\\_architecture.html](https://www.oreilly.com/library/view/the-ims-ip/9780470019061/9780470019061_sip_architecture.html)

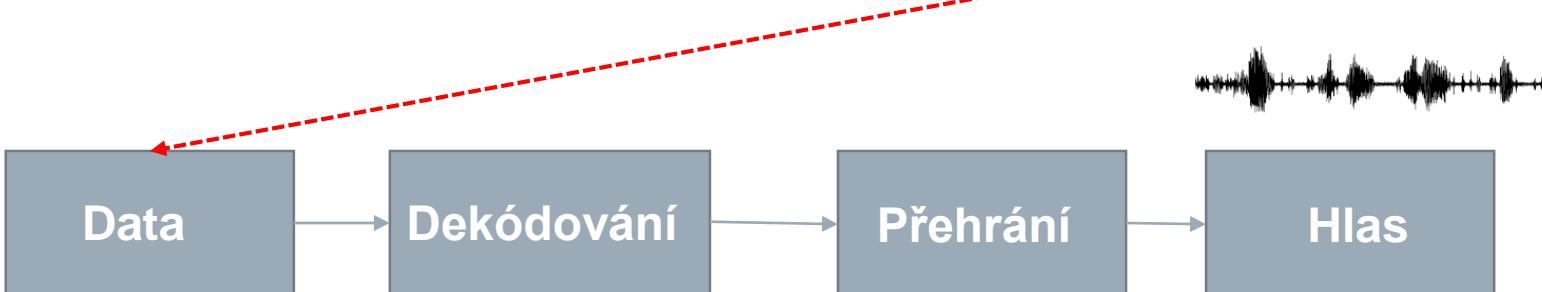
# RTP – komunikační schéma



Vysílací strana



Počítačová síť



Přijímací strana

# Počítačové sítě

11. Přednáška – Základy bezdrátových sítí

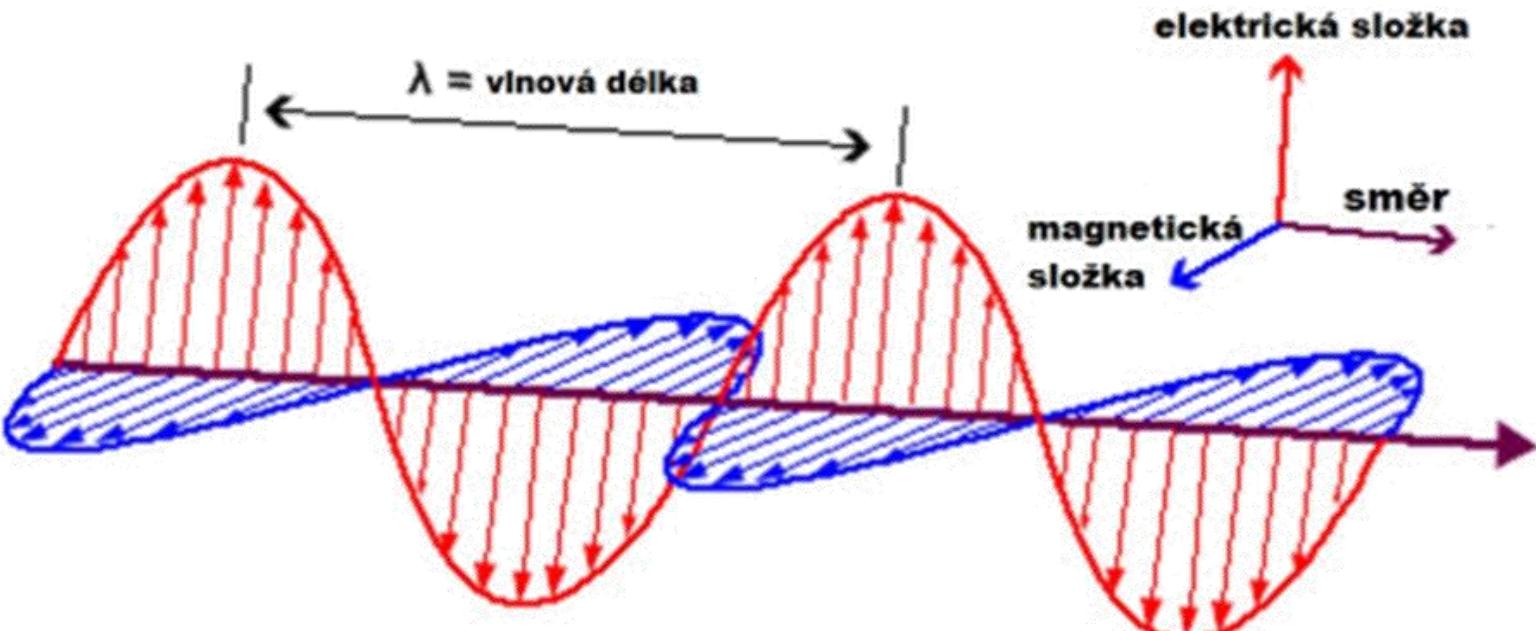


# Bezdrátové sítě a motivace jejich využití



- Použité přenosové médium je **vzduch či vakuum**.
- Pro přenos dat využívají **antény**.
- Stanice bezdrátových sítí mohou být **mobilní**.
- **Komunikační kanály (KK)** = komunikační linky bezdrátových sítí.
- KK jsou **méně odolné vůči rušení** v porovnání s linkami metalických či optických sítí.
- Vlivem fyzikálních limitů KK nedosahují stejných propustností jako linky kabelových či optických sítí.
- Výstavba bezdrátových sítí je velmi rychlá a jednoduchá oproti metalickým či optickým sítím.
- Existují **různé implementace** (dle určení)
  - Wifi (přenos dat, přístup k Internetu)
  - Bluetooth (propojení zařízení místo kabelu)
  - Zigbee (propování, sběr dat v průmyslu)
  - Radio Frequency Identification (RFID)
- Možnosti využití bezdrátových sítí podléhají právním předpisům, které se pro různé země obecně liší.

# Elektromagnetické pole a vlny



Převzato z <https://elektro.tzb-info.cz/>

- **Elektromagnetická (elmag.) vlna** = šíření elektromagnetického pole prostorem v určitém čase.
- Elektromagnetická vlna se šíří prostorově ve dvou **navzájem kolmých rovinách**. Tyto roviny označujeme jako elektrickou a magnetickou složku elmag. vlny.
- Elektromagnetickou vlnu je možné popsát **vlnovou rovnicí**.
- **Vlnová délka ( $\lambda$ )** = vzdálenost 2 bodů, které kmitají ve fázi.
- Vlnová délka souvisí s frekvencí ( $f$ ) a rychlosťí světla  $f = \frac{c}{\lambda}$  [hz].

# Antény a jejich parametry, přenosový kanál



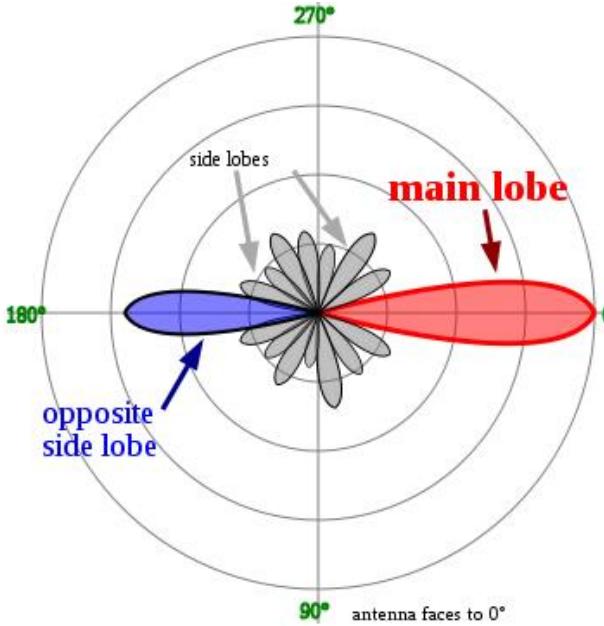
- **Antény** = prvky v bezdrátových sítích, které vysílají a přijímají elmag. vlny (signál).
- **Antény umožňují:**
  - určit směr šíření či příjmu elmag. vln.
  - zesílit intenzitu vysílaných či přijímaných elmag. vln.
- **Parametry antén:**
  - **Zisk antény** = úroveň zesílení intenzity elmag. vln po průchodu anténou.
  - **Pracovní pásmo** = frekvenční rozsah, pro který byla anténa navržena a má největší zisk.
  - **Směrovost antény** = směr vysílání či příjmu elmag. vln danou anténou, popisuje se vyzařovacím diagramem (viz slajd 6).
- Konstrukce antén jsou **fyzicky přizpůsobené pro různé frekvenční rozsahy** a z tohoto důvodu je nemožné zkonstruovat jedinou univerzální anténu, která by fungovala stejně pro řádově odlišné frekvenční rozsahy.
- **Přenosový kanál** = pracovní pásmo antény, na kterém se přenášejí data. Typickými parametry přenosového kanálu jsou: základní frekvence a šířka. Šířka přenosového kanálu souvisí přímo s jeho propustností.
- **Propustnost (někdy kapacita) přenosového kanálu** = počet bitů za vteřinu, které je možné přenosovým kanálem přenášet současně.

# Polarizace antény



- **Polarizace antény** = rovina, ve které se šíří elektrická složka elmag. vlny.
- Polarizace antény souvisí s polohou antény a může být tedy **vertikální či horizontální**.
- Složením vertikální a horizontální polarizace vznikne polarizace **kruhová**.
- Výhodou kruhové polarizace je to, že je odolnější vůči rušení oproti vertikální či horizontální polarizaci.

# Vyzařovací diagram (radiation pattern) antény



Převzato z [https://www.tutorialspoint.com/antenna\\_theory/antenna\\_theory\\_radiation\\_pattern.htm](https://www.tutorialspoint.com/antenna_theory/antenna_theory_radiation_pattern.htm)

- Pro umisťování více antén na stožárech je potřeba vědět, jakou část prostoru daná anténa pokryje resp. zaruší.
- **Vyzařovací diagram (VD)** = rovinné znázornění vyzařovacích směrů dané antény. Pro kvantifikaci směru šíření se používají vyzařovací úhly.
- **Hlavní lalok (main lobe)** = primární vyzařovací směr antény, značí oblast kterou anténa pokrývá
- **Opačný lalok (opposite side lobe)** = opačný vyzařovací směr antény, značí oblast kterou anténa ruší.
- **Postranní lalok** = směr která anténa pokrývá či ruší, který vznikl její nedokonalou konstrukcí.
- Vyzařovací diagramy jsou 2 (pro horizontální a vertikální rovinu).



# Druhy antén dle tvaru vyzařovacího diagramu

- **Všesměrové (360°)**
  - Používají se na připojení klientských stanic okolo určitého místa (10ky a 100ky metrů).
- **Sektorové (30-180°)**
  - Používají se na připojení klientských stanic v místech, ve kterých je vyšší rušení a není možné používat všesměrové antény (1ky kilometrů).
- **Směrové (10-30°)**
  - Používají se na propojování hlavních částí (páteřních oblastí) bezdrátových sítí na kratší až střední vzdálenosti (100ky metrů až 1ky km).
- **Úzcesměrové (< 5°)**
  - Používají se na propojování hlavních částí na dlouhé vzdálenosti (10ky km).

# Druhy antén dle typu konstrukce - FYI



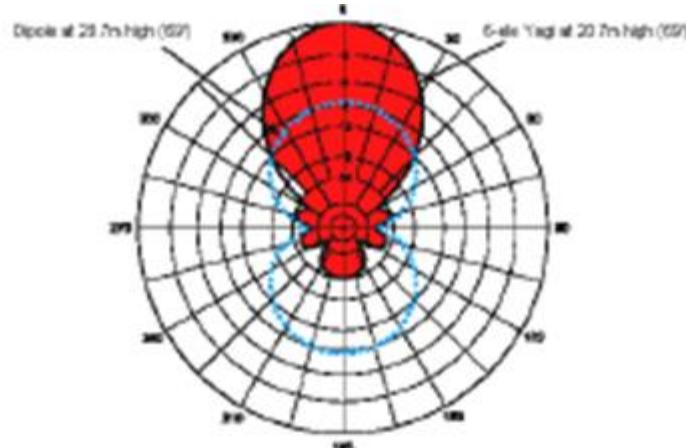
- Zig-Zag antény (cik-cak)
- Yagi antény
- Paraboly
- Spirálové antény

# Yagi antény - FYI

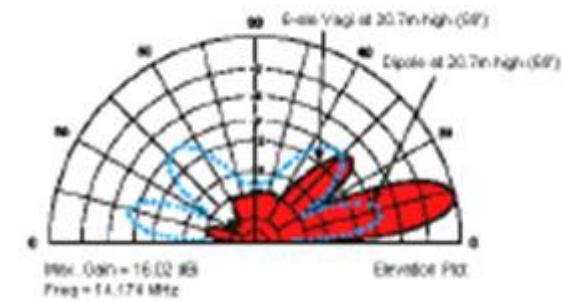
## Konstrukce



Vyzařovací diagram



Horizontální



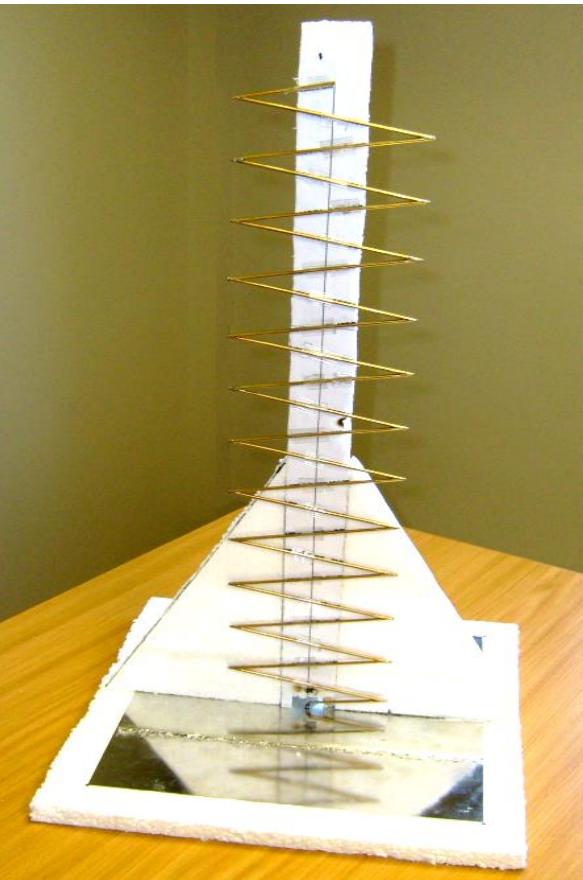
Vertikální

Převzato z [https://www.changpuak.ch/electronics/yagi\\_uda\\_antenna\\_DL6WU.php](https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php)

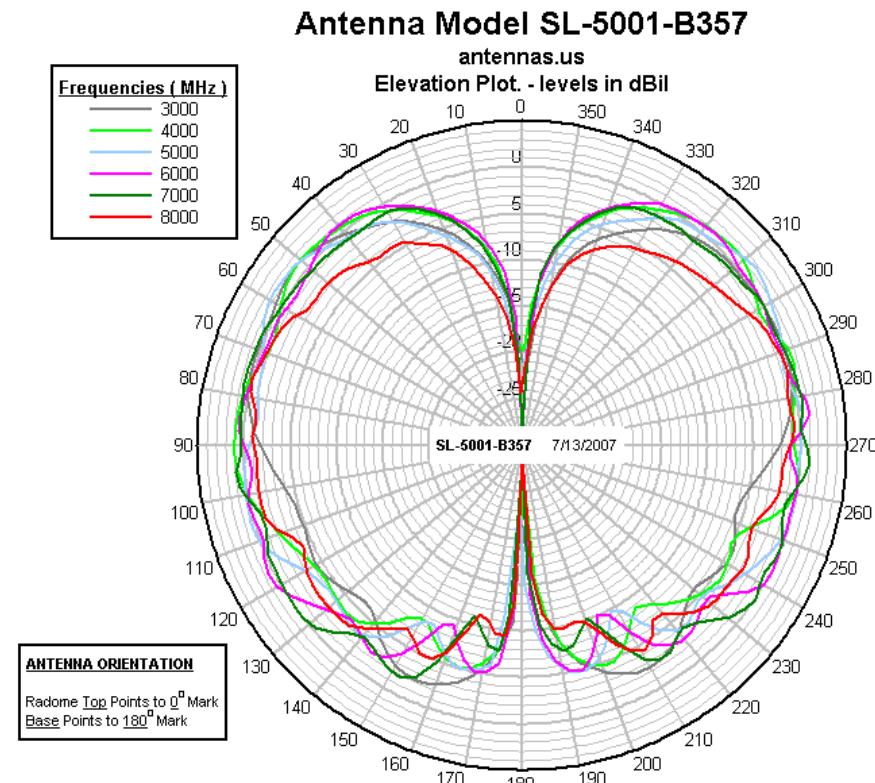
# Zig-Zag antény - FYI



## Konstrukce



## Horizontální vyzařovací diagram



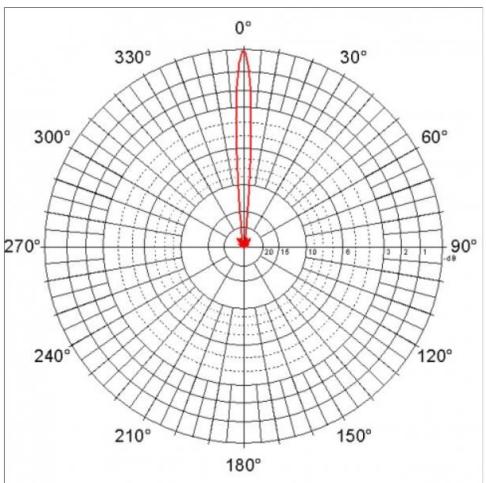
Převzato z [https://www.changpuak.ch/electronics/yagi\\_uda\\_antenna\\_DL6WU.php](https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php)

