Počítačová komunikace
Algoritmy a mechanismy směrování v sítích
Řízení toku v uzlech sítě a koncových zařízeních
Protokoly Internetu, adresace, programové rozhraní
Propojování sítí a funkce propojovacích prvků

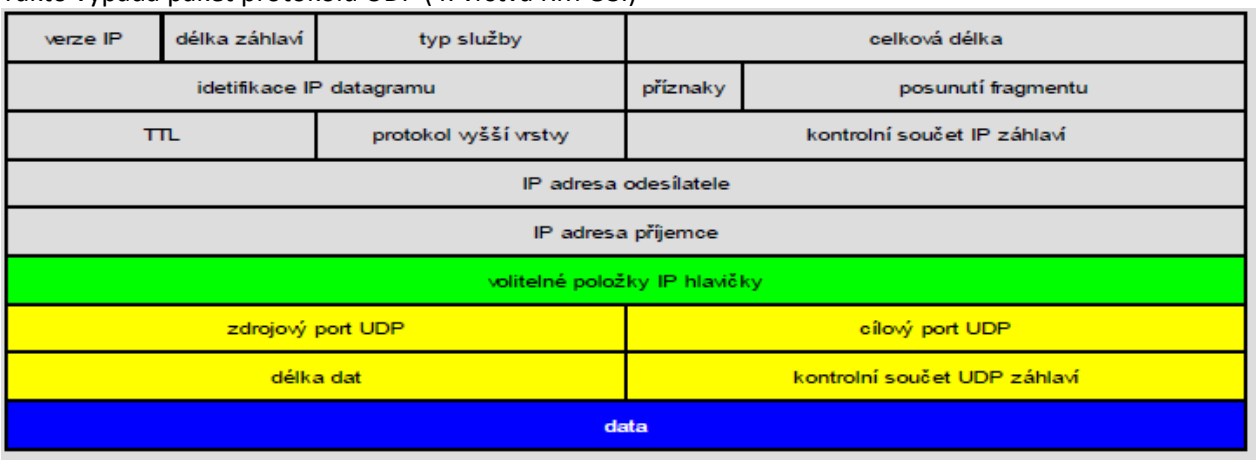
Obsah

Počítačová komunikace

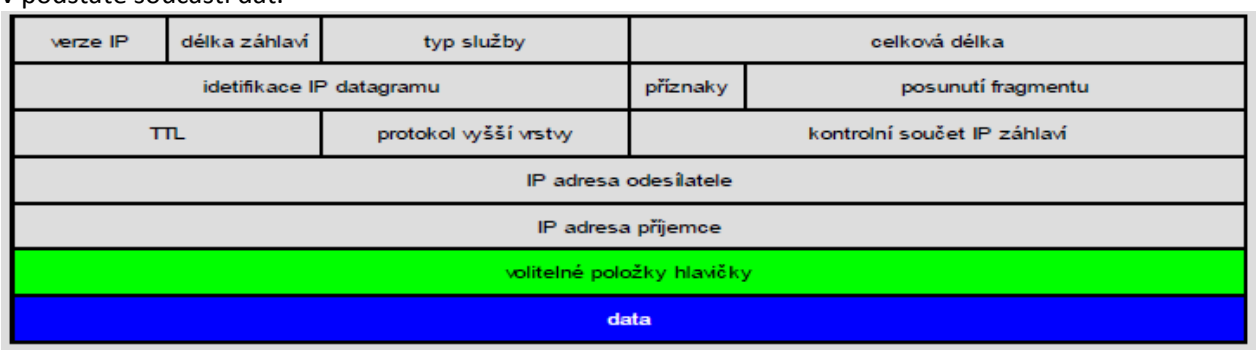
Pod pojmem počítačová komunikace si můžeme představit vzájemnou komunikaci mezi počítači za použití datových sítí. Pro vzájemnou komunikaci mezi počítači je důležité definovat obecně závazná pravidla, tak aby všechny stanice v síti byla schopna rozpoznat informace, které v datových sítích putují a dokázali je správně interpretovat.
Motivace: sdílení souborů, tiskáren, výpočetního výkonu, přenos informací atd.
Rozlišujeme dva druhy komunikace mezi počítači:

PEER-TO-PEER(P2P) rovnocenné propojení počítačů bez centrálního prvku – spojení každý s každým
CLIENT-SERVER propojení počítačů za použití centrálního prvku, počítače tedy nejsou propojeny každý s každým, ale přes centrální prvek (server)

K tomu, aby spolu počítače dokázali komunikovat, je potřeba zařídit, aby spolu mluvili stejným jazykem. Právě pro tyto účely byl navržen referenční model OSI/ISO (resp. v praxi použitý TCP/IP model). Díky standardizaci komunikačních modelů a komunikačních protokolů (rozuměj pravidel počítačové komunikace) si dnes můžeme stahovat soubory z internetu, pouštět video a posílat zprávy.
Komunikace mezi počítači je založena na posílání datových jednotek mezi počítači a jejich interpretací cílovou stanicí a následnou reakcí. Tyto datové jednotky se nazývají datagramy (IP packety). Když si připodobníme datagram k něčemu z reálného světa, jedná se vlastně o telegram, tedy krátkou zprávu, která musí obsahovat jisté náležitostí. Mezi tyto náležitosti patří adresát, odesílatel a samotná zpráva. Protože se ale nacházíme v počítačovém světě, umožnili nám inženýři komunikačních protokolů schovat do záhlaví zprávy o něco více informací, než kolik jich můžeme poslat v telegramu. Každá zpráva v počítačové komunikace obsahuje hlavičku zprávy a tělo zprávy (data). Hlavička zprávy obsahuje údaje důležité pro různá zařízení, přes která musí zpráva projít (síťová karta, routery, bridge atd.), data pak obsahují již jen samotnou informaci která je přenášena. A právě tady nastupuje role modelů komunikace. Při konstrukci zprávy na straně odesílatele se postupuje modelem shora dolů, zatímco při jejím čtení na straně příjemce se postupuje zdola nahoru. Jednotlivé vrstvy modelu, přes které je potřeba při tomto procesu projít, jsou určeny komunikačním protokolem, který je při komunikaci použit. Obecně funguje tento proces tak, že se na úrovni vrstvy komunikačního protokolu připraví záhlaví zprávy a připojí se data, poté se prochází modelem směrem dolů a připojují se do záhlaví zprávy další informace, které slouží nejen cílové stanici, ale také komunikačním prvkům přes které musí zpráva projít ke správnému zacházení s datagramem. Asi nejjednodušším způsobem, jak ukázat funkci tohoto principu je uvedení příkladu:
Takto vypadá paket protokolu UDP (4. vrstva RM OSI)



