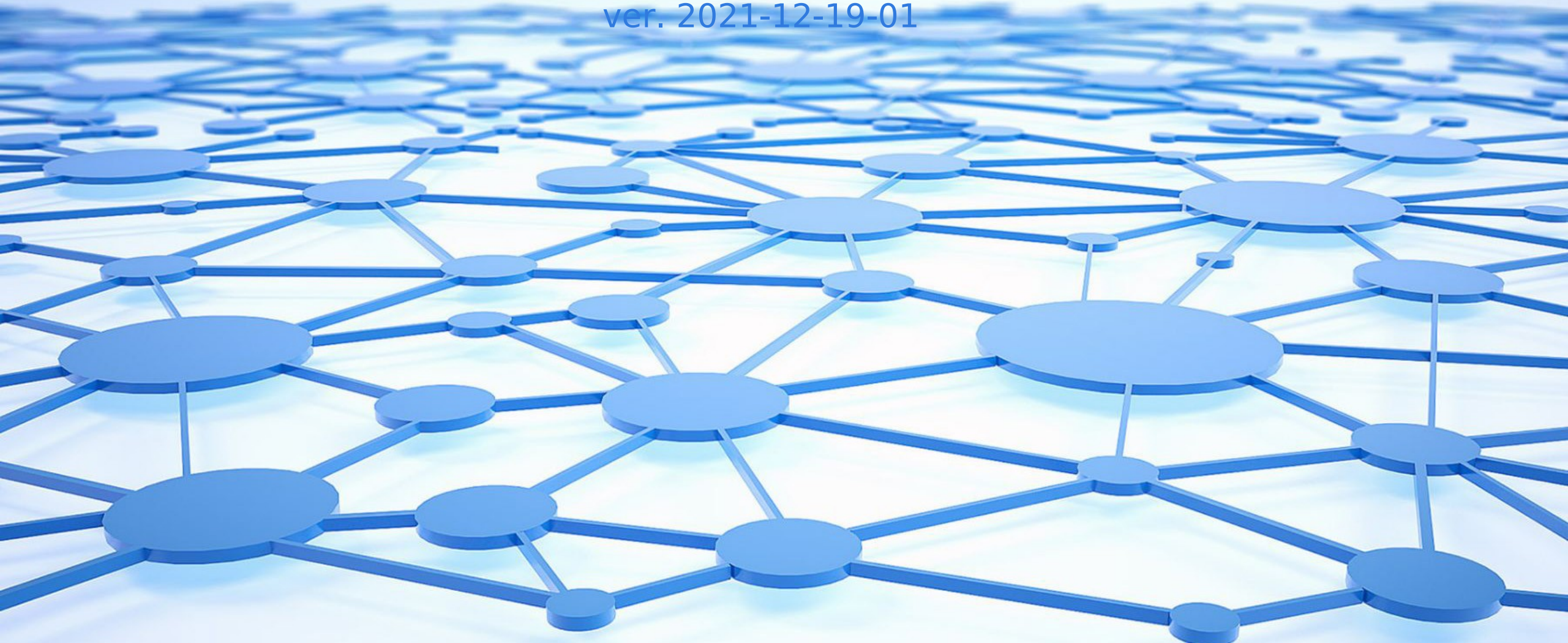
Úvod do počítačových sítí

Přednáška 11

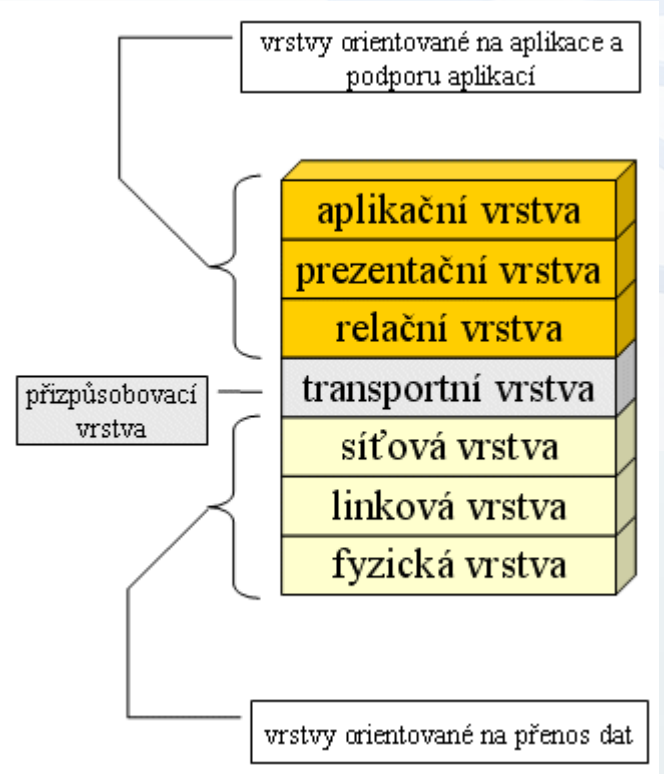
(2021/2022)