# Parabolické antény - FYI

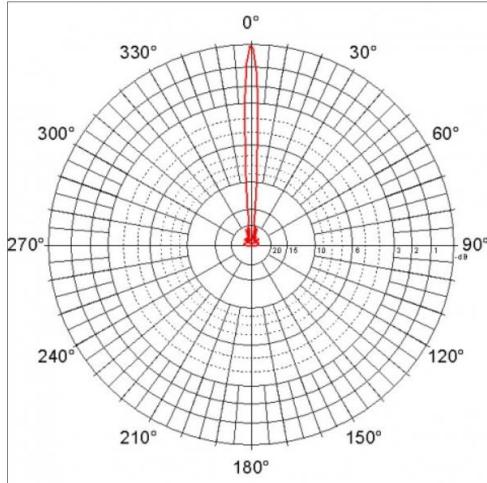


Konstrukce

Vyzařovací diagram



Vertikální



Horizontální



Převzato z <https://cz.jirous.com/anteny-5ghz/jrc-24/>

# Spirálovité antény - FYI



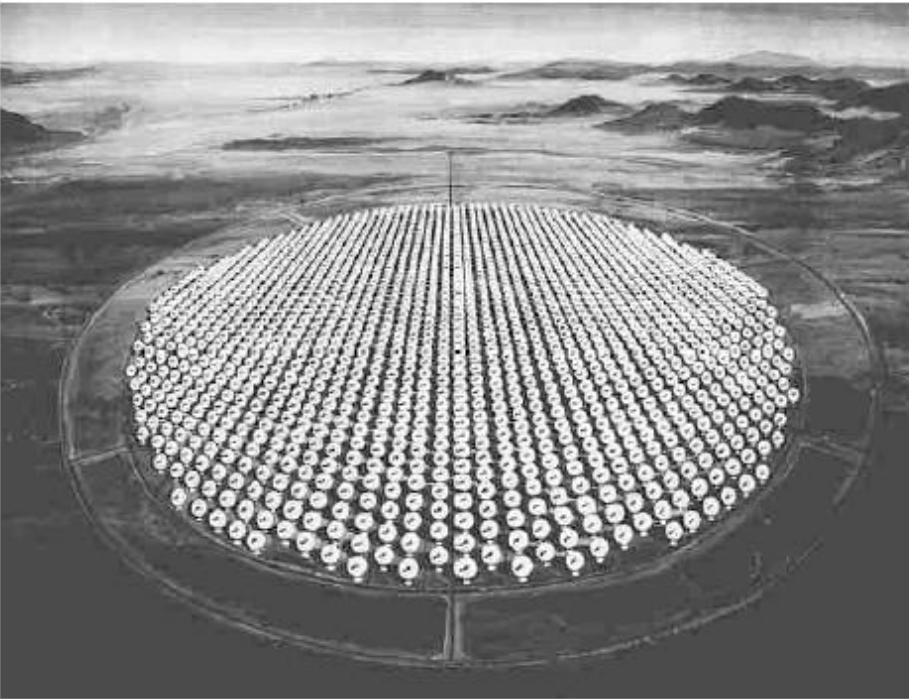
Konstrukce



Převzato z [https://www.changpuak.ch/electronics/yagi\\_uda\\_antenna\\_DL6WU.php](https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php)

Tyto antény dokáží vysílat elmag. vlny v obou rovinách – kruhová polarizace (viz slajd 5).

# Anténní pole



Převzato z <https://www.elektormagazine.com/news/novel-phasedarray-antenna-is-an-order-of-magnitude-better-than-alternative-solutions>

**Anténní pole** = soustava antén, která spojuje jednotlivé samostatné antény v jeden homogenní celek (jednu anténu). Celkový vyzařovací diagram má tvar součtu jednotlivých vyzařovacích diagramů konkrétních antén.

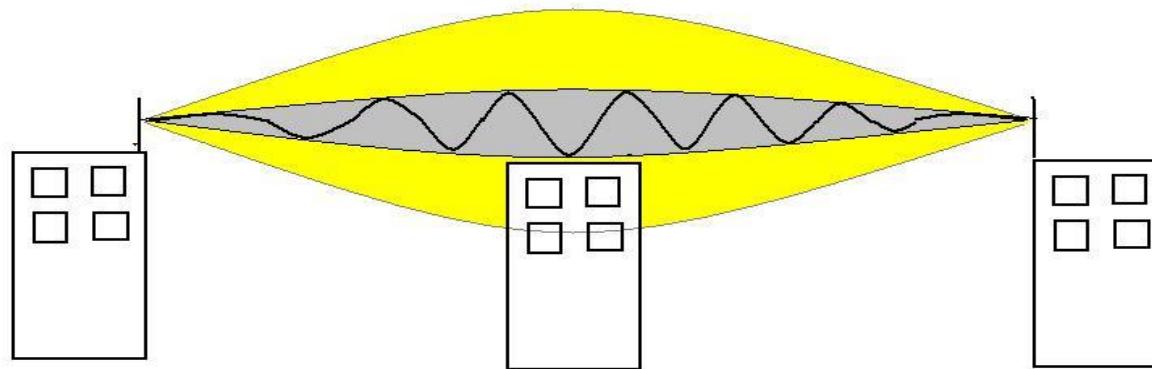
# Fresnelova zóna



## Situace 1:



## Situace 2:



Převzato <http://hackairs.wz.cz/index.php?stranka=wifi>

- Standardním **jménem požadavkem** pro vytváření bezdrátových sítí je **přímá viditelnost** mezi anténami (Situace 1). Reálný signál se však nešíří po přímce, ale ve **Fresnelově zóně** (žlutá oblast).
- Pokud je překážka ve Fresnelově zóně (Situace 2), znamená to vždy pokles intenzity signálu, ačkoli mezi místy existuje přímá viditelnost.

# Multiplex v bezdrátových sítích



- **Multiplex** = současné sdílení stejného přenosového média více stanicemi (viz 2. přednáška, slajd 6.)
- Nejvíce používané typy jsou tyto:
  - **Frekvenční multiplex (FDM)**
  - **Časový multiplex (TDM)**
  - **Prostorový multiplex (SDM)**
  - **Ortogonalně frekvenčně dělený multiplex (OFDM)**
  - **Skákové frekvenční spektrum (FHSS = Frequency Hopping Spread Spectrum)**

# Ortogonalně frekvenčně dělený multiplex (OFDM)



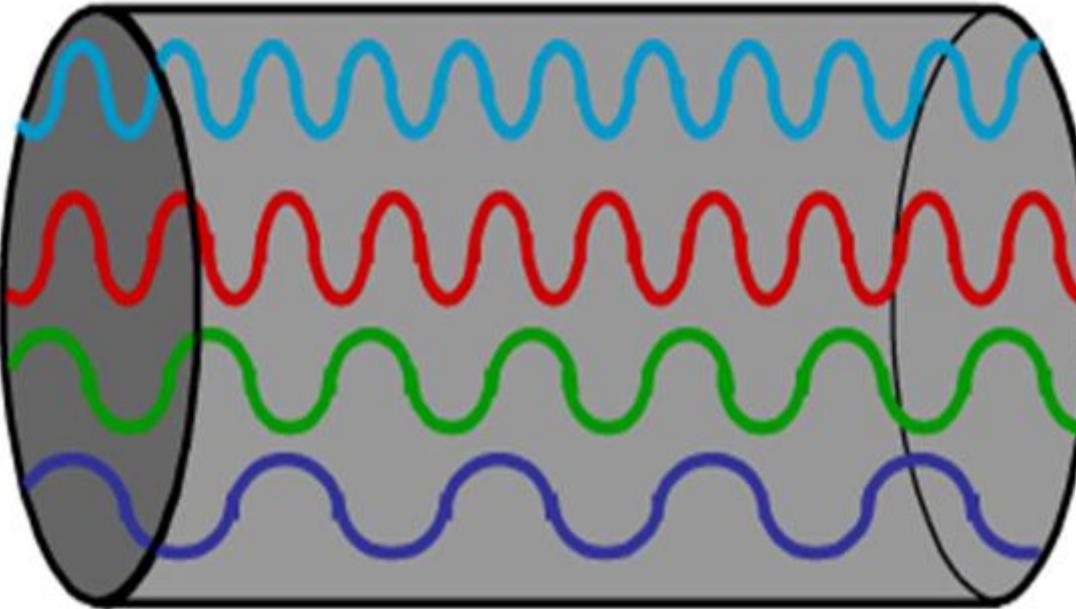
Podkanál 1

Podkanál 2

Podkanál 3

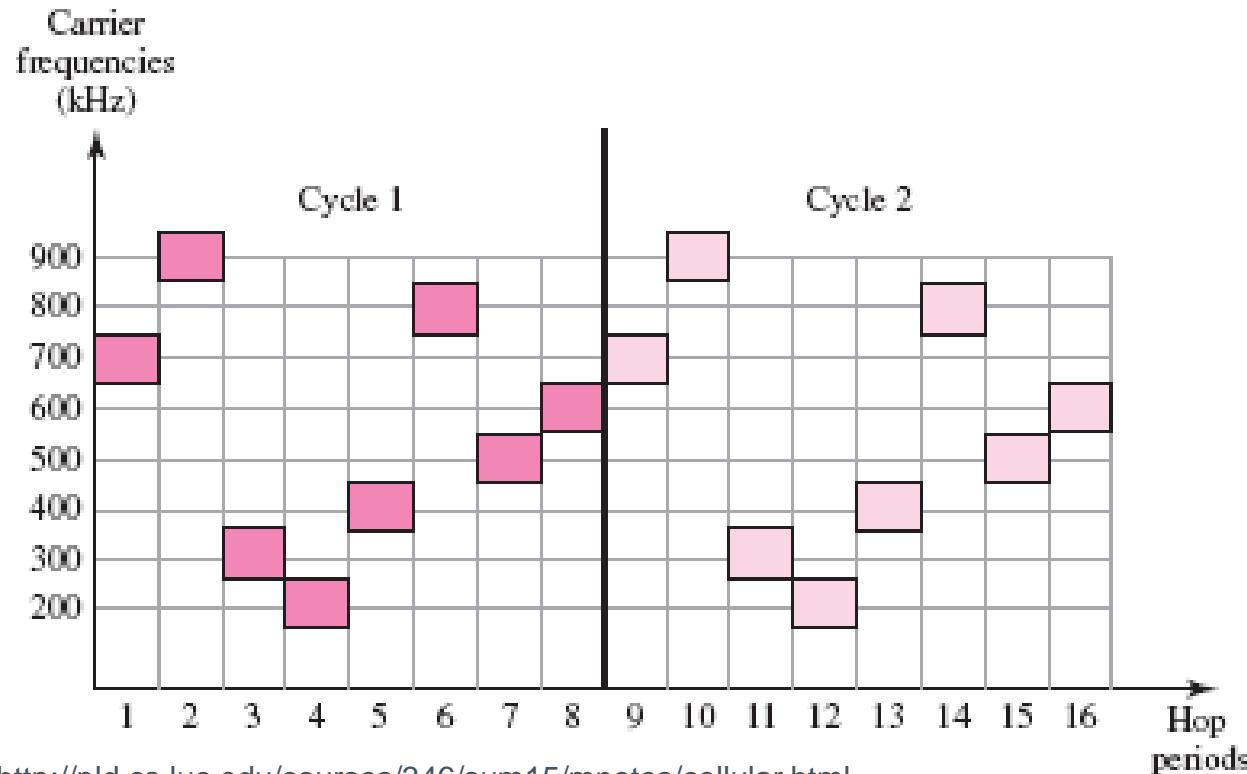
Podkanál 4

Přenosový kanál



- **Přenosový kanál** = množina za sebou spojených stejně širokých podkanálů. Celková šířka přenosového kanálu se dělí rovnoměrně mezi podkanály.
- Přes jednotlivé podkanály se data přenáší samostatně. V závislosti na počtu podkanálů se navýšuje propustnost kanálu, avšak na úkor odolnosti rušení jednotlivých podkanálů, jelikož užší pásmo je snadněji zarušitelné.

# Frequency hopping spread spectrum (FHSS)



Převzato z <http://pld.cs.luc.edu/courses/346/sum15/mnotes/cellular.html>

- Přenosový kanál se rozdělí na velké množství podkanálů, mezi kterými se přelaďuje podle předem stanoveného schématu.
- **Výhodou této metody je velká odolnost vůči rušení**, jelikož podkanály se využívají jen na velmi krátké časové intervaly a není snadné je zarušit.
- **Nevýhodou této metody je nízká propustnost**, která souvisí s neustálým nutným přelaďováním.

# Modulace signálu

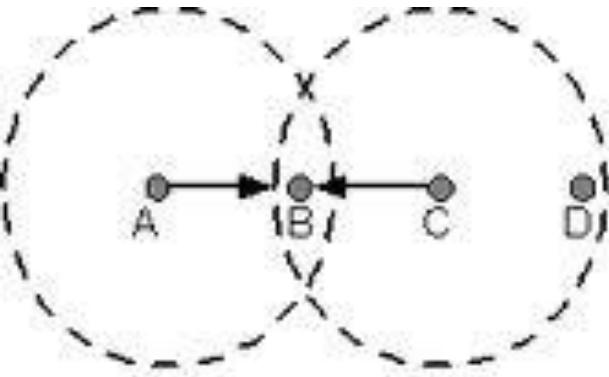


- Na úrovni fyzické vrstvy OSI modelu přenášejí data elmag. vlny.
- Každá vyslaná elmag. vlna může nést samostatnou informaci.
- Nejmenší a nejjednodušší přenášitelná datová jednotka je jeden bit.
- Pokud vhodně změníme tvar elmag. vlny, je možné přenášet prostřednictvím elmag. vlny více bitů.
- **Modulace** = přizpůsobení signálu, který nese datovou informaci do frekvenčního pásma antén, pro které jsou konstruovány.
- Existují různé druhy modulace
  - **Jednoduché**
    - Amplitudová, frekvenční, fázová.
  - **Složené**
    - Kombinují více jednoduchých modulací do sebe.
    - Umožňují přenášet více bitů v rámci jedné elmag. vlny najedou.
    - Quadratic Amplitude Modulation (**QAM**), Direct Sequence Spread Spectrum (**DSSS**)
- **Symbol** = datová jednotka, která je přenášena jedinou elmag. vlnou.
- **Modulátor** = zařízení, které provádí modulaci odesílaného signálu. Odesílaný signál nesoucí informaci modulátor přizpůsobí frekvenčnímu rozsahu použitých antén.
- **Demodulátor** = zařízení, které získá z přijímaného signálu původní datovou informaci.
- Obecně platí, čím složitější použitá modulace, tím menší odolnost vůči rušení.



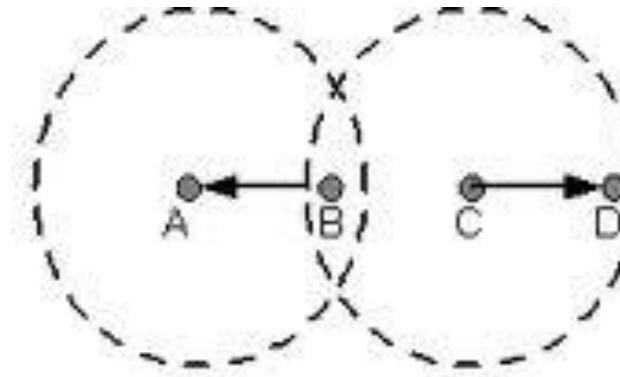
# Problém skrytého/vystaveného uzlu (Hidden/Exposed node problem)

Převzato z <https://www.geeksforgeeks.org/collision-avoidance-in-wireless-networks/>



Hidden node problem

- Uzel C ovlivňuje příjem uzlu B od A, ačkoli A a C nejsou navzájem dosažitelné.
- C je v tomto případě **skrytým** uzlem pro A.



Exposed node problem

- Uzel B chce komunikovat s uzlem A a současně uzel C s D.
- Ačkoli se A a D neslyší, současná komunikace není možná. B a C jsou **vystavenými** uzly.

