

1. HODINA

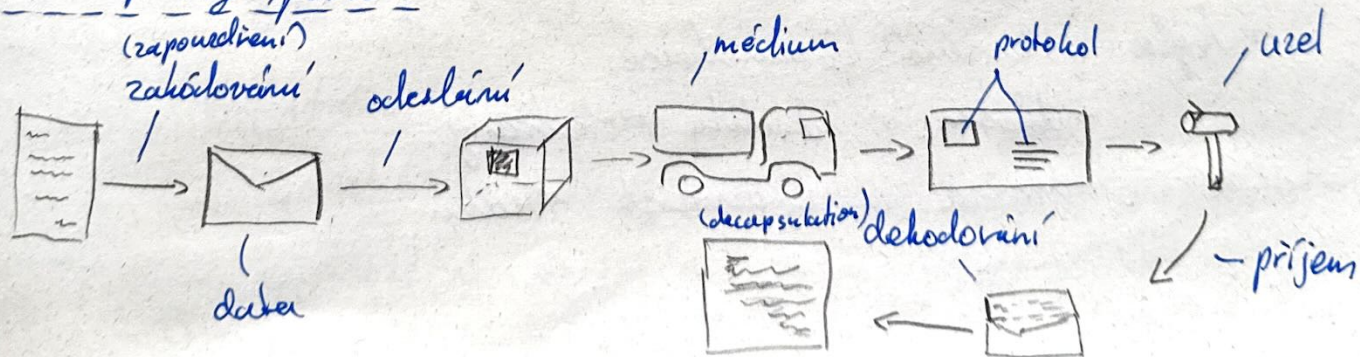
Obecné atributy komunikace

- identifikace → musejí se mezi sebou najít
- metoda → hlas, znaková řeč, psaná forma
- jazyk → dohoda na jazyce
- rychlost → rychlost komunikace, občas třeba zpomalit
- proces → např. porvání k prezidentské, nějaký rituál společnosti

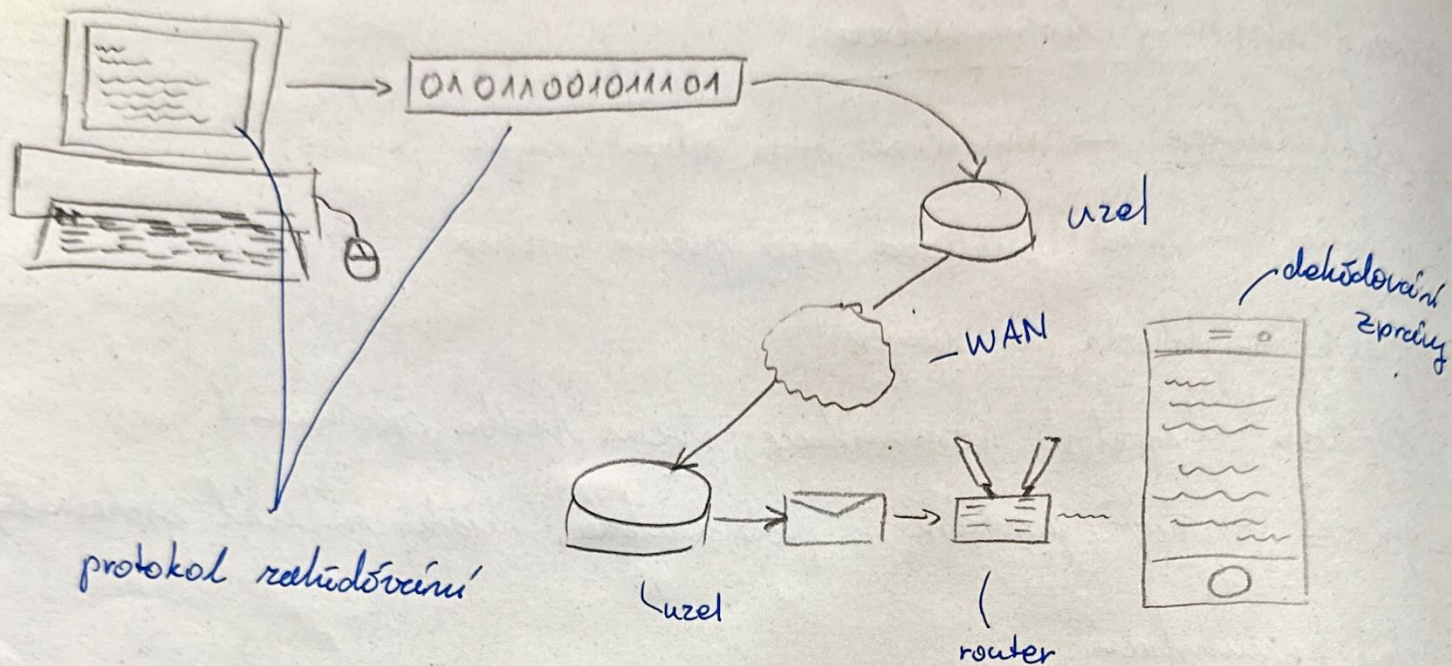
Typy komunikací

- lečinná → hlas, písmo, signály
- telekomunikace → zavedená pravidla, vše má pod kontrolou síť, telefon umožňuje přístup
- počítačová síť → volně dostupná pravidla
→ síť se stará o přenos a pravidla a logiku se věnuje konečné realizaci
- konvergovaná síť → počítače používají linky (drůbež)
→ pak telefony používají počítačovou síť (Voice over IP)
→ pak ale mobilní data zase vrátily počítačovou síť přes telefonní kanály

Přenos zpráv poštou



Přenos zpráv (e-mail)

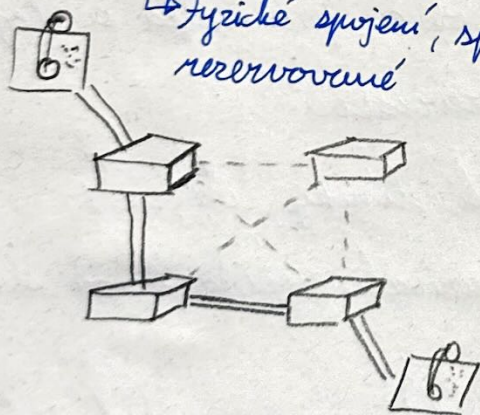


Poradavky - odolnost

↳ americké ministerstvo obrany → řeší výpadek telefonních sítí v případě útoků, protože přepínali obutky

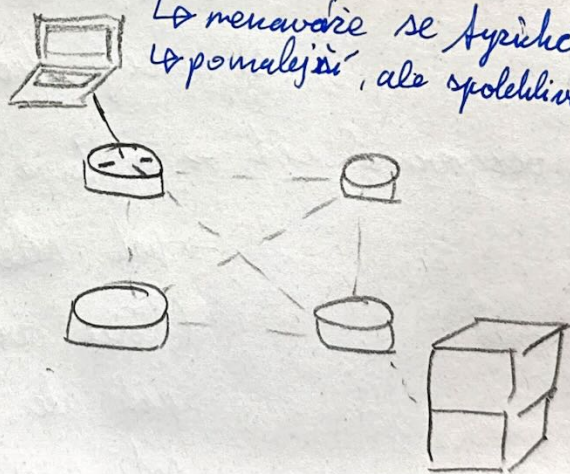
přepínání obutků

↳ fyzické spojení, spojení rezervované



přepínání paketů

↳ nenavazuje se fyzické spojení
↳ pomalejší, ale spolehlivější



Poradavky - bezpečnost

↳ řešila se fyzická bezpečnost, zavedla se ochrana dat

- ↳ byla:
- odevřená komunikace
 - důvěra v identitu protistrany
 - důvěra ve správnost obsahu

↳ rizika bezpečnosti:

1) fyzické napadení

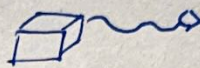
2) DoS útok (Denial of Service) → útok masivní komunikací

3) ukradení dat nebo zfalšování

↳ jak zabezpečit infrastrukturu?

1) omezení přístupu

2) záložní zdroje napájení



3) záložní servery s daty



↳ jak zabezpečit data?

1) ověřování uživatelů a ověření serverů / počítačů

2) kryptografie (šifrování dat)

3) inspekce dat (přes antivirusy)

Poradanky - rozšiřitelnost (LAN)

↳ „local area network“

↳ vytvoření 3 vrstev: → umožní jednoduché zapojení dalších rozšíření

1) core

↳ síť je připojena k ISP → „internet service provider“

↳ v místnosti s klimatizací a UPS → „uninterruptible power source“ neboli náhradním generátorem

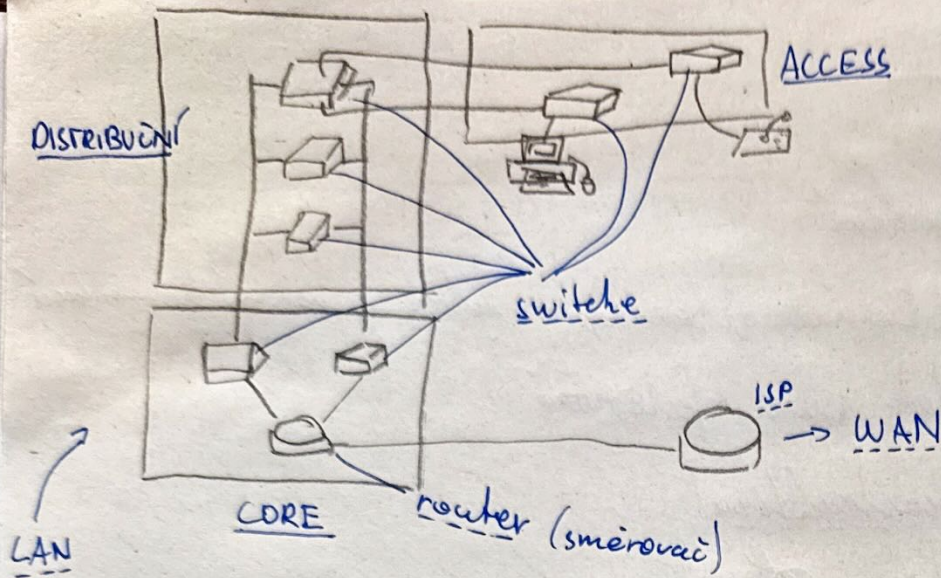
↳ má router připojený rovnou k ISP a na druhé straně ke switchům → propojují další velké sítě

2) distribuce

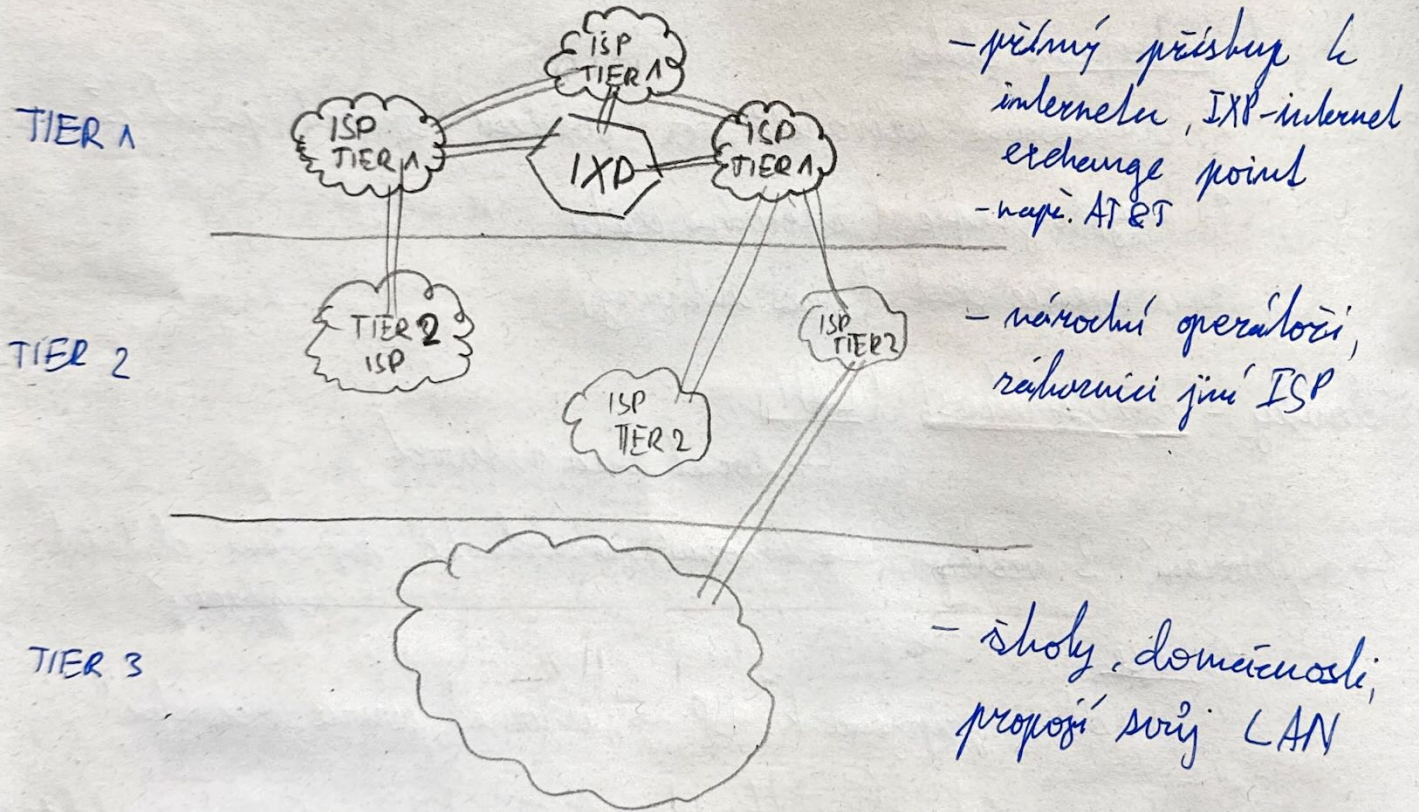
↳ po sítích

↳ „vertikální“ → obsahuje switche a propojuje další switchy v access vrstvě

3) access (přístup) → „horizontální“, propojuje počítače, listy...



Pořadanky - rozšiřitelnost (WAN)



Pořadanky - kvalita služeb

- přenosové parametry: → různé aplikace, různé požadavky:
- latence → rozdělení od zdroje k obdržením
 - YouTube - ztrátovost reálných
 - email - ale pravidelně
 - nesmí být ztrátová
 - pravidelnost doručování → jitter nebo-li jak objem holisa
 - stratovost dat → jak často není paket doručen
 - šířka pásma → kolik dat lze přenést, rychlost světla je vždy stejná

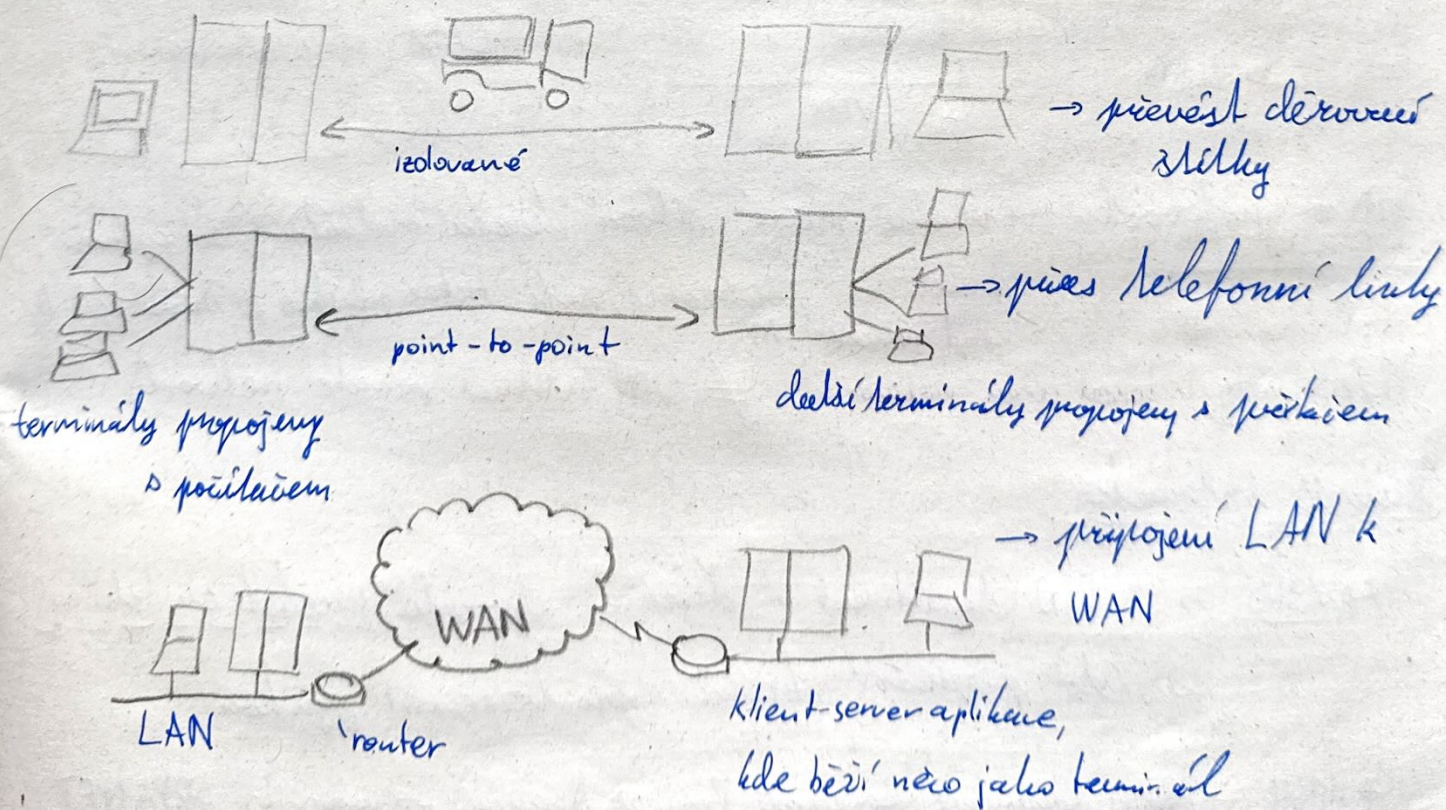
cíl:

- doručit rychle prioritní pakety
- navrhnout pro aplikace dostatečnou síťovou pásmu pro normální fungování

jak:

- pakety označit QoS (quality of service), abe měly navrhnout, že se aplikace budou chovat fčr
- vyhradit síťovou pásmu pro aplikace, abe měly se stát, že aplikace ji nevyužívá a pak obvykle zabírá místo
- prioritní fronty → tak, že správy jdou do fronty, abe priorita se dá zjistit až po načtení celého paketu, takže ostatní čekají

Typy počítačových sítí



Dělení sítě

- Lokální síť (LAN)

↳ malé vzdálenosti (kampus), jednotný vlastník

- Rozlehle síť (WAN)

↳ vzdálený, rozlehlý, mnoho vlastníků

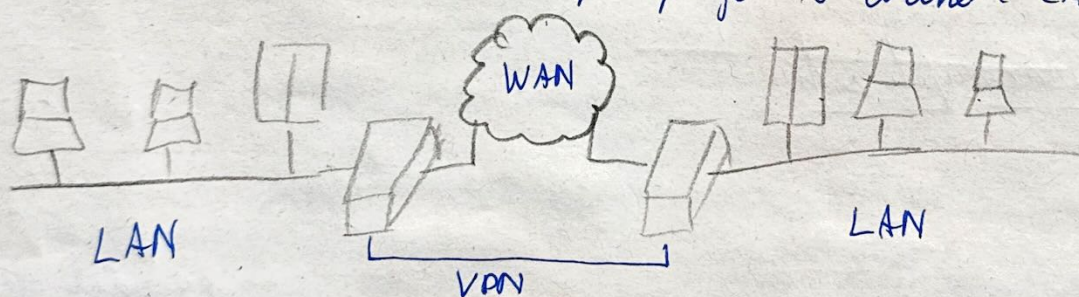
- Dnes → rozdíly jsou menší, rozdíl právě ve vlastnictví

Veřejné a privátní síť

↳ propojení dvou LAN sítí → pomocí VPN (virtuální privátní síť), aby
nešlo přes WAN přímo ale tunelem

síť se pak chová
jak jediná LAN

VPN → šifruje před odesláním a dešifruje
při přijetí v druhém LAN přes tunel



VPN v notebooku → vlastně nová síťová karta virtuální, co vytvoří
tunel v soukromé síti, takže vlastně se tváří jako by
byl součástí LAN

VLAN (virtual local area network) $\neq$ VPN (virtual private network)

Historie internetu

↳ 1960 → US DoD (Department of Defense) vytvořilo packet switching
rameno přepínání obrátek pro bezpečnost fyzikou

↳ 1969 → ARPA (Advanced Research Project Agency) → vytvořilo ARPANET,
kdy síť terminálů vytvořili pomocí point-to-point

→ první telefonní linky

1974 → vytvořen pojem internet (internetworking)

1977 → k ARPANET připojena další síť

1983 → TCP/IP nahraduje NCP (network control program) v ARPANETu

1985 → BSD (Berkeley System Distribution) přidali TCP/IP do Unixu

Request for comments (RFC)

IAB → IETF, RTF → RFC

↳ od 70. let → dnes ryklivá Internet Advisory Board pomocí
Internet Research Task Force a Internet Engineering Task Force

↳ aktualizace dostanou nové číslo po schválení komisí

↳ vlastně standardizace internetu, co by se mělo dít

↳ na ietf.org/rfc.html

Úlohy: ①

1) Výhody / nevýhody přepojování paketů.