Když se podíváme na IP packet(3. vrstva), vidíme, že žluté položky z předchozího případu tam nejsou – stali se v podstatě součástí dat.



A na tomto principu to celé funguje, protokol vyšší vrstvy (v našem případě UDP) přidává do hlavičky údaje potřebné pro cílovou stanici na úrovni UDP, tyto údaje ale nejsou zajímavé pro zařízení, která komunikují na nižší vrstvě (IP protokol), IP protokol je k ničemu nepotřebuje.

Komunikačních protokolů existuje celá řada a slouží k různým účelům, uveďme si alespoň ty nejznámější:

Protokoly 2. Vrstvy RM OSI

Zajišťují přenos dat mezi přímo propojenými systémy, dělení proudu bitů na jednotku informace, kontrola integrity dat, adresace v rámci segmentu, zapouzdření dat vyšší vrstvy, bitově a znakově orientované protokoly

SLIP definuje pouze zapouzdření paketů na sériové lince
nedefinuje: adresaci, typ paketů, detekci chyb, kompresi, informace ke konfiguraci

CSLIP Compressed SLIP – vychází se SLIP, slouží tedy ke komunikaci po sériové lince, využívá faktu, že se během komunikace některé položky protokolů vyšších vrstev nemění a umí zredukovat hlavičky IP protokolu, čímž šetří šířku pásma

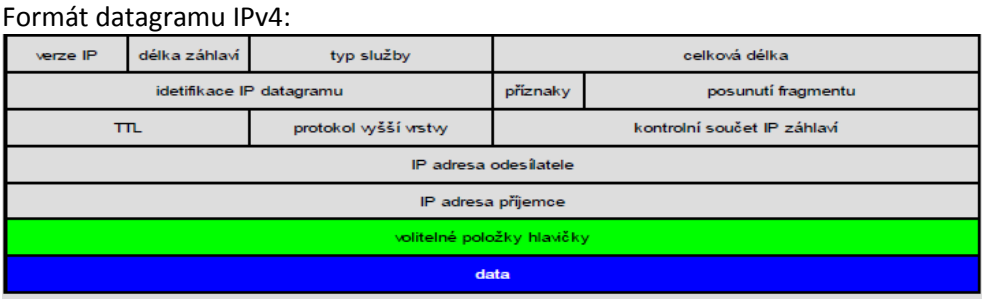
PPP Point-to-Point protokol – slouží k propojení dvou sítí po telefonní lince, příp. ISDN. Zapouzdřuje dva protokoly LCP – pro navázání spojení, NCP – pro přenos dat

ETHERNET II Tzv. DIX, definuje MAC adresy (HW adresy zařízení) a zapouzdření protokolů vyšších vrstev a přenos rámců v datových sítích

IEEE 802.3 v podstatě totéž co předchozí protokol, jen standardizovaný IEEE. Drobné odlišnosti v implementaci. Vzhledem k tomu, že se hojně používají obě varianty, musí být současná zařízení schopna zpracovávat rámce obou protokolů

Protokoly 3. Vrstvy RM OSI

IP podpora fragmentace (rozbití datagramů na menší celky, které jsou lépe přenositelné sítěmi), fragmentaci provádí libovolný směrovač, defragmentaci provádí cílová stanice. Jedná se o základní protokol síťové vrstvy, zapouzdřuje protokoly vyšších vrstev
Zavádí pojem IP adresa (viz adresace). Existují dvě varianty pro IPv4 a IPv6, liší se formátem použité IP adresy a drobně i formátem datagramu.
Formát datagramu IPv4:



verze IP – 4
délka záhlaví – po 32b
typ služby (ToS) – opomíjeno
- priorita 3b
- nízké zpoždění 1b
- vysoká propustnost 1b

celková délka – omezení na 64KB
identifikace

- příznaky
  - 0
  - DF … 1 nefragmentovat
  - MF … 0 poslední fragment
- posunutí fragmentu 0 první
- TTL – 0 likvidace paketu
- protokol vyšší vrstvy
  - 1 ICMP, 2 IGMP, 6 TCP, 17 UDP
  - 4 IP over IP, 97 Eth within IP
- volitelné položky – zahazuje se

ICMP Internet control message protocol. Přenáší se v IP datagramu, slouží k přenosu řídících a kontrolních informací

Protokoly 4. Vrstvy **viz. Protokoly internetu**
TCP/IP
UDP

Algoritmy a mechanismy směrování v sítích

Řeší otázku kam s ním? Myšleno s datagramem (packetem).

