1. HODINA

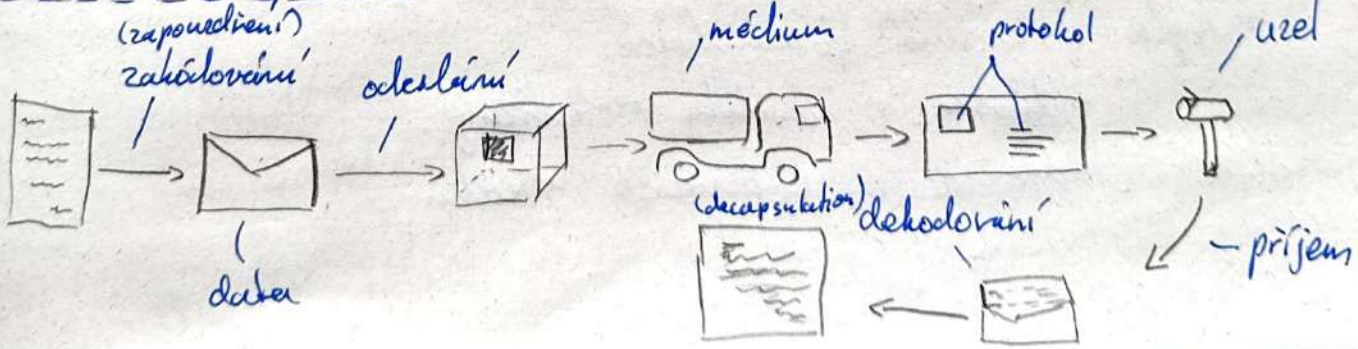
Obecné atributy komunikace

- identifikace → musejí se mezi sebou najít
- metoda → hlas, znaková řeč, psaná forma
- jazyk → dohoda na jazyce
- rychlost → rychlost komunikace, občas třeba zpomalit
- proces → nejen. pozvání k prezidentovi, nějaký rituál společenský

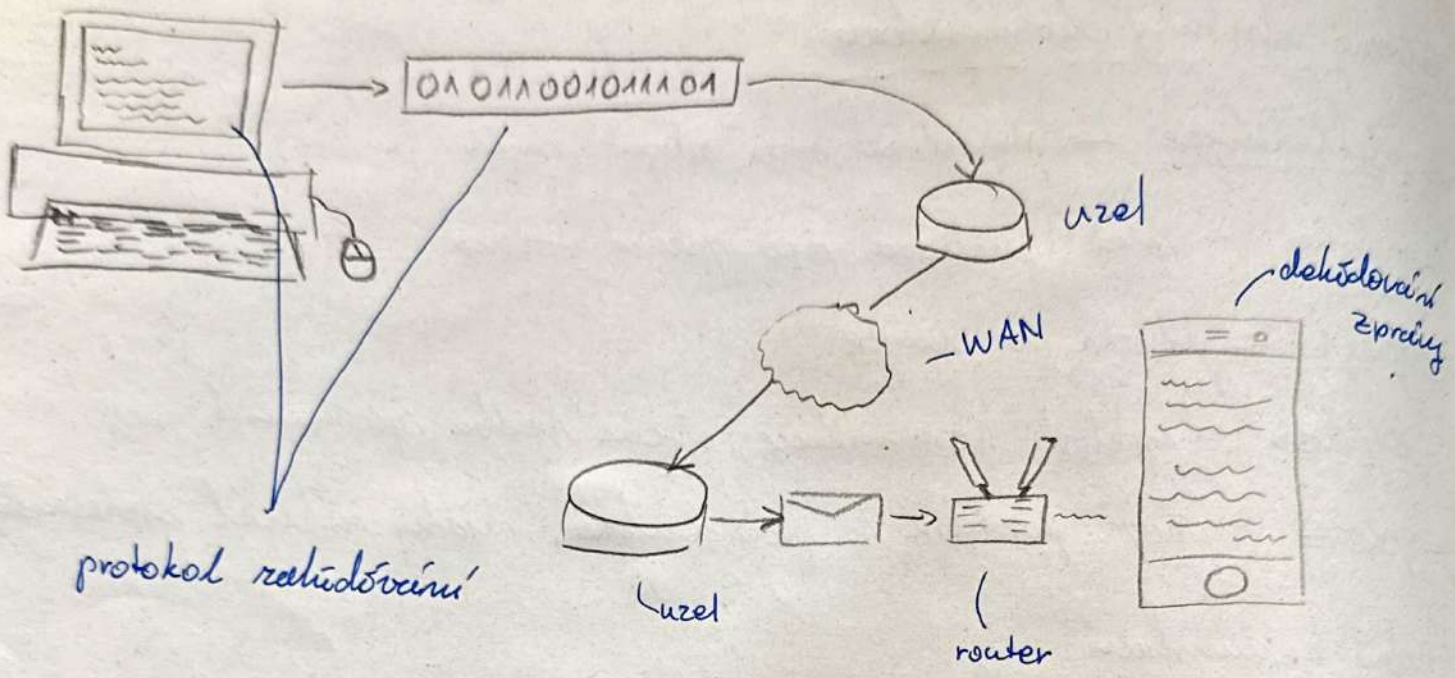
Typy komunikací

- řečová → hlas, písmo, signály
- telekomunikace → zabudovaná pravidla, vše má pod kontrolou síť, telefon umožňuje přístup
- počítačová síť → volně dostupná pravidla
  → síť se stará o přenos a pravidla a logiku si vrství konečné zařízení
- konvergovaná síť → počítače používali linky (dříve)
  → pak telefony používali počítačovou síť (Voice over IP)
  → pak ale mobilní data zase vrátily počítačovou síť přes telefonní kanály

Přenos zprávy poštou



Přenos zprávy (e-mail)

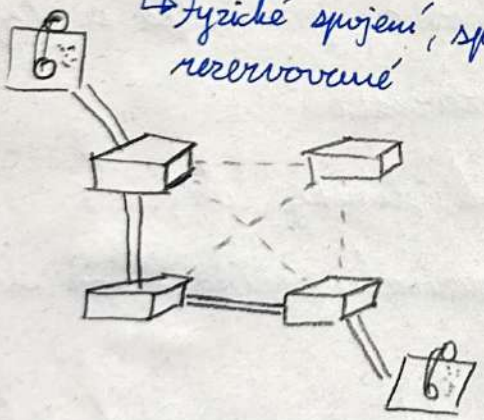


Požadavky - odolnost

↳ americké ministerstvo obrany → řeší výpadek telefonních sítí v případě útoku, protože přepínali obvody

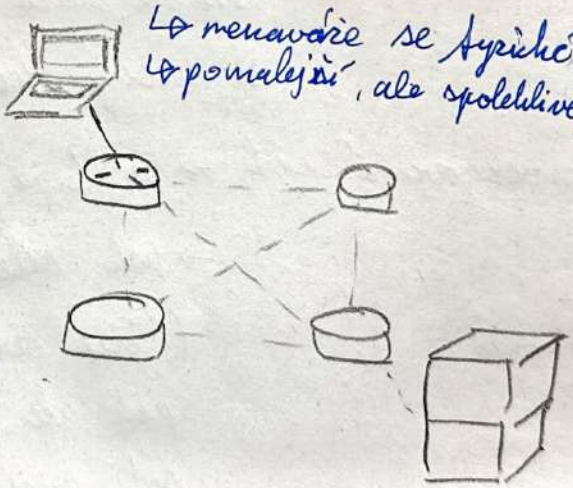
přepínání obvodů

↳ fyzické spojení, spojení rezervované



přepínání paketů

↳ nenavazuje se fyzické spojení
↳ pomalejší, ale spolehlivější



Požadavky - bezpečnost

↳ řešila se fyzická bezpečnost, zmenšila se ochrana dat

↳ byla: - otevřená 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) omezit přístup ⊘
2) záložní zdroje napájení 
3) záložní servery s daty  ...

↳ jak zabezpečit data?

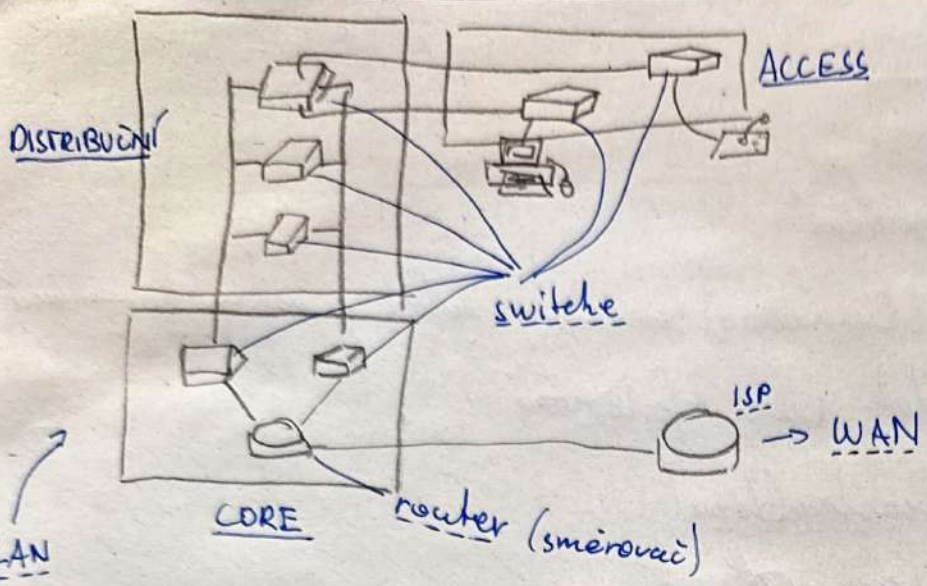
1) ověřování uživatelů a ověření serverů / počítačů
2) kryptografie (šifrování dat)
3) inspekce dat (přes antiviry)

Požadavky - rozšiřitelnost (LAN)

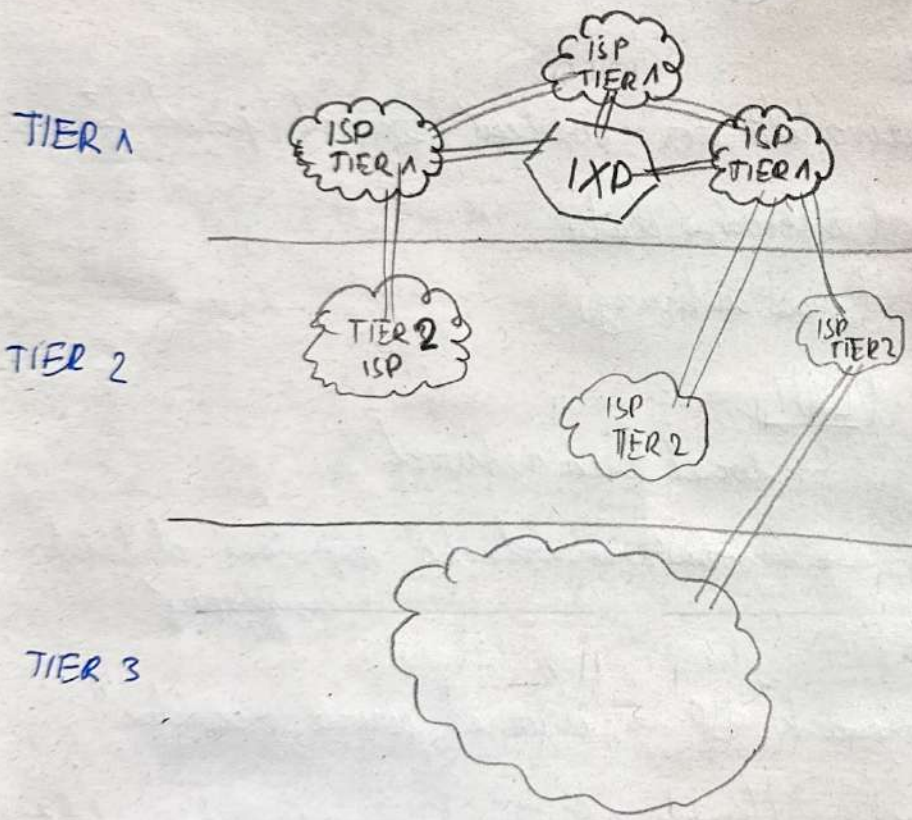
↳ „local area network"

↳ vytvoření 3 vrstev: → umožní jednoduché rozpojení dalších rozšíření

1) core
   ↳ síť je připojena k ISP → „internet service provider"
   ↳ v místnosti s klimatizací a UPS → „uninterruptable power source" neboli náhradním generátorem
   ↳ má router připojený rovnou k ISP a na druhé straně ke switchům → propojují další uzly sítě
2) distribuční
   ↳ „vertikální" → obsahuje switche a propojuje další switche v access vrstvě (po patrech / v patře)
3) access (přístup) → „horizontální", propojuje počítače, tiskárny...



Požadavky - rozšiřitelnost (WAN)



- přímý přístup k internetu, IXP-internet exchange point
- např. AT&T

- národní operátoři, zákazníci jiní ISP

- školy, domácnosti, propojí svůj LAN

Požadavky - kvalita služeb

přenosové parametry: → různé aplikace, různé požadavky:
- Youtube - ztrátovost nevadí, ale pravidelně
- email - nesmí být ztrátový

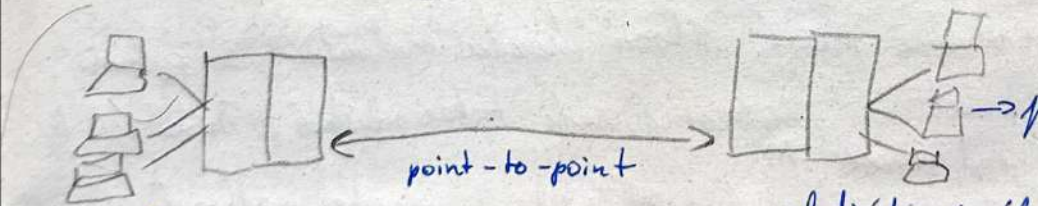
- latence → zpoždění od doručení k obdržení
- pravidelnost doručování → jitter nebo-li jak objem kolísá
- ztrátovost dat → jak často není paket doručen
- šířka pásma → kolik dat lze přenést, rychlost světla je vždy stejná

cíl:
- doručit rychlé prioritní pakety
- zaručit pro aplikace dostatečnou šířku pásma pro normální fungování

jak:
- pakety označit QoS (quality of service), ale nelze zaručit, že se aplikace budou chovat fér
- vyhradit šířku pásma pro aplikace, ale může se stát, že aplikace ji nevyužívá a pak zbytečně zabírá místo
- prioritní fronty → fakt, že zprávy vidí do fronty, ale priorita se dá zjistit až po načtení celého paketu, takže ostatní čekají

Vznik počítačových sítí

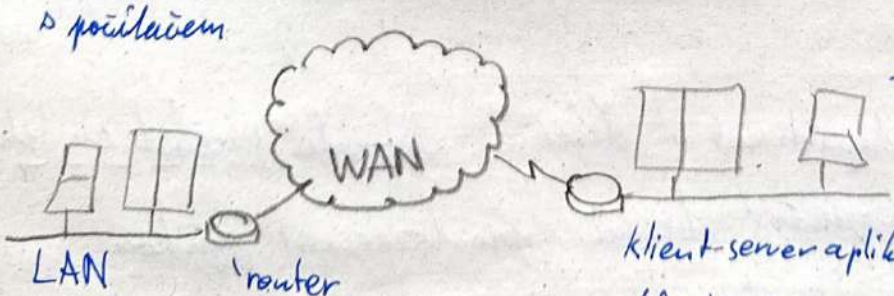
→ přenášet děrovací štítky



→ přes telefonní linky

terminály propojeny s počítačem

delší terminály propojeny s počítačem



→ přípojení LAN k WAN

klient-server aplikace, kde běží něco jako terminál

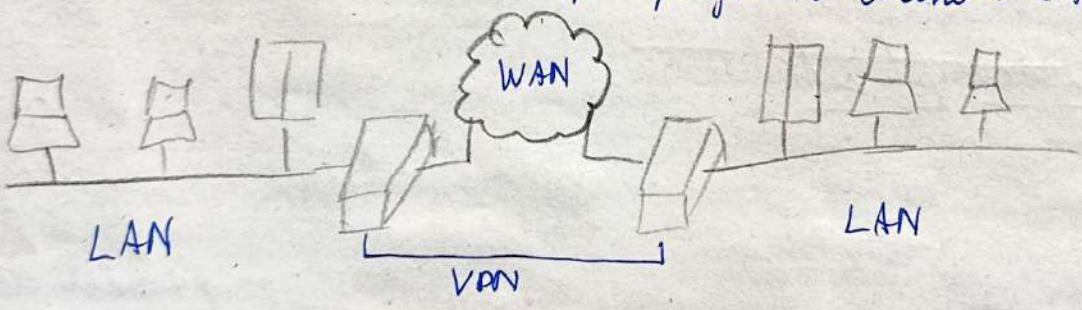
Dělení sítí

- Lokální sítě (LAN)
  - malé vzdálenosti (kampus), jednotný vlastník
- Rozlehlé sítě (WAN)
  - vzdálený, rozlehlý, mnoho vlastníků
- Dnes → rozdíly jsou menší, rozdíl právě ve vlastnictví

Veřejné a privátní sítě

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

síť se jeví jako jedna jak jedlová 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) ≠ VPN (virtual private network)

Historie internetu

- 1960 → US DoD (department of defense) stvořilo packet switching namísto přepínání okruhů pro bezpečnost fyzikou
- 1969 → ARPA (Advanced Research Project Agency) → vytvořilo ARPANET, kdy síť terminálů vytvořili pomocí point-to-point

→ pronajali telefonní linky

1974 → vytvořen pojem internet (internetworking)

1977 → k ARPANET připojena další sítě

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

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

Request for comments (RFC) ← IAB → IETF, IRTF → RFC

↳ od 70. let → dnes vydávají 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 dělat

↳ na ietf.org/rfc.html

Otázky: (1)

1) Výhody/nevýhody přepojování paketů.
2) přístup na síťových protokolech, že vznik sítí inicializovala armáda?
3) požadavky na škálovatelnost sítě
4) různé emailu a telefonu přes síť → rozdíly
5) definice LAN
6) definice VPN

2. HODINA

Vrstvená filozofie

- Zapisovatel
  ↳ musí zapsat správně → něco jako protokol
- Sekretářka
  ↳ dopis zabalí a doplní adresu a pošle na podatelnu
- Podatelna
  ↳ vezme balík a vyhrne strategii odesílání
- 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)
↳ rozdělaje práci mezi vrstvy

Síťová architektura (protocol suite) → např. TCP/IP (sada protokolů)
↳ vezmu síťový model a přidáme další technologie, např. protokoly

OSI (Open Systems Interconnection)

↳ nepoužívá se už, jen k výuce
↳ zahrnuje referenční model + protokoly
↳ nevyužívaná rozhraní

Vrstva		Funkce
7	aplikační	předání informací uživateli
6	prezentační	řeší rozluštění dat po přenosu – např. endianita
5	relační	řídí 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

↳ v Microsoft Exchange Post

↳ složitá struktura adres, ale jednoznačné

G=Libor S=Forst
O=charles university } dohromady info
...