ver. 2021-12-19-01



L5 – Relační vrstva
L6 – Prezentační vrstva
L7 – Aplikační vrstva

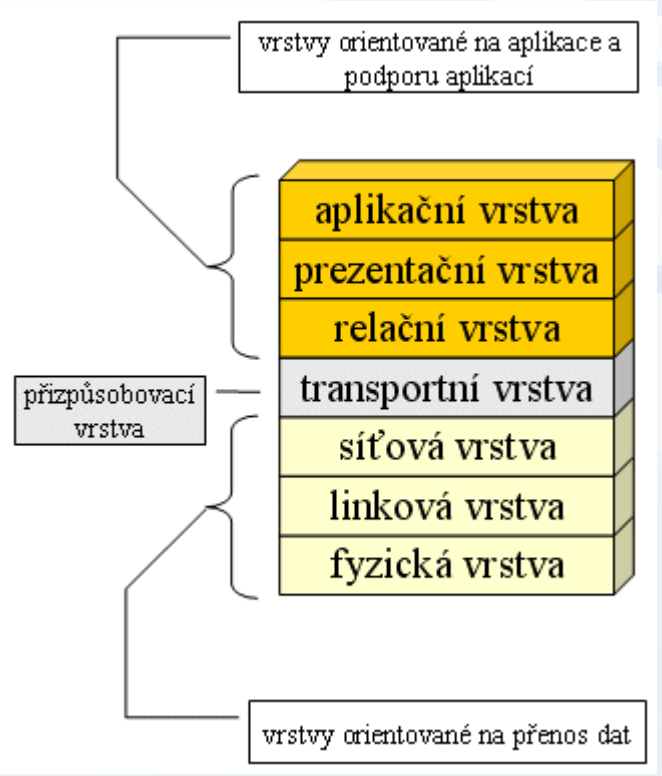
- Vrstvy orientované na podporu aplikací
- L5 - Relační / session vrstva
  - Podpora služeb pro řízení relace – dialogu
  - Reálně dnes není implementována
- L6 – Prezentační vrstva
  - Prezentace dat – transformace dat z formátu, který potřebuje aplikace do formátu vhodného pro přenos sítí
  - Reálně dnes není implementována
- L7 – Aplikační vrstva
  - Vrstva umožňující síťové fungování aplikací
  - Postupem času výrazně zobecněná podoba



zdroj: https://www.earchiv.cz/b06/b0300001.php3

L5 – Relační vrstva

- Relační / session vrstva
- Zajišťuje spolehlivý přenos mezi aplikacemi
  - Tedy může opět překrývat nedostatky nižší vrstev
- V rámci ISO/OSI byla vrstva navržena, ale reálně je dnes její funkce zajišťována až na aplikační úrovni
  - Tedy funkce této – a prezentační vrstvy také – jsou třeba a jsou požívané, ale jsou realizované v konkrétních aplikacích různě, nikoliv prostřednictvím služeb relační vrstvy
- Základní úkoly
  - Tvorba, udržování a ukončování relace – session
  - Řízení dialogu
  - Realizace synchronizačních bodů pro přenos
  - Řízení transakcí
- Příklady protokolů relační vrstvy
  - X.225, X.215
- Příklad protokolu, který relační vrstvy „nahrazuje“ v rámci TCP/IP může být například RPC



zdroj: https://www.earchiv.cz/b06/b0300001.php3

L5 – Relační vrstva: Udržování relací

- Zajistit udržení relace pro dvě komunikující aplikace bez ohledu na reálné chování L4 vrstvy
  - Jedna relace v jednom L4 spojení
    - Nejjednodušší situace – navážu jedno spojení – a realizuji jeden HTTP request – například
      - Není nic před a nic po – relace si odpovídají 1:1 a tedy není „téměř“ třeba nic řešit
  - Jedna relace je realizována ve více L4 spojeních
    - Například v e-shopu koš – dáte položku do koše v rámci jednoho TCP spojení, ale pak odejdete od PC a až se vrátí vaší TCP spojení je již díky timeoutu uzavřené, ale svůj košík stále chcete
      - Řešení pomocí session na úrovni aplikace / nebo cookie nebo kombinace, ale vše na aplikační vrstvě
  - Více relací prostřednictvím jednoho L4 spojení
    - Tedy zas v rámci e-shopu například možnost změnit identitu – odhlášení a přihlášení na jiného uživatele
      - Opět reálně realizované na L7 v rámci session / záznamech v DB

L5 – Relační vrstva: Řízení dialogu

- Možnost zavádět dodatečná omezení komunikace
  - Dodatečná – realizovaná NAD L4
  - Tedy L4 komunikace může podporovat full-duplex, ale na L5 nemusí být podporován
- Realizace může být řešena formou „předávání pověření“
  - Podobně jako na L2 v Token Ringu - „Jen ten kdo má token – slovo – smí vysílat“
  - Reálně se samozřejmě předávání řeší jen pro polo-duplexní model

- Možné způsoby realizace
  - Simplexní
    - Komunikace vedená jen jedním směrem
  - Polo-duplexní
    - Zde je třeba řídit pověření – na úrovni L4 může být komunikace full-duplexní, ale charakter aplikace vyžaduje jeho omezení na polo-duplexní – princip „otázek a odpovědí“
      - V rámci HTTP posílám požadavek a čekám na odpověď – sice můžu poslat více požadavků paralelně, ale ty tvoří samostatné session
  - Full-duplexní
    - Komunikace je možná oběma směry bez omezení