Směrovací metody:

Záplavové pošle data do všech portů kromě portu, ze kterého data přišla
Největší výhodou tohoto způsobu je nalezení nejkratší cesty k cíli bez nutnosti cokoliv nastavovat nebo počítat. Velkou nevýhodou je zahlcení všech segmentů sítě tedy i těch kde se cílová stanice nenachází. Je nutné řešit způsob jak likvidovat nadbytečné packety např. nastavením TTL (TimeToLive)

Náhodné náhodně vybere odchozí port
Tento způsob směrování nezaručuje omezenou dobu doručení

Izolované směrovač rozhoduje se na základě vnitřních mechanizmů, např. posílá do portu s nejkratší frontou (Horký brambor) nebo využívá metodu zpětného učení, kdy si směrovač buduje tabulku (ta obsahuje odesílatele, čas a směr odkud packet přišel) je nutná kombinace s jinou metodou

Statické někdo/něco napevno nastaví směrovací tabulky při budování sítě případně při změně topologie
Je možná kombinace s předchozími metodami
Nereaguje na změny topologie, je nutný zásah do konfigurace
Používá se například při výpadcích sítě nebo přetížení

Adaptivní směrovač si udržuje směrovací tabulku a neustále jí aktualizuje. Na základě tabulky potom směruje data v sítí

Hierarchické Využívá se v rozlehlých sítích, rozděluje síť na autonomní oblasti, přestupy mezi oblastmi obstarávají hraniční směrovače

Směrovací tabulky

Jednou z možností jak sestavovat směrovací tabulky na směrovačích je ruční konfigurace. Tento způsob směrování lze s úspěchem použít v málo rozlehlých sítích a v sítích kde nedochází příliš často ke změnám. Nevýhodou tohoto způsobu je potom to, že je při každé změně v síti (příp. výpadku stanic) nutná změna konfigurace směrovačů. Existují sice možnosti, jak ručně konfigurovat směrování i alternativními cestami, ale představa správy takové sítě je poměrně děsivá ;-)

Existují ovšem způsoby jak sestavovat směrovací tabulky automaticky pomocí kombinace zpětného učení a výměny informací mezi směrovači

Pro sestavení směrovací tabulek se používají dvě třídy protokolů, každý z nich je založen na trochu jiném principu jak nalézt nejlepší cestu packetu sítí.

Distance Vector

Tyto protokoly jsou založené na výpočtu vzdálenosti, resp. počtu přeskoků (hops). Při sestavování tabulek je používán např. Bellmanův-Fordův algoritmus pro určení nejkratší cesty v síti.

Hlavním zástupcem této rodiny je protokol

RIP **Routing information Protocol**

Tento protokol je založen na výměně kompletních směrovacích tabulek mezi směrovači. Každý směrovač pravidelně rozesílá informace o svých směrovacích tabulkách (obsahují informace o známém okolí). Ostatní směrovače tyto informace přijmou a doplní si informace ve svých tabulkách, na základě hoops count/počtu přeskoků potom aktualizuje směrovací tabulky podle nejlepší/nejkratší cesty. Je potřeba vyřešit problém výpadku linky. Tento problém se řeší např. použitím techniky Split horizon (Split horizon znamená, že router nebude posílat svůj update na rozhraní, ze kterého mu update zrovna přišel - až od RIPv2)

Byly vydány tři verze Routing Information Protocol (směrovacího protokolu) a to RIPv1, RIPv2 a RIPng.

RIP verze 1

Originální specifikace RIPu dle RFC 1058 používá směrování podle původních tříd IPv4 adres A, B nebo C. Periodické aktualizace směrování nezahrnují informace o masce sítě, protože podle původního systému je maska dána příslušností IP adresy do jedné ze tříd. Chybí tak podpora pro CIDR (Classless Inter-Domain Routing), což znemožňuje existenci různě velkých podsítí uvnitř jedné třídy IP adres. Všechny podsítě musí být stejně velké (tj. se stejnou maskou). Neexistuje zde podpora pro vzájemnou autentizaci routerů, a proto je protokol RIPv1 napadnutelný nejrůznějšími útoky.

RIP verze 2

Kvůli nedostatkům originální specifikace RIP, byla v roce 1993 vyvinuta druhá verze (RIPv2) a naposledy upravena v roce 1998. Zahrnovala možnost přenášet informace o masce sítě, tudíž podporovala CIDR (Classless Inter-Domain Routing). K udržení zpětné kompatibility, zůstalo omezení 15 skoků (hop count). Při správné konfiguraci může být RIPv2 plně kompatibilní se starší verzí. K lepšímu kompatibilitě slouží vlastnost compatibility switch. Ve snaze vyhnout se zbytečnému zatížení na hostiteli, jež se neúčastní směrování, RIPv2 vysílá celou směrovou tabulku všem sousedním směrovačům na adrese 224.0.0.9 (multicast), čímž se liší od RIPv1, který používá broadcast). Unicastové adresování je stále povoleno pro mimořádné účely.
RIPv2 včlenilo podporu pro vzájemnou autentizaci routerů. Hesla jsou však přenášena v nekódovaném textu, což je nedostatečné pro bezpečnou komunikaci v síti internet. Šifrovací (MD5) autentizace pro RIP byla představena v roce 1997.RIPv2 je Internetový Standard STD-56.

RIPng (RIP další generace)

který je definován v RFC 2080, je rozšířením RIPv2 zahrnující podporu IPv6 (Internetového Protokolu další generace). Hlavní rozdíly mezi RIPv2 a RIPng jsou:

- podpora IPv6 síťování
- podpora aktualizovaných autentizací (verifikace ověření identity osoby nebo procesu zabezpečovacím systémem) - RIPng nepodporuje (nahrazeno IPsec)
- připojování libovolných tagů k směrovačům (routerům) - RIPng nepodporuje
- kódování dalších skoků do každého směrovacího záznamu - RIPng vyžaduje specifické kódování na dalším skoku, pro set směrovacích záznamů

Omezení

- počet skoků (hop count) nesmí přesáhnout 15, v případě, že přesáhne, bude považován za neplatný
- uvnitř jedné třídy IP adres nemohou v RIP verzi 1 existovat různě velké podsítě

Mezi další reprezentanty **Distance vector** patří protokoly:

IGRP - Interior Gateway Routing Protocol

je proprietární protokol vyvinutý firmou CISCO. Patří také do rodiny distance-vector směrovacích protokolů. Odstraňuje některé limitace RIP protokolu (např. zvládá více hopů než původních 15) a vylepšuje vypočítávání metriky (zahrnuje více parametrů). Tento protokol používá rozdělení sítí na třídy (*classful routing*), které vede k plýtvání IP adresami. V současné době je tento protokol považován za zastaralý a byl nahrazen EIGRP.