2) příjev na síťových protokolech, že vznik síť inicializovala armáda?

3) požadavky na škálovatelnost sítě

4) náklady emailu a telefonu přes síť → rozdíly

5) definice LAN

6) definice VPN

Ústřední filozofie- Zapisovatel

↳ musí psát správně → něco jako protokol

- Sekretářka

↳ dopis zabalí a doplní adresu a pošle na podepsání

- Podatelna

↳ vezme balík a vybere strategii odeslání

- Výhody

↳ o změnách nemusí ostatní vrstvy vědět

↳ každá vrstva dělá jednoduchý úkol (dekompozice)

Síťový model, síťová architektura

Síťový (referenční) model → např. OSI (Open Systems Interconnection)

↳ rozděluje práci mezi vrstvy

Síťová architektura (protocol suite) → např. TCP/IP (sada protokolů)

↳ vezmeme síťový model a přidáme další technologie, např. protokoly

OSI (Open Systems Interconnection)

↳ neporovnává se s tím, jak se vyvíje

↳ základní referenční model + protokoly

↳ navržená řešení

Vrstva	Funkce
7	aplikační předání informací uživateli
6	prezentací řešení rozluštění dat po přenosu - např. endianita
5	relační řídit dialog mezi aplikacemi při spojení
4	transportní spojení mezi 2 uzly a přenos dat mezi nimi
3	síťová posílání paketů do různých sítí
2	linková zdroj a cíl přenosu mezi dvěma uzly □-□
1	fyzická přenos bitů fyzický jako fyzikálně

X.400 (Message Handling System)

- ↳ implementace pošty pomocí OSI
- ↳ Microsoft Exchange Post
- ↳ složitá struktura adres, ale jednoznačné

G = Libor S = Forst
O = Charles University } vhodné info
...

X.500 (Directory Access Protocol)

- ↳ adresářová služba, něco jako telefonní seznam s možností vyplnit oblíbený nápoj

X.509 (specifika veřejných klíčů)

- ↳ asymetrická kryptografie
- ↳ každý klíč musí mít jedinečný identifikátor

LDAP (Lightweight DAP)

- ↳ databáze údajů o osobách a službách

Rodina protokolů TCP/IP

OSI	Vrstva	Příklady protokolů
7	<u>aplikační</u> zobrazení webu odesílání mailů	FTP HTTP SMTP
6		(doménový systém) DNS SIP
5		NFS XDR RPC
4	<u>transportní</u>	TCP UDP
3	<u>síťová</u>	IP NAT ICMP
2	<u>síťové rozhraní</u>	Ethernet, Wi-Fi, FDDI
1		ATM, SLIP, PPP

ping (diagnostika sítě)
Internet Control Message Protocol

nalezení Mac adresy pomocí IP
optika
Address Resolution Protocol

nejšora v TCP/IP
↳ jsou to jen způsoby přenášení
na koncové aplikace / uživatele

data v malých bunčích přes vyhrazené seriové linky

- ↳ navržená zespoda, nebo-li od nejjednodušších aplikací

Spojované / nespojované služby

Spojované (TCP)

- ↳ spolehlivé doručení dat
- ↳ implementace aplikace je jednoduchá, ale TCP je složitější
- ↳ takové telefonní spojení

Nespojované (UDP)

- ↳ něco jako poštovní spojení
- ↳ aplikace musí tedy sama řešit potvrzení doručení a případné opětovné odeslání
- ↳ jen pošle paket a o nic víc se nestará UDP

Aplikační modely

Model klient-server

- ↳ navázání spojení → klient podle adresy najde server
- ↳ klient řadí požadavky
- ↳ server obsluhuje většinou více klientů
- ↳ upload: klient na server
- ↳ download: server na klienta
- ↳ př. DNS, WWW, SMTP

Model peer-to-peer (P2P)

- ↳ nejsou vybraněné role
- ↳ není potřeba znát adresu zdroje dat
- ↳ Napster, Gnutella, BitTorrent

Adresování počítačů

↳ různé typy adres podle vrstvy, kde komunikace probíhá

↳ HW (linková vrstva) → MAC (Media Access Control) adresy

↳ dříve udělovány výrobci, dnes nastavitelné

↳ ethernetové adresy → 3 bajty prefix výrobce

→ 3 bajty číslo síťové karty

↳ není pro komunikaci mezi sítěmi, nerespektuje

topologii nebo-li mac adresu bude mít jiné
stejně

↳ IP adresy (Internet protokol) → SW (síťová vrstva)

↳ používá v protokolu TCP/IP

↳ 2 typy: 1) IPv4

2) IPv6

↳ respektuje topologii → v adrese je: a) adresa sítě pro komunikaci
mezi sítěmi

b) adresa počítače pro identifikaci
rozlišení uvnitř sítě

jednoznačně určí síť a
počítač

↳ doménová adresa → aplikační vrstva (pro lidi)

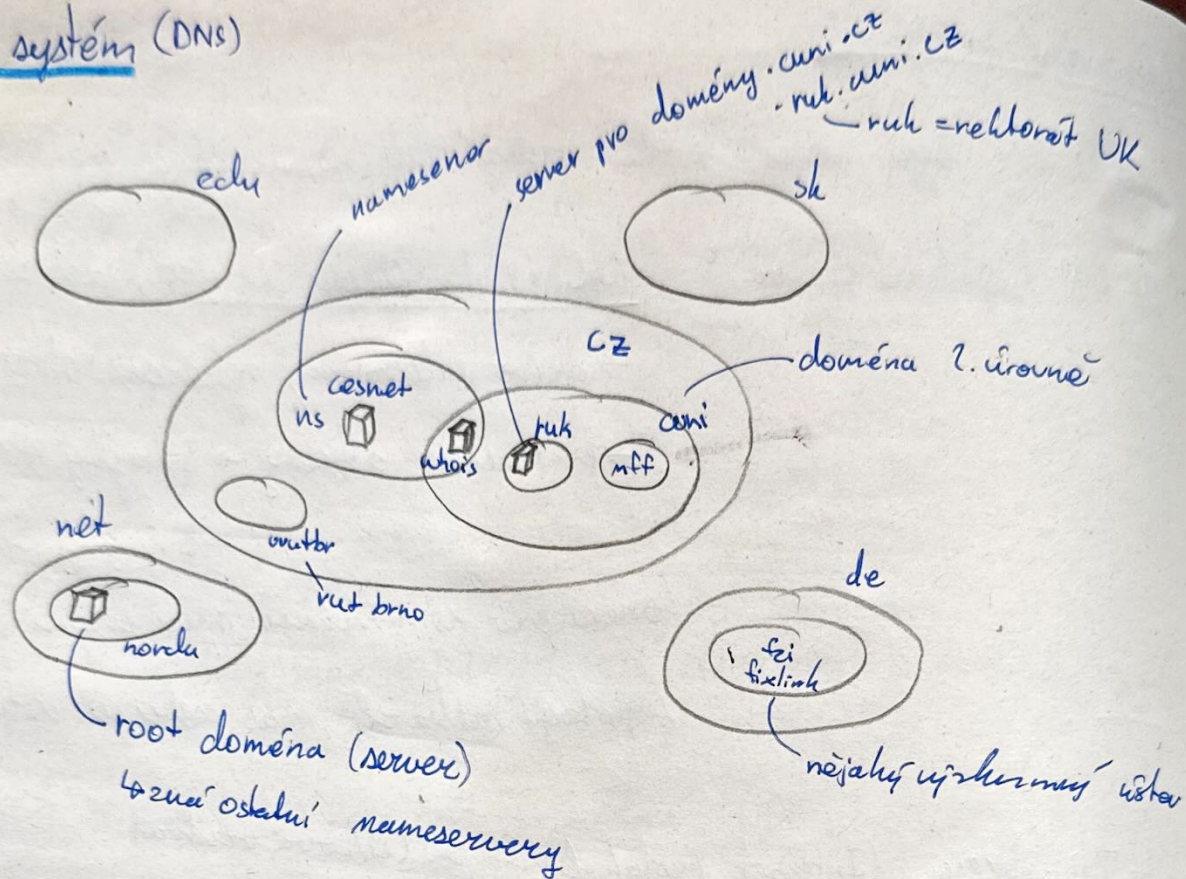
↳ hierarchie je zprava doleva www.mff.cuni.cz

↳ poslední je TLD → doména nejvyšší úrovně (kód země)

ARP → konverze mezi síťovými (IP) a fyzickými (MAC) adresami
→ Address Resolution Protocol

DNS → konverze mezi síťovými a doménovými
→ Domain Name System

Doménový systém (DNS)



Správa domén

- ↳ ICANN (Internet Corporation for Assigned Names and Numbers)
 - ↳ spravuje domény nejvyšší úrovně, RFC 920
 - ↳ arpa ("ARPA net") → technické
 - ↳ definování systému doménových jmen a TLD
 - ↳ ISO kódy zemí (CZ, sk, uk, eu, nu, tv, ...)
 - ↳ nejsou ISO kódy
 - ↳ rezortní -5 (com, org, edu, mil, gov, net)
 - ↳ komerční, nekomerční, armáda, vláda, síťové organizace
 - ↳ přidání dalších (info, biz, aero, ...)

↳ TLD .cz

- ↳ nemá rezortní architekturu
- ↳ spravuje CZ.NIC → cca 1 mil. jmen

SLD

- ↳ second level domain
- ↳ spravuje vláda



IP adresy

↳ koncové uzly v síti TCP/IP musí mít IP adresu

↳ IPv4 → 4 byty (např. 135.113.19.71)

↳ IPv4 → 16 bytů (např. 2001:718:1e03:a01::1) ← ^{mily ::}

↳ prefix → definuje síť

→ veřejné adresy pro přístup k internetu přidělí ISP

→ privátní adresy si můžeme rovněž přidělit sami → registry

bezpečnost, ale interoperabilita → směrovací adresy NAT

network address
translation

↳ přidávání adresy:

↳ rovně specifika sítě

↳ statické → předefinovaná IP adresa

↳ dynamické → přidávána na vyžádání

→ musí být ověřeno, že se smí připojit

→ DHCP registrační přetváření

↳ Dynamic Host Configuration Protocol

↳ link-local → počítač si přidělí sám v rámci sítě → problém
duplicity

Port, socket

Port

↳ 16-bitové číslo, co určuje aplikaci / službu na koncovém uzlu
komunikace

↳ destination-port < 1024 (většina)

↳ port z well-known services

↳ např. webové služby → 80

↳ source-port

↳ počítač systém o přidávání generického portu, který používá
v komunikaci

↳ aby bylo možné identifikovat vše klienti nájednou
pro všechny služby

Socket

- ↳ označení jednoho konce komunikačního kanálu
- ↳ $\langle \text{IP adresy, Port} \rangle$
- ↳ TCP/IP $\langle \text{zbojová IP, zbojový port, cílová IP, cílový port, protokol} \rangle$
- ↳ dva různé kanály stejného serveru se musí lišit aspoň v jednojím pruhu
- ↳ stejná čísla portů lze použít pro různé typy protokolů

Průběhy Well-known services (znat!)

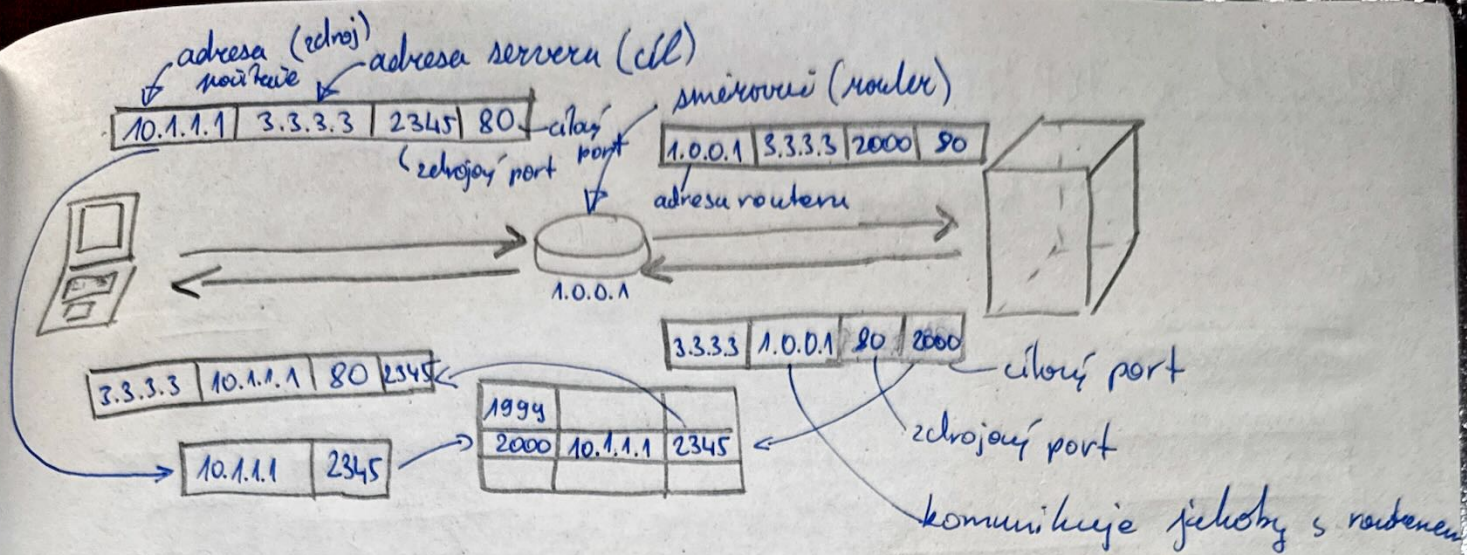
↳ používá primárně TCP, ale lze i UDP

- FTP (file transfer protocol) → 21 / TCP
↳ přenos souborů
- SSH (secure shell) → 22 / TCP
↳ přenos souborů a přihlášení
- SMTP (simple mail transfer protocol) → 25 / TCP
- DNS (domain name system) → 53 / *
↳ protokol je jedno UDP/TCP
- HTTP (hypertext transfer protocol) → 80, 443 / TCP
↳ 80 nešifrované
↳ 443 šifrované
- SIP (session initial protocol) → 5060, 5061 / *
↳ VoIP, IP telefon

Průběh adres (NAT) → network address translation

→ „IP masquerading“ (maskování IP)

- ↳ lokální síť používá privátní adresy a veřejné má jiné



→ vlastně port se remapuje k privátní adrese do registru (zdrojový počítač)

a server si připočítá, se komunikuje s routerem, který nahradí jak port, tak IP adresu za veřejnou routera, tak vlastně mění socket < IP adresa, port >

Adresování služeb

Uniform Resource Identifier (URI) → RFC 3986

→ záměnné s URL (Uniform resource locator), což málo komplementovat URN (Uniform Resource Name)

→ jednotný systém odkazů pro více služeb (FTP, WWW)
 ↳ http (není protokol!)

URI = schéma : [//] autorita [cesta] [?dotaz] [#fragment]

↑
obras

↑
server (doména, plněná)

↑
nестandardní port, je třeba ho zde uvést jinak well-known services

↑
v autoritativním systému

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

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

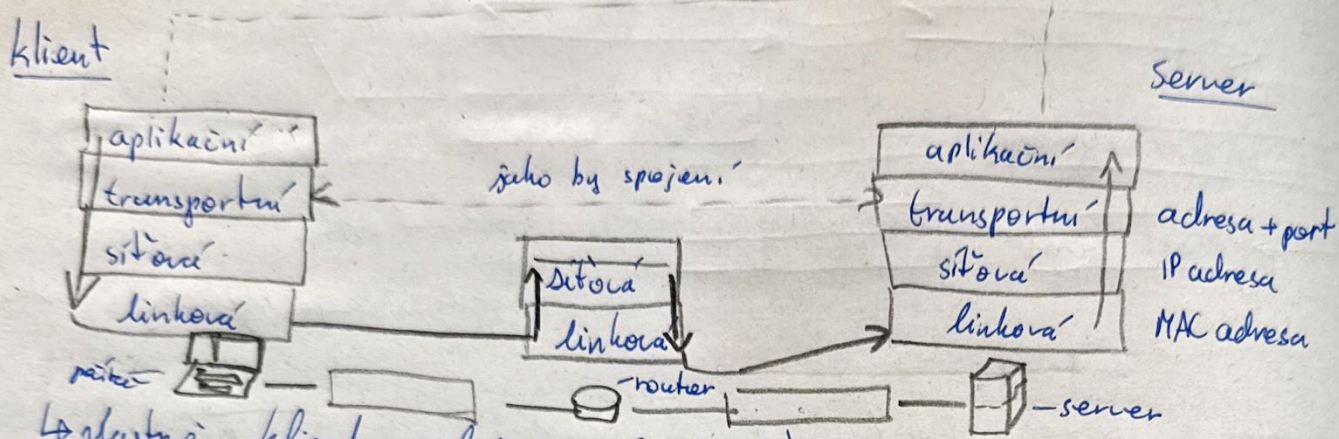
sip: 221911111@voip.cz

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

mailto:forst@cuni.cz

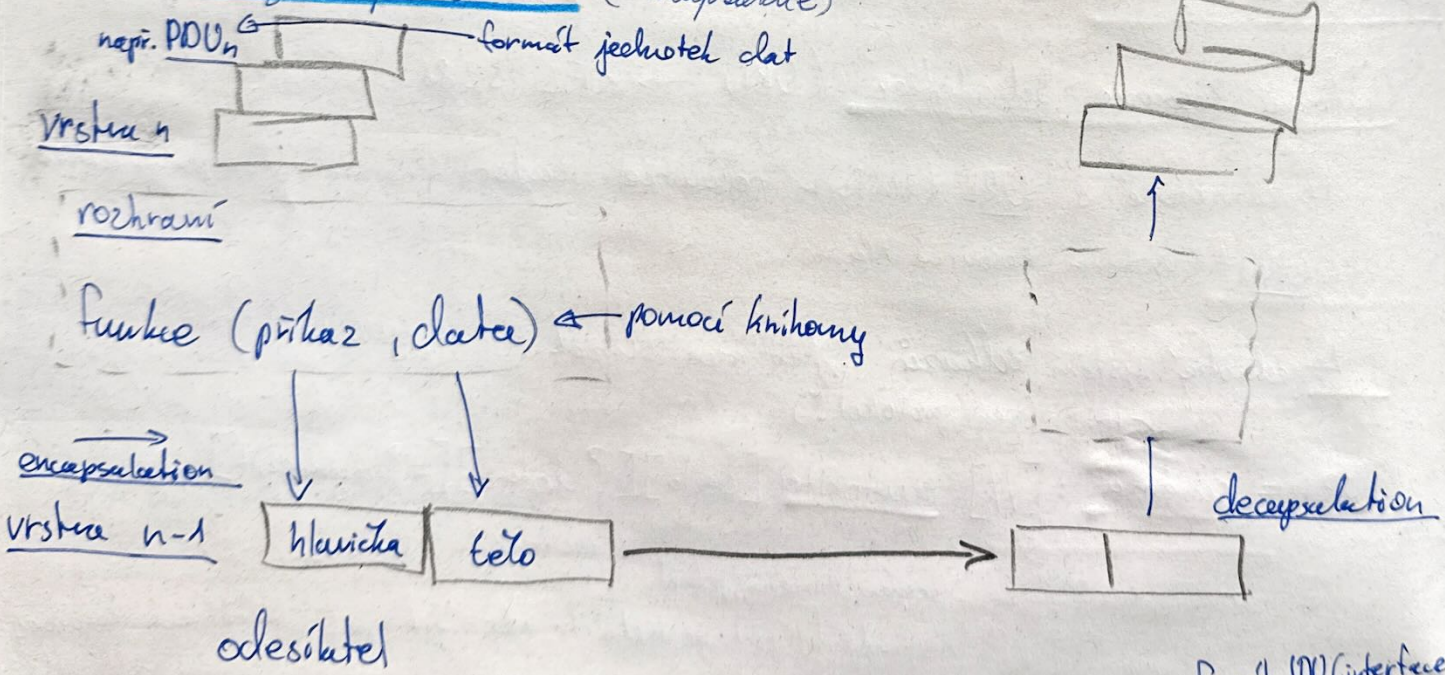
Dalový lok v TCP/IP

URI (URL)
http://www.mff.cuni.cz



→ vlastně klient zadá URI (aplikační) a aby navázali spojení s adresou a portem služby je potřeba najít IP adresu serveru, takže ve stejné síti nejde nejbližší next-hop uzl, což je server jinak musí přes router do jiné síti a to jde přes síťovou vrstvu posíláním dat a nalezení next-hop uzlu → a to pak nespěchá

