Lekce 1 - Úvod

1) Historie vzniku TCP/IP, vztah k NCP a ARPANETu, vývoj "vlády nad Internetem!

TCP/IP – vychází ze staršího experimentálního NCP...určen pro rutinní provoz sítě, vývoj akademická sféra, finance armáda...první síť – ARPANET...původně běží na NCP...1.1.83 přechází na TCP/IP (první zmínka 1973, 1981 – definován v RFC 791 a 793)...původně monolitický, později rozdělení transportní úkoly – rozdělování dat na bloky - (TCP) a síťové úkoly - přenos bloků – (IP)...dnes je to řada dalších protokolů řešící jednotlivé problémy...další: adresování (IP adresy), jmenný prostor (DNS), standardizace...původně za státní peníze, nyní peníze od komerčního sektoru...zůstává ve veřejném vlastnictví (standardy veřejné a veřejně vyvíjené, rozhodnutí plynou z diskuse a konsensu)

2) Dokumenty RFC, STD, FYI a BCP

RFC – obecný publikační mechanismus na internetu (standardy, info, návrhy, experimenty)...nikdy se nemění...změny se projeví vydáním nového RFC, který „zastará“ dřívější...číslování pořadovými čísly - RFC 1 z roku 1969, dnes přes 5000...standard (skutečné standardy internetu) nebo off-track (jiné)

STD – ty RFC, které skutečně jsou standardy (víc je těch, co nejsou)...jakýsi „pořadač“, nahoře je vždy nejvyšší aktuální RFC (může být i víc – standard pokrývá víc oblastí popsaných více RFC)

FYI – totéž pro informační dokumenty RFC

BCP – best current practice (doporučení a návody „jak na to“)

3) Standardy TCP/IP a standardizační orgány

Nejdřív – řízeno ARPA, lidé kolem V. Cerfa
1979 – ICCB (mění se na IAB)
1985 – IAB
1992 – ISOC (nad IAB)

IETF, IRTF – mezistupeň mezi vedením a pracovními skupinami (engeneering – implementace a research – budoucí vývoj)
IESG, IRSG – dozor a organizace nad pracemi
RFC editor – publikace standardů

Dříve – standardy z akademické sféry, dnes – standardy pocházejí od firem...než je něco přijato za standard, musí se ukázat, že to funguje v praxi...plus veřejná diskuse

4) Role ISOC, ICANN a WSIS ve vývoji Internetu a jeho standardů

ISOC – 1992, akademická organizace, zastřešuje standardizační proces navenek, morální autorita, ale malý vliv v politické a komerční sféře
ICANN – důsledek odejití vlády USA – náhradní orgán přebírající dozor nad internetem...stará se o jmenný a adresový prostor, číslování portů atp., provozuje

kořenové servery, nestará se o obsah internetu
WSIS – snaha omezit „nadvládu“ vyspělého světa (a hlavně USA) prostřednictvím ICANN, snaha o převedení pravomocí, nic moc se nevyřešilo

5) Jak je řešena správa TLD .cz? Včetně historického vývoje …

Nejdřív spravuje VŠCHT, poté firma CoNET (přísná a restritivní pravidla)...1.11.1997 (formálně, v praxi až 1999) – správu přebírá CZ.NIC, sdružení českých ISP => liberalizace internetu v čr
2002 – změna pravidel pro registrace domén (napřed zaplať, pak užívej – omezení spekulativních registrací)
2003 – decentralizace správy TLD domény - přechází na registrátory
2005 – rozhodnutí – nechceme tu zatím IDN (názvy domén s diakritikou)
2006 – podepsáno memorandum s MI – řeší legislativní otázky s vládou ČR, stát se dostává do vedení CZ.NIC

Lekce 2 – Architektura TCP/IP

1) V čem se lišil přístup autorů TCP/IP a ISO/OSI při vzniku obou architektur?

ISO/OSI – všechno uděláme sami (nebo alespoň převezmeme a přepracujeme do vlastního standardu (např. Ethernet –> ISO 8802.3)...od složitějšího k jednoduššímu (chceme nejdřív všechno, co nejde, to postupně odebírat)
TCP/IP – co je rozumné, to převzít...soustředit se na provázání s cizí technologií (př. IP over Ethernet)...postupný vývoj po malých krocích (začít s málem, postupně nabalovat další, když to jde)

2) Srovnejte způsob řešení otázky spolehlivosti v TCP/IP a ISO/OSI

ISO/OSI se snaží vždy zajistit co nejvyšší – dokonce i na více vrstvách nad sebou (=> vyšší režie)...inteligentní musí být síť...
TCP/IP – dává aplikacím si vybrat...protokol TCP (spolehlivý, ale složitější) a UDP (nespolehlivý, ale jednoduchý)...inteligence je tak na koncových uzlech

3) Koncepce vrstvy síťového rozhraní v TCP/IP, srovnání s ISO/OSI

TCP/IP se nestará o to, co je na síťové vrstvě a níž, pouze propojuje síťové rozhraní se síťovou vrstvou (výjimka – protokoly pro přenos po dvoubodových spojích SLIP a PPP)...takže funguje nad vším...ISO/OSI naproti tomu definuje i nižší vrstvy, čímž se stává omezenější

4) Koncepce síťové vrstvy TCP/IP

katenetový model – svět stvořený z dílčích sítí, propojení přes routery (dříve gateways)...prvek sítě je buď hostitelský počítač nebo směrovač (router)...nemělo by být obojí naráz
poklička – důsledně odděluje samotnou implementaci sítě od přenosových protokolů...tvořena: jednotným adresováním pomocí IP adres + protokolem IP (všude

stejným) – nespojovaný, nespolehlivý, best effort...prochází pouze MTU (max. velikost linkového rámce) – omezení fragmentace posílaných dat
součásti síťové vrstvy – adresy (IP adresy), převody mezi síťovou a linkovou adresou, fragmentátor, podpora fungování síťové vrstvy
související - směrovací protokoly, přidělování IP adres, přidělování doménových adres
nové – překlad adres, privátní adresy, dělení a sdružování adres, bezpečnost, podpora mobility...