EIGRP - Enhanced Interior Gateway Routing Protocol

je nástupce IGRP, který pracuje s takzvaným beztřídním směrováním (*classless routing*), umožňující vytvoření různě velikých sítí. Také implementuje Diffusing Update Algoritmus (DUAL), který zlepšuje routování a zabraňuje vytvoření smyček

Link State

Protokoly založené na výměně informací o stavu linek. Všechny uzly si udržují totožnou mapu sítě a na základě informací z ostatních uzlů ji aktualizují. Jednou z výhod oproti Distance Vector je menší zatížení sítě při výměně informací. Informace o změně stavu se posílají ihned. Nečeká se na aktualizační interval. Není omezená rozlehlost sítě (u DV max 15 přeskoků). Je možné zvolit i jinou metriku než jen počet přeskoků, např. kvalita nebo cena linek atp.

Hlavním zástupcem této rodiny je

OSPF Open shortest path first

je adaptivní hierarchický distribuovaný routovací protokol, provádějící změny v routovacích tabulkách na základě změny stavu v síti. Jedná se o nejpoužívanější routovací protokol uvnitř autonomních systémů.
Routery, používající tento protokol, si v pravidelných krátkých intervalech zvláštními zprávami (ECHO) kontrolují spojení se svými sousedními routery. Při zjištění jakékoliv změny zasílá oznámení všem routerům v síti, ty si pak podle nové informace přepočítají nové cesty v síti a podle toho upraví routovací tabulky.
Výpočet nejkratších cest se provádí Dijkstrovým algoritmem.
Dalším vylepšením tohoto protokolu je rozdělení autonomního systému na několik oblastí (proto hierarchický), ve kterých si routery vzájemně vyměnují sdělení o změnách v síti, ale mimo svou oblast je neposílají. O výměnu souhrnných informací mezi oblastmi se starají hraniční routery. Touto technikou se zamezuje zahlcování rozlehlých sítí informacemi o změnách při velkém počtu routerů v autonomním systému.

Výhody OSPF:
velmi rychlá konvergence
možnost členit velké oblasti na menší zóny

Řízení toku v uzlech sítě a koncových zařízeních

Pod tímto pojmem si lze představit soubor prostředků sloužících k zajištění doručení dat v potřebné kvalitě (QoS = Quality of Services)
V této oblasti se budeme zabývat způsoby, jak zajistit požadovanou kvalitou služeb. Tento problém není vždy zcela řešitelný. QoS se uplatňuje zejména při přenosu multimédií, IP telefonii atp.
V jednoduché síti se všichni uživatelé dělí o prostředky stejným dílem, tj. 5 uživatelů sdílí 10MBit linku, pak má každý uživatel k dispozici 2Mbit. Za normálních okolností není menší rychlost problém. Prostě jenom déle čekáte, než se vám načte webová stránka, případně stáhne soubor. Co ale dělat v případě, že hodláte využít službu, která potřebuje větší přenosovou kapacitu než ony 2MBit?
Odpověď je QoS.

QoS umožňuje:

- Rezervovat přenosovou kapacitu pro daný kanál
- Nastavit vyšší prioritu některým službám (např. ssh) a zkrátit jejich odezvu
- Omezit přenos na definovaný limit (např. omezení FTP, aby bylo možno přistupovat na WWW)
- Definovat maximální zpoždění dat

Jak to zařídit?
Řízením provozu
omezování provozu (traffic policing)

- ořezávání provozu
- absolutní přenos, průměrný přenos, špičkový přenos
- např. token bucket

tvarování provozu (traffic shaping)

- dodržení podmínek pro omezování provozu
- náročné na paměť
- změna zpoždění

Typem služeb
rozlišované služby (DiffServ)

- značkování paketů
- řízení mezi sousedy

integrované služby (IntServ)

- rezervace prostředků
- špatná adaptibilita

maximální snaha (best effort)

- původní řešení
- nejlépe jak to jde, všem stejně

Plánovací mechanizmy používané při řízení toku
FIFO -First in first out
nejjednodušší přístup -pakety jsou zpracovávány v pořadí jejich příchodu
Prioritní FIFO
pakety jsou rozděleny do dvou prioritních tříd: důležité pakety a méně důležité pakety. Pro každou třídu existuje speciální fronta, přičemž platí, že libovolný paket z fronty důležitých paketů má přednost před libovolným paketem z fronty méně důležitých paketů. Vzájemné mezi pakety z každé skupiny je jednotlivými frontami zachováno. Front (a tím pádem i prioritních tříd) může být i více než 2
WFQ -Weighted fair queuing
každý datový tok má svojí frontu. Pokud jeden tok např. posílá rychleji nebo větší pakety, než ostatní, projevuje se to jen na zpracování onoho toku a netrpí tím ostatní
Leaky Bucket
metaforicky se dá přirovnat ke kyblíku s dírou ve dně. Packety odtékají rychlostí, podle díry ve dnu a pokud kyblík přeteče, pakety se zahazují. Neumožní narozdíl od Token bucketu špičky v sítovém přenosu, protože díra ve dnu je pořád stejná

Token Bucket

v zásobníku na tokeny je určitý počet tokenů. Pokud přijde paket o velikosti N bytů na řadu, je odebráno ze zásobníku N tokenů. Pokud jejich počet nestačí, je s paketem nakládáno dle implementace (zahodí se atd), jinak je odeslán

Round Robin

funguje jako bariéra v datovém přenosu. Mechanismus Round Robin postupně po dobu krátkého časového kvanta propouští pakety z jednotlivých datových toků, které dorazily k bariéře, přičemž některé toky mohou mít přednost

Protokoly Internetu, adresace, programové rozhraní

Opakování je matkou moudrosti ;-)

Protokoly na linkové vrstvě

Zaručují přenos dat mezi přímo propojenými systémy.