X.500 (Directory Access Protocol)

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

X.509 (správa veřejných klíčů)

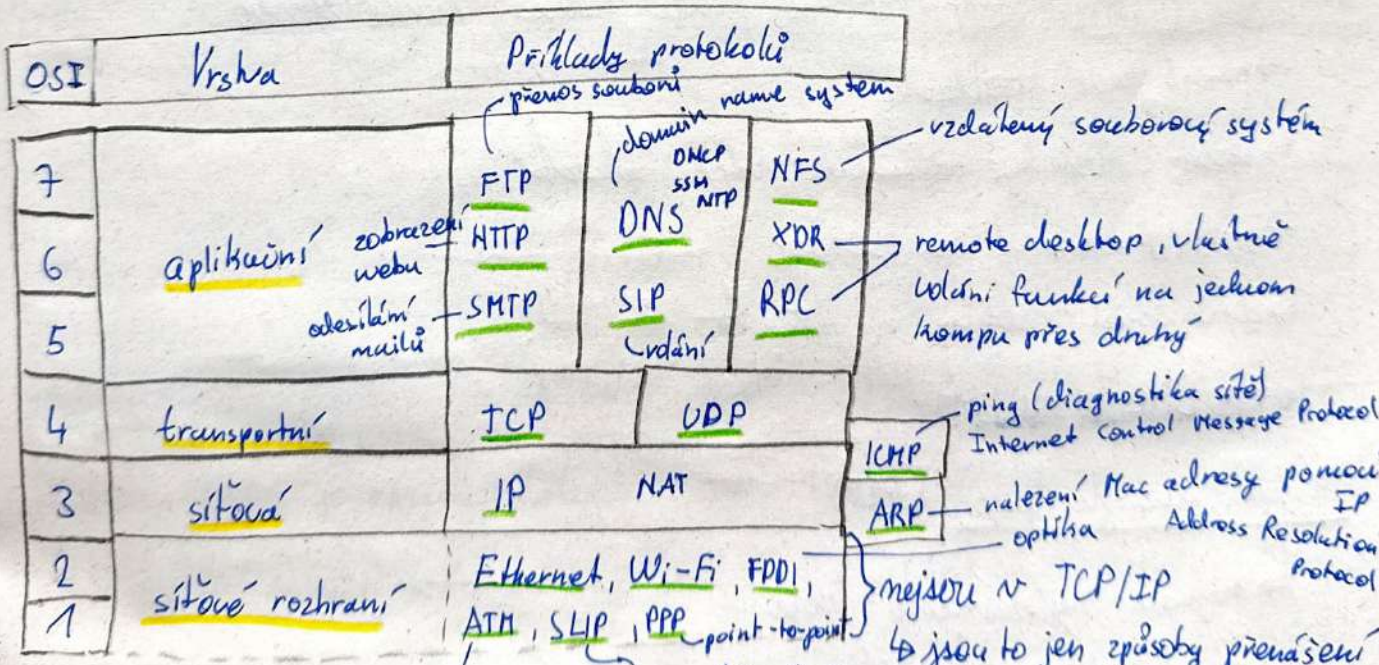
↳ asymetrická kryptografie

↳ každý klíč musí mít jednoznačný identifikátor

LDAP (Lightweight DAP)

↳ databáze údajů o osobách a službách

Rodina protokolů TCP/IP



↳ navrženo zespoda, nebo-li od nejjednodušších aplikací

Spojované / nespojované služby

Spojované (TCP)

↳ spolehlivé doručení dat

↳ implementace aplikace je jednoduchá, ale TCP je složité

↳ takové telefonní spojení

Nespojované (UDP)

↳ něco jako poštovní spojení

↳ aplikace musí vždy sama řešit potvrzení doručení a případné opětovné odeslání

↳ jen pošle paket a o nic více se nestará UDP

Aplikační modely

Model klient-server

↳ navázání spojení → klient podle adresy najde server

↳ klient zadá požadavky

↳ server obsluhuje většinou více klientů

↳ upload: klient na server

↳ download: server na klienta

↳ př. DNS, WWW, SMTP

Model peer-to-peer (P2P)

↳ nejsou vyhraněné 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 nadělovány výrobcem, dnes nastavitelné

↳ ethernetové adresy → 3 bajty prefix výrobce
→ 3 bajty číslo síťové karty

↳ není pro komunikaci mezi sítěmi, nerespektuje topologii nebo-li mac adresa bude mít jinde stejnou

↳ IP adresa (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 v rámci určité sítě

jednoznačně určí síť a počítač

↳ doménová adresa → aplikační vrstva (pro lidi)

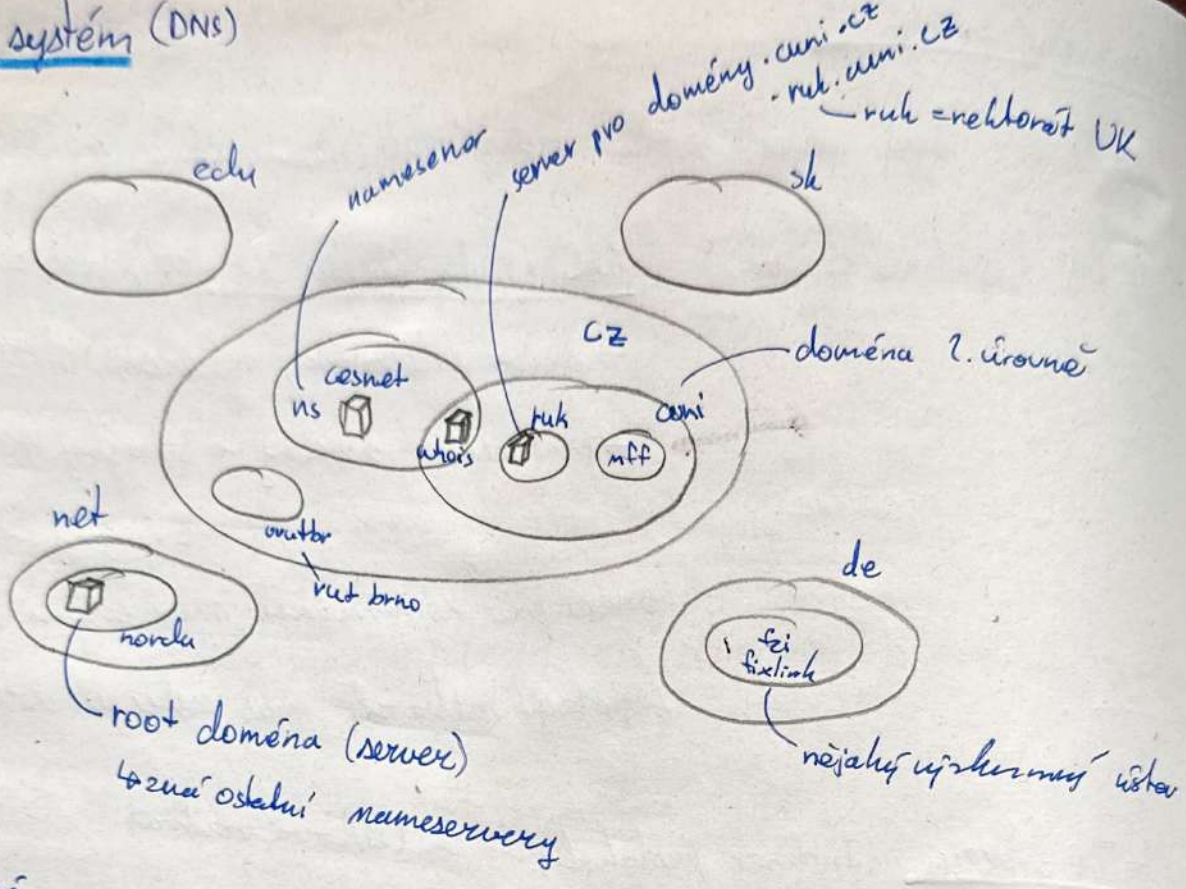
↳ hierarchie je zprava doleva www.mff.cuni.cz

↳ 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
↳ definování systému doménových jmen a TLD
↳ arpa ("ARPA net") → technické
↳ 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
↳ přidány další (info, biz, aero, ...)

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

SLD
↳ second level domain
↳ spravuje vlastník
SLD
ms.mff.cuni.cz

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 ← nuly ::)

↳ prefix → definuje síť

→ veřejné adresy pro přístup k internetu přiděluje ISP

→ privátní adresy si může tvorit síť přidělit sám → zvyšuje bezpečnost, ale interoperabilitu → směrovací adres NAT
network address translation

↳ přiřazení adresy:

↳ ručně správa sítě

↳ statické → předdefinovaná IP adresa

↳ dynamické → přiřazena na vyžádání

→ musí být ověření, že se smí připojit

→ DHCP zajišťuje přiřazování

↳ Dynamic Host Configuration Protocol

↳ link-local → počítači 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 (většinou)

↳ port z well-known services

↳ např. webové služby → 80

↳ source-port

↳ přidělí systém o přiřazení generického portu, který používá v komunikaci

↳ aby bylo možné identifikovat více klientů najednou pro jednu službu

Socket

↳ označení jednoho konce komunikačního kanálu

↳ < IP adresa, Port >

↳ v TCP/IP < zdrojová IP, zdrojový port, cílová IP, cílový port, protokol >

↳ dva různé kanály stejného serveru se musí lišit aspoň v zdrojovém portu

↳ stejná čísla portů lze použít pro různé typy protokolů

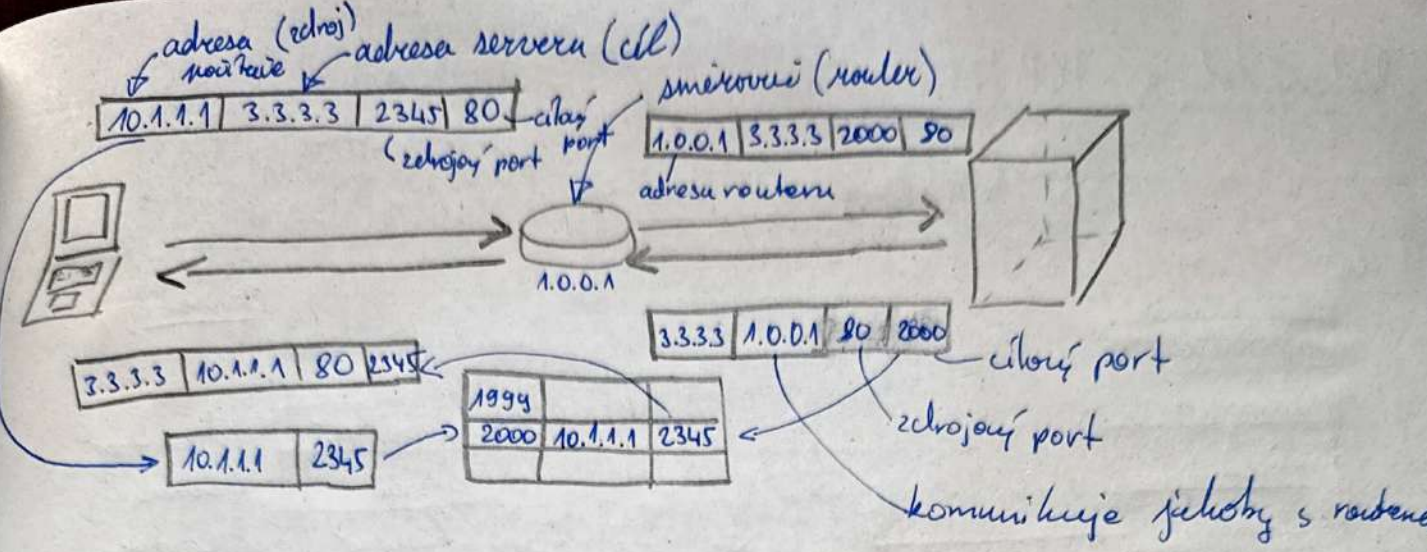
Příklady Well-known services (znát!)

- FTP (file transfer protocol) → 21/TCP ← používá primárně TCP, ale lze i UDP
  ↳ 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

Překlad adres (NAT) → network address translation

→ „IP masquerading" (maskování IP)

↳ lokální síť používá privátní adresy a navenek má jiné



↳vlastní port se zaměstnává k přiváté adrese do registru (zdrojový počítače

a server si připadá, že komunikuje s routerem, který nahrazuje jak port, tak IP adresu za veřejnou routeru, tak vlastně mění socket <IP adresa, port>

Adresování služeb

Uniform Resource Identifier (URI) → RFC 3986

↳ zaměnitelné s URL (Uniform resource locator), co může komplementovat URN (Uniform Resource Name)

↳ jednotný systém odkazů pro více služeb (FTP, WWW)

http (není protokol!)

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

obsah — server/doména, pokud nestandardní port, je třeba ho zde uvést jinak well-known services — v souborovém systému

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

př. ftp://sunsite.mff.cuni.cz/Net/RFC

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

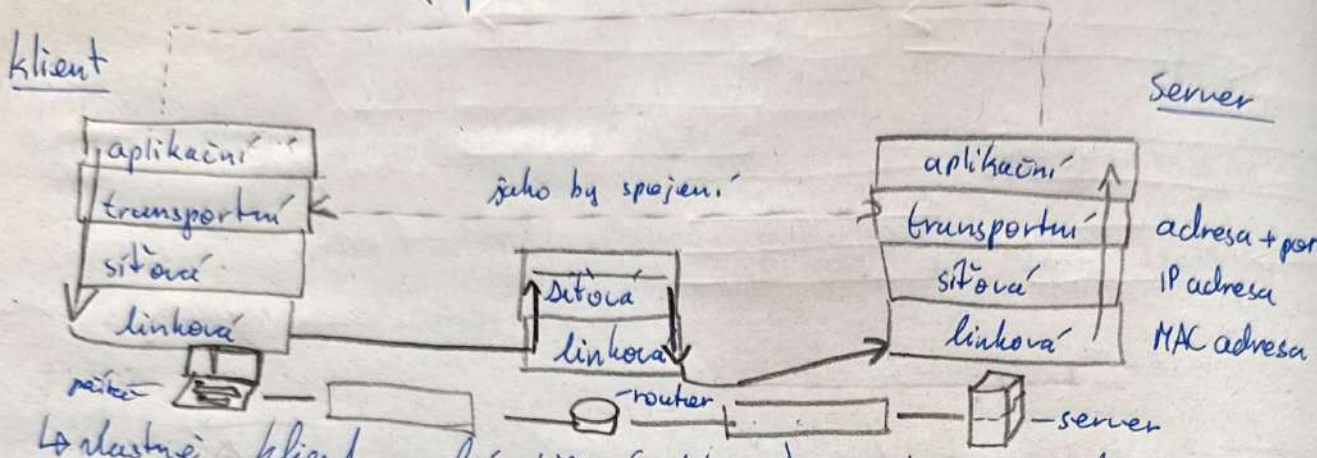
mailto:forst@cuni.cz

sip:22191111@voip.cz

Datový tok v TCP/IP

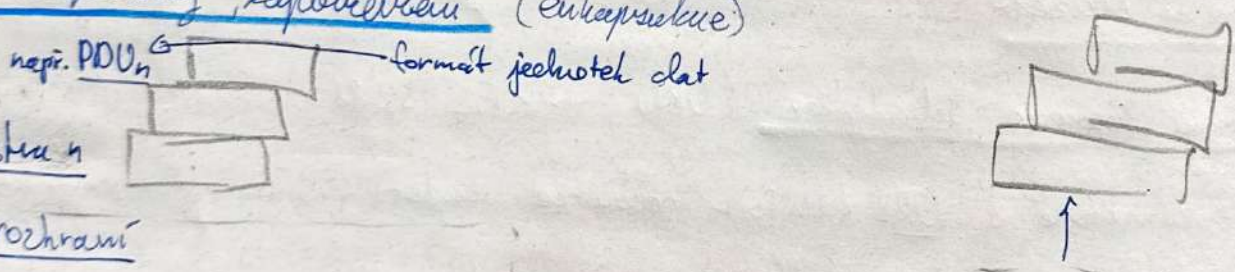
URI (URL)

http://www.mff.cuni.cz

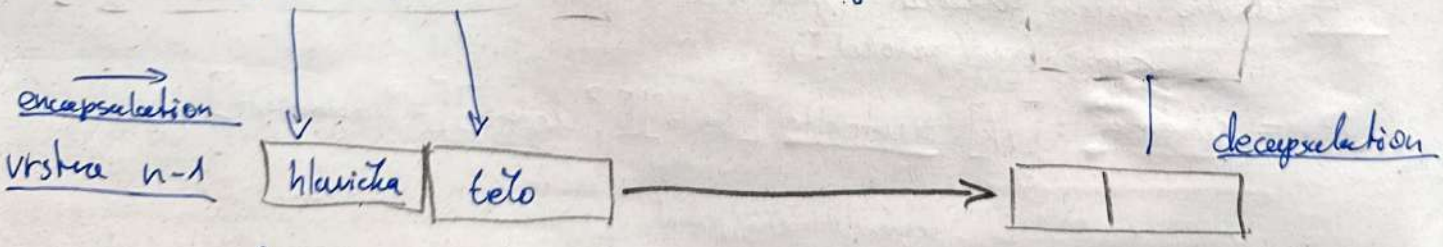


↳ klientské klient zadá URI (aplikační) a aby navázali spojení s adresou a portem služby je potřeba mít IP adresa serveru, takže ve stejné síti mají nejbližší next-hop uzel, což je server jinak musí přes router do jiné sítě a to jde přes síťovou vrstvu posílání dat a nalezení next-hop uzlu → a to pak mezipočet

Multiplexing zapouzdření (encapsulace)



funkce (příkaz, data) ← pomocí knihovny



odesílatel

↳ PDU (protocol data unit)