5) Koncepce transportní vrstvy TCP/IP

End-to-end komunikace
2 různé protokoly – TCP (spojovaný, spolehlivý, adaptibilní, složitý, data po bytech), UDP (nadstavba IP, nespojovaný, nespolehlivý, data po blocích) + později další specializovanější protokoly SCTP (mezi TCP a UDP) a DCCP
zajištění multiplexu/demultiplexu dat

6) Koncepce aplikační vrstvy TCP/IP , způsob řešení prezentačních a relačních služeb

TCP/IP předpokládá, že stačí jednotná (a jednoduchá) aplikační vrstva – aplikace, která potřebuje být složitější, si relační nebo prezentační úkoly zajistí sama...vyjímky jsou samostatné protokoly RPC (relační) a XDR (prezenční) – ovšem aplikace, které je nechtějí jimi nejsou nijak zatížené

7) Požadavky multimediálních aplikací, možné přístupy ke QoS v TCP/IP

potřebují – malé a pravidelné zpoždění...to TCP/IP sám o sobě nezaručuje...
řešení – Kvantitativní – nárůst přenosové kapacity tak „aby to stačilo“ (snižuje pravděpodobnost problému) vs. Kvalitativní – zavést podporu QoS (řeší 100%, ale drahé a náročné)
implementace QoS – priorizace (různá priorita posílaných dat), rezervace (vyhrazení potřebné kapacity), hrubá síla (prostě zvětšovat kapacity)

8) Řešení bezpečnosti a mobility v TCP/IP

bezpečnost – původně neřešeno (předpoklad řešení na aplikační úrovni)...později framework IPSEC (na úrovni síťové vrstvy) – důvěrnost (šifrování), integrita (data se po cestě nezmění)...režimy transport (zabezpečení v hlavičce) a tunnel (datagram vložen do zabezpečeného datagramu)
mobilita – řešení přenositelnosti IP adres, metoda „agentů“ - směrování na původní místo a odtud na nové přes agenta, který ví kam, ale jen pro omezenou mobilitu, vyžaduje stále statické IP adresy...podpora mobility nově zavedena v protokolu DCCP

9) Co přináší a řeší IPv6?

Řeší problém vyčerpání adresního prostoru (u 128bitového adresování vychází na každého člověka 4 miliardy)...nový formát datagramu...více strategií přidělování

adres...podpora hierarchického směrování, bezpečnosti, QoS

Lekce 3 – IP Adresy

1) Jaká je koncepce IP adres? (návaznost na linkové adresy, složky IP adresy, způsob přidělování IP adres hostitelským počítačům a směrovačům)

žádná návaznost...síťová a uzlová část (představa – svět = sítě, síť = uzly), odděleny bitovou hranicí...adresa každého uzlu musí být unikátní...a v rámci jedné sítě musí být stejná síťová část...síti se přidělí větší množství adres se stejnou síťovou částí...v rámci sítě se pak přidělí libovolně...potenciálně hodně zůstane nevyužitých

2) Dělení IP adres do tříd A až E a důsledky na rychlost čerpání adresového prostoru

A – velké sítě...8 bitů na síťovou část, 24 na uzlovou...je jich jen pár...1.x.x.x – 127.x.x.x
B – střední sítě...16:16...128.0.x.x – 191.255.x.x
C – malé sítě...24:8...těch je nejvíc...192.0.0.x – 223.255.255.x
D – vyhrazena pro skupinový přenos...224.0.0.0 – 239.255.255.255...není logicky dvousložková – lze jednotlivé přidělování
E – vyhrazena pro budoucí využití (ale fakticky nevyužita) – 240.0.0.0 – 255.255.255.255

Rozdělení na třídy rychlost čerpání výrazně snížilo, i když to stále není dokonalé řešení

3) IP adresy se speciálním významem (broadcast, síť jako celek, loopback, multicast adresy)

řízený broadcast: x|11...11 (týká se sítě se síťovou částí adresy x)
omezený broadcast: 11...11|11..11 (jen aktuální síť)
síť jako celek: x|0
loopback: 127|x.x.x (nejde „ven“)
multicast: třída D

4) Způsob distribuce IP adres (přidělovatelé …) a jeho vývoj v čase

žádná IP adresa nesmí existovat dvakrát (výjimka – privátní IP adresy (pozdější))
centrální distribuce...původně SRI NIC (Stanford), později IANA a regionální přidělovatelé (RIPE, ARIN...)

5) Principiální možnosti řešení úbytku IP adres (včetně IPv6/IPNG)

problém: stále nedostatečně pružné (méně uzlů v síti než adres k dispozici)
okamžité řešení: přidělovat pouze části tříd (větší směrovací tabulky), subnetting (jeden vstupní bod a zbytek uzlů má posunutou bitovou hranici)
dočasné řešení: privátní adresy (pokud provoz nejde „ven“, můžou tam být adresy už použité, směrem ven řeší firewall nebo překládací protokol NAT/PAT), CIDR (přidělování adres v kvantech mociny dvojky, pomocí masky, která říká, jak velká je síťová část)
konečné řešení: IPv6 – přechod na 128 bitový adresní prostor

6) Princip subnettingu a jeho využití

posun bitové hranice v lokální části sítě pomocí masky – rozdělení na více skupin (mociny dvojky)...ví se o tom jen v lokální síti...navenek jde vše přes směrovač se standardní adresou...pro soustavy více sítí v rámci jedné třídy adres...je potřeba jeden vstupní bod (stromová struktura sítě)

7) Privátní IP adresy a způsob jejich využití

problém duplicity adres – směrovač neví, co platí...ale když nebude vědět o existenci uzlu na druhém konci světa se stejnou IP, to to nevadí...takže pokud spolu 2 uzly nekomunikují, můžou mít stejnou adresu...je nutné nepustit nic ven – přes brány (firewall)
jsou dány doporučené adresy, které mají být užívány jako privátní – kdyby se firewall rozbil a šířil data dál, tak nejbližší správně nakonfigurovaný po cestě to pozná a data zarazí...

8) Mechanismus NAT a jeho fungování (včetně NAT/PAT)

Zajišťuje průběžný překlad privátních IP adres na veřejné - průchod dat - na sítové vrstvě...není vidět pro aplikace, aplikačně nezávislý (je možné to dělat na aplikační vrstvě, ale to musí každý sám pomocí proxy brány)...stojí na rozhraní mezi veřejnou a privántí sítí, skrývá privátní adresy před světem...statický (pevně dáno jak překládat) vs. Dynamický (přiřazování vnější/vntiřní adresa se děje náhodně dle potřeby)
PAT – mapuje adresy vnitřních uzlů do 1 vnější IP adresy...info o jednotlivých uzlech v číslech portů
výhody – jedna IP adresa pro „hodně“ uzlů
nevýhoda – nejde se připojit z venku dovnitř, nemusí fungovat (když je IP adresa i jinde než v hlavičce – řeší „inteligentní“ NAT)

