

• Proces předání správy

- pouštění → list papíru - databáj jednotka
 - vložení do obálky - zařazení / encapsulation
 - napsání adresy + malé číselní rámečky - dodržování protokolu
 - hození do schránky - odeslání
 - pošta doručí správu - svaří nebo propojí různými módami
 - obdrží ji příjemce - cílový uzel
 - při jeho otevření obálku - rozbalení / decapsulation
 - rozbalíme správu - správa doručena

- e-mailem → rozdělení a zařazení podle správného protokolu ⇒ databáj jednotka
 - po místní síti doručíme správu sítové infrastruktury
 - ta ji nadále doručí až na konečný uzel (počítač příjemce)
 - příjemce ji rozbalí, dešifruje a zobrazí na displeji

• Odolnost

- americká armáda řešila problém výpadku telefonních sítí v případě války
- telekomunikační sítě - přepojování okruhů
 - vlastní: síť najde nejkratší vlnu (okruh), kterou spojí obě strany
 - ⇒ rychlé, plynule, ale při výpadku vlny se spojení rozpadne
 - chyběli sítě odolnost vůči chybám
- přepojování paketů
 - data rozdělíme na malé bloky - pakety - a každý paket si najde vlastní cestu k cílovému uzel
 - pokud je některý uzel napaden, tak si paket najde jinou cestu
 - ⇒ formálnější, proměnlivá doba přenosu, spolehlivější

• Bezpečnost

- na počátku internetu se mimo jiné objevily softwarového útoku
 - ⇒ nejdříve protokoly nepoužívaly šifrování a důvěrovatelnost i obsah dat
- bolesti se fyzického útoku na infrastrukturu - uky a kabely - bomby

• bezpečnostní rizika

- fyzické napadení infrastruktury
- útok na data - neoprávněná manipulace
- DoS (Denial of Service) - zahľadom' zdrojuje dat - nemůže fol komunikacii s uživateli
- DDoS (Distributed DoS) - útočník využije cizí servery, aby zvýšený prúd generovaly nevidomky místo nej

• Rozšíritelnost

- pridání počítače / sítě musí byt snadné

• LAN (Local Area Network)

- Core vrstva - je připojena na infrastrukturnu ISP (Internet Service Provider)
 - router = uzel propojující různé sítě
 - místní router je připojený k routernu ISP
 - několik klavních switchů propojujících klavní router se zbytkem LAN

• Distribuční vrstva - vertikální

- distribuuje konektivitu do všech částí budovy

• Access vrstva - horizontální

- umožňuje připojení koncovým zařízením

• WAN (Wide Area Network) - opak rozdílení na 3 vrstvy

- Tier 1 - klicoví hráči internetu připojení přímo na páteř internetu
- Tier 2 - společnosti (národní operátori), jejímiž rákarníky jsou jiní ISP
- Tier 3 - ISP, kteří připojují koncové rákarníky - společnosti, domácnosti a LAN

• Kvalita služeb

→ prenosové parametry sítě

- Latence - zpoždění = doba doručení'
- Jitter = rozptyl zpoždění - pravidelnost doručování'
- Záťatkovost - je to často docházejí k tomu, že nějaký paket nemá doručen
- Síra písma - rychlosť - kolik dat lze rozdělat a přenášet = bandwidth

→ různé aplikace mají různé požadavky

- multimedialní aplikace - pravidelnost doručení'
- prenosy dat (email) - nízká záťatkovost dat

Kvalita služeb

- cíl - garance vymenovaného toku pro konkrétní typ proudu
- garance rychlejšího doručení prioritních zpráv

implementace

- data obsahují klasifikaci QoS (Quality of Service)

strategie garance kvality

- část kanálu vyhradíme jen pro prioritní zprávy
→ různá kvalita, plýtvání kapacitou

strategie best effort

- u každého uzel je prioritní fronta
→ efektivní využití médií, není různá kvalita

Vznik počítačových sítí

1) oddělené počítače - přenos dat na děrných štítcích

2) počítače + terminály - point to point komunikace

3) vznik LANer, které dohromady utvořily WAN

→ vznik client-server aplikací, kde je část výkonu vykonává klient

Základní dělení sítí

- LAN - sdílení zdrojů na malé vzdálenosti - jednotliví vlastníci a řízení
- WAN - globální sítě, vzdálený přístup, mnoho vlastníků - neponáří ji ten, kdo ji využívá
→ rozdíly se dnes stírají, ale sítě LAN byly jen soukromé

Veřejné a privátní sítě

- když chceme propojit dvě LAN sítě přes veřejnou síť → problém s bezpečností
- ⇒ VPN (Virtual Private Network) - na hranici obou privátních sítí děláme rozvratí, který enciuluje / dešiuluje zprávy ⇒ vytvoří VPN tunel mezi veřejnou sítí
- pro počítače v obou částech LAN se celá síť hraví jako jedna LAN
- případně může být jedna strana tunelu nahrazena SW na počítači

RFC (Request For Comments)

- standardizace internetu - veřejné
- obsah dokumentu se nemění
- ne všechny RFC jsou dodrženy

- sítový model - popisuje vrstvy, jejich strukturu a vztopy - OSI model
- sítová architektura - model + konkrétní služby, technologie, protokoly,... TCP/IP
- OSI (Open Systems Interconnection)
 - model - vhodný pro dokumentaci a výuku
 - | protokoly - nepraktické, budovány shora
 - 1) fyzická - fyzický prenos bitů mezi vrstvami - hub, repeater
 - 2) linková - prenos datových rámci mezi sousedními vrstvami - switch
 - 3) sítová - směrování mezi sítěmi / mezi vzdálenými vrstvami - router
 - ↳ prenos datových bloků s proměnlivou, ale omezenou délku - paketu
 - 4) transportní - prenos dat neomezené délky mezi aplikacemi
 - segmentace příliš velkých bloků
 - 5) relační - řídí dialog mezi aplikacemi
 - 6) prezentativní - datové konverze pro aplikace
 - 7) aplikativní - komunikace mezi programy, interakce mezi uživatelem a aplikací
- X.400 (Message Handling System) = OSI pošta, komplikovaná, ale jednoznačná adresace
- X.500 (Directory Access Protocol) - adresářové služby, telefonní seznam ↳ adresace podle X.400
- následující:
 - X.509 (Public Key Infrastructure) - správa veřejných klíčů
 - LDAP (Lightweight DAP) - databáze informací o uživatelích a službách

Rodina protokolů TCP/IP

- nároh odspranu, praktické → od jednoduchých k složitým
↳ prototypy

OSI	VRSTVA	PROTOKOLY		
7	aplikací	FTP	DNS	NFS
6		HTTP	SIP	XDR
5		SMTP		RPC
4	transportní	TCP		UDP
3	síťová		IPv4 / IPv6	
2	síťové	Ethernet, WiFi, ATM		
1	rozhraní	FDDI, SLIP, PPP, ...		

→ většinou je leží definovat všechny
v aplikací vrstvě, ale jsou
výjimky → NFS + XDR + RPC

↳ složí mimo hierarchii

← Internet Protocol

← protokol podle média

TCP (Transmission Control Protocol)

- pro spojené služby - telefonní signál
- zaručeno spolehlivé / reliable doručení dat
 - TCP data segmentuje na pakety, počítáje seřízení domácím, případně je posílá znova
 - implementace TCP je složitá, ale aplikace je jednoduší
 - aplikace nemusí řídit komunikaci

UDP (User Datagram Protocol)

IP je vše 'unreliable'



- pro nespojené služby - pošta
- není zaručeno doručení ani pořadí paketů → unreliable = nespolehlivé
- kontrolu musí provádět aplikace → může řídit komunikaci
- UDP je jednoduché, ale aplikace složitá

Aplikací modely

Klient - server

- klient má svou adresu serveru
- klient navazuje komunikaci, zadává požadavky
- server obvykle obsluhuje více klientů
- download (S → K), upload (K → S)
- např. DNS, WWW, SMTP

peer-to-peer - P2P

- partneři neznají své adresy zdrojedat
- nejsou vymezeny role → každý je zároveň klient i server
- BitTorrent

sjednačování aplikací

→ může tam uplodat
nelegální data

→ sdílet si je může download

→ nelegální distribuční

Adresování počítačů

6B

- MAC adresa - HW, linková vrstva, ethernet: $\underbrace{8:0:20:AE:6:1F}_{6B}$
 - (Media Access Control)
 - nerespektuje topologii sítě
 - současné síťové karty mají MAC adresu v paměti \Rightarrow lze zjistit
- IP adresa - SW, síťová vrstva, IPv4: $\underbrace{195.113.19.41}_{4B}$
 - přidělována počítači podle topologie sítě
 - určuje jednoznačnou síť a v jejím rámci počítač
- Domeinová adresa - lidé, aplikativní vrstva, $\underbrace{\text{www.mff.cuni.cz}}_{\leftarrow}$
 - přidělená podle organizační struktury, hierarchie \leftarrow
- DNS (Domain Name System) IP \leftrightarrow Domain
- ARP (Address Resolution Protocol) IP \leftrightarrow MAC

DNS

- hierarchická struktura zón, jež obsahují info o podřízených počítačích a zonách
- tyto informace jsou uloženy v databázi nameserveru - poskytuje klientům odpovědi
- každý počítač by měl mít přiřazené doménové jméno - ale může mít i více jmen
 - ↳ prode domény, mají sliby na něm být
- správa domén

• TLD (Top Level Domain) - spravuje ICANN - původně příslušné polohy

→ Technické - arpa

→ rezortní pro USA - com, org, edu, mil, gov, net

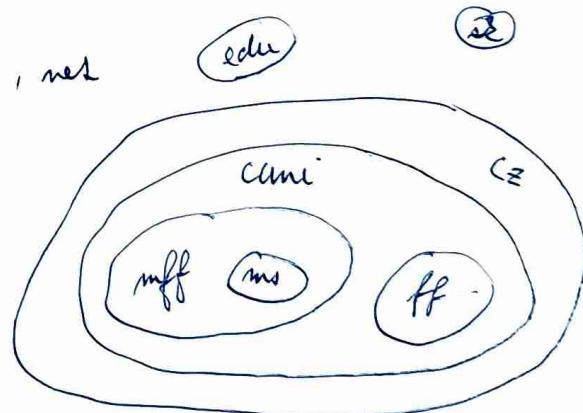
→ ISO kódem řemi - cz, sk, ..., uk, en

• SLD (Second Level Domain) - spravuje ji majitel

→ doménu cz spravuje CZ.NIC

↳ což znamená, že CZ je SLD

→ hierarchická správa domén



⇒ TLD spravuje centrální organizaci a pak ji reprezentuje celá hierarchie

• IP adresy

- každý koncový uzel v síti TCP/IP musí mít IP adresu
- IPv4 : 4 byty \rightarrow 195.113.19.71
IPv6 : 16 bytů \rightarrow 2001:418:1E03:Q01::1 \leftarrow hex pr 2B
- nejvíc náročnějších bloků nahradit ::
- → verejné adresy (středním komunikoval se různým světa) přiděluje ISP
- soukromé adresy (střední lze používat jen v rámci LAN) přiděluje správce LAN
- ↑ přiřazení adresních bloků sítě - prefix
- přiřazení adresy počítači v síti
 - statičtí - každý uzel má vždy stejnou adresu
 - dynamické - adresa je přiřazena na výžádání
 - volné - k síti se může připojit kolikkoliv
 - omezené - pro připojení je třeba se autentizovat
 - platí i pro privátní adresy
 - ↳ výjimka: link-local adresa - počítač si vyměňuje vlastní adresu v rámci segmentu sítě ve které je

• Port - 16 bit. číslo identifikující jeden konec spojení - aplikaci, která má příchozí porty

- destination-port musí klient znát \rightarrow well-known services
- source-port přiděluje operační systém k neobsazených čísel portů
- spojení v TCP/IP: <zdrojová IP, zdrojový port, cílová IP, cílový port, TCP/UDP>
 - dva různé kanály stejné aplikace se musí lišit alespoň ve zdrojovém portu
 - stejná čísla portů lze použít pro 2 různé kom. kanály, pokud používají různé protokoly
- příklady well-known services
 - 21/TCP - FTP (File Transfer Protocol) - přenos souborů
 - 22/TCP - SSH (secure Shell) - vzdálené přihlašení a přenos souborů
 - 25/TCP - SMTP (Simple Mail Transfer Protocol) - přenos elektronické pošty
 - 53/* - DNS (Domain Name System) - předlady mezi doménami a IP
 - 80/TCP - HTTP (Hyper Text Transfer Protocol) - přenos webových stránek
 - 443/TCP - HTTPS - šifrované HTTP

• Socket - 1 konec komunikačního kanálu mezi klientem a serverem

- adresa 1 konce kanálu \Rightarrow <IP adresa, port>

NAT (Network Address Translation)

- princip: Edy lokální síť používá privátní adresy a ven se představuje nějakou verejnou adresou = IP masquerading
- implementace i terminologie se v detailech liší
- princip: Edyže klient z LAN něco poslat ven, tak router na perimetru LAN upraví obsah paketu tak, aby server odpočítel jemu
⇒ router si uloží socket klienta - edyž přijde odpověď, tak mu ji přepošle

Adresní služeb

- URI (Uniform Resource Identifier) - jednotný systém odkazů

↳ historicky URL (umístění) a URN (název) dnes $URI \sim URL$
pro re file systemu identifikator bodu na stránce (název)

$URI = \text{schema} : [//] \text{autorita} [\text{cesta}] [?\text{dotaz}] [\#\text{fragment}]$ $[] = \text{volitelné}$

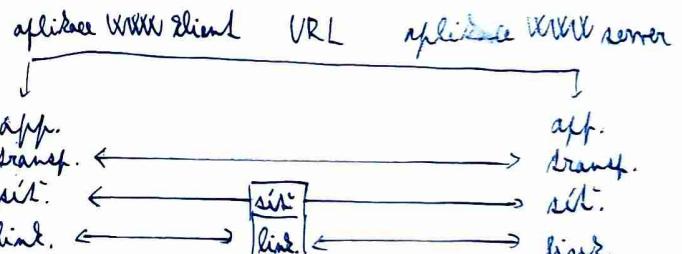
autorita = [jméno [: host] @] adresa [: port]

info o uživateli ↑ Doména ↑ poleh server nepřevezme well-known port

ftp://sunsite.mff.cuni.cz/Net/RFC

http://1.2.3.4:8080/?ID=123#Local

mailto: foxt@uni.cz



Client ↔ switch ↔ router ↔ switch ↔ server

Další krok v TCP/IP

- aplikativní vrstva - klient adresuje server pomocí URL

→ aplikativní vrstva předává data spolu s cílovým socketem transportní vrstvy

- transportní vrstva - identifikace obou konci komunikačního socketu

→ předává data a cílovou adresu sítové vrstvě

- sítová vrstva - identifikace IP adresami

→ předává data a MAC adresu next-hop vrstvy

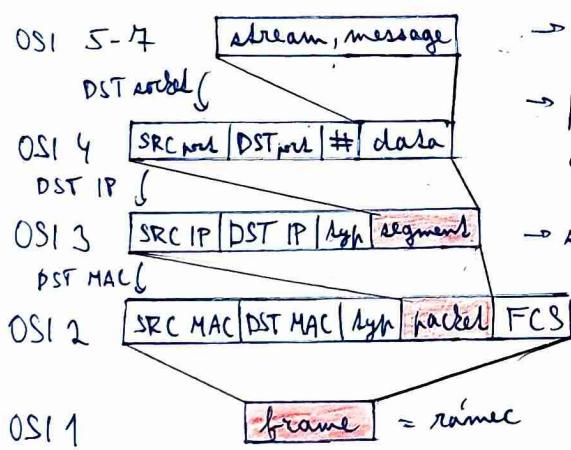
- linková vrstva - identifikace MAC adresami

→ fyzické fyzická vrstva doručí data na next-hop vrstvu, tak bud' to je cílový server,

zakří data předáme vysokém vrstvám. Nebo router → data dostane sítová vrstva → hop...

- Multiplexing - několik komunálních vnitřních vstřívá stejný kanál v různé vlny
- Protocol - obě strany musí dodržovat protokol.
→ na 1 vlně může probíhat komunikace ve více protokolech současně
- Encapsulation
 - majme na vlně n protokol, který definuje PDV (Protocol Data Unit) kdo vlny.
 - SW n-té vlny přidá k PDV n řídící informace a zavola službu následující vlny
 - tato funkce pořadí n-té vlně info o selhání / úspěchu n-1. vlny
⇒ kdo výměně dat mezi vlnami se nazývá interface a jejímu formátu (prikyry + PDV n) se říká IDU (Interface DV)
 - vlna n-1 převzme řídící informace a sestaví PDV n-1, jde o PDV n a header s řídícími informacemi ← reforwarding
 - header musí obsahovat identifikátor vlny n → příjemce provede decapsulation a demultiplexing a předá data vlně n.