# Počítačové sítě

11. Přednáška - IEEE 802.11b/g/n/ac/ax (Wi-Fi)



# IEEE 802.11 – přehled verzí



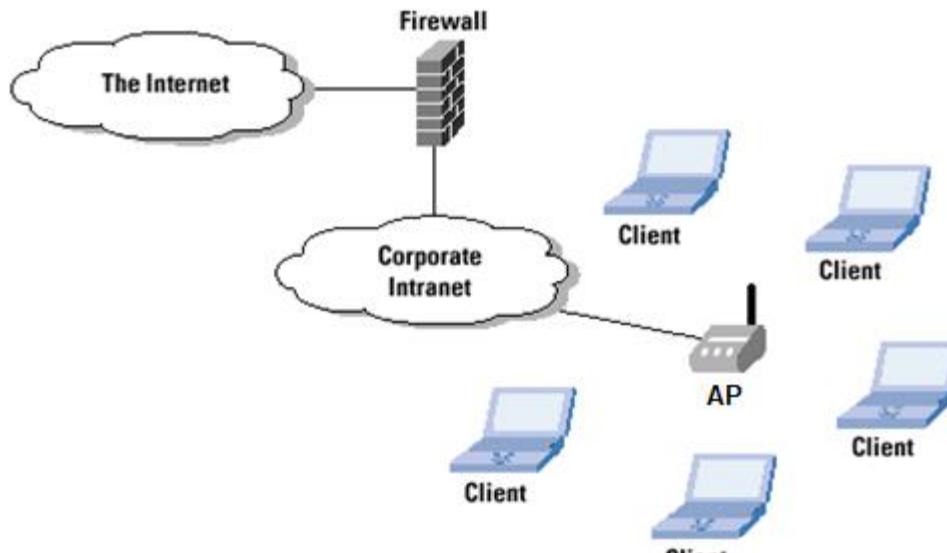
Standard	Označení	Rok vydání	Pásma [GHz]	Maximální rychlosť [Mbit/s]	Fyzická vrstva
původní IEEE 802.11	-	1997	2,4	2	DSSS a FHSS
IEEE 802.11a	Wi-Fi 1	1999	5	54	OFDM
IEEE 802.11b	Wi-Fi 2	1999	2,4	11	DSSS
IEEE 802.11g	Wi-Fi 3	2003	2,4	54	OFDM
IEEE 802.11n	Wi-Fi 4	2009	2,4/5	600	MIMO OFDM
IEEE 802.11y	-	2008	3,7	54	
IEEE 802.11ac	Wi-Fi 5	2013	5	3466,8	MU-MIMO OFDM
IEEE 802.11ad	-	2012	60	6757	
IEEE 802.11ax	Wi-Fi 6	2019	2,4/5/6	10530	MU-MIMO OFDMA

- IEEE 802.11 (Wi-Fi) patří do podobné skupiny sítí jako např. Ethernet IEEE 802.2. Někdy se IEEE 802.11 označuje také jako **bezdrátový Ethernet**.
- **Všechny verze IEEE 802.11 podporují přístupovou metodu CSMA/CA** (viz 2. přednáška slajd 12).
- Zařízení podporující určitou verzi IEEE 802.11 by dle normy měla podporovat i všechny verze předešlé. V praxi ale toto tvrzení vždy neplatí.

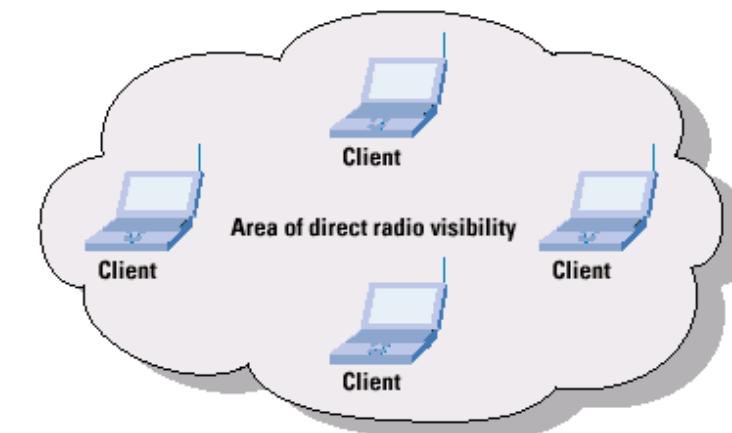
# IEEE 802.11 – topologie



- Z hlediska existence centrálního uzlu se sítě dle topologie dělí na:
  - **Infrastrukturní**
    - Centrální jednotka (AP = Access Point) řídí komunikaci mezi stanicemi.
  - **Adhoc**
    - Stanice spolu komunikují napřímo, bez AP.



Infrastrukturní síť



Adhoc síť

Převzato z [https://www.researchgate.net/figure/Ad-hoc-mode-vs-Infrastructure-mode-IEEE80211-introduced-many-types-of-the-Wi-Fi\\_fig1\\_316175326](https://www.researchgate.net/figure/Ad-hoc-mode-vs-Infrastructure-mode-IEEE80211-introduced-many-types-of-the-Wi-Fi_fig1_316175326)

# IEEE 802.11 rámec - detailně



- Network Allocation Vector (NAV) = synchronizace



Data protokolů vyšších vrstev

- MTU maximálně 2048 bajtů
- Max 256 bajtů pro hlavičku protokolu vyšší vrstvy

# Druhy rámců v IEEE 802.11

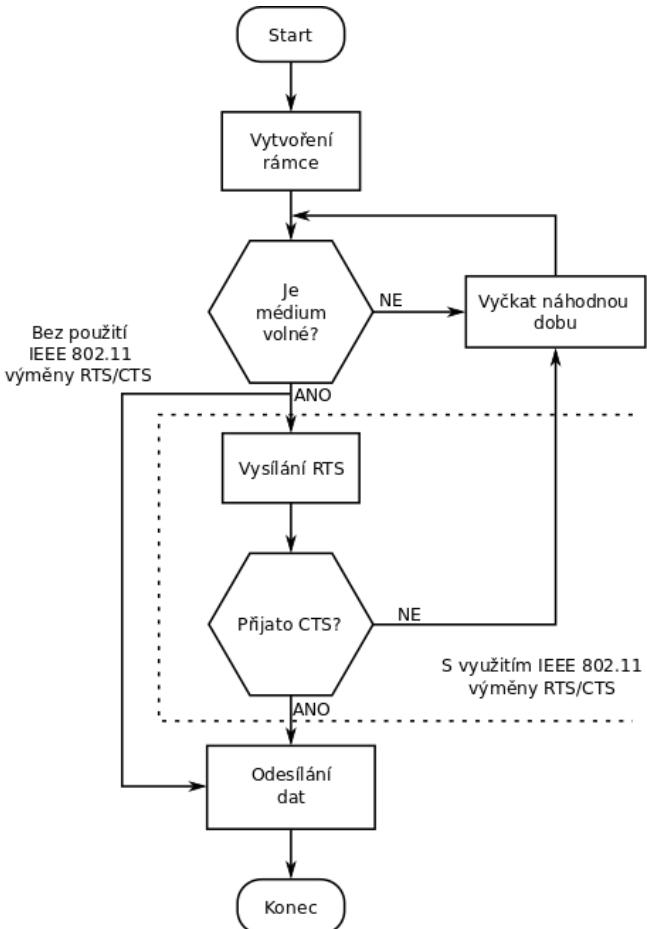


- Rámců (na rozdíl od Ethernetu) existuje více druhů:
- **Kontrolní**
  - Slouží pro připojení stanice do sítě (přihlášení k AP).
    - **Probe** (připojení) **request/responce**, **Authentication** (autentizace) **request/responce**.
  - Zasílají parametry, které stanice musí podporovat pro připojení k síti.
    - **Beacon** (obsahuje i název + parametry přenosu) .
- **Řídicí**
  - Řídí přístup stanic k médiu, tzn. určují která stanice může vysílat.
    - Např. **RTS/CTS**, **ACK** (viz dále).
- **Datové** (přenášení dat)
  - Přenášejí data mezi AP a stanicí, popř mezi stanicemi.
- Další informace jsou již nad rámec tohoto předmětu a odkazují zájemce na předmět **NI-BPS**.

# RTS/CTS mechanismus



- Problém skrytého uzlu je v praxi velmi běžný, avšak standardní metoda **CSMA/CA** jej nedokáže vyřešit. Z tohoto důvodu byl zaveden RTS/CTS mechanismus, který se CSMA/CA může (a obvykle to tak je) ale nemusí kombinovat.
- **RTS** = Request to Send, **CTS** = Clear to Send.



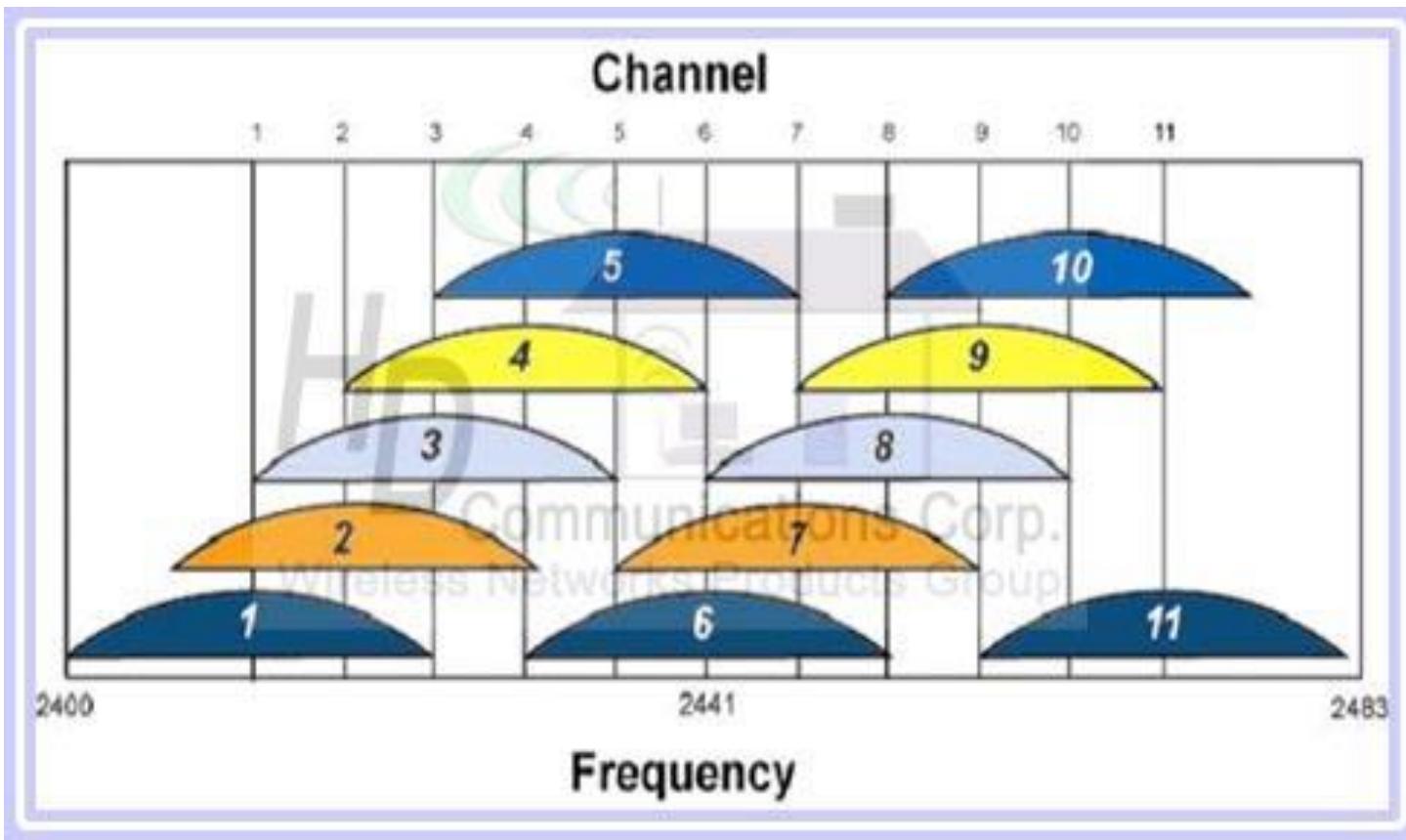
1. Stanice, která chce odvysílat data jiné stanici či AP zašle AP rámec **RTS**.
2. Pokud žádná jiná stanice dříve o vysílání nežádala, odešle AP stanici rámec **CTS**.
3. Rámec CTS vyslaný AP však **obdrží všechny stanice**, které jsou k AP připojeny (AP vysílá všem jednou anténou).
4. Stanice, které obdrží rámec CTS, a zjistí, že není určený pro ně, vyčkají definovanou dobu.
5. Stanice, která obdržela pro sebe určený CTS rámec začne vysílat.

# Vlastnosti IEEE 802.11b - FYI



- Frekvence 2,4 GHz.
- Šířka kanálu 22 MHz.
- Celkem kanálů – 14 (závislé na regionu).
- V ČR – 1-13, Japonsko – pouze kanál č. 14.
- Modulace DSSS+BPSK modulace (přenáší se 1 bit).
- Max propustnost přenosového kanálu je 11 Mbitů.

# IEEE 802.11b – překryv kanálů



- V případě použití této normy se přenosové kanály (channels) **překrývají**.
- Standardní šířka přenosového kanálu je **22 Mhz**, avšak přenosové kanály jsou odstupňovány pouze po **5 Mhz** → využitelné bez vzájemného rušení jsou pouze 3 kanály.
- Reálně v ČR lze tedy využít pouze kombinace **{1,6,11}** , **{2,7,12}** a **{3,8,13}**.

# Vlastnosti IEEE 802.11g - FYI



- Oproti IEEE 802.11b se používá navíc:
  - **OFDM** (viz slajd 16).
  - **QPSK** modulace (přenáší se najednou 2 bity).
- Maximální propustnost přenosového kanálu je 54 Mbit/s.
- Existuje zpětná kompatibilita s IEEE 802.11b
  - Jakmile se objeví jediná stanice podporující pouze starší verzi, začnou všechny stanice podporující normu IEEE 802.11g komunikovat dle normy IEEE 802.11b.

# Vlastnosti IEEE 802.11a - FYI



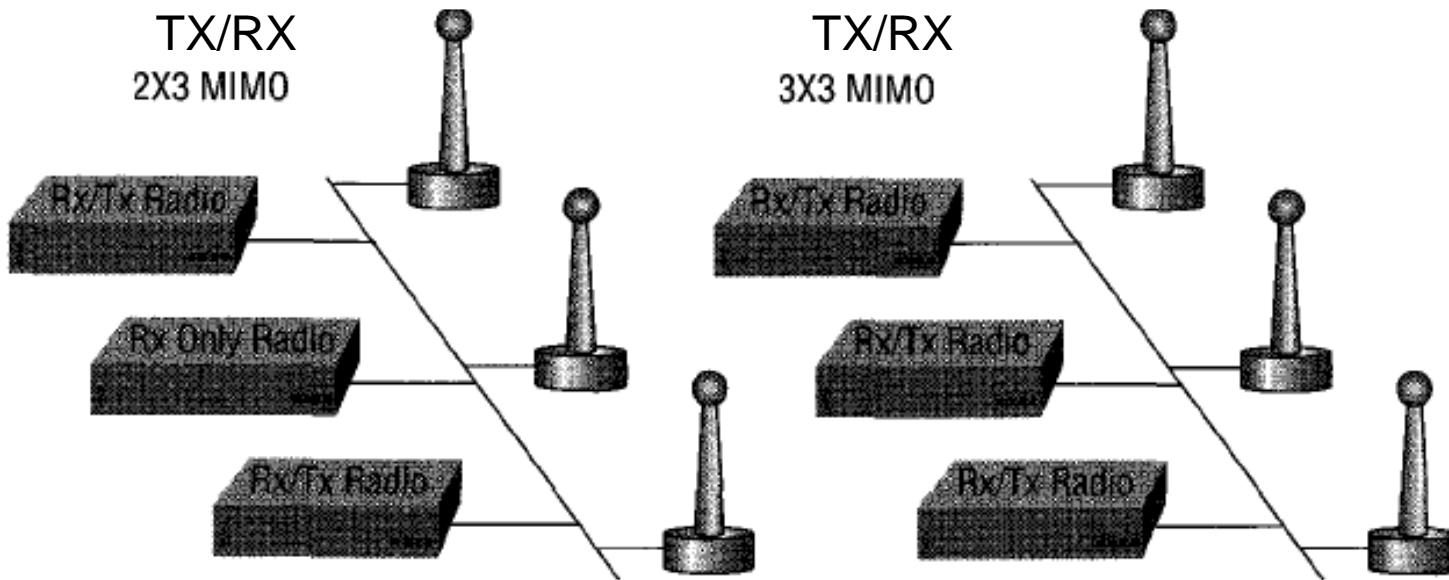
- Pracuje v pásmu 5.1-5.9 GHz.
- Používá se OFDM, QPSK a 16-QAM modulace (přenáší se současně 2 či 4 bity)
- Šířka přenosového kanálu je 20 MHz.
- Přenosové kanály se nepřekrývají.
- Domovní (5180-5320 MHz) a venkovní frekvence (5500-5700 MHz).

# Vlastnosti IEEE 802.11n



- 2,4 GHz nebo 5,1-5,9 GHz.
- Používá se OFDM, QPSK a 16-QAM modulace (přenáší se současně 2 či 4 bity).
- Je podporovaná více antén, **MIMO** systémy (viz dále, slajd 31).
- Je možné využívat **beamforming** (viz dále, slajd 34).
- Přenosové kanály je možné **spojovat** (viz dále, slajd 35).
- Propustnost přenosového kanálu závisí na počtu použitých antén, šířce kanálu a použité modulaci (max. 600 Mbit/s).