L5 – Relační vrstva: Synchronizace

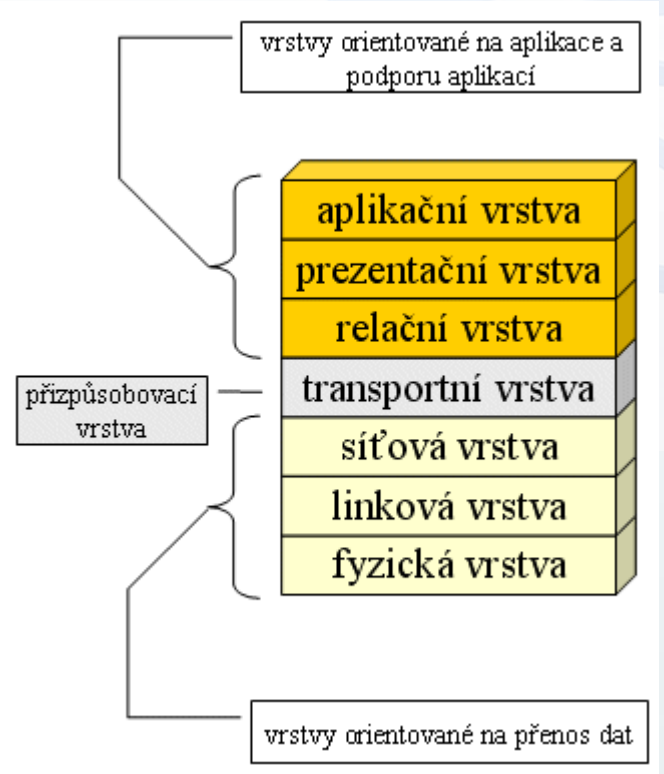
- Vytváření synchronizační body v komunikaci
- Snaží se řešit situaci, kdy jsou data na úrovni L4 v pořádku přenesena, ale v rámci aplikace pak nejsou v pořádku zpracována
  - Například všechna přijímaná data tisknu na tiskárně a dojde papír
- Aby bylo možné navázat na předchozí případ, zavádí se kontrolní synchronizační body – checkpointy
- Tyto body mohou být dvojího druhu
  - Hlavní synchronizační body – major - potvrzované
    - Definují bod v rámci komunikace, ke kterému už ale odesílatel nemá připravená data
    - Mezi hlavními body jsou vedlejší kontrolní body
  - Vedlejší kontrolní body – minor - nepotvrzované
    - Leží mezi dvěma hlavními body
    - V rámci komunikace je možné se k jednotlivým vedlejším kontrolním bodům vracet a odesílatel musí mít tato data připravena k znovu odeslání
  - Poslední hlavní kontrolní bod definuje oblast kam až se můžu v datech vrátit
  - Jednotlivé vedlejší kontrolní body až k poslednímu hlavnímu kontrolnímu bodu definují kroky po kterých se mohu vracet

L5 – Relační vrstva: Řízení transakcí

- Relační vrstva zavádí
  - aktivity - činnost například přenos souboru
    - Činnost od připojení k ukončení spojení
    - Aktivita je ohraničena hlavními synchronizačními body
  - dialogová jednotka
    - Přenášená data – například přenos jednoho souboru je jedna dialogová jednotka
    - Dialogová jednotka může obsahovat více vedlejších synchronizačních bodů
- Jednotlivé aktivity jsou na sobě nezávislé
  - Zda proběhla předchozí aktivita nemá žádný vztah k následující aktivita
- Každá dialogová jednotka musí proběhnout celé nebo vůbec
  - Tedy se jedná o atomické operace

L6 – Prezentační vrstva

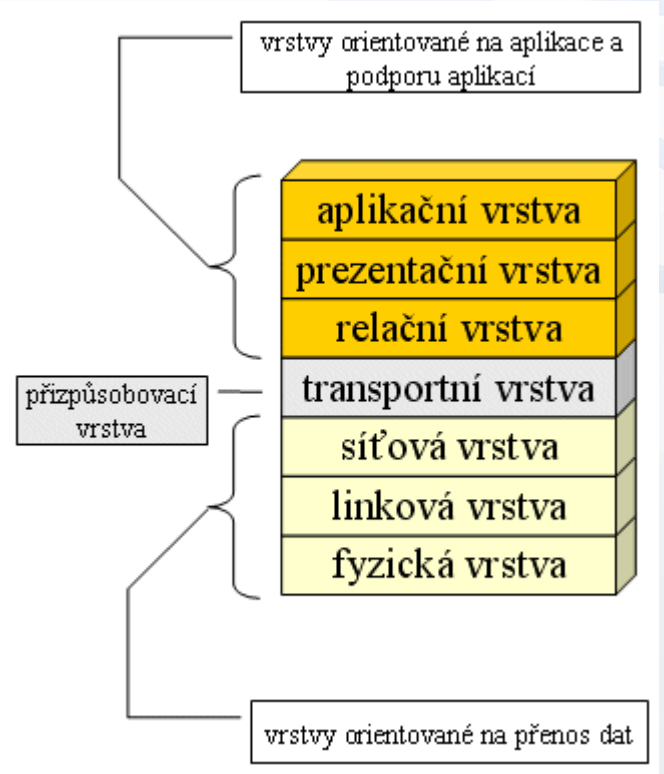
- Zajišťuje konverzi aplikačních dat do podoby vhodné pro přenos počítačovou sítí
  - Univerzální formát – tedy vlastně popis jaká data přenáším a jak mohou vypadat
    - Můžeme mít dva přístupy:
      - „Jednofázová“ - data nepopisuji, ale pouze přenáším, ale z definice protokol vím, o jaká data jde a jak s nimi pracovat
        - XDR – External Data Representation - Přenáší jen data – musím vědět jaká data jsou přenášena
      - „Dvoufázové“ řešení
        - Abstraktní syntaxe
          - Popis dat / datových struktur
          - ASN.1 – Abstract Syntax Notation
        - Přenosová syntaxe
          - Jak bude vypadat ten samotný přenos
          - BER – Basic Encoding Rules
            - TLV – type Length Value
              - Nemusím vědět jaká data přenáším, protože o jaké data se jedná je součástí přenášených dat
  - Komprese data
    - Snaha snížit velikost přenášených dat
  - Šifrování dat
    - Snaha zabezpečit data během přenosu
- Pro každý řešený problém může existovat a být podporováno více variant
  - Na začátku musí proběhnout dohoda jak přenášet
  - Obě strany znají několik možností / protokolů a musí se dohodnout na nejvýhodnější kombinaci
  - Dnes využíváno například pro šifrování SSL