Typy PDU a TCP/IP



- zpráva (nespojovaná app), proud (spojující app)
- proud aplikace vyžádá TCP, takže blok dat segmentuje a připraví header s # offsetem dat v rámci proudu
- síťová vlna & segment přidá header s číslem protokolu transportní vlny ⇒ packet
- linková vlna & packetu přidá header s číslem protokolu na síťové vlně a footer obsahující FCS (Frame Check Seq.), což je číslo vyplňané ze zbytku frame / rámcem

- v úložném uzel fyzická vlna data dekóduje a předá linkové vlně
- linková vlna přepočítá FCS a kontroluje, jestli se hodi (proud ne, obsah se změnil)
 - ↳ kontroluje cílovou MAC a podle čísla protokolu předá síťové vlně rozbalený paket
- síťová vlna kontroluje IP a předá segment příslušnému SW transf. vlny
- rozsah práce transformní vlny závisí na použitém protokolu
 - UDP - zpráva je předána aplikaci
 - TCP - segment je uložen a celý datový blok bude aplikaci předán až po přijetí všech segmentů

Bespečnost

uživatelské

- Autentizace = proces ověření identity subjektu ← kdo jsem? server
 - Autorizace = proces, když uživatel přiřazuje identifikovanému subjektu oprávnění
- lokalné lze autentizovat pomocí:

- enzlostí - heslo, PIN, ...
 - ⊕ snadná implementace, jednoduché sdílení
 - ⊖ důvěryhodný fakt může být rozpracován
- technických prostředků - HW token, elektronický klíč ← co mám?
 - ⊕ je to bezpečnější
- biometrie - otisk prstu, sken očnice
 - ⊕ nejbezpečnější

Vzdálená autentizace

- problém: kanál může být odpojován. ⇒ rozpracování hesla
- ⇒ OTP (One Time Password)
- nebo pomocí kryptografie vyhovující bezpečnosti kanál
- problém: většina protokolů nemá build-in ověření identity
- ⇒ navrhuje framework SASL (Simple Authentication and Security Layer), který dokáže do většiny důležitých protokolů včlenit různé metody ověřování
- po každý protokol 3 profily, který určuje, jak autentizaci v daném protokolu provádět
- možnost využít autentikačního serveru, který poskytuje speciální protokol na komunikaci s klientem a serverem
- ⇒ server = proklytoratel služby, a. server = poskytovatel identity
- a. protokoly: LDAP, RADIUS, NTLM, Kerberos, SAML
- OTP = mechanismy umožňující nepřekrutebnou plain-textovou autentizaci uživatelské
- 1) historyky → vyklopný seznam jednorázových hesel
 - 2) challenge-response → server posle uživateli na klienta jedinečný kód a uživatel jej pomocí svého hesla s kalkulačkou s kombinuje se svým heslem → odpovídá
 - 3) HV/Tokeny → uživatelský dostane speciální autentikační zařízení, které je sesynchronizováno se serverem a generuje kódy pro identifikaci
- platnost kódů je několik tisíc a 1 použití

Kryptografie

→ velmi důležitá - pro šifrování a el. podpis se využívají 3 základní typy algoritmů

symetrické šifrování

- historie: addiční, transpoziční, substitucní sifry, šifrovací maticy, ...

- dnes: metody založené na matematické teorii

→ pro šifrování i dešifrování se používá stejný klíč

→ příklady: DES, Blowfish, AES, RCG

⊕ rychlé, vhodné pro velká data

⊖ partneři si potříd musí nejde bezpečně předat klíč

asymetrické šifrování

→ pro šifrování a dešifrování se používá páru nazájem neodvratitelných klíčů

→ odesílatel veřejným klíčem zprávu rozšifruje a příjemce ji sám vlastní dešifruje

→ matematický základ - jednocestné funkce

↳ lze šifrovat žádat
a dešifrovat něco jiného

• násobení $x = a \cdot b$ X rozklad na prvočinitele

• diskrétní logaritmus $y = g^x \text{ mod } q \rightarrow x=?$

→ příklady: RSA, DSA, ECDSA

⊕ veřejný klíč lze šířit, sám však není problém schránkování

⊖ pomalejší, vhodné jen pro malá data

→ veřejný klíč je třeba pečlivě chránit !

hashovací funkce

→ vyhroží první kód z daného textu

→ široké uplatnění → kontroly shod, hashovací tabulky → CRC, MD5

o kryptografii:

→ malá změna textu = velká změna hashe ← slovo jednoznačné hashování

→ jednoznačnost → text je z hashe neodvratitelný

→ SHA

→ malým textu se stejným hashem je obtížné

Sifrování dat - sym + asym

- text sifrujeme symetrickou sifrou a její klíč sifrujeme veřejným klíčem příjemce a toto ale' mu odesleme
- příjemce svým tajným klíčem dešifruje klíč a pomocí něj i správu

Elektronický podpis - asym + hash → je jedno jestli je reč o reálném sifrování

- odesílatele znali hashovací funkci, nevěděl text a vypočítá jeho hash
- ↳ ten hash sifruje svým soukromým klíčem
- ⇒ text, sifrovaný hash a hashovací funkci odesle příjemci

- příjemce si sám spočítá hash přijatého textu a dešifruje přijatý hash veřejným klíčem odesílatele. Potom oba hashe sedí, tak:

1, někdo nemanipuloval s textem ← vysílající hash

2, správně sloučeně odesídal někdo s přístupem k soukromému klíči odesílatele

meant me odesílatele

Diffie-Hellmannův algoritmus

napi. symetrický klíč

- rozložit vyměněnou informaci nezabezpečeným kanálem, aby oba kresli sdílené tajemství
- používá diskrétní logaritmiku

1, A vygeneruje tajné číslo a a veřejná (pivo) čísla p, q

2, A spočítá číslo $A = p^a \text{ mod } q$ a posle $p, q, A \rightarrow B$

3, B zná tajné číslo b , spočte $B = p^b \text{ mod } q$ a posle $B \rightarrow A$

4, A spočítá $s = B^a \text{ mod } q$ a B také spočítá $s = A^b \text{ mod } q$

→ princip: $B^a = (p^b)^a = p^{ba} = p^{a+b} = (p^a)^b = A^b$

- bez znalosti a, b a při volbě velkých p, q je spočítání s nerešitelné

→ při odchycení A,B

Autentizace veřejných klíčů

- je třeba ověřit, že jméno = identifikacím enaké patří ke klíči

- autentizaci ověří třetí strana a připojí svůj podpis

1) Web of Trust - nevratelé potvrzují autentizaci klíčů dalších nevratelů

↳ když to podvodil někdo komu důvěřujeme → ✓

2) Public Key Infrastructure - klíče poštovají speciální organizace

X.509 → Certification Authorities (CA) → poštovají klíč + směrka a den, když důvěřuju CA provozuje na důvěryhodný i klíč

• Certifikát

- = klic + identifikace vlastníka ← podepsaný vydařatelem - např. CA
- pokud důvěřujeme vydařateli, tak i klici
- řetězec důvěry: je třeba věřit CA → kontáme se na certifikáty CA do kde nedospíjeme & nejdále CA, které důvěřujeme

→ struktura certifikátu podle PKI

- certifikát - verze certifikátu
- sériové číslo
- vydařatel
- doba platnosti
- vlastník nejmenšího klíče
- info o klíci (algoritmus a klíč)

- algoritmus pro elektronický podpis
- elektronický podpis

→ můžete stahovat certifikáty
a kontrolovat podpis pomocí
verejného klíče CA, která
ho vydala

• SSL & TLS

- SSL (Secure Socket Layer) se ve verzi 3.0 přejmenovala na
TLS (Transport Layer Security) 1.0 → dnes se používají TLS 1.1 +
- speciální mezivrstva mezi transportní a aplikacní vrstvou umožňují
autentifikaci a šifrování
 // HTTP + SSL
- využívá to řada starších protokolů → HTTPS na portu 443
- princip:
 - 1) klient pošle požadavek na SSL s pojemem + parametry
 - 2) server pošle odpověď + parametry + svůj certifikát
 - 3) klient ověří server a vygeneruje rácklad šifrovacího klíče → pošle ho serveru
 - 4) server rozšifruje rácklad klíče. Z toho ráckladu vygeneruje server i klient celý klíč
 - 5) klient a server si naročují potvrdit, že od této chvíle jejich komunikace šifrována

zajištěny metody
sítěm souboru

Aplikační vrstva TCP/IP

- spojuje funkce OSI 5, 6 a 7
- protokol na aplikační vrstvě definuje
 - průběh dialogu - kdo iniciouje spojení, ...
 - formát správ - textový / binární, struktura
 - semantiku správ a informačních polí - která část správy znamená co
 - typy správ - jaké jsou požadavky a odpovídají na ně
 - interakce s transportní vrstvou - TCP / UDP - kdy, jak?
- Domain Name System - DNS doménových IP
 - klient - server aplikace pro překlad jmen na adresy a naopak
 - binární protokol nad UDP i TCP, port 53
 - běžné dotazy (odpověď do 512 v novém EDNS) se vykonávají v UDP
 - větší dotazy výměnou probíhají v TCP
 - Client se obrací na DNS servery, jejichž adresy jsou zadány ve své konfiguraci
 - národní se dozvídá co potřebuje
 - počet odpovědí neobsahuje potřebné info, měla by obsahoval odpaky za servery, kterých je třeba se plátkat dál
 - jednotka dat je zářenam (Resource Record - RR) např:
mff.cuni.cz 3600 IN A 195.113.19.48 TTL = doba platnosti v sekundách
↳ jméno zářenamu ↳ TTL ↳ typ ↳ data
 - každá správa obsahuje hlavičku a množství položek zářenamů
 - Typy DNS zářenamů
 - SOA (Start Of Authority) - výrodní zářenam → informace pro dalšího posledního změny, ...
 - NS - zářenamy definující nameservery, které udržují databázi zářenamů dané domény
 - A - IPv4 adresa pro dané jméno
 - AAAA - IPv6 adresa — } } poslouží jmenu na adresy
 - PTR - reverzní zářenam pro poslání adres na jména
 IPv4: 1.2.3.4 → 4.3.2.1.in-addr.arpa
 IPv6: ::1 → 1.0...0.0.ip6.arpa ← poslouží odlehlejším sečína
 - CNAME - zářenam pro tvorbu aliasů → alias --> kanonické (strukturní) jméno počítače
 - MX (Mail Exchanger) - řídká, který server přijímá pro danou doménu (počítač) poslán

Typ	Jméno zájnamu	Data
SOA	jméno domény	obecné informace o doméně
NS	jméno domény	jméno nameserveru domény
A	jméno počítače	IPv4 adresa počítače
AAAA	jméno počítače	IPv6 adresa počítače
PTR	reverzní jméno (např. pro IP adresu 1.2.3.4 je to 4.3.2.1.in-addr.arpa, pro ::1 je to 1.0...0.ip6.arpa)	doménové jméno počítače
CNAME	jméno aliasu	kanonické jméno počítače
MX	jméno domény/počítače	jméno poštovního serveru a jeho priority

Aplikační vrstva TCP / IP

- spojuje funkce OSI 5, 6 a 7
- protokol na aplikační vrstvě definuje

- průběh dialogu - kdo iniciuje spojení, ...
- formát správ - textový / binární, struktura
- semantika správ a informačních polí - která část správy znamená co
- typy správ - jaké jsou požadavky a odpovědi na ně
- interakce s transportní vrstvou - TCP / UDP - kdy, jak?

Domain Name System - DNS

doménových IP

- klient - server aplikace pro překlad jmen na adresy a např.
- binární protokol nad UDP i TCP, port 53
 - běžné dotazy (odpověď do 512 v non EDNS) se vykonají v UDP
 - větší dotazy výměnou probíhají v TCP
- Client se obrací na DNS servery, jejichž adresy jsou zadány ve své konfiguraci
 - uživatel se dozvídá co potřebuje
 - pokud odpověď neobsahuje potřebné info, měla by obsahovala odkazy na servery, kterých je třeba se plátkat dál
- jednotka dat je ráčenam (Resource Record - RR) např:

mff.cuni.cz 3600 IN A 195.91.3.19.48 TTL = doba platnosti v sekundách
↳ jméno ráčenamu ↳ TTL ↳ typ ↳ data

- každá správa obsahuje klauzulu a některý počet ráčenám

Typy DNS ráčenám

- SOA (Start Of Authority) - úvodní ráčenam → informace jako datum poslední změny, ...
- NS - ráčenam definující nameservery, které udržují databázi ráčenám dané domény
- A - IPv4 adresa pro dané jméno } pokud jmena na adresy
- AAAA - IPv6 adresa ——— }
- PTR - reverzní ráčenam pro převod adres na jména
 - IP v4: 1.2.3.4 → 4.3.2.1.in-addr.arpa
 - IP v6: ::1 → 1.0...0.0.ip6.arpa ← nulby byly odděleny čárkami
- CNAME - ráčenam pro tvorbu aliasů → alias --- kanonické (skutečné) jméno počítání
- MX (Mail exchanger) - říká, který server přijímá pro danou doménu (počítací) poštov

• Servery DNS

- primární (master) server - spravuje záznamy o doméne
- sekundární - pravidelně stahuje a zálohují aktuální obsah databáze ↗
- caching-only - pokud se k záznamům dostanou i nějaké další servery, tak si je jen dočasně uloží do cache - dokud je potřebují

→ každá doména (róna) musí mít alespoň 1 autorizační (prim./sek.) server
→ v SOA záznamu je uvedeno, jak často mají sekundární servry aktualizovat
databázi → výzadují se data od primární
→ primární server může minovitě rychle sekundářům akt. se aktualizují
→ pro výměnu dat se používá TCP

• Vyřizování DNS dotazu

- vývratel rada www.mff.uni.cz
- vygeneruje dotaz pro nameserver r. domény, kde dotaz venek
- bude rekurzivní = server převzme odpověď od svého výzadu
- pokud tento nameserver nemá v cache info o hledané doméně → hledání
 - ↳ neví nic o mff.uni.cz, uni.cz ani cz
- ⇒ obrať se na korenový nameserver - který ale neposkytuje rekursivní odpověď
- ve své databázi najde nejrelevantnější položku a tu pošle
 - ↳ „posli svůj dotaz nameserveru s adresou ...“.
- server si tento info uloží do cache a počítá s dotazem k nově získanému serveru
- makrve se dostane k nějakému autorizačnímu serveru, který zná odpověď
- tento konecnu odpověď pak přepoše klientovi

• DNS dotaz a odpověď

• Dotaz

ID - náhodné 2B číslo

FLAGS - funkce

QUERY - 1 záznam bez dotazové části

• Odpověď

ID

FLAGS

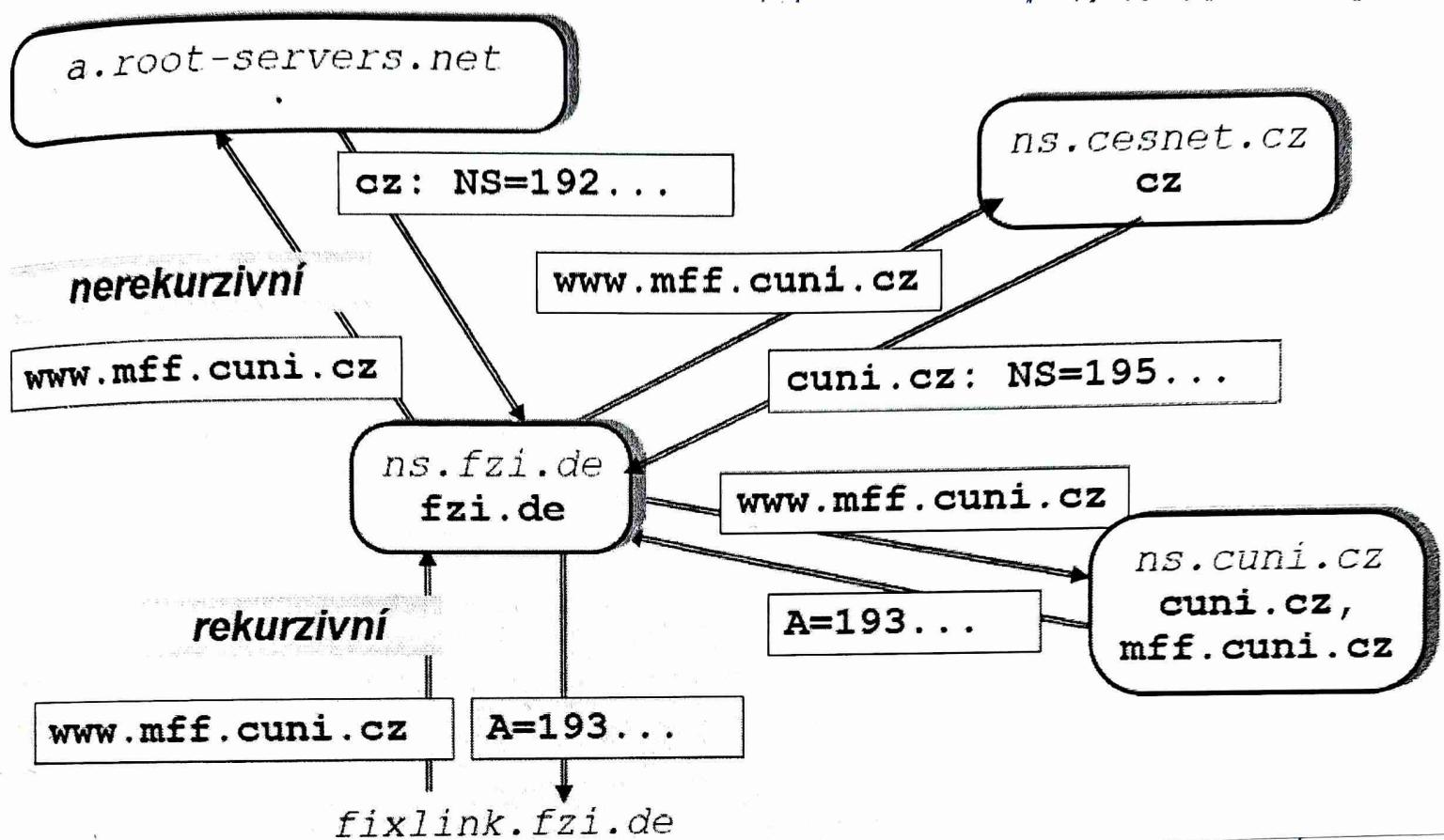
QUERY

ANSWER - RR s odpovídající

AUTHORITY - seznam nameserverů, které mají
dále autorizační odpověď / informaci

ADDITIONAL - adresy nameserverů

↳ adresy MX serverů a ANSWER ...