SLIP -Serial Line IP

definuje **pouze** zapouzdření paketů na sériové lince

CSLIP -Compressed SLIP

pouze redukce záhlaví TCP a IP přenáší se změny položek záhlaví ignorují se změny záhlaví

PPP -Point to Point Protocol protokol umožňující autentizaci, šifrování a kompresi

Etherner II (DIX)

Protokoly na síťové vrstvě

IP -Internet Protocol

k čemu se používá

- základní protokol síťové vrstvy a celého Internetu
- vysílání datagramů na základě síťových IP adres obsažených v jejich záhlaví - každý datagram je samostatná datová jednotka, která obsahuje všechny potřebné údaje o adresátovi i odesilateli a pořadovém čísle datagramu ve zprávě datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě doručení datagramu není zaručeno, spolehlivost musí zajistit vyšší vrstvy (TCP, aplikace)
- poskytuje vyšším vrstvám síťovou službu bez spojení

funkce a činnosti vykonávané protokolem IP

- adresace koncových uzlů a sítí v IP intersíti
- vytváření IP paketů z paketů protokolů vyšší vrstvy
- směrování IP paketů přes IP intersíť
- fragmentace IP paketů
- stará se o segmentaci a znovusestavení datagramů do a z rámců podle protokolu nižší vrstvy (např. Ethernet)
- protokol IP je základním přenosovým prostředkem pro protokoly TCP/IP

fragmentace

- umožňuje vložení IP paketu do kratších rámců nižší vrstvy (MTU -Maximum Transmission Unit) fragmentaci provádí libovolný směrovač
- defragmentaci provádí koncový uzel možnost zakázání fragmentace

ICMP -Internet Control Message Protocol

- slouží k přenosu řídících hlášení, která se týkají chybových stavů a zvláštních okolností při přenosu(př. ping, traceroute) přenos chybových i řídících informací
- testování dostupnosti (Echo Req/Rep) řízení zahlcení a toku změna směrovací tabulky informace o masce časová synchronizace
- omezená implementace zahazování z bezpečnostních důvodů
- některé služby, např. echo, redirect...
- Echo Request … požadavek na odpověď, každý prvek v síti pracující na IP vrstvě by na tuto výzvu měl reagovat. Často to z různých důvodů není dodržováno.
- Echo Reply … odpověď na požadavek Destination Unreachable … informace o nedostupnosti cíle, obsahuje další upřesňující informaci
- Net Unreachable … nedostupná cílová síť, reakce směrovače na požadavek komunikovat se sítí, do které nezná cestu Host Unreachable … nedostupný cílový stroj Protocol Unreachable … informace o nemožnosti použít vybraný protokol Port Unreachable … informace o nemožnosti připojit se na vybraný port
- Redirect … přesměrování, používá se především, pokud ze sítě vede k cíli lepší cesta než přes defaultní bránu. Stanice většinou nepoužívají směrovací protokoly a proto jsou informovány touto cestou. Funguje tak, že stanice pošle datagram své, většinou defaultní, bráně, ta jej přepošle správným směrem a zároveň informuje stanici o lepší cestě.
- Redirect Datagram for the Network … informuje o přesměrování datagramů do celé sítě Redirect Datagram for the Host … informuje o přesměrování datagramů pro jediný stroj

- Time Exceeded … vypršel časový limit Time to Live exceeded in Transit … během přenosu došlo ke snížení TTL na 0 aniž byl datagram doručen Fragment Reassembly Time Exceeded … nepodařilo se sestavit jednotlivé fragmenty v časovém limitu(např pokud dojde ke ztrátěčásti datagramů)

Protokoly sloužící pro přidělení IP adresy
ARP -Address Resolution Protocol
RARP -Reverse Address Resolution Protocol
spolu s IP protokolem se používá k získání ethernetové MAC adresy sousedního stroje z jeho IP adresy
přidělení adresy bezdiskové stanici nepoužívá se
BOOTP -Bootstrap Protocol
starší protokol statické přidělení parametrů. Protokol BootP slouží k získání konfigurace. Klient ve svém požadavku (BOOTREQUEST) uvádí svou hardwarovou adresu a volitelně svou IP adresu, jméno serveru a požadovaný boot soubor. Server v odpovědi (BOOTREPLY) pošle požadovanou IP adresu, IP adresu gateway, svou IP adresu a jméno.
DHCP -Dynamic Host Configuration Protocol
DHCP protokol umožňuje prostřednictvím jediného DHCP serveru nastavit všem stanicím sadu parametrů nutných pro komunikaci v sítích používajících rodinu protokolů TCP/IP včetně parametrů doplňujících a uživatelsky definovaných. Významným způsobem tak zjednodušuje a centralizuje správu počítačové sítě (například při přidávání nových stanic, hromadné změně parametrů, nebo pro skrytí technických detailů před uživateli). DHCP servery mohou být sdruženy do skupin, aby bylo přidělování adres odolné vůči výpadkům.