zdroj: https://www.earchiv.cz/b06/b0300001.php3

L7 – Aplikační vrstva

- V obou modelech je aplikační vrstva, ale má dramaticky jiný význam
  - OSI/OSI – jedná se o službu poskytovanou aplikacím
    - Společné služby
      - Vytvoření asociace – vzájemná komunikace – navázání spojení
      - Typy služeb
        - » Volání vzdálených podprogramů
        - » Transakční zpracování – atomické operace
        - » Spolehlivý přenos dat
    - Specifické služby
      - Konkrétní služby, které požadujeme
      - Např
        - » Přenos souborů(FTP, TFTP, …)
        - » Adresářové služby (Novell, MS AD, …)
        - » Vzdálený přístup k terminálu (RDP, SSH, Telnet, …)
        - » Přenos zpráv(Email, Instant messaging)
  - TCP/IP – jedná o prostředí, kde opravdu běží samotná aplikace
    - Jednotlivé aplikace si výše uvedené problémy řeší samostatně
    - Můžeme definovat dva typy aplikací
      - Systémové – to co je třeba pro fungování sítě
        - » DNS, LDAP, DHCP, SNMP, …
      - Uživatelské – to co požaduje uživatel
        - » FTP, HTTP, SMTP, SSH, …



zdroj: https://www.earchiv.cz/b06/b0300001.php3

L7 – Aplikační vrstva:Systémové aplikace

- Může dále dělit dle použití
  - Adresářové služby
    - Specifické typy databází
    - DNS, LDAP, „MS AD“
  - Konfigurační služby
    - BOOTP, DHCP
  - Služby síťového managementu
    - SNMP, RMON
  - Bezpečnostní služby
    - Kerberos, SASL, „MS AD“

L7 – Aplikační vrstva:Systémové aplikace: DNS

- DNS – Domain Name System
- Převod jména na IP a zpět
  - proteus.fav.zcu.cz < – > 147.228.63.11
- Databáze s více typy záznamů
  - A, AAAA, CNAME? MX, TXT, …
- Přenáší data nešifrovaně
- Používá TCP i UDP protokol a ve výchozím stavu port 53
- Odpověď může být cachovaná
- Odpověď může být
  - autoritativní
    - Od serverů uvedených v WHOIS DB – autoritativní jmenné servery
  - Neautoritativní
    - Od cachujícího serveru

L7 – Aplikační vrstva:Systémové aplikace: DNS II.

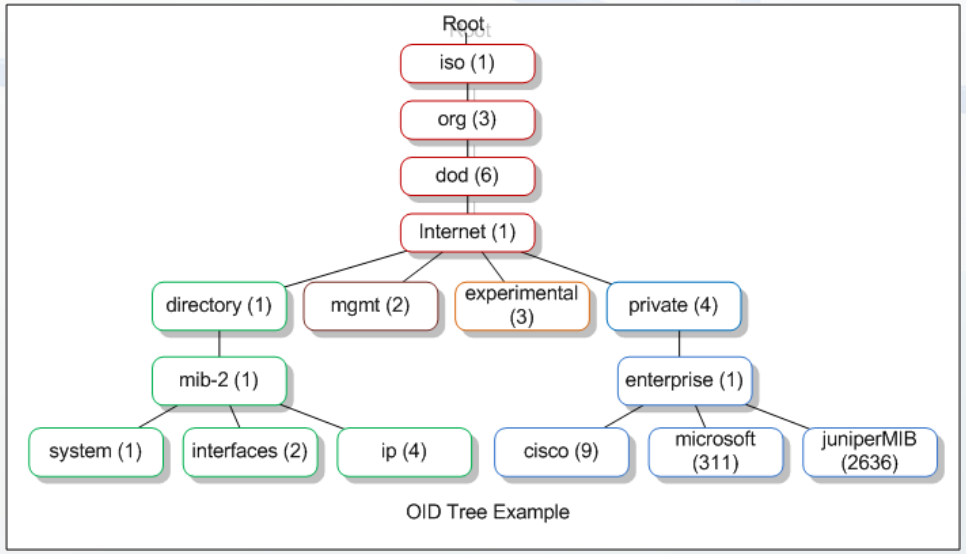
- Typické využití TCP a UDP v DNS
  - UDP – klasické komunikace – dotaz odpověď
    - Předpokládáme malá data – nečekáme, že dojde k chybě
    - Těch dotazů může být hodně a snadno se mohou opakovat
  - TCP – typicky synchronizace zón
    - U přenosu zónových souborů se jedná o větší celky dat
    - Tento přenos je třeba aby byl spolehlivý a nejde zde o čas jak rychle se povede
      - Rozuměj, vteřiny nehrají roli
- DNS není v základu nijak zabezpečené a je možné poměrně snadno
  - Podvrhnout odpověď – pokud jsem na vhodném místě mohu odpovídat a jiný DNS server a tím přesměrovat provoz
  - Podvrhnout dotaz – jedná se o UDP, tedy datagramy, které nemají navazované spojení, pokud vytvořím falešný – s falešným odesílatelem – DNS servery na něj odpoví, ale NE mě, ale podvrženému odesilateli a tím dojde na základě „zrcadla“ k DDOS
- Zabezpečení DNS se řeší až s příchodem DNSSEC a podepisováním zónových záznamů
  - Zatím není povinné a tedy ani masivně rozšířené