* Client posílá nádej s příkazy, server řádce s odpověďmi

→ SMTP Only

FTP (404 - stránka nebyla)

a, budou ještě další odpovědi,

• Dotaz:

ID: n
 FLAGS: Recursion Desired
 QUERY: www.cuni.cz. IN A

• Odpověď:

ID:	n		
FLAGS:	Authoritative Answer		řádkové jméno
QUERY:	www.cuni.cz. IN A		www.cuni.cz
ANSWER:	www.cuni.cz. IN CNAME tarantula		
	tarantula IN A 195.113.89.35		řádky - mají heslo)
AUTHORITY:	cuni.cz. IN NS golias		server → neplatí se)
ADDITIONAL:	golias IN A 195.113.0.2		- smysl příkaz opakovat)

řádky - mají heslo)

server → neplatí se)

- smysl příkaz opakovat)

Beezpečnost DNS

→ problém nárovníka: jak se dostat ke zdroji dat, abych mohl např. poslat falešnou odpověď?

→ je těžké ho odchytit

→ vyhnout si ho nemůže - nahodilý zdrojový kód + ID

cache-poisoning

- když klient požádá o serveru nárovníka, tak nárovník může správně naplnit sekci ANSWER, ale do AUTHORITY a ADITIONAL přidat falešné údaje o jiné doméně
⇒ riziko kompletní kontroly nad dotazy směřující do domény (vítka) (z)

- řešení: poskytovat od root serveru a plátku se pravce autorizačních serverů

DNSSEC = DNS zabezpečené podpisy

- je konfliktní a rozšiřuje se pomalu

Diagnostika DNS

→ cmd: nslookup

→ UNIX: dig

File Transfer Protocol - FTP

- jeden z nejstarších protokolů

- původně sloužil ke vzdálenému přístupu k vlastním datům pomocí otevřeného leska !!!

⇒ dnes hlavně anonymní přístup - uživatel anonymous/ftp, heslo je email

→ uživatel má rádce přístup k volně dostupným datům

→ je to tekutý protokol → klient může s různými spojeními na server na portu 21

↳ klient posílá rádce s příkazy, server rádce s odpověďmi

Kódy odpovědí

↗ SMTP kód

- každá odpověď začíná XXX kódem → převzal to Iciba HTTP (404 - stránka nebyla)

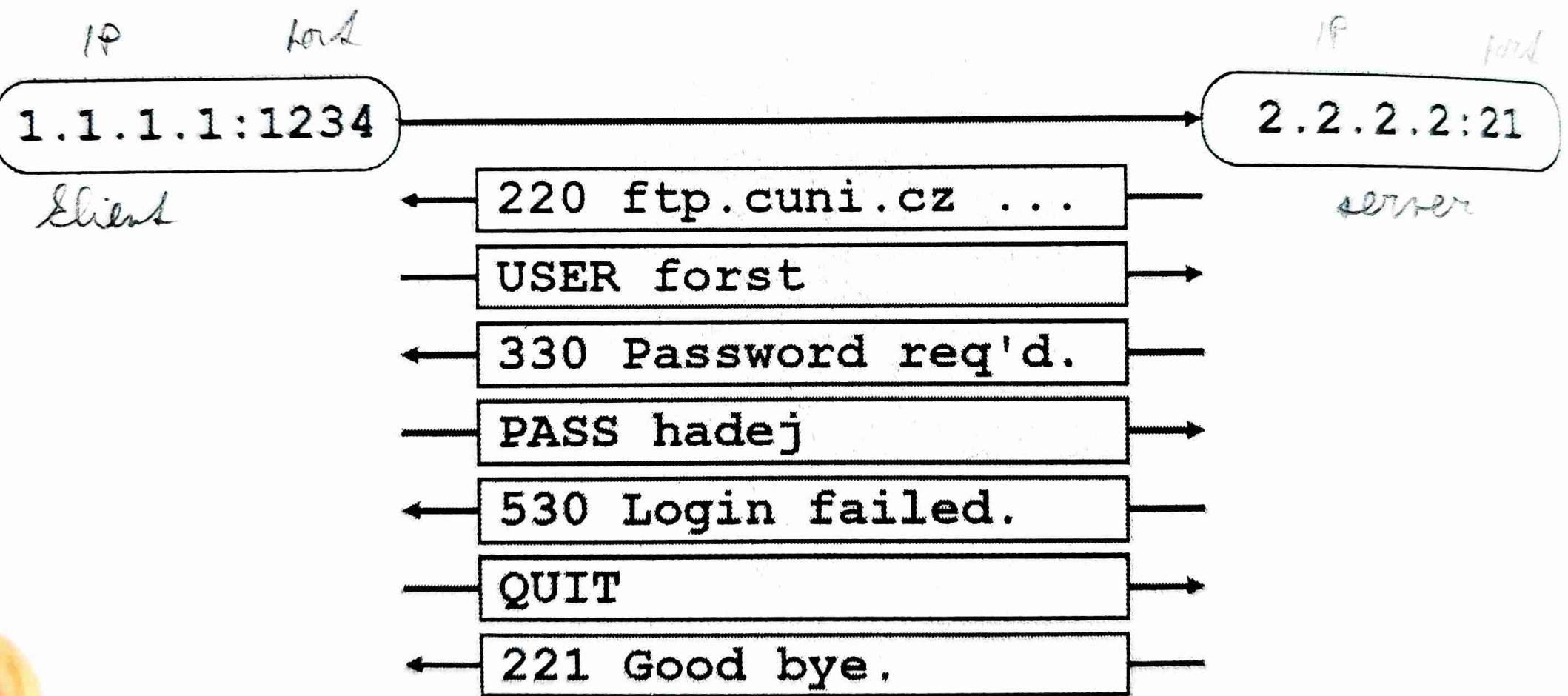
- 1XX = předbežná klidná odpověď (akce byla zahájena, budou ještě další odpovědi)

- 2XX = definitivní klidná odpověď

- 3XX = neúplná klidná odpověď (jsou nutné další příkazy - např. heslo)

- 4XX = dočasná záporá odpověď (Iciba je přetížený server ⇒ nepovedlo se)

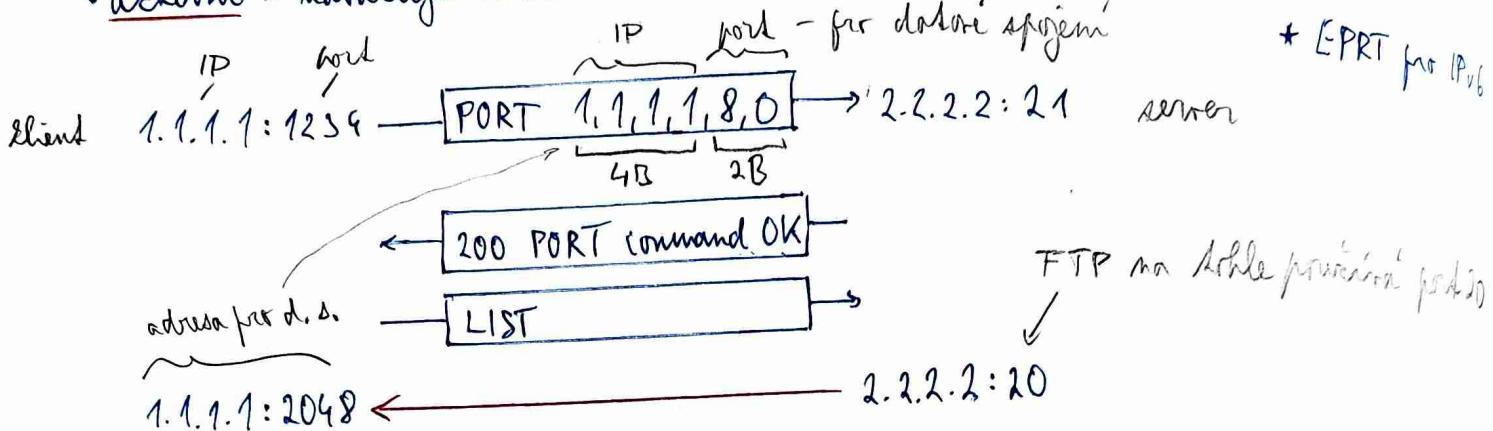
- 5XX =紕valá záporá odpověď (nepovedlo se a nemá smysl příkaz opakovat)



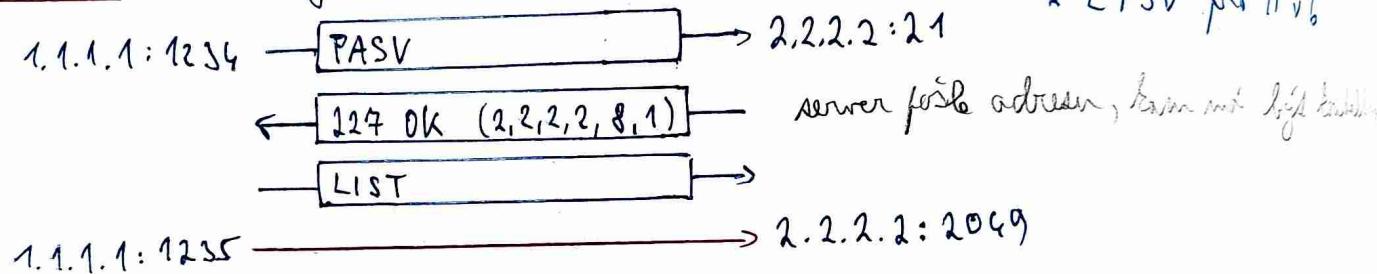
aktivní / pasivní datové spojení - FTP

- prenos dat probíhá po krv. datovém spojení → přes TCP
- maximálně se dohodnout, kde bude kanál sloužit a na jaký socket

• aktivní - navazuje server



• pasivní - navazuje klient



→ po skončení prenosu dat se datové spojení uzavře

→ aplikace pro FTP: WWW prohlížeč, správci souborů, cmd příkaz ftp

Elektronická pošta

- obecná služba existující i mimo internet
- předávání e-mailů (pozdeji i souborů) v lokální síti \Rightarrow offline
- offline přístup k informačním službám - FTP archiving
- na internetu používá SMTP na TCP portu 25 \leftarrow sekvenční protokol
- e-mailová adresa v internetu (typicky):

login@počítac

foret@ms.mn.mff.cuni.cz

mátej poštovní schránky
server, kde schraňtu leží

↳ bezpečnostní nároky

alias @ doména

libor.forejt@cuni.cz

- původně binární e-maily (64 kB), později posílání souborů

Příjem a odeslání pošty v SMTP

→ uživatel využívá příkaz k odeslání dopisu

1) poštovní program kontrolouje adresu za @ a sjistí, který server bere poštu pro tento doménu

a) je možné, že meri klientem a serverem některí rádce pletácky ⇒ přímé doručení

2) mail submission - program předá dopis fórovému SMTP nejprve serveru, který má uloženou konfiguraci → nazývá se mail-forwarder ↑ MTA

3) vtedy, když přijmaje a doručuje poštu nazýváme Mail Transfer Agent (MTA)

⇒ jednotlivé MTA si pomocí SMTP předávají dopis

⇒ server, který dopis přijme, si ho vtedy dočasně uloží do fronty a pak pošle dál

4) pokud je cílový mailbox na serveru přístupném z internetu, tak se poslední MTA pokusí doručit dopis na tento server

→ správce toho serveru může v DNS nastavit nějaké mail-exchangers

⇒ potom k poslednímu MTA bude dopisy posílat na ně ↓

MX je ochotný přijmout dopis
a doručit ho do ale

Přístup k poště z pohledu uživatele

→ uživatel přistupuje k poštovnímu systému pomocí nějakého poštovního programu

⇒ Mail User Agent (MUA) - 2 možnosti připojení

• přímé připojení - uživatel se připojuje na MTA, kde má svůj mailbox

- aplikace má přístup k dopisu uživatele ⇒ příjem

- zároveň ————— s sloučením MTA ⇒ odesílané správy rádu přímo do fronty MTA

• připojení pomocí poštovního protokolu - POP, IMAP

- na MTA server se připojí klient toho protokolu a zadává uživatelskou polohu

⇒ bylo voleno, ale slouží pouze ke čtení doručené pošty

- ten program se srovná s připojení na nějaký další MTA server, kde fórové SMTP dopisy odesílá

Václav SMTP protokolem

→ MAIL FROM <fovor@uni.cz>

moy dopis

450/550

← 250 ...

obrálek

přijemci jednotlivě → server je potvrzuje (250), může odmítnout

→ RCPT TO <medved@uni.cz>

(pokud odpoví 250 → fiktivní odpovědnost za doručení dopisu

→ pokud se to neprodele, následuje Delivery Status Notification (DSN)

← 250 ...

→ DATA

→ rázřídk textu = 10, co příjemce vidí

→ fóvor na bázi jiného CR LF řádku!

← 554 Enter mail, end with ".."

↓ konec dopisu

→ From: <fovor@uni.cz>

→ To: <medved@uni.cz>

↓

↓ 250...

↓ QUIT

→ odesíatel může do řádku dopisu například jiného odesíatele nebo například do MAIL FROM!

→ Edyř MTA odposídlil 250 na nějželého adresáta a nevedlo se mu k doméně, kde posle DSN, kde nevyplňuje MAIL FROM ⇒ takové dopisy se často méně kontroloují ⇒ používají se spamovací enginy ⇒ některé správci posléz kalici přijem dopisů v různých formátech, což nemá dobré

• Struktura dopisu

- dopis se sčítává re ráhlem, které obsahuje hlavičky - jen ASCII 0-127
- řetězec je separátor (prázdná řádka) a následuje Text dopisu - původní byly
 - ⇒ dnes rozšířený protokol ESMTP, kde je ASCII 0-255
 - ⇒ je možné definovat strukturu a vnitřek těla dopisu pomocí MIME ⇒ souboje

• Soubory a diakritika v pošti

- původně byly povoleny pouze ASCII 0-127 → kodovaný soubor / řetězec s non-ASCII znaky

UUENCODE

→ nejdříve se výdaj ZB → rozdělí se na 4x6b. a tyto 6b. řady se převodem na 4 nelineární znaky pomocí speciální tabulky

⇒ 64 znaku: 26 velkých písmen, 10 číslic a 28 dalších znaku

→ nejfrekventnější velikost je 33% - je to tentýž protokol Čau → R&%V

→ kodovaný soubor se vkládá mezi řádky begin, end

⇒ problém: aby chom soubor nashli, musíme projít celý dopis

• Multipurpose Internet Mail Extension - MIME

- umožňuje strukturovat dokument (soubor) na hlavičku a tělo
 - ↳ hlavička mimo jiné obsahuje typ (Text, image, audio) → diakritický je multifakt
- multifakt dokument obsahuje více MIME dokumentů
- pro těden část umožňuje použití typu a formátu (Text/html), když jsou soubor a kodovaný
- mail s přílohami se posle jako multifakt, kde první část je Text a druhá část je filtry
- umožňuje používat diakritiku i v některých hlavičkách dopisů - např. předmět
- dnes používají i mimo formát

- Base64 - mychari z UUENCODE, jiná tabulka (a...z, A...Z, 0...9, +, /) a formát řádek → +33%
- Quoted-Printable - ascii znaky jsou uložení bez úměny ⇒ lepsi čitelnost Textu
→ non ASCII se vkládají jako "=HH", kde HH je jejich hex hodnota ⇒ je třeba kodovat i
⇒ pro non ASCII máme +200% ⇒ vzhledem k pouze ASCII souboru

nejdůležitější Hlavíčky dopisu

Date:	datum pořízení dopisu
From:	autor (autoři) dopisu
Sender:	odesilatel dopisu
Reply-To:	adresa pro odpověď
To:	adresát(i) dopisu
Cc:	(carbon copy) adresát(i) kopie „na vědomí:“
Bcc:	(blind cc) tajní adresáti kopie
Message-ID:	identifikace dopisu - sloučí k vybraním nášení
Subject:	předmět dopisu
Received:	záznam o přenosu dopisu

Bespečnost pošty - vězniak

- dopis se může dostat k hodně lidem, když nebyl určen + SMTP není sifrování řešení: sifrovat obsah dopisu → PGP (Pretty Good Privacy)
- někdy nemá jistý odesílatel → užaje a obává se že mohou být ručené částečné řešení: Sender Policy Framework
řešení: systém challenge-response, elektronický podpis

Bespečnost pošty - klient + server

→ svých

- poštovní server, aby měl posílat maily lokálních klientů / vězniaků domény a ostatní maily (příslušná rovná) pouze lokálním vězniakům
 - ⇒ ignoroval maily od cizích lidí pro určití lidi + nedovolil všechna posílat maily
- poštovní server dovolí komunikaci aby se připojil a poslal mail kamkoliv, takže je open relay
 - ⇒ brání riziku zneužití pro rozšíření hromadných mailů
 - ⇒ Existují organizace, které vyhrazují svému open-relay serveru - ignorace dopisů od nich
- když chec lokální vězniak poslal mail vzdáleně, tak ho server bere jako cizího
- ⇒ ESMTP umožňuje autentifikaci vězniaků, což SMTP nemá built-in
 - ↳ je to součást SASL profily pro SMTP → příkaz AUTH

- klient může pomocí ESMTP příkazu STARTTLS počítat s rohajím SSL/TLS spojením

Ochrana proti spamu

- Grey-listing - spam enginy obvykle nepoužívají porty o doméně
 - ⇒ poštovní server udržuje databázi triplets $\langle \text{IP}^{\text{mail}} | \text{mail} | \text{mail} \rangle$
 - ⇒ naopak odmítne poslat mail recipientovi odpovídící 450
 - ⇒ Client se po cca 15 min. pokusí mail poslat znova ⇒ tento určitý počet přijme 250
- Sender Policy Framework - doména publikuje pomocí DNS jeho poštovní servery
 - ⇒ poštovní odesílatel málo sdané domény z jiných serverů ⇒ ignorujeme hr
 - ⇒ problém: když má někdo nastavené forwardování pošty na jiné místo, tak to selže, protože odesílatel je stejný, ale nejdřív to posílá jiný stroj ⇒ dnes se nepraktikuje
- Domain Keys Identified Mail (DKIM) - podobná myšlenka - se svými poštovními servery
 - ⇒ odesílatel odesílá podepsání → forwardování funguje
- Antispam algoritmy - server na základě nastavené heuristiky rozhoduje jestliže mail je spam → diskutabilní věcnost a riziko false positive