Protokoly na transportní vrstvě
UDP -User Datagram Protocol
nepotvrzovaná služba bez spojení
použití (i příklady aplikací): stream -proud -tok dat vysílání internetových rádií a televizí(nezáležá na tom, zda nějaký packet nepřijde, prostě se jede dál)
Vlastnosti
rychlejší jak TCP, neřeší kontrolu došlých packetů a ani pořadí ve kterém přicházejí, může se stát, že z řady odeslaných packetů 1,2,3,4,5 vám dojdou ve stavu 4,2,5,3 s tím, že 1 vůbec nedošla
Porty
UDP používá porty, aby bylo možné rozlišit v počítači jednotlivé aplikace a správně jim doručit data, i když jich komunikuje v počítači více. Port je 16 bitová hodnota, která umožňuje používat porty z rozsahu 0-65535. Port 0 je rezervován, ale je možné ho použít, pokud odesílající proces neočekává žádnou odpověď Porty 1-1023 jsou tzv. dobře známé (anglicky well known ports) a na Unixech a odvozených operačních systémech jsou potřeba práva uživatele root, aby je bylo možné použít. Porty 1024-49151 jsou registrované porty. (Nemělo by se jich používat pro jiné aplikace, než jsou registrované u IANA.) Porty 49152-65535 jsou používány pro komunikaci klienta se serverem

TCP –Transmission Control Protocol
Použití (i příklady aplikací): posílání souborů (záleží na aplikaci, třeba u Torrentů funguje jak TCP tak UDP) HTTP protokol, neboli WWW stránky
Vlastnosti
kontrola došlých dat využívá algoritmu klouzající okénko, navázání spojení, záleží na pořadí
Porty
Několik příkladů: FTP (port 21 a 20), SMTP (port 25), DNS (port 53) a HTTP (port 80). Registrované porty jsou typicky používané aplikacemi koncových uživatelů při otevírání spojení k serverům jako libovolné čísla zdrojových portů, ale také mohou identifikovat služby. Dynamické/privátní porty mohou být také používány koncovými aplikacemi, ale není to obvyklé
TCP spojení

navázání	třífázový protokol - klient pošle SYN na server, ten pátky pošle SYN + ACK a klient zpátky pošle ACK -v ideálním případě se spojí
komunikace	řazení dat do správného pořadí, znovuposílání ztracených rámců, odstranění duplikací, kontrola přetečení zásobníku pro zpracování, využití klouzavého okénka
ukončení	třífázový protokol -klient pošle FIN na server, ten pátky pošle FIN + ACK a klient zpátky pošle ACK -v ideálním případě spojení skončí

rámcová znalost řízení toku=změna velikosti okénka, Nagleův algoritmus (pro využití kapacity kanálu při malém zatížení), odesílání potvrzení po větších blocích, přenastavení timeoutu atp.

Rozdíly mezi TCP a UDP

TCP spojově orientovaný protokol Spojení může otevřít klient nebo server a pak mohou už být posílána jakákoliv data oběma směry

Spolehlivost – TCP používá potvrzování o přijetí, opětovné posílání a překročení časového limitu. Pokud se jakákoliv data ztratí po cestě, server si je opětovně vyžádá. U TCP nejsou žádná ztracená data, jen pokud několikrát po sobě vyprší časový limit, tak je celé spojení ukončeno, zachování pořadí – jestliže se odešlou 2 zprávy, jedna po druhé, první dorazí nejdřív k serveru. Pokud data dorazí ve špatném pořadí, TCP protokol se postará o to, aby některá data pozdržela a finálně je předala správně seřazená vyšší režie

TCP protokol potřebuje tři pakety jen pro otevření spojení, což však umožňuje zaručit spolehlivost celého spojení

UDP je jednodušší protokol založený na odesílání nezávislých zpráv bez záruky

protokol už neumožňuje ověřit, jestli došla zamýšlenému příjemci. Datagram se může po cestě ztratit. UDP nemá žádné potvrzování, přeposílání, ani časové limity, nezachovává pořadí, jestliže odešleme dvě zprávy jednomu příjemci, nemůžeme předvídat, v jakém pořadí budou doručeny

jednoduchost – nižší režie, než u TCP (není zde řazení, žádné sledování spojení atd.)

Protokol síťové vrstvy ještě jednou, ale trochu jinak

IPv6

Formát datagramu

datagram v IPv6 odráží snahu tvůrců vytvořit co nejmenší záhlaví a to pouze se základním obsahem. Tvůrci přesunuli málokdy používané části datagramu z IPv4 do tzv. „rozšiřujících hlaviček“ v IPv6. Jde zejména o položky potřebné k fragmentaci datagramu (ty byly v každém IPv4 datagramu) Pro základní záhlaví byla stanovena nová pevná velikost, takže položka „velikost záhlaví“ byla vynechána. Při dnešní nízké chybovosti a vysoké propustnosti linek, ztratilo opodstatnění pole „kontrolní suma“. Principy použité v IPv4 (Fragmentace, Směrování) prošli jen nepatrnými změnami

Řetězení hlaviček

k základnímu záhlaví v IPv6 je možné připojit i rozšiřující hlavičky a to prostřednictvím položky „další hlavička“ v základním záhlaví. Tuto položku obsahuje i každá rozšiřující hlavička, přičemž je možné tímto způsobem připojit několik doplňujících záhlaví. Hodnota položky „další hlavička“ taktéž zastupuje položku „protokol“ z IPv4, kterou se určuje typ údajů nesený za celým záhlavím

Fragmentace

MTU (Maximum transfer unit) je hodnota udávající maximální velikost přenosové jednotky na úrovni vrstvy síťového rozhraní (nejnižší vrstvy) komunikačního modelu TCP/IP.