9) Mechanismus CIDR a jeho přínosy

„inverzní subnetting“ - posun bitové hranice doleva, do síťové části...síť může dostat „přesně“ (mocniny dvojky) tolik uzlů, kolik potřebuje...princip – agregace adres - shodné částí síťové adresy umístit do CIDR bloků: 194.213.228/24 – odpovídá adrese třídy C – 24 bitů je síťová část, první tři byty tvoří společný „prefix“...směrování - příchozí IP pakety, které začínají prefixem, se směrují tam, kam ukazuje CIDR blok...a tam už je zas k dispozici další, jemnější členění...přínosy – šetří IP adresy i velikost směrových tabulek

10) IP adresy v IPv6, srovnání s IPv4

základ – rozšíření adresního prostoru na 128 bitů (IPv4 – 32 bitů)
opravuje nedostatky – multicast (skupina uzlů) místo broadcastu (všechny uzly)
přidává vlastnosti – Qos, bezpečnost, mobilita, anycast (víc uzlů se stejnou adresou, ozvat se má ten nejbližší)
eliminuje „pomůcky“ jako NAT
je zpětně kompatibilní s IPv4, opačně ne...

11) Vyhrazené adresy v IPv6, symbolický zápis IPv4 a IPv6 adres

vyhrazené adresy:
000... - různé účely
001... - unicast adresy
111... - místní adresy a multicast
loopback - ::1 (samé nuly, poslední jednička)
nespecifikováno - :: (samé nuly – např. při dotazování se na přidělení IP adresy)
FE:xxx – privátní (site local (v rámci soustavy sítí zákazníka) vs. link local (nejde přes žádný směrovač))
::IPv4 adresa – embedded IPv4 (zleva doplněná nulami)

zápis:
stright hex – 8x čtyřmístné hexadec. číslo
leading zero supressed – nulová slova se zkracují na 0
zero compressed – nulová slova se vynechávají
mixed notation – poslední 4 byty se zapisují decimálně jako v IPv4

Lekce 4 – Systém DNS

1) Soubor hosts a použití symbolických doménových jmen před DNS, plochý prostor doménových jmen

Hosts – obsahuje záznamy ve formátu „Jméno=IP adresa“, uložen na centrálním serveru
Původní doménová jména byla jednosložková...o změny se staral jeden centrální server...to bylo použitelné, jen dokud byl malý počet uzlů v síti

2) Vymezení pojmu "doména", Top Level domény, princip delegace pravomoci, pojem zóny

hierarchická (stromová) struktura jednotlivých jmenných prostorů – 1 prvek = doména...top level – TLD (ccTLD (cz, us) a gTLD (com, org)), stojí úplně nahoře stromu...pravomoci – v jedné doméně nesmí být 2x stejné jméno...každá doména může automaticky vytvořit subdoménu „www“...zóna – okruh působnosti jedné autority (má právo provádět změny), může měnit velikost (nové subdomény, delegace pravomocí)...zone file – info o doménových jménech v zóně

3) Syntaxe doménových jmen, plně kvalifikovaná doménová jména, způsob začlenění prostoru IDN do jmenného prostoru DNS

jednotlivá dílčí jména (max 63 znaků) oddělená tečkou...písmena, číslice, pomlčka (jen uvnitř), ignore case...celkem max. 255 znaků...nejnižší doména je nejvíc vlevo...plně kvalifikovaná – obsahuje celou posloupnost dílčích jmen až po nejvyšší doménu...IDN (národní abecedy a další znaky) – systém se nemění, překlad pomocí klientských aplikací

4) Vztah name serverů a domén, primární a sekundární name servery, kořenové name servery

Name server – jeden prvek domény, který má všechny informace o struktuře domény...1 doména = 1 name server (logicky, fyzicky nejméně zdvojeny), 1 pc = name server pro 0-n

domén (pro 1 zónu)...primární ns – pracuje přímo s zone file na local disku...sekundární ns – záloha, data si bere z primárního (zone transfer)...kořenové – 13 po světě, většinou v USA

5) Princip překladu symbolického doménového jména na IP adresu

nějaký nameserver někde v síti zná překlad doménového jména na IP adresu...tazatel se ptá svého (nejbližšího) name serveru...ten buď odpoví, pokud informaci má a nebo „postoupí dotaz výš“ - buďto se sám ptá (postupuje od nejvyššího „root“ name serveru) – rekurzivní dotazování a nebo tazatele odkáže jinam – iterativní dotazování

6) Architektura systému DNS (servery a resolvery, optimalizace fungování, autoritativní a neautoritativní odpovědi)

klient/server princip...name server – odpovídá na dotazy klientů...resolver – zprostředkovává roli klienta pro dotazy mezi name servery (je jak u běžných uzlů, tak u name serverů – pro komunikaci s jinými ns)
optimalizace – replikace (každý ns má aspoň jednu zálohu, v praxi i víc, funkčnost, snížení zátěže), cache – ukládání výsledků do mezipaměti (největší optimalizace), forwarding server (rekurzivní dotazy - přijímá, ale nevyhodnocuje)
info z cache je „neautoritativní“ (nedává ho přímo autorita – name server)...uzly můžou chtít autoritativní dotaz, ten pak musí nameserver přepočítat

7) Resource Records v rámci DNS, jejich typy

Formát uložení dat v DNS – věta v databázi
Jméno – typ – třída – ttl (max. doba uložení do cache) – velilkost dat – data
Typy – SOA (zóna), A (ip uzlu), AAA (ipv6 uzlu), NS (nameserver), MX (doručování pošty) ...

8) Protokol DNS, formát zprávy

Jednoduchý klient/server protokol...nejčastěji UDP...port 53
paket DNS query – dotaz i odpověď stejný – HEADER – QUESTION – ANSWER – AUTHORITY - ADDITIONAL

9) Domény a diakritika, systém IDN

původní DNS to nepovoluje...ale může to být žádané – nadstavba IDN – definuje překlad z UNICODE do ASCII na straně aplikací...nové domény začínající „xn-“, následuje definovaný překlad libovolného unicode jména...vyžaduje specializovaný software

10) Princip alternativních DNS stromů, přednosti a nevýhody

náhrada oficiánlích kořenových DNS serverů vlastními => vlastní TLD domény...je potřeba donutit uzly ptát se těchto alternativních

11) Princip dynamického DNS

Na koncovém uzlu běží DNS klient, který průběžně informuje svět o změnách své IP adresy – v případě častých změn (např. dynamické přidělování IP), typicky placené

12) Princip překladu IP adresy na doménové jméno, reverzní domény

Existuje doména in-addr.arpa, jejíž subdomény následně tvoří odpovídající části IP adresy (v obráceném pořadí)...a na konci této strutury je doménové jméno (CNAME) uzlu s danou IP adresou – reverzní doména

13) Koncept ENUM, jeho princip a možnosti využití

Snaha provázat s DNS telefonní čísla – projekt ze světa spojů, ponechat číslice, obrátit jejich pořadí, oddělit tečkami a na konec přidat .e164.arpa – a máme jméno v DNS
využití: přidávání dalších informací k tel. Číslu, telefon místo www adresy, posílání mailů na telefon

Lekce 5 - Protokol IP et al.