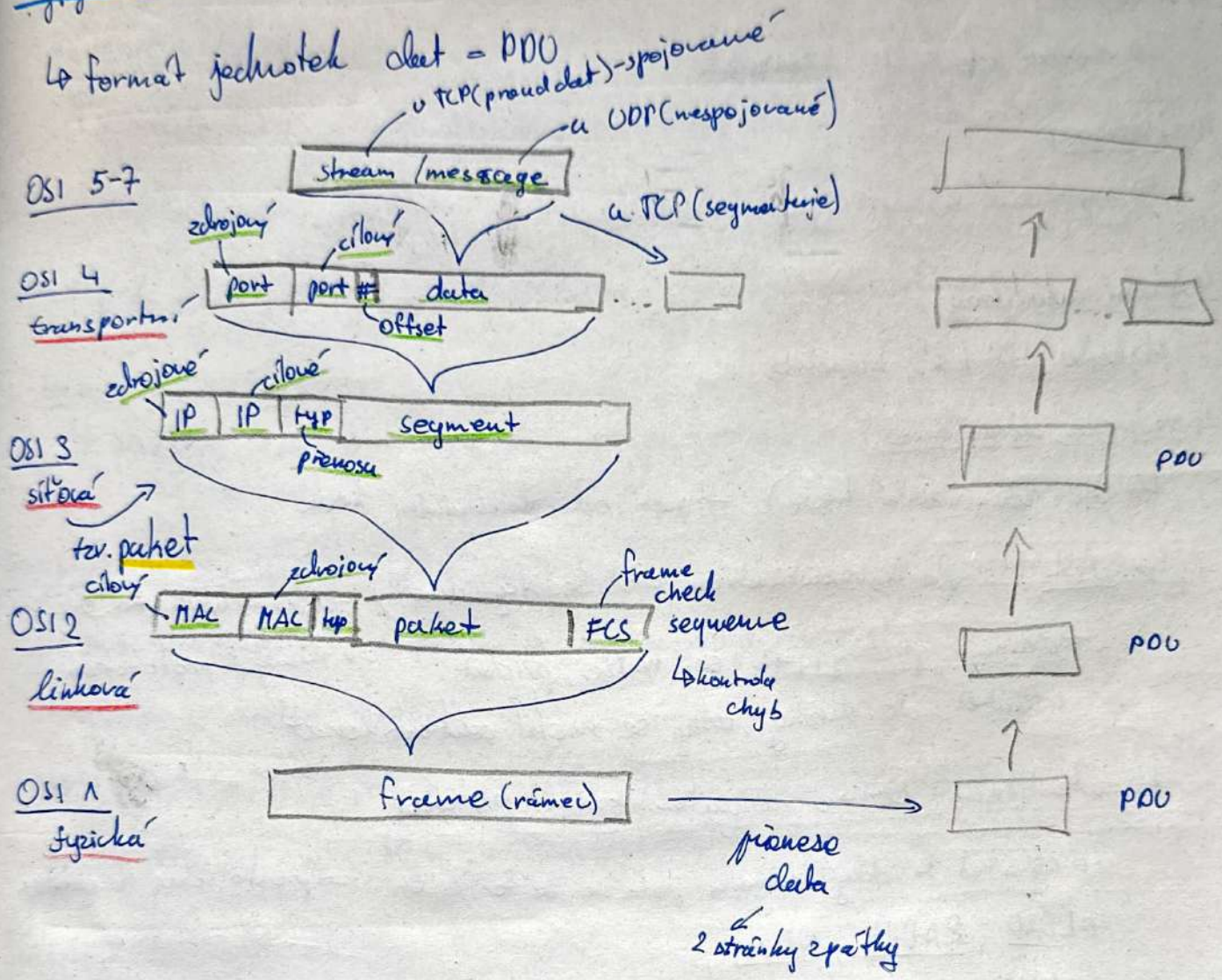
↳ vrstva n pošle do n-1 nebo do jakékoliv jiné vrstvy, kde se výměna dat informace zapouzdří díky rozhraní (knihovně) a podle protokolu se pošle → různé by měly např. IPv4 a IPv6

Formát IDU (interface data unit)

↳ přijemce to nakonec otevře (decapselation)

Typy PDU v TCP/IP

↳ formát jednotek dat = PDU



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

↳ UDP → zpráva doručena aplikaci

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

Otázky: 1) Popište rozdíl mezi TCP, UDP, IP?
2) Jaké adresy se používají na aplikační, transportní, síťové, linkové a fyzické vrstvě?
3) Jaký je rozdíl mezi klient-server a peer-to-peer modelem.
4) Jak jsou spravovaná doménová jména?
5) Jak se přidělují IP adresy?
6) Jaký je smysl a princip NAT? 8) Co je multiplexing a zapouzdření?

3. HODINA

Autentikace, autorizace

↳ autentikace
  ↳ ověření identity uživatele

↳ autorizace
  ↳ vymezení služeb po autentikaci

↳ lokální autentikace
  ↳ heslo, HW klíčů, biometrie

↳ vzdálená autentikace
  ↳ OTP (jednorázová hesla) → proti odposlouchávání hesla
  ↳ SASL ("Simple Authentication and Security Layer") → framework na ověření v různých protokolech
    ↳ definuje profily, jak se může přihlásit uživatel do služby, aby se mohl autentikovat

↳ autentikační server a autentikační protokol
  ↳ uživatel se díky serveru prokáže, že to je on → prostě cas i guess
  ↳ LDAP, RADIUS, NTLM,
    ↳ lightweight directory access protocol
    ↳ katalogy uživatelů, služeb
    přihlašování do Wi-Fi
    starší autentizace Microsoft

one time password

offline
↳ nechal vygenerovat seznam hesel a jako po druhém zadávale

challenge - response
↳ místo hesla dostal řetězec, ten vložil do SW, přidal heslo a vypadl jiný řetězec a to bylo místo hesla
↳ to semné, jak přijde přihlášení mailem přes odkaz

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 používání

kryptografie - asymetrické šifrování

↳ pro šifrování a dešifrování se používá jiný klíč
↳ tajný a veřejný
↳ netriviální inverzní funkce (matematický základ - jednocestné funkce):
- násobení prvočísel a pak jejich rozklad → dlouho hard
- diskrétní logaritmus $m = p^k \bmod q$

↳ řádově pomalejší než symetrické
↳ někdo zašifruje veřejným klíčem a pošle data a jí si je otevře 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

↳ kryptografii:

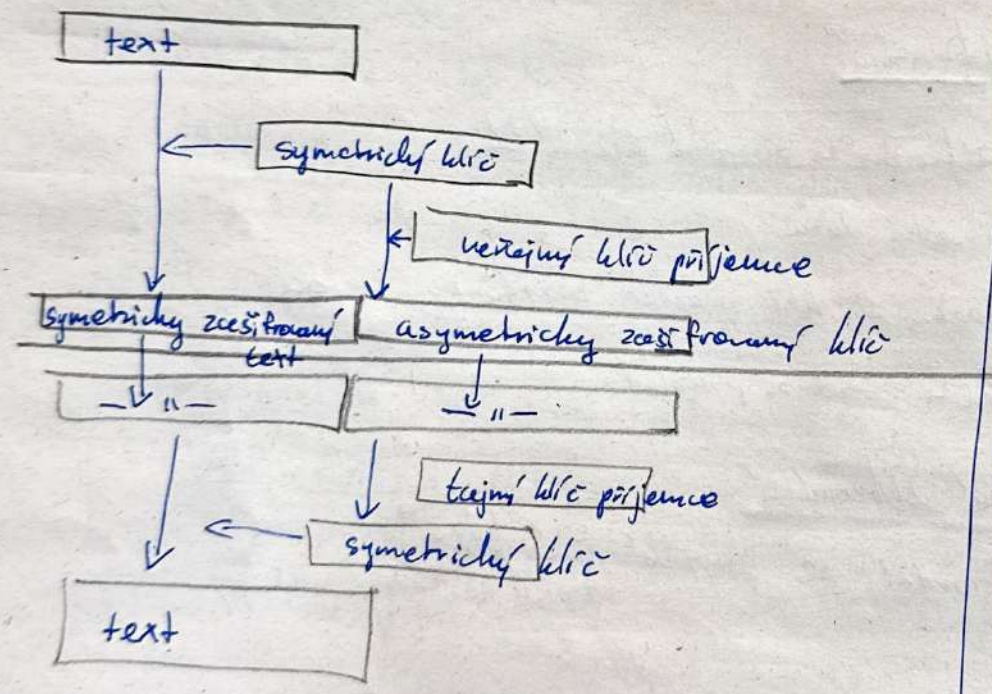
↳ jednosměrné

↳ malá změna textu → velká změna hashe

↳ text neodhadnutelný z hashé

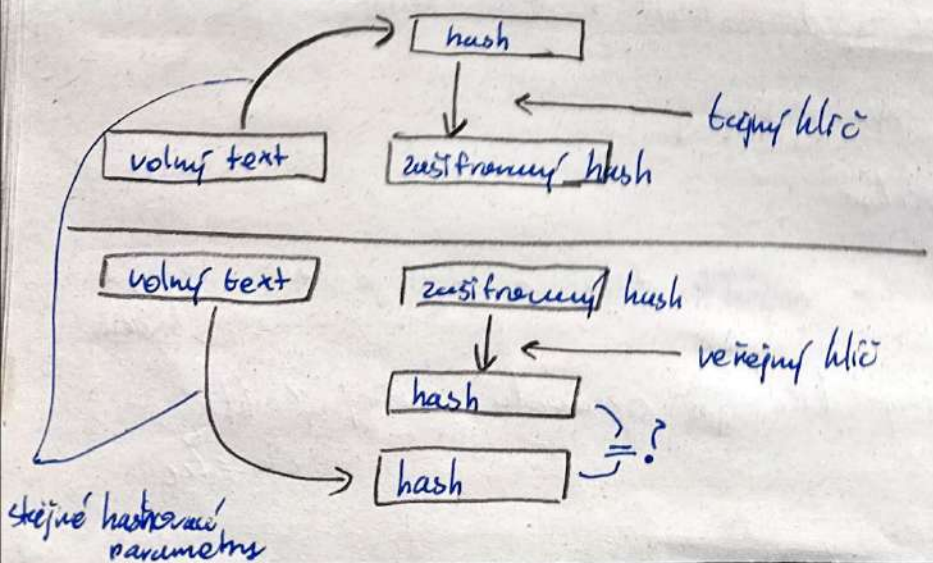
↳ SHA (secure hash algorithm)

Šifrování dat



výhoda:
- šifruje asymetricky jen malý klíč
- pro více příjemců tak zašifruji jen jednou, ale klíč umím zašifrovat pro každého z vlášť

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 → $s = B^a$
↳ Osoba 2 → $s = A^b$
} $s = s$

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

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

Autenticita veřejných klíčů

↳ zapečetěný autoritou

↳ PGP (pretty good privacy) → web of trust → někdo komu věřím podepíše, že ten klíč je správný
→ pošlu klíč a identifikační značku a požádám o ověření

↳ CA (Certification Authorities)
↳ vydá certifikát

Certifikát → dodaný k softwaru seznamu důvěryhodných CA

↳ certifikát je identifikace majitele klíče ověřená autoritou

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

struktura
- verze certifikátu
- sériové číslo certifikátu
- vydavatel
- 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
↳ dnes nové

↳ mezi aplikační a transportní

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

↳ princip:

1) pošle požadavek na SSL spojení
2) server pošle odpověď + certifikát
3) klient pošle zašifrovaný (veřejným) základ šifrovacího klíče
4) server rozšifruje tajným klíčem základ šifrovacího klíče
5) probíhá šifrované spojení

Aplikační vrstva TCP/IP

↳ spojuje OSI 5, 6, a 7

↳ pravidla komunikace mezi klientem a serverem

↳ protokol definuje:

- průběh dialogu
- formát zpráv → binární / textové
- typy zpráv → požadavky / odpovědi
- sémantika → každá zpráva má jen jednu interpretaci
- interakce s transportní → jestli použije TCP nebo UDP

Domain Name System

- překlad jmen na adresy a naopak
- nad UDP i TCP
  - do 512B
  - větší objem
  - dnes i více díky EDNS
- pokud odpověď od serveru delší → příznak truncated a klient navýší 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 ptát kteréhokoli nameserveru

DNS záznamy

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

pro IPv6 ::1 → 1.0...0 .ip6.arpa
(32 nul)

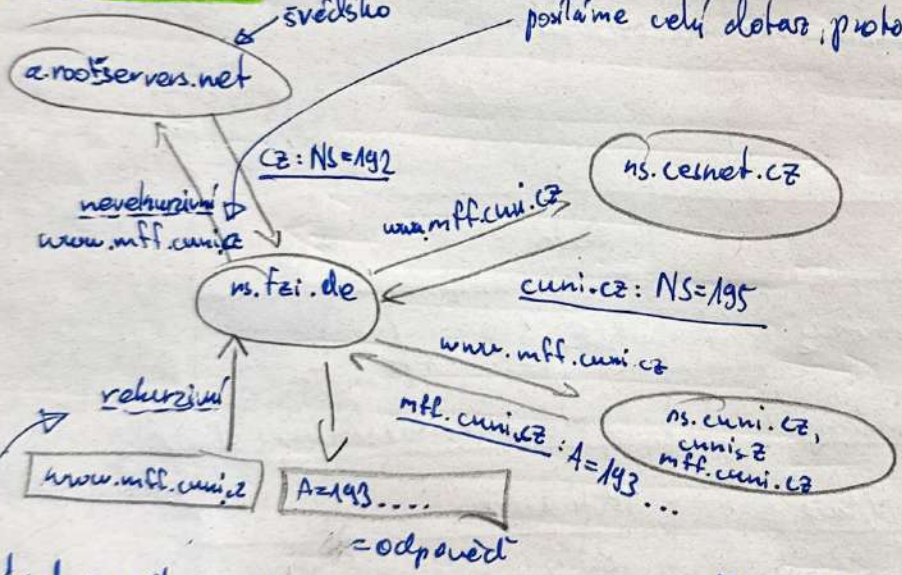
Servery DNS

typy:

1) primární → ten má databázi jmen
2) sekundární → zálohy, stahují si z master primárního serveru
3) caching-only → po nějakou dobu si je uloží

↳ každá doména by nejlépe měla mít více nameserverů registrovaných
↳ sekundární server se aktualizuje podle času uvedeném v SOA, ale je možné i že primární server informuje o změně dat

Vyřizování DNS



až budeš mít, pošli odpověď = rekurzivní

↳ pak si to uloží do cache

DNS dotaz a odpověď (i když je binárně)

Dotaz: ID: n

příznaky – FLAGS: Recursion

QUERY: www.cuni.cz IN A ← mě IPV4

Odpověď

ID: n

zapamatuj FLAGS: Authoritative Answer

→ QUERY: www.cuni.cz IN A

ANSWER: www.cuni.cz IN CNAME tarantula

tarantula IN A 195.113.89.35

AUTHORITY: cenia.cz IN NS golias
ADDITIONAL: golias IN A 195.113.0.2 ← pro příště

Bezpečnost DNS

↳ protože je jednoduché efektivní dotaz

obrana je, že ID a zdrojový port → celkově tak 4 miliardy možností a to nejde tak rychle sthnout

↳ může napadnout lokální síť

↳ cache poisoning -

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

↳ řešení: používat jen root servery a ne cache
↳ jen tehdy pokud z autoritativních

↳ DNSSEC: - podepisování záznamů klíčem nadřazeného serveru → útočník nejde unést cz nameserver
- rozšiřuje se pomalu → komplikovaný

Diagnostika DNS

nslookup → set type
→ server
→ ls
→ exit
} vypíše nameservery

dig / drill → [doména]

Otázky: 1) 3 základní skupiny kryptografických algoritmů a rozdíly mezi nimi.
2) princip šifrování a elektronického podpisu.
3) účel a podstata Diffie-Hellmanova algoritmu
4) jak ověříme pravost veřejného klíče.
5) rozdíl mezi HTTP a HTTPS
6) Kdy se používá UDP a kdy TCP?
7) účel a chování sekundárního nameserveru
8) postup řešení DNS dotazu
9) bezpečnostní rizika DNS

4. HODINA

File transfer protocol (FTP)

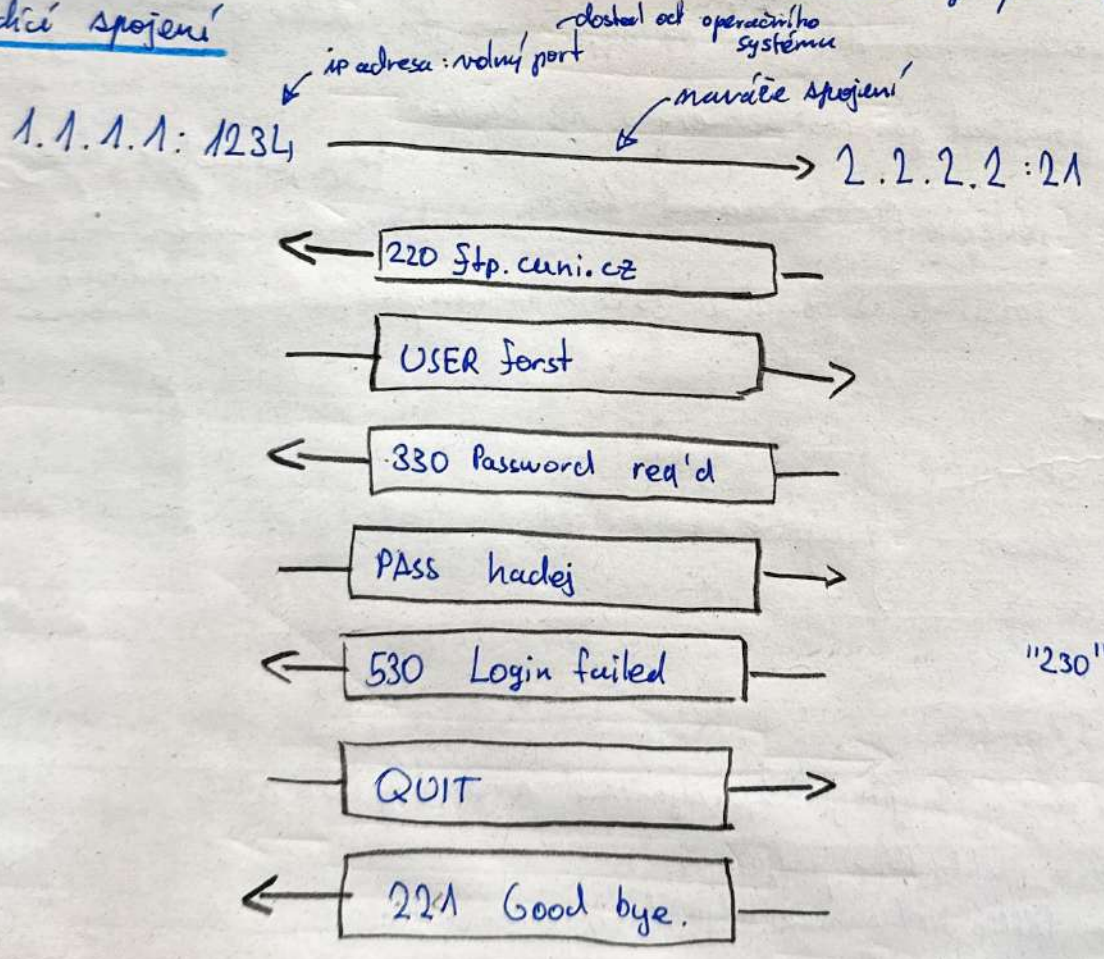
↳ jeden z nejstarších protokolů (RFC 959), ale platí dodnes

↳ původně: - vzdálený přístup k účtu, aby bylo možné přenášet soubory
- autentikace byla přenosem otevřeného hesla – bezpečnostní riziko

↳ dnes: - anonymní přístup k veřejně dostupným souborům
- uživatel poskytne jen e-mail ze statistických důvodů
↳ jako heslo

↳ přišlo kdysi funkci HTTP, ale i dnes některé servery používají protokol FTP

Řídicí spojení



"230" – povedlo by se login

↳ FTP začal rozlišovat odpovědi hromadným kódem → zjednodušuje komunikaci

Třídy odpovědí

↳ 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 požaduje doplňující 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 příkaz znovu zopakovat a je šance, že se provede

↳ např. přetížení 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 => stejné problémy