- Post Office Protocol - POP → Textový protokol
 - posthorní protokol pro přístup uživatele k mailboxu - dále je IMAP
 - starý protokol → dnes podporován hlavně kvůli zpětné kompatibilitě → dnes
 - hlavní nevýhody
 - odvídání parolární hesla → Z rozšiřujícího příkazu pro řešení autentikaci → ne mohou použít TLS
 - dopisy je nutno stahovat ke serveru cíli
 - ↳ Aby bylo možné použít rozšíření pro starý jiný nastavení - tedy někde implementovány
- Internet Message Access Protocol - IMAP
 - modernější, ale složitější nástupce POP - má ho většina dnešních MUA
 - hlavní výhody
 - server uchovává info o dopisech (star)
 - podpora více schránek (složek)
 - protokol umožňuje využádat funkce částečného dopisu a vyhledávat v dopisech
 - built-in možnost používání TLS
 - například spojení na vyhrazený port - příkaz STARTTLS
 - tedy textový, ale uživatelé mají převést výhody
 - možnost zadat více příkazů následně
- Princip distribuované databáze
 - databáze informací uložených na obvyklem monolithickém serveru
 - jsou prováděny tak, že uživatel přechází pomocí odkazu ze serveru na server
 - ⇒ Gopher - 1. celosvětově rozšířena → aniz by se dalo
↳ něco jako web dneska → při přihlášení se zobrazí menu s odkazy, které vedou
lze na další menu / text / formulář
 - poskytoval jen textové informace → soubory si uživatel musel stahovat
- HyperText
 - Text obsahující varby určující pokračovat čtením prohlubnější informace
 - poskytuje myšlenku doplněnou textem o obrázků, svazcích, ...
 - realizace 1989 v CERNu → služba World Wide Web

- World Wide Web - WWW

- distribuovaná hyperTextová databáze
 - všechny jednotlivé informace je hyperTextová stránka (document)
 - ↳ server ji posílá na řádost klientům
 - dokumenty jsou psány v HTML - popisuje obsah i formu
 - ↳ konkrétní rozbalení je v rezisi klienta resp. webovém prohlížečku
 - dokumenty - statické - cesta v URL pot význam odpovídá skutečné cestě na disku serveru
dynamické - generují se dynamicky podle požadavků klienta
 - přenos stránek zajišťuje HTTP - chybějící zabezpečení \Rightarrow TLS \Rightarrow HTTPS

• HTTP v.1

- Hypertext protocol, převážuje verze 1.1, port 80
 - Klient může požádat na určité funkce prováděvat:
 - výrodní řádečka - metoda (GET), cesta, verze protokolu
 - klavíry - Host = jméno serveru, na který se klient obrací
 - fajgy, kódování, starší stránky
 - data pro aktualizaci, ...
 - file - náhledovní → např. Edýz klient uploaduje na server dokument

→ Server von odpsn

- výrodní řádky - verze protokolu, číslo odvěření (200), slovní popis (OK)
 - hlasovací řádky - formální název → čas poslední změny, ...
 - detaily protokolu - zprávy o přenosu, ...
 - vlastnosti posílaného dokumentu - např. jeho MIME hlasovací řádky
 - kdo - pořadový dokument / sešit chybou zprávy

→ kódy odpovědi - jazyk FTP

$\rightarrow 1,2,3$ výrodelecké stojné, 4_{xx} = chyba na straně klienta
 5_{xx} = chyba na straně serveru

→ melody HTTP

def: Metoda je bezpečná \Leftrightarrow nemá obsah dokumentu - nízky

- bezpečná = nemá obsah dokumentu - nikdy
- ideomofobická = opakování fráze má stejný efekt

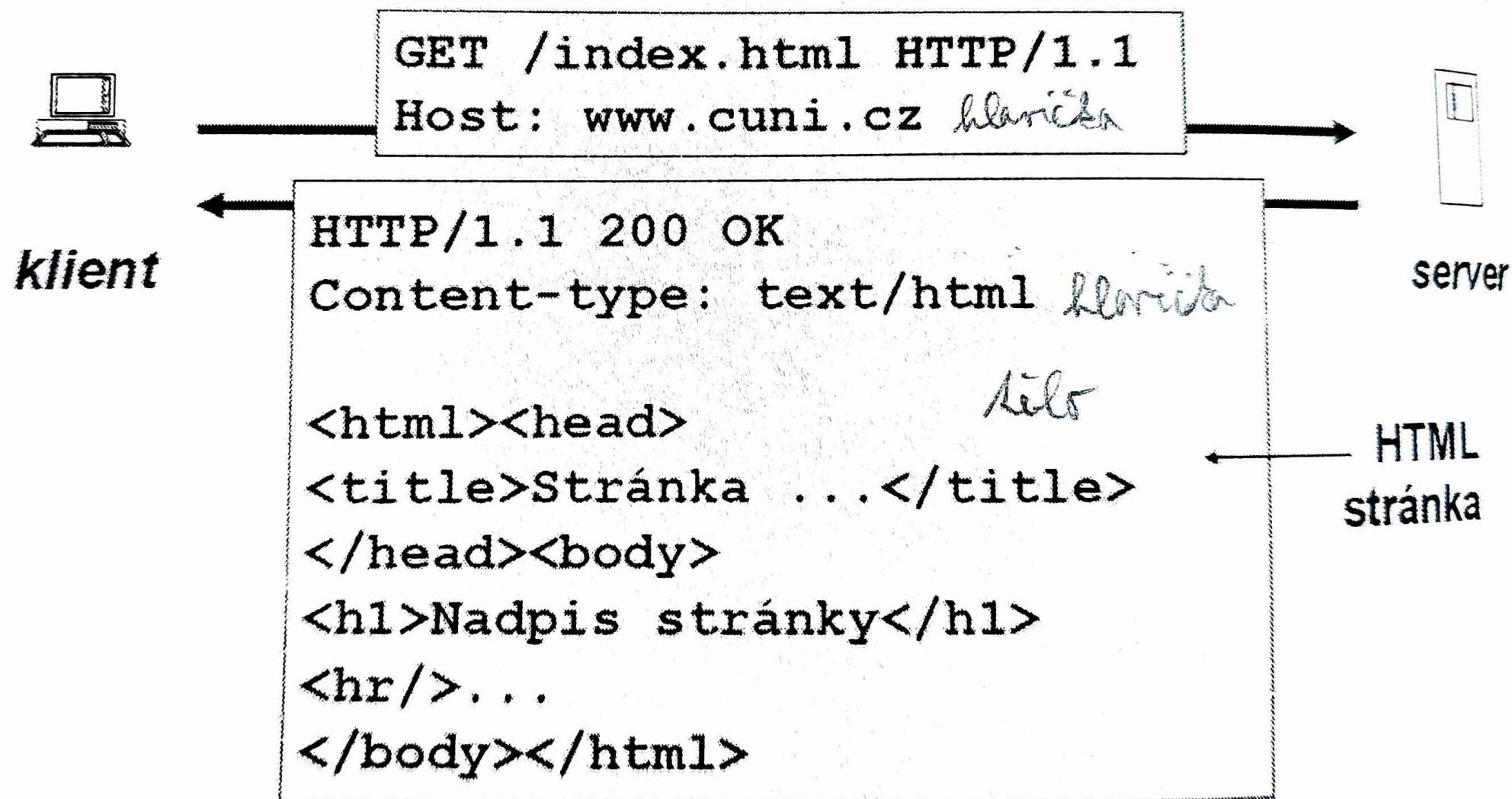
metoda	číslo požadavku	číslo odpovědi	bezpečnost	identifikace	
GET	—	document	✓	✓	← get doc.
HEAD	—	—	✓	✓	← get head
POST	parametry	document	✗	✗	← e.g. základny formulář
PUT	document	—	✗	✓	fórmuláře
DELETE	—	výsledek	✗	✓	← smazání
CONNECT					

vykročí ← tunel → když je možné se HTTP realizovat jiné spojení než jinou protokolu

→ bezpečnostní rizika
→ univerzální obecnost
omezení daná bezpečnostní
funkcionalitou sítě

Ukázka protokolu HTTP

URL: <http://www.cuni.cz/index.html>



• Vlastnosti HTTP v.1

- požadavky jsou merávány ⇒ pokud se WWW stránka skládá z sestra a 2 obrázků, pak lze být dva s merávanými požadavky
- v 1.1: persistentní spojení = po skončení požadavku nemusí klient zavírat TCP spojení
→ pro 1 spojení může jít pořadně několik požadavků
 - + klienti obvykle otevírají současně několik spojení
- celá komunikace je bezstanová ⇒ server nemá, které požadavky patří k sobě,...
⇒ pokud uživatel předá serveru nejprve info potřebné pro další práci (nastavení), tak by se server musel ohlásit u každého požadavku znova
- ⇒ cookies = data, které server vygeneruje na základě info od uživatele + posle je klientovi pomocí hlaviček Set-Cookie
 - prohlížeč si je uloží a při následujícím dalším požadavku je serveru poslána ve formě hlavičky Cookie ⇒ server rozpozná správné nastavení,...
 - cookies nepředstavují přímé nebezpečí, ale server je může využít pro abnormální info o uživateli + mohou být odcizeny - jsou uložena v PC

• HTTP v.2

- binární protokol - lze na něj přejít v rámci HTTP/1 spojení
- bloková modelovac → některá propustnost (rychlosť)
- metody - vlastní multiplexing → více streamů v rámci jednoho TCP spojení
 - ↳ streamy se neblokují + se dají prioritizovat
- server může poslat více dat, než klient požadal, pokud má vlastní, že je bude potřebovat → posle tyto bloky na stránce různou (push)
- kvůli autentifikaci může stránka obsah hlaviček + často mají podobný obsah
⇒ lze ji efektivně komprimovat

• HTML

- 2014 verze 5
- sestraj obsah stránek je doplněn knazkami
 - ↳ strukturní → rozlozec
 - ↳ semantické → adresa
 - ↳ formační → sestiné
- je předchůdcem XML = Extensible ML

- Telnet - Telecommunication Network
 - velmi starý protokol
 - první řešení vzdáleného přihlašování na jiný stroj
 - rozšířená emulace terminálů (Network Virtual Terminal)
 - ⇒ protokol přenáší příkazy a reakce tak, že uživatel má "fiktivní" prostředí jako na reálném systému.
 - ⇒ klient a server se musí domluvit kdo bude dležit práci
 - když uživatel stiskne klávesu, někdo musí odpovědět její zobrazením na obrazovce
 - ⇒ DO ECHO / DONT ECHO + WILL ECHO / WONT ECHO
 - vznikají problémy :: protokol neobsahuje příkaz, zda jde o parsel nebo odparsel
 - hlavní nevýhoda: sloučený přenos dat → jedná se o rozdíl mezi počtem dat → posíláme rozdíl, ale až moc počtu
 - dnes - obsas se používá, když nechceme odhalit heslo - segmenty LAN
 - ↓ hledání jiných protokolů, připojením na jejich server

• Secure Shell - SSH

Telnet

FTP

- bezpečná náhrada starých protokolů pro vzdálené přihlašování / přenos souborů
- klient ověřuje server + komunikace je šifrována
- aktuální verze 2, port 22
- SSH v2 matic umožňuje:
 - otevírat paralelně více zašifrovaných kanálů
 - tunelovat zašifrovaným kanálem jiný port - možnost obcházet firewally
 - přístupový file system tak, že se jeví jako lokální - SSHFS
- Windows → klienti: putty, winscp
- UNIX → příkazy: ssh, scp

šifruje uživatel
↓
současné NVT
a přenášení souboru

šifruje uživatel
↓

• Bespečnost SSH

1, klient ověřuje server - na ráckodě zpravidla uživatelské konto / certifikáty

2, server ověřuje uživatele - pomocí hesla / OTP nebo bez hesla?

- uživatel si může pro každé dvojice (klient, server) vygenerovat dvojici klíčů, které jsou většinou uloženy na serveru

⇒ server požádá zašifrovanou výzvu a klient odpovídá plain textem

- bezpečnostní riziko: když uživatel ráckodě přistupuje k účtu, tak se může přihlašovat na všechny stroje používající stejnou dvojici klíčů nebo pokud je možné se se strojem A přihlašovat na B a reciprocem B → A pomocí stejné dvojice → ŠPATNĚ

⇒ princip internodních červů - řešení: chránit každý klíč heslem

Voice over IP - VoIP

- obecně označení technologií pro přenos hlasu pomocí TCP/IP
- lze realizovat různými náročnostmi nekompatibilními sítěmi
 - standard H.323
 - standard SIP
 - rozšíření obecnějšího protololu - SKYPE + HTTP
- celá řada problémů
 - digitalizace hlasu, nalezení partnera, domluva vlastnosti zářivem'
 - propojení s běžnou telefonní sítí

H.323

→ telekomunikacím společnost

- komplexní řešení multimedialní komunikace od ITU (International Telegraph Union)
- binární protokoly - sítě se každým bitem - založeno na ASN.1
- celá řada dílčích protokolů - ne všechny jsou volně dostupné
 - H.225 / RAS (Reg./Adm./Status) pro vyhledávání partnera pomocí X.500
 - Q.931 řešení mezičinných spojení
 - H.245 řešení různého (dohodnut používaných vlastnostech zářivem')
 - RTP kanály (Realtime Transfer Protocol) → přenos multimedialních dat - audio / video
 - RTCP (RTP Control Protocol) → řídí RTP kanály
- dnes postupně nahrazováno SIP

Abstract Syntax Notation 1 - ASN.1

- metoda, jak definovat nejednotlivou strukturu / obsah jednotlivého dat pomocí formální definice → velmi strukturální nástroj
- problém je implementace v H.323 - přirodě se někdo zapojí jen tehdy když je nutná → autorizace budoucí mechanismus umožňující budoucí rozšíření
 - extrémně složitá implementace
 - ⇒ kupuj se knihovny, které z elektronického zápisu v ASN.1 vykrojuj kód, který realizuje zápis a čtení R.323
- používá ho i X.509 - Public Key Infrastructure (PKI)

Session Initiation Protocol - SIP

- načrada složitěho H.323 jednoduchím protokolem port TCP/UDP 5060
 - architektura protokola HTTP, informace se přenáší ve formě klávesic
 - nejvíce vlastní přenos dat → využívá RTP + RTCP
 - řeší jen myslidelný partnera a nastavení sponzí
 - dohoda o parametrech datových kanálů řeší Session Description Protocol (SDP)
 - ↳ řetězec protokol → řádky formátu keyword = value
 - přenášení pomocí SIP zpráv
 - koncový uzel se může registrovat u registrátora ⇒ lze se propojit na telefonní síť
 - posřední proxy servery - menadžují komunikaci přes hranice různých sítí
 - ↳ jako u SMTP se během přenosu vkládají do zprávy klávesy s cestou
 - ⇒ posílána může zprávu směrovat odpočátku ↳ Via, Record-Route
 - ukázka SIP hovoru
 - 1) volající vysíle INVITE s rolagenym URL a nabídkou datových kanálů jazér SDP zpráv
 - 2) příjem dorazí na nejbližší proxy ⇒ ráčne řešit nálezení cíle - podle své konfig. + URL
 - ↳ nejoblížší aktuální proxy, ...
 - 3) proběhne nejakej významnější - viz obrázek
 - 4) když to rolagený zveďne, tak posle 200 Ok + SDP zpráv s nabídkou datových kanálů
 - ↳ ráčením to bude požadat proxy a sen
 - ↳ bude připraven další → nečeká NAT
 - 5) volající to potvrdí ACK
 - ↳ proxy se roze píše
 - 6) od tohoto klávesiku obě strany využívají datové kanály ↳ RTP/RTCP

Příklad SIP session

volející



INVITE (+SDP)

100 Trying

180 Ringing

200 OK (+SDP)

ACK

BYE

200 OK



hívaj

INVITE (+SDP)

100 Trying

180 Ringing

200 OK (+SDP)

ACK

BYE

200 OK

volaný



zvídav

AT

RTP/RTCP

- Schílení systému souborů
 - vzdálene připojení cílového file systému transparentně do lokálního
- Network File System - NFS
 - původně z firmy Sun Microsystems, dnes otevřený → R.FC
 - primárně UDP, dnes i TCP
 - připojený disk je identifikován jako server: cesta
 - autentifikace pomocí protokolu Kerberos
 - má relacím (RPC - Remote Procedure Call) a prezentacím (XDR) vrstvy
- Server Message Block - SMB
 - původně vyrábala IBM, potom Microsoft ⇒ nemí open
 - reverse engineering ⇒ Existuje implementace Samba - umožňuje použití Win
 - identifikace disků: UNC //server/číslo
 - autentifikace pomocí obvyklého uživatelského jména a hesla - Win
- Network Time Protocol - NTP
 - synchronizace času mezi různými sítěmi UDP
 - ⇒ stejné timestampy souborní + posouzávání času vzdálostí na různých přístupech
 - Client kontaktuje NTP servory místní v konfiguraci → ty kontaktují NTP servory dál
 - zdroje / NTP servory mají klasifikaci:
 - první rázum = stratum 0 → atomové hodiny
 - server stratum N → řízený podle zdroje stratum N-1
 - problém: zdroj poslé čas, ale nemí přesný kvůli latenci sítě
 - ⇒ v odporech jsou timestampy srovnávány interval, kde když se zlepší čas
 - ⇒ když je zdroj víc, tak se hledá průnik těch intervalů pomocí Marschnera
- Bootstrap Protocol - BOOTP
 - starý protokol, sloužil k přiřazování IP adres bezdistribučně stanován → přesněji nazývaný IP → MAC
 - stanice pošle všem uzelům v lokální síti svůj MAC
 - ⇒ BOOTP server najde klienta v seznamu a poslé IP
 - pak je odděluje router, tak musí umět tak rychlost poslat dál komu BOOTP server
 - = BOOTP forwarding
 - požaduje se uživatel, že klientem by se mohly i další info:
 - adresy routerů, nameserverů, NTP serverů, mail forwarderů, ...
 - protokol se rozšířil až následný DHCP

Dynamic Host Configuration Protocol - DHCP

- vnitřek je BOOTP
- stejný formát správ \Rightarrow BOOTP Client může komunikovat s DHCP serverem
- stejná statická alokace adres i dynamická - MAC se mohou měnit + hodnoty klientů
- časově omezený pronájem adres - lease-time
- kooperace mezi servry v síti \Rightarrow mohou se sdílet několika nabídkami adres
- průběh DHCP
 - ↳ klient počle broadcast DHCP DISCOVER

\Rightarrow jednotlivé DHCP servry v síti mu ráčí poslat své nabídky DHCP OFFER

\rightarrow klient chce zde a sbírá odpovědi \Rightarrow počte z nich vybere tu nejlepší

2) klient počle DHCP REQUEST s IP, kterou si ^{↳ ideálně tu, ve které posledně, po dobu doby} _{sebral} vysílá \rightarrow počle doby pronájmu

\hookrightarrow počle broadcast, aby ostatní servry odhlásily svoje nabídky pro něj

\Rightarrow server k počtu DHCP ACK, že adresa je opravdu stále volná

3) od této okamžiny začíná doba pronájmu

\rightarrow po polovině této doby klient počle stejnou sítě vysílá DHCP REQUEST

a, dostane odpověď \Rightarrow startuje mu nový interval doby pronájmu

b, pokud odpověď nedostane, tak po $\frac{7}{8}$ doby pronájmu počle nový

DHCP REQUEST, tentokrát broadcastem

\Rightarrow pokud ani tentokrát adresu nedostane, tak po uplynutí doby pronájmu

Převodník v systému - OSI 6

- funkce v rámci modelu komunikací kodování

- datových typů, datových struktur, ...