1) Charakteristika a vlastnosti protokolu IP

vlastnosti - přenosový protokol síťové vrstvy – univerzální, jediný...používá virtuální pakety – IP datagramy, zajišťuje směrování a přenos...funguje nespolehlivě, nespojovaně...ipv4 a ipv6
charakteristika - univerzální, jednotné služby (společné minimum, skrytí odlišností)...proměnná velikost paketů (velikost určuje odesilatel)...jednoduchý, efektivní, rychlý...na principu best-effort

2) Formát IP datagramu, význam položek v hlavičce

proměnná velikost (576B – 64kB)...hlavička a datová část (obojí proměnná velikost)
hlavička – version (4), length (velikost hlavičky - 4bity), type of service (ignorováno), total length (délka datagramu – 16bitů), identification (řeší fragmentaci), flags (zákaz fragmentace, pokračují další fragmenty), frag offset (pozice dat v původním celku), ttl (počet přechodů přes směrovače), protocol (typ nákladu), header checksum (kontrolní součet), IP zdroje, IP příjemce, option field(s) (volitelná rozšíření), padding (doplněk hlavičky na násobek 32 bitů)

3) Způsob řešení fragmentace v IPv4

směrovač rozdělí datagram na menší (podle parametru MTU) – každý kus dostane stejnou identification jako původní, nastaví se more fragments=1 (krom posledního kusu), nastaví

se potřebný offset...
tato fragmentace může mít libovolně mnoho úrovní – zpětně se sestavuje až u příjemce (když něco nedorazí, všechno se zahazuje)

4) Koncepce protokolu ICMP, jeho vztah k IP a zařazení do vrstev

informuje odesilatele (resp. jeho IP protokol) o nestandardních situacích (zahození datagramu) – vysvětluje co se stalo (zahlcení, timeout, nedosažitelné, přesměrování, atp.)
+ některé další funkce (ne přímo chybové hlášky – tracerotue, echo request)
ICMP je pevná součást IP – ICMP pakety cestují v rámci IP paketů
vrstvy – měly by být na transportní vrstvě (kvůli povaze IP), ale to by je neviděly směrovače (ty jsou na síťové)...řešení – kompromis – ICMP je posíláno jako IP paket (transportní vrstva), ale jsou zároveň brány jako součást síťové vrsty...porušení vrstevnatého modelu – ale má to dobrý důvod

5) Způsob využití protokolu ICMP v utilitách traceroute a ping

traceroute – posílá ICMP datagram (typu traceroute, s ttl=1) na neexistující port...první směrovač zachytí datagram, shodí ttl na 0, zahodí datagram a pošle ICMP datagram odesilateli – ten tak zjistí kam datagram šel...a pošle další s ttl=2...poslední krok – UDP datagram na neexistující port

ping – dotazující posílá ICMP typu „echo request“, dotázaný typu „echo reply“...test dostupnosti, RTT (času cesty tam a zpět) a počtu skoků přes směrovače (TTL)

6) Překlad IP adres na linkové adresy - možné přístupy, protokol ARP a formát jeho zprávy

tabulkový převod – tabulka „IP=linková adresa“...rychlé, ale problém jak udržet tabulky aktuální
přímý výpočet – existuje funkce na převod...ale funguje jenom když jde nastavit HW adresu
dotazy a odpovědi – někdo se ptá, kdo ví odpovídá...centrálně (jeden uzel zná všechno) nebo distribuovaně (každý uzel zná svoji hw adresu a o té informuje)

ARP – dotazy a odpovědi, distribuovaná varianta...dotaz šířen broadcastem...odpovídá ten, komu je určen...definuje formát dotazů/odpovědí a způsob přenosu

ARP zpráva – typ hw adresy, typ adresy protokolu, délka hw adresy, délka protokolové adresy, operace (1 dotaz, 2 odpověď), hw adresa odesílatele, protokol adresa odesílatele, hw adresa příjemce, protokol adresa příjemce

7) Princip IP multicastu, protokol IGMP

jeden paket doručit více příjemcům současně – linkový (uzly v přímém dosahu, snadné, IP multicast (kamkoliv v dosahu IP protokolu, složité)
využití multicastových IP adres – uzly mají dynamické členství ve skupinách (trvalé či dočasné)...skupina má danou multicastovou adresu...paket se pošle na danou adresu a

doručí všem aktuálním členům...linkový multicast – přenos v rámci uzlů jedné sítě...multicast agent – řeší přeposílání do jiné sítě

IGMP – ovládání multicastových skupin...pakety cestují v IP paketech...formát: typ žádosti (vytvoření skupiny, přidáni uzlu atp.), code (veřejná/privátní), checksum, identifikátor, multicast ip adresa, klíč (pro přístup do privátních skupin)

8) Hlavní změny v IPv6 oproti IPv4

větší adresní prostor – 128 bitů oproti 32
jiná struktura paketu – více hlaviček s pevnou délkou, hlavičky jednodušší, není checksum
podpora QoS, zabezpečení (rozšiřující hlavičky), směrování
jinak řešená fragmentace
multicast místo broadcastu, anycast...

9) Způsob řešení fragmentace v IPv6, odlišnosti od Ipv4

fragmentace jen u odesilatele (na směrovačích je to náročné na provoz)...pokud někde nepůjde přenést, zahodit a vygenerovat ICMP chybu „too big“...odesilatel si buď zjistí maximální propustnost spojení (Path MTU discovery) a nebo využije minimální velikost paketů 1280 bytů (garantovaná)
fragmentace indikována rozšiřující hlavičkou

Lekce 6 - IP směrování

1) Přímé a nepřímé doručování v IP, způsob využití směrovacích tabulek, možnosti jejich aktualizace (dynamické a statické směrování)

přímé – odesilatel a příjemce ve stejné síti (stejná síťová část adresy)...nemusí se rozhodovat o žádném směru, pošle se přímo
nepřímo – různé IP sítě, odesilatel hledá nejvhodnější směrovač a pošle jemu

směrovací tabulky – seznam cílových sítí (nebo jejich CIDR prefixů – agregace údajů) a „next hop“ - tj. adresa nejbližšího směrovače (nebo pokyn „pošli přímo“)...prohledává se odshora dolů (nejvíc obecný pokyn musí být dole)...host specific route (konkrétní instrukce pro konkrétní cílový uzel) vs. default route (použije se pro všechno ostatní, co není v tabulce)

2) Možnosti optimalizace směrovacích tabulek (zmenšování počtu položek)

agregace do CIDR bloků – pokud má skupina uzlů stejnou část adresy a posílá se přes stejný směrovač, dá se spojit do jednoho záznamu
default route – stačí definovat cesty jen k pár uzlům a zbytek ať chodí podle default route...