↳ přenos dat probíhá po jiném spojení než řídicí → u FTP, SIP

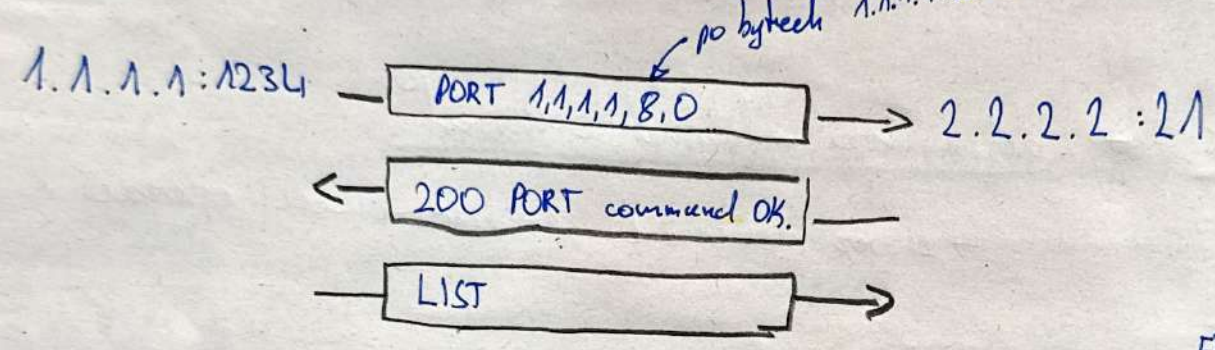
↳ existují 2 druhy, kde pasivní znamená, že server je pasivní a datové spojení navazuje klient

aktivní znamená, že server je aktivní a navazuje datové spojení na klienta

↳ používá se pasivní kvůli bezpečnosti

aktivní

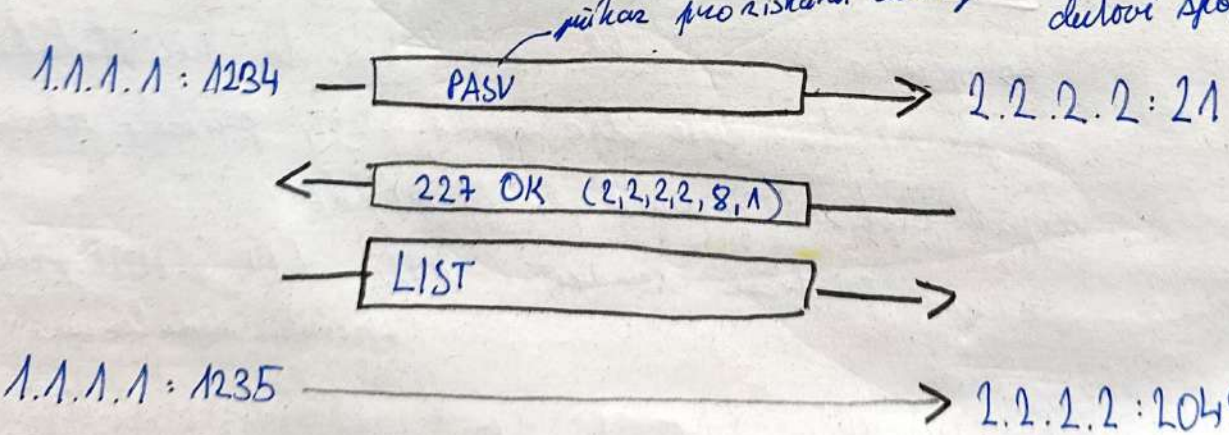
2048 = 256 · 8



FTP port 20 je pro datové spojení, 21 pro řídicí

↳ nebezpečné, protože klient začne přijímat spojení a ne otevírat. Firewall často nedovolí. Většinou se to dovolí na 5 sekund jen na určitý port a IP adresu z určitého portu a IP adresy

pasivní



↳ ty zprávy v datovém spojení úhrnují jen směr navázání spojení, ale data potečou na obě strany

Aplikace pro FTP

↳ WWW prohlížeče → umožňují 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 běh vlákna řídící kanál a datový kanál v ohni

↳ příkaz **ftp**

↳ příkazy v promptu

↳ navázání relace: open, user

↳ ukončení relace: close, quit, bye

↳ vzdálené příkazy: cd, pwd, ls, dir
(change directory
└ print working 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
└ stáhni └ nahraj └ multiple → pro více souborů

↳ typ přenosu souboru: ascii, binary

! CR LF → ! ↳ pozor mezi přenos různými OS → ascii → textový
binary → binární

↳ pomocné: prompt, hash, status, passive ...
↳ server připojuje ke klientovi → passive off a reverse
↳ aktuální stav FTP klienta
↳ pro zobrazení, co se děje
↳ ptát se před akcí?

LF → UNIX

CR → Macintosh

CRLF → Windows

Elektronická pošta

↳ SMTP protokol 25 ← port → textový protokol na TCP

↳ obecná služba, co existuje i mimo internet

↳ off-line předávání krátkých zpráv ← do 64kB i souborů

↳ diskusní kluby (mailing-listy, konference)

↳ off-line informační služby → mailem se poslalo: chci vědět, kde je tenhle soubor a po dlouhé době přišla odpověď

↳ RFC 821 → formát protokolu, pak 2821, 5321
↳ RFC 822 → formát zpráv, pak 2822, 5322 } dnes platí

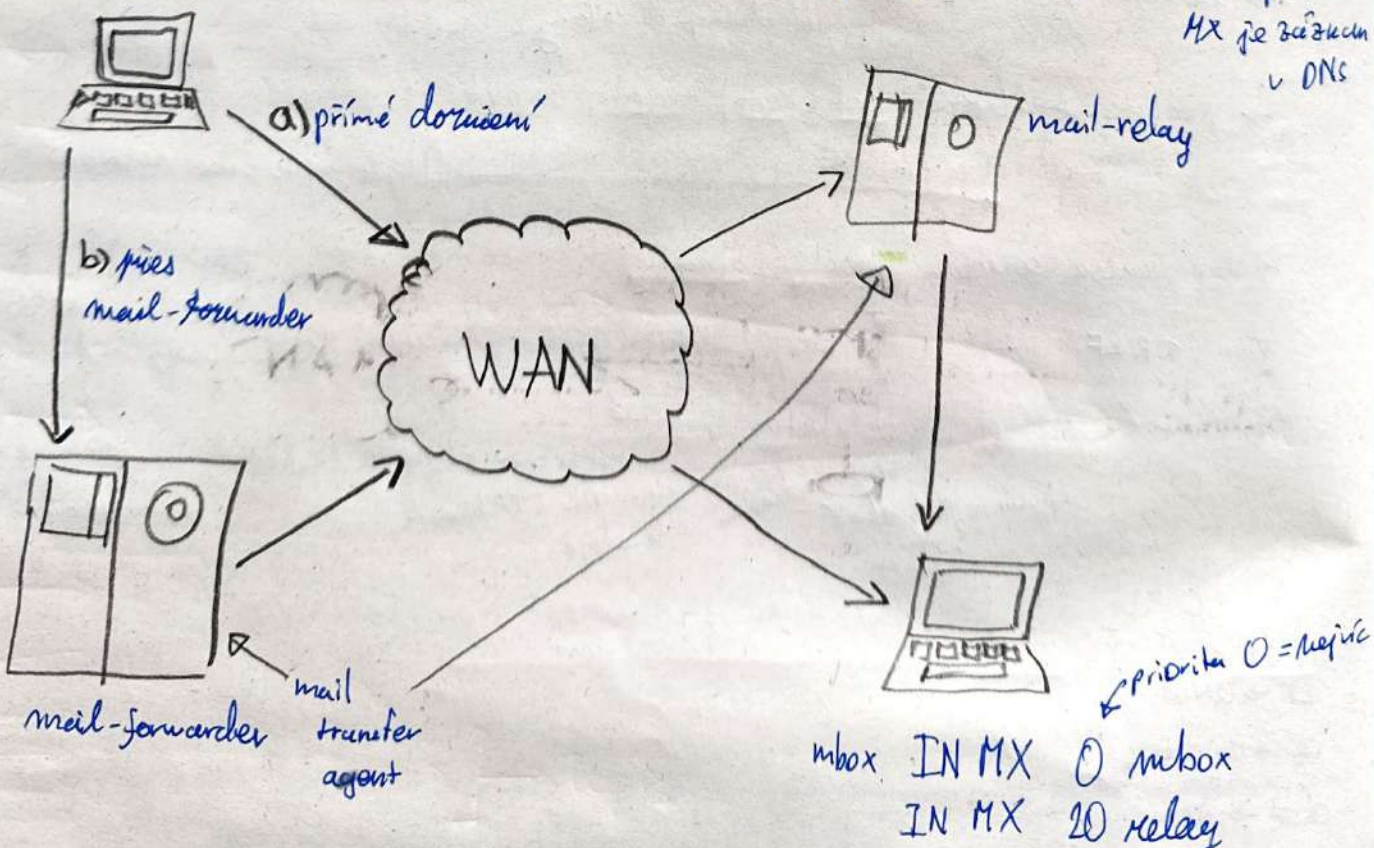
↳ mailová adresa:

login @ počítač → forst @ ms.ms.mff.cuni.cz
(mailserver mateřská strana ← nebezpečnější)

alias @ doména → Libor.forst @ cuni.cz

Příjem a odeslání pošty v SMTP

MX-definuje ne jakové stroje má mail přijít
MX je záznam v DNS



↳ a) přímé doručení

↳ mezi klientským počítačem a cílovým serverem (dalším počítačem) nejsou překážky a pomocí SMTP příkazů se mail předá

↳ dnes není časté

b) přes mail - forwarder

↳ na lokální síti se mail pošle na mail forwarder, který mám za úkol to poslat dál hned nebo se zpožděním a pak to přeposílá nebo rovnou pošle přes WAN a zkusí to poslat na mlox cílového serveru a pokud to nepůjde, tak to pošle na relay v síti, který to odešle na cílový server

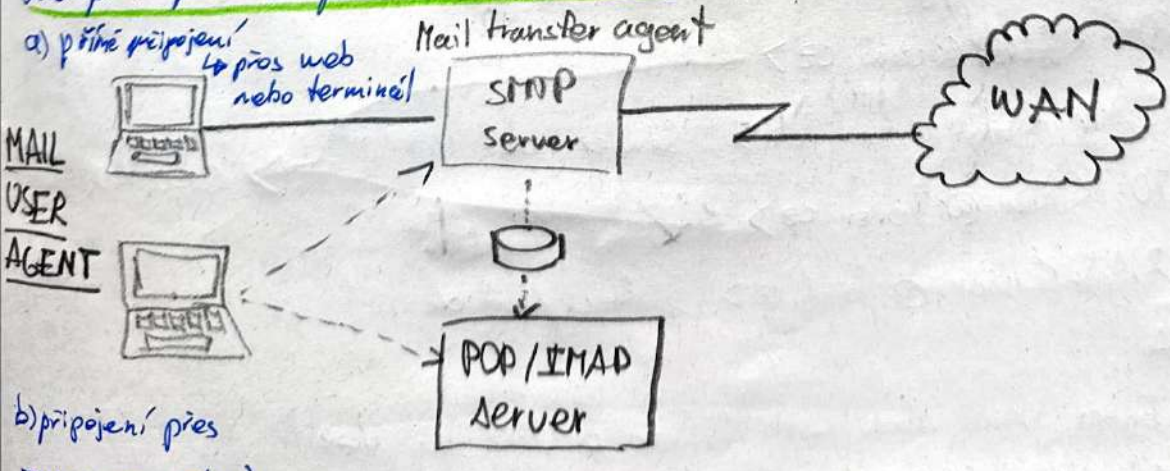
↳ MTA (mail transfer agent) je mail - forwarder a mail - relay

↳ dočasně odesílající MTA jako klient a přijímající MTA jako cílový server

↳ není vhodný pro větší soubory

↳ celá cesta mailu se dá vyjít z hlavičky

Přístup k poště z pohledu uživatele



a) přímé připojení k SMTP serveru

↳ přes web nebo terminál (SSH)
↳ přímý přístup k mailboxu a lokální služby MTA, takže rovnou zařazuje do odesílání zprávy do fronty MTA

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

↳ POP/IMAP → poštovní protokoly, co stahuje zprávy z SMTP serveru
↳ na odesílání není potřeba autentikace, na čtení ano
↳ open-relay → člověk se připojí zvenku na server a sám si pošle dopis ven, nedovolený → kvůli spamu
↳ doručení je: - z lokální sítě ven
- z venku nenávazat SMTP spojení (bez autentikace není email a heslo) a poslat email do/mimo sítě

Ukázka SMTP protokolu

↳ textový protokol

← 220 alfik.ms.mff.cuni.cz ESMTP Sendmail... ← může vždy rovnou napsat 500 a odmítnout spojení
→ 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>
→ ...
→ .

obálka

→ dopis

← 250 2.0.0 h9869Fxt Message accepted for delivery — může odmítnout, např. vir, moc velký atd.

→ QUIT

← 221 2.0.0 alfik closing connection

↳ jakmile převezme dopis, řekl 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 nevěsti odesílatel

↳ v hlavičce jsou informace o cestě, ale je řekne SMTP server o to požádá

↳ MAIL FROM → pak špičaté závorky, to samé u RCPT TO, řekl může být libovolně mnoho a server akceptuje každého zvládá

↳ Delivery Status Notification (DSN) → pokud se nepodařilo odeslat, řekl doslune odesílatel

↳ bez adresy → může projít bez kontroly viru, ale servery zakazují obecně

↳ DATA → pak dopis ukončen řečkou, jinak lze poslat 1 řečky a jedna se umaže, pokud je na řádce jen CR LF řečka

Elektronický dopis podle RFC 822

Received: from alfik.ms.mff.cuni.cz
by bobynka.ms.mff.cuni.cz ← kdo napsal
Date: Thu, 16 Nov 1995 00:54:11 +0100
To: student1@ms.mff.cuni.cz
MIME-version: 1.0 ← pro posílání souborů
Content-Type: multipart/mixed; boundary="=_XXX_="

From: Libor Forst <forst@cuni.cz> ← zapomněl
Subject: Test pošty
Cc: student2@ms.mff.cuni.cz

hlavička → striktně jen 7 bitů z ASCII 0x00–0x7F
↳ pro informaci a pro mailové programy

← prázdná řádka

z MIME
--=_XXX_=
Content-Type: text/plain; charset=Windows-1250
Content-Transfer-Encoding: 8bit

Čau Pepíčku!
...
--=_XXX_=

obsah dopisu
↳ kdysi řádky jen 7 bitů, ale v ESMTP protokolu se mohli domluvit na 8bit kódování
potřeba domluva

Hlavičky dopisů

Date → datum pořízení dopisu v americkém (pevném) formátu

From → všichni autoři dopisů, nejsou: • „komentář <adresa>"
• „(komentář) adresa"

→ ve spamu bývá více adres → protože si myslí, že to projde spíše pokud tam bude důvěryhodná 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 tito adresáti obdrželi dopis → Blind carbon copy

Message-ID → vyrobí odesílací program a příjemce použije při odpovědi, aby se dohledaly maily sdružovat

Subject → shrnutí obsahu dopisu, pozor jen v 7-bit ASCII jako zbytek hlavičky

Received → doplňuje MTA (= uzel), pokud dopis přijme a odešle dál (= pro sledování dopisu)
→ je tam název MTA, čas, lokální identifikátor atd.

Soubory a diakritika v poště

↳ kódování pomocí UUENCODE (z UUCP = „unix-to-unix copy")

↳ předávalé před internetem

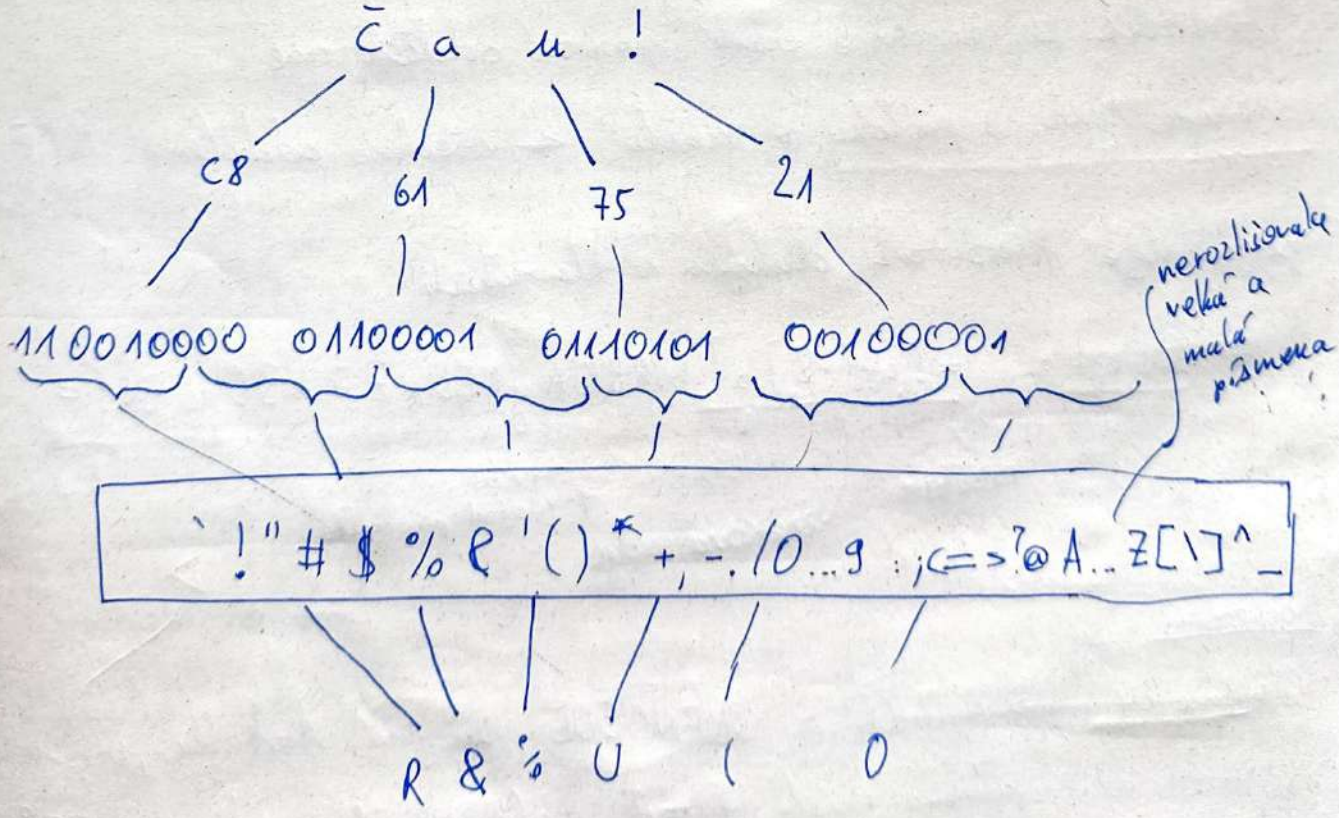
↳ převod binárního souboru (nebo textu) do textu ASCII 7-bit

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

↳ soubor začínal v textu dopisu → begin a končil → end

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

↳ zakódovaný soubor se zvětšil o 33 %



Multipurpose Internet Mail Extension (MIME)

↳ umožňuje strukturovat dokument, používá se i v jiných protokolech např. HTTP

↳ RFC 2045 – 2049

↳ dokument v MIME: obsahuje hlavičku a tělo

- ↳ v hlavičce je:
  - typ dokumentu → z hlavního typu a podtypu → text/html, image/jpeg
  - znaková sada
  - způsob kódování
  - původní název souboru
  - předpokládaný způsob zobrazení → jako přílohu, text, ...