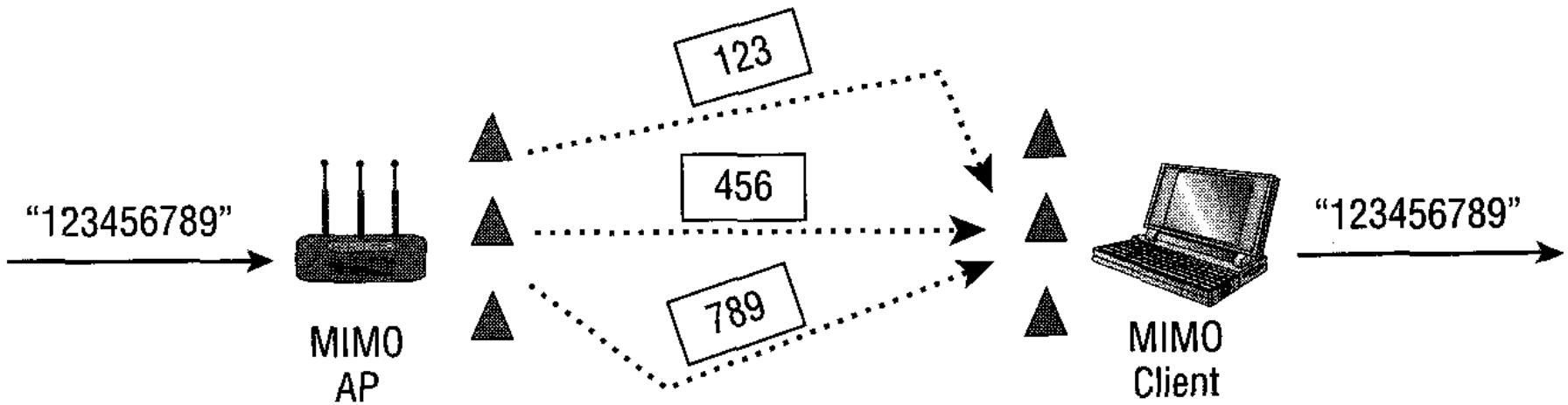
# MIMO systém



Převzato z <https://www.cisco.com/c/en/us/td/docs/wireless/>

- Klasická IEEE 802.11 se označuje jako **SISO = Single Input Single Output**, používá se jedna anténa pro příjem a vysílání.
- **MIMO** = Multiple Input Multiple Output, pro vysílání a přijímání se používá více antén současně. Někdy se tato norma označuje jako HT = High Throughput.
- Způsobu využití je více: **prostorový multiplex, diverzita antén** (viz dále).
- IEEE 802.11n podporuje současné použití až 4 antén.

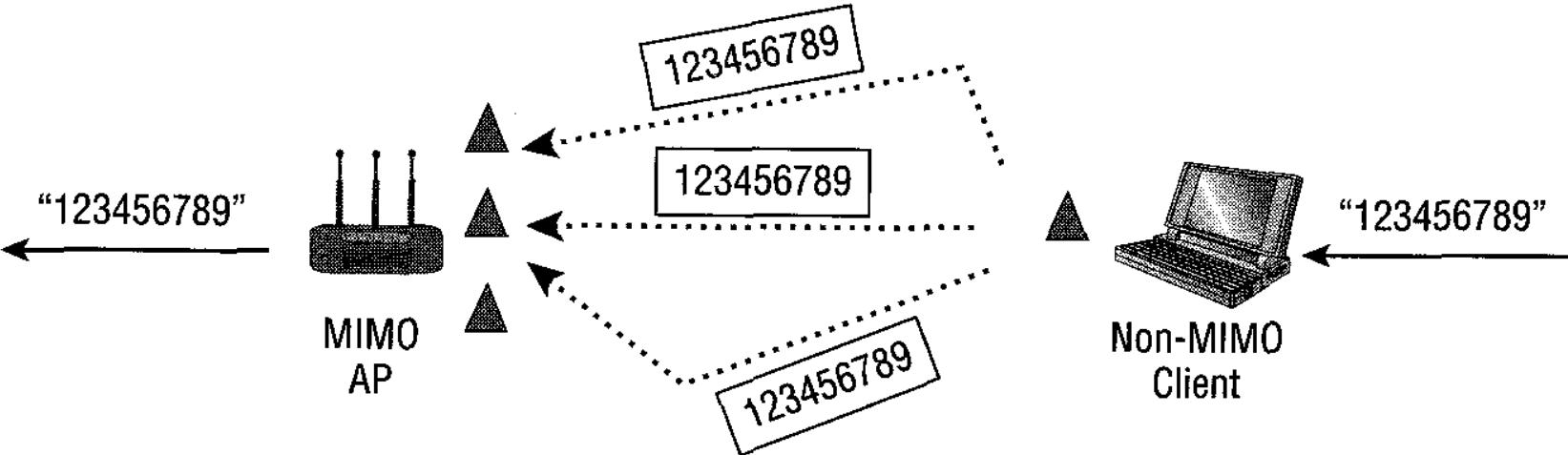
# Prostorový multiplex



Převzato z <https://www.cisco.com/c/en/us/td/docs/wireless/>

- Data se před vysíláním rozdělí a jsou vysílána po částech různými anténami. Důsledkem tohoto je několikanásobné navýšení propustnosti přenosového kanálu.

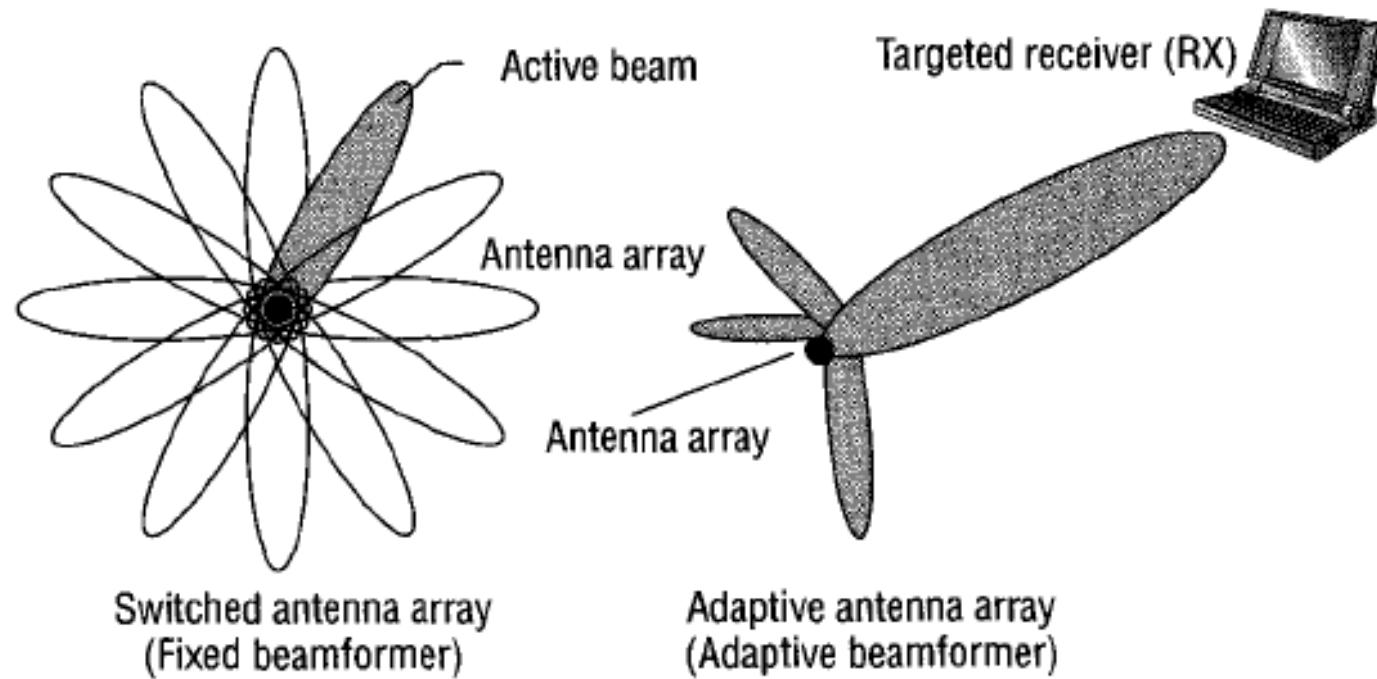
# Diverzita antén (Maximal Ratio Combination)



Převzato z <https://www.cisco.com/c/en/us/td/docs/wireless/>

- V tomto případě se k AP, které podporuje technologii MIMO, připojuje klient, který však technologií MIMO nepodporuje. AP přijímá stejný signál více anténami a vybere si ten, který je nejlepší.

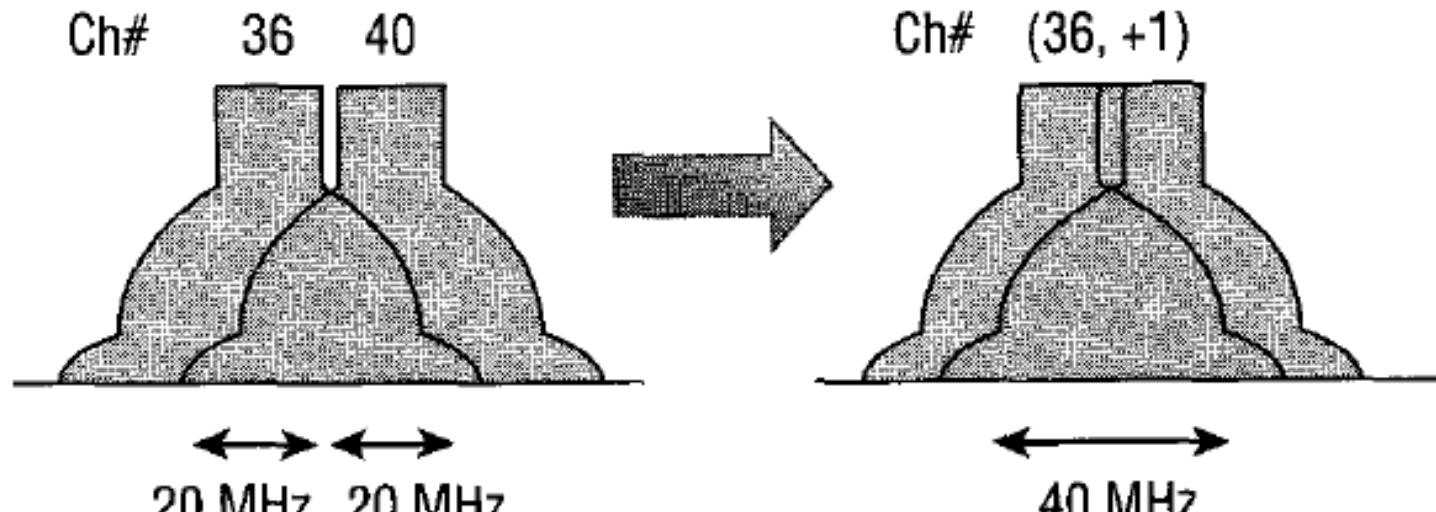
# Beamforming



Převzato z <https://www.cisco.com/c/en/us/td/docs/wireless/>

- Beamforming = vysílání elmag. vln v úzkém směru, který lze dynamicky měnit.
- Beamforming lze realizovat prostřednictvím **anténního pole resp. úzce-směrově nastavitelné antény**.
- U anténního pole pro komunikaci se stanicí AP zvolí takovou anténu (např. na základě intenzity signálu), v jejímž směru se stanice nachází.
- Díky **volbě optimálního směru** dochází k menšímu rušení ostatních stanic kvůli využití užšího vyřazovacího směru.

# Spojování kanálů (Channel Bonding)



**Primary channel = 36**  
**Secondary channel = 40**

Převzato z <https://www.cisco.com/c/en/us/td/docs/wireless/>

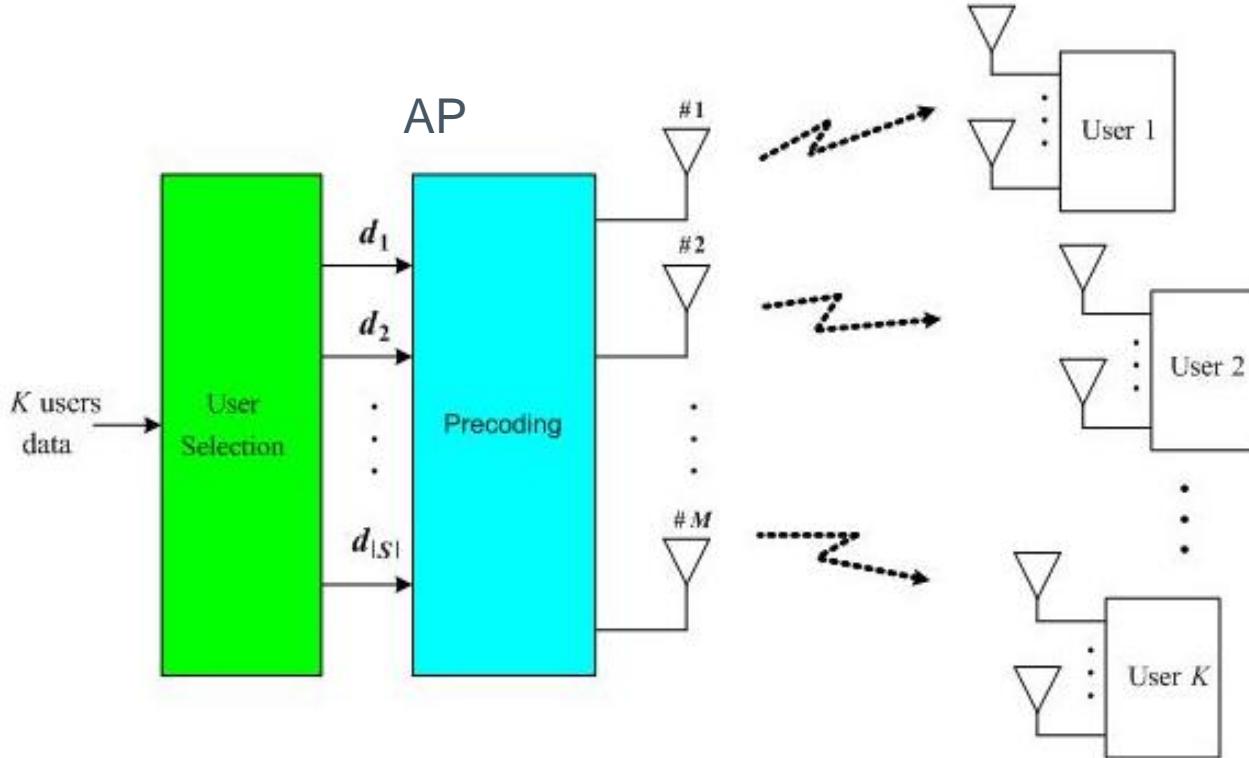
- Více přenosových kanálů se využije současně k vytvoření jediného širokého přenosového kanálu.
- Část jednoho kanálu se používá pro řízení přenosu.

# IEEE 802.11ac/ax (Wi-Fi 6)



- Lze použít současně až 8 antén.
- Podporuje mechanismus **MU-MIMO** (viz dále).
- Používají se 16, 64, 256 QAM modulace (přenáší se 4, 6 či 8 bitů najednou).
- 20/40/80/160 Mhz široké přenosové kanály.
- Pro jednotlivé klienty se používá prostorový multiplex řešený pomocí beamformingu.
- Pro jednu síť je možná koexistence ve frekvenčních pásmech 2,4 a 5 Ghz (využívají se obě pásmá současně).
- Nové metody zabezpečení (WPA3, zatím neprolomeno, malá podpora).