3) Základní algoritmus směrování v IP

- zjistit, zda se příjemce nenachází ve stejné síti
- pokud ano, přímé doručování
- pokud ne – prohledávat směrovou tabulku
- pokud se prefix hledané adresy shoduje se záznamem – pošli nepřímo dle instrukce, jinak pokračuj v hledání
- pokud je dána default route – pošli nepřímo tam
- pokud není, skončit s ICMP chybou „Unreachable“

4) Role hostitelských počítačů při směrování, využití protokolu ICMP při směrování

hostitelské počítače mají směrovací tabulky a rozhodují se podle nich...ale neaktualizují jejich informace, to dělají pouze směrovače, hostitelé jen přebírají...host musí znát alespoň jeden směrovač...postupně se učí (směrovač mu řekne „tohle je lepší posílat jinam“)

ICMP Redirect – prostředek směrovačů, jak upozornit hosty, že neposílají optimální trasou a že by měli svá data začít posílat jiným směrem...hostitel se má poučit...není to ale důvod k zahození paketu
ÏCMP Router solicitation – dotaz „jaké jsou v okolí směrovače?“ (překlenutí toho, že host na začátku nezná nic – prostě se takto zeptá...)
ICMP Router advertisement – buďto odpověď na solicitation a nebo „reklama“ samotného směrovače

5) Směrování v rozsahu celého Internetu a jeho vývoj v čase

nejdřív – centrální směrovače s úplnou informací (pro malý internet dostačující)
později – 2 úrovně – core s úplnou informací (páteřní síť) a noncore s částečnou informací (znaly jen své podsítě)...protokoly GGP a EGP pro komunikaci...požadavek na stromovou strukturu
ještě později – dekompozice na malé AS (v rámci sebe mají detailní informace, ale nešíří je ven)

6) Princip autonomních systémů, EGP a IGP

potřeba udržovat směrovací informace rozumně velké – rozpad na malé bloky...uvnitř nich detailní směrovací informace...ven se šíří jen informace typu „uvnitř mě jsou sítě A, B, C“ (fyzicky rozsah adres, resp. CIDR bloky)...komunikace skrz omezený počet vstupních bodů...libovolná struktura => mj. podpora peeringu

EGP – dříve přímo protokol pro komunikaci mezi jednotlivými AS (funkčnost spíše pro centralizovaný internet), dnes označení pro celou rodinu protokolů – používá se BGP (propojení s cykly, podpora CIDR...)
IGP – rodina protokolů pro komunikaci uvnitř AS...protokoly RIP a OSFP

7) Protokol RIP a jeho fungování

řeší směrování uvnitř jednotlivých AS...typu vektor-distance – pevnou metrikou je počet přeskoků, updaty každých 30 sekund – jen sousedům (neúplná informace o topologii),

pokud nepřijde odpověď do 180s, uzel je označen za nedostupný (nekonečná délka cesty), nekončící výpočet, který je distribuovaný – závislý na chybách ostatních, nepřipouští alternativní cesty (lepší cesta musí být ostře kratší)...implementován jako aplikace...vývoj – RIP2 (nové vlastnosti), RIPng (pro IPv6)

8) Protokol OSPF a jeho fungování

řeší směrování uvnitř jednotlivých AS...typu link-state...uzly testují dostupnost svých sousedů (stav linky)...link state paket – od každého uzlu všem uzlům v síti, rozesílán při změně nebo po 30 minutách pro refresh...úplná informace o topologii, výpočet není závislý na druhých, umožňuje alternativní cesty, podpora další dekompozice

Lekce 7 - Transportní protokoly

1) Dobře známé porty - podstata a význam, kdo udržuje konvenci?

Porty na kterých jsou poskytovány služby...význam – jednotné připojování pro klientské aplikace...porty 0-1023...spravuje a přiděluje IANA...veřejně dostupné (dříve RFC, dnes internet)

2) Čím je definováno aplikační spojení? Koncept portů a socketů.

Port – logické zakončení spojení, na jeden může být připojena jen jedna entita, entita může být asociována s více porty, na jeden uzel a port může vést víc spojení, spojení je typu fronta
Socket – styl práce s porty (z BSD unixu), analogie brány vedoucí ke službám

Aplikaní spojení – transportní protokol, IP1, port1, IP2, port2

3) Způsob práce se sockety při nespojované komunikaci (naznačte postup)

Funkcí SOCKET se založí nový socket, funkcí BIND se asociuje s nějakým portem (implicitně žádný), zavírá se funkcí CLOSE...umí dvě elementární operace SENDTO a RECVFROM

4) Způsob práce se sockety při spojované komunikaci (naznačte postup)

vytvoření a bindování stejné...dále funkce LISTEN (čekání na žádost o spojení), CONNECT (žádost o spojení), ACCEPT (přijetí spojení) a funkce SEND a RECV pro přenos dat

5) Protokol UDP - jeho vlastnosti, formát datagramu, výpočet kontrolního součtu hlavičky

velmi jednoduchý - jen multiplex/demultiplex, kontrolní součet hlavičky, malá režie
nespolehlivý, nespojovaný, iluze blokového přenosu, bezstavový
udp datagramy – vkládané do IP datagramu, v praxi malé...source port, destination port, délka udp, checksum, data...volitelná pseudohlavička (jen pro účely přenosu, ochrana proti nesprávnému doručení)...
kontrolní součet – v jedničkovém doplňku (nulový – samé jedničky, žádný – samé nuly), datagram s chybným součtem je zahozen (bez ICMP zprávy)