↳ častý typ dokumentu: multipart

- ↳ rozdělí se na více částí pomocí oddělovače
- ↳ např. text a přílohy v mailu ← všechno oddělené MIME dokumenty

↳ umožňuje používat diakritiku v hlavičkách

- ↳ Subject: =?utf-8?b?SVRBVCAyMEDxZ7A30gc96?=
  - =? … začátek
  - utf-8 … znaková sada
  - b … kódování – b-base64, q-quoted printable
  - ?= … konec

Kódování

↳ Base 64 → vychází z UUENCODE, jen jiná tabulka, viz. velká, malá písmena + čísla + + /, ale pořád zvětšuje o 33 %

↳ Quoted-Printable → převážně pro jazyky s latinkou, jinak 300 % originálu
→ normální text, občas "=HH", což je hexadecimální hodnota znaku

Etika poštovního styku → RFC 1855 (Netiquette Guidelines)
Net

↳ přečíst všechny maily, než odpovíte
↳ neodpovídat hned pokud je v Cc
↳ nechat příjemci čas na odpověď, ale později se připomenout
↳ odpovídat rychle jako potvrzení
↳ správná volba subjektu
↳ zkontrolovat adresáty
↳ nezachovávat kolik původního textu v odpovědi
↳ respektovat ©
↳ soubory zvážit, zda poslat, raději na dropbox
↳ podepsat se

Bezpečnost pošty (uživatel)

↳ dopis je otevřená zásilka, proto je potřeba šifrovat obsah dopisu → např. Pretty Good Privacy (PGP)
↳ programem - šifrování veřejným klíčem a odšifrování privátním
↳ ten symetrický klíč

↳ někdy není jistý odesílatel ani údaje v obálce a textu
↳ protože díky otevřenému protokolu tam může kdokoliv dopsat cokoliv
↳ částečné řešení: Sender Policy Framework - data sorce name
↳ poštovní server pošle DSN zprávu uvedeným odesílateli, vlastně kontroluje které servery smí odesílat za které domény
↳ řešení: • systém výzva/odpověď → takže doručí na výzvu, že dopis odeslal
• elektronický podpis → pomocí šifrování

Bezpečnost pošty (klient, server)

↳ open-relay → správný server jen odesílat maily lokálních uživatelů kamkoliv a maily odkudkoliv jen lokálním uživatelům
→ jinak ostatní servery zablokují komunikaci
→ odkudkoliv pošle kamkoliv ⇒ open-relay

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

↳ STARTTLS → příkaz v ESMTP, který zahájí SSL/TLS spojení mezi dvěma servery prošifrování spojení, jinak je šifrování spíš problém uživatele

Ochrana proti spamu

spam → inzerce nebo obtěžování lidí ← spam enginy to neograbují
grey-listing → na RCPT TO
na první pokus odpoví 450, neúspěšně a pokud
vědný dopis, leč klient se znova pokusí o doručení
se sledným tripletem → a pak email dojde, potvrdí 250, běžně po 15-ti minutách
<klient, sender, recipient> → pak uloženo v paměti na měsíc, bez zpoždění

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

DomainKeys Identified Mail (DKIM) → odesílací stroj podepíše dopis a přijímající server to zkontroluje
→ forward není problém, protože každé MTA může mít vlastní klíč

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

→false positive

Otázky:

1) Jaká jsou nejdůležitější bezpečnostní rizika FTP?
2) Co je podstatou problému při otevírání dodatečných datových kanálů?
3) Vysvětlete pojmy MTA a MX a jejich roli v přenosu pošty.
4) Co je „obálka" a jak souvisí s obsahem hlaviček dopisu?
5) Jak lze používat non-ASCII znaky v hlavičkách 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. HODINA

Post office protocol

problémy
↳ otevřené posílání hesel, pro šifrovanou autentifikaci APOP

↳ dopisy je třeba stahovat celé, TOP - stahne jen hlavičky, ale ručně implementovaná
↳ omezené možnosti práce s mailboxem
↳ nahrazuje se IMAP protokolem

↳ aktuální verze 3

Internet Message Access Protocol (IMAP)

↳ modernější než POP
↳ port 143

↳ výhody:
↳ zabudované šifrování → přes port 993 (TLS), nebo přes příkaz STARTTLS
↳ vyhledávat v dopisech
↳ podpora stahovat jen hlavičky

↳ používá dnes každá aplikace

IMAP4 protokol

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

→ 1 LOGIN Sorst heslo
← 1 OK User authenticated
→ 2 LIST "" "*"
← * LIST (Has No Children) "/" "INBOX"
← 2 OK LIST COMPLETED
... atd.

↳ musí mít SMTP spojení pro odesílání

Princip distribuované databáze

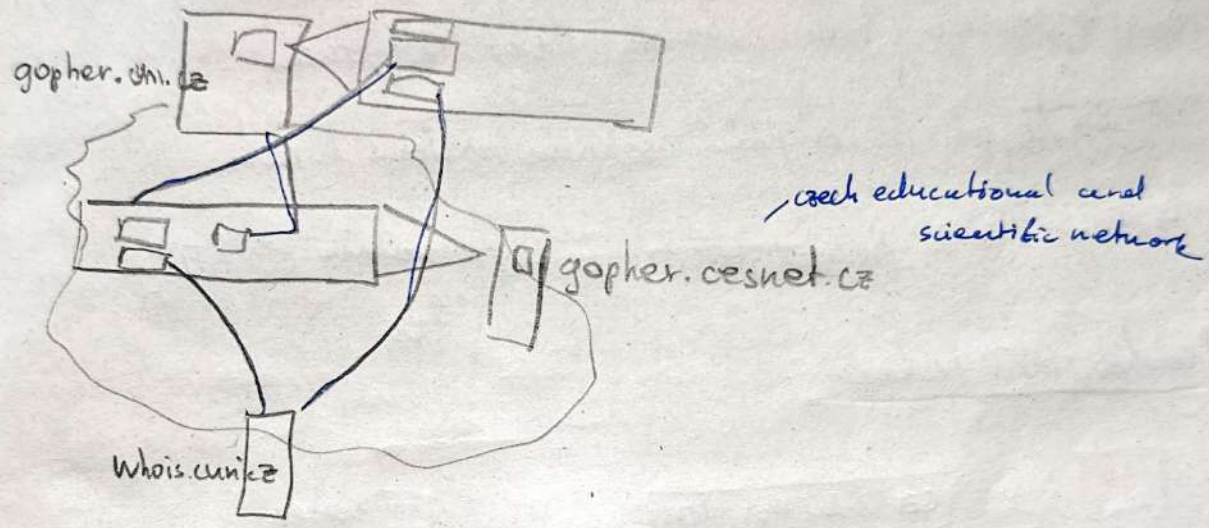
↳ Gopher

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

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

nebylo vědět, jestli se přesunul na jiný server

pomocí formuláře



czech educational and scientific network

↳ pár let po gopheru se objevil web

Hypertext

↳ oproti gopheru přidal místo textové, čistě hypertextovou databázi

↳ 1945 - první myšlenka

↳ text odkazy na vysvětlující/přidružený text

↳ 1965 - rozšíření

↳ o interaktivní informace → obrázky, zvuk, video

↳ 1989 - praktický

↳ WWW z Cernu

World Wide Web

↳ distribuovaná hypertextová databáze

↳ základní jednotka je hypertextová stránka (dokument)

↓

posílá na žádost klientům (statické)

on demand stránku vygeneruje z URL požadavky (dynamické)

↳ ↑ jazyce HTML

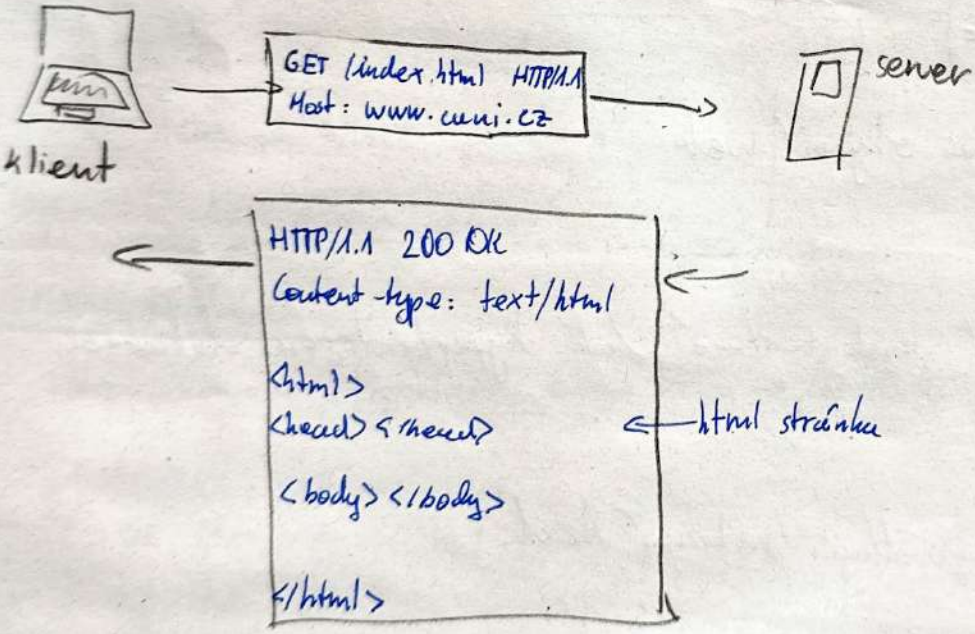
↳ popisuje obsah - obrázky, text a formu

↳ to, jak to vypadá rozhoduje rozšíření klienta

↳ přenos stránek pomocí HTTP se šifrováním → HTTPS

↑ TLS

Protokol HTTP (nekódem)



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)

- tělo dokumentu - volitelné

Kódy odpovědí:

1xx → informační (požadavek přijat k zpracování)
2xx → hlavní
3xx → další info od klienta potřeba (přesměrování)
4xx → chyba na straně klienta
5xx → chyba na straně serveru

Metody HTTP

METODA	TĚLO POŽADAVKU	TĚLO ODPOVĚDI	Bezpečná (nemění stránku)	Idempotentní (opakované použití má stejný efekt)
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 taky stránku html s chybou

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

POST - řeknu o dokument, ale posílám data, html formuláře → GET/POST (data v query URL)

PUT - posílám dokument, nežádám si o stránku, měním server, ale efekt by měl být stejný

DELETE - smaže dokument, dostanu zpětnou zprávu

CONNECT - vytvořit spojení v jiném protokolu → bezpečnostní riziko

Vlastnosti HTTP v. 1

↳ na jeden požadavek, jeden dokument (stránku, obrázek)

↳ 1 GET → 1 stránka a 3 obrázky

↳ na každý obrázek nový GET

↳ v HTTP ver. 1.1 → vytvořeno perzistentní spojení, kde lze posílat více požadavků

→ občas otevřou více kanálů (klienti), aby mohli poslat více požadavků GET najednou → mají. by obrázky

↳ komunikace je bezestavová → by čtyři GETy jsou nezávislé operace

↳ proto si servery ukládají cookies (nenesou kód, žádná hesla) → aby to vypadalo stavově:

↳ server vytvoří identifikaci a uloží si data o uživateli a pošle je do hlavičky

↳ při každém požadavku ukládá do hlaviček

↳ pro shromažďování informacích o uživateli

↳ bezpečnostní riziko sekundární → cílená reklama

→ informace získá jiná osoba

Hypertext transfer protocol v.2

→ pomalu nej nej přichází

↳ úplně nový protokol, změnil z textového na binární → ono je to vlastně jedno, protože je to stejně zase převedené

↳ motivace: větší propustnost → prostě rychlejší načítání

↳ metody:

↳ server pushne více, než klient požádal, protože očekává, že to stejně bude žádat → styly, obrázky

↳ prioritizované streamy v rámci jednoho TCP, aby se neblokovaly

↳ komprimace hlaviček → při požadavcích se stejně nebude jazyk kódováním měnit, takže se neposílají pořád

↳ povinné šifrování neprošlo

Jazyk HTML

↳ značkovací jazyk

↳ nyní verze 5 (odstavec, adresa, tučně)

↳ značky → strukturální, sémantické, formátovací

str. 89 - 98 jsou HTML stejdy v prezentaci

Dynamické stránky (server)

↳ dynamika řízená na serveru, ne u klienta (u klientů neběží kód)

Dynamické stránky (klient)

↳ přes skripty, v javě → ten převede na meziкód a podle klienta ho interpretuje jazyk, co má na počítači, takže knihovny pro každý jazyk na interpretaci → řeší javascript

Telnet

- protokol pro vzdálené přihlášení k počítači, **port 23**
- Telecommunication Network, RFC 97 ; dnes už RFC 2946
- přenáší přes síťový virtuální terminál (NVT) - emulace terminálu
  - přenáší oběma směry znaky a příkazy
  - musí se domluvit na posílání -> kdo udělá co
- otevřený přenos dat -> řeší až RFC 2946, dnes ale raději SSH
- Dnes využití:
  - ladění jiných protokolů
  - přístup na síťová zařízení v rámci LAN -> nehrozí odposlouchávání hesel
- problém s terminálem -> není příznak příkaz nebo odpověď
  - protože někdo musí zobrazit znak pokud se stiskne klávesa echo
  - aby může se zacyklit -> DO ECHO / WILL ECHO
    - pokud jeden server zapomene, že to už poslal

Secure Shell (SSH)

- bezpečná náhrada pro přihlášení ke vzdálenému počítači nebo přenos souborů
  - klient důsledně ověřuje server, pokud ale po měsících přihlašování řekne mám nový klíč -> upozornit
    - man in the middle
    - jinak nový klíč - netřeba ověřovat a uloží se
  - komunikace je šifrovaná
- port 22, aktuální verze 2, -> RFC 4250-4254
- SSHv2
  - více kanálů paralelně zabezpečených
  - tunelování -> jiný provoz jiným zabezpečeným kanálem
  - souborový systém -> SSHFS
- putty, WinSCP -> SSH aplikace Windows, na Unixu -> ssh a scp příkazy

Bezpečnost SSH

↳ klient ověřuje server
  ↳ zkontroluje klíč (zkontroluje uživatel)
  ↳ certifikátem (přes autoritu)

↳ server ověřuje klienta
  ↳ pomocí hesla
  ↳ použití výzev a odpovědí (OTP)
  ↳ pomocí klíče → server pošle zašifrovanou zprávu veřejným klíčem a uživatel pošle plain text

↳ strategie používání klíčů
  ↳ ověřit klíč serveru, man-in-the-middle
  ↳ zaheslovat si klíč tak, že pokud někdo získá můj stroj, tak nemá přístup k serverům, protože nezná heslo
  ↳ reciprocita (A → B, B → A) → internetoví červi, kdy chtějí veřejný klíč za přihlášení → většinou nejde, ale může se stát, že někdo nechá přihlášení zapnuté i naopak

Voice over IP

↳ není protokol, ale obecně jakýkoliv nástroj, co přenáší hlas přes TCP/IP, ne třeba Skype přes HTTP
↳ protokoly patřící do standardu H.323
  ↳ další (jiné, nekompatibilní):
    - SIP standard
    - proprietární

→ používané pro VoIP

↳ problémy:
  ↳ digitalizace hlasu, nalezení partnera, možnosti (obou) více zařízení a propojení s běžnou telefonní sítí - tedy použití čísla tel.

H.323

↳ šetřily každým místem → binární protokoly, po bitech (↙ založeno na ASN.1)

↳ vytvořila ITU – international telegraph union

↳ dílčí protokoly: (nepotřeba)

- H.225 /RAS (Registration /Admission /Status) pro vyhledávání partnera
- Q.931 (síťová vrstva ISDN) → navazuje spojení
- H.245 → 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 mnemonickým identifikátorům

↳ zapisuje do tolika bitů kolik potřeba → předchází bit co signalizuje kolik bude bitů

→ z 80. let = tragédie

↳ používá v H.323 a X.509

Session Initiation Protocol (SIP)

SDP ← session description protocol
↳ textový protokol
↳ keyword = value

↳ jednodušší než H.323

↳ nahrazuje navazování spojení mezi zařízeními, dohodu o vlastnostech řeší SDP

↳ textový protokol narozdíl od binárního → podoba se HTTP

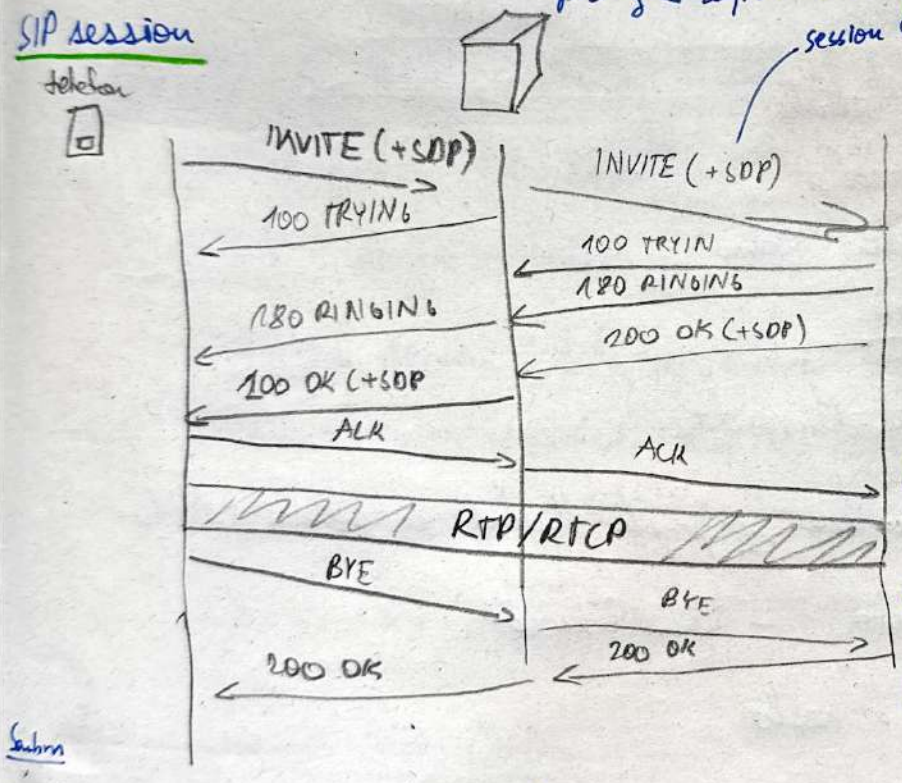
↳ port 5060 pro UDP i TCP

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

↳ přes běžnou telefonní síť přes registrátora

proxy ← zajímavost, že tam je, nahrává běžně

SIP session



aby vše fungovalo
↓
, tohle proxy upraví pomocí NAT

- zašle INVITE + SDP
  ↳ doraží na proxy, která řeší nalezení cíle
  ↓
  přes úseky a další proxy
  ↳ 180 ringing → end-to-end, proto proxy jinak 100 trying je next-to-node
  ↳ 200 OK + SDP
  ↳ proxy zase upraví
  ↳ pak ACK