Multiplexing, zapouzdření (enkapsulace)

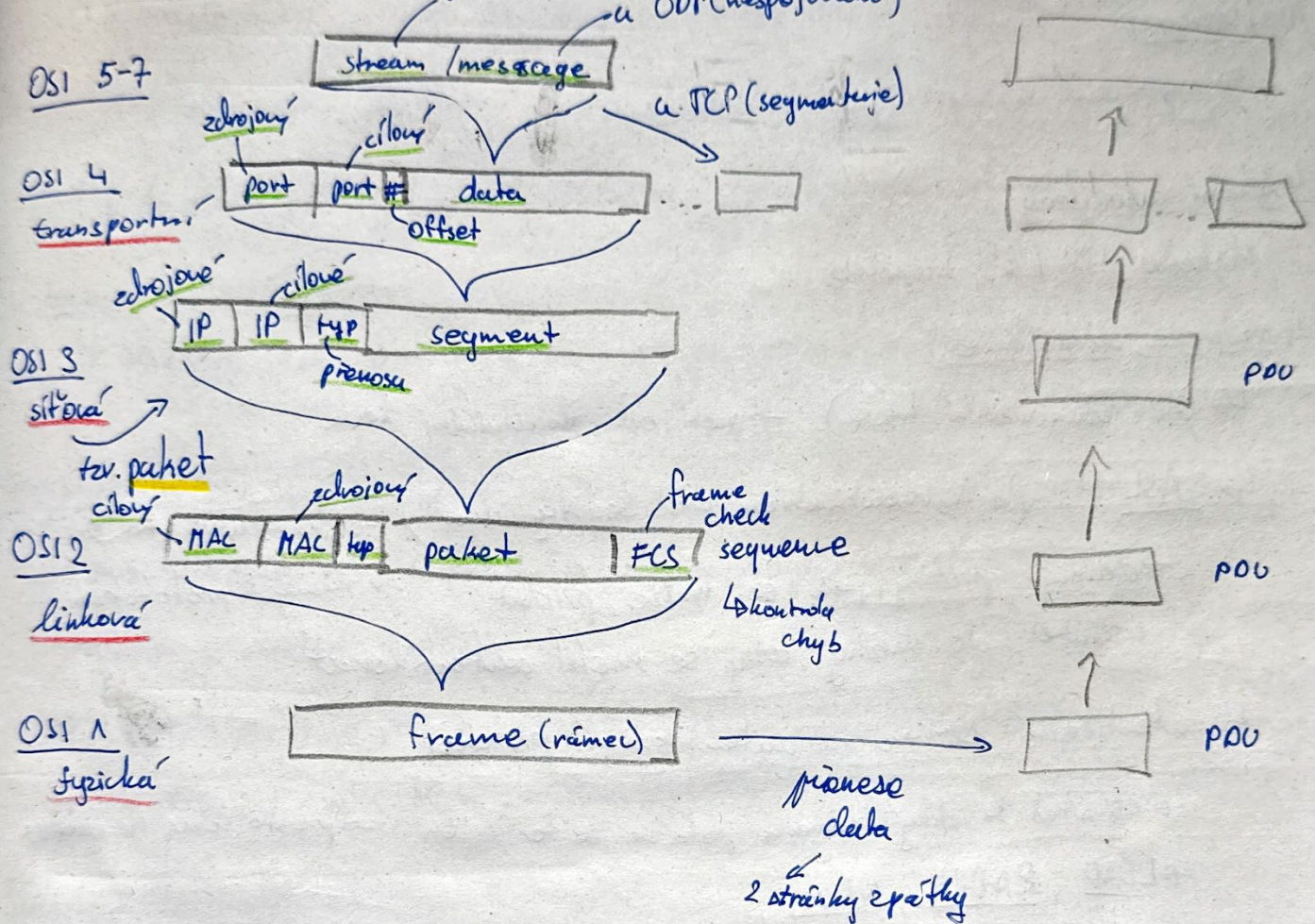


→ PDU (protocol data unit)
→ vrstva n posle do n-1 nebo do jakéhokoliv jiné vrstvy, kde se informace zapouzdří díky rozhraní (hraničce) a podle protokolu se posle → různé by měly navr. IPv4 a IPv6

↳ přijímáme to nahoru otevře (deapsulation)

Typy PDU v TCP/IP

↳ formát jednotek dat = PDU
v TCP (proud dat) - spojované
a UDP (nespojované)



↳ po přenesení dat, FCS zkontroluje, zkontroluje MAC adresu a poté se rozhodne kterým hop - uzelu paket předá

↳ UDP → zná doména aplikací

↳ TCP → segment uložen a datový blok předán až po přijetí všech segmentů

Γ Otázky: Popište rozdíl mezi TCP, UDP, IP?

2) Jaké adresy se používají na aplikacím, transportní, síťové, linkové a fyzické vrstvě?

3) Jaký je rozdíl mezi klient-server a peer-to-peer modelem?

4) Jak jsou spravována doménová jména?

5) Jak se přidělují IP adresy?

6) Jaký je smysl a princip NAT? 8) Co je multiplexing a zapojení?

3. HODINA

Autentikace, autorizace

↳ autentikace

↳ ověření identity uživatele

↳ autorizace

↳ vyměření služeb po autentikaci

↳ lokalní autentikace

↳ heslo, HW klíč, biometrie

↳ vzdálená autentikace

↳ OTP (jednorázová hesla) → proti odposlouchání hesla

↳ SASL ("Simple Authentication and Security Layer") → framework na ověření

↳ definuje profily, jak se může připojit v různých protokolech
uživatel do služby, aby se mohl autentikovat

↳ autentikační server a autentikační protokol

↳ uživatel se díky serveru prokáže, že to je on → prostě čas i guess

↳ LDAP, RADIUS, NTLM,

↳ lightweight directory access protocol

↳ katalog uživatelů, služeb

přihlašování do Wi-Fi

stanice autentizace Microsoft

one time password

offline

↳ mechal vygeneroval server hesel a jedno po druhém zaslával

challenge-response

↳ místo hesla dostal řetězec, ten vložil do SW, přidal heslo a
vypadal jiný řetězec a to bylo místo hesla

↳ to samé, jak přijde přihlášení mailu přes server

token

↳ generuje časově omezený kód v synchronizaci se serverem

Kryptografie - symetrické šifrování

↳ pro šifrování a dešifrování se používá stejný klíč

↳ rychlé a vhodné na velká data

↳ nevýhoda je, že je potřeba si klíč předat bezpečnou cestou

↳ DES, Blowfish, AES, RC4 → dnes používá

Kryptografie - asymetrické šifrování

↳ pro šifrování a dešifrování se používá jiný klíč

↳ tajný a veřejný

↳ matematické inverzní funkce (matematický základem - jednostranné funkce):

- násobení prvočísel a pak jejich rozklad → docela hard

- diskrétní logaritmus $m = p^k \bmod q$

↳ každé pomalejší než symetrické

↳ někdo šifruje veřejným klíčem a pošle data a já si je otevřu tajným klíčem

↳ problém ověření, že veřejný klíč je opravdu veřejný klíč
člověka

↳ RSA, DSA, ECDSA

Kryptografie - hashovací funkce

↳ pevný kód z daného textu

↳ mělo by být jednoznačné

↳ kryptografie

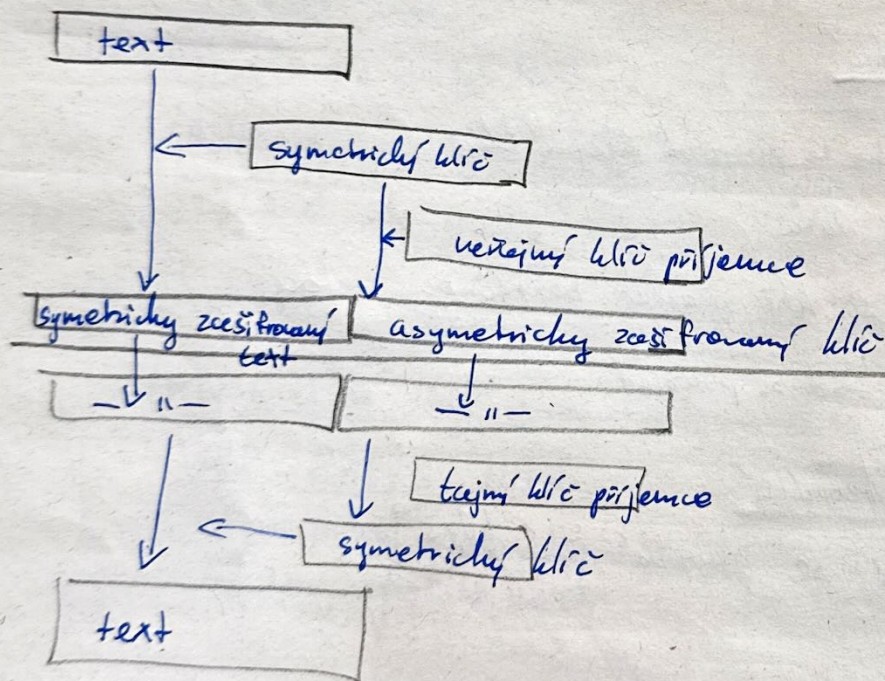
↳ jednosměrné

↳ malá změna textu → velká změna hashe

↳ text neodvratitelný z hashe

↳ SHA (secure hash algorithm)

Šifrování dat

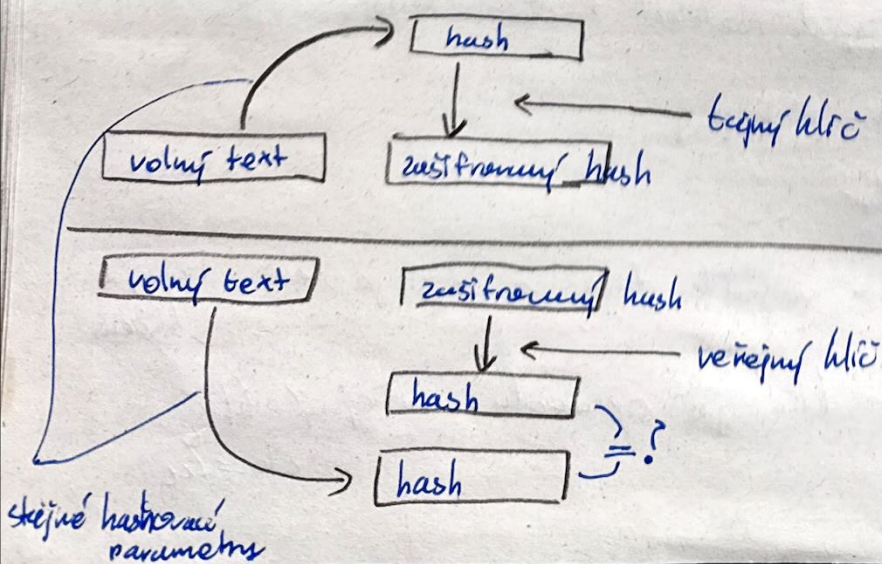


výhoda:

- šifruje asymetricky jen
malý klíč

- pro více příjemců tak
zašifruje jen jednou,
ale klíč musí
zašifrovat pro každého
zvlášť

Elektronický podpis



Diffie Hellmanův algoritmus

↳ Osoba 1 zvolí a a prvočísla p, q

↳ $A = p^a \bmod q$ a A, p, q pošle osobě 2

↳ Osoba 2 zvolí b a spočítá $B = p^b \bmod q$ a pošle B osobě 1

↳ Osoba 1 $\rightarrow s = B^a$
↳ Osoba 2 $\rightarrow s = A^b$ } $s = s$

↳ $A^b = (p^a)^b = p^{ab} = p^{ba} = (p^b)^a = B^a$

↳ bez znalosti b a a nelze dojít ke stejnému s a sítnu prolomit
→ pro větší a, b

Autenticita veřejných klíčů

↳ zabezpečení autoritou

↳ PGP (pretty good privacy) → web of trust → někdo komu věřím podepíše, že ten klíč je správný

→ poslu klíče a identifikaci zveřejněnou a porovnám o ověření

↳ CA (Certification Authorities)

↳ vydá certifikát

Certifikát → dle seznamu důvěryhodných CA

↳ certifikát je identifikace vlastníka klíče ověřená autoritou

↳ certifikát podle X.509 (použit v SSL) v SSH se používá jiný strukturu

- verze certifikátu

- sériové číslo certifikátu

- subjekt

- doba platnosti

- vlastnosti veřejného klíče, informace o veřejném klíči (algoritmus)

↳ i CA může ztratit důvěru

SSL, TLS

↳ Secure Socket Layer 3.0 → Transport Layer Security 1.0 → 1.1, 1.2, 1.3
↳ chmes nové

↳ medzi aplikačnou a transportnou

↳ HTTPS (HTTP přes SSL/TLS) → přes port 443
↳ chmes

Principy:

- 1) pošte požiadavek na SSL spojenie
- 2) server pošte odpovedi + certifikát
- 3) klient pošte zasifrovaný ^{verejným} zariadením šifrovacieho kľúča
- 4) server rozšifruje tajným ^{zariadením} šifrovacieho kľúča
- 5) probíha šifrovanie spojenia

Aplikačná vrstva TCP/IP

↳ spojuje OSI 5, 6, a 7

↳ pravidla komunikácie medzi klientom a serverom

↳ protokol definuje:

- priebeh dialogu
- formát zprávy → binárny / textový
- typy zprávy → požiadavky / odpovede
- sémantika → každá zpráva má len jednu interpretáciu
- interakcie s transportnou → jestli použije TCP alebo UDP

Domain Name System

↳ překládá jména na adresy a naopak

↳ používá UDP i TCP

↳ do 512B větší objem

↳ dnes i více díky EDNS

↳ pokud odpověď od serveru delší → přikaz truncate a klient naváže TCP pro kompletní

↳ klient kontaktuje servery v konfiguraci a postupně zvyšuje timeout

↳ jednotka záznamu: (resource record - RR)

↳ ns.cuni.cz 3600 IN A 195.113.19.78

jméno

↳ doba platnosti typ a data

↳ může se psát kteréhokoliv nameserveru

DNS záznamy

	Typ	JMÉNO	Data
start of authority	SOA	jméno domény	obecné info
nameserver	NS	jméno domény	jméno nameserveru domény
address IPv4	A	jméno počítače	IPv4 adresa
address IPv6	AAAA	jméno počítače	IPv6 adresa
pointer record	PTR	pro 1.2.3.4, to je 4.3.2.1. in-adetr = arpa	doména PC
canonical name	CNAME	jméno aliasu	každé jméno kompu
mail exchanger	MX	jméno domény	jméno poštovního serveru a priorita

pro IPv6: 1:1 → 1.0...0 . ip6.arpa
32 bit

Levery DNs

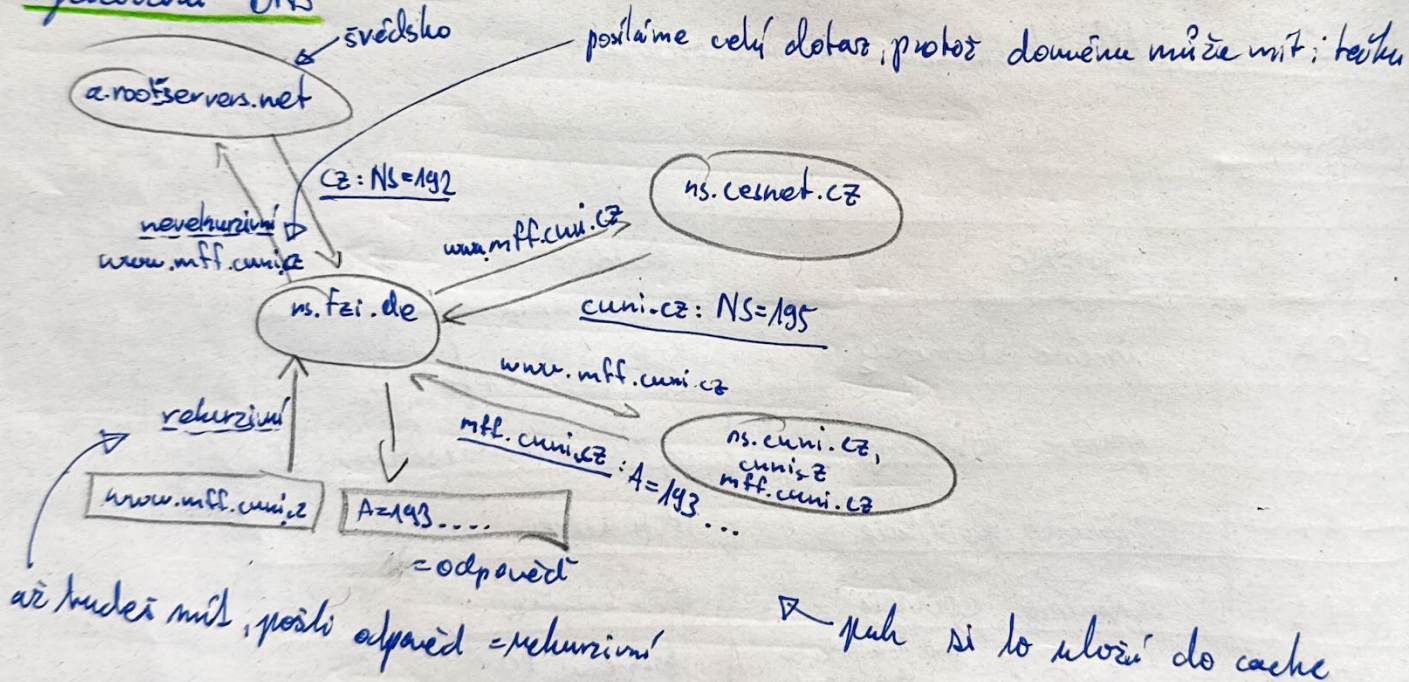
lypy:

- 1) primærui → ten ma datatäsi jinen
- 2) sekundærui → Balokky, slahyji si k master primæruiho serveru
- 3) caching-only → po mejahou dobu si je ulozi

↳ kućala domaćina by najčešće imala nult ili nimeserreni registracijski

↳ sekundarni server se aktualiziruje podle času uvedeném v SOAP,
ale je možné i že primární server informuje o změně dat

Thyrisovani DNS



DNS dotaz a odpověď (i když je binární)

Dotaz: Id: n

průběhy - FLAGS: Recursion

QUERY: www.unilz IN A

Odpověď

3p: n

zopakuj FLAGS: Autorative Answer

→ QUERY: www.univ.cz IN A

ANSWER: www.cuni.cz IN CNAME tarantula

tarantula N A 195. 113. 89. 35

AUTHORITY: ceni.cz IN NS golias
ADDITIONAL: golias IN A 195.113.0.2 ← pro přístě

Bezpečnost DNS

↳ protože je jednoduché falsifikovat dotazy

pochrana je, že ID a zdrojový port → celková tak c. miliardy možností
a to může být rychle stihnout

↳ může napadnout lokální síť

↳ cache poisoning

↳ domku klienta aby mu poslal DNS dotaz na server pomocí reklamy
ve spamu → pak pošle korektní odpověď s jiným autoritativním
nameserverem pro doménu c. např. → tím může ovlivnit další jeho
dotazy

↳ řešení: používat jen root servery a ne cache

↳ jen tehdy pokud je automaticky

↳ DNSSEC: - podepisování známých klíčem veřejného serveru → utvořit
rozšiřuje se pomalu → komplikovaný
například c. nameserver

Diagnostika DNS

nslookup → set type

→ server

→ ls

→ exit

} napíše nameservery

dig / drill → [doména]

úkoly: 1) základní principy kryptografických algoritmů a rozdíl mezi nimi.

2) princip šifrování a elektronického podpisu.

3) účel a podstata Diffie-Hellmanova algoritmu

4) jak ověřuje pravost veřejného klíče.

5) rozdíl mezi HTTP a HTTPS

6) kdy se používá UDP a kdy TCP?

7) účel a chování rekursivního nameserveru

8) postup řešení DNS dotazu

9) bezpečnostní rizika DNS

4. MOBITNA

File Transfer Protocol (FTP)

↳ jeden z najstarších protokolů (RFC 959), ale platí dodnes

↳ princip: - vzdálený přístup k úložišti, aby bylo možné přenášet soubory

- autentikace byla přenosem otevřeného hesla - bezpečnostní riziko

↳ dnes: - anonymní přístup k veřejně dostupným souborům

- uživatel pošle jen e-mail se statistickými důvody
jeho hesla

↳ plněto kóduje funkce HTTP, ale i dnes některé servery používají protokol FTP

Účinné spojení

1.1.1.1: 1234 ↓ dostal oct operačního systému
↑ navrácení spojení → 2.2.2.2: 21

← 220 Ftp.cuni.cz

→ USER forst

← 330 Password req'd

→ PASS hadej

← 530 Login failed

→ QUIT

← 221 Good bye.