6) Protokol TCP - jeho vlastnosti, způsob navazování a ukončování spojení

dobře řeší složité přenosové problémy...spojovaný, plně spolehlivý, dvoubodové spojení, řízení toku, iluze bytové roury, ochrana před zahlcením, stavový – navazuje a ukončuje spojojení
softwarová emulace spojovaného charakteru – mezi koncovými uzly se posílá nespojovaně, příjemce odesílá potvrzení, odesilatel čeká na jeho doručení po určitou dobu...doba se mění – adaptibliní podmínkám...přenos pomocí bufferu odesílatele a přijemce, součástí TCP datagramu je aktuální pozice v proudu (na začátku se synchronizuje), pouští se jen tolik dat, kolik příjemce dokáže zpracovat (metoda okénka)
spojení pomocí třífázového handshake - „připojit“ - „připojuji“ - „ok“ a „zruš“ - „ruším“ - „ok“...spojení se ruší pokud proběhne komunikace a nebo po vypršení časového limitu

7) Zajištění spolehlivosti v protokolu TCP

potvrzování – příjemce posílá potvrzující pakety, odesilatel na ně čeká
synchronizace – buffery bytového proudu u příjemce i odesilatele – do TCP datagramu se dá informace o pozici
ochrana před zahlcením – každá ztráta brána jako zahlcení, po ztrátě neposílá dál, ale čeká na potvrzení, když potvrzení přijde, pošle 2x tolik dat (a pomalu nabíhá na původní úroveň)
v novějších variantách TCP řešeno ještě lépe

8) Možnosti zajištění QoS na úrovni transportní a aplikační vrstvy

transportní – síťová vrstva dává „všem stejně“, TCP ani UDP neřeší...typicky řešení „hrubou silou“ (prostě zvedat kapacity, aby k problémům nedocházelo (docházelo méně často)...pořád je to nejsnazší)
aplikační – client buffering – data proudí různě rychle, ale cílová aplikace si je bufferuje a z bufferu bere konstantní rychlostí

9) Protokol RTP/RTCP

transportní podpora QoS...balení dat do specifických paketů s dodatečnými informacemi – typ obsahu, pořadí, čas vzniku, info o proudu...posíláno přes UDP, podpora multicastu
RTCP – informuje obě strany o stavu RTP přenosu

10) QoS: Differentiated a Integrated Services

integrated – garantovat, co kdo potřebuje – specifikace při navazování spojení, když síť nemůže vyhovět, tak nespojí...v podstatě návrat k přepojování okruhů (je třeba předem vyhradit kapacitu)...realizace pomocí R-Spec (požadavky) a T-Spec (jak bude vypadat provoz) – směrovače zjistí, zda můžou zajistit – protokol RSVP

differentiated – nabídnout alespoň nějaký kompromis v rámci možností, třídy provozu (priority), podpora na síťové vrstvě, pakety označeny „nálepkou“ s druhem služby, např. Expedited Forwarding (2 třídy) a Assured Forwarding (4 třídy, 3 úrovně)

Lekce 8 - Telnet, FTP a NFS

1) Možnosti a přístupy ke vzdálenému přihlašování v TCP/IP (rozdíl Telnet vs. rlogin, ICA …)

telnet – maximálně univerzální (nezávislý na prostředí, jednoduché služby), terminálnové relace, nepodporuje přihlášení, pouze znakové rozhraní
rlogin – vázán na BSD UNIX, trusted hosts – info z okolí – přihlašování
ICA – nové řešení s podporou grafického rozhraní

2) Princip fungování Telnetu, význam a role NVT

na aplikační úrovni (není součást OS) – snadno se modifikuje, na některých OS nepodporován, neefektivní
NVT – odbourává rozdíly mezi koncovými terminály, řeší formát odesílaných dat, koncové terminály se mu přizpůsobují, společné minimum – jednoduchý řádkový znakový terminál, přenos dat pomocí TCP, jen poloduplexně, přenáší 7-bitové ASCII znaky
lepší podmínky lze dohodnout s klientem přes options Telnetu

3) Příkazy Telnetu, mechanismus rozšiřování

tři druhy příkazů – editační (práce s terminálem), řídící (práce s procesy telnetu), dohadovací (nastavení terminálu oproti NVT)
rozšiřování – dobrovolné...mohou navrhovat obě strany...přes příkazy WILL (žádost – já chci) a odpovědi DO/DONT a DO (žádost – já bych rád abys ty) a odpovědi WILL/WONT...rozšíření nejsou nijak omezena

4) Koncepce protokolu FTP (pohledy na soubor, režimy fungování, uživatelé a práva, srovnání s protokolem TFTP)

FTP má jednotný formát dat pro přenosy (ale koncové uzly se mohou dohodnout na jiném)
Obecně posílá 8-bitové znaky jako proud
Soubor je pro něj defaultně nestrukturovaná posloupnost znaků (alternativně: posloupnost stejně velkých záznamů, množina stránek)
Přenáší implicitně stream dat (alterntivně: bloky (se zarážkami pro případy výpadků), komprese (eliminace opakujících se znaků))
Klient/server model – klient posílá příkazy, server odpovědi (trojmístná čísla + textový

popis) a případně provádí...
Řídící (s vlastním jazykem – řídící a uživatelský) a datové spojení
Ví o uživatelích - vyžaduje autentizaci uživatele (aspoň jako „annonymous“)
TFTP – osekaná verze, nezná uživatele, typicky pro natažení „boot image“, používá UDP (FTP TCP), nedělá žádné akce na vzdáleném pc – je třeba zadat úplnou cestu k souboru

5) Implementace a fungování protokolu FTP (spojení a jejich navazování, pasivní režim, druhy klientů)

řízeno vlastním jazykem – příkazy textové, přenášené podobně jako telnet, řízení přístupu (login), nastavení parametrů (změny čísel portů), výkonné příkazy (přenos souborů, přejmenování, mazání)
textové odpovědi – stačí první číslice (1-5), zbytek info navíc
spojení je třeba zaštítit nějakým uživatelem jehož jménem FTP pracuje
existují řádkové i grafické klienty

6) Příkazy a odpovědi protokolu FTP, dialog mezi klientem a serverem

příkazy - definován pevně daný řídící jazyk, dají se vysílat ručně přes telnet na příslušný ftp port, ale obvykle je implementováno nějaké rozhraní (které může mít definováno vlastní „uživatelské“ příkazy)
odpovědi – na každý příkaz alespoň jedna, textový formát - tři číslice + popisný text, důležitá je pouze 1 číslice (1-5), zbytek jen pro „chytré“ klienty
dialog – klient vysílá příkazy, server zkouší provádět a vrací odpovědi