# Mechanismus Multi User (MU) MIMO



Převzato z [https://en.wikipedia.org/wiki/Multi-user\\_MIMO](https://en.wikipedia.org/wiki/Multi-user_MIMO)

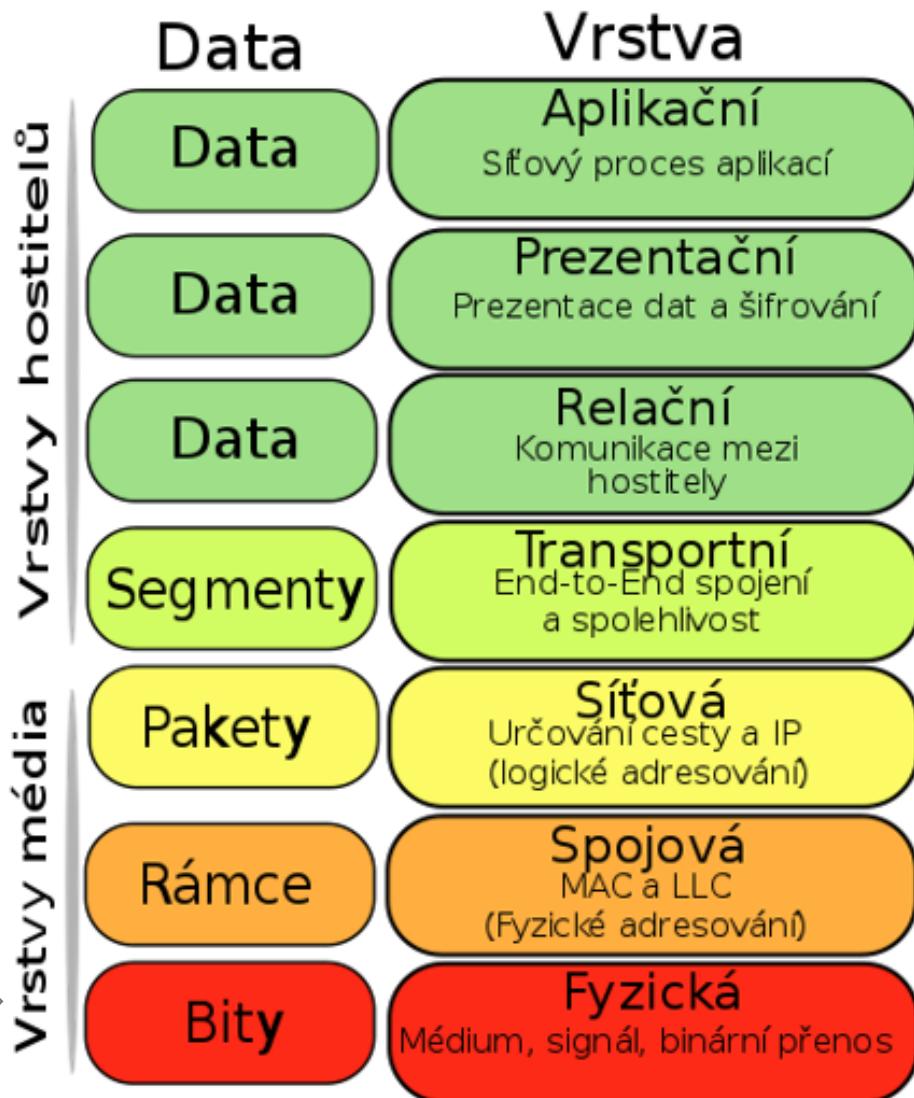
- **MU MIMO** = pro každého uživatele je vyhrazena část z celkového počtu antén.
- S každým uživatelem se komunikuje paralelně (MIMO), avšak směry komunikace jednotlivých uživatelů jsou prostorově odděleny.

# Počítačové sítě

Přednáška 12 - Internet poslední míle a optické sítě



# Internet poslední míle a technologie



- **Poslední míle** (last mile) = oblast mezi koncovým uživatelem a poskytovatelem Internetu.
- Realizace připojení poslední míle probíhá prostřednictvím metalických, optických a bezdrátových spojů.
- U metalických sítí je možné využít datové vodiče pro i pro napájení zařízení.
- Optické sítě = sítě, u kterých je informace přenášena prostřednictvím světelného paprsku šířeného optickým vlákном (světlovodem).
- Vid (mode) = cesta, kterou se šíří paprsek v optickém vlákně.
- Dle počtu možných vidů se optická vlákna dělí na jednovidová anebo vícevidová.
- Při přenosu dat v optických vlákech vznikají různé poruchy, kterým se říká **disperze**.
- Dle způsobu přeposílání dat mezi síťovými zařízeními se optické sítě **dělí na aktivní a pasivní**.



# Technologie Internetu poslední míle dle typu přenosového média

- **Metalika** = koaxiální a elektrické kabely.
  - Metalické kabely umožňují **současný přenos dat a napájení zařízení**.
- **Optika** = optická vlákna.
  - Dnes nejrychlejší přenosové médium.
  - Neumožňuje napájení.
- Bezdrátové spoje = antény (viz přednáška 11).

# Metalické kabely

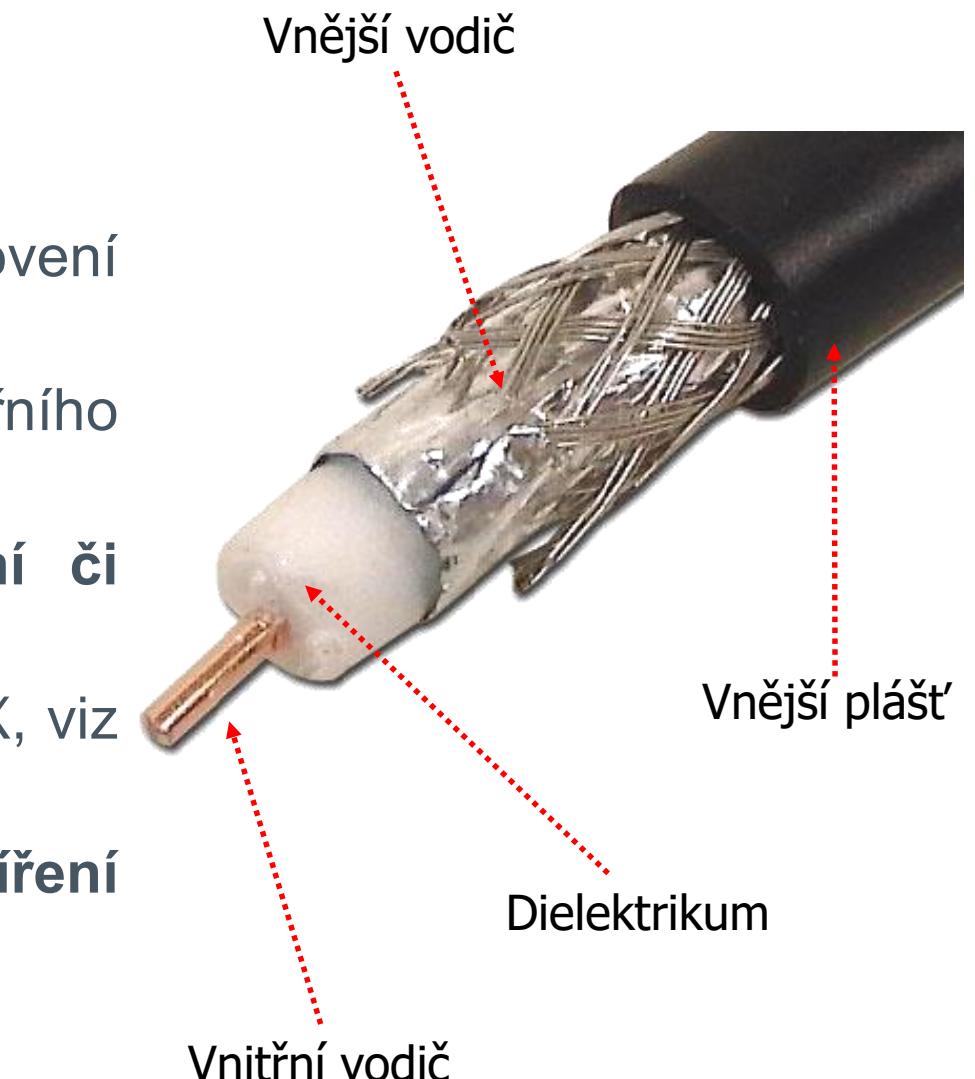


- Materiálem média je **ušlechtilý kov** (většinou měď).
- V minulosti používány hojně.
- V dnešní době se pro šíření Internetu od nich **upouští** a nahrazují je bezdrátové či optické spoje (metalika není odolná vůči přepětí).
- Oproti optickým kabelům jsou levnější a práce s nimi je snazší.
- Druhy metalických kabelů **dle požitých typů vodičů** se dělí na:
  - **Asymetrické** = různé vodiče v rámci 1 kabelu.
    - Označují se běžně jako **koaxiální** kabely.
  - **Symetrické** = stejné vodiče v rámci 1 kabelu.
    - **Kabely s kroucenými páry vodičů.**
- Druhy metalických kabelů **dle ochrany vodičů** se dělí na:
  - **Nestíněné** = pro vnitřní použití.
  - **Stíněné** = pro vnější použití, resp. i pro vnitřní v místech s vysokým stupněm rušení.

# Asymetrické (koaxiální) kabely



- Pro posílání **dat** se využívá **vnitřní vodič**.
- **Vnější vodič** slouží jako refereční pro stanovení úrovně signálu šířeného vnitřním vodičem.
- Vnější vodič se používá i pro **stínění** vnitřního vodiče.
- Dnes se používají zejména pro **televizní či satelitní rozvody**.
- Kdysi se používal i pro **Ethernet** (10 Base X, viz 2. přednáška slajd 22).
- **Koaxiální vodiče je možné použít pro šíření signálu o frekvenci od 30 KHz do 6 Mhz.**



# Šíření signálu v koaxiálních kabelech



- Při šíření signálu koaxiálním kabelem dochází postupně k jeho slábnutí. Tomuto jevu se říká **útlum**.
- Útlum souvisí s charakteristikou impedancí kabelu ( $Z$ ), kterou lze popsát následující rovnicí.

$$Z = \frac{Z_0}{2\pi\sqrt{\epsilon_r}} \cdot \ln\left(\frac{D}{d}\right) \approx \frac{60}{\sqrt{\epsilon_r}} \cdot \ln\left(\frac{D}{d}\right)$$

Charakteristická impedance kabelu

Impedance volného prostředí

Průměr vnějšího vodiče

Průměr vnitřního vodiče

Permitivita materiálu

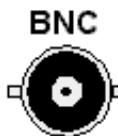
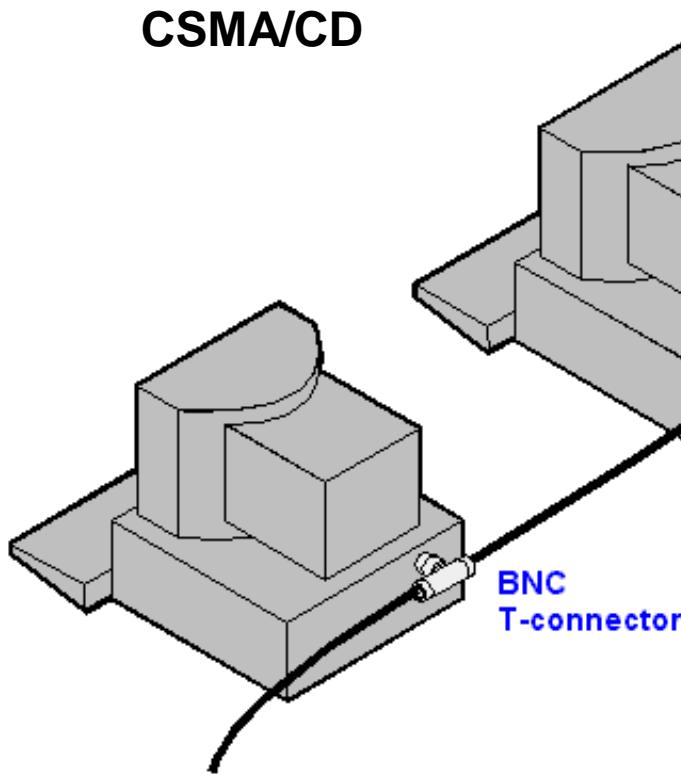
The diagram shows the formula for characteristic impedance Z. An arrow points from the term Z\_0 to the text "Charakteristická impedance kabelu". Another arrow points from the term 2π√ε\_r to the text "Permitivita materiálu". Arrows also point from the terms D and d in the ln argument to the text "Průměr vnějšího vodiče" and "Průměr vnitřního vodiče" respectively. A final arrow points from the first term Z\_0/2π to the text "Impedance volného prostředí".

- Důležitý fakt, který z rovnice plyne, je to, že  $Z$  je menší v případě použití kabelu s větším průměrem vnitřního vodiče. Z tohoto důvodu je obecně lepší využívat kably se **větším průměrem**.
- Útlum se projevuje **zejména u delších kabelů** (desítky či stovky metrů).

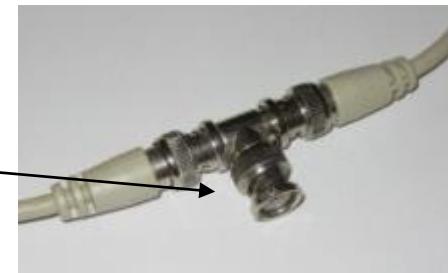
# Ethernet přes koaxiální kabel (10 Base 2) - FYI



From Computer Desktop Encyclopedia  
© 1998 The Computer Language Co., Inc.

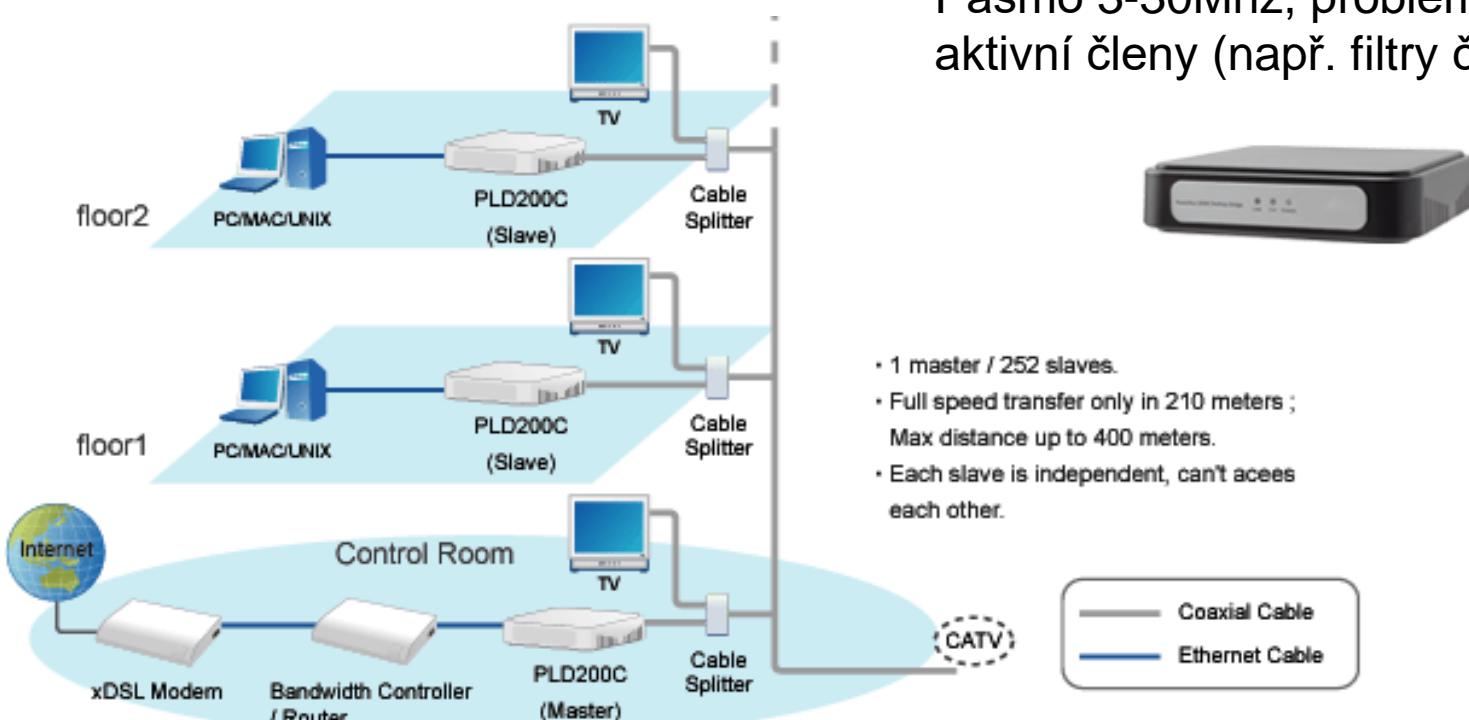


BNC = Bayonet Neill-Concelman



**Terminátor**

# Ethernet přes STA (10/100/500 Mbitů) - FYI



Převzato z <https://morce.emp-centauri.cz/ethernet-over-coax/?lang=cs>

- **STA** = Společná televizní anténa.
- Výrobců této technologie je více, nicméně řešení jsou mezi sebou většinou nekompatibilní, jelikož technologie není standardizovaná.