L7 – Aplikační vrstva:Systémové aplikace: DHCP

- Protokol sloužící k auto-konfiguraci nastavení sítě
- Používá UDP protokol dva porty
  - 67/UDP – port na kterém se klienta ptá serveru
    - Respektive na začátku se ptá všech broadcastem – protože žádné nastavení sítě nemá
    - Tím že nastavení sítě nemá, jedná se L2 broadcast
    - Zdrojová MAC je nastavena na MAC klienta
    - Zdrojová IP je 0.0.0.0 a cílová je 255.255.255.255
  - 68/UDP – port na kterém server odpovídá
    - Odpověď jde opět broadcastem(ale může i s cílovou MAC pro klienta) se zdrojovou MAC adresou serveru, IP zdrojová je server, cílová je 255.255.255.255 (dokud není konfigurace hotová, klient IP nemá)
- Provoz, tím že se používá broadcast je omezen na LAN
- Pokud v dané LAN není DHCP server, je možné požadavek „předat“ dále do jiné sítě, pokud je routeru DHCP Realy Agent
  - Mimo vlastní síť jde požadavek běžně unicastově – protože může projít pře více LAN
- V síti by měl být pouze 1 DHCP server jinak nastává problém
  - Ono jich tedy může být více, ale musí tvořit cluster – spolupracovat jako fail-over / balancer
  - Pokud je jich více síť nemusí fungovat správně
    - Od koho adresu mám jde najít v logu nebo tcpdumpu
    - Servery mohou mít i stejnou IP – pak je můžeme rozlišit pomocí ARP - MAC

L7 – Aplikační vrstva:Systémové aplikace: SNMP

- SNMP - Simple Network Management Protocol
  - Protokol sloužící pro podporu správy sítě a sběru dat
- Používá více portů
  - Nezabezpečená komunikace agenta UDP 161
    - Například klasické dotazy agenta - „Kolik dat proteklo přes interface eth0?“
  - Asynchronní nezabezpečené trapy UDP 162
    - Pokud v síti nastane nějaká monitorovaná událost, umí o tom SNMP asynchronně informovat server – zařízení kde problém nastal kontaktuje server
    - Velice výhodné tam, kde máme málo událostí – šetříme linku – a například adresy za NAT a tedy „nemůžeme“ jiné monitoringy jako např Icinga2
  - Zabezpečená komunikace agenta UDP 10161 – dostupné až od verze SNMPv3
  - Zabezpečené asynchronní trapy UDP 10162 – dostupné až od verze SNMPv3
- Informace shromažďuje ve vlastní konfigurovatelné databázi MIB
  - Nemá pevný obsah, ale definuje identifikátory OID a k nim hodnoty
    - Např. 1.3.6.1.2.1.2.2.1.5
  - Využívá k popisu dat podmnožinu ASN.1
    - Tedy „jako by“ prezentační vrstvu
- Nejčastěji se používá jako podkladová platforma pro sběr nejrůznějších dat o IT systémech
  - Protože ve velkém množství zařízení je přímo podporován
- Sice k němu existuje „modernější“ nástupce – RMON, ale ten se zatím masivně neprosadil



zdroj: https://www.fi.muni.cz/~kas/pv090/referaty/2015-podzim/snmp.html

L7 – Aplikační vrstva:Uživatelské aplikace

- Může dále dělit dle použití
  - Přenos souborů
    - TFTP, FTP, SCP, Rsync, HTTP
  - Vzdálený přístup
    - Telnet, X-window, SSH, RDP, VNC
  - Přenos zpráv
    - SMTP/POP/IMAP, Skype, Jabber, …

L7 – Aplikační vrstva:Uživatelské aplikace: TFTP

- TFTP - Trivial File Transfer Protocol
- Používá UDP port 69
- Nepodporuje žádné zabezpečení či ověření
- Nepodporuje žádné složitější operace než download a upload souboru
  - To je ovšem na jednu stranu výhodné, protože je možné jej integrovat do HW či použít na velice nízké úrovni v OS, protože je jednoduchý
- Typicky se používá na
  - přenos základní konfigurace routeru/switche
  - Boot bezdiskových stanic v kombinaci s DHPC a např NFS
    - Přes DHCP zjistíme nastavení sítě a info, odkud máme stáhnout kernel – TFTP a odkud máme připojit rootFS – NFS
- Díky omezeným možnostem se příliš nehodí na větší a časté přenosy

L7 – Aplikační vrstva:Uživatelské aplikace: FTP

- FTP - File Transfer Protokol
- Slouží pro přenos souborů – filozoficky se jedná o zdokonalení TFTP
- Používá TCP protokol
  - Protože potřebuje mít jistotu správného přenosu
- Používá více TCP portů – v základu 20/TCP a 21/TCP a další náhodně zvolené dle zvoleného módu
- Kromě prostého přenosu souborů umožňuje i další operace
  - Přenos může být binární nebo textový
  - Je možné procházet a vylistovávat adresáře
  - Je dostupné podpora ověření jménem a hesel
    - Ale je zachována i možnost anonymního přístupu
  - Je dostupné podpora šifrování pomocí SSL
    - Jen v novějších verzích – v základní je jméno a heslo posíláno v plain podobě
    - SSL verze nepoužívá extra port, ale je v rámci řídícího portu 21 dohodnuto, zda je SSL podporované a požadované oběma stranami
  - Jsou dostupné dva operační módy
    - Aktivní/normální mód
    - Pasivní mód

L7 – Aplikační vrstva:Uživatelské aplikace: FTP: Aktivní mód

- Aktivní / normální móde
  - Jedná se o výchozí chování
  - Klient se portu 21 připojí k serveru provede ověření atd
  - Před přenosem zvolí klient port na kterém začne poslouchat / očekávat spojení od serveru
    - Tento port je na server přenesen v rámci session na portu 21 a příkazem PORT
  - Server se připojí k klientovy na zvoleném portu s odchozím port 20 na straně serveru
  - Problém nastává v případě, že je klientem za NATem
    - Klient na privátní IP sice poslouchá na portu, ale server se k němu nemůže pře veřejnou IP připojit, protože v NAT není adekvátní záznam pro nový port
    - Lze řešit pomocí speciálních modulů pro NAT, které poslouchají provoz na porty 21 a dovolují doplnit NAT tabulky pro realizaci přenosu
    - Problém nastává ve chvíli, kdy je provoz na jiném portu nebo kdy je provoz tunelován
      - Protože pak přídavný modul nemá jak zjistit potřebné parametry
  - Problém může nastat i v případě použití state/established connection firewallu na serveru
    - Ten blokuje veškeré odchozí spojení, kterému nepředchází požadavek a navázání komunikace ze strany klienta
    - Cílem je blokování backdooru, ale zde se vlastně otevře náhodný port na který může kdokoliv přistoupit