\Rightarrow velmi složitě - dle až dle definice / hodně

- položky v reálnosti: ASN.1 - položky dobré, ale strojově složitá implementace

\Rightarrow TCP/IP vlastně potřebuje protokoly \rightarrow konverzi provádí aplikace

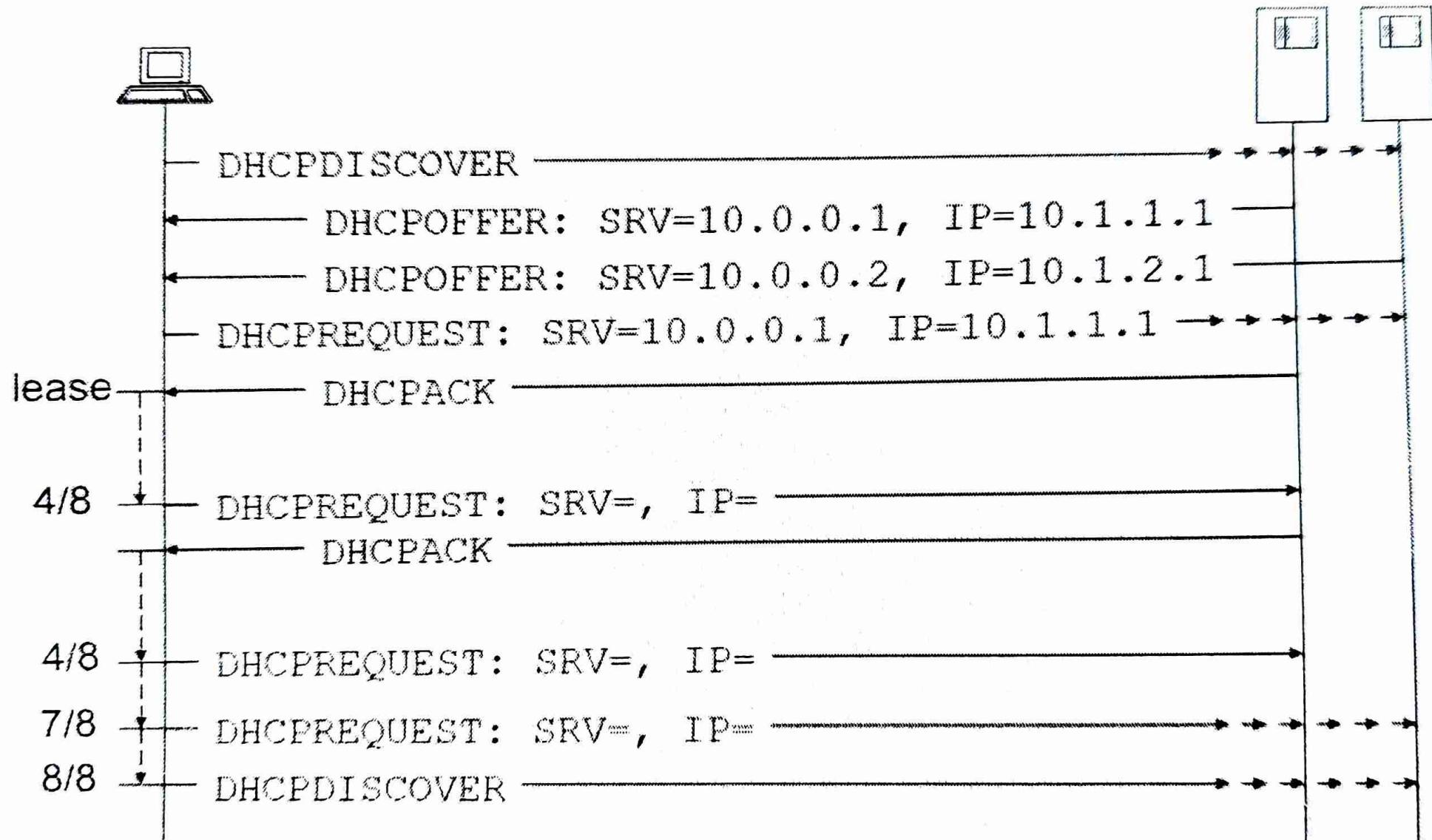
praktické problémy:

- konverze řádek: Win CR (0x0D) + LF (0x0A) vs. UNIX LF

- big endian vs. little endian

\Rightarrow TCP/IP používá big endian \Rightarrow MSB se posílá jdele první

\Rightarrow žádoucí je konverze



Relační vrstva - OSI 5

- představa o obecném modelu dialogu
 - 1 dialog může obsahovat několik spojení
 - pro 1 spojení může probíhat několik dialogů

⇒ TCP / IP rozděluje dialog do aplikacích protokolů

- SMTP - v rámci 1 spojení může být vyřízeno několik mailů (postupně)
- SIP - inicializuje dialog prvního příslušného spojení a 2 dalskými čárkami
↳ analog = 11. dia

Transformní vrstva - OSI 4

- zodpovídá za end-to-end přenos dat mezi koncovými aplikacemi
- zprostředkovává služby sítě aplikacím protokolům
- umožňuje paralelní více aplikací (clientů a serverů) na stejném vrstvě ← parity
- volitelně zabezpečuje spolehlivost přenosu dat } TCP
- volitelně segmentuje data a jezdíme je sládá } multiplexing
- volitelně řídí tok dat - flow control ~ rychlosť vysílání

TCP - Transmission Control Protocol

- pro spojení služby (telefonní hovor)
- Client naváže spojení → data se odesílají ve streamu
- spojení řídí a zabezpečuje TCP
- data ve spojení proudují oběma směry, protože protištítana, počítá se druhém segmentu
- menší pravidelné spolehlivé bezkontaktné spojení

UDP - User Datagram Protocol

- pro nespojené služby - správa
- neexistuje spojení, data se posílají jen meziadile správy
- UDP je jednoduché, relaci řídí aplikace
- pravidelný tok za cenu vysší chvatnosti

SCPT, DCCP, MPTCP - další modifikace či kombinace

Struktura UDP datagramu

- v UDP hlavičce se přenáší pouze informace o multiplexingu - SRC a DST port a řídicí informace - délka a kontrolní součet

↳ num. Src port copied to +1
↳ retrans. for bytes
↗ SEQ number

Struktura TCP packetu

- aby TCP mohlo garantovat kompletnost přenosu, každý segment musí mít ID = offset
- pro potvrzení packety: ACK number
- flags obsahují příkazy
- Urgent pointer je pro out-of-band přenos
 - ⇒ aplikace očekávající daty jde urgentní přenosem URG
 - ⇒ taková data se vrátí do normální komunikace a jejich koncová adr. v rámci datového bloku

↳ nr from urgent pointer

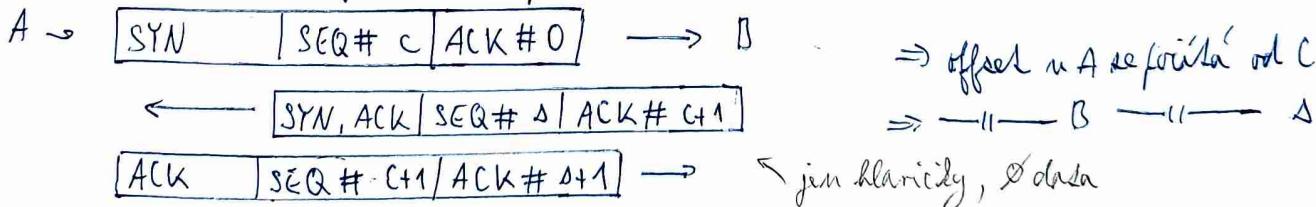
TCP window

- TCP posílá více bloků najeďdom → pravidlo 1 blok = 10 B & okno = 40 B
- ⇒ protiúhrada k poslání příkazem ACK a hodnotou ACKnumber nastavenou na offset koncu dat, která byla doručena
- může A poslat v rámci nějakého datového paketu - jinak by k tomu neexistovalo
- ⇒ když dorazí ACK, tak posílající posune okno
- ⇒ když se okno naplní, tak příští odesíláme a čeká na ACK
- když ACK nepřijde, tak znova pošle první nepotvrzený block dat

zavádění TCP spojení - three-way-handshake

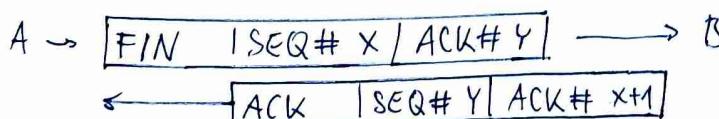
srdečí SEQ number si rozdělují
↗ 1. posílání

- sekvencí čísla ("offset") z korespondenčních dvojicí mezičísla od 0 ⇒ následné číslo

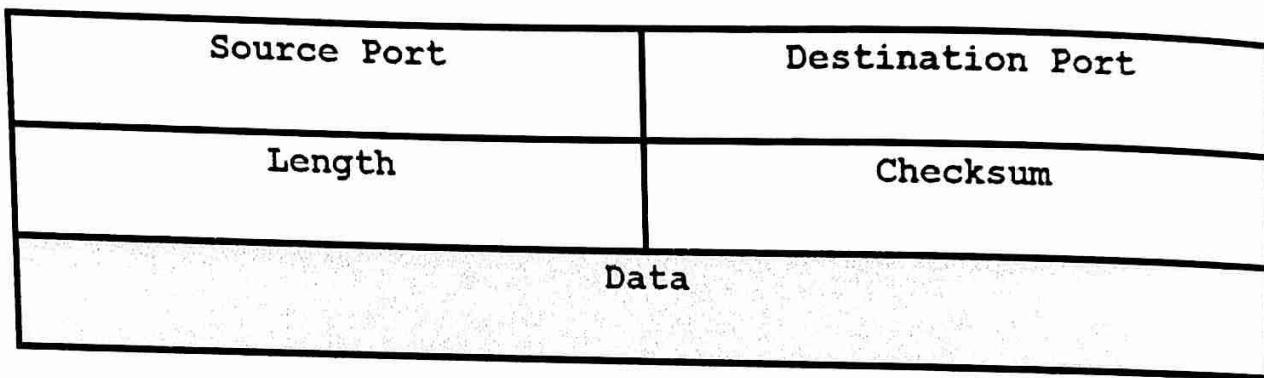
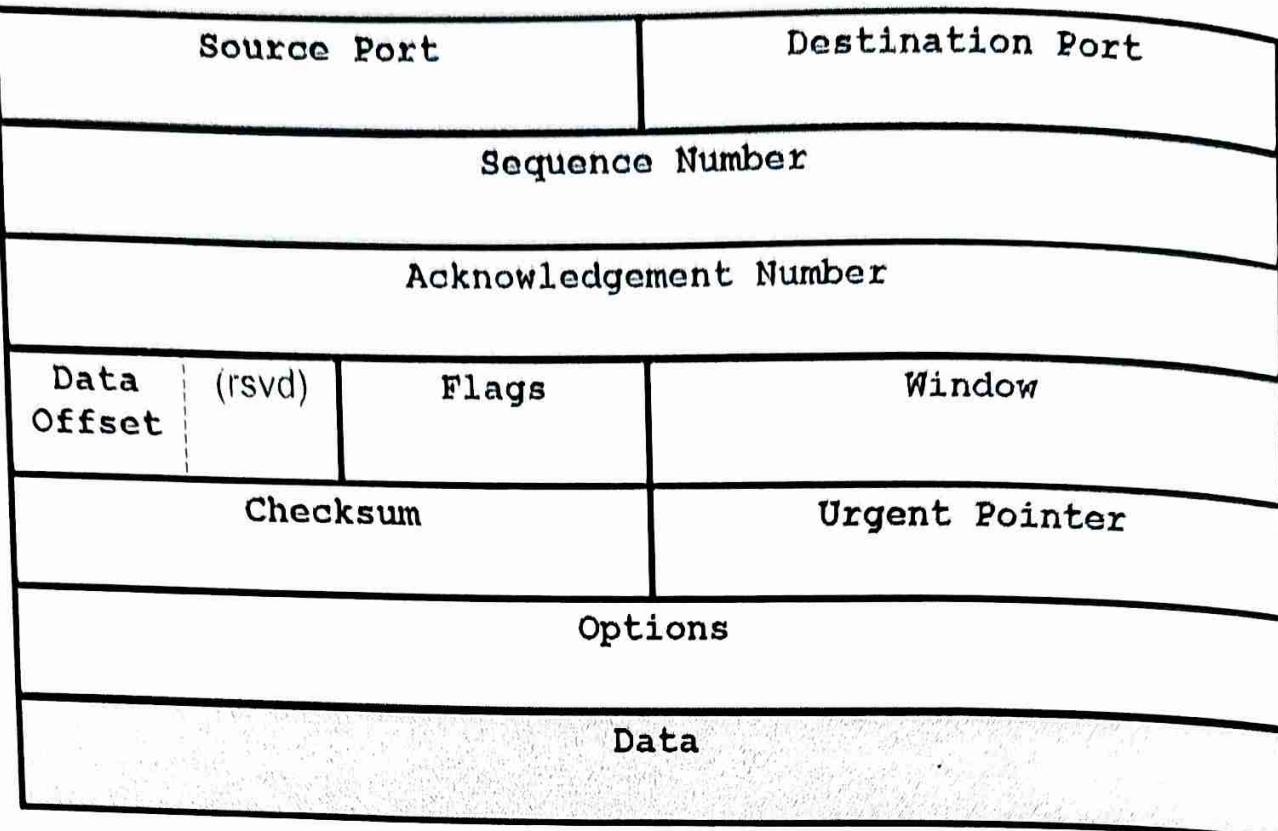


zavádění TCP spojení - jednostranné

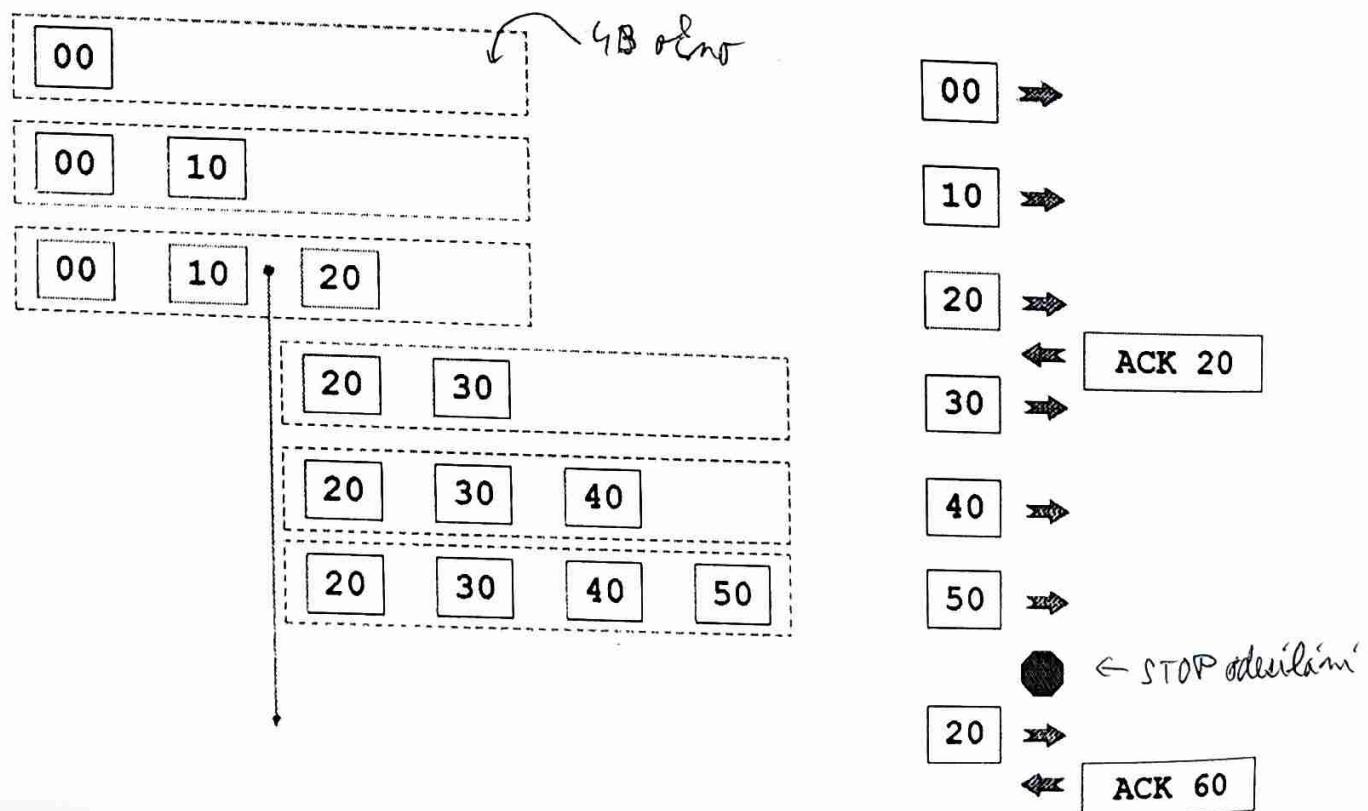
- A posle FIN packet ⇒ říká, že ně nebude posílat žádná data
- než B posle FIN, tak potřebuje znova posílat data ⇒ A k tomu bude posílat ACK packety



→ protiúhrada konec / posleji provede řádku



i segmentuje data a ještě může být soubor



- TCP přírny

- SYN - paket slouží k synchronizaci čísel segmentů - inicializace SEQ num
- ACK - paket potvrzuje doručení všech paketů až po ACK num (najma)
 - ↳ paket může i nemusí obsahovat data
- URG - paket obsahuje urgentní data, jejich házená adresa je v urgent pointeru
- FIN - odesílatek někdy nechce poslat řádná data
- RST - reset - odesílatek odmítá přijmat spojení \Rightarrow oznamuje ukončení sítě
- PSH - informuje příjemce, že obdržel kompletní blok a má ho předat aplikaci

- nápis existujících socketů

cmd: netstat -an \rightarrow vypíše seznam všech TCP i UDP serverů a otevřených TCP up.