"230" - povedl by se login

↳ FTP stále rozlišoval odpovědi kojmístním kódem → zjednodušuje komunikaci

Kódové odpovědi

↳ první číslice vyjadřuje závažnost odpovědi

1xx → předběžná kladná odpověď

↳ server přijal požadavek a začal ho řešit, ale budou ještě další zprávy od serveru, takže klient nemusí nic dělat → klient čeká

2xx → definitivní kladná odpověď

↳ server dokončil požadovanou operaci

3xx → neúplná kladná odpověď

↳ server operaci přijal, vše je v pořádku, ale poskytl ještě dodatečné informace, např. heslo

↳ v tomto případě server čeká, až klient odpoví

4xx → dočasná záporná odpověď

↳ operace se nepovedla, ale není to nic fatálního a po nějakém čase lze úplně stejný požadavek znovu zopakovat a je šance, že se povede

↳ např. přístizání serveru, nebo 404 u HTTP protokolu, stránka nenalezena

5xx → definitivní záporná odpověď

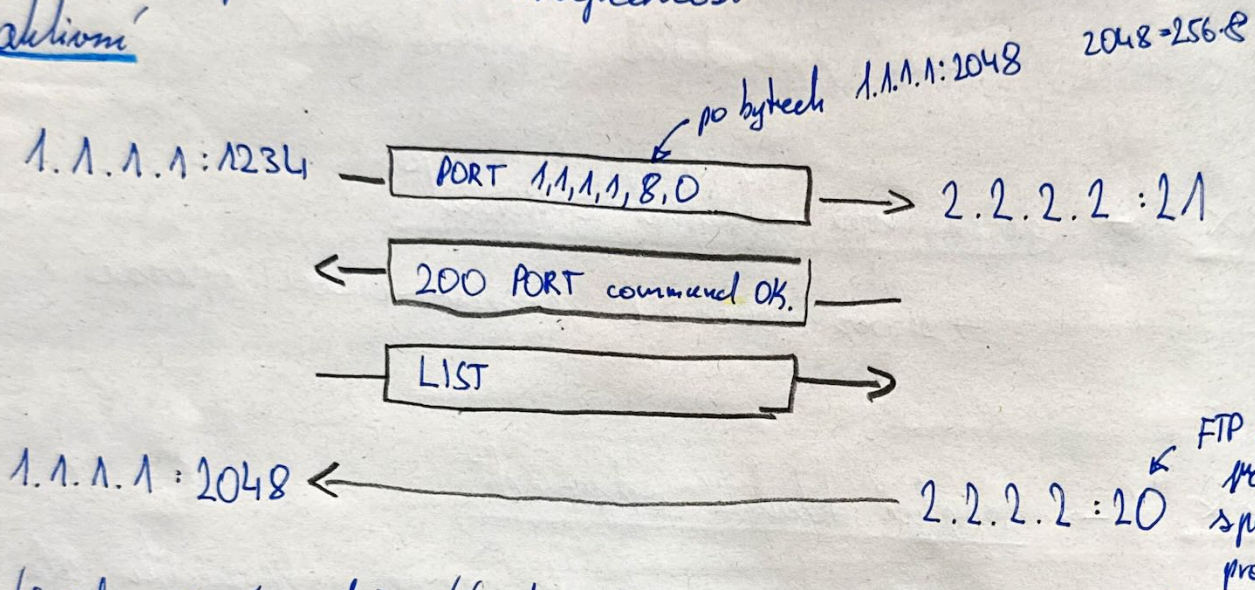
↳ operace se nepovedla a nemá cenu to opakovat, protože znovu neprojde → např. špatné heslo

Aktivní / pasivní datové spojení

voice over IP, spojení přes
TCP/IP $\Rightarrow$ slouží pro
problem

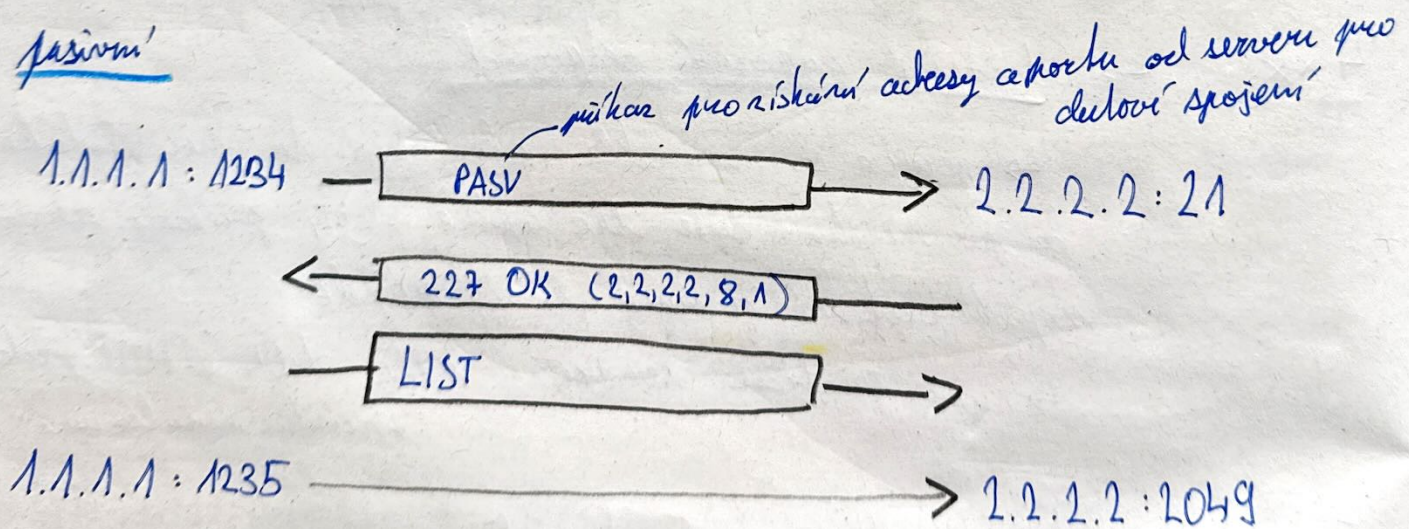
- ↳ přenos dat probíhá po jiném spojení než řídicí $\rightarrow$ u FTP, SIP
- ↳ existují 2 druhy, kde pasivní znamená, že server je pasivní a datové spojení naváže klient
aktivní znamená, že server je aktivní a

navazuje datové spojení na klienta
↳ používá se pasivní kvůli bezpečnosti
aktivní



↳ nebezpečné, protože klient může přijímat spojení a ne otevírat. Firewall často nedovolí. Řešením se to dovede na 5 sekund jasně určit port a IP adresu z určitého portu a IP adresy

pasivní



↳ ty směry v datovém spojení ukazují jen směr navázání spojení, ale data ptečou na obě strany

Aplikace pro FTP

- ↳ WWW prohlížeče → umožňuje navázat FTP spojení, často jen pro download, pro upload je třeba rozšíření
- ↳ správci souborů → např. Total Commander dokáže pomocí FTP protokolu provádět běžné operace na FTP serveru
 - ↳ umožňuje taky vidět různé manuály a další věci k nim v online

↳ příkaz ftp

- ↳ příkazy a proměnné
- ↳ navázání relace: open, user
- ↳ ukončení relace: close, quit, bye
- ↳ vzdálené příkazy: cd, pwd, ls, dir
 - ↳ print working directory
 - ↳ change directory
- ↳ práce se soubory: delete, rename, mkdir, rmdir
 - ↳ remove directory
- ↳ lokální příkazy: lcd, pro ostatní: !command
 - ↳ local change directory
- ↳ přenos souborů: get, put, mget, mput
 - ↳ stejní
 - ↳ nahraj
 - ↳ multiple → pro více souborů
- ↳ typ přenosu souborů: ascii, binary
- ! CRLF → !
 - ↳ pro rozlišení přenosů různými OS → ascii → textový, binary → binární
- ↳ pomůcky: prompt, hash, status, passive ...
 - ↳ pro zobrazení, co se děje
 - ↳ aktivní stav FTP klienta
 - ↳ server připojuje ke klientovi → passive off a neoff

LF → UNIX

CR → Macintosh

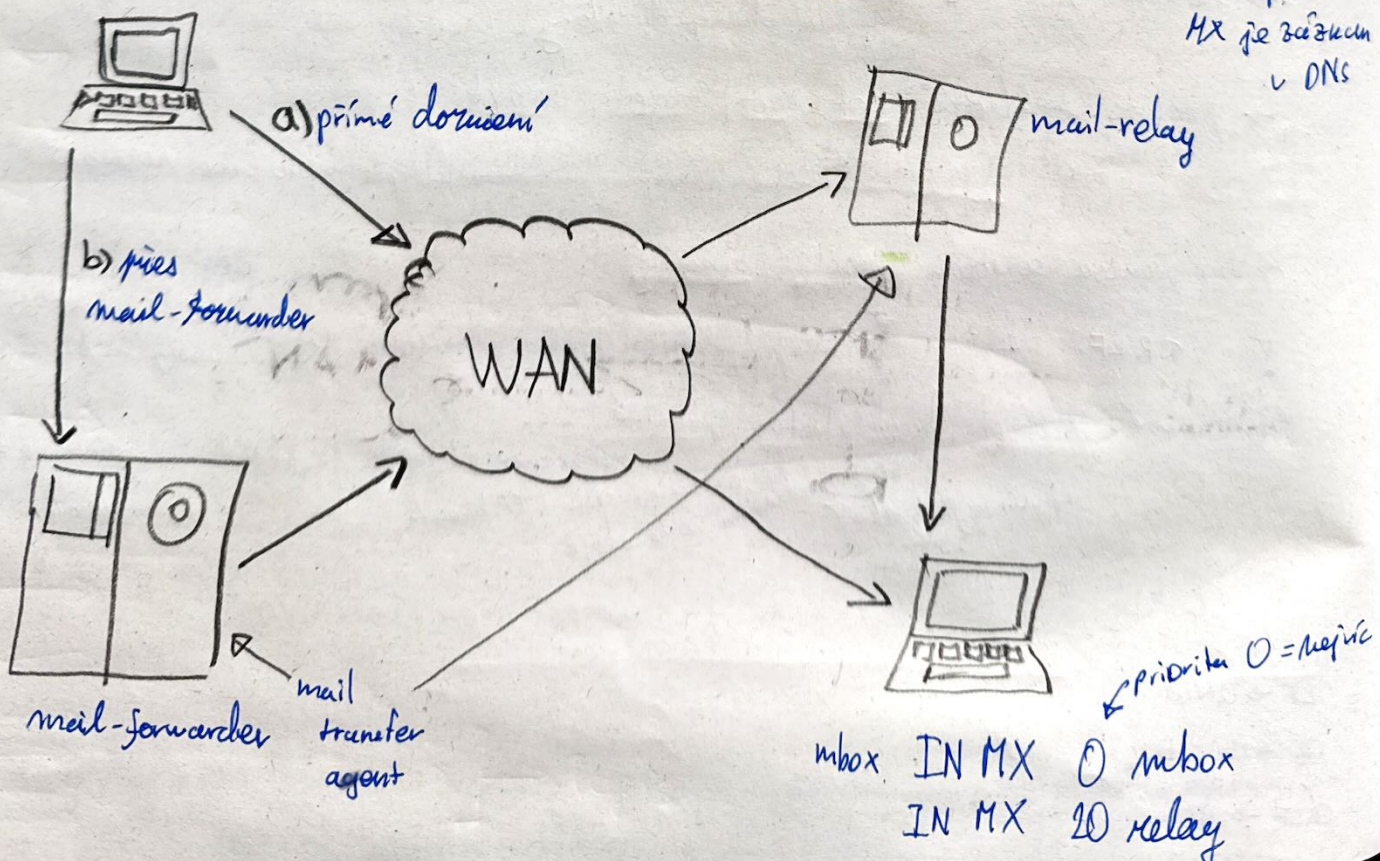
CRLF → Windows

Elektronická pošta

- ↳ SMTP protokol 25 → textový protokol na TCP
- ↳ obecná služba, co existuje i mimo internet
- ↳ off-line předávání krátkých zpráv i souborů → do 64kB
- ↳ diskusní kluby (mailing-listy, konference)
- ↳ off-line informační služby → mailum se posílalo: chi vědět, kde je senhle soubor a po dlouhé době přišla odpověď
- ↳ RFC 821 → formát protokolu, pak 2821, 5321
- ↳ RFC 822 → formát zprávy, pak 2822, 5322 } dnes platí
- ↳ mailová adresa:
 - login @ počítač → forst @ ms.ms.mff.cuni.cz
 - alias @ doména → Libor.forst @ cuni.cz

Příjem a odeslání pošty v SMTP

MX - definuje na jakém stroji má mail přijít
MX je začátek v DNS



a) přímé doručení

- ↳ mezi klientským počítačem a cílovým serverem (dávčím počítačem) nejsou přechody a pomocí SMTP přikročí se mail přímě
- ↳ dnes není časté

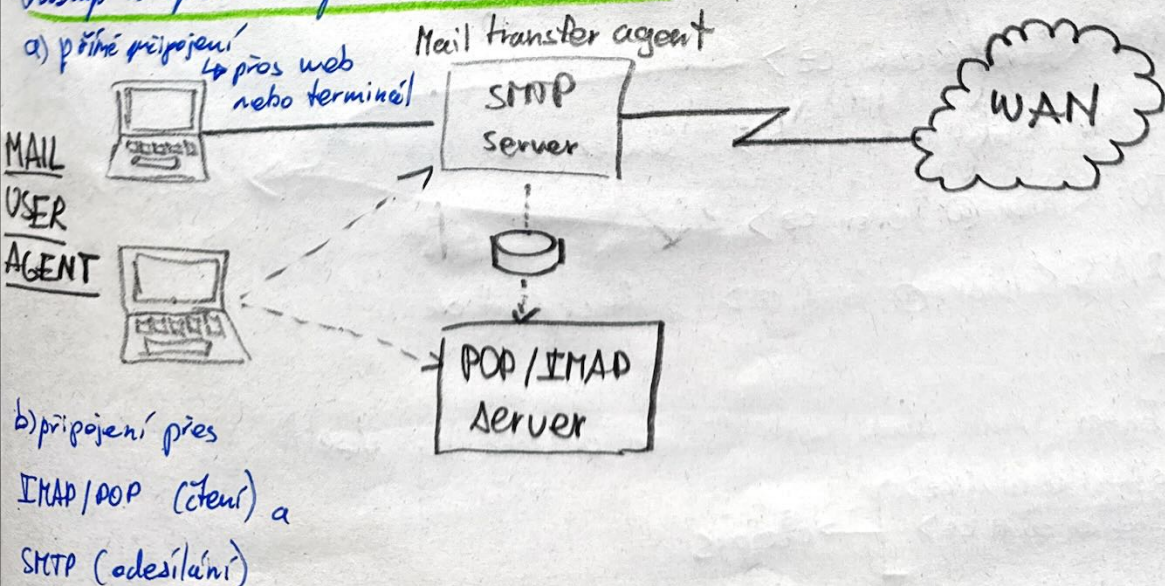
b) přes mail-forwarder

- ↳ na lokální síti se mail posle na mail forwarder, který mu dá náhod to poslat dle čísla hned nebo se spojení a pak to přeposílá nebo rovnou posle přes WAN a zkusí to poslat na mloz cílového serveru a pokud to nepůjde, tak to posle na relay v síti, který to odešle na cílový server
- ↳ MTA (mail transfer agent) je mail-forwarder a mail-relay
 - ↳ dočasné odesílající MTA jako klient a přijímající MTA jako cílový server

čtení a odesílání pro větší soubory

- ↳ celá cesta mailu se dá vyřít z hlavníky

Přístup k poště z pohledu uživatele



a) přímé připojení k SMTP serveru

↳ přes web nebo terminál (SSH)

↳ přímý přístup k mailboxu a lokální službě MTA, takže rovnou zašleme do odesílací správy do fronty MTA

b) připojení přes POP/IMAP (čtení) a SMTP (odesílání)

↳ POP/IMAP → poštovní protokoly, co slouží e-mailům k SMTP serveru

↳ na odesílání není potřeba autentikace, na čtení ano

↳ open-relay → člověk se připojí zvenku na server a snaží se poslat dopis ven, nedovolený → kvůli spamu

↳ dovolený je: - z lokální sítě ven

- z venku nemá SMTP spojení (bez autentikace není email a heslo) a poslat email dovnitř sítě

Ukážka SMTP protokolu

↳ textový protokol

← 220 alfik.ms.mff.cuni.cz ESMTP Sendmail...

→ HELO betynka

← 250 alfik Hello betynka, pleased to meet you

→ MAIL FROM: <forst@cuni.cz>

← 250 2.1.0 <forst@cuni.cz>... Sender ok

→ RCPT TO: <libor@forst.cz>

← 250 2.1.5 <libor@forst.cz>... Recipient ok

→ DATA

← 354 Enter mail, end with "." on a line by itself

→ From <forst@cuni.cz>

→ To <libor@cuni.cz>

→ ...

} → dopis

← více vždy rovnou napsat 500 a odmluvit spojení

→ obálka

← 250 2.0.0 49869FXT Message accepted for delivery směřo odmitnut, napiš. vir, moc velký atd.
 → QUIT
 ← 221 2.0.0 atk closing connection

- jedním převeze dopis, tak je server zodpovědný za doručení dopisu a ne klient, který obdrží číslo, pod kterým si ho server uloží
- sender a recipient se můžou v hlavičce (obálce) a dopise lišit, pokud nevíš odesílatel
- v hlavičce jsou informace o cestě, ale je třeba SMTP server o to přičíst
- MAIL FROM → pak špičaté zářezky, to samé u RCPT TO, těch může být libovolně mnoho a server akceptuje každého zvlášť
- Delivery Status Notification (DSN) → pokud se nepodařilo odeslat, špičkaty → může přijít bez kontroly více, ale dostane odesílatel
- DATA → pak dopis uložený někde, jinak lze poslat 1 řádek a ještě se uzavře, pokud je mu třeba jen CR LF řádek

Elektronický dopis podle RFC 822

Received: from atk.ms.mff.cuni.cz

by kelyuka.ms.mff.cuni.cz ← kdo napsal

Date: Thu, 16 Nov 1995 00:54:31 +0100

To: student1@ms.mff.cuni.cz

MIME-version: 1.0 ← pro posílání souborů

Content-Type: multipart/mixed; boundary = "-XXX-"

→ MIME
-XXX-

Content-Type: text/plain; charset=Windows-1250

Content-Transfer-Encoding: 8bit

Čau Petře!