Cvičení

1) K čemu slouží IMAP protokol?
2) K čemu slouží HTML?
3) Popište, co se odehrává v HTTP 1.1 protokolu, pokud uživatel požádá o stránku se dvěma vloženými obrázky?
4) Stručně popište účel a vlastnosti nejznámějších metod 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 systému souborů

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

↳ Network File System (NFS)

↳ mezi nejpoužívanější v UNIXu, verze 2049

↳ AFS - Andrew File System → náhrada

↳ server : cesta ← odkaz na připojený disk

↳ vyvinul v Sun Microsystems → dnes IETF

↳ autentikaci přes Kerberos

↳ umí UDP (pravděpodobně), umí TCP záložka

↳ kvůli rychlé reakci

↳ RPC - volání, XDR - prezentační - jedno

↳ Server Message Block (SMB)

↳ vyvinulo IBM, převzal Microsoft

↳ Samba → open implementace, propojila Microsoft se světem UNIXu

↳ UNC → \\ server \ cesta - identifikace

↳ autentikace → jméno a heslo

Network Time Protocol

↳ aby na všech počítačích v síti byl stejný čas

↳ důležité:

↳ hlavně při sdílení souborů → timestampy souborů

↳ logy při řešení problémů sítě

↳ z stratum 0 (atomové hodiny)

↳ kvůli přenosu a zpoždění

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

↳ NTP funguje tak, že je tam latence uvnitř stratum serverů a chyby Marzullaově algoritmu se dá spočítat interval

BOOTP a DHCP

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

↳ Bootstrap Protocol (BOOTP)

↳ RFC 951

↳ protože disky byly drahé, tak se kupovali počítače bez disků, pročež nebylo možné si uložit konfiguraci a IP adresu → problém

↳ řešením byl BOOTP → počítač poslal do sítě broadcast, kdy všem svojí fyzickou (MAC) adresu síťové karty - kdyby vypadená v kuloáru

↳ BOOTP server odpověděl 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 zpětky klientovi

→ normálně nedovoleno, aby se broadcast šířil po internetu

DHCP (Dynamic Host Configuration Protocol) - automatické alokace IP adres, v nastavení mobilů

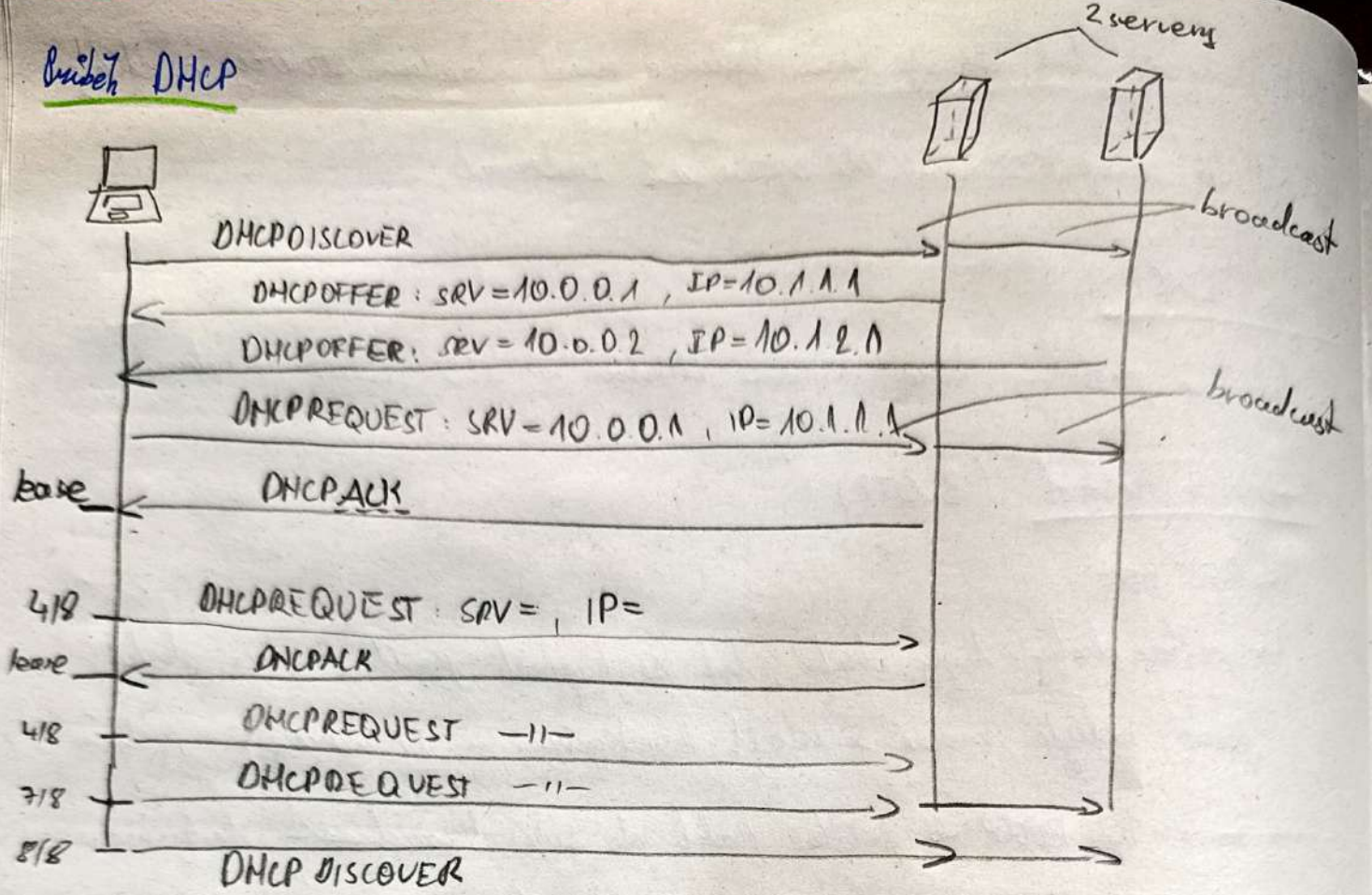
↳ stroj neobdrží fixní adresu podle MAC adresy, protože jich bylo omezený počet a kdy se přiřazuje dynamicky

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

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

↳ zpětně kompatibilní s BOOTP

Průběh DHCP



DHCPDISOVER – broadcast žádost o IP adresu

DHCPOFFER – servery v síti pošlou nabídky

DHCPREQUEST – říkám serverům, aby dali vědět, co si berou za IP adresu ← nejčastěji podle toho, kterou už měl ← uvolnil ten nejdřívější

DHCP ACK – potvrzení od serveru

DHCP REQUEST – v polovině 4/8, pak 7/8, aby si prodloužil dobu pronájmu
↓
rady už broadcastem

Prezentační vrstva (OSI 6)

↳ vrstva, která řeší rozhraní dat pro přenosu

↳ všeobecný model s kódováním → datové typy a datové struktury

↳ ASN.1 → v H.323
→ snažilo se být univerzální, ale bylo moc složité

↳ TCP/IP přeneslo dekódování na programátory do aplikací (aplikační vrstvy)
↳ prostě zaroláním Network to host a host to network → jak je to uložené v síti

↳ problémy:
↳ CR LF na konci řádek → $\emptyset x \emptyset D$ a $\emptyset x \emptyset A$
↳ BE a LE → zapisování bytů

Relační vrstva (OSI 5)

↳ řízení dialogu mezi aplikacemi, nehledě na to kterou aplikaci

↳ třeba RPC v NFS (zajímavost)

↳ TCP/IP zase přeneslo zodpovědnost do aplikační vrstvy (protokolů)

↳ např. SMTP → (jedním) do TCP (spojení) předávání více mailů než jeden otčas

SIP → 1 řídící spojení + 2 datové → video se enkóduje

Transportní vrstva (OSI 4)

↳ za end-to-end spojení a přenos dat podle požadavků → TCP (potvrzí doručení)

→ UDP (jen odeslání potvrdí)

zabezpečuje → TCP → spolehlivost přenosu dat

↳ multiplexing

↳ pomocí portů lze identifikovat protokol, službu

↳ segmentace dat → TCP

↳ rozdělí na bloky, které opět složí pokud přizpůsobuje velikost

↳ řídí tok → TCP

↳ pro efektivnější přenos ovlivňuje, co se kdy odešle

↳ strana může požádat o pomalejší přenos dat

Transportní vrstva v TCP/IP

TCP (Transmission Control Protocol)

↳ pro spojované aplikace

↳ klient navádí spojení a data běží ve formě proudu → stream

↳ TCP řídí a zabezpečuje doručení → proto jsou aplikace jednoduché

↳ po odeslání musí čekat na potvrzení nebo než vyprší timeout, takže pomalejší než UDP

UDP (User Datagram Protocol)

↳ pro nespojované služby

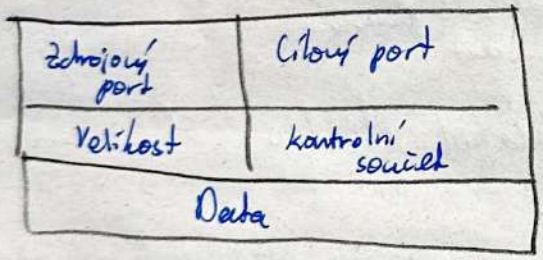
↳např. odesílání mailu

↳UDP je jednoduché → všechy aplikace musí být složité a řešit doručení a spolehlivost

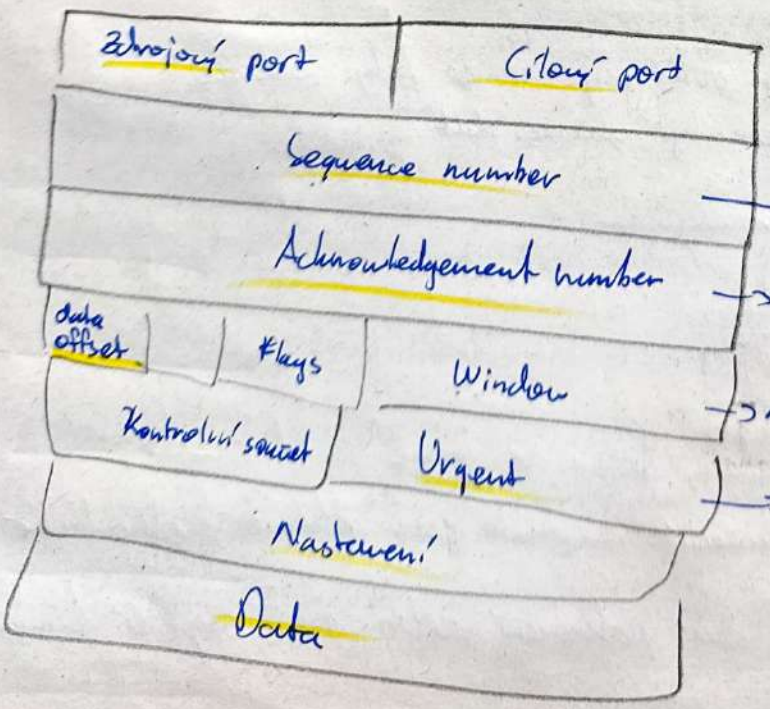
↳ale rychlejší, protože nemusí čekat na potvrzení

- SCTP, DCCP, MPTCP — multipath TCP
  - Stream Control Transmission Protocol
  - datagram congestion control protocol

Struktura UDP datagramu



Struktura TCP paketu



Data offset → pro identifikaci segmentů

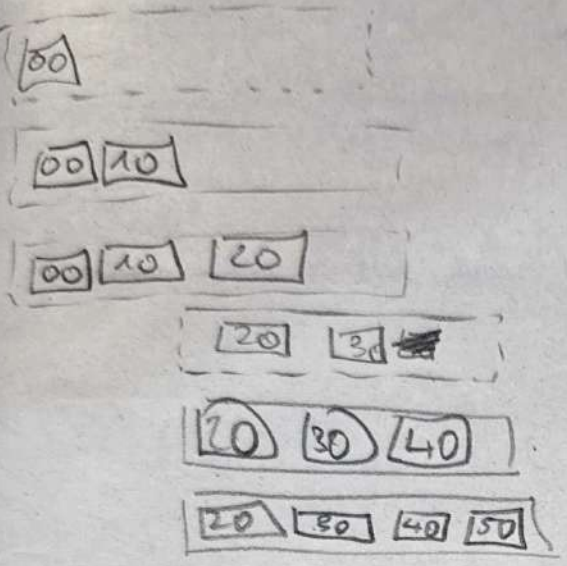
→ na identifikaci

→ na potvrzení → posun od začátku komunikace → potvrzuje kolik přijal a může poslat dál

→ řízení rychlosti

→ důležitost

TCP okno



00 →

10 →

20 →

← ACK 20

30 →

40 →

50 →

?

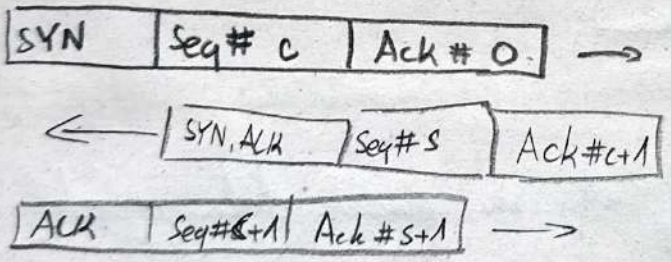
20 → přijelo 60 B

← ACK 60

↳ posírá se okno místo čekání na každý ACK pro každý paket

Zahájení a ukončení spojení

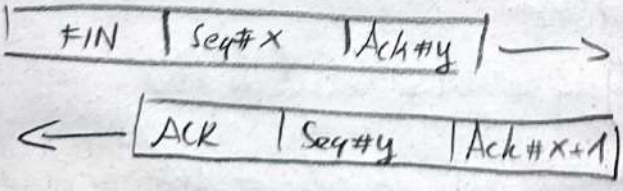
navázání → three-way-handshake



SYN → synchronizace sequence number od náhodného čísla → ne od nuly pro horší napadení

↳ 3 pakety

ukončení → jednostranné



FIN → ukončení, že nebude posílat data, ale může přijímat

→ např. GET v HTTP, uzavře spojení a počká na film i guess

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

TCP příznaky

SYN → synchronizace, s sequence number

ACK → acknowledgement

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 → třeby, jak vypadá spojení TCP

↳ v prezentaci, not important?

Výpis existujících socketů

↳ pro zobrazení všech otevřených spojení -> netstat

↳ jsou tam sloupy - protože TCP je stavové

Síťová vrstva (OSI 3)

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

adresace → adresa počítače atd.

zapouzdření (encapsulation) → data, adresy se musí vložit do PDU ← přičíst

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

forwarding (přeposílání) → předání paketů dál k uzlům, aby se dostal k počítači

decapsulation → vyjmutí dat, které se předají transportní vrstvě

↳ protokoly: IPv4, IPv6, IPX, Appletalk

Internet protokol (IP)

- nespojovaná služba → datagramy se doručují nezávisle
- nespolehlivá → spolehlivost řeší vyšší vrstvy
- nezávislá na způsobu přenášení (médiu) → nemusí řešit použitou technologii

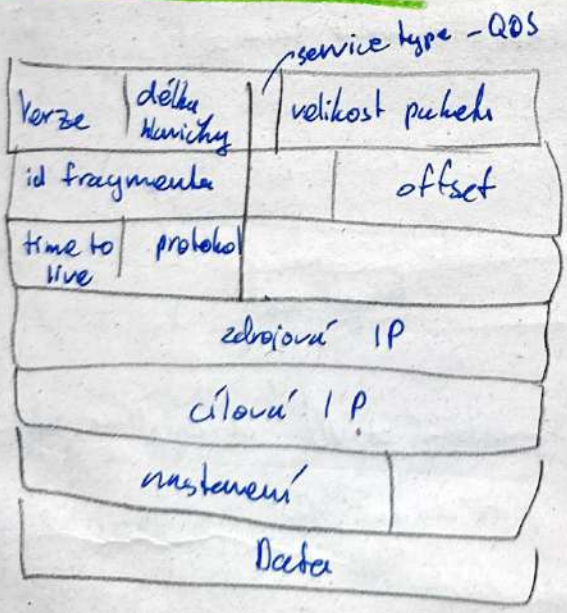
Adresy:

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

přidělování adres:

- IANA (Internet Assigned Numbers Authority), součást ICANN (Internet Corporation for Assigned Names and Numbers)
- nadřazená správa sítě → privátní (správce sítě)
  → veřejné (ISP)
  - nad ISP právě IANA
- IANA má pod sebou 5x regionálních registrátorů, RIPE NCC
  - pod nimi ISP
  - každému dá blok adres

Struktura IPv4 diagramu

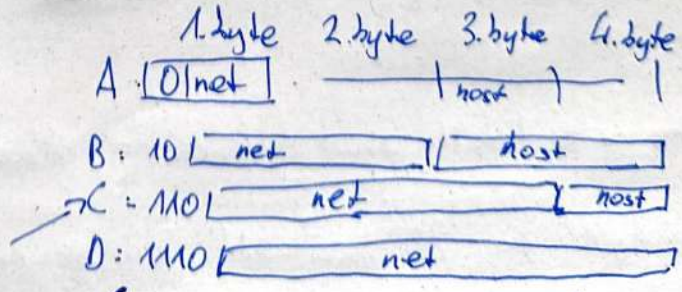


IPv4 adresy

4 byty

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

. . . . } multicast experimental



používá se s malým počtem počítačů

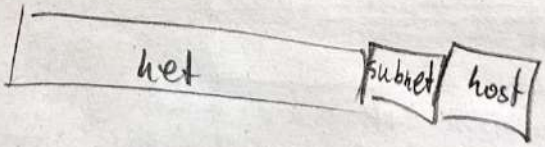
Speciální IPv4 adresy

- 0.0.0.0
  - použitelná jako zdrojová, když je třeba komunikovat, ale ještě nezná svoji adresu, třeba DHCP dotaz
- 127.0.0.1 / loopback
  - pro mě, takže komunikuje se jako na síti ale jen v počítači, klidně bez síťové karty
- adresa sítě: <adresa sítě> <samé nuly>
- network broadcast: <adresa sítě> <samé jedničky> → všem v síti sítě
- limited broadcast: 255.255.255.255
  - všem v téhož 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 přiřadit stanice sama
  - nelze 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 lomítkem počet bitů, co tvoří síťovou část
- 6 podsítí po 30 počítačích, protože nedoporučuje all zeros a all-ones
- VLSM → variable length subnet mask
- dnes nepoužívání, protože nahromadění privátních adres