7) Koncepce protokolu NFS, přínosy a důsledky jeho bezestavovosti, význam mount serveru

transparentní sdílení souborů...od firmy Sun...není vázán na platformu...bezstavový – uzavřené požadavky klienta (vykonají se a stav serveru se nezmění), robustnost NFS – proto úspěšný, připouští pouze ty operace, které lze víckrát opakovat a skončí stejně...zakázáno OPEN, CLOSE, APPEND – mění stav
identifikace souborů pomocí file handles – trojice systém souborů, soubor, instance – jednoznačná definice
mount server – řeší těch pár otázek, které nejsou bezstavové (připojení adresového stromu, evidence zpřístupněných adresářů), bývá aplikační, zatímco sám NFS přímo v jádru (rychlý)

8) RPC - význam a princip fungování, vztah k NFS, způsob implementace

iluze, že akce probíhají u klienta – volání lokálních procedur (podprogramů), které ale odkazují kamsi na vzdálený server, lokální procedura skončí, až když skončí vzdálená...klient ani nepozná, že je na síti...zjednodušuje implementaci, je to jakási mezivrstva mezi serverem a klientem

Lekce 9 - Elektronická pošta

1) Filosofie a architektura SMTP pošty, hlavní protokoly a standardy

začít s málem (jen krátké ASCII zprávy), postupně se obohacovat (národní abecedy, přílohy)
model klient/server – server: transport, shraňování nedoručených zpráv...klient: stahování doručených, odesílání nových

SMTP – vlastní přenos zpráv
POP3,IMAP – stahování doručených zpráv se serveru ke klientovi
MIME – rozšíření
RFC 822 – doporučení formátování zpráv a adres

2) Struktura zprávy el. pošty, hlavička její nejdůležitější položky

hlavička – tělo – (volitelná) příloha...hlavička oddělena od těla prázdnou řádkou
v hlavičce – příjemce, odesilatel, datum, předmět, x-dalších atributů... formát definován standardem RFC 822
tělo – samotná text zprávy (původně ASCII text)
příloha – cokoliv co je soubor (až s MIME)

3) Adresy v SMTP poště a doručování podle MX záznamů

dříve – schránka@počítač (staticky vázané), dnes – uživatel@doména (DNS doména)
MX záznam – DNS resource record typu MX...uchovává IP adresu uzlu, kam se má mail doručit na základě toho, jaká je daná mailová adresa...cílový server ještě rozlišuje jednotlivé uživatele a maily „hází“ do příslušných schránek (seznam aliasů)

4) Doručování el. pošty s využitím tzv. doménového koše

Všechno jde do jedné schránky, odpadá seznam aliasů...pošta se třídí až v rámci doménového koše (uživatel definuje pravidla)...doménový koš nemusí být na stejném uzlu jako koncové schránky

5) Příkazy a odpovědi protokolu SMTP

vzájemný dialog odesílatel/příjemce...příkazy textového charakteru, odpovědi trojčíslí jako u FTP

6) SMTP dialog mezi poštovními servery

spojení navazuje odesílatel na základě MX záznamů DNS serverů...SMTP protokol běží na portu 25...napřed dochází k výměně identifikačních údajů, poté se přenáší samotná zpráva...přenos probíhá „online“

7) Podstata problému Open Relay v el. poště

Obecně SMTP přijme/odešle poštu odkudkoliv/kamkoliv – nebezpečí spamu
Zavést omezení – SMTP přijme zprávy jen od „jeho“ sítě/uživatelů, nebo přijímá jen zprávy pro „jeho“ sítě/uživatele

8) Problém netextových přenosů v SMTP, příčiny a možnosti jeho řešení

SMTP kóduje do 7-bitového ASCII...to může zničit (není to garantováno) 8-bitové záznamy (typicky UNICODE text nebo soubory)
princip – převést 8-bitů na 7-bitů...ale jak? Nesystematicky (pouze přílohy) – UUENCODE, BinHex...systémové řešení – standard MIME

9) Standard MIME, co řeší a jakým způsobem

existence příloh, národní abecedy, provázání s aplikacemi

definuje 2 kódování (8-bit na 7-bit) – Quoted Printable (jen netisknutelné kóduje do -=HEXCODE), Base64 (kóduje vše – binární kódy všecho, rozdělit na šestice (čísla 0-63) a ty jsou namapovány na převodní tabulku znaků)
typování – MIME type dat (dvousložkové – např. text/html, image/gif)
rozšíření formátování zpráv – nové položky v hlavičce

Lekce 10 - World Wide Web

1) Princip hypertextu, historie WWW

hypertext – provázání různých nesouvisejících témat (stránek) pomocí klíčových prvků (slova, obrázky), brouzdání mezi nimi
www – původně textová služba se skromnými prezentačními schopnostmi, nyní podporuje mnoho formátů (dříve řešily plug-iny, nyní přímo implementováno do browserů), výborné prezentační schopnosti...a je to platforma sdružující mnoho dalších služeb

2) Služba Gopher - koncepce, přednosti a nevýhody, srovnání s WWW

koncepce – existují menu (seznam prvků – buďto další menu a nebo odkaz na nějaký konkrétní soubor), obsah oddělen od menu...gopher servery umí odkazovat na sebe navzájem – jsou provázané a to celosvětově

u www splývá menu a obsah...to bylo atraktivnější a úspěšnější

3) Koncepce jazyka HTML a jeho vývoj

html – říká, co se má zobrazovat, ale ne jak se to má zobrazovat (to dělá až sám koncový browser)
původně jen velmi jednoduché možnosti, jednosměrná prezentace (ze serveru ke

klientovi)...později přibývají možnosti formátování...zpětná vazba na server (formuláře)...CSS (šablony pro vzhled)...skripty (programování v rámci html...např. Php)...i celé programy (Java applety, Active-X prvky)

4) Koncepce protokolu HTTP, rozdíly mezi verzemi 0.9, 1.0 a 1.1

jednoduchý protokol pro přenos – textová data přes TCP, port 80, bezstavový, komunikace: žádost – odpověď, jednoduché příkazy, číselné odpovědi (jako FTP), každý prvek 1 spojení

0.9 – velmi jednoduché, minimální dialog, jen www stránky
1.0 – hlavičky (domluva klient-server), i jiné objekty (podobně jako MIME typy), přetrvávají nedostatky
1.1 – bohatší hlavičky, virtuální www servery (víc na jedné IP), perzistentní připojení (více prvků v jednom spojení), pipelining, cache a proxy paměť, zabezpečení, výběr verze obsahu

5) Příkazy, odpovědi a hlavičky HTTP

GET – žádost o poskytnutí WWW stránky
HEAD – žádost o poslání hlavičky
POST – posílá data na server + žádost o odpověď
další příkazy se nepoužívají

hlavičky – upřesnění požadavků nebo odpovědí...např. REFER – adresa, odkud pochází obsah, EXPIRES – do kdy jsou data braná za platná (a cacheována), CONTENT-TYPE (typ obsahu)...