# Power Line Communication (PLC)



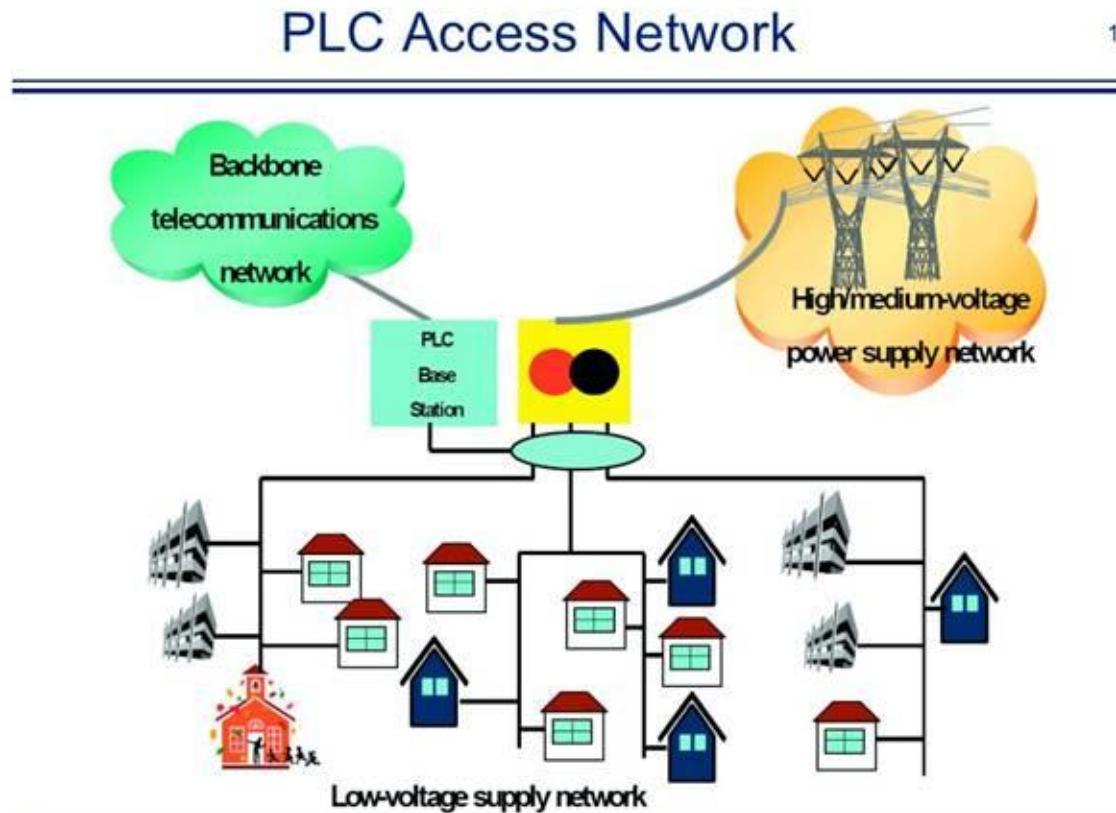
- Signál je šířen stejnými vodiči, jako je vedeno elektrické napětí.
- Někdy označované také jako **PBL** (Power Broadband Link).
- Pro Ethernet se používá ozačení **Ethernet over 230 (Eo230)**.
- Standardizované v roce 2010 v normě **IEEE 1901**, tudíž zařízení různých výrobců by měla být vzájemně kompatibilní.
- Používá se **QPSK modulace** (přenos 2 bitů v rámci 1 symbolu, viz. přednáška 11, slajd 18).
- **Pro sdílení média se používá přístupová metoda TDMA** (viz 2. přednáška, slajd 8).
- Původní myšlenka byla, že tato technologie měla být ekvivalentní klasickému **Dynamic Speed Line (DSL)** připojení, nicméně se nenaplnila.

# PLC - nevýhody



- **Řada zdrojů rušení** – další aktivní zařízení v el. síti.
- **Omezený počet účastníků kvůli použití TDMA.**
- Nefunguje dobře mezi **různými fázemi** v rámci jedné **rozvodné el. soustavy v jednom domě.**
- Trafostanice jsou pro PLC zcela neprůchozí, jelikož fungují jako galvanická oddělení.
- **V ČR užívané téměř výhradně pro domácí použití.**

# Struktura sítě, využívající PLC zařízení - FYI



Zařízení Eo230 pro vytváření dvoubodových spojů

Převzato z [http://www.oas.org/en/citel/infocitel/2009/junio/plc\\_i.asp](http://www.oas.org/en/citel/infocitel/2009/junio/plc_i.asp)

# Kabely s kroucenými páry vodičů

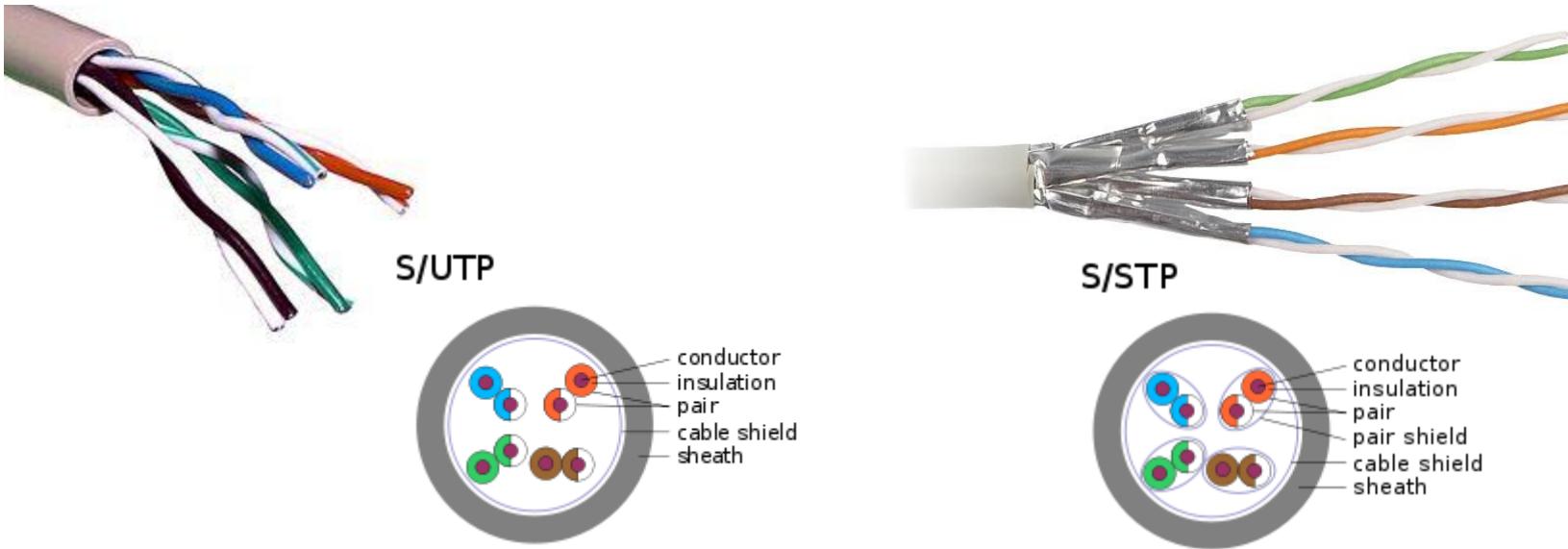


- Jedná se o **symetrické kably**, všechny vodiče jsou stejné.
- Někdy označované též jako kroucené dvoulinky.
- Jsou **levné**, nicméně v porovnání s aktuálními jinými technologiemi méně výkonné.
- Parametry kabelů jsou **průměr vodiče a perioda kroucení**.
- **Důvod kroucení = ochrana proti rušení.**
- Čím **kratší perioda zakroucení**, tím vyšší odolnost vůči rušení, ovšem i vyšší cena.
- Typicky **2,4,22,24,26 vodičů v jednom kabelu.**

# Druhy kabelů s kroucenými páry vodičů



- **UTP** = Unshielded Twisted Pair, nestíněné jednotlivé páry vodičů.
- **STP** = Shielded Twisted Pair, stíněné jednotlivé páry vodičů.



Převzato z <https://www.indiamart.com/proddetail/cat-6-cable-utp-stp-19856138091.html>

- **(S)UTP/STP** = Shielded UTP/STP, kabel obsahuje okolo vodičů stínící pletivo.
- **(F)UTP/STP** = Foiled UTP/STP, kabel obsahuje okolo vodičů speciální ochrannou kovovou fólii.

# Konektory používané u kabelů pro kably s kroucenými páry vodičů



- **RJ11**
  - Určený pro 3 páry vodičů.
  - Používaný pro telefony či DSL připojení, použitelný pro rychlosti do **100Mb/s**.
- **RJ45**
  - Určený pro 4 páry vodičů.
  - Používaný pro běžné počítačové sítě, použitelný pro rychlosti do **10 Gb/s**.
- **ARJ45/CG45**
  - Konektor určený pro 6 párů vodičů.
  - Zpětně kompatibilní s RJ45, zapojené jsou pak pouze 4 páry.
  - Pro přenosové rychlosti do **40 Gbit/s**.



RJ11



RJ45



ARJ45

# Kategorie kabelů s kroucenými páry vodičů



- **5e (Cat 5e, Class D)**
  - Pro Fast Ethernet (100 Mbps, využívá dva páry) a Gigabit Ethernet (využívá všechny čtyři páry), používá stíněnou i nestíněnou kabeláž, šířka pásma 100 MHz, konektor RJ45.
- **6 (Cat 6, Class E)**
  - Podporuje 10 Gbps Ethernet, ale pouze do 55 metrů, používá stíněnou i nestíněnou kabeláž, šířka pásma 250 MHz, konektor RJ45.
- **6a (Cat 6a - augmented, Class EA)**
  - Podporuje 10 Gbps Ethernet na plnou vzdálenost 100 metrů, používá stíněnou i nestíněnou kabeláž, šířka pásma 500 MHz, konektor RJ45.
- **7 (Cat 7, Class F)**
  - Neujala se, pro podporu 10 Gbps Ethernet, vyžaduje stíněnou kabeláž (S/FTP) a konektory ARJ45, šířka pásma 600 MHz, nicméně výrobci raději zvolili Cat 6a a RJ45.
- **7a (Cat 7a, Class FA)**
  - Rozšíření Cat 7, mohlo by podporovat 40 Gbps Ethernet na vzdálenost 50 metrů, šířka pásma 1000 MHz.
- **8 (Cat 8)**
  - Aktuálně ve vývoji (ISO technické doporučení z roku 2013), podporuje 40 Gbps Ethernet, šířka pásma 1600 až 2000 MHz, dvě varianty, jedna používá stíněné kably (U/FTP, F/UTP) a konektor RJ45, druhá kably (F/FTP, S/FTP) a konektor ARJ45.

# Power over Ethernet (PoE)



- **PoE** = napájení zařízení je realizováno přes stejný kabel, přes který jsou přenášena data.
- **Různé standardy** IEEE 802.af (PoE), IEEE 802.at (PoE+), IEEE 802.bf (PoE++), IEEE 802.bt (HighPower PoE).
- Hlavní rozdíly spočívají v maximálním **možném napájecím výkonu**.
- PoE existují v **aktivní** (upravují přenášené napětí) a **pasivní** (neupravují přenášené napětí) formě.
- V praxi se toto řešení používá velmi často v místech, kde je **problém s externím napájením**.
  - Např. pro IP kamery, bezdrátové přístupové prvky (AP) či přepínače.

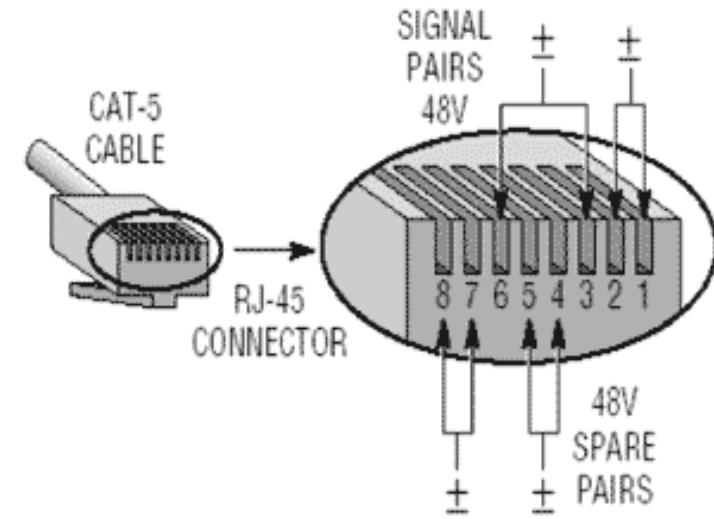
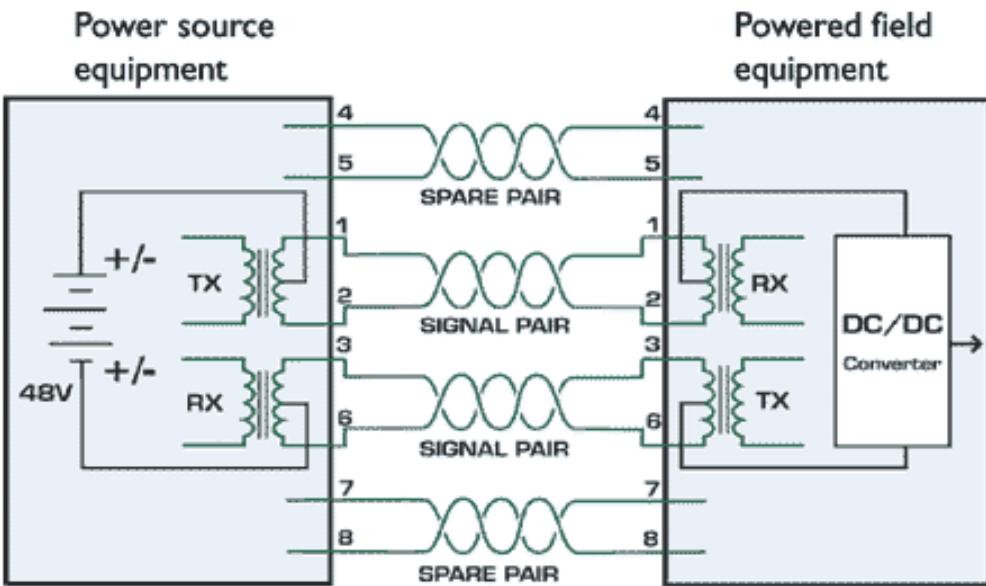
# Verze PoE dle možného přenášného výkonu - FYI



	PoE	PoE+	PoE++, UPoE	High-Power PoE
Name	802.3af	802.3at	802.3bf	802.3bt
Standard	15.4W	30W	60W	100W
Port Power				
Devices	VoIP Phones	PTZ Camera	Mgmt Devices	LED Lighting

Převzato z <https://www.fastcabling.com/2019/11/28/what-is-poe-part-2/>

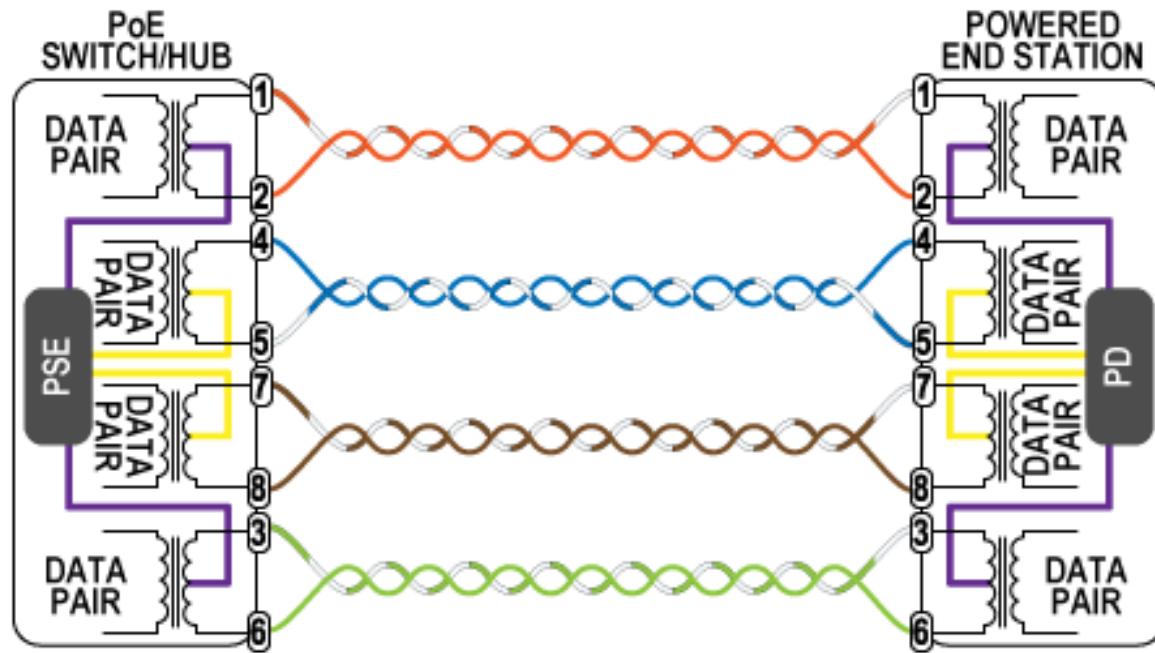
# Princip PoE (Fast Ethernet)



Převzato z <https://www.programmersought.com/article/94904678111/>

- Standardní UTP/STP kabel pro Ethernet obsahuje 4 páry vodičů. Pro Fast Ethernet se používají pouze 4 vodiče, zbylé 4 vodiče jsou použity pro napájení.

# Princip PoE+ (Gigabitový Ethernet)



Převzato z <https://www.programmersought.com/article/94904678111/>

- PSE = Power Sourcing Equipment (napáječ), PD = Powered Device (napájené zařízení).
- Data i napájení se přenášejí po stejných vodičích. To je možné díky tomu, že **frekvence napájení** (60 Hz či méně) a **frekvence signálu přenášejícího data** ( $10^6$  –  $10^7$  Hz) se **zásadně liší** a tudíž nedochází k vzájemnému rušení.