...
-XXX-

zupomněl

From: Libor Forst <forst@uni.cz>

Subject: Test pošty

Cc: student2@ms.mff.cuni.cz

hlavička → struktura

jen 7 bitů z ASCII 0x00-0x7F

→ pro informaci a pro mailové programy

→ pravidla řádků

obsah dopisu

→ když tak jen 7 bitů, ale v SMTP protokol se mohli domluvit na 8bit kódování

potřeba domluvy

Hlavní části dopisu

Date → datum posílení dopisu v americkém (penném) formátu

From → název autorů dopisu, napsané: „komentář <adresa>“
• „(komentář) adresa“

→ ve spamu bývá více adres → protože si myslí, že to projde
pokud tam bude chvilkyhodná adresa, a ta stejná adresa v From
i To

Sender → kdo dopis odeslal, jen jedna adresa

Reply-To → kam má od příjemce přijít odpověď

To → adresáti

Cc → seznam adresátů, komu je poslána kopie dopisu → Carbon copy

Bcc → přidá se pouze do obálky, ale ne do textu, tak že ostatní
nevědí, že tyto adresáti obdrželi dopis → Blind carbon copy

Message-ID → vytvoří odesílatel program a příjemce použije při odpovědi,
aby se dokázaly maily sledovat

Subject → stručný obsah dopisu, pozor jen v 7-bit ASCII jako
rychlý klíč

Received → doplní MTA ^{uzel}, pokud dopis přijme a odesle dál
→ je tam název MTA, čas, lokální identifikátor ad.
pro složení
dopisu

Soubory a diakritika v Unixu

↳ kóčování pomocí UUCODE (≠ UUCP = „unix-to-unix copy“)

↳ nebo serku

↳ předávací před interakcí

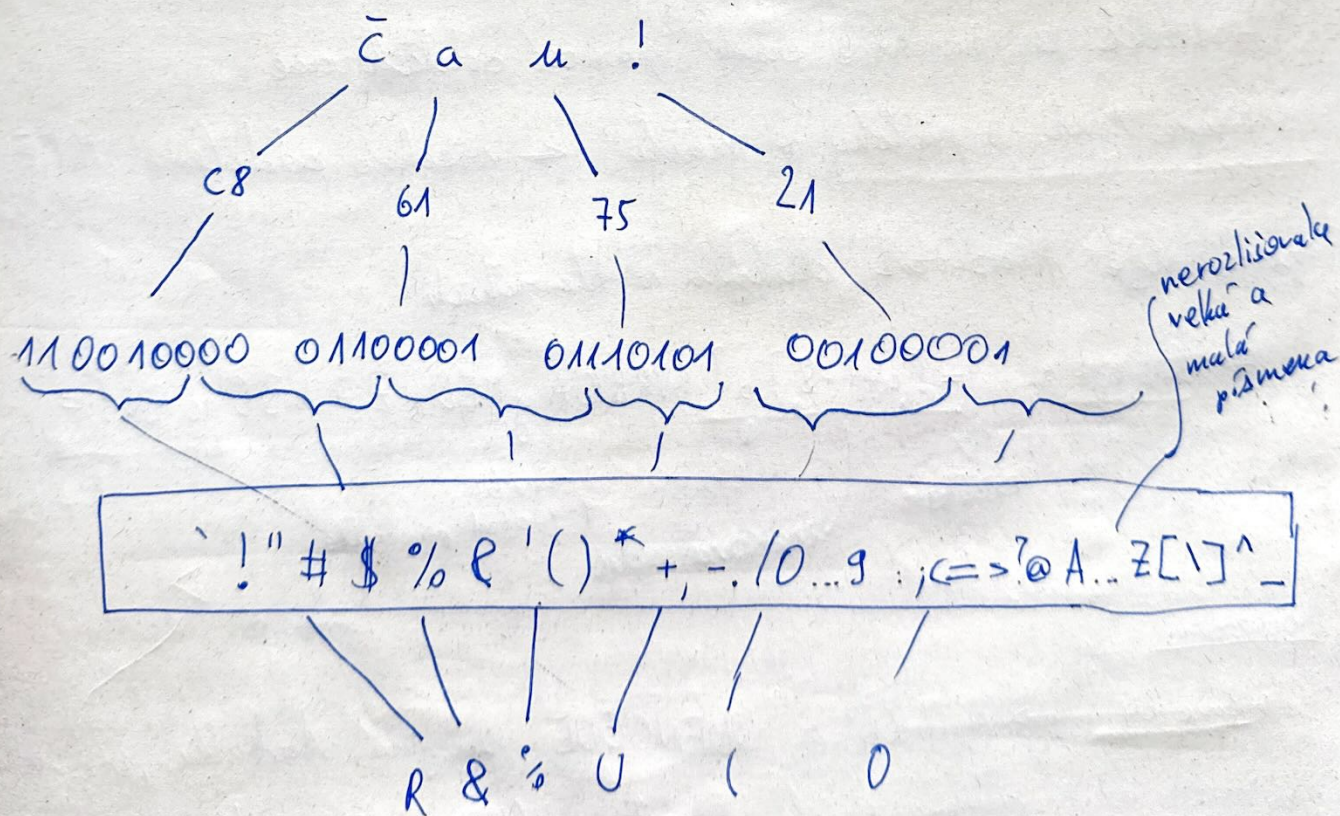
↳ převod binárního souboru do textu ASCII 7-bit

↳ vezmou se 3 bajty souboru (24 bitů) → rozdělí se na 4 skupiny po 6 bitech, které se převedou na čtyři licházní znaky

↳ soubor začíná v textu dopisem → begin a končí → end

↳ také se musel prohledávat celý text dopisu, aby se našel soubor

↳ zakódovaný soubor se zvětšil o 33 %



Multipurpose Internet Mail Extension (MIME)

Etika poštovního styku → RFC 1855 (Netiquette Guidelines)

- ↳ přičíst všechny maily, než odpovíte ^{Net}
- ↳ neodpovídat hned pokud je vCc
- ↳ nechat příjemci čas na odpověď, ale podějit se připomenout
- ↳ odpovídat rychle jako potvrzení
- ↳ správná volba subjektu
- ↳ kontrolovat adresáty
- ↳ nezachovávat text původního textu v odpovědi
- ↳ respektovat ©
- ↳ soubojy zvážit, zda poslat, raději na dropbox
- ↳ podepsat se

Bezpečnost pošty (uživatel)

- ↳ dopis je otevřená zpráva, proto je potřeba šifrovat obsah dopisu → mpi. Pretty Good Privacy (PGP)
 - ↳ program - šifrování veřejným klíčem a odšifrování ^{privátním} symetrickým klíčem
- ↳ někdy není jistý odesílatel ani údaje v obálce a textu
 - ↳ protože díky otevřenému protokolu tam může kdokoliv dopsat obsah
- ↳ částečné řešení: Sender Policy Framework - data same name
 - ↳ poštovní server pošle DSN správci uvedenému odesílateli, vlastně kontroluje které servery smí odesílat za které domény
- ↳ řešení: systém výzva/odpověď → takže doménu na výzvu, že dopis odeslal
 - elektronický podpis → pomocí šifrování

Bezpečnost pošty (client, server)

- ↳ open-relay → správny server jen odesílat maily lokálních uživatelů komunikativ a mailů odůhodnotiv jen lokálním uživatelem
 - jinak ostatní servery rablohuji komunikaci
 - odůhodnotiv pošle kaniholiv ⇒ open-relay

- ↳ MSA (mail submission server) → může požadovat při povolání vložením mailu autentikaci pomocí na portu 587 ← ESMTP pomocí příkazu AUTH
 - ↑
z SASL

- ↳ STARTTLS → příkaz v ESMTP, který zahájí SSL/TLS spojení mezi dvěma servery pro šifrované spojení, jinak je šifrování spíš problémem uživatelů

Ochrana proti spamu

- spam → inzerce nebo obtěžování lidí
- grey-listing → na první pokus odpoví 450, neúspěšně a pokud následný dopis, tak klient se znovu pokusí o doručení
- se slajunim tripletem → a pak email dojde, pokusí 250, třeba po 15-ti minutách
- <client, sender, recipient> → pak uloženo v paměti ne měsíc, bez zpoždění

- ↳ Sender Policy Framework → které servery mají právo odesílat pomocí kterých domén
 - problém s forwardem někam jinam
 - kanišen

- ↳ Domain Keys Identified Mail (DKIM) → odesílatel stroj podepíše dopis a přijímající server to kontroluje
 - forward není problém, protože každé MTA může mít vlastní klíč

Antispamy → algoritmy, co se dohadují pomocí heuristiky, zda
daný algoritmus je spam
→ false positive

Úkoly:

- 1) Jaká jsou nejčastější bezpečnostní rizika FTP?
- 2) Co je problémem při otevírání dodatečných
dalších kanálů?
- 3) Vysvětlete pojmy MTA a MX a jejich roli v přenosu pošty.
- 4) Co je „obálka“ a jak souvisí s obsahem klaviček dopisu?
- 5) Jak lze používat non-ASCII znaky v klavičkách a těle dopisu?
- 6) Jak se liší UUENCODE, Base64 a Quoted-Printable?
- 7) V čem spočívá problém open-relay serverů?
- 8) Vysvětlete podstatu grey-listingu a DKIM.

5. MODINA

Post office protocol

problémy

↳ odesílaní hesel, pro šifrovanou autentikaci APOP

↳ dopisy je třeba stahovat celé, POP - stáhne jen hlavičky, ale třeba implementované

↳ omezení možnosti práce s mailboxem

↳ nahrazení se IMAP protokolem

↳ aktuální verze 3

Internet Message Access Protocol (IMAP)

↳ modernější než POP

↳ port 143

↳ výhody:

↳ zabudované šifrování → přes port 993, nebo přes přikaz STARTTLS

↳ vyhledávání v dopisech

↳ podpora stahovat jen hlavičky

↳ používá chci hodně aplikací

IMAP4 protokol

↳ znakování (tagování) příkazů → lze tedy poslat více příkazů najednou

→ 1 LOGIN first heslo

← 1 OK User authenticated

→ 2 LIST "" "*"

← * LIST (Has No Children) "/" "INBOX"

← 2 OK LIST COMPLETED

... atd.

↳ musí mít SMTP spojení pro odesílání

Princip distribučované databáze

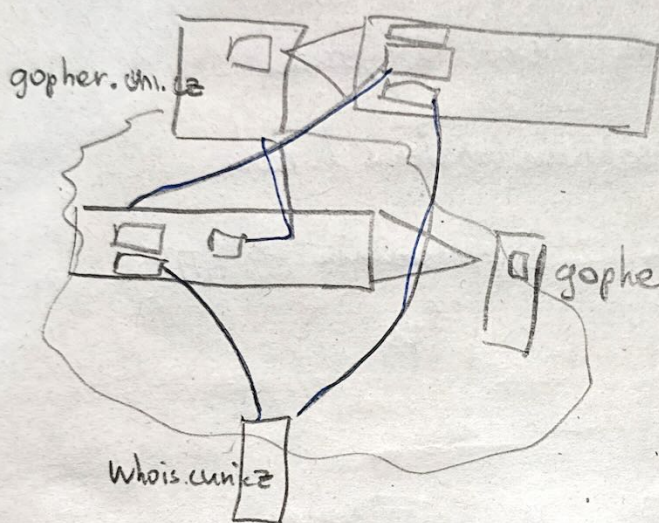
↳ Gopher

↳ databáze informací, které byly navzájem provázány a na různých serverech → vlastně web

↳ pouze jen text a ne obrázky → tedy bylo potřeba ho stáhnout

nebylo oděť, jestli se přesunul na jiný server

pomocí formuláře



czech educational and scientific network

↳ pár let po gopheru se objevil web

Hypertext

↳ opuch gopheru provided místo textové, tak hypertextovou databázi

↳ 1945 - první myšlenka

↳ text odkazy na vysvětlující / příbuzný text

↳ 1965 - rozšíření

↳ o intertextové informace → obrázky, zvuk, video

↳ 1989 - prakticky

↳ WWW z Cernu

World Wide Web

↳ distribuovaná hypertextová databáze

↳ základní jednotka je hypertextová stránka (dokument)

↓
nosiť na začiatok klientom (statické)

on demand stránka vygeneruje URL požiadavky (dynamické)

↳ jazyk HTML

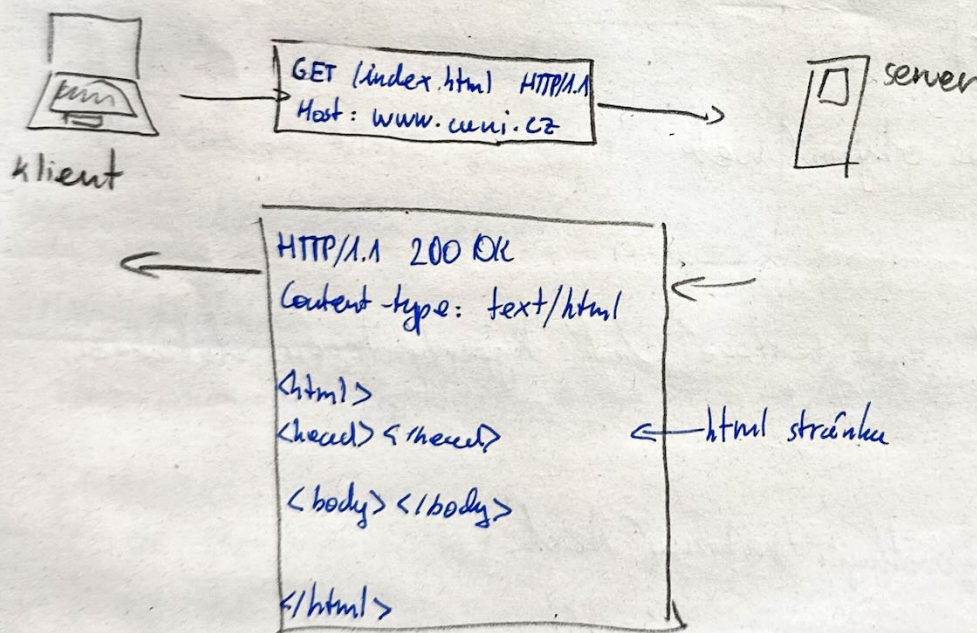
↳ popisuje obsah - stránky, text a foruny

↳ to, čo je potrebné riešiť riešenie klienta

↳ prenos stránok pomocou HTTP se šifrovaním → HTTPS

↑ TLS

Protokol HTTP (ukážka)



Hypertext Transfer Protocol v.1

↳ dnes verze 1.1, port 80, textový protokol

↳ formát zpráv:

- úvodní řádka → požadavek a pak odpověď

- doplňující (hlavičky) → ten Content-type

→ kódování, jazyk, stáří (požadavek)

→ typ, expirace, kódování (odpověď)

- tento dokumentu - voliteľné

kódy odpovedí:

1xx → informácia (prípadne prijat k spracovaniu)

2xx → úspech

3xx → ďalšia info od klienta potrebná (presmerovanie)

4xx → chyba na strane klienta

5xx → chyba na strane serveru

Metody HTTP

METODA	TĚLO POŽADAVKU	TĚLO ODPOVĚDI	nemění stránku opakované použití má stejný efekt	
			Bezpečná	Idempotentní
GET	—	dokument	ano	ano
HEAD	—	—	ano	ano
POST	parametry	dokument	ne	ne
PUT	dokument	—	ne	ano
DELETE	—	výsledek	ne	ano
CONNECT	←	tunnel	→	

GET - dostanu html dokument, v případě chyby také stránku html s chybou

HEAD - dostanu hlavičky, jako velikost stránky, název, jazyky

POST - pošlu o dokument, ale pošlu data, html formuláře → GET/POST ^{skládá v query URL}

PUT - pošlu dokument, nešíkám si stránku, měním server, ale efekt by měl být stejný

DELETE - smažu dokument, dostanu zpět zprávu

CONNECT - vytvořit spojení v jiném protokolu → bezpečnostní riziko

Vlastnosti HTTP v.1

- ↳ na jeden požadavek, jeden dokument (stránka, obrázek)
- ↳ 1 GET → 1 stránka a 3 obrázky
 - ↳ na každý obrázek nový GET
- ↳ v HTTP verzi 1.1 → vytvořeno persistentní spojení, kde lze posílat více požadavků
 - občas otevře více kanálů (klienti), aby mohli poslat více požadavků GET najednou → mají ty obrázky
- ↳ komunikace je bezstavová → ty čtyři GETy jsou nezávislé operace
 - ↳ proto si servery ukládají cookies ← nenesou kód, zápis hesla → aby to vypadalo stavově:
 - ↳ server vytvoří identifikaci a uloží si data o návštěvě a poše je do hlavičky
 - ↳ při každém požadavku ukládá do hlaviček
 - ↳ pro shromažďování informací o návštěvě
 - ↳ bezpečnostní riziko uchování → cílená reklama
 - informace riskují jiná osoba

HyperText Transfer Protocol v.2

→ pomalu ne nej přetváří

↳ úplně nový protokol, změnil se textového na binární → ono je to vlastně jedno, protože se to stejně rozšiřuje

↳ motivace: větší propustnost → prostě rychlejší načítání

↳ metody:

↳ server posílá více, než klient požádal, protože očekává, že

o to stejně bude řídit → mapy, obrázky

↳ prioritizované streamy v rámci jednoho TCP, aby se neblokovaly

↳ komprimace hlaviček → při požadavcích se stejně nebudou změnit, takže se neposílají pořád

↳ novinné diprování neprošlo

Jazyk HTML

↳ značkovací jazyk

↳ byla verze 5, odstavce, adresa, tučně

↳ značky → strukturační, sémantické, formátovací

str. 89-98 jsou HTML stránky v prezentaci

Dynamické stránky (server)

↳ dynamika řízení na serveru, ne u klienta

→ ne klientovi neběží kód

Dynamické stránky (klient)

↳ přes skripty, v jave → ten převede na mariohód a podle klienta to interpretuje jazyk, co má na počítači, takže knihovny pro každý jazyk na interpretaci → teď javascript

Telnet

- ↳ protokol pro vzdálené přihlášení k počítači, port 23
- ↳ Telecommunication Network, RFC 57, dnes už RFC 2946
- ↳ přenáší přes síťový virtuální terminál (NVT) → emuluje terminál
- ↳ přenáší oběma směry znaky a příkazy
- ↳ musí se domluvit na posílání → kdo udělá co
- ↳ ovládnutí přenosu dat → řeší už RFC 2946, dnes ale raději SSH
- ↳ Dnes využít:
 - ↳ ladění jiných protokolů
 - ↳ přístup na síťové zařízení v rámci LAN → netvoří odposlouchávání hesel
- ↳ problém s terminálem → nemá přiznání příkaz nebo odpověď
- ↳ protože někdo musí zohrnout znak pokud se stiskne klávesa echo
- ↳ aby mi se se zachytil → DO ECHO / WILL ECHO
 - ↳ pokud je ten server zapomeno, že to už poslal

Secure shell (SSH)

- ↳ bezpečná náhrada pro přihlášení ke vzdálenému počítači nebo přenos souborů
- ↳ klient důkladně ověřuje server, pokud ale po měsících přihlázení रहे मैं नया की → zpozornit
- ↳ komunikace je šifrována
- ↳ port 22, aktuální verze 2, → RFC 4250-4254
- ↳ SSH v2
 - ↳ více kanálů paralelně zabezpečených
 - ↳ tunelování → jiný provoz jiným zabezpečeným kanálem
 - ↳ souborový systém → SSHFS
- ↳ putty, WinSCP → SSH aplikace Windows, na Unixu → ssh a scp příkazy

Bezpečnosť SSH

↳ klient overuje server

- ↳ kontroluje kľúč (kontroluje vzťah)
- ↳ certifikátom (prieš autoritu)

↳ server overuje klienta

- ↳ pomocou hesla
- ↳ pomocou výzvy a odpovedí (OTP)
- ↳ pomocou kľúča → server pošle zašifrovanou správu verejným kľúčom a užívateľ pošle plain text

↳ Strategie používania kľúčov

- ↳ overiť kľúč serveru, man-in-the-middle
- ↳ zakaždým si kľúč ľah, že pokiaľ niekto zistí mój stroj, tak nemôžem prísť k serveru, pretože musím heslo
- ↳ reciprocita ($A \rightarrow B$, $B \rightarrow A$) → internetové čerwi, kedy chcú verejný kľúč a prihlásenie → väčšinou nejde, ale môže sa stať, že niekto nemá prihlásenie rovnaké i naopak

Noise over IP

- ↳ nemá protokol, ale obecné jazykovo nástroj, čo preverí hlas přes TCP/IP, nevieha Type přes HTTP

↳ protokoly patria do standardek 11.823

↳ ďalšie (jiné, nekompatibilné):

- SIP standard

- proprietárny

→ používa sa VoIP

↳ problémy:

- ↳ digitalizácia hlasu, nahrať partnera, možnosti (obom) vice rozlíšení a propojení s bežnou telefonnou sieťou - teda používať čísla tel.

H.323

↖ založeno na ASN.1

↳ šetrný kardinál mýslím → binární protokoly, po káblech

↳ organizace ITU - international telegraph union

↳ další protokoly: (nepotřebá)

- H.225 /RAS (Registration /Admission /Status) pro vyhledávání partnera

- Q.931 (síťová vrstva ISDN) → nastavuje spojení

- H.215 → řízení hovoru

- RTP (Realtime transfer protocol) → pro přenos multimediálních dat

- RTCP ^{←+control} → zabezpečuje { přenáší vlastní data

↳ dnes nahrazuje SIP
↳ lepší pro VoIP

ASN.1 → Abstract Syntax Notation 1

↳ pro popis datové struktury nebo obsah dat díky mnemotickým identifikátorům

↳ zapisuje do kolik bitů kolik potřebá → přechází bit co signalizuje kolik bude bitů

→ z 80. let = tragédie

↳ používá v H.323 a X.509

Session Initiation Protocol (SIP)

↳ jednoucháň než H.323

↳ nahrazuje navazování spojení mezi zařízeními, dohodou o vlastnostech

↳ textový protokol navořel od binárního → podobá se HTTP

↳ port 5060 pro UDP i TCP

↳ používá RTP / RTCP pro přenos dat, SIP jen navazuje spojení

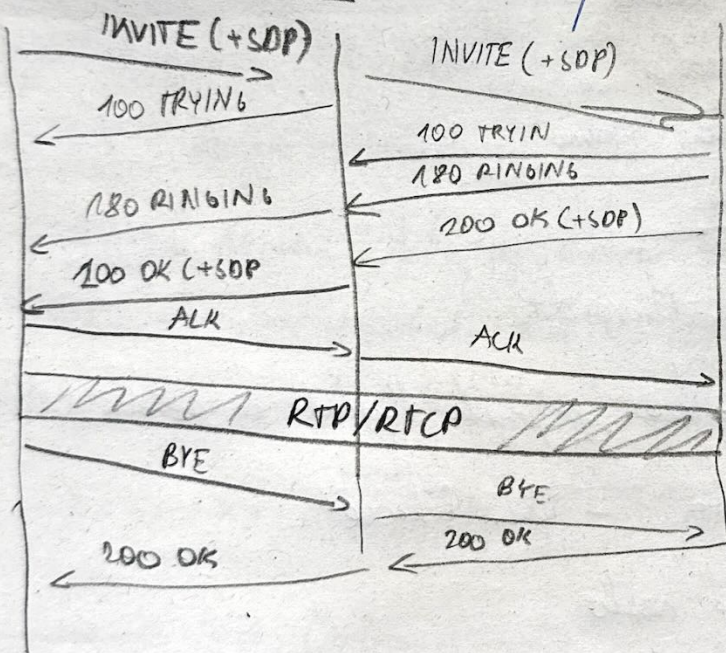
↳ přes běžnou telefonní síť přes registrátora

proxy ← zajímavost, že tam je, kobříá hčšná

session description protocol

SIP session

telefon



telefon



- začne INVITE + SDP

↳ hledá na proxy, která řeší nalezení cíle

↓
přes ústředny a další proxy

↳ 180 ringing → end-to-end, proto prople
jinak 100 trying je next-to-node

↳ 200 OK + SDP

↳ proxy zase upravit

↳ pak ACK

aby vše fungovalo

↓
takle proxy upraví pomocí NAT

Seznam

1) K čemu slouží IMAP protokol?