L7 – Aplikační vrstva:Uživatelské aplikace: FTP: Pasivní mód

- Pasivní móde
  - Je řešením problému aktivního módu a NATu
  - Klient se připojí k serveru na portu 21 a provede ověření atd.
    - A přepne komunikaci do pasivního módu příkazem pasv
  - Server zvolí náhodný port na kterém začne očekávat přenos
    - Zvolen typicky z dynamického rozsahu portů
  - Klient se druhým spojením připojí k serveru na dohodnutý port s odchozím portem 20
  - Zde problém s NATem nenastává, protože veškerá komunikace je inicializována ze strany klienta
  - Do pasivního módu je nutné se přepnout ještě před přenosem samotným i jinými příkazy
    - Protože i například vylistování adresáře je realizováno mimo port 21 a tedy by bylo blokováno

L7 – Aplikační vrstva:Uživatelské aplikace: Telnet / SSH

- Protokoly pro vzdálené připojení na terminál (ale ne jen to)
- Telnet starší a nešifrovaný, TCP port 23
  - Přesto, že jej dnes SSH nahradilo, stále se používá tam, kde SSH není podporováno, například switche či routery
    - Tím, že není šifrovaný je jednodušší na implementaci
  - Jelikož se jedná o protokol založený na přenosu text, je dnes hojně využíván k testování odolně orientovaných protokolů
    - Například HTTP, ale především POP a IMAP
- SSH modernější a nativně šifrované
  - Používá TCP port 22
  - Přenos může být nejen šifrovaný, ale i komprimovaný
  - Podporuje různé druhy autentizace
    - Jméno a heslo
    - Klíče např RSA
    - Externí ověření ala Kerberos
  - Umožňuje „tunelovat“ další protokoly
    - ssh root@muj_server -L 3307:localhost:3306
      - Na localhost:3307 vám bude „poslouchat“ mysql ze server muj_server, ale reálně bude spojení tunelované pomocí SSH

L7 – Aplikační vrstva:Uživatelské aplikace: SCP / Rsync

- SCP umožňuje přenos souborů prostřednictvím SSH spojení
  - Stejně jako SSH používá TCP port 22
  - Na rozdíl od FTP podporuje od počátku šifrování celé komunikace – ne jen přihlášení
  - Funguje ve dvou režimech
    - SCP - dávkové zpracování příkazů – umí jen download/upload souborů
    - SFTP – Secure FTP- umožňuje interaktivní zpracování požadavků, stejně jako FTP klient
  - Podporuje chroot
    - Tedy při zadání cd / v klientovi neskončíte v rootu serveru, ale v přeneseném root, který je specifický pro každého klienta
- Rsync je program, který pomocí SSH dokáže přenášet soubory
  - Tedy obdobná funkce jako FTP/SCP, ale „chytřejším způsobem“
  - Rsync umí přenášet jen rozdíly nebo jen zobrazit jaká data být přenášel
  - Rsync umí navázat na předchozí přenos

L7 – Aplikační vrstva:Uživatelské aplikace: VNC

- VNC – protokol sloužící k zobrazení vzdáleného grafického terminálu
  - Nativně používá port TCP 5900
  - Ověření probíhá na základě sdíleného hesla
    - Ale to není povinné, je možný i neautentizovaný přístup
  - Protokol není v základu nijak šifrován
    - To je možné obejít například použitím SSH tunelu/VPN či speciálních rozšíření
  - Provoz není komprimován a přenáší se celé obrazovky
    - Velice náročný na přenos, především tam, kde je změn hodně
    - Velice často má problém se synchronizací myši
      - Dat/změn je hodně a musí kromě stahování změn přenášet zpět na server pozici myši, což se děje se zpožděním – myš pak „ujíždí“
- V zásadě funguje jako „vzdálený pohled na obrazovku“, tedy je společné pro všechny uživatele
  - Na rozdíl od RDP, které vytváří separátní session pro každého uživatele
- VNC se dnes samo o sobě na vzdálený přístup příliš nepoužívá a nahrazováno modernějším RDP
- Ale stále je používáno jako součást dalších systémů
  - Například pro vzdálený přístup na konzolu virtualizovaných systému, kde požadován grafický interface – Microsoft Windows / MACOS

L7 – Aplikační vrstva:Uživatelské aplikace: RDP

- RDP – Remote Desktop protokol - „Vzdálená plocha“ – protokol sloužící k zobrazení vzdáleného grafického terminálu
  - Nahrazuje VNC a řeší jeho nedostatky
    - Spojení je šifrované
    - Podporuje komprese
    - Přihlášení je ověřené na serveru na kterém služba běží
      - Tedy nemá své extra jméno a heslo jako heslo u VNC, ale reálně dojde k přihlášení uživatele / vytvoření session/relace, na kterou je možné i navázat
      - Session může být na serveru i více pro různé uživatele
  - Ve výchozím stavu funguje na TCP portu 3389
  - Původně bylo jen součástí Microsoft Windows Serverů
  - Dnes dostupné na všech Microsoft Windows, ale u v dalších systémech jako je např Linux
    - A to jak klient – rdesktop / xfreerdp tak i server xrdp
  - Nepřenáší kompletní obrazovku, ale jen změny
    - Výrazně úspornější než VNC