Local, Local Address, Foreign Address, State

• Sítová vrstva - OSI 3

- hlavní funkce je přenos dat předních transportní vrstvou od zdroje k cíli

- rytína:

- adresace - protokol síťové vrstvy definuje formu a strukturu adres ve sítí
- segmentování - řídí data potřebná pro přenos se přidají k PDU
- routing - vyhledání nejvhodnější trasy k cíli přes možné sítě
- forwarding - router, který nemá konečný předávací pakety k zdroji dalšího
- dekapitulace - vybalení dat a přední transformní vrstvou

- příklady protokolů: IPv4, IPv6, IPX, AppleTalk

• Internet Protocol - IP

- vláznosti

- nestojaná služba - datagramy se doručují nerávně
- best effort - nespolehlivá, spolehlivost není vysoká vrstva
- nerávně na média - různé vrstvy nerávně mají média

- adresy - obsahují část s adresou sítě a část s adresou užív.

- IPv4 \rightarrow 32B, IPv6 \rightarrow 128B

- adresy přiděluje

- na vrcholu hierarchie stojí IANA (Internet Assigned Numbers Authority)
- 5 světových regionů \rightarrow regionální registrátoři \rightarrow Evropa je pod RIPE NCC
- dále ISP různých úrovní $\begin{cases} \text{první - druhá} \\ \text{region - ISP} \end{cases}$
- v LAN přiděluje IP adresy lokální správa sítě $\begin{cases} \text{Automaticky - DHCP} \end{cases}$

Struktura IPv4 datagramu

- délka hlavičky se udává v 32 bit slozech
- fragmentace - když síťová vrstva dostane packet dležejší než je její MTU (Maximum Transmission Unit) rafouzením by rozložil rámec delší než max. povolená délka pro danou linku vrstvy - MTU (Maximum Transmission Unit)
 - ⇒ pakom je třeba packet fragmentovat na více datagramů a poslat je postupně
- TCP se chec fragmentaci vyhnout ⇒ Push MTU - pakety se posílají s příznakem Do not fragment → odesílatele se dozvídá o problémech s velikostí MTU a může správně učinit velikost segmentu
- TTL, verze, číslo protokolu, kontrolní součet hlavičky, délka hlavičky
- IP adresa odesílatele a příjemce

IPv4 adresy

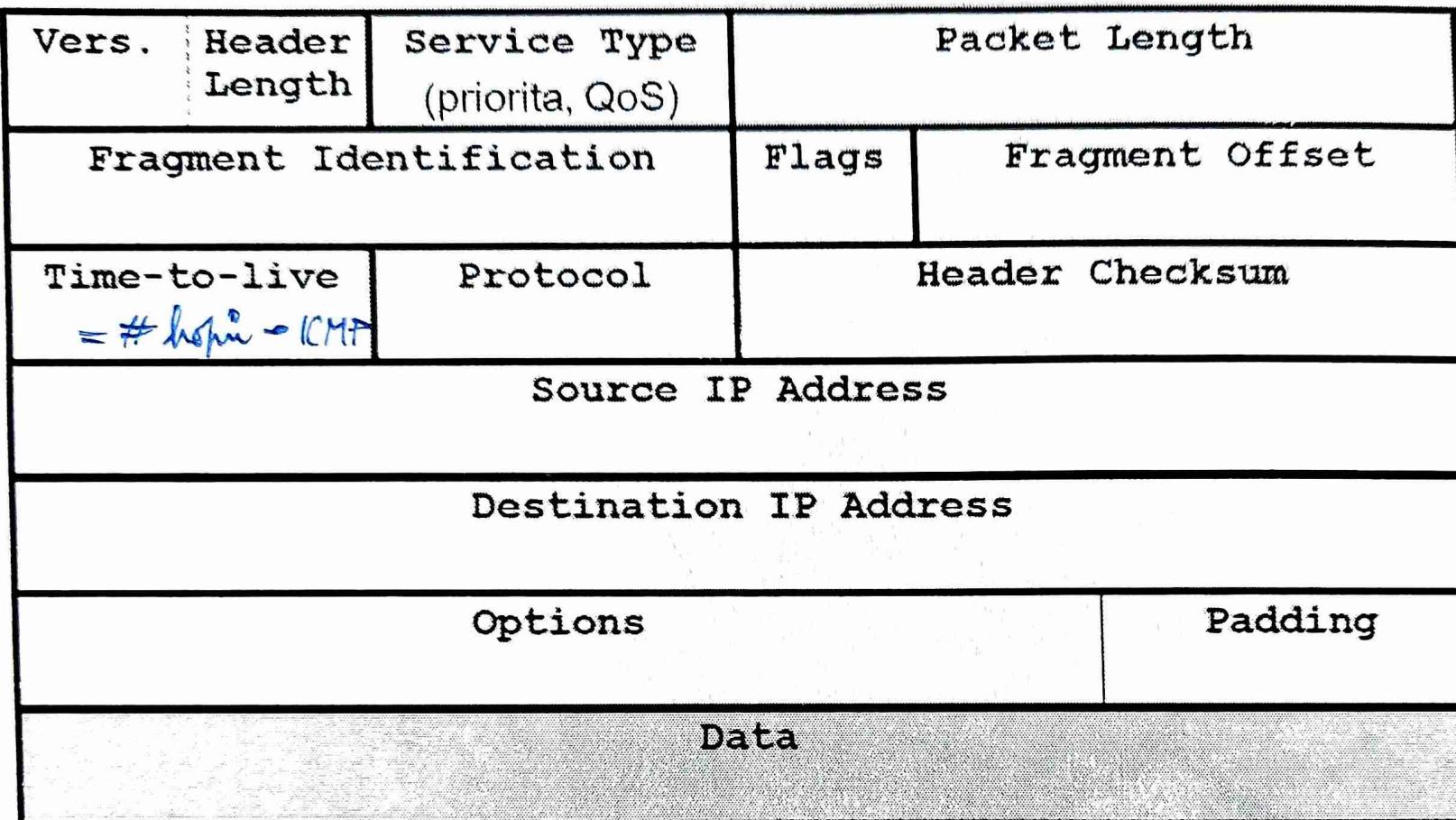
- původně 1B, pakom 3B, pakom 1B+3B, malmez výšky ABC → ABCD → ABCDE
- speciální adresy - by design
 - this host - adresou počítací s danou reprezentací adresou: 0.0.0.1/8
 - loopback - adresa "sobota počítací" pro užívání sítě: 127.0.0.1/8
 - adresa sítě - <adresa sítě><samej masky>
 - network broadcast - <adresa sítě><samej podmíky> ← všechny v dané síti
 - limited broadcast - 255.255.255.255 ← všechny v několika sítích
- speciální adresy - by definition
 - privátní adresy - pro provoz v lokální síti, přiděluje správce, není pouze NAT
 - ↳ 1A: 10.0.0.0/8 1B: 172.16-31.0.0/16 256C: 192.168.*.0/24
 - link-local adresy - pouze pro spojení v rámci segmentu sítě, uzel si ji volí samý
 - ↳ 169.254.1-254.0/16

Subnetting = rozdělení sítě na podsítě rozšířením síťové části adresy

C: 24/8 → 24/3/5:

net	net	net	sub net	host
-----	-----	-----	---------	------

- formou síťové masky: 1<=>sítě 0<=>host ↗ 255.255.255.224 → adresa & maska = adresa sítě
- nedoporučuje se používat subnet all-zeroes a all-ones ⇒ 6 × 30 adres (70%)
- část se ignoruje výšky (A, B, C) ⇒ classless mode
 - a mnohých se jen počítají blízké adresy sítě ⇒ např. 193.84.57.1 /24
- potřebuje se v síti používat různé masky → VLSM (Variable Length Subnet Mask)
- posun hranice sítě opačným směrem: supersubnetting



Třída	1.byte	2.byte	3.byte	4.byte	1. byte	Sítí	Adres
A	0	net	host		1-126	126	~16 M
B	10	net	host		128-191	~16 k	~64 k
C	110	net	host		192-223	~2 M	254
D	1110	net			224-239		multicast
E	1111				240-255		experimental

Třídy
IPv4 adres

• Kriče internetu

- preplňování routoracích tabulek

⇒ podstatný problém: velký počet neovlivně přidělených bloků rychle plní routorací tabulky

⇒ částečné řešení: realokace adres ⇒ CDR (Classless InterDomain Routing) agregace

- vyčerpávání adresního ř.

⇒ kvůli shrnutímu členění síti dochází k vyčerpání

⇒ částečné řešení: přidělování bloků adres bez ohledu na řídky

vracení nevyčerpaných bloků

privátní adresy + NAT ⇒ LAN → 1000 pravidelných + NAT = 1 miliarda

• IP verze 6

- dleší vývoj, z IPv4 adaptována řada dodatečných nástrojů

- přechod z IPv4 menadluje současnou IPv4 a IPv6

- koncová podoba adres: 128 b. (16B) → 8·2B

- zápis: FEC0::1:800:5A12:3456/64

- druhy adres:

• unicastové - adresy 1 sítě + rohové adresy:

• Loopback (::1/128)

• Link-Scope (FE80::/10) - dvíře link-local

• Unique-Local (FC00::/7) ~ privátní adresy v IPv4

• multicastová - adresa sloužící mnoha

• anycastová - de facto unicastová adresa přidělena více světě

⇒ více serverů po světě mají stejnou adresu a my chceme s nimi nejbližší
⇒ routování za nás vrátí do určitého města nejbližší

• chybějí broadcastové - posílají se multicastové

Směrování / Routing

- při směrování nejakehákho paketu chome výběr najít next-hop router

⇒ <u>směrovací tabulka</u>	směr do sítě	maska	router
- <u>default gateway</u>	destination	mask	gateway
- další ráčnany:	127.0.0.0	/8	127.0.0.1

→ next-hop router vybereme jalo den nejspeciálnější ráčnam

→ forwarduje masku ⇒ rozdává adresu sítě kam vede ⇒ formáme s koncovou adresou

• průmě ráčnany = ráčnany popisující první přijaté sítě / hosty

→ tyto ráčnany vznikají z tabule automaticky po konfiguraci sítového rozhramí

→ jaro gateway je uveden vlastní adresa, aby bylo jasné, že nemá kříba hledat next-hop router → taž adresa je v křídě sítě jiná

→ formální sítové rozhramí s loopback adresou - 1. ráčnam v případě *

• nepřímé ráčnany = ráčnany pro nepřímou propojení sítě / podsítě

→ gateway je centrální souběžně adresou next-hop routeru

adresa nějakého vlastního
sítového rozhramí

→ podle venku dělíme ráčnany na

• implicitní - venkovem automaticky na konfigurovaném sítovém rozhramí

• explicitní - dr tabulky se radí příkazem - ručně / ho zavádí OS při startu PC

• dynamické - ráčnam se vytvoří v průběhu práce pomocí info od dalších velkých sítí

→ směrovat by měla umět každá stanice (host?) v TCP/IP síti

→ maska určuje rozsah adresy sítě. ~ rozdělenou část adresy destination

~ směrovací algoritmus

najdi v tabulce všechny vyhovující ráčnany

↓
existuje → No route to host ⇒ paket se vrátí - maskou jediné složky nemusí být
defualt gateway

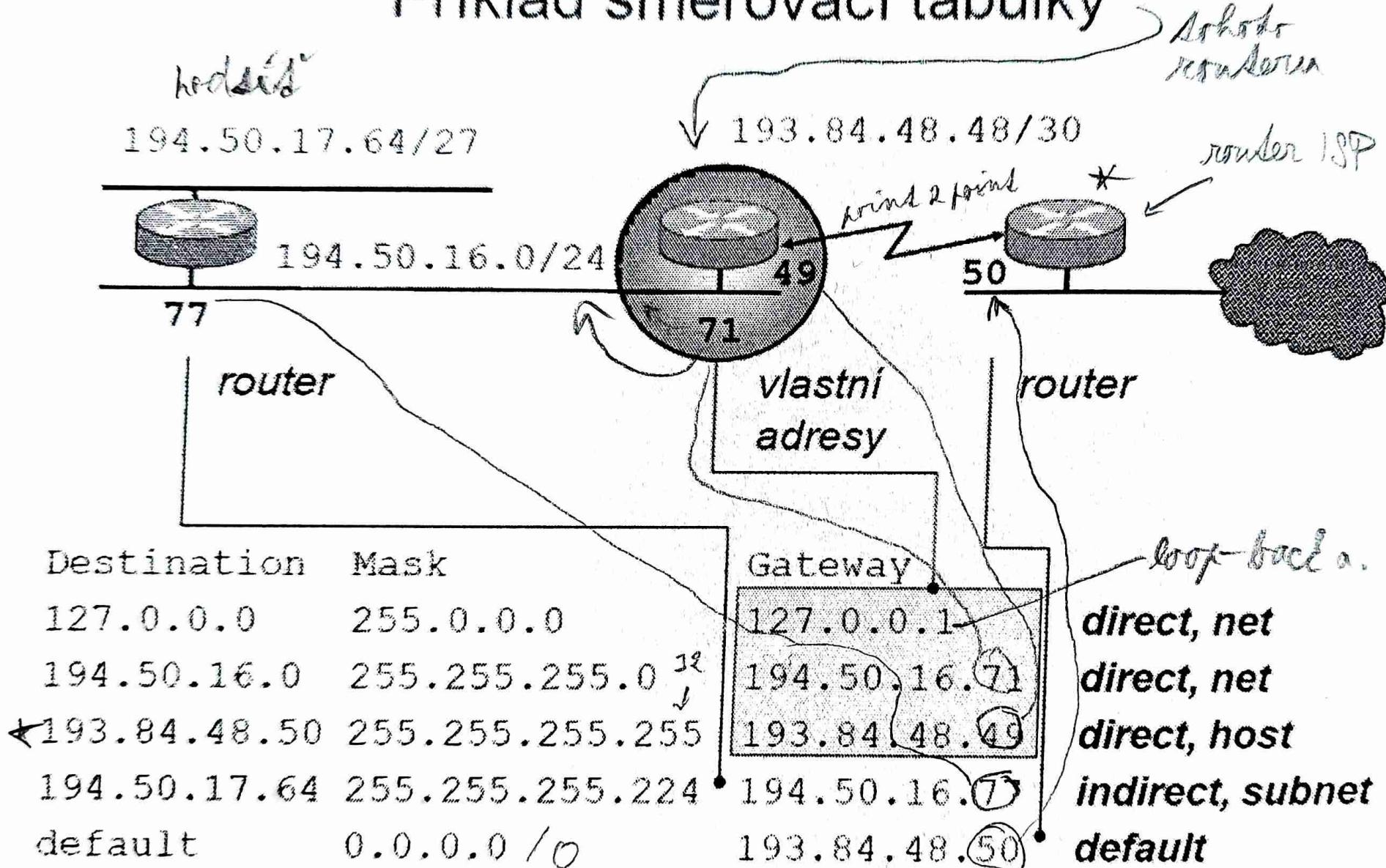
vrátí nejspeciálnější ráčnam (největší maska, nejdelší prefix)

↓
muj sítě → vrátíš na vrstvu - jalo by se právě přistál ve sítě

muj sítě → jde příjemci - průmě ráčnam

↓ ne → jde next-hop router - nepřímý ráčnam

Příklad směrovací tabulky



Konfigurace sítě

- UNIX - IP adresa → ifconfig interface IP-addr [netmask mask]
 - defaultní router → route add default gateway
 - DHCP → dhclient interface
- Windows - přes nastavení
 - v cmd je ipconfig, route