2) K čemu slouží HTML?

3) Popište, co se očekává v HTTP 1.1 protokolu, pokud klient požádá o stránku se dvěma vloženými obrázky?

4) Stručně popište účel a vlastnosti nejznámějších metad HTTP.

5) K čemu slouží cookies a jaké představují rizika?

6) Porovnejte protokoly telnet a SSH.

7) Porovnejte protokoly H.323 a SIP

6. HODINA

sdílení systémů souborů

↳ připojit celý disk jiného počítače a pracovat s ním jako lokální

↳ Network File System (NFS)

↳ mezi nejpopulárnější v UNIXu, norm 2049

↳ AFS - Andrew File System → náhrada

↳ server: cesta a odkaz na připojený disk

↳ vyvinut v Gen Microsystems → dnes IETF

↳ autentikaci přes Kerberos

↳ nad UDP ^{práci}, nad TCP zůstává

↳ kvůli rychlé reakci

↳ RPC volání, XOR - přesávání - jedno

↳ Server Message Block (SMB)

↳ vyvinuto IBM, převzal Microsoft

↳ Samba → open implementace, propojila Microsoft se systémem UNIXu

↳ UNC → \\server\cesta - identifikace

↳ autentikace → jméno a heslo

Network Time Protocol

↳ aby na všech počítačích v síti byl stejný čas

↳ důležitost:

↳ hlavně při sdílení souborů → timestampy souborů

↳ logy při řešení problémů sítě

↳ z stratum 0 (atomové hodiny)

↳ přebírá stratum 1 a tak dále, vždy přebírá od N-1 stratumu

↳ kvůli přesnosti a
rozptylení

↳ NTP funguje tak, že je tam balence mezi stratum servery a díky Marzullac algoritmu se dá spočítat interval

BOOTP a DHCP

↳ technické protokoly, které umožní klientovi v síti získat IP adresu

Bootstrap Protocol (BOOTP)

↳ RFC 951

↳ protože díky tomu bylo možné, že se hypotali počítače bez disků, proto nebylo možné si uložit konfiguraci a IP adresu → problém

↳ řešením byl BOOTP → počítač poslal do sítě broadcast, tedy všem svoji fyzickou (MAC) adresu síťové karty - tedy vyhledal v karte

↳ BOOTP server odpovídal s IP adresou, kterou může začít používat pomocí tabulky, kterou měl o síti

↳ BOOTP Forwarding → pokud není v síti ^{odděluje je router} BOOTP server, tak musí poslat do podsítě, kde je a pak zpět klientovi
→ normálně nedovoleno, aby se broadcast šířil po internetu

DHCP (Dynamic Host Configuration Protocol) - automatické alokace IP adres, v ^{restauraci} ^{mobile}

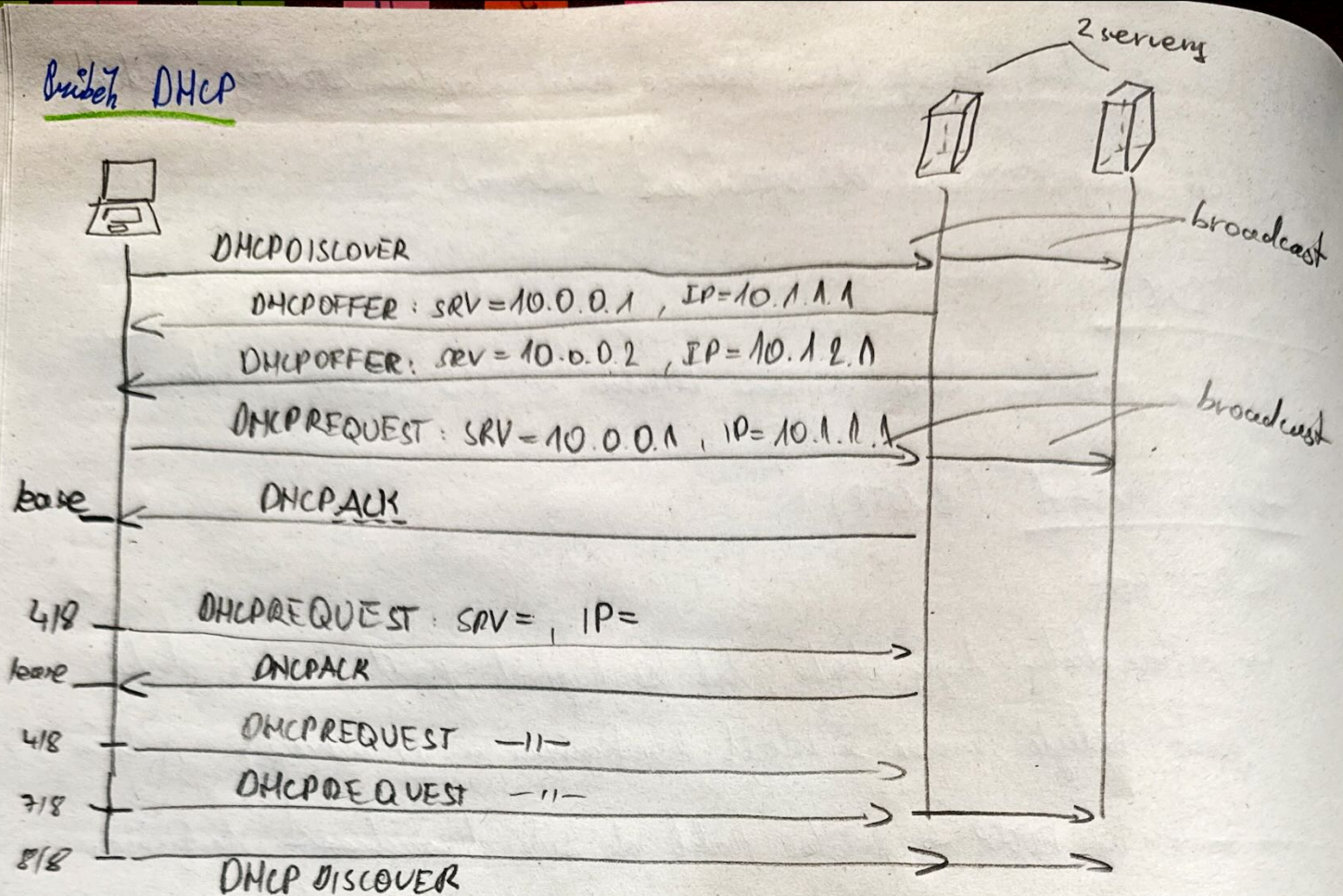
↳ stroj nedostane fixní adresu podle MAC adresy, protože jich bylo omezený počet a tedy se přiváděje dynamicky

↳ časové omezení pronájmu adres → lease-time, třeba 1h - 12h

↳ více serverů v síti, co spolupracují a klient si může vybrat podle služby

↳ opětově kompatibilní s BOOTP

Průběh DHCP



DHCPDISCOVER - broadcast začlost o IP adresu

DHCPOFFER - servery v síti pošlou nabídky

DHCPREQUEST - všem sděním, aby dal vědět, co si bere za IP adresu

DHCP ACK - potvrzení od serveru

DHCPREQUEST - v pokračování 418, pak 718, aby si prodloužil dobu používání
↓
tedy v broadcastem

Prezentací veslo (OSI 6)

↳ veslo, která řeší rozhodnutí dat pro přenos

↳ všeobecný model s kódováním → datové typy a datové struktury

↳ ASN.1 → v H.323

→ mohlo se být univerzální, ale bylo moc složité

↳ TCP/IP přeneslo dekodování na programátory do aplikací (aplikací veslo)

↳ problémy: prostě zavolání Network to host a host to network → jak je to realizováno v síti

↳ CR LF na konci řádek → $\Phi \times \Phi \Phi$ a $\Phi \times \Phi A$

↳ RE a LE → zapisování bytů

Relační vrstva (OSI 5)

- ↳ řízení dialogu mezi aplikacemi, nehlídá na to kterou aplikaci
- ↳ třeba RPC v NFS (zajímavost)
- ↳ TCP/IP zase převedlo rozdílnost do aplikační vrstvy (protokolů)
 - ↳ např. SMTP → ^{zpráva} po TCP ^{spojení} předá více mailů než jeden obsah
- SIP → 1 řídící spojení + 2 datové → video a audio

Transportní vrstva (OSI 4)

- ↳ se end-to-end spojení a přenos dat podle přecpanosti
 - ↑
potřebí doručení
→ TCP
 - UDP
↓
je-li odeslán potvrdit
- ↳ multiplexing
 - ↳ pomocí portů lze identifikovat protokol, službu
- ↳ segmentace dat → TCP
 - ↳ rozdělí na bloky, které opět složí pokud přesáhne velikost
- ↳ řídí tok → TCP
 - ↳ pro efektivnější přenos ovlivňuje, co se kdy odesle
 - ↳ stream může pozdržet o pomalejší přenos dat

Transportní vrstva v TCP/IP

TCP (Transmission Control Protocol)

- ↳ pro spojované aplikace
- ↳ klient navádí spojení a data lečou ve formě proudu → stream
- ↳ TCP řídí a zabezpečuje doručení → proto jsou aplikace jednodušší
- ↳ po odeslání musí čekat na potvrzení nebo než vyšší lineout, takže pomalejší než UDP

UDP (User Datagram Protocol)

↳ pro nespojovné služby

↳ např. obsluha mailu

↳ UDP je bezobslužný → řezy aplikace musí být šifrovat a řídit doručení a spolehlivost

↳ ale rychlejší, protože nemusí čekat na potvrzení

↳ - SCTP, DCCP, MPTCP - multipath TCP

↳ stream datagram congestion control protocol
control transmission protocol

Struktura UDP datagramu

zdrojový port	Cílový port
Velikost	kontrolní součet
Data	

Struktura TCP paketu

<u>Zdrojový port</u>		<u>Cílový port</u>	
<u>Sequence number</u>			
<u>Acknowledgement number</u>			
<u>data offset</u>		<u>Flags</u>	<u>Window</u>
<u>Kontrolní součet</u>		<u>Urgent</u>	
<u>Nastavení</u>			
<u>Data</u>			

data offset → pro identifikaci segmentů

→ pro identifikaci

→ pro potvrzení

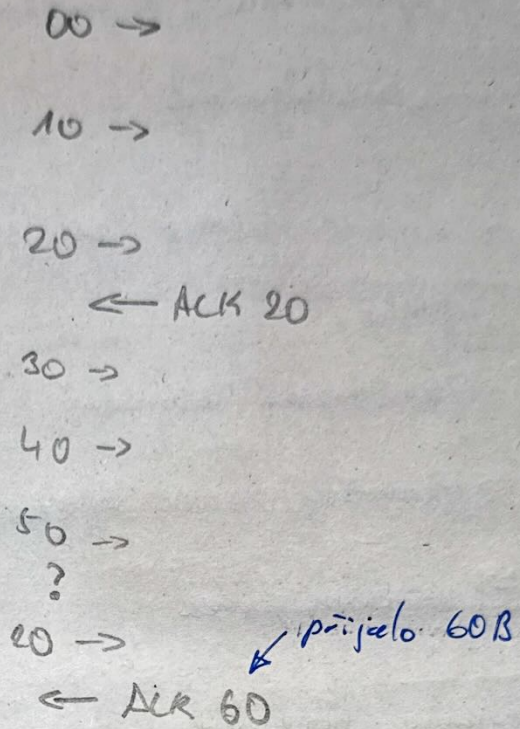
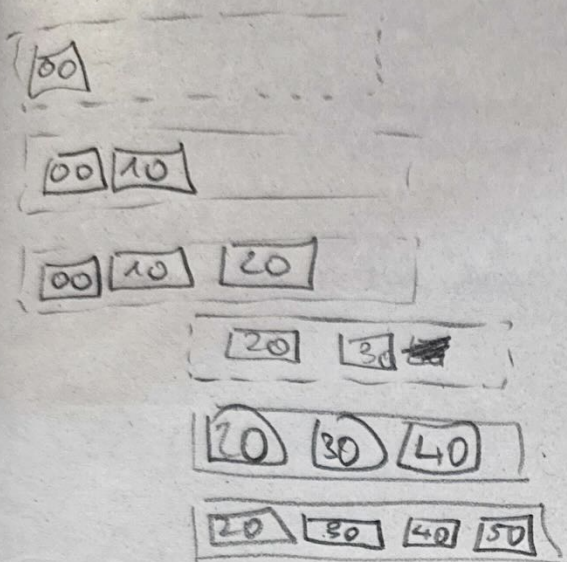
→ řízení rychlosti

→ důležitost

posun od začátku komunikace

potvrzuje kolik přijal a může posílat dál

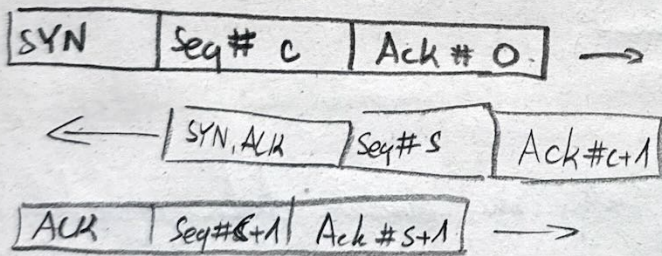
TCP okno



↳ pravidla se ohledně místa čekání na každý ACK pro každý paket

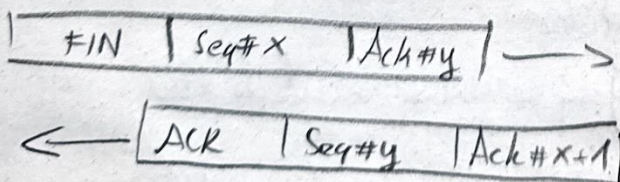
Zahájení a ukončení spojení

navázání → three-way-handshake



SYN → synchronizace
sequence number od nějakého
čísle → ne od nuly pro lepší
určení
↳ 3 pakety

uzavření → jednostranné



FIN → ukončení, že nebude posílat
data, ale může přijímat
→ např. GET a HTTP, uzavře spojení
a počká na film i guess

↳ protistrana poté provede totéž, takže jde o ukončení spojení, tak
ještě musí posílat ACK pakety protistranně, protože je to jen
half-closed

TCP přiznaky

SYN → synchronizace, s sequence number

ACK → acknowledgment

PSH → informuje příjemce, že už obdržel blok dat a může předat aplikaci

FIN → jednostranné ukončení

RST → okamžitý konec spojení

URG → urgentní paket

Výpis programu tcpdump → tedy, jak vypadá spojení TCP

↳ v prezentaci, not important?

Výpis existujících socketů

↳ pro zobrazení všech otevřených spojení → netstat

↳ jsou tam slovy - protože TCP je stavové

Šifrová vrstva (OSI 3)

↳ najít cestu v síti k počítači, aby mu předala data, co dostala z transportní vrstvy

adresace → adresa počítače adst.

zapouzdření (encapsulation) → data, adresy se musí vložit do PDV

↳ šifrování

routing (směrování) → hledání nejkratší cesty k cíli v síti

Forwarding (přeposílání) → přelání paketů dle k uzlům, aby se dostal k počítači

decapsulation → vyčtení dat, která se přelají transportní vrstvou

↳ protokoly: IPv4, IPv6, IPX, Appletalk

Internet protokol (IP)

- ↳ nespojovací služba → datagramy se doručují nezávisle
- ↳ nespolehlivá → spolehlivost řeší vyšší vrstvy
- ↳ nezavislá na způsobu přenosu (medium) → nemusí řešit konkrétní technologii

Adresy:

- ↳ každá s adresou sítě a každá s adresou uzlu
- ↳ IPv4 → 4 byty IPv6 → 16 bytů

Přidělování adres:

- ↳ IANA (Internet Assigned Numbers Authority), součást ICANN

internet corporation
for assigned numbers
and members

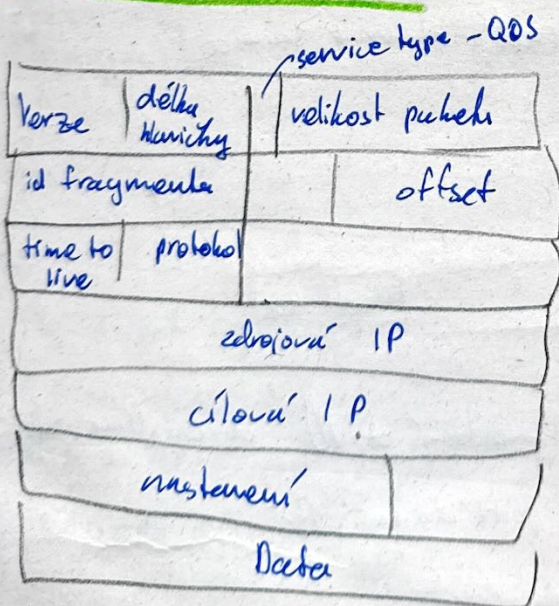
- ↳ rozchozíje správa sítě → privátní (speciální síť)
- veřejné (ISP)

↳ nad ISP právě IANA

- ↳ IANA má pod sebou 5x regionálních registrátorů, RIPE NCC

Struktura IPv4 diagramu

↳ každému pod ním ISP
dává blok adres

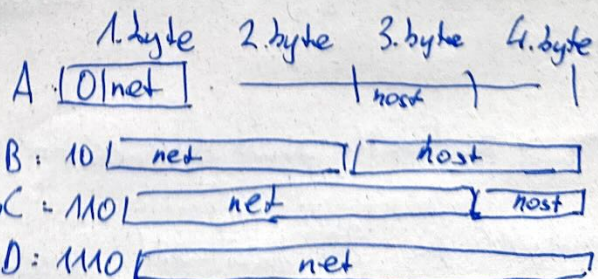


IPv4 adresy

- A 0 net host
- B 10 net host
- C 110
- D 1110
- E 1111

2 4 byty

multicast experimental



používá se s mnohým portem počítačů

E: 1111

Speciální IPv4 adresy

↳ 0.0.0.0

↳ používá jako zdrojová, když potřeba komunikovat, ale ještě neznám svoji adresu, třeba DHCP dotaz

↳ 127.0.0.1 (loopback)

↳ po mě, takže komunikace jako na síti ale jen v počítači: klíčová síťová karta

↳ addressite: <adresa sítě> <samé nuly>

↳ network broadcast: <adresa sítě> <samé jedničky> → všem uvnitř sítě

↳ limited broadcast: 255.255.255.255

↳ všem v této síti

↳ privátní adresy

10.0.0.0/8

172.16-31.0.0/16

192.168.*.0/24

} normálně nejsou ve veřejné síti

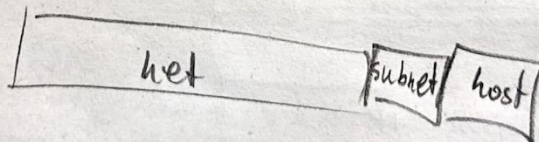
↳ link-local adresy → 169.254.1-254.0/16

↳ v rámci segmentu sítě, může přiznat stavie sama

↳ nete komunikovat mimo síť

Subnetting

↳ rozdělení sítě na podsítě



↳ pomocí síťové masky → 255.255.255.224 → bitový AND a zjistíme, zda patří do naší sítě

↳ classless → za limitu počet bytů, co tvoří síťovou část

↳ 6 podsítí po 30 počítačích, protože nedopoujíme all zeros a all ones

↳ VLSM → variable length subnet mask

↳ adres nepoužívá, protože nehomogenně privátních adres

↳ supernetting → merge → spojen sítě

Strana internetu

↳ propínají se směrovací tabulkami:

↳ sítě co spolu sousedily byly agregovány

↳ Classless Inter Domain Routing (CIDR) dle množství se maximálně přibližovat sítě

↳ používání privátních adres a NAT → řešilo vyčerpávání adresního prostoru

↳ došly adresy asi 2012, 2017

IPv6

↳ 16 bytů

Adresy

↳ unicastová → loopback :: 1/128

→ link-scope fe80::/10 → čistě link-local

→ unique-local fc00::/7 → čistě site-local, takové privátní adresy IPv4

↳ multicastová → ff00::/8 → různá broadcastových adres

↳ anycastová → unicastová adresa přiřazena více větám
→ např. senery po světě → nalezení nejbližšího

↳ tunelování IPv4 a IPv6

Ostatky:

1) Proč a jak se synchronizují čas na počítačích v síti?