Protokol IP umožňuje rozdělení velkého IP paketu na menší celky (fragmenty). Každý fragment je samostatný IP paket -má svou (novou) IP hlavičku. Každý fragment potom putuje k cíli samostatně, nezávisle na ostatních. U příjemce jsou jednotlivé fragmenty poskládány a je z nich sestaven původní celek -IP paket. Každý fragment je samostatný IP paket, z čehož vyplývá, že může být podle potřeby opět rozdělen na další fragmenty, provádí se pouze u odesilatele informace o fragmentaci v rozšiřující hlavičce hlavičky až k fragmentační – nefragmentovatelné minimální MTU (Maximum Transmission Unit) pro IPv6 je 1280B (ale očekává se, že klient provede MTU Path discovery) vyhledávání MTU cesty (MTU Path discovery)

ICMPv6 pravidelné opakování (cca 10min) nemusí se implementovat

Automatická konfigurace

dva typy automatické konfigurace

Stavová konfigurace

charakterizuje jí v IPv4 používaný systém DHCP, který je upravený pro potřeby IPv6 a nese název DHCPv6

nevyužívá broadcast

1 DHCP Unique Identifier (DUID) jednoznačně identifikuje uzel linková adresa a čas přiděleno výrobcem

2 Identity Association (IA) -jednoznačně identifikuje rozhraní

3 vyhledání všech serverů (FF02::1:2 – adresa agenta) –> solicit -advertise

4 oslovení zvoleného serveru podle DUID (FF02::1:2)! –> request – reply

5 obnovení -renew, rebind, release, confirm

6 rekonfigurace vyvolaná serverem -reconfigure

Bezstavová konfigurace

Kreativní novinkou je konfigurace bezstavová, která nevyžaduje žádné servery. Jejím základním pilířem je tak zvané objevování sousedů (neighbor discovery). Základní myšlenka je celkem prostá: každý směrovač v určitých intervalech rozesílá do sítí, k nimž je připojen, tak zvané ohlášení

směrovače. V něm jsou obsaženy základní informace -především prefixy adres dané sítě a zda on sám může sloužit pro předávání paketů ven (jako implicitní směrovač, default gateway) Z ohlášení směrovačů (o které může při startu aktivně požádat pomocí výzvy směrovači) se počítač dozví, jaké adresy používá zdejší síť. K nim si doplní zbývající část (typicky 64 bitů), která se jednoznačně generuje z jeho Ethernetové adresy. Tak získá platné IPv6 adresy pro své rozhraní. Také si vytvoří seznam implicitních směrovačů, kterým bude předávat pakety směřující mimo síť. Pokud je jich víc, zpočátku je prostě střídá a směrovací tabulku si postupně vylepšuje na základě jejich upozornění (ICMP přesměrování), pokud paket k určitému cíli poslal nevhodným směrovačem.

Mobilita

Základní charakteristikou mobilních zařízení je, že za provozu přecházejí z jedné sítě do druhé (např. když komunikujete ze svého notebooku během cesty z Prahy do Vídně) a tudíž mění svou IP adresu. To nepředstavuje problém, pokud spojení navazuje mobilní počítač -prostě použije svou aktuální adresu. Potíže nastanou, pokud se někdo chce spojit s cestujícím počítačem. Řešení je poměrně prosté. Každý počítač má svou domácí síť a jí odpovídající domácí IP adresu. Tato adresa je zanesena v DNS a tu také použije externí stroj při pokusu o spojení. Pokud je počítač právě na cestách, zastupuje jej domácí agent. Tuto roli hraje jeden ze směrovačů v domácí síti mobilního stroje (podpora mobility zahrnuje i způsob jeho automatického výběru, aby nemusel být konfigurován staticky). Domácí agent na sebe přesměruje data určená mobilnímu stroji (při objevování sousedů odpovídá místo něj). Mobilní uzel průběžně informuje svého domácího agenta o aktuální IP adrese

Navázání spojení probíhá následovně:

- externí počítač zašle paket s žádostí o navázání spojení na domácí (trvalou) adresu mobilního uzlu
- jeho domácí agent ji zachytí a předá tunelem mobilnímu uzlu.
- ten odpoví žadateli a zároveň zahájí proces nazvaný optimalizace cesty. Jeho cílem je informovat protějšek o aktuální adrese mobilního uzlu (Situaci komplikují bezpečnostní mechanismy, aby se kdokoli nemohl prohlásit za jiný počítač na cestách)
- jakmile se protějšek dozví aktuální adresu mobilního uzlu, probíhá další komunikace přímo mezi nimi

Rozdíly oproti IPv4

Zjednodušení hlavičky

zjednodušuje původní hlavičku odstraněním přebytečných polí a zřetězuje volitelné podhlavičky se základní IP hlavičkou

mimo základní hlavičky se používají volitelné podhlavičky pro směrování, fragmentaci a ověřování přístupu např. podhlavičky s názvem Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encalupsulating Security Payload tyto se používají jen v případě požadavku na danou funkci

Rozšíření adresního prostoru IP adres

IPv6 rozšiřuje adresní prostor z původních 32 bitů na 128 bitů. (to už je hodně adres ☺ to by mohlo chvíli stačit)

Automatická konfigurace uzlů

protokol má zapracované mechanismy automatického předělování konfiguračních údajů sítě jeho jednotlivým uzlům odpadá tak manuální předělování IP adres na lokálních uzlech, respektive směrovačích sítě

Bezpečnostní procedury

nový protokol má přímo zabudované bezpečnostní procedury ověřování přístupu (Authentication) a kódování (Encryption) na úrovni IP komunikace. IPv6 umožňuje volitelný výběr bezpečnostních metod a parametrů, implicitně se však používají metody autorizace přístupu MD5 a kódování podle DES (Data Encryption Standard) IP má bezpečnostní vlastnosti implementované prostřednictvím volitelných podhlaviček AH (Authentication Head) a ESP (Encapsulating Security Payload)

podpora multimediálních aplikací

nárůst aplikací požadujících komunikaci v reálném čase (real-time) v síti Internet si vyžádal i zvýšení podpory tohoto druhu komunikace na straně IP protokolu -IP za tímto účelem používá metodu označení toku návěští (Flow label) tokem rozumíme IP pakety přenášené mezi zdrojem a cílem v IP síti se speciálními požadavky na přenos přiřazením číselného návěští danému toku prostřednictvím protokolu RSVP (Resource Reservation Protocol), může IP směrovač obsluhovat přenos paketů různých toků odlišným způsobem