↳ supernetting → směrování → spojování sítí

Jádro internetu

↳ přeplňují se směrovací tabulky:
↳ sítě co spolu sousedily zkly agregovat
↳ Classless Inter Domain Routing (CIDR), ale muselo se mainově přečíslovat síti
↳ 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 → dnes link-local
→ unique-local fc00::/7 → dnes site-local, takže privátní adresy IPv4
↳ multicastová → ff00::/8 → náhrada broadcastových adres
↳ anycastová → unicastová adresa přiřazena více uzlům
→ např. servery po světě → měření nejvhodnějšího
↳ tunelování IPv4 a IPv6

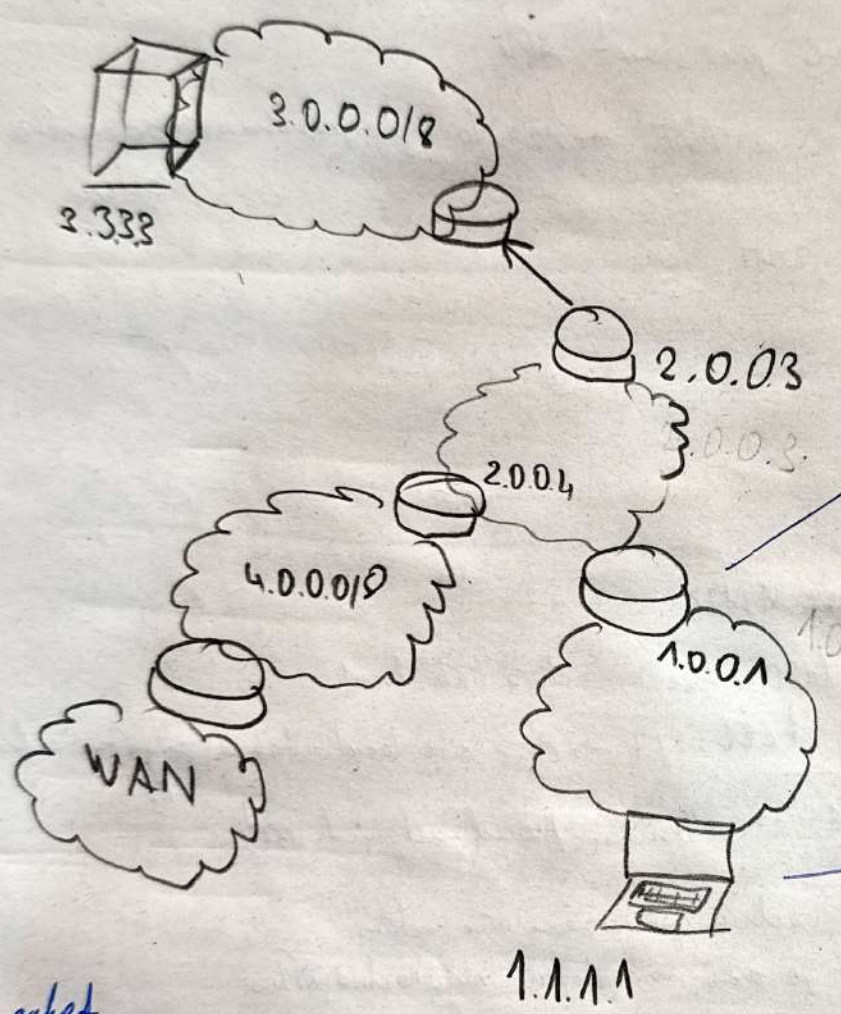
Otázky:
1) Proč a jak se synchronizuje čas na počítačích v síti?
2) K čemu slouží DHCP?
3) Jak se v TCP/IP sítích chybějící prezentační vrstva?
4) rozdíl mezi TCP a UDP
5) pojmy potvrzování a TCP okno. Jak souvisí?
6) three-way-handshake?
7) Jak souvisejí třídy IPv4, subnetting a CIDR?
8) Proč je v síti C pouze 254 adres pro počítače? → samé nuly a samé jedničky, odpověď
9) Jaké typy broadcastových adres známe?

7. HODINA

Směrování (silnice) – vlastně pořád síťová vrstva

↳ rozhoduje se podle křižovatek

Směrování (sítí)



cíl	next-hop router
Destination	Gateway
3.0.0.0/8	2.0.0.3
4.0.0.0/8	2.0.0.4
default	2.0.0.4

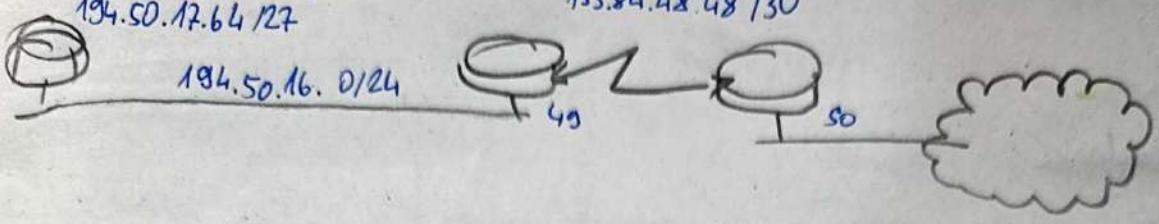
Destination	Gateway
default	1.0.0.1

paket

↳ hledá next-hop router pomocí směrovacích tabulek

↳ paket chce například na adresu 3.3.3.3 → porovnává vždy prefix s maskou 18 například a podle toho se vydá jinak default

Základ směrovací tabulky



Destination	Mask	Gateway
127.0.0.0	255.0.0.0	127.0.0.1 (direct) → loopback adresa
194.50.16.0	255.255.255.0	194.50.16.71 → adresa karty C, na hlavní síť LAN
193.84.48.50	255.255.255.255	193.84.48.49 → stroj připojený k 1 routeru, jediný v LAN
194.50.17.64	255.255.255.224	194.50.16.77 → ukazuje na podsítě
default	0.0.0.0	193.84.48.50 → router → ven

Principy směrování

↳ každá stanice v TCP/IP síti

↳ ve směrovací tabulce: - cíl (destination)

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

↳ záznamy:

implicitní - jsou po vytvoření sítě

explicitní - manuálně zadané

dynamické - vytvoření používáním sítě

Směrovací algoritmus

↳ najdi vyhovující záznamy → nenajde? → paket nelze doručit, ale nepadá program

musíte mít bude obsahovat aspoň default

↳ pak vybere destination s nejširší maskou

↳ direct → pošle příjemci

↳ indirect → pošle na next-hop router } doručit se na linkové vrstvě

Konfigurace sítě

UNIX

- IP adresa : ifconfig
- defaultní router: route add default router
- DHCP : dhclient

Windows

↳ nějak v nastavení, nebo ipconfig a route

Internet Control Message Protocol

↳ 3. vrstva síťová OSI → protokol ICMP → k posílání řídících IP informací

↳ ping, echo, echo reply

↳ destination unreachable → nedostupný stroj, pro UDP důležité

↳ time exceeded → time to live překročen

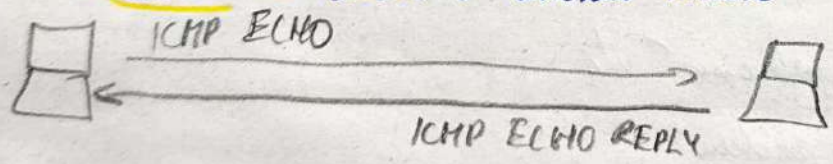
↳ posílání zprávy protokolu NDP → neighbor discovery protocol
→ zjistit stav okolní sítě

↳ source quench - žádost o snížení rychlosti

Ping ↳ router solicitation - vyhledávání routerů

↳ redirect - k změně v routovací tabulce

↳ Parameter problem - chyba v hlavičce IP datagramu



↳ pro diagnostiku sítě - dostupnost uzlu

↳ ping → posílá periodicky ICMP echo dokud nedorazí odpověď

↳ nezaručuje dostupnost služeb → jen síťovou vrstvu

↳ ping, pong několikrát, a pak se udělá statistika latence

↳ nemusí jsem mít speciální software

Time To Live (IP)

→ v DNS je TTL sekundy a znamenají kolik sekund ještě záznam platí

↳ Time Exceeded

↳ není o čase!, měří se v hopech

↳ při dosažení 0 se pošle time exceeded

↳ chrání dlouhé v routovacích tabulkách → aby se nezacyklilo donekonečna

↳ každý router sníží TTL a tím mění obsah IP hlavičky

Diagnostika směrování

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

↳ kontrola cesty → traceroute, tracert

↳ pošle packet TTL 1 → dostane zpět první router
TTL 2 → dostane zpět druhý nexthop router
...
→ poslední směrování je většinou problém

Statické řízení směrovacích tabulek

↳ cesty se nastaví při startu ← defaultní

↳ například DHCP → dostane router a nic jiného nepotřebuje

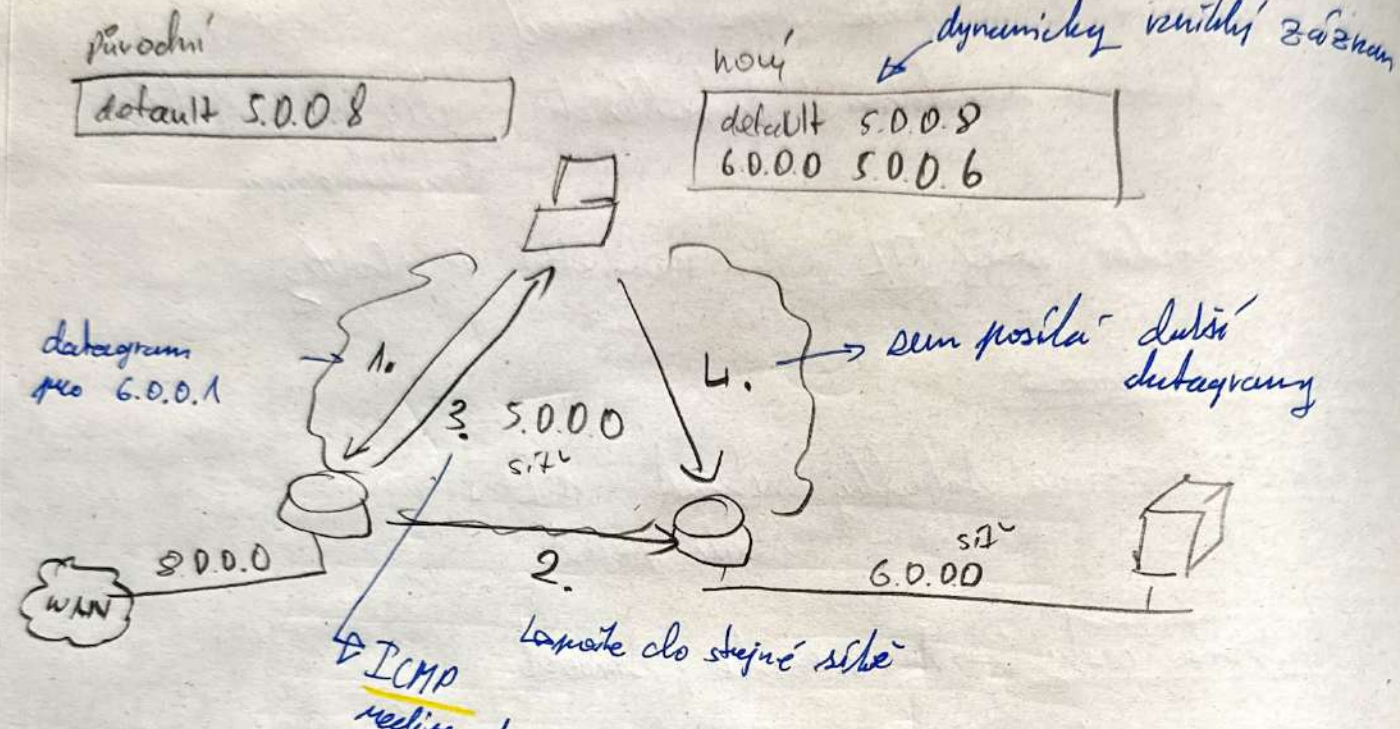
↳ ale nepružné při změnách, subnettingem nejde, šířce katalogu je

↳ dají tak jen v malých sítí

Redirekce

↳ další ICMP paket

↳ používá se k tomu, když existí, že routovací tabulky jsou špatně



ICMP redirect, že si má počítač změnit routovací tabulky

↳ co když ale ICMP redirect je falešný? → pak je potřeba to zakázat

Dynamické řízení směrovacích tabulek

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

↳ síť se dokáže sama opravovat

↳ jednodušše se změní konfigurace a automaticky se aktualizuje

↳ ale citlivější na útoky

↳ protokoly: routed, gated, BIRD

↳ dělí se na - distance - vektor protokoly (RIP) (např.)

- link - state protokoly (např. OSPF)

Distance vector protokoly

↳ hledá pokud neexistuje něco blíž než dosavadní cesta, má k ní uložené i „vzdálenosti" → počet hopů

↳ pak informuje sousedy a ty si to taky upraví atd.

↳ jednoduché

↳ pomalu reaguje na chyby, nezohledňuje vlastnosti linek (↳ rychlost, cenu, spolehlivost), omezení velkých sítí, chyba ve výpočtu ovlivní celou síť

Routing Information Protocol (RIP)

↳ velmi starý distance vektor protokol

↳ počet hopů omezen na 15 → špatné pro jednodušší sítě

↳ používá Bellman-Fordův algoritmus → ne nejrychlejší Dijkstrův

↳ protože kopíruje počet ze směrovací tabulky, jen ho trochu upraví než aby spouštěl celý algoritmus znova

↳ protože funguje díky zprávám od ostatních uzlů

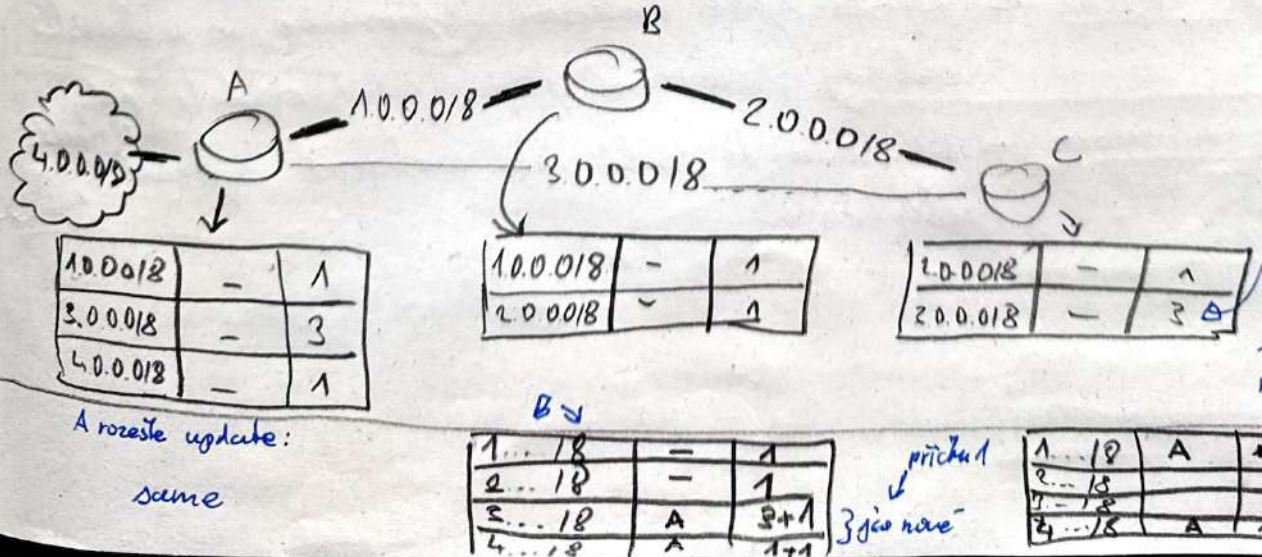
↳ verze 2

↳ umí subnetting s proměnlivou maskou (VLSM)

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

↳ funguje na všech systémech

Metrika a kvalita linek

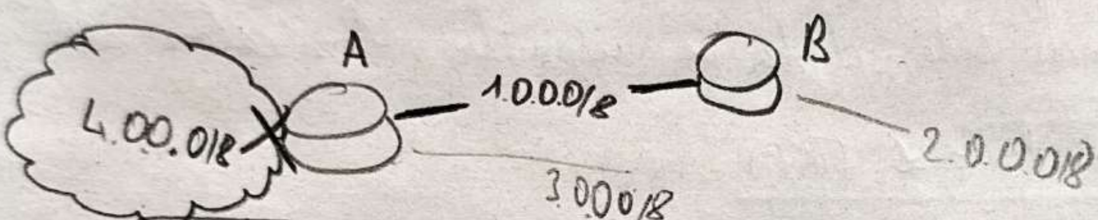


B rozešle update

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

↳ pokud mají lepší cestu, tak si updatnou tabulku se vzdáleností

Counting to infinity



1..18	–	1
2..18	B	2
3..18	–	3
4..18	–	1

1..18	–	1
2..18	–	1
3..18	A	4
4..18	A	2

↳ vypadne linka na 4.0.0.0/8 → tak se odráží tam a zpátky a v tabulce růstne 16, neboli ∞

↳ 7×30 sec → proto je ∞ tak malý, protože to trvá půl minuty

↳ řešení techy:

- triggered updates → posílá update okamžitě
- split horizon → neposílá update partnerovi informace, co se dozvěděl od něj, ale musí připravit celý update zprávy (samostatně a ne kopírovat)
- poison reverse → s metrikou 16 posílá data partnerovi, vylepšení split horizon

Link-state protokoly

↳ každý router zná celou mapu sítě

↳ při updatu si sdělí stav linek a každý router si aktualizuje svoji mapu sítě

↳ nevýhody: pomalejší a náročnější

↳ výhody: - dobře neovlivní ostatní

- reagují dobře na změnu sítě
- nemusí si vyměňovat celé své databáze
- rozdělit na menší podsítě

Open shortest path first (OSPF)

↳ link state protokol, nejrozšířenější

↳ používá Dijkstrův algoritmus nalezení nejkratší cesty

↳ při průchodu grafu se přidá jeden uzel do neustálených vrcholů a hledá max. kolik má vrcholů

↳ rozhlédá se na oblasti → aby se nemusel algoritmus použít pro celou síť

↳ hierarchický model sítě:

↳ area 0 → páteř

↳ ostatní oblasti jsou připojeny na páteř

↳ každý router zná cestu k páteři a mapu své oblasti

↳ path cost

↳ lze přiřadit, určuje administrátor podle šířky pásma, ceny, propustnosti

Autonomní systémy (AS)

↳ systém, jak zjednodušit routování na celém globálním internetu

↳ na globálním internetu se pracuje s bloky adres uskupeny do autonomních systémů

↳ uvnitř AS se používají ty interní routovací protokoly

↳ definice: množina sítí, 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 nezacyklil, tím, že tvoří posloupnost AS a nejde přes tu, kde už byla → path-vektor

↳ v ČR nejsou stovky

IP filtrování – firewall

↳ filtrování na routeru, který připojuje síť k internetu (intranet / internet), tedy na transportní vrstvě, v TCP/IP, metoda dovolování provozu na základě portů

↳ stanoví pravidla, tedy které porty jsou dovoleny z a do sítě

↳ přísná → tedy kdo naváže spojení, kdo pošle SYN paket v TCP

↳ ven vybrané, dovnitř nic

↳ např. HTTP, SMTP ok

↳ FTP, SIP → více kanálů, problém ← musí být spolupráce s aplikační vrstvou

↳ obvyklé

↳ ven všechno, dovnitř nic

↳ u protokolů s mnoha kanály → SIP nepoužitelné

↳ u FTP s aktivním přenosem problém

↳ DMZ (demilitarizovaná zóna) → lépe chovali jak k nebezpečnému internetu

↳ pro provoz www serveru nebo pošty, kde se otevírají kanály dovnitř

↳ zóna, kde je přístup zvenku v pohodě, ale oddělený od sítě, dnes hosting

Proxy server

↳ software, co kontroluje provoz určitého protokolu z vnitřní do vnější a obráceně

transparentní
↳ chová se jeho 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 podpora pro daný protokol

bezpečnostní → filtruje operace na aplikačním protokolu
výkonnostní → pokud odejde více požadavků, tak místo, aby odesílala všechny, tak odešle jeden a pak rozešle požadavky v síti tím, kdo požádali
30 dětí → 1 stránka → 30 odeslat → 30 přijmout, místo toho jen jednou

Otázky
1) Jak funguje směrovací algoritmus?
2) Popište typy sloupců a různých směrovací tabulky
3) Čím se liší statické a dynamické řízení správy směrovacích 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 ovnečuje jejich IP filtrování?
10) K čemu slouží proxy servery?