# Optická vlákna (světlovody)



- Přenos informace probíhá pomocí **světelných** či **laserových diod** (v principu podobné, rozdíl pouze ve vlnových délkách).
- Diody jsou instalovány v zařízeních, které se označují jako **optické transcievery**.
- Pro šíření signálu v optických vláknech jsou důležité 2 fyzikální principy:
  - **Index lomu** (viz slajd 22)
  - **Snellův zákon** (viz slajd 23)
- **Vid** = cesta paprsku světla šířený optickým vláknem.
- Dle počtu současně přenášených vidů se optická vlákna dělí na **jednovidová** (viz slajd 25) a **vícevidová**.
  - Vícevidová vlákna se dle konstrukce dále dělí na:
    - Vlákna se **skokovým indexem lomu** (viz slajd 26).
    - Vlákna s **gradientním indexem lomu** (viz slajd 27).

# Vlastnosti optických vláken



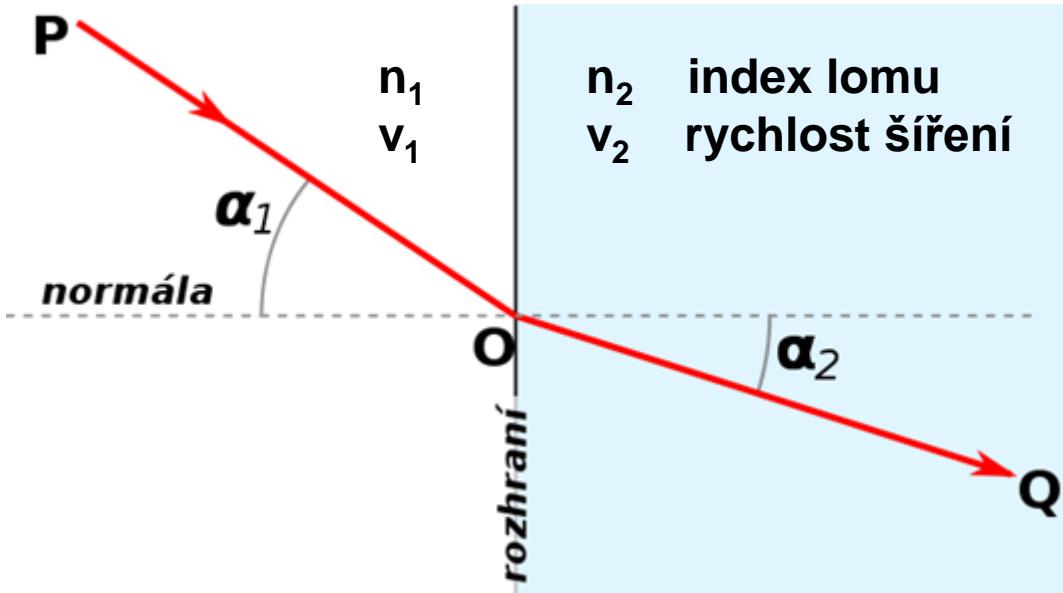
- Možné frekvence vysílaného signálu až do **200 THz**.
- **Dokonalé galvanické oddělení** stanic a síťových prvků (nepřenáší se žádné el. výboje).
- Neumožňují přenášet napájení.
- **Mají nízký útlum** signálu v porovnání s metalickými kably.
- **Nutnost opakování** signálu až po několika **desítkách km vlivem poruch (disperzí, viz slajdy 28-30)**.
- Používají se primárně v páteřních linkách poskytovatelů Internetu.
- Nejlepší současné technologie dokáží prostřednictvím optických vláken přenášet 100ky Gbit/s.



# Fyzikální principy využívané v optických vláknech: Index lomu

- Rychlosť světla je v různých homogenních prostředích obecně různá.
- Index lomu  $n = c/v$ , **vždy  $> 1$**  mimo vakuum.
  - $v$  je rychlosť světla v daném prostředí,  $c$  je rychlosť světla ve vakuu.
- Měme dvě prostředí, u kterých index lomu v prvním prostředí je roven  $n_1$  a ve druhém  $n_2$ .
- Relativní index lomu při přechodu paprsku z prvního prostředí do druhého je roven  $n_{21}=n_2/n_1=v_1/v_2$ .

# Fyzikální principy využívané v optických vláknech: Snellův zákon



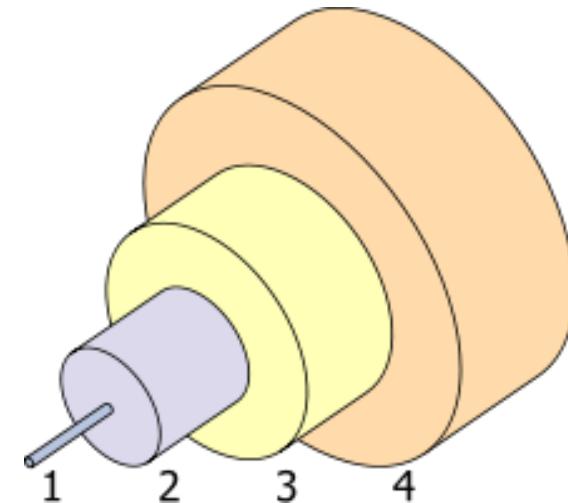
$$\frac{\sin \alpha_1}{\sin \alpha_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1}$$

- $n_2 > n_1$  nastává lom od kolmice .... z hustšího prostředí do řidšího.
- $n_2 < n_1$  nastává lom ke kolmici .... z řidšího prostředí do hustšího.
- Platí-li  $\alpha_2 = 90^\circ$ ,  $\sin(\alpha_m) = v_1/v_2$ , nastává totální reflexe (vše se odrazí zpět).

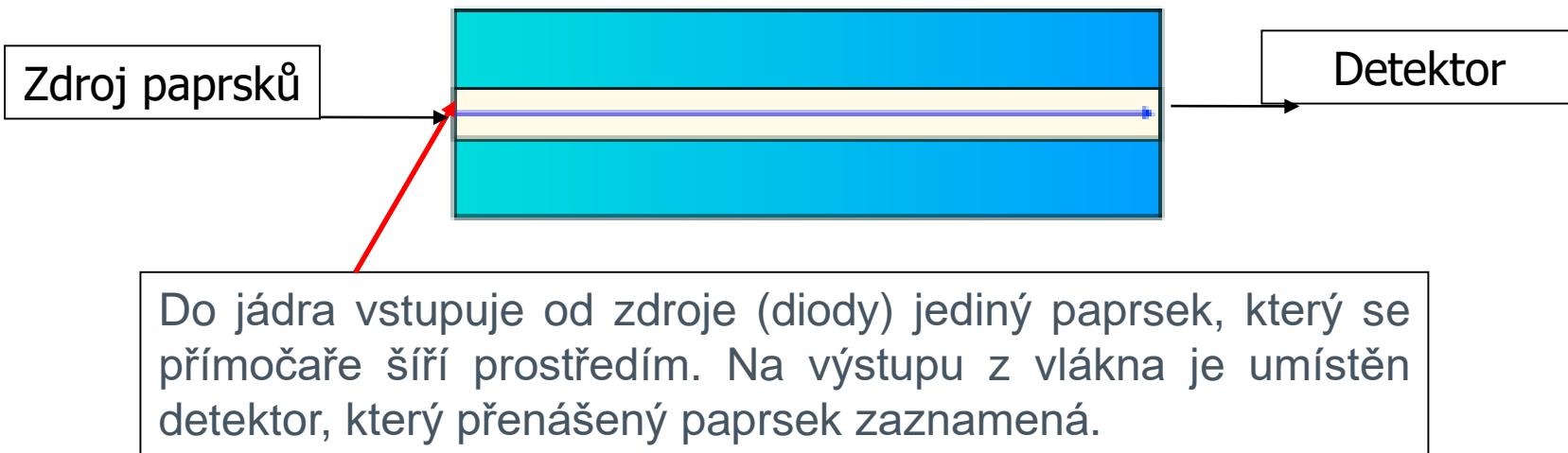
# Optické vlákno - struktura



- Typicky se skládá ze čtyř částí
  1. **Jádro** (core)
  2. **Plášt'** (cladding)
  3. **Ochranná vrstva** (isolation)
  4. **Obal** (jacket)

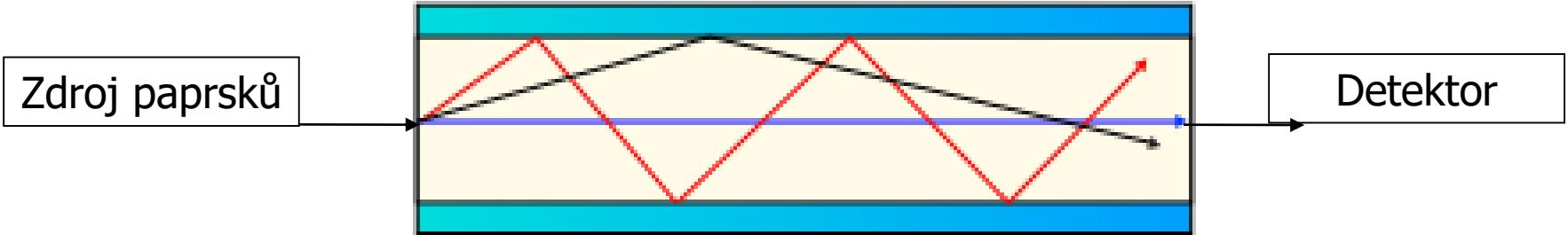


# Jednovidová optická vlákna



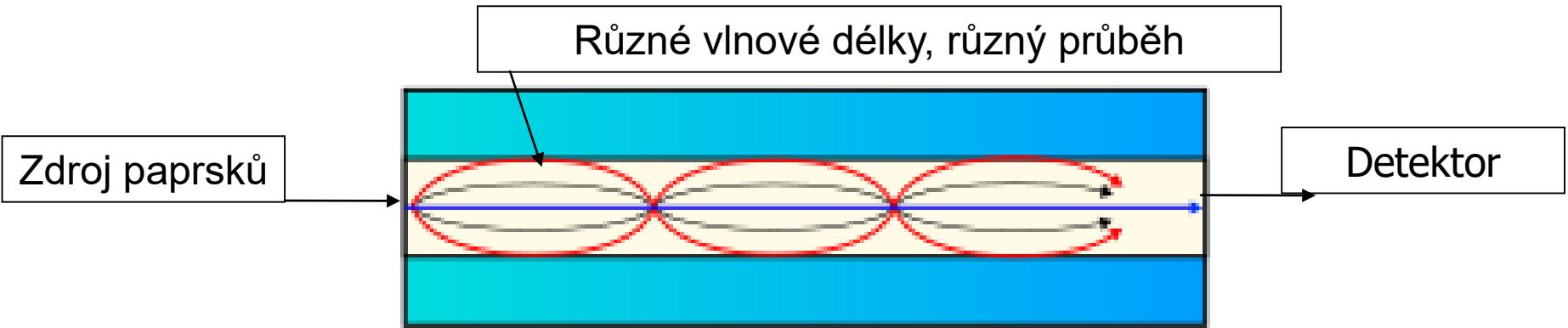
- **Poměr mezi průměrem jádra a průměrem pláště je obvykle 8/125.**
- Používají se na spoje pro vzdálenosti v řádech **km až desítek km**.
- Pro nasvícení se používají **laserové diody**. Oproti metalice jsou jednovidová optická vlákna poměrně drahá.

# Vícevidová vlákna se skokovým indexem lomu



- Paprsky mohou vstupovat ze zdroje (diody) do vlákna pod více úhly, pomocí **totální reflexe** dochází k vytváření cesty paprsku.
- Na druhé straně vlákna se nachází detektor, který registruje jejich součet.
- Používají se pouze na **krátké vzdálenosti**, jelikož vykazují větším útlum signálu v porovnání s jednovidovými.
- Jsou levná, jako zdroj světla se používají **světelné diody**.
- **Typická hodnoty poměru průměru jádra k průměru pláště je 62.5/125 či 50/125.**

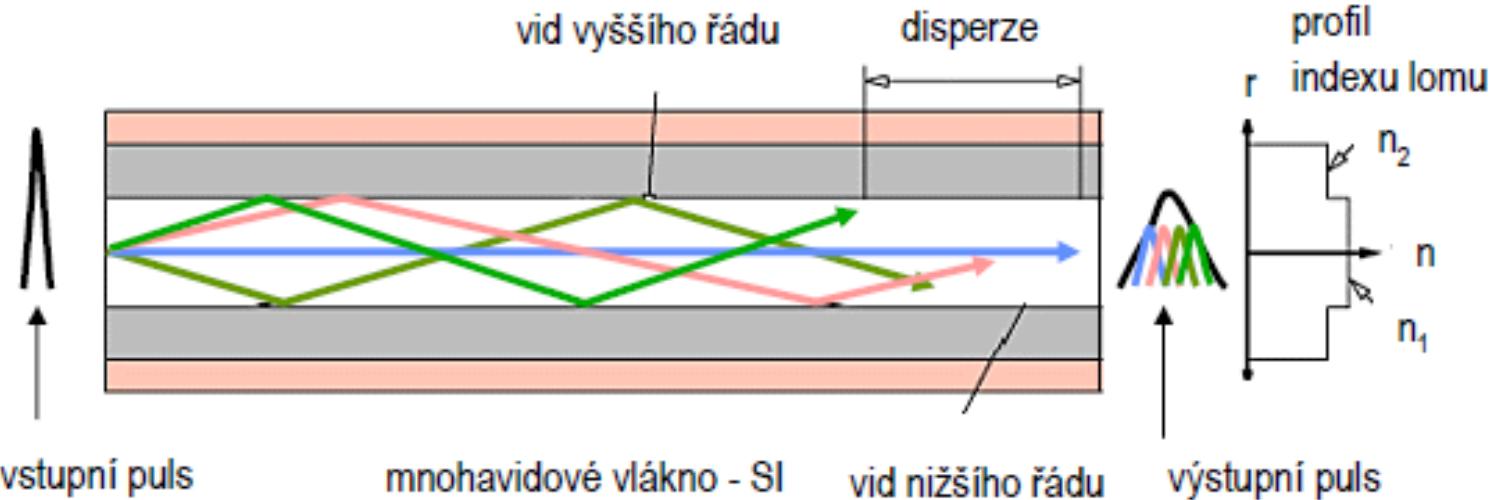
# Vícevidová vlákna s gradientním indexem lomu



- Jádro je tvořeno z tisíců různých vrstev, které postupně mění index lomu tak, aby se signál šířil ve formě sinusovky.
- Důvodem pro jejich vývoj byla **eliminace vlnové disperze (viz dále)**.



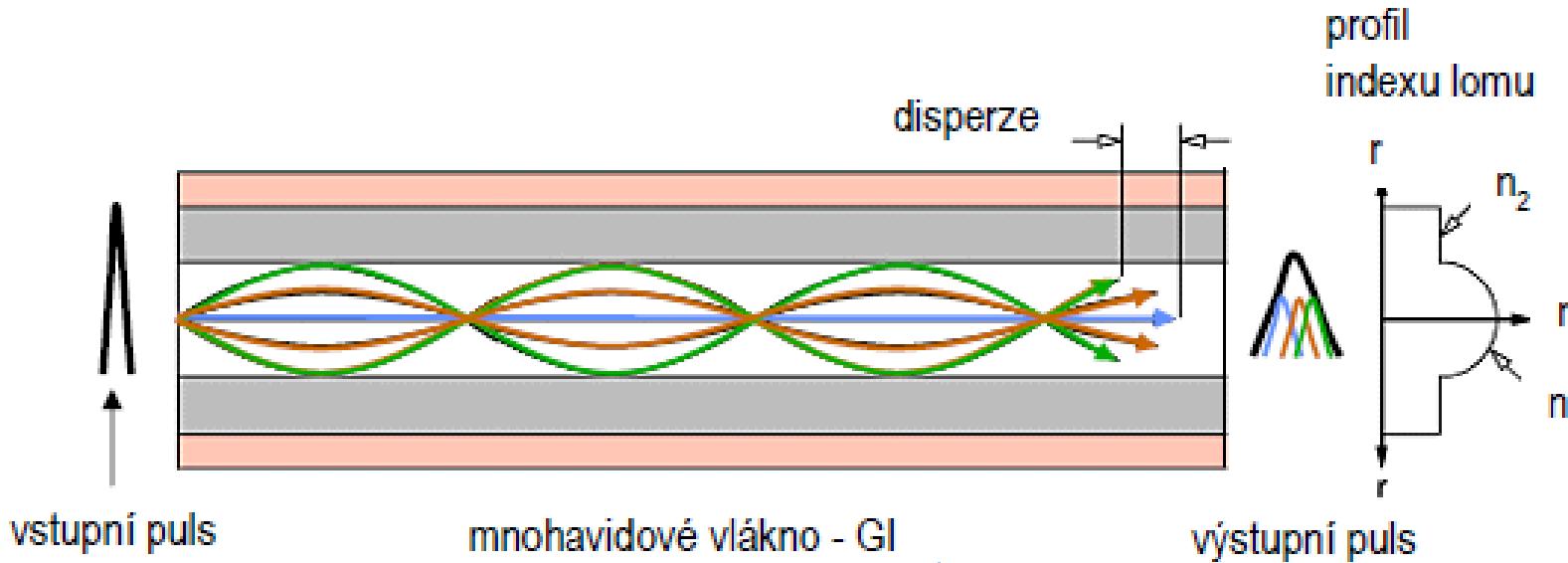
# Vlnová disperze u vícevidových vláken se skokovým indexem lomu



- **Puls** = skupina paprsků různých vlnových délek šířící se současně optickým vláknem.
- Disperze **vzniká tak**, že **index lomu** určitého homogenního prostředí **je pro paprsky s různou vlnovou délkou různý**. Na začátku vystupují všechny paprsky z jednoho místa. Vlnová disperze způsobí to, že se každý paprsek odráží jinak a na výstupu se neprotnou v jednom bodě .
- Výstupní puls je oproti vstupnímu pulsu vnímán jako širší.