L7 – Aplikační vrstva:Uživatelské aplikace: Emailové služby

- Emailové služby jsou složeny z více komponent
  - MUA – Mail User Agent – například Outlook nebo Thunderbird
    - Slouží k obsluze pošty na straně klienta, tedy stahování, zpracování a tvorbu emailů
    - Používá více protokolů – POP3/POP3s, IMAP/IMAPS, SMTP/SMTPS, HTTP/HTTPS, MAPI, …
  - MTA – Mail Transfer Agent
    - Slouží k přenosu emailu od klienta k serveru a mezi servery samotnými
    - Používá protokol SMTP/SMTPS
  - MDA – Mail Delivery Agent
    - Slouží k doručení email od MTA do schránky uživatele
    - Může se jednat o komunikaci pomocí pipe nebo pomocí LMTP protokolu

L7 – Aplikační vrstva:Uživatelské aplikace: Emailové služby: POP3/POP3S

- Nejstarší protokol na stahování pošty
  - Neumí pracovat se složkami na serveru
  - Stahuje celé zprávy
  - Umí nechat na serveru kopie zpráv X dnů zpět
- Používat může dva porty
  - TCP 110 - pro nešifrované spojení
  - TCP 995 – pro šifrované spojení
    - To je realizováno na základě SSL certifikátu – o těch si řekneme více u HTTPS
- Komunikace je založena na principu přenosu textu, takže v nešifrované podobě je možné přímo vidět jméno a heslo
- Tím, že POP je textový protokol, je k jeho ověření/ladění možní použít telnet nebo openssl
  - Nešifrovaná verze:
    - telnet localhost 110
    - login pepa
    - pass pepa
  - Šifrovaná verze :
    - openssl s_client -connect localhost:995
    - Příkazy jsou pak stejné, protože i protokol je stejný, jen je zabalený do SSL

L7 – Aplikační vrstva:Uživatelské aplikace: Emailové služby: IMAP/IMAPS

- Modernější náhrada POP protokolu
  - Podporuje složky na serveru – včetně vnořených
  - Podporuje stahování pouze hlaviček emailu
  - Dovoluje přesouvat/filtrovat zprávy na serveru bez nutnosti stažení zprávy
  - Dokáže synchronizovat obsah mezi více klienty
- Používat může více portů
  - TCP 143 či TCP 220 - pro „nešifrované“ spojení
    - „nešifrované“ protože už umožňuje versi STARTTLS a tedy, že i na portu 143 může být provoz šifrován, pokud se klient a server dohodnou, že je to požadováno a podporováno oběma stranami
  - TCP 993 – pro šifrované spojení
    - To je realizováno na základě SSL certifikátu – o těch si řekneme více u HTTPS
- Vnitřní komunikace je stejně jako u POP textová a tedy dovoluje ověření pomocí telnet/openssl
  - Nešifrovaná verze:
    - telnet localhost 143
    - . login pepa password pepa
  - Šifrovaná verze :
    - openssl s_client -connect localhost:993
    - Příkazy jsou opět stejné, protože i protokol je stejný, jen je zabalený do SSL

L7 – Aplikační vrstva:Uživatelské aplikace: Emailové služby: SMTP/SMTPS

- Protokol sloužící k přenosu emailu od klienta na server a následně mezi servery
- Používat může více portů
  - TCP 25 - pro „nešifrované“ spojení
    - „nešifrované“ protože už umožňuje STARTTLS a tedy, že i na portu 25 může být provoz šifrován, pokud se klient a server dohodnou, že je to požadováno a podporováno oběma stranami
  - TCP 465 – pro šifrované spojení
    - To je realizováno na základě SSL certifikátu – o těch si řekneme více u HTTPS
  - TCP 587 – pro šifrované spojení s podporou autentizace
    - Šifrování je řešeno pomocí STARTTLS
  - Dnes už je možné jak autentizaci tak šifrování provozovat i na portu 25
    - Ale používají se o zbylé dva a to jednak kvůli zpětné kompatibilitě, druhak proto ,že port 25 bývá často při odchodu provozu z ISP blokován
- Stejně jako pop/imap je i SMTP textově orientován, takže základní připojení možné opět ověřit pomocí telnet/openssl
  - Nešifrovaně: telnet muj_server 25
  - Šifrovaně: openssl s_client -connect muj_server:25

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS

- Hyper Text Transfer Protokol
- Běžně funguje na dvou portech TCP
  - http – nešifrovaná verze port 80
  - https – šifrovaná verze port 443
    - Jedná se o HTTP, který je „obalen“ šifrováním, ale uvnitř je stejný
- Původní verze přijala dotaz, poslala odpověď + návratový kód a spojení ukončila
  - To se v situaci kdy jedna HTML stránka má X částí jak JS, CSS, obrázky, atd ukázalo jako extrémně nevýhodné
    - Protože režie navázání spojení, forku procesu na serveru atd je drahá – dlouhá
  - Nyní podporuje keep-alive spojení – spojení zůstává otevřeno definovaný čas i po vyřízení požadavku, protože se očekává, že přijdou další – šetříme zdroje
- HTTP je bezstavový, stavovost – relaci – přináší až podpora cookie / session
  - Session může fungovat i bez podpory cookie, ale není o to obvyklé
  - Cookie představují bezpečnostní riziko – proto se dnes musí „potvrzovat“, že o nich víte a souhlasíte

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS – typy požadavků

- GET
  - Nejběžnější požadavek – dej mi soubor / stránku / data
  - Celá požadovaná URL např /login.php?username=pepa&heslo=pepa se zaznamenává do logu
    - Proto se nepoužívá k přenosu formulářů či k přihlašování
- POST
  - Běžně používaný k přenosu dat od klienta k serveru
    - Může jít o formulář s daty, ale i soubor či soubory
  - Do logů se propisuje jen URL, např /login.php, ale předávaná data už ne
- PUT / DELETE
  - vytvoření/smazání objektu na serveru
- HEAD
  - Stejně jako GET, ale vrací jen hlavičku odpovědi a ne data
  - Používá se k testovacím účelům