8. HODINA

Address Resolution Protocol

↳ umožňuje překlad 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` → vypsání cache

↳ síťový může poslat gratuitous ARP - tedy nevyžádanou odpověď

↳ nelze ověřit správnost odpovědi

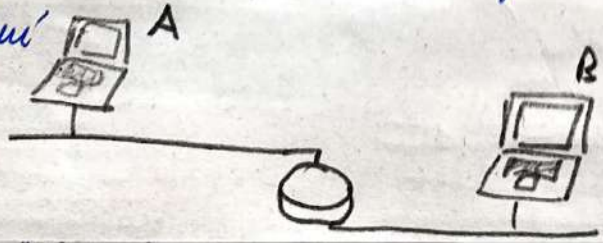
↳ může být více MAC adres v síti oddělené routery

Proxy ARP

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

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

↳ jak se stane, že router má v ARP cache pro stejnou IP adresu víc MAC adres → OK pokud je tam proxy, jinak je to nepoužití



Linková vrstva (OSI 2) → množina uzlů sdílejících stejné médium

↳ 2 podvrstvy:

LLC (logical link control)

↳ multiplexing, řeší různé síťové protokoly se dokáží uložit a pak předat po linkové vrstvě

MAL (Media Access Control)

↳ adresaci a řízení, kdo kdy smí něco posílat v rámci sítě na ní linkové vrstvě

↳ datová jednotka linkové vrstvy → frame (rámec) ← PDU

↳ obsahuje: - synchronizační pole

↳ určitá sekvence bitů, co umožní cílovou stanici a hlavně odlišit se od šumu, který tam normálně je

- hlavičku

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

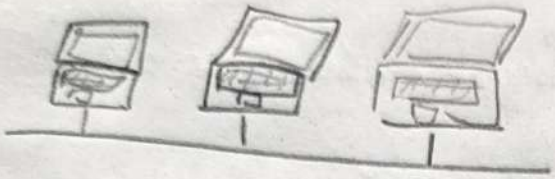
- data

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

Typy síťových topologií

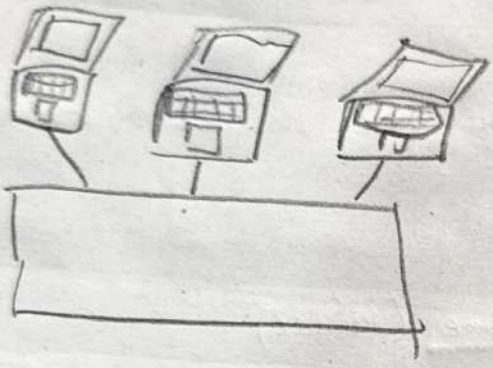
Multipoint

↳ komunikuje spolu více počítačů najednou



← Sběrnice

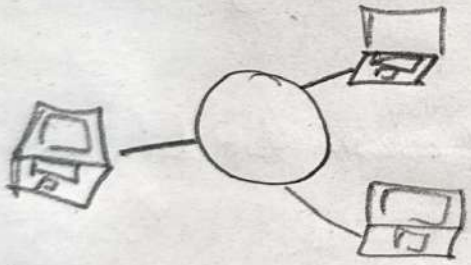
↳ spojena kabelem, ale odesílání více zpráv mohlo způsobit kolizi
↳ poškození kabelu zničilo síť
↳ koaxiální kabel



← Hvězda

↳ centrální prvek, porouší se jen kabel, ne celá sběrnice →
↳ fyzicky - tam může být sběrnice
- to, co doopravdy
logicky - to je hvězda
- to, jak spolu uzly komunikují

Wi-Fi →



← Kruh

↳ kabel jde kolem dokola mezi počítači
↳ Token-ring, FDDI
↳ žeton posílá a kdo ho zrovna má, může vysílat

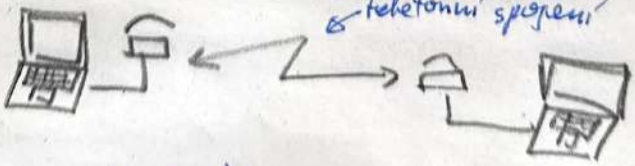
Point-to-point

↳ komunikace mezi dvěma počítači

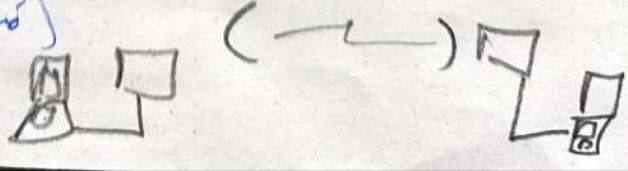
↳ přímé připojení kabelem např. RS-232 →

↳ Ethernet → plná sběrnice

↳ propojení přes modem



↳ bezdrátové připojení (laser, radiovlnění)



Způsoby řízení přístupu k médiu

Multipoint

↳ deterministický způsob řízení

↳ někdo řídí, kdo smí a nesmí vysílat

↳ token (žeton) nebo někdo řídí centrálně

↳ nedeterministický

↳ nikdo neřídí, takže můžou vzniknout kolize

Point-to-point

half-duplex

↳ nedokáže současně přijímat a vysílat

full-duplex

↳ síťová karta

↳ dokáže vzeh současně přijímat i vysílat, tak se nastaví 2 simplexní přenosné linky

Řešení kolizí

CSMA (Carrier sense with multiple access)

↳ nosná volná

↳ uzel sleduje jestli na médiu neprobíhá přenos a pokud ne, začne vysílat, ale nezaručí, že nebudou k. lize, protože se můžou 2 uzly rozhodnout vysílat současně

CSMA/CD (Collision detect) -> Ethernet

↳ během vysílání detekuje kolize a kdy tak rozhodí přenos a počká náhodnou dobu, která roste exponenciálně při každém pokusu

↳ zvětšuje se interval

↳ ale doba vysílání rámce musí být delší než doba šíření mezi uzly, aby uzel detekoval kolizi a rámec se neztratil

↳ max. délka segmentu a min. délka rámce

CSMA/CA (Collision Avoidance) -> Wi-Fi

↳ pošle se rámec pokud je nosná volná, odešle rámec a čeká na ACK, pokud nepřijde, tak exponenciálně čeká

Ethernet

↳ vznikl ve firmě Xerox, převzalo IEEE

↳ CSMA/CD → po 16 s čekání chyba

↳ adresy: 6 bytů

↳ 3 byty prefix výrobce, 3 byty adresa

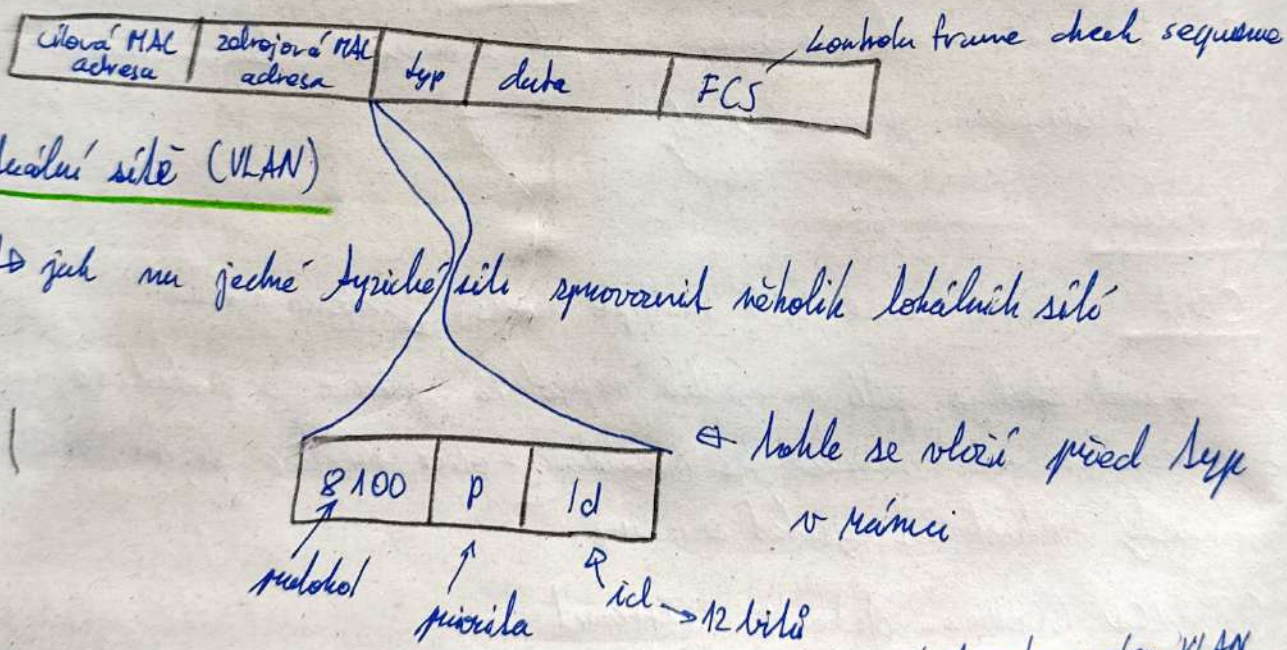
↳ dříve vypálená, dnes se dá změnit

Standardy IEEE 802.3

↳ pak to vyhrál ethernet, dohlížel držel krok s HW

↳ dnes optický kabel → 100 Gb/s

Struktura ethernetového rámce



Virtuální sítě (VLAN)

↳ jak na jedné fyzické síti spravovat několik lokálních sítí

↳ tohle se vloží před typ v rámci

↳ stroj pošle rámec, switch k tomu přidá tag, pošle dál po síti, dojde na jiný switch a za ním už je koncové zařízení, tak odebere tag a pošle, protože o tom zařízení, virtuálně není

→ pohled porty pro ten VLAN

↳ router musí vidět všechny VLANY

↳ do trunk portu pošle neotagované data

Cyklický kontrolní součet (CRC) - cyclic redundancy check

↳ v Frame Check Sequence

↳ hashovací funkce založena na dělení polynomu, protože ho lze udělat HW

1 1 0 → ... + 1.x^{29} + 1.x^{27} + 0.x^{26} + ..

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

↳ vydělím a vezmu zbytek po dělení, to sedí FCS a na konci to zase přepočítá a zjistí zda se shoduje / neshoduje

↳ dokáže detekovat chyby velmi dobře

Wi-Fi

↳ není protokol, je to označení technologie, rady standardů protokolů IEEE 802.11

↳ různá pásma 2,4 GHz; 5 GHz

↳ různé rychlosti 1 až 600 Mbps

↳ na hvězdicové topologii

↳ CSMA/CA

↳ SSID → service set id, pro rozlišení sítě

↳ problém zabezpečení

Fyzická vrstva (OSI 1)

↳ přenos signálu po kontrolním médiu

↳ bity na analog a obráceně

↳ metalické: el. pulzy (kabely)

↳ optické: světelné pulzy (optika)

↳ bezdrátové: vlny

Druhy přenosu dat

modem : D -> A (modulator / demodulátor) - bity
codec : A -> D (coder / decoder) - zvuk třeba

↳ vše je analogový, jen je potřeba interpretovat a převést

↳ baseband

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

↳ tzv. Manchester přenos - používá Ethernet, ve slotu stoupne -> 1
klesne -> 0

↳ broadband

↳ vysílá vlna, co se moduluje

↳ AM, FM -> změna amplitudy vlny

Nestíněná kroucená dvoulinka (UTP) - unshielded twisted pair

↳ kolik je tam drátů -> 8 vodičů

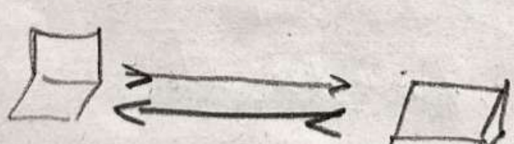
↳ 4 páry Cu vodičů pravidelně zakroucené

↳ zakroucení sníží vyzařování a příjem elmag. záření

↳ protože na obou kabelech se tvoří stejná změna a ta může pak odečíst

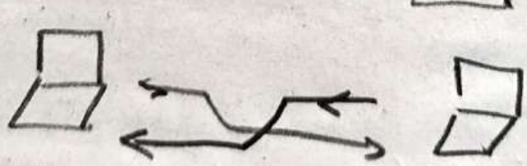
↳ 100Mb Ethernet používá jen 2 páry, protože lze rozdělit

↳ 2 různá zařízení



přímý

↳ 2 stejná zařízení



křížený kabel

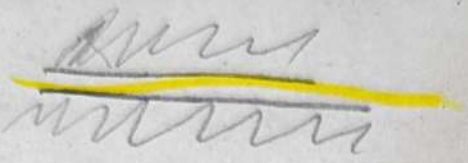
↳ ale oni se dohodnou protokol mezi karty MDI / MDIX

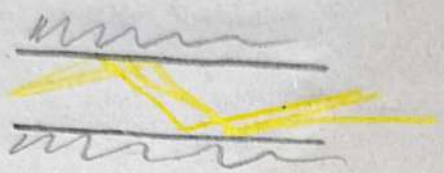
STP -> kabel s kovovým stíněním
-> shielded twisted pair

Optická vlákna

↳ velká šířka pásma, vysoká frekvence a žádné rušení

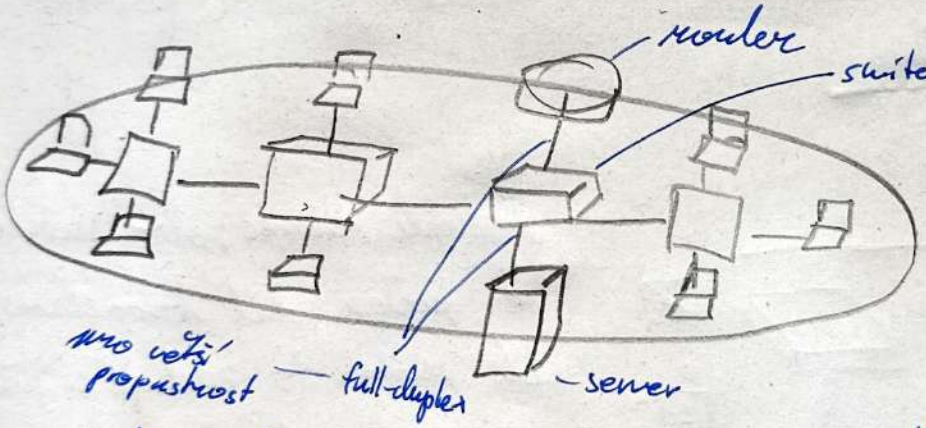
↳ nehodlat do kabelen, vyšší cena, horší manipulace

↳ jednovidová  – laser, větší dosah a přenosová rychlost

↳ multimode  – více lámou, pomalejší, LED

↳ vlákno z křemíku

Segmentace sítě



repeater – hub – rozbočovač -> horší propustnost & více kolizí, protože mu to trvá

↳ distribuuje fyzický signál aniž by mu rozuměl, zvětší dosah

bridge (most)

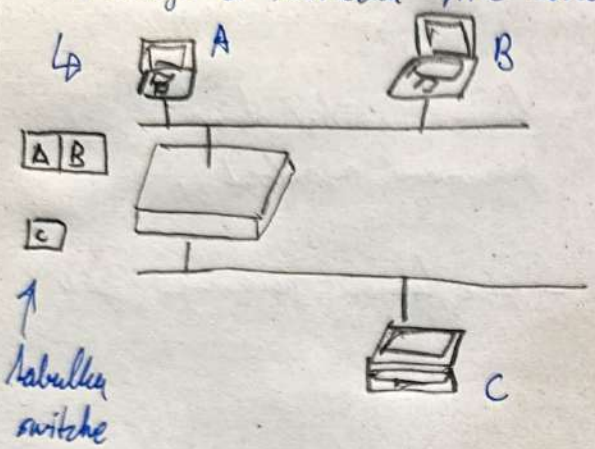
↳ zlepšuje propustnost, protože rozumí MAC adresám a různě mu linková vrstva posílá kam, kam je to určené

↳ tzv. switch

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

↳ tabhle umí switche

↳ udržuje si tabulku MAC adres stanic



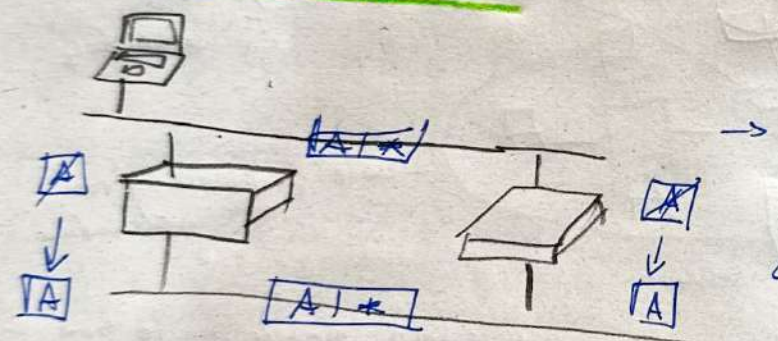
↑ tabulka switche

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

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

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

Spanning Tree Algorithmus



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

↑ důvodem je cyklus

↳ řešením je nejít acyklickou množinu → kostru (spanning tree)

↳ STP (spanning tree protocol)

↳ některé switche jsou v režimu blocking, kdy neposílají rámce, ale jen monitorují, zda nevypadl druhý switch (o je ve stavu forwarding a kdy rámce přeposílá

↳ pomalé → proto počítač může udělat faststart a zaručí, že se nevytvoří cyklus

Otázky:

1) Popište účel a bezpečnostní rizika protokolu ARP.

2) Popište účel vrstev 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 médiu.

8) Popište typy metalických a optických kabelů

9) Co je learning bridge a STP?