Internet Control Message Protocol - ICMP

- slouží pro posílání řídících informací pro IP síť:
- Echo, Echo reply - testování dosažitelnosti počítače
- Destination unreachable - nedostupný stroj / služba / síť, rozkáraná fragmentace
↳ problem při routování / problem s UDP
- Time exceeded - vypřel TTL \Rightarrow chyba v routování
- Source quench - zádost o snížení rychlosti přen os datagramu
- Router solicitation, Router advertisement - vyhledávání routerů
- Redirect - nýgra ke změně rácknamu v routovaci tabulce
- Parameter problem - chyba v rácknamu datagramu
- používá IP datagramy, ale nemá so transportní protokol \Rightarrow ~OSI 3.5 \Rightarrow stejně mimo
- ICMPv6 rozšířen → např. správy Neighbour Discovery Protocolu

Ping - ráckladní program pro diagnostiku sítě

- program s periodon 1s vysílá ICMP echo správy - dlead hr nepravidelné ...
- když správa dorazí na cílový stroj, ten odpoví ICMP echo reply
- pokud dorazí reply, tak ping vypočte řádky s časem cesty = round-trip time
- může vypočít round-trip min / avg / max / std-dev
- na alterném učku nemusí být k dispozici speciální program
- nezávislé dostupnost sloužeb - prvek sítě v reakci
- pokud odpovídá neplatné, tak nezávislé dostupnost sloužeb - neplatný router může odstranit ping paketu

ICMP Time Exceeded

- povídá pole TTL v IP rážkovi, aby nedocházelo k racykem paketu mezi routery
- TTL udává # hopů, kterých se paket ještě může vzdálit
- ⇒ při každém hopu router sníží TTL ⇒ musí upravit header IP paketu
 - if TTL == 0: nahradí paket a posílí odesílatele ICMP Time Exceeded zprávu
 - else: posílí paket next-hop routeru
- dnes je TTL defauktně 64

Diagnostika směrování

například obraz

- výpis routovací tabulky: netstat -r [n] = route print
- ping - něčím neponává - možná máz
- traceroute - 3x vysíle paket s TTL = 1 ⇒ zjistí 1. router
3x vysíle paket s TTL = 2 ⇒ zjistí 2. router
⋮

⇒ zjistí kdy se přestane vracet Time Exceeded ⇒ přiblíženě na jakém routeru je problém

Statické řízení routovacích tabulek

- počítá možné mítelé vložení všech potřebné ráženiny a pro každou si je přidá do tabulky
 - např. když jsi DHCP dostane adresu defaultního routeru pro danou síť
 - ① nepružné při změnách sítě, problém se subnetingem
 - ② méně citlivé na problémy v síti, dostupné v libovolné síti
- ⇒ vhodné pro jednoduší, stabilní sítě

ICMP Redirect

- umožňuje staticky řízeným routovacím tabulkám pořídit lepšíjší síť

1) chceme poslat paket do sítě 6.0.0.0

→ default gateway 5.0.0.2

2) paket dojde na router 5.0.0.8 a ten nádi, že ho má poslat router 5.0.0.6, které do stejné sítě, se které posílá!

⇒ posle mn hr, ale...

3) posle odesílatele (nám) ICMP redirect zprávu, abychom si do tabulky přidali novou ráženinu pro síť 6.0.0.0 ⇒ novou sítě je přiřazeno D

původní obsah tabulky:

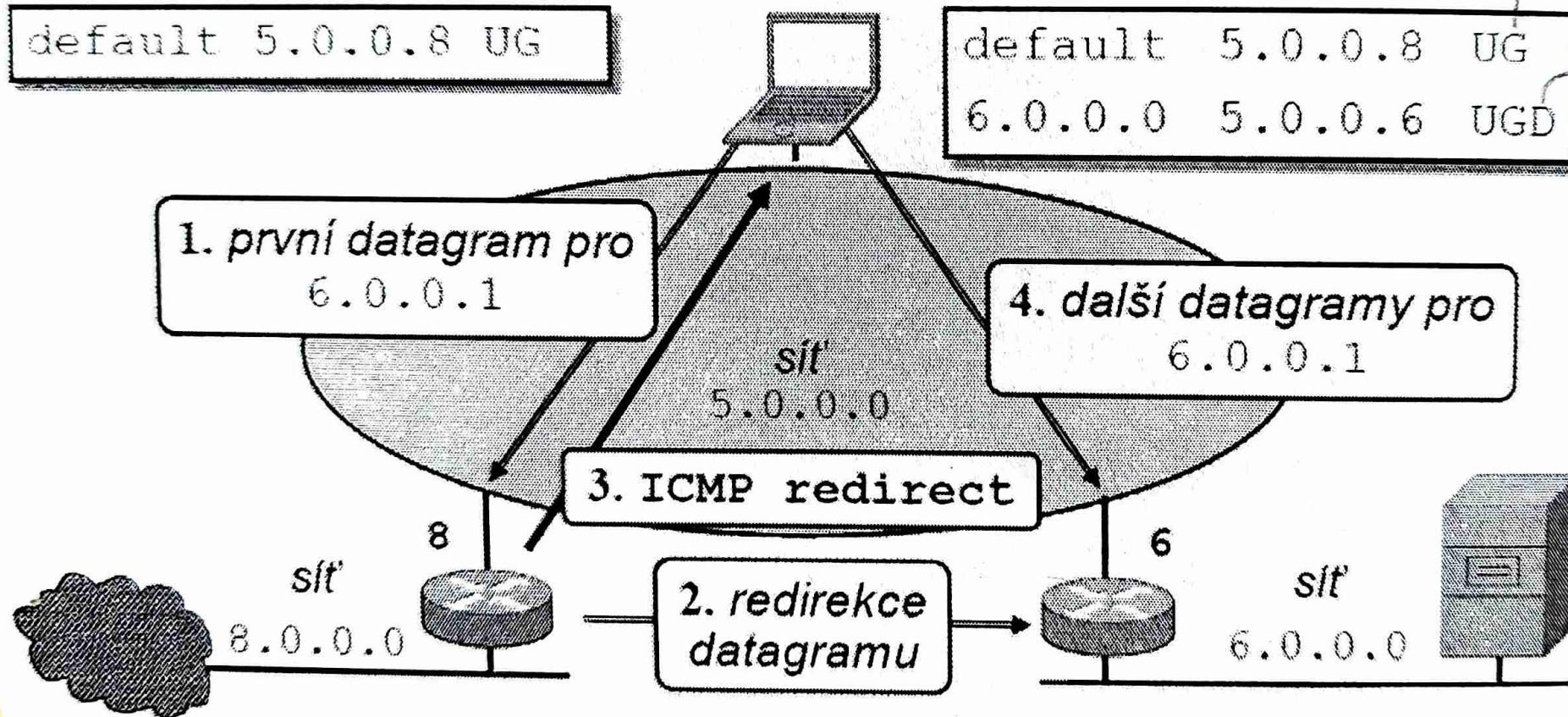
default 5.0.0.8 UG

nový obsah tabulky:

default 5.0.0.8 UG
6.0.0.0 5.0.0.6 UGD

Gathering

Dynamic



Dynamické řízení směrovacích tabulek

- sousední routery si vyměňují info o síti prostřednictvím routeing protocolů
 - stanice se jím mohou řídit sítě, ale v režimu read-only

- (+) jednoduché změny konfigurace, síť se sama opravuje, routeing tabulky se udržují *auto.*
- (-) citlivost na problémy, vlohy

- na počítání musí být program obsluhující protokol → BIRD - MFF
- routeing protokoly pro lokální síť se dělí na:
 - Distance-vector protokoly - RIP
 - Link-state protokoly - OSPF

↳ interní routeing protokoly

Distance vector protokoly

- router má v roznamí i „vzdálenost“
- směr tabulek periodicky posílá sousedům, kteří si upraví svoji tabulku

- (+) jednoduché, snadno implementovatelné

- (-) pomalá reakce na chyby + chyba reaguje na vývoj celého sítě, omezený rozsah sítě, metrika nezohledňuje vlastnosti linek (rychlosť, spoľahlivosť, ...)

Routing Information Protocol - RIP

- nejstarší směrovací protokol

A → C přímo přijíma routery mají metriku 1

vlastnosti

- metrikou je # routerů v cestě = hop count

$A \rightarrow D \rightarrow C \Rightarrow \text{cesta} \geq \text{hop} \geq 2$

- rozsah sítě je omezen na 15 hopů, 16 = inf

- pro výpočet nejkratších cest používá Bellman-Fordův algoritmus

Alternativní verze 2 - OSPF

- umí subnetting včetně VLSM

- obsahuje mechanismus na výchlovou detekci chyb

- starý → dostupný skoro všude

- nepoužitelný pro velké, složité a ne dynamické sítě

metrika a čísla linek

- hop count nereflektuje vlastnosti linek → metrika některých formulejících linek umírá svítání

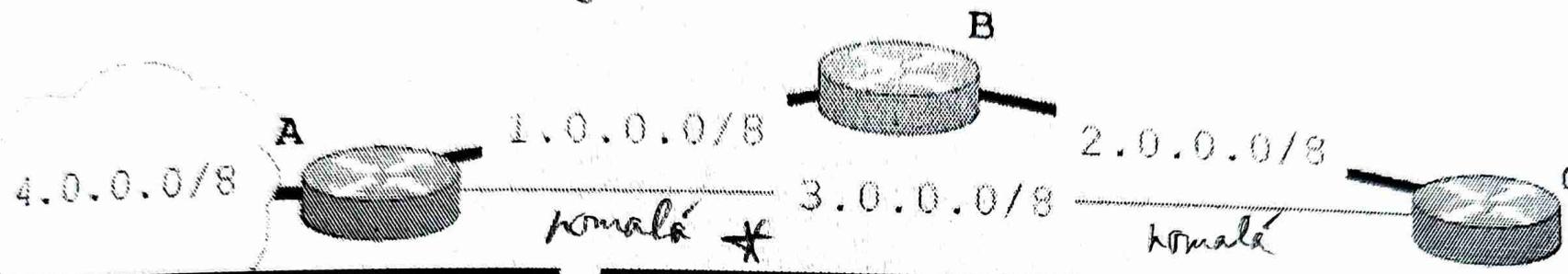
- na začátku se všechny routery inicializují hodnotami pro první přijaté sítě

- každý router posílá update a ostatní routery počítají metriky pro nové sítě, takže

$$n(A, C) = n(A, B) + n(B, C)$$

→ přičtení 2 domů vzdálenost sítě, reagovat na update

* metoda linky merí AC vzdálość + 2 $\Rightarrow r(A,C) = 3$



1.../8	-	1
3.../8	-	3
4.../8	-	1

1.../8	-	1
2.../8	-	1

2.../8	-	1
3.../8	-	3

A rozesílá update:

1.../8	-	1
2.../8	-	1
3.../8	A	3+1
4.../8	A	1+1

1.../8	A	1+3
2.../8	-	1
3.../8	-	3
4.../8	A	1+3

B rozesílá update:

$r(A_B)$

1.../8	B	1+1
2.../8	-	1
3.../8	-	3
4.../8	B	2+1

\rightarrow počká bude říšské C posílat update

C píše jazyk router

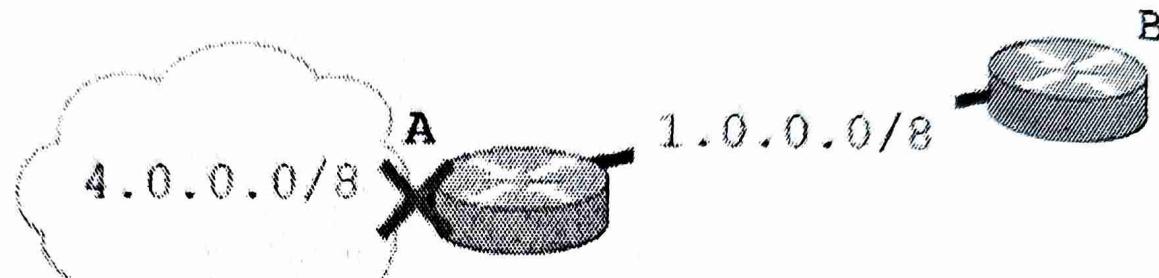
- counting to infinity - důvod, proč je měření řek malé
 - když v síti objde členitá mřížka přijedou informace do nějaké jiné sítě, tak na to ten router A reaguje měřením metriky dané sítě na inf (16)
 - ⇒ neříkáme ale výpršení periody kdy bude moct rozslat svou tabulkou, tak jiný router B rozsle update se starou vzdáleností ⇒ A si upraví tabulkou ⇒ 16 nezahrani
 - ⇒ po výpršení periody (30s) A rozsílá update ⇒ B má ten ráznam pro A, záleží si kde zvítězí ⇒ B rozsílá update ⇒ A + = 1, ...
 - ⇒ po nějaké době dojde i $A=B=16$
 - ⇒ kdyby inf bylo něčeho, teď by starost mřížky byla něčeho
 - race condition = situace, když 2 mřížky jenž mohou nastat v mřížkovém prostoru
 - țízení nebo RIP v2
 - Triggered updates - router při detektaci problému rozsílá update všechny
 - ⇒ riziko race condition se sníží, ale neodebrání
 - Horizon split - router neposílá sousedovi info o sítích, o kterých se dozvídá od něj
 - ⇒ problém je zobrazení, ale pro každého souseda musí rámci připravit záložku
 - Poison reverse - router sousedovi info o „jeho“ sítích posílá, ale s metrikou 16

Link state protokoly

- každý router má „mapu“ cílové sítě
- routery si navzájem sdělují pouze star svých linek ⇒ každý si sám modifikuje svou mapu
- (-) výpočet mapy je náročnější na výkon CPU i paměť
 - při startu a na nestabilních sítích může výpočet vytáhnout celou sítě
- (+) první reakce na změny topologie, výpočet dat pouze při změnách sítě
 - každý si mapu počítá sám, chybou mřížkového oslabení
 - sítě je možné rozdělit na podsítě ⇒ ↑ rychlosť výpočtu

Open Shortest Path First - OSPF

- k mřížkové neplánované cestě používá Dijkstrův algoritmus
- fwčírová hierarchický model sítě
 - oblast 0 = fáterové sítě
 - vlastní oblasti se přiřazují pouze na fáter ⇒ k routeru značí mapu své oblasti a vzdálu fáteru
 - metrika je možné konfigurovat → implicitně to je path cost = \sum cen na cestě, cena = $f(\text{bandwidth})$



1.. /8	-	1
2.. /8	B	2
3.. /8	-	3
4.. /8	-	1

1.. /8	-	1
2.. /8	-	1
3.. /8	A	4
4.. /8	A	2

Výpadek linky A/4:

4.. /8	-	16
--------	---	----

4.. /8	B	2+1
--------	---	-----

← přes 3 dan. vede křížící cesta

4.. /8	A	3+1
--------	---	-----

...

Stav po 7x30sec:

4.. /8	-	16
--------	---	----

4.. /8	-	16
--------	---	----

Autonomní Systémy

- Definice: blok sítí se společnou routovací politikou. \Rightarrow nemá zase tak lečit o příslušenství AS
- \rightarrow v roce 1982 pro enesí routování na globální úrovni
 - \rightarrow identifikátor AS je jeho číslo, dřív 16 bit, dnes 32 bit.
 - \rightarrow v ČR: na příčkách 2, dnes skorší
 - \Rightarrow routování mezi jednotlivými AS využívá Externí routovací protokoly - EGP - BGP
 - \Rightarrow v AS Interní routovací protokoly - IGP - RIP, OSPF
 - \rightarrow dnes je nejrozšířenějším EGP Border Gateway Protocol - BGP

IP filtrace

- router na perimetru má v konfiguraci uvedeno, jaký provoz je povolen a za jakých podmínek
- prázdná konfigurace: není vybrané, domluvíme
 - OK pro protokoly s 1 datagramem kanálem - HTTP, SMTP
 - problém u protokolu s více kanálů - FTP, SIP
- standardně: ven voleli, domluvíme / spojení může mít server
 - mazání např. u FTP \rightarrow aktuálním přenosem
 - nepovolené u protokolu s mnoha kanálů - SIP
- řešení: SW na routeru musí částečně rozumět protokolu na aplikativní vrstvě
- problém se sháněním určité sítě, ke které by měli mít přístup všechni z internetu
 - \hookrightarrow mapí www server, portka
 - \rightarrow problemem výjimek je násobkovitost
- \Rightarrow lepsi je rozdělit oddělený segment sítě, do kterého je přístup povolen pouze k němu

\Rightarrow DMZ = Demilitarizovaná zóna

Proxy servry

- / pracuje na hranici mezi dvěma sítěmi
- Transparentní - SW na routeru zachytí požadavek klienta, mazáče svým jménem správce serveru a požadavek odesle
 - odpořec přijde zpět na router, ten ji uloží do cache a paké přesoudu může zadateli
 - Netransparentní - klienty je třeba nafigurovat, aby požadavky posílali přímo, ale proxy server
 - \oplus proxy nemusí být mezi sítěmi
 - \ominus je nutná podpora v daném protokolu
 - jsou nejčastěji používány bezpečnostním a nějčetnějším prověrem sítě
 - umožňuje správce sítě kontrolovat činnost klientů, filtrovat šířící obsah zpráv
 - může mít i cache pro zlepšení dostání klientům

• Address Resolution Protocol - ARP

- umožňuje překladať mezi linkovými a sítovými adresami ethernet MAC \leftrightarrow IP
- neznámé adresy se registrují broadcastem výrovnou s všemi MAC adresami ff:ff:ff:ff:ff:ff
- hledaný uzel (když se chybí jeho ARP server) odpovídá výrovnou ARP odpověď s představenou MAC a průduší info (IP a MAC) a řadí do svého ARP cache
- řadatel (ARP Client) si odpočítá vloží do ARP cache

! Nelze mít více ARP odpovědí ke stejnému // ARP odpověď ke žádosti

! Gratifikace ARP - nevyžádané ARP - rychlejší změna dynamické sítě, riziko

- výpis ARP table: arp -a

• Proxy ARP



Achce něco poslat B

1, host A posílá broadcastem ARP request s IP adresou B

2) router počítá, že došlo k B místy nedostání, takže samy posle ARP reply se svou vlastní MAC adresou

3, host A si k IP adrese B nezapisuje ARP cache přiřadí MAC routeru

4, host A posílá data pro B s MAC adresou routeru

• Linkové vrstva - OSI 2

- dělí se na 2 podvrstvy

• Logical Link Control (LLC) - umožňuje různým protokolům sítivé vrstvy přístup k stejném médium \Rightarrow multiplexing

• Media Access Control (MAC) - řídí adresací adresu a přístup k médiu
 \Rightarrow kdo, kdy a jak může data odesílat a jak je přijímat

- TCP/IP už se kontrolem nezabývá - součást sítivého rozhraní

Definice: Sítový segment = množina všech sousedících stejných médií.