2) K čemu slouží DHCP?

3) Jak se v TCP/IP řeší chybnější prezentace ústava?

4) rozdíl mezi TCP a UDP

5) pojmy potvrdění a TCP okno, jak se využívá?

6) three-way-handshake?

7) jak se využívá třídy IPv4, subnetting a CIDR?

8) Proč je v síti C pouze 254 adres pro počítače? → same nuly a same jediný, odpověď

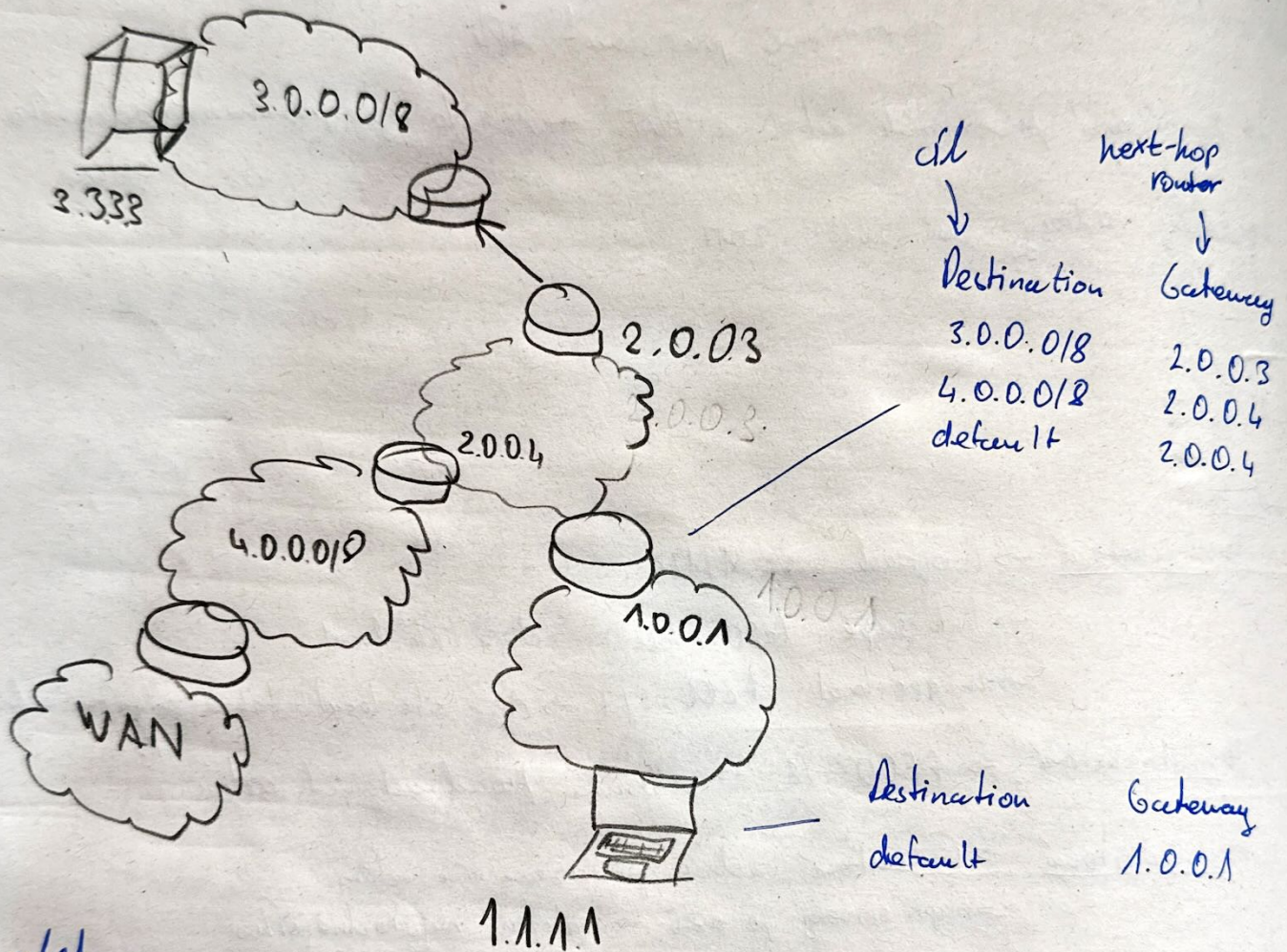
9) Jaké typy broadcastových adres známe?

7. HOPINA

Směrování (silnice) - vlastně pomocí síťové masky

↳ rozšiřuje se podle křížovatek

Směrování (síť)



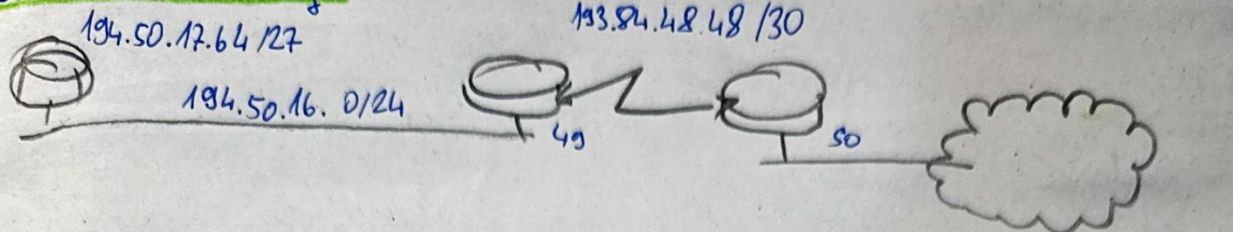
oček

↳ hledá next-hop router pomocí směrovacích tabulek

↳ pakel chce například na adresu 3.3.3.3 → proto vždy porovná prefix

s maskou 18 například a podle toho se vybere jiná default

Průběh směrovací tabulky



Destination	Mask	Gateway	direct
127.0.0.0	255.0.0.0	127.0.0.1	→ loopback adresa
194.50.16.0	255.255.255.0	194.50.16.71	→ adresa hrady C, na hlavní síti LAN
193.84.48.50	255.255.255.255	193.84.48.49	→ stroj připojený k ISP routeru, jediný v LAN
194.50.17.64	255.255.255.224	194.50.16.77	→ ukazuje na pod síť
default	0.0.0.0	193.84.48.50	→ router → ven

Principy směrování

→ kódová stránice v TCP/IP síti

→ ve směrovací tabulce :

- cíl (destination)
- maska
- gateway → přímý (direct) → nějaký jeden cíl
- nepřímý (indirect) → next-hop router

→ rozdělení:

implicitní - jsou po vytvoření sítě

explicitní - manuálně zadání

dynamické - vytvořené používáním sítě

Směrovací algoritmus

→ najdi vyhovující rozdělení → nemůže? → pokud už došlo, ale neprojde podobně

protože rozdělení bude obsahovat aspoň default

→ pak vybere destination s největší maskou

→ direct → podle přímého

→ indirect → podle next-hop router

} dočítat se na linkové úrovni

Konfigurace sítě

UNIX

- Adresa : ifconfig
- defaultní router: route add default router
- DHCP: dhclient

Windows

↳ nejde v nastavení, nebo ipconfig a route

Internet Control Message Protocol

↳ 3. vrstva síťové OSI → protokol ICMP → k posílání různých informací

↳ ping, echo, echo reply

↳ destination unreachable → nedostupný stroj, pro UDP důležitě

↳ time exceeded → time to live překročen

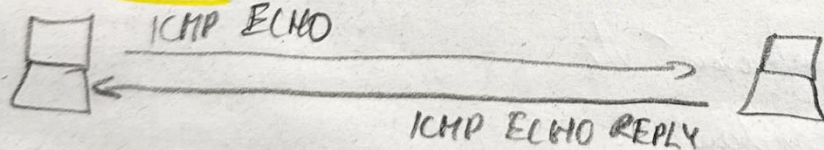
↳ posílání správy protokolu NDP → neighbor discovery protocol
→ zjistit stav okolní sítě

↳ source quench - zátěž o snížení rychlosti

Ping ↳ router solicitation - vyhledávání routerů

↳ redirect - k změně v routovací tabulce

↳ parameter problem - chyba v hlavici IP datagramu



↳ pro diagnostiku sítě - dostupnost uzlů

↳ ping → posílá periodický ICMP echo dolů nadolní aparát

↳ nezaručuje dostupnost služeb → jen síťovou vrstvu

↳ ping, pong neholikrát a pak se udělá statistikaidence

↳ nemusí být speciální software

Time to Live (TTL)

→ v DNS je TTL sekundy a znamená kolik sekund ještě zůstaná platit

↳ Time Exceeded

↳ není o case, měří se v hopech

↳ při dosažení 0 se pošle time exceeded

↳ kvůli dubbě v routovacích tabulkách → aby se nevytřásla konekce

↳ každý router snižuje TTL a tím mění obsah IP hlavičky

Diagnostika směrování

↳ troubleshooting směrovací tabulky netstat -r [n]
route print

↳ kontrola cesty → traceroute, tracert

↳ poše packet TTL 1 → dostane zpět první router

TTL 2 → dostane zpět druhý next hop router

...

→ poslední směrování je většinou problémem

Úplné řízení směrovacích tabulek

↳ cesty se nastaví při startu ↖ defaultní

↳ například DHCP → dostane router a nic jiného nepotřebuje

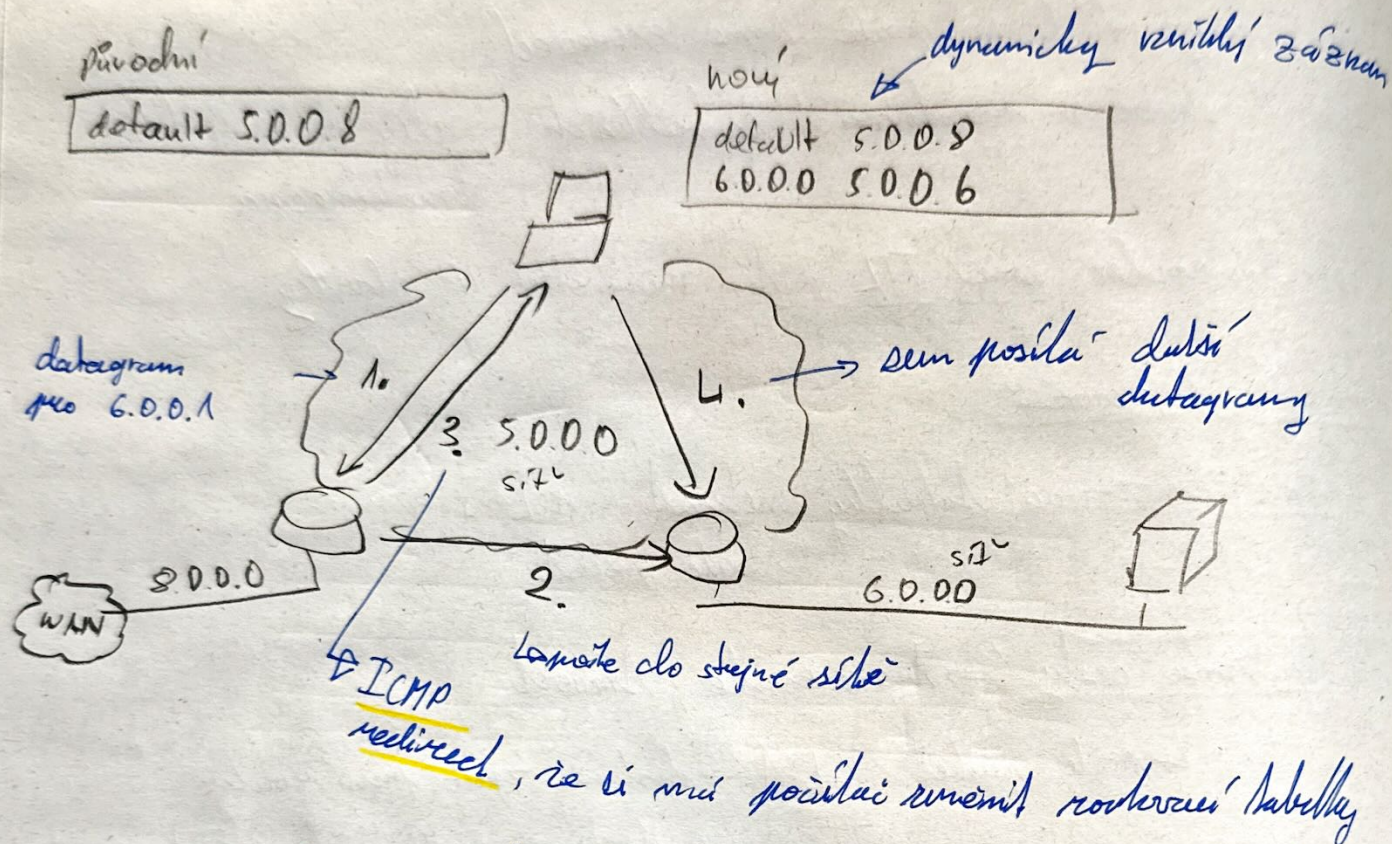
↳ ale nepoužitelné při eměních, subnettingem majte, takže většinou

↳ asi tak jen v malých sítích

Redirekce

↳ další ICMP paket

↳ porovná se k tomu, když zjistí, že routovací tabulky jsou špatné



↳ co když ale ICMP redirect je falešný? → pak je potřeba ho zaházet

Dynamické šíření směrovacích tabulek

↳ routery si vyhoví směrovací tabulky pomocí komunikace se sousedními routery a pomocí routovacích protokolů

↳ sít se dobře sama upravovat

↳ jednoduše se změni konfigurace a automaticky se aktualizuje

↳ ale citlivější na síťky

↳ protokoly: routed, gated, BIRD

↳ dělí se na-distance-vector protokoly (RIP) ^{např.}

- link-state protokoly (např. OSPF)

Distance vector protocols

↳ hledá pohled neexistuje něco bližší než dosavadní cesta, má tam měření i „vzdálenosti“ → počet hopů

↳ přehled informací soustředěný a ty si ho každý upraví atd.

↳ jednoduché

↳ pomalu reaguje na chyby, nerozhleduje vlastnosti linek,

↳ rychlost, cena, spolehlivost

oměření rozsah sítě, chyba ve výpočtu ovlivní celou síť

Routing Information Protocol (RIP)

↳ velmi starý distance vector protokol

↳ počet hopů omezen na 15 → proto vhodné pro jednoduchší sítě

↳ používá Bellman-Fordův algoritmus → ne rychlejší Dijkstra

↳ protože kopíruje počet ze směrovací tabulky, jen ho trochu upraví než aby spouštěl celý algoritmus znovu
↳ protože funguje díky zprávám od ostatních uzlů

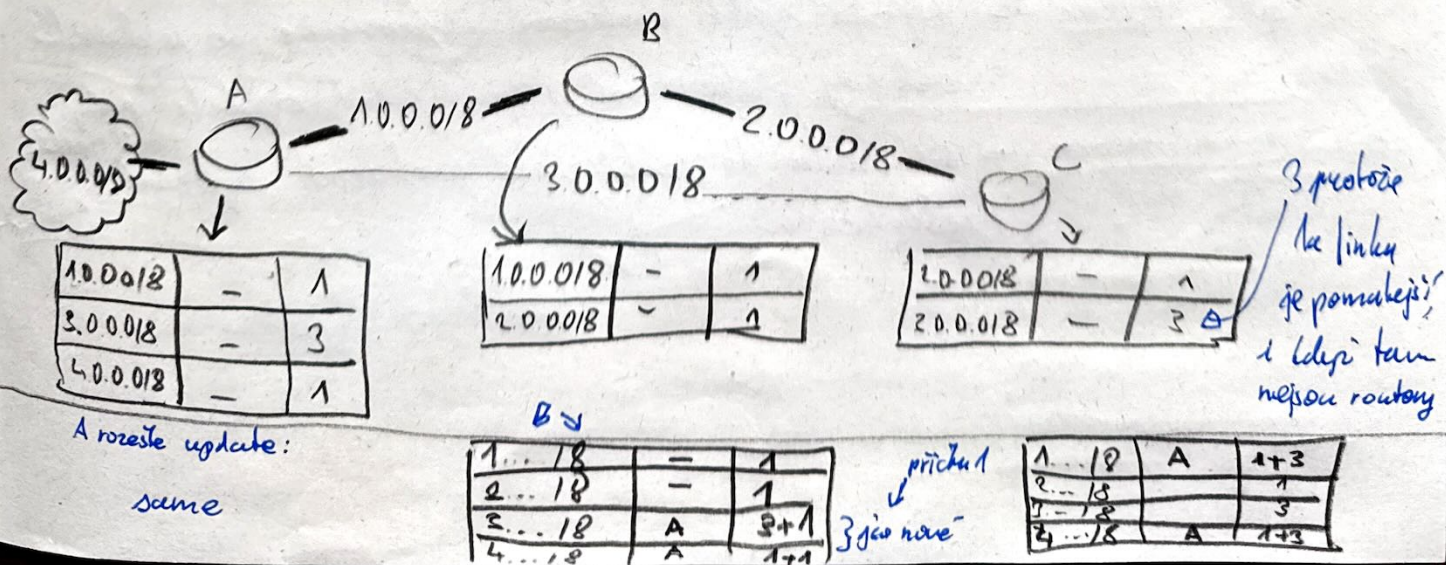
↳ verze 2

↳ má subnetting s proměnlivou maskou (VLSM)

↳ urychlení detekce chyb → triggered updates, split horizon, poison reverse

↳ funguje na všech systémech

Metrika a kvalita linek

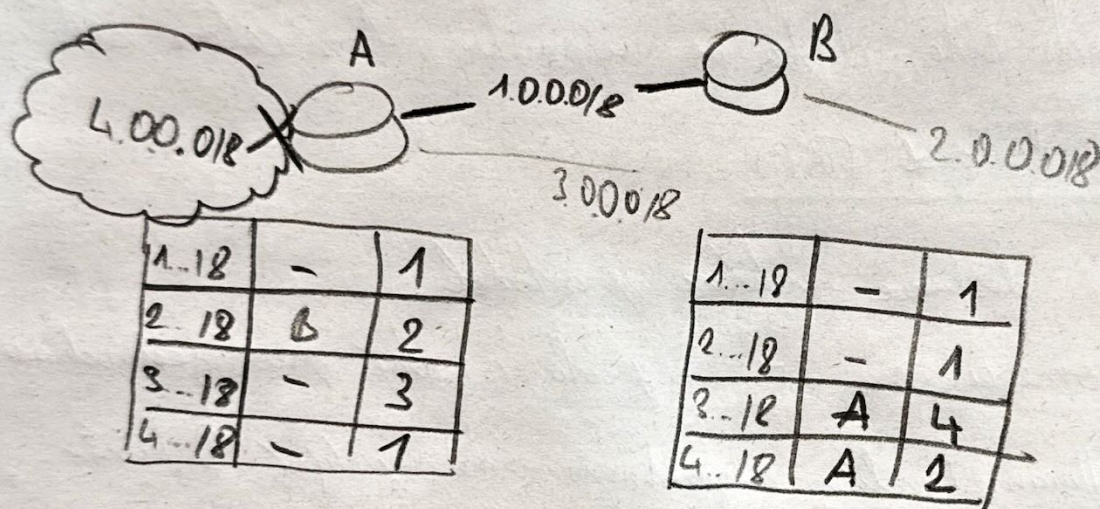


B rozšířte update

1...18	B	1+1
2...18	-	1
3...18	-	3
4...18	B	2+1

→ pokud mají lepší cestu, tak si aktualizují tabulku se vzdáleností

Counting to infinity



→ vypadne linka na 4.000/8 → tak se ochrání sama zpátky a v tabulce růstne 16, nebo-li ∞

→ 7x30 sec → proto je ∞ tak malý, protože to trvá půl minuty

→ řešeními tedy:

triggered updates → posílá update okamžitě

split horizon → neposílá update partnerovi informace, co se dává dál než, ale musí připravit další update zpátky

poison reverse → o metrikou 16 posílá data partnerovi, výhledem ^{zavědla} a ne ^{kapitál}

Link-state protocols

- ↳ každý router zná celou mapu sítě
- ↳ při updatu si sítě staví linky a každý router si aktualizuje svoji mapu sítě
- ↳ nevýhody: pomalejší a náročnější
- ↳ výhody: - chyba neovlivní ostatní
 - reagují dobře na změny sítě
 - nemusí si vyměňovat celé své databáze
 - rozdělit na menší podsítě

Open shortest path first (OSPF)