6) HTTP dialog mezi klientem a serverem

Klient vysílá požadavky, server odpovídá tříčíselnými kódy (zase jako FTP a SMTP)

1xx – informační odpověď
2xx – kladná
3xx – očekává se další aktivita
4xx – chyba klienta
5xx – chyba serveru

7) Bezestavový charakter HTTP komunikace a možnosti pro uchovávání historie relace

samo o sobě nejde...řešení: ukládat informace o relaci do URL (přidat tam tolik parametrů, kolik je potřeba)...klient to ale musí alespoň jednou inicializovat

lepší: cookies – RFC 2109...krátké textové soubory uchovávané u klienta, ve kterých se uchovávají informace o předchozím průběhu relace...přijímání cookies na počítači lze vypnout

8) Problematika virtuálních WWW serverů a její řešení

více samostatných WWW serverů na jednom PC – s různým obsahem...

IP-based řešení – každý server má svou vlastní IP adresu
name-based – stejná IP adresa, rozlišení pomocí hlavičky HOST (od http 1.1 povinná, většinou ji ale prohlížeče generují i pro 1.0)

9) Statické, dynamické a aktivní HTML dokumenty

statický dokument – v pevné podobě na serveru, jenom se zobrazuje (klasická www stránka v html)
dynamický – sám o sobě neexistuje (nejde cacheovat), vytvoří se až když klient podá požadavek – na základě aktuálního stavu, nejde ale dál měnit (např. php skripty)
aktivní – není dokončen, vytváří se až za chodu u klienta (aplety, activeX prvky, skripty)

10) Optimalizace fungování WWW, cache servery a jejich řízení

cache a proxy servery – optimalizují datové toky, ale komplikují komunikaci server-klient...je třeba rozlišovat mezi „end-to-end“ a „hop-by-hop“ (v hlavičkách)

cache – u klienta (efektivní, rychlé, ale nikdo další z toho nic nemá...), proxy-caching (data se cacheují někde uprostřed cesty, společné pro víc klientů), v rámci serveru (např. u generování dynamických stránek se to vyplatí – zrychluje odpovědi)...problémy – zahlcování, vypršení platnosti...ale zase je to optimálnější...není jasně daná hranice...řízení: pomocí hlaviček (Date – datum vytvoření, Expires – za jak dlouho objekt zastará, Cache-control – jak cacheování řešit, několik možných nastavení)

Lekce 11 - VOIP a IP telefonie

1) Vysvětlete pojmy VOD, VOIP, IP telefonie, internetová telefonie

VOD – voice over data, obecný pojem pro přenos zdigitalizovaného hlasu datovými sítěmi
VOIP – voice over IP, přenos zdigitalizovaného hlasu po protoklu IP (obecné označení pro libovolnou takovou technologii)
IP telefonie – obecné označení pro službu (obyčejně realizovanou pomocí VOIP)
internetová telefonie – varianta IP telefonie, přenosy přes veřejný internet

2) Architektura H.323, její hlavní části, jejich role a úkoly

Koncept od ITU, plnohodnotné řešení – robustní, pokrývá vše, ale drahé a náročné...celková architektura více protokolů...řeší – správu terminálů a zóny, kódování hlasu, řízení hovorů, signalizaci, přenos dat

gatekeeper – správce, terminálový adaptér – přepujue do sítě klasický telefon, MCU – řeší komunikaci více uzlů naráz, voice gateway – spojení mezi PSTN a IP sítí

3) Charakterizujte signalizaci a řízení hovorů v IP telefonii, naznačte jejich řešení v H.323

přes hovorový kanál (Call Channel) – spojovaný nad TCP, data přenášena samostatně pomocí UDP

signalizace – zařizování, vedení, ukončování spojení, H.225 – definuje formát zpráv, Q.931 – systém signalizace převzatý z ISDN, RAS – komunikace s gatekeeperem
řízení hovorů – paramtery spojení (kodeky, schopnosti zařízení, porty), H.245

4) Popište roli a způsob fungování gatekeeperu v H.323

Gatekeeper – nepovinný, pokud ale je – musí registrovat všechny uzly

zajištění – překlady adres, správu, řízení přístupu a přenosové kapacity
volitelně - zprostředkovová vyhledání volaného, autorizaci hovorů...vlastní komunikace přes něj nejde

5) SIP - k čemu slouží, jeho architektura (servery a jejich role)

pouze signalizační protokol (jen jeden) – řeší sestavení, dohled a ukončení relace mezi účastníky (2+), neřeší přenos dat a řízení hovoru (to dělá navazující protokol – SDP)
textový charakter, blízký http a filozofii WWW

proxy server – hledání spojení
redirect server – směruje žádosti o navázání spojení
location server – zná umístění koncových stanic
registrar server – správa členství koncových stanic

User Agent – v každém terminálu, UAC (klient) a UAS (server)...komunikace podobná jako u WWW

6) Metody a odpovědi protokolu SIP

opět podobné jako u FTP nebo http...odpovědi trojčíselné (zde 1-6)

INVITE – žádost o spojení
ACK – potvrzení žádosti
BYE – ukončení
CANCEL – zrušení (nenavázaného)
REGISTER – registrace UA do systému
OPTIONS – dotaz na možnosti

7) Způsob navazování relací v SIP-u, s využitím proxy a redirect serveru

zná umístění volaného – přímé oslovení přes třífázový handshake

nezná – osloví proxy server – ten zajistí nalezení sám – předá žádost o spojení dál k cílovému UA
nezná – osloví redirect server – ten mu řekne umístění volaného UA
ukončení spojení – můžou obě strany

8) SIP adresy a jejich překlad, využití při navazování relací

varianta url adres – sip:user@doména
překlad zajišťuje DNS, který najde registrar a location server dané domény...přes nový typ RR záznamu – SRV...UA se ptá proxy/redirect serveru – ten si najde záznam ve svém DNS a zeptá se příslušného registrar/location serveru...pak buď přímo naváže spojení (proxy) a nebo pošle informaci zpět k UA (redirect)

9) Protokol MGCP - jeho koncepce a využití, vztah k SIP a H.323

Odděluje a propojuje přepojování hovorů (dělá Media Gateway) od rozhodování o směrování hovorů (Call Agent)...SIP a H.323 fungují peer-to-peer, MGCP jako klient-server...je to doplněk SIP/H.323