- PDV na linkové vrstvě = rámec / frame

- hší se podle použitěho média

- obecně obsahuje: Synchronizační pole (\sim start condition?),

• hlavička - MAC adresy, řídící info LLC, datové pole

• pohlídky - Frame Check Sequence (FCS) - detekce chyb

Typy sítových topologií

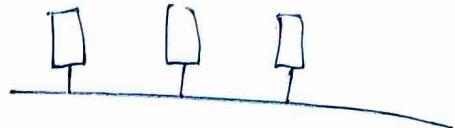
- topologie = uspořádání uzelů

Multipoint

Starnice (Ethernet)

- když se zábel převíží, všechno se rozbírá

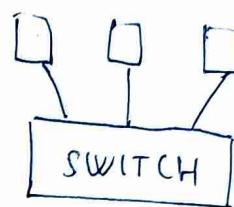
- Záblesk = když 2 počítače chcejí vysílat různou



Hvězda (ATM)

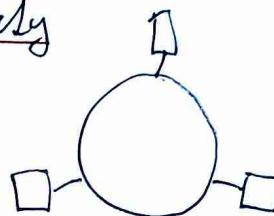
- centrální prvek je dobře chráněný ^{homonymum!}

- je to switch - jednotlivé rášavky = forty



Kruh (FDDI, Token-ring)

- vše je propojeno do kruhu



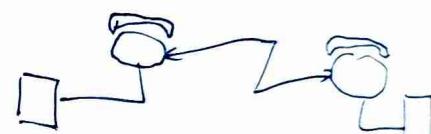
Point-to-point

Přímé spojení kabelem (RS-232)



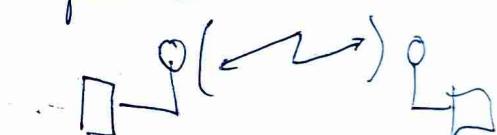
Propojení přes modemy

↳ modem moduluje spojení aby bylo přenosné po telefonní síti



Beskálové propojení - laser, radiové vlny

↳ WiFi má hvězdicovou topologii



Způsob řízení přístupu k médiu

Multipoint

Deterministicky - vždy je určeno, kdo má vysílat \Rightarrow rečie pouze určitým uzelům

- Token-ring - řídící prvek = speciální poket (token) - během řízení

- vysílající odchylí token \Rightarrow přijímající ho uzele vysíle

- nebo je speciální řídicí uzel, který ostatním uzelům posílá signály, kdežto mohou vysílat

Nedeterministicky - Ethernet, musí se nejprve řešit kola

Point-to-point - halfduplex - řízeno kolou

fullduplex - např. pro ethernet se oboustranné kolou \Rightarrow \uparrow provoznost

• Řešení kolizií

• CSMA (Carrier Sense with Multiple Access)

- užel poslouchá „nosnou“ a pokud není volno, čeká

• CSMA/CD (Collision Detection) - Ethernet

- během vysílání užel taky poslouchá nosnou → detektuje případnou kolizi

- při kolizi (tb) stanice rastají vysílání, upozorní ostatní, počkají retransmisi (náhodnou!) dobu a poté opakují

→ pokud před dojde ke kolizi, tak se exponenciálně zvětšuje interval čekání

- podmínka: doba vysílání rámců > doba šíření po segmentu (= kolisní okno)

⇒ určuje max. délku segmentu sítě a min. velikost rámců

• CSMA/CA (Collision Avoidance) - WiFi

- WiFi používá hvezdovou topologii - centrální pylon = Access point
⇒ každý uzel je vlastně point-to-point

- když je nosná volná, vysílá se celý rámec a čeká se na ACK
↳ pokud ACK nedosáhne, rabiájí se exponenciálně čekání

• Ethernet

- historie - v Xerchu → standardizaci provádělo IEC ⇒ 2 formáty IEEE 802.3

- místní technologie pro lokální sítě

- dleší pravé reagovat na výroj HW

- šíření přístupu metoda CSMA/CD

- při kolizi užel vysílá „jam signál“

- exponenciálně čekání končí po 16 pokusech chybou

- MAC adresy - 3 byty prefix výrobce, 3 byty adresa

- dříve využívala na kartě, dnes nastavítelem

- strukturna ethernetového rámců

- původní koncept - 2B typ sítěho protokolu

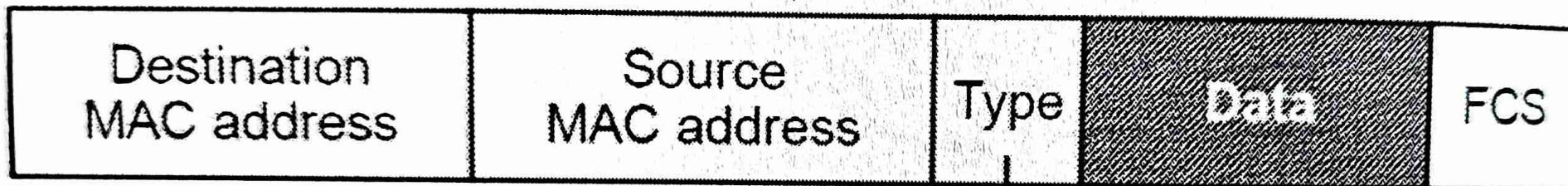
- IEEE - délka + speciální LLC rácklan'

Ethernet II
IEEE 802.3
nejnovější! musí se shodit
normy IEEE

→ 1 segmentu sítě neexistuje
→ byl 2x stejná MAC

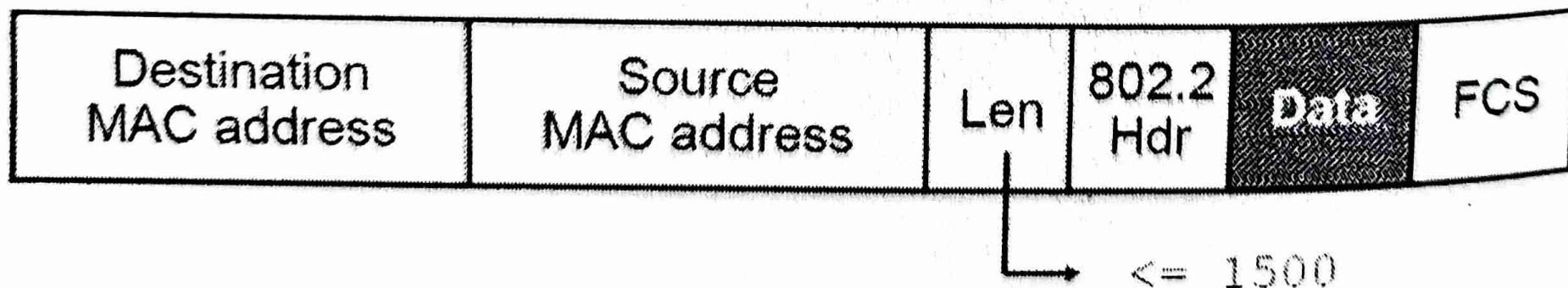
Struktura ethernetového rámce

Ethernet v2:



IP 0x0800
ARP 0x0806
RARP 0x8035
IPX 0x8137

IEEE 802.3



<= 1500

• Virtuální sítě - VLAN

Virtuální sítě oddělují segmenty a uživatelé
dle VLAN ID

- prostředek jak pro 1 fyzickou síť provozovat více logických sítí
- virtuální síť jeva označený 12 b. identifikátorem → VLANID
- Ethernetový rámec se prodlouží o 32 b. dlonky tag obsahující (protokol ID, QoS priority a VLANID) = VLAN Tag - vloží se před Type
- virtuální síť lze z pohledu koncových stanice provozovat transparentně
 - ↳ switch ří, že na nejdeš portu má VLAN s určitým VLANID
- když přijde rámec s tím VLANID, tak tag odstraní a zbytek posle do té VLAN
- když přijde rámec z té VLAN, tak do méj přidá tag a posle ho přej
- Když nějaký uzel potřebuje mít přístup k rámcům ze všech sítí, tak se jeho port konfiguruje jako trunk a switch s rámci mezi sedláčkami ⇒ obsluha tagů nechá na uzel
- rámců se přidáním tagu prodlouží o 4B ⇒ musíme byt schopni pracovat s rámcem delšími než max. retransmit max na max - 4B

• Cyklický kontrolní součet - CRC = Cyclic Redundancy Check

- hashovací funkce používaná pro kontrolu konsistence dat - FCS, kontrolní součet IP header
- posloupnost bitů ⇒ polynom $10110 \sim x^4 + x^2 + x$
- vydělí se charakteristickým polynomem sítových stupňů, kolik b. má kontrolní pole
 - ↳ pro CRC-16 když $x^{16} + x^{15} + x^2 + 1$
- zbytek po dělení se píše do bitů a poskytuje hash
- jednoduchá HW implementace
- velká síla - n-bit CRC dešifruje na 100% chyby s hligem # bitů, aby byly kontrolní bity

• WiFi = WLAN (Wireless LAN)

- mnoho různých variant pro IEEE 802.11
 - ↳ různá frekvence 2.4 až 5 GHz
 - ↳ různé rychlosti 2 až 600 Mbps
- struktura sítě

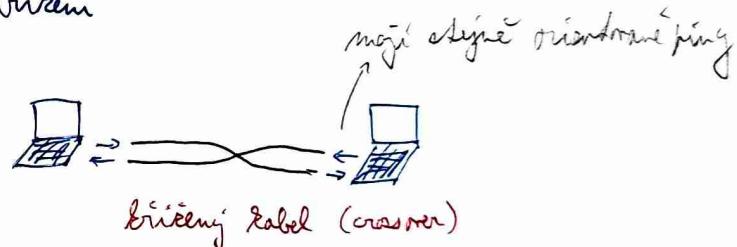
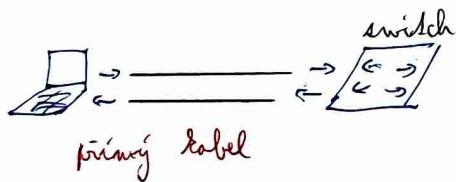
- infrastruktura hranicích Access point
 - ↳ ad-hoc peer-to-peer sítě - komunikují spolu mimo 2 rádiusy
- SSID (Service Set ID) = rozlišec pro rovnoběžní sítě
- problém se záberem frekvencí ?

Fyzická vrstva - OSI 1

- funkce: přenod digitální info ne analogovou a obnovení + priesad pro komunikaci mezi
- typy médií
 - metálkicí → el. pulzy
 - optické → světelné pulzy
 - běžatkové → modulace vln
- baseband priesad - přenáší primitivní signál a kóduje ho
 - Ethernet používá kód. Manchester $0 := \text{falling edge}$, $1 := \text{rising edge}$
- broadband priesad - přenáší rozložený signál a moduluje ho - frekv., amplituda, frekvenci

Nestiněná bronečná dvojlinky - UTP = Unshielded Twisted Pair

- dnes standardní kabel (metálkicí)
- 4 páry Cu vodičů navzájem proti sebe zavrtané - snižuje kruhovou propagační zpoždění a příjem elektrozvuku
- 100 Mbps Ethernet používá jen 2 páry ⇒ je možné využít i 2.
- konektory: RJ 45
- při připojení je třeba rohlednout pořadí závisek



→ dnes zvykle už není potřeba → používá se autokonfigurace MDI / MDIX

- alternativa s koncovým zámkem: STP

Optická vlákna

SiO₂

- signál se přenáší jaro viditelné světlo s vlnovou délkou vlnoviny vlákna
 - vysoké frekvence, velký bandwidth (rychlos), malý náklad, žádoucí rozsah
- nevýhody: 1 cena, náročný manipulace (malé ohýbání), nedostatek kabelu
- druhý vlastnost
 - jednoridová (singlemode) - svítí se laserem → 1 paprsek, ↑ dosah + bandwidth + cena
 - mnohoridová (multimode) - svítí se i LED

• Segmentace sítě

- repeater - řeší vzdálost signálů } příliš dál všem
- ne strukturované tabulácií hub
- bridge - spojuje segmenty na binární mřížce
 - řeší vzdálost propustnosti (rozděluje solitní doménou)
 ⇒ do kohoutí
 - ne strukturované tabulácií switch

} příležitě jen všechny
MAC adresy

zde se v místech někdy

- full duplex ⇒ mezijská propustnost
- celá síť oddělena frontovou pidełkoříží 1 IP síť a také 1 broadcast doména

• Learning bridge - BUM (BVS)

- režim práce switche, když si vám do své MAC tabulky přidává info o tom, za jakým portem jsou jaké adresy
- když si zaplní celou tabulku, tak už bude všechny rámců posílat do správných portů
 - ↳ s výjimkou broadcastu, mezinárodních unicastů a multicastů - ty bude posílat všem

⇒ BVS (Broadcast and Unknown Service) / BUM (Broadcast, Unknown and Multicast)

• Spanning Tree Protocol - STP

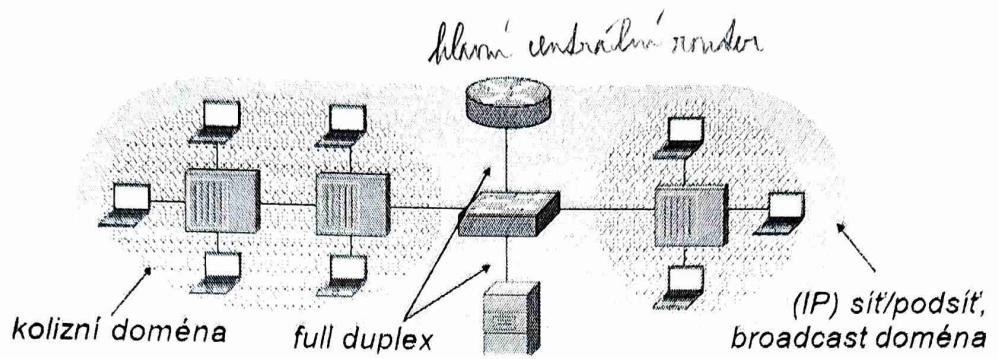
- někdy chceme mít pro vzdálost robustnost 2 segmenty propojené dvěma switchi
 - ↳ kdyby by switchy fungovaly současně, tak learning bridge by selhal

Důvod: graf je cyklický

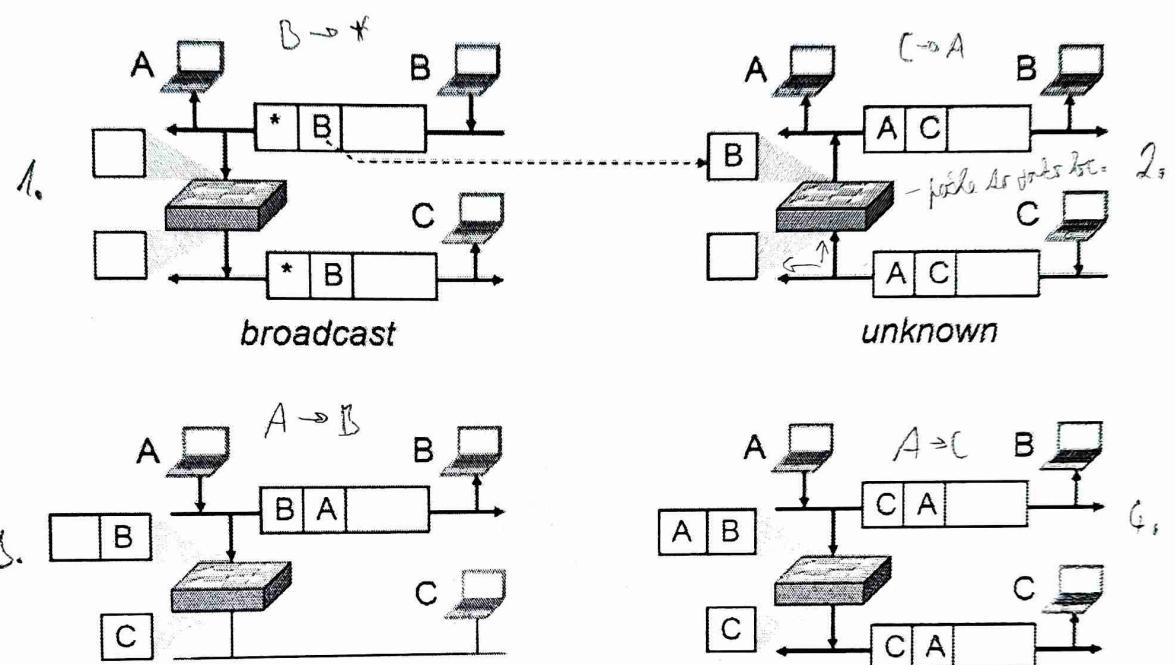
Rешení: najít kostru = spanning tree

⇒ switchy se musí držet protokolu, který z nich bude v režimu forwarding a Elegy a v režimu blocking - monitorovat, jestli nedostal žádoucí

- STP má několik sémantik, které mohou být použity
 - ⇒ obvykle lze STA na portu počítat. (first port) - je to na administrativní



LEARNING
BRIDGE



→ nice bridge