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS – návratové kódy

- Každá odpověď obsahuje informaci o stavu vyřízení požadavku
  - Odpověď v rámci L7 - níže může být více chyb, které jsou pro HTTP díky TCP transparentní
- Návratové kódy jsou členěné „po stovkách“ do skupin
  - 1xx – informativní
  - 2xx – úspěšné
    - Nejčastěji 200 OK
  - 3xx – přesměrování
    - Nebyla vrácena požadovaná odpověď, ale požadavek byl přesměrován na novou adresu, kde snad odpověď najdeme
  - 4xx – chyba vyvolaná / způsobená klientem
    - Přesněji chyba vzniklá v důsledky chování klienta, např
      - 401 – přístup na stránku, která vyžaduje autentizaci
      - 403 – požadavek na soubor, kterou server nemůže otevřít kvůli oprávnění na FS
      - 404 – požadavek na neexistující stránku
  - 5xx – interní chyba serverů

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS – zabezpečení

- HTTP protokol nepodporuje šifrování
- To přichází až s rozšířením na HTTPS
- Šifrování zajišťují SSL certifikáty
  - Před samotným přenosem dat je vytvořeno šifrované spojení a tím je pak přenášen HTTP protokol
  - Certifikáty mohou být
    - Vlastní nepodepsané – každý si jej pomocí openssl může vystavit sám
      - Problém je, že prohlížeč mu typicky nevěří
    - Podepsané vlastní certifikační autoritou
      - Pomocí openssl si vytvoříte i certifikační autoritu a tu naimportujete do prohlížeče
      - Váš prohlížeč už certifikátu věří, ale jen váš
    - Podepsané obecné známou certifikační autoritou
      - Placené – např Thawte, kupují se na 1 rok, pak je třeba jej obnovit
      - Zdarma – iniciativa Let`s Encrypt – vystavují se zdarma na kratší dobu – např 3 měsíce
- Certifikát - „složité heslo s přidanou hodnotou“ je možné použít jak k šifrování tak i k ověření identity proti strany
  - Pokud certifikát podepsala nějaká obecně uznávaná autorita, byl nejprve jeho žadatel prověřen, takže pak máte vyšší jistotu, že se opravdu bavíte avizovanou protistranou
    - Samozřejmě i to lze za určitých okolností – například pomocí AD wildcard certifikátu zneužít

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS – Proxy/cache

- Téměř každý HTTP klient – browser – umí cachovat požadavky
  - Cílem je snížit opakovaný přenos dat a zrychlit odezvu webu
  - Vzniká zde problém neplatného/zastaralého obsahu cache
- HTTP požadavek nemusí jít na server přímo, ale může jít přes HTTP Proxy
- „běžná“ proxy funguje podobně jako cache na klientovi, ale jednotná pro celou síť
    - Opět s cílem snížit duplicitní přenosy, ale zde pro celou síť
    - S možností filtrace požadavků – např ***porno*** je zakázaný požadavek
    - S možností autentizace/autorizace požadavků – např na facebook.com se může, ale jen pokud znám jméno a heslo
      - Myšleno jméno a heslo k přístupu přes proxy server – následně se ještě logujete na facebook.com
    - Může zde docházet i k filtraci na úrovni IP adrese
      - Tedy vybrané IP – VLAN pana ředitele – smí na NET bez omezení, jiná VLAN má část obsahu zakázaný
      - Defakto se jedná o „firewall“, ale na L7 vrstvě
        - » Na L7 znamená, že nutně musím rozumět danému protokolu
- Proxy může logovat veškerá požadavky a používá se tak často jako podklad pro report provozu v rámci organizace
- Typickým představitelem proxy je například Squid

L7 – Aplikační vrstva:Uživatelské aplikace: HTTP/HTTPS – Reverzní proxy

- Druhým typem proxy je reverzní proxy
  - Běžná proxy zaštiťuje organizaci a propouští provoz dále do internetu
  - Reverzní proxy je typicky na straně serverů a zprostředkovává komunikaci klienta s jedním či více vybranými servery
  - Cíle nasazení jsou částečně shodné s klasickou proxy
    - Cachování požadavků – proxy může například statická data jako JS/CSS odbavovat výrazně rychleji než server za ní
    - Filtrace požadavků na základě URL
    - Přepis požadavků – tedy změna části nebo celé URL či přesměrování
  - Zároveň je možné reverzní proxy použít pro zvýšení stability / dostupnost
    - Proxy pak může fungovat ve dvou režimech a rozdělovat provoz mezi dva a více zdrojů
      - Fail-over
        - » Dostupných zdrojů mám sice více, ale používám jen jeden a další přepnu až v případě výpadku primárního zdroje
      - Balancer
        - » Mám více zdrojů, které používán současně a rozděluji mezi ně provoz
        - » Krom toho, že zvyšuji dostupnost jako fail-over, dovedu zde i provoz škálovat do šířky
        - » Dnes typická konfigurace většiny větších serverů

L7 – Aplikační vrstva: Obecně známé a používané protokoly

- Aplikačních protokolů je dnes obrovské množství
- Ale základních a nejčastějších byste měli vědět
  - K čemu slouží a jak cca fungují
  - Na jakých portech a jakými protokoly fungují
- Jedná se především o :
  - HTTP(TCP/80) / HTTPS(TCP/443)
  - POP3(TCP/110) / POP3S(TCP/995) / IMAP(TCP/143/220) / IMAPS(TCP/993)
  - SMTP(TCP/25) / SMTPS (TCP/587)
  - TFTP(UDP/69)/ FTP(TCP/20+21) / SCP(TCP/22) / RSYNC (TCP/873/22)
  - Telnet (TCP/23) / SSH(TCP/22) / RDP(TCP/3389) / VNC (TCP/5900)
  - DHCP (UDP/67+68) / DNS (TCP+UDP/53)