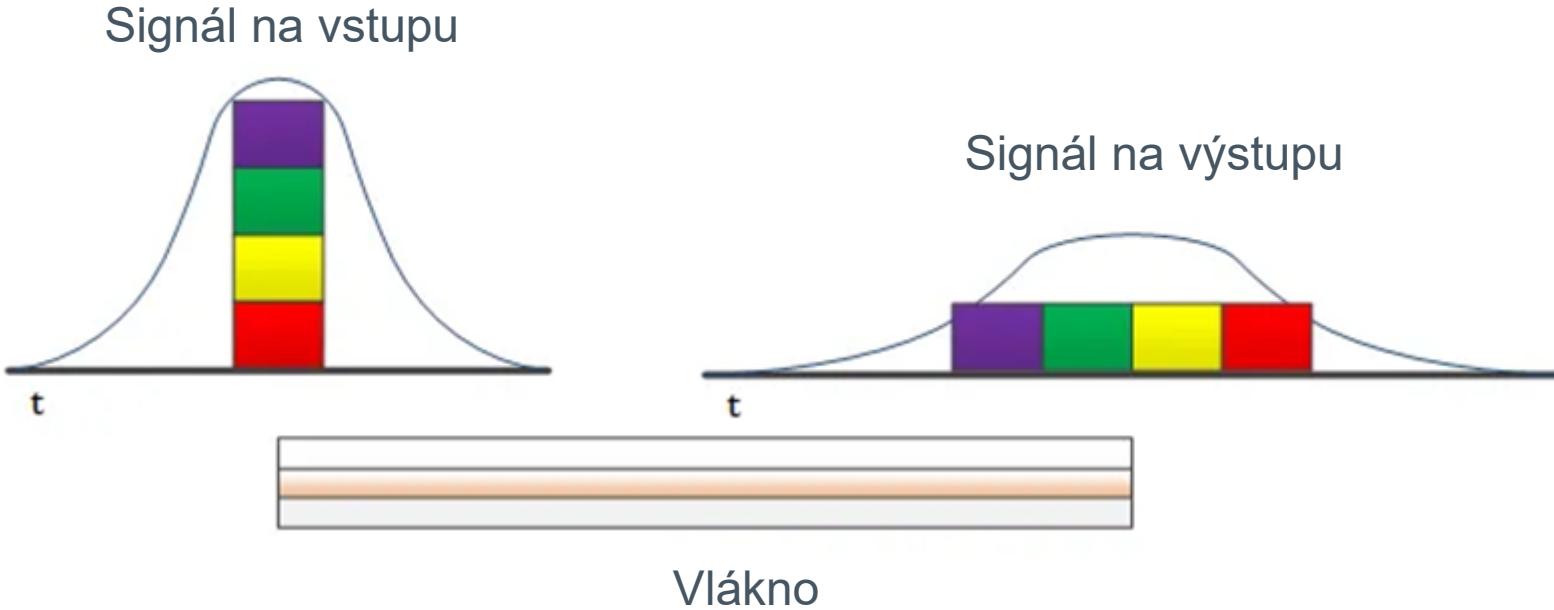


# Vlnová disperze u vícevidových vláken s gradientním indexem lomu



- Vzhledem k tomu, že **hodnota indexu lomu** se mění pozvolna a velmi často, je výsledná disperze oproti vláknům se skokovým indexem lomu podstatně menší.

# Chromatická disperze

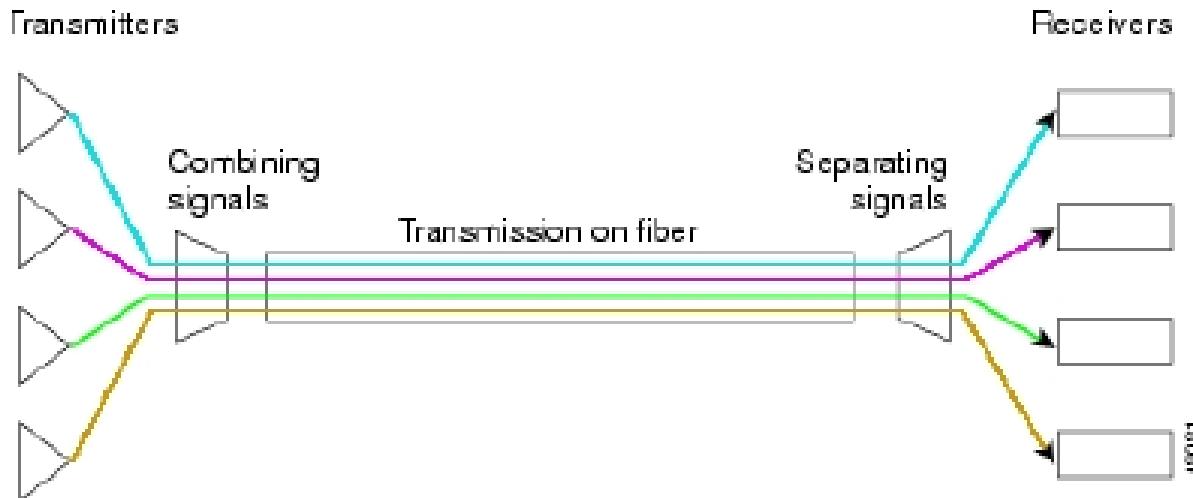


- Důvodem této disperze je to, že různé barvy světla se ve stejném prostředí šíří různou rychlostí.
- Na vstupu jsou **vyslány** všechny paprsky **současně**, na výstupu se objeví **postupně**.
- Tento jev je důsledkem vlastností materiálu konkrétního optického vlákna. Pro eliminaci chromatické disperze se používají aktivní prvky, které se nazývají **kompenzátory chromatické disperze**.

# Vlnový multiplex (WDM)



- WDM = **Wavelength Division Multiplex**, tj. **multiplex** který je speciálním případem **frekvenčnímu multiplexu**.
- Do optického vlákna pustíme více paprsků o různých vlnových délkách. Vlnová délka souvisí s různou frekvencí a barvou daného paprsku.
- **U běžných optických transcieverů dochází k přenosu dat pouze na jedné vlnové délce** (to vyžaduje samostatné vlákno pro vysílání a příjem).
- Pokud potřebujeme zvýšit kapacitu spoje a je neekonomické tahat další vlákna, je vhodnější využít WDM.
- WDM lze použít jak u single tak u více vidových vláken.
- WDM dokáže vyřešit po jednom vlákně současně jak **vysílání tak i přijímání dat**.



Převzato z <https://www.technopedia.com/2018/08/principle-of-wdm-wavelength-division.html>

# Varianty WDM – další vývoj



- **Coarse WDM (CWDM)**
  - původní název pro WDM, u kterého se používaly dvě různé vlnové déky (1550 a 1310 nm).
  - Později došlo k tomu, že byl stanoven rozestup vlnových délek po 20 nm pro 18 různých vlnových délek.
- **Dense WDM (DWDM)**
  - Používá až 160 vlnových délek, délky jsou odstupňovány po jednotkách nm.
  - Technologie využívá čistě optická zařízení, tzn. signál se upravuje pouze optickou nikoli elektronickou cestou.
  - Pro běžného uživatele je **cenově nedostupná** (stojí řádově miliony korun).



# Optické trasy pro Internet poslední míle (FTTX)

- Optická trasa = připojení realizované optickým vláknem.
- Pro zakončení optických vláken se používají SFP konektory:
  - SFP = Small Form-Factor Pluggable
  - SFP, SFP+, QSFP, SFP28 a SFP28+ konektory.
- **FTTX = Fibre to the X.**
- **FTTN – Fibre to the Node**
  - Optická trasa je zakončena v optickém zařízení- rozváděči. Ke klientovi vede metalika (DSL).
- **FTTC – Fibre to the Cabinet**
  - Obdoba FTTN, jen vzdálenost ke klientovi je nižší – max. 300m (kabelové televize).
- **FTTB – Fibre to the Building**
  - Optické trasa vede do budovy, ve které se metalicky rozbočuje pro zákazníky.
- **FTTH – Fibre to the Home**
  - Optická trasa vede až do domu zákazníka, který vlastní optické zařízení.

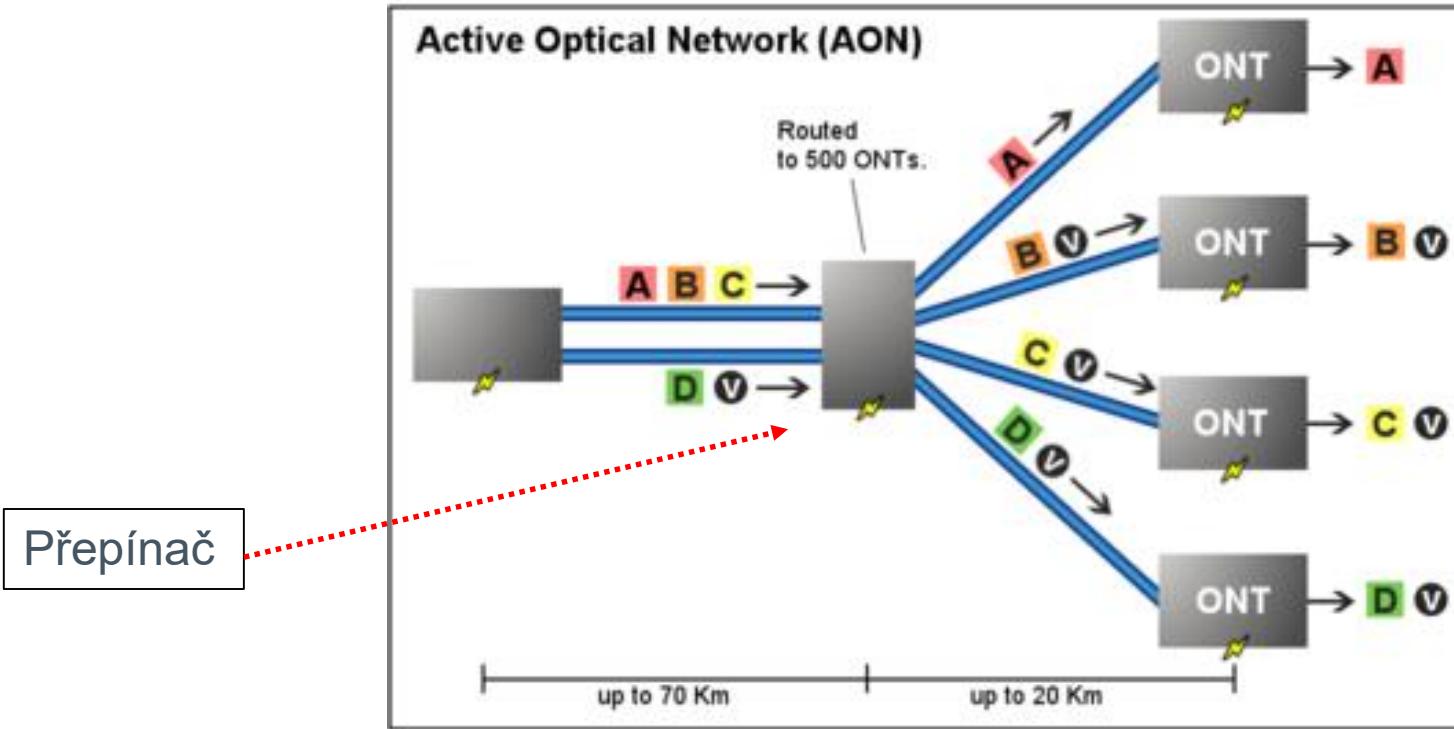


# Druhy optických sítí FTTX



- **Aktivní optická síť** (AON = Active Optical Network)
  - Používají se pouze aktivní zařízení.
  - Aktivní optická zařízení (přepínače) vybírají pro další šíření dat jeden konkrétní výstupní port dle cílové MAC adresy.
  - Výstavba aktivních sítí je dražší – použitá zařízení jsou složitější oproti PON.
- **Pasivní optická síť** (PON = Passive Optical Network)
  - Používají se levnější pasivní prvky.
  - Pasivní optická zařízení (rozbočovače) rozesílají stejná data na všechny výstupní porty.
  - Stanice dostávají data určená i pro další stanice a musejí si samy data filtrovat např. výběrem dle koncové **MAC adresy**.
  - V důsledku duplicitních dat jsou PON oproti AON méně propustné.

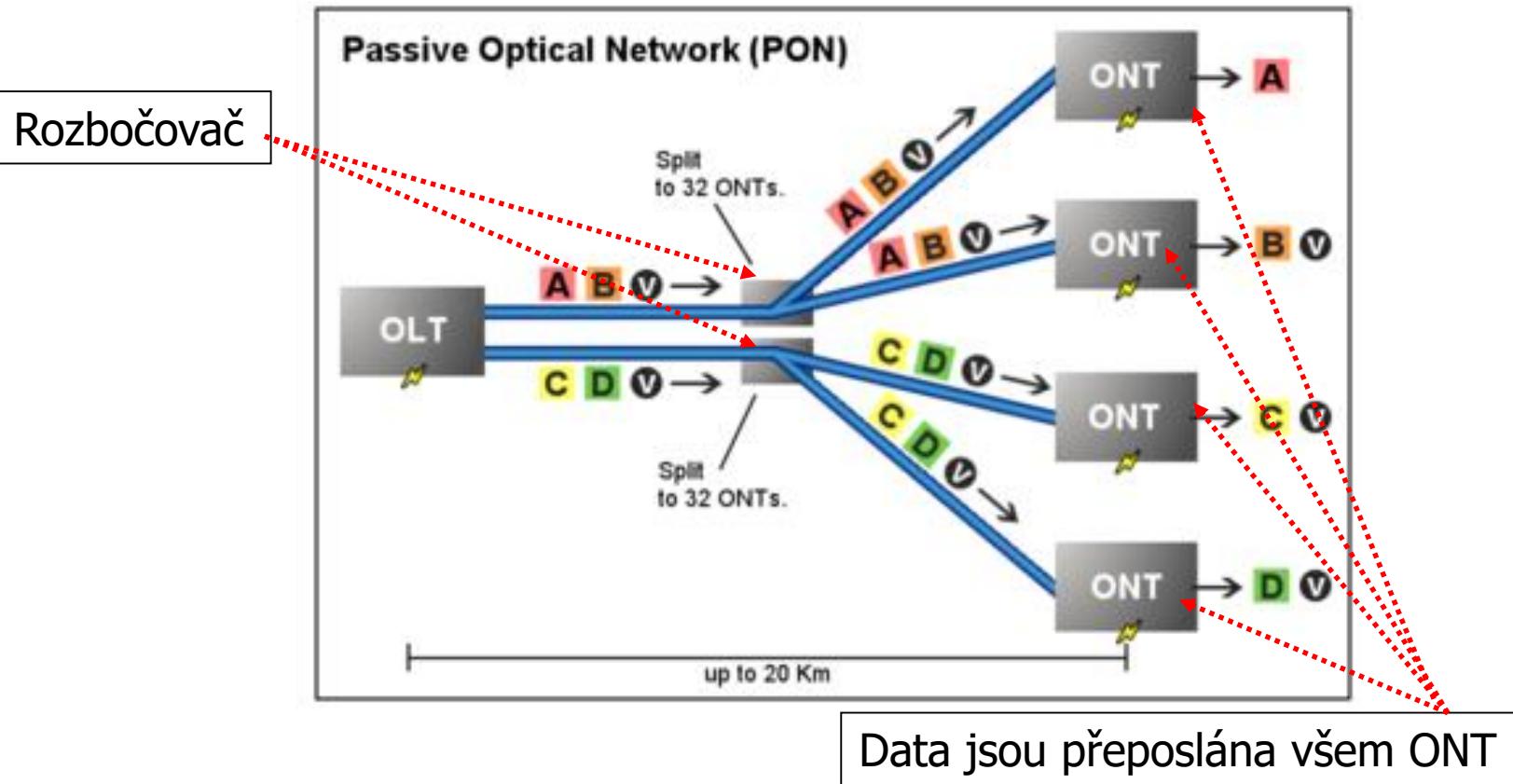
# Princip aktivní optické sítě (AON)



Převzato z <http://www.fiberopticshare.com/ftth-access-networks-aon-vs-pon.html>

- A,BC,D – unicastová data pro konkrétní stanice, v - multicastová data.
- ONT = Optical Network Terminal, zakončení optické sítě pro konkrétní koncovou stanici.
- Ke každému přepínači je vedeno jedno nebo více optických vláken. Přepínač přeposílá data dále přes jeden konkrétní výstupní port.

# Princip pasivní optické sítě (PON)



Převzato z <http://www.fiberopticshare.com/ftth-access-networks-aon-vs-pon.html>

- **A,B,C,D – unicastová data pro konkrétní stanice, v - multicastová data.**
- **OLT** = Optical Line Terminal, zakončení optické trasy, která je určena pro více konkrétních koncových stanic. Stanice jsou připojeny prostřednictvím OLT.