- ↳ link state protokol, nejrozšířenější
- ↳ používá Dijkstrův algoritmus na hledání nejkratší cesty
 - ↳ při průchodu grafu se přechází jen z rel. do sousedních vrcholů a každý max. holík má vrcholů
- ↳ rozkládá se na oblasti → aby se nemusel algoritmus počítat pro celou síť
- ↳ hierarchický model sítě:
 - ↳ area 0 → páteř
 - ↳ ostatní oblasti jsou připojeny na páteř
 - ↳ každý zná cestu k páteři a mapu své oblasti
- ↳ path cost
 - ↳ lze přizpůsobit, mění administrátor podle síťky pásma, délky, propustnosti

Autonomní systémy (AS)

- ↳ systém, jak sjednotit routerů na celém globálním internetu
- ↳ na globálním internetu se pracuje s lokální adresy vstupující do autonomních systémů
- ↳ uvnitř AS se používají ty interní routovací protokoly
- ↳ definice: množina sítě, co používá stejnou routovací politiku
- ↳ skoro každý ISP má svůj autonomní systém
- ↳ mezi nimi se používají externí routovací protokoly
- ↳ EGP (externí routovací protokoly)
 - ↳ např. BGP (Border Gateway Protocol)
 - ↳ hlavní, aby se nezacyklilo, tím, že nově posloupnost AS a nejde přes to, kde už byla → path-vector
- ↳ v ČR myslí starší

IP filtrování - firewall

(intranet / internet)

- ↳ Filtrování na routeru, který připojuje síť k internetu, tedy na transporthní vrstvě, v TCP/IP, protože dovolení provozu na určité porty
- ↳ Stanovit pravidla, tedy které porty jsou dovoleny z a do sítě
- ↳ přísma
 - ↳ není vybráno, dovolit nic
 - ↳ např. HTTP, SMTP ok
 - ↳ FTP, SIP → více kanálů, problém
- ↳ oboupří
 - ↳ není všechno, dovolit nic
 - ↳ u protokolů s mnoha kanály → SIP nepovítelno
 - ↳ u FTP s aktivním přenosem problém
- ↳ DMZ (demilitarizovaná zóna) → takže chováli jak k nebezpečnému internetu
 - ↳ pro provoz www serveru nebo porty, kde se otevřely kanály dovolit
 - ↳ zóna, kde je přístup zvenku v podobě, ale oddělený od sítě, dnes hosting

Proxy server

↳ software, co kontroluje provoz určitého protokolu z vnějšku dovnitř a obráceně

transparentní

↳ chová se jako server ke klientovi, vyhodnotí bezpečnostní politikou sítě a jako klient naváže spojení se serverem pokud ok

↳ pak jakmile přijde odpověď, tak zpracuje a pokud ok, tak pošle klientovi

↳ není třeba konfigurovat → uživatel o tom ani neví, že jde proxy
→ proxy server běží na routeru

netransparentní

↳ klient se musí nakonfigurovat, tedy musí specifikovat, že posílá proxy serveru a ne přímo

↳ proxy server nemusí být router a potřebuje tam být porty pro daný protokol

bezpečnostní → filtruje operace na aplikacím protokolu

vykonnostní → pokud odejde více požadavků, tak místo, aby odesílala všechny, tak odešle jeden a pak rozešle požadavky v síti tím, kdo potřebuje
30 dětí → 1 stránka → 30 odeslat → 30 přijmout, místo toho jen jednou

Otázky

- 1) Jak funguje směrovací algoritmus?
- 2) Popište typy sloupků a různé směrovací tabulky.
- 3) Čím se liší statické a dynamické řízení směrových tabulek?
- 4) K čemu slouží routovací protokoly?
- 5) Porovnejte distance-vector a link-state protokoly.
- 6) Co je autonomní systém?
- 7) Vysvětlete význam zpráv ICMP Echo, Time Exceeded a Destination Unreachable.
- 8) K čemu slouží pole TTL v IP záhlaví?
- 9) Co omezuje pojem IP filtrování?
- 10) K čemu slouží proxy server?

8. HODINA

Address Resolution Protocol

↳ umožňuje překládat síťové adresy na linkové adresy

↳ tedy konverze MAC (např. Ethernet) a síťových (např. IP) adres

↳ broadcastová výzva v rámci jednoho routeru

↳ pošle (FF:FF:FF:FF:FF:FF) všem a IP adresu a komu patří daná IP adresa pošle unicastově zpět svoji MAC adresu

↳ ARP klient si to uloží do cache, protože pravděpodobně s ní ještě bude komunikovat

↳ arp -a → vyprázdí cache

↳ síťovník může poslat gratuitous ARP - tedy nepřetlakovou odpověď

↳ může ověřit správnost odpovědi

↳ může být více MAC adres v síti oddělené routery

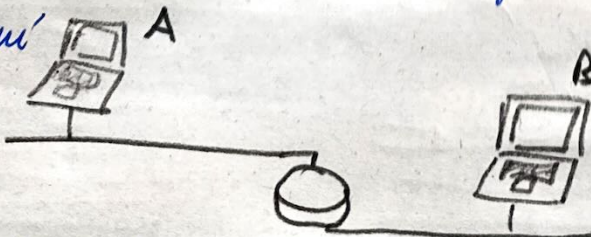
Proxy ARP

→ ale jiné podsíti

↳ pokud host A chce poslat hostu B paket, tak pošle ARP request s jeho IP adresou, ale namířuje na to router, který odešle svoji MAC adresu

↳ host A si tedy uloží MAC adresu routeru do ARP cache a jemu pošle pakety, které pak router pošle do jiné podsíti hostu B

↳ pak se stane, že router má v ARP cache pro stejnou IP adresu více MAC adres → Ok pokud je sám proxy, jinak je to napadení



Linková vrstva (OSI 2) → minimální úroveň sdílení stejného médiumu

↳ 2 podvrstvy:

LLC (Logical Link Control)

↳ multiplexing, tedy různé síťové protokoly se dohází uložit a pak předat po linkové vrstvě

MAC (Media Access Control)

↳ adresaci a řídí, kdo kdy smí něco posílat v rámci sítě na té linkové vrstvě

↳ PDU

↳ datová jednotka linkové vrstvy → frame (rámec)

↳ obsahuje: - syndronizační pole

↳ určité sekvence bitů, co probudí cílovou stanicí a tedy odliší se od šumu, který tam normálně je

- hlavička

↳ cílovou a zdrojovou MAC adresou, typ dat ...

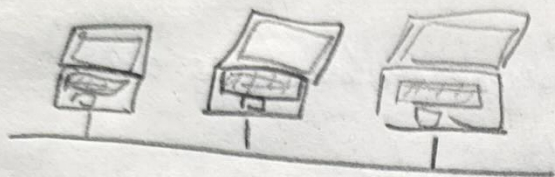
- data

↳ Frame Check Sequence (FCS), slouží ke kontrole správnosti

Typy síťových topologií

Multipoint

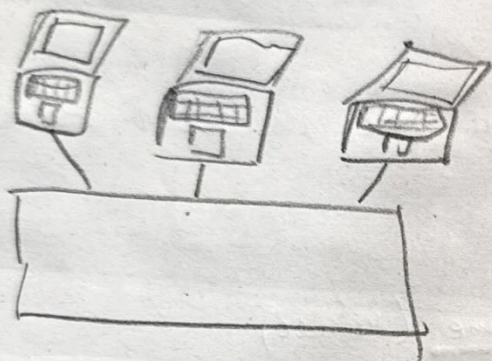
↳ komunikuje spolu více počítačů nejednou



← Stěrnice

↳ spojena kabelem, ale odeslané více zpráv mohlo způsobit kolizi

↳ požadavek kabelu změní síť
↳ koaxiální kabel



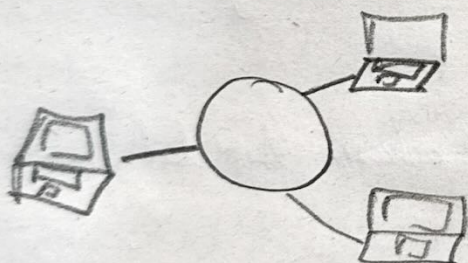
← Hvězda

↳ centrální prvek, poruší se jen kabel, ne celá síť

↳ fyzicky - tam může být stěrnice
to, co doopravdy

logicky - to je hvězda
- to, jak spolu všechny komunikují

Wi-Fi →



← Kruh

↳ kabel jde kolem dokola mezi počítači

↳ Token - ring, FDDI

↳ každý posílá a kdo ho přijme, může vyslat

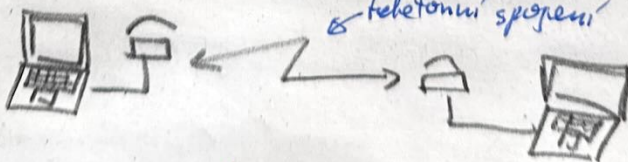
Point-to-point

↳ komunikace mezi dvěma počítači

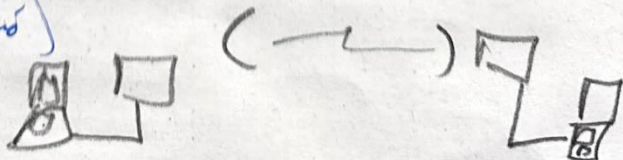
↳ přímé připojení kabelem např. RS-232 →

↳ Ethernet → plus stěrnice

↳ připojení přes modem



↳ bezdrátové připojení (laser, rádiové)



Epizody řešení přístupu k médium

Multipoint

↳ deterministický epizod řešení

↳ mělo řídit, kdo smí a nesmí vysílat

↳ token (prvek) nebo mělo řídit centrálně

↳ neterministický

↳ mělo neřídit, takže možou vzniknout kolize

Point-to-point

half-duplex

↳ nedobře současně přijímat a vysílat

full-duplex

↳ sítová karta

↳ dobře vzájemně současně přijímat i vysílat, tak se nastaví 2 simplexní přenosové linky

Řešení kolizí

CSMA (Carrier sense with multiple access) → možná volná

↳ měl sleduje jestli na médium neprobíhá přenos a pokud ne, začne vysílat, ale nezaručí, že nebudou kolize, protože se mohou 2 uzly synchronizovat vysílat současně

CSMA/CD (Collision detect) → Ethernet

↳ během vysílání detekuje kolizi a když tak zastaví přenos a počká náhodnou dobu, která podle exponenciálního při každém pokusu

↳ ale doba vysílání rámce musí být delší než doba šíření

mozi uzly, aby měl detekovat kolizi a rámec se nevrátil

↳ max. délka segmentu a min. délka rámce

CSMA/CA (Collision Avoidance) → Wi-Fi

↳ podle se rámec pokud je možná volná, odešle rámec a čeká na ACK, pokud nepřijde, tak exponenciálně čeká

Ethernet

↳ vznikl ve firmě Xerox, převzal IEEE

↳ CSMA/CD → po 16s čekání chyba

↳ adresy: 6 bytů

↳ 3 byty prefix výrobce, 3 byty adresa

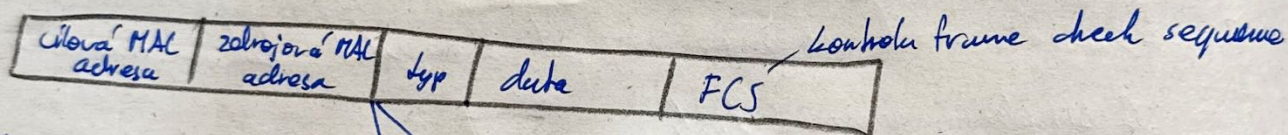
↳ drát napájen, dnes se dá změnit

Standardy IEEE 802.3

↳ půlno vytvářel ethernet, dokonal drát krok s HW

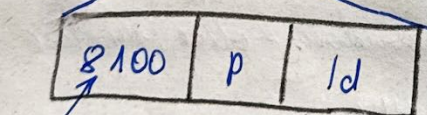
↳ dnes optický kabel → 100 Gb/s

Struktura ethernetového rámce



Virtuální síť (VLAN)

↳ jak na jedné fyzické síti spouštět několik lokálních sítí



↑
subnet

↑
priorita

↑
id → 12 bitů

↳ může se vložit před typ v rámci

↳ stroj pošle rámec, switch tam přidá ten tag, přoudeľ po síti, dokle na jiný switch a na něm to je koncové zařízení, tak odebere tag a pošle, přelo o tom zařízení, výrobce ani není

↳ router musí vidět všechny VLANY

↳ do trunk portu pošle neotevorený data

Cyklický kontrolní součet (CRC) - cyclic redundancy check

↳ Frame Check Sequence

↳ kashovací funkce založena na dělení polynomu, protože to lze udělat HW

$$\boxed{110} \rightarrow \dots + 1x^{20} + 1x^{27} + 0x^{26} + \dots$$

↳ vezmeme charakteristický polynom např. CRC-16 to je $x^{16} + x^{15} + x^2 + 1$

↳ vycházíme z vezmeme zbytek po dělení, to se dělá FCS a na konci to zase přepočítá a zjistí zda se shoduje / neshoduje

↳ detekce detekce chyby velmi dobře

Wi-Fi

↳ není protokol, je to označení technologie, tedy skupinu protokolů IEEE 802.11

↳ frekvence přenosu 2,4 GHz; 5 GHz

↳ maximální rychlosti 1 až 600 Mbps

↳ má hvězdicové topologii

↳ CSMA/CA

↳ SSID → service set id, pro rozlišení sítě

↳ problém zabezpečení

Fyzická vrstva (OSI 1)

↳ přenos signálu po konkrétním médiu

↳ byly na analog a oběma

↳ metallické: el. pulzy (kabely)

↳ optické: světelné pulzy (optika)

↳ bezdrátové: vlny

Druhy přenosu dat

modem: $D \rightarrow A$ (modulátor (demodulátor) - bity
codec: $A \rightarrow D$ (coder (decoder) - zvuk tržbou

↳ vše je analogový, jen se potřeba interpretovat a převést

baseband

↳ přenáší signál, který kóduje proudem asi a kóduje pro synchronizaci

↳ kv. Manchester přenos - používá Ethernet, ve slotu stoupne $\rightarrow 1$
klesne $\rightarrow 0$

broadband

↳ vysílá vlna, co se moduluje

↳ AM, FM $\rightarrow$ změna amplitudy vlny

Nestíněná kroucená dvojlinka (UTP) - unshielded twisted pair

↳ kolik je pár drátů $\rightarrow 8$ vodičů

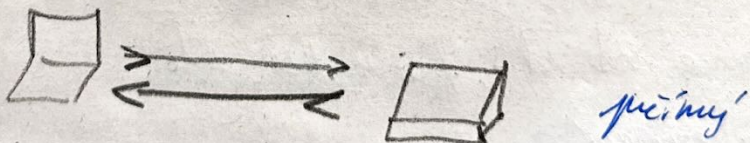
↳ 4 páry Cu vodičů pravidelně zkroucené

↳ zkroucení snižuje vyzařování a příjem elektromagnetického záření

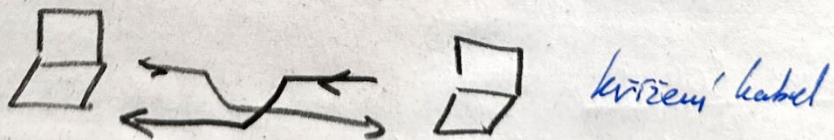
↳ protože na obou kabelech se proudí stejná směrem a ke snížení pole odlehčí

↳ 100Mb Ethernet používá jen 2 páry, protože lze rozdělit

↳ 2 různá zařízení



↳ 2 stejná zařízení



↳ ale oni se dohodnou pokud mají hardy MDI/MDIX

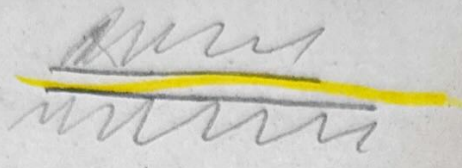
STP $\rightarrow$ kabel s kovovým stíněním
 $\rightarrow$ shielded twisted pair

Optická vlákna

↳ velká šířka pásma, vysoká frekvence a větší rozsah

↳ nekočkat do kabelu, vyšší cena, horší manipulace

↳ jednoválcová



- laser, větší dosah a
přenosová rychlost

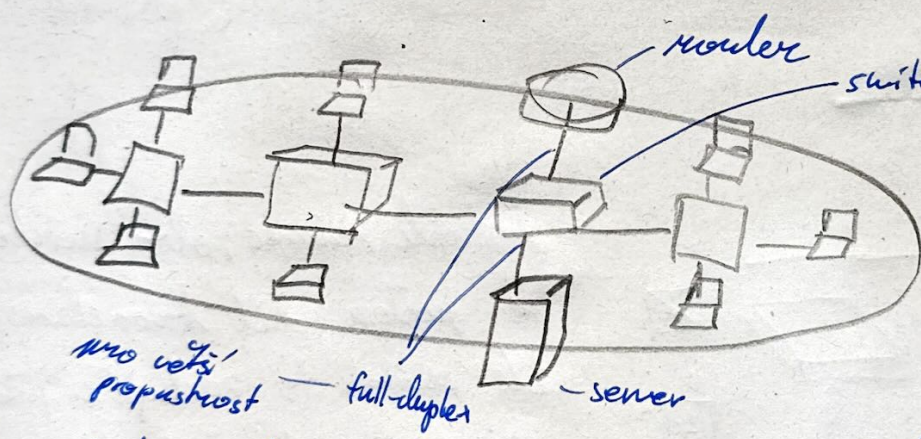
↳ multimode



- více lámen, pomalejší, LED

↳ sluková z trémítu

Segmentace sítě



- celé je broadcastová doména

repeater - hub - rozlišovací -> zvyšší propustnost a více kolizí, protože nám to trvá

↳ distribuuje fyzický signál cizí by nám rozeměl, větší dosah

bridge (most)

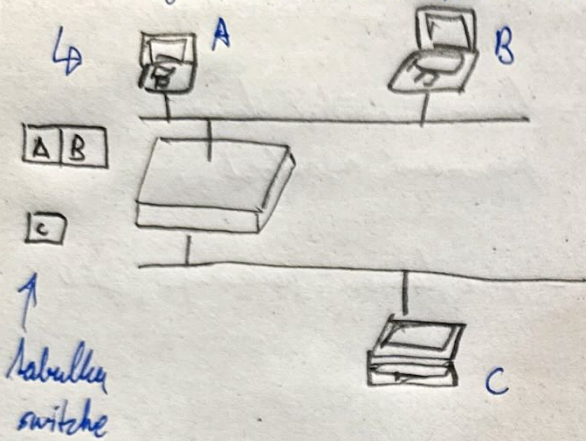
↳ zlepšuje propustnost, protože rozumí MAC adresám a máme nám
liniové vestru poslat kam, kam je to nutné

↳ br. switch

Learning bridge, BUM (BUS) - broadcast and unknown services

↳ Každé přenosové switchy

↳ udržuje si tabulku MAC adres stanic

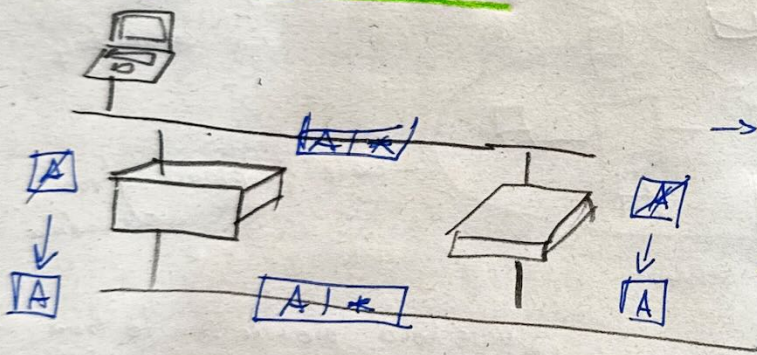


→ B pošle C, switch není a pošle broadcast všem, ale uloží si B do tabulky

→ C pošle A, switch není, zase broadcast, uloží si C

→ A pošle B, switch ví, proto nepošle na jinou linku k C, ale rovnou pošle B

Spanning Tree Algorithms



→ A pošle rámec, oba switchy to pošou dál broadcastem, ale switch to zachytí a přesune si počítač do tabulky jinam, takže přestal existovat

↑
důvodem je cyklus

↳ řešením je najít optimální množinu → kostru (spanning tree)

STP (spanning tree protocol)

↳ některé switchy jsou v režimu blocking, kdy nepřeposílá rámce, ale jen monitoruje, zda nevypadne druhý switch, což je ve stavu forwarding a tedy rámce přeposílá

↳ pomalé → proto celou síť může udeřít failstart a zavaz, že se vytvoří cyklus

Odpisy:

- 1) Popište účel a bezpečnostní mechanismy protokolu ARP.
- 2) Popište účel vektor LLC a MAC
- 3) Jaký je rozdíl mezi fyzickou a logickou topologií sítě?
- 4) Co je kolize, kde se vyskytuje a jak se řeší?
- 5) K čemu slouží a jak se realizuje VLAN?
- 6) K čemu slouží CRC a jak pracuje?
- 7) Popište způsob přenosu dat v závislosti na médii.
- 8) Popište typy metalických a optických kabelů
- 9) Co je learning bridge a STP?