Adresace

IP adresa je v číslo, které jednoznačně identifikuje síťové rozhraní v počítačové síti, která používá IP (internetový protokol). V současné době je nejrozšířenější verze IPv4, která používá 32bitové adresy zapsané dekadicky po jednotlivých oktetech (osmicích bitů), například 192.168.0.1. Z důvodu nedostatku IP adres bude nahrazen protokolem IPv6, který používá 128bitové IP adresy.
IP adresa slouží k rozlišení síťových rozhraní připojených k počítačové síti. Síťovým rozhraním může být síťová karta (Ethernet, Wi-Fi), IrDA port, ale může se jednat i o virtuální zařízení (loopback, rozhraní pro virtuální počítač a podobně).
Zkratka IP znamená Internet Protocol, což je protokol, pomocí kterého spolu komunikují všechna zařízení v Internetu. Dnes nejčastěji používaná je jeho čtvrtá verze (IPv4), postupně se však bude přecházet na novější verzi 6 (IPv6). V jiných protokolech se adresování jednotlivých zařízení může provádět jinak (viz např. MAC adresa).
IP protokol byl původně vyvinut pro potřeby komunikace v Internetu. IP adresa musí být v dané síti jednoznačná (jedno rozhraní může mít více IP adres, ale stejná IP adresa nemůže být na více rozhraních), avšak lze používat NAT a privátní IP adresy (viz níže).
Veškerá data jsou mezi síťovými rozhraními přenášena v podobě IP datagramů.
Jelikož by pro běžné uživatele počítačových sítí bylo velice obtížné pamatovat si číselné adresy, existuje služba DNS (*Domain Name System*), která umožňuje používat snadněji zapamatovatelná doménová jména počítačů, která jsou automaticky převáděna na IP adresy.

Adresy v IPv4
V IPv4 je adresou 32bitové číslo, zapisované po jednotlivých bajtech, oddělených tečkami. Hodnoty jednotlivých bajtů se zapisují v desítkové soustavě, např. **192.168.48.39**
Takových čísel existuje celkem 2^{32} = 4 294 967 296. Určitá část adres je ovšem rezervována pro vnitřní potřeby protokolu a nemohou být přiděleny. Dále pak praktické důvody vedou k tomu, že adresy je nutno přidělovat hierarchicky, takže celý adresní prostor není možné využít beze zbytku. To vede k tomu, že v současnosti je již znatelný nedostatek IP adres, který řeší různými způsoby: dynamickým přidělováním (tzn. např. každý uživatel dial-up připojení dostane dočasnou IP adresu ve chvíli, kdy se připojí, ale jakmile se odpojí, je jeho IP adresa přidělena někomu jinému; při příštím připojení pak může tentýž uživatel dostat úplně jinou adresu), překladem adres (Network address translation) a podobně. Ke správě tohoto přidělování slouží specializované síťové protokoly, jako např. DHCP.

Struktura adresy
Adresa se v IPv4 dělí na tři základní části:

adresa sítě	*adresa podsítě*	*adresa počítače*

Koncepce internetu jako sítě složené ze sítí a tomu odpovídající struktura adres patří mezi novinky zavedené IP. Má velký význam pro směrování – mimo cílovou síť se směruje podle adresy sítě, a až když je IP datagram doručen do ní, začíná se brát ohled i na detailnější části adresy.
Původní koncept adresace nepočítal s podsítěmi, definoval jen adresu sítě a počítače. Později se však toto členění ukázalo jako příliš hrubé a lokální část adresy se rozdělila na podsíť a počítač. Obecně platí, že mezi adresami ve stejné podsíti (mají totožnou adresu sítě a podsítě) lze data dopravovat přímo – dotyční účastníci jsou všichni propojeni jedním Ethernetem či jinou lokální sítí. Jakmile se adresa cíle nachází v jiné síti, bude potřeba datagram předat příslušnému směrovači, aby jej dopravil dál – viz směrování.
Adresu sítě pro danou koncovou síť přiděluje poskytovatel připojení (oficiálně ji přiděluje lokální registrátor, ale tím bývá právě poskytovatel). Je třeba o ni požádat prostřednictvím standardních formulářů. Strukturu lokální části adresy – zda bude rozdělena na podsítě a jaká její část bude případně věnována adrese podsítě a jaká adrese počítače – určuje správce dotyčné sítě. Ten také přiděluje adresy.
Hranici mezi adresou podsítě a počítače určuje **maska podsítě (subnet mask)**. Jedná se o 32bitovou hodnotu zapisovanou stejně jako IP adresa. V binárním tvaru obsahuje jedničky tam, kde se v adrese nachází síť a podsíť, a nuly tam, kde je počítač. Maska podsítě je společně s IP adresou součástí základní konfigurace síťového rozhraní, často se předává protokolem DHCP.

Adresování sítí a podsítí
V úplných začátcích Internetu bylo toto rozdělení adresy na síť a lokální část fixní: prvních osm bitů adresy určovalo síť, zbytek pak stroj v síti. To však umožňovalo pouze 256 sítí (v každé však mohlo být přes 16 milionů stanic), takže s nástupem lokálních sítí bylo zřejmé, že bude potřeba tento systém změnit. Adresy se proto rozdělily do tříd podle

toho, jaká část adresy určuje síť a jaká určuje stanici v síti (přičemž dvě třídy byly vyhrazeny pro zvláštní účely). Odpovídající třída se poznala podle hodnoty prvních několika bitů (a pro člověka podle prvního bajtu):

Třídy IP adres

Třída	začátek (bin)	1. bajt	standardní maska	bitů sítě	bitů stanice	sítí	stanic v každé síti
A	0	0–127	255.0.0.0	7	24	$2^7 = 128$	$2^{24} = 16\,777\,216$
B	10	128–191	255.255.0.0	14	16	$2^{14} = 16384$	$2^{16} = 65\,536$
C	110	192–223	255.255.255.0	21	8	$2^{21} = 2\,097\,152$	$2^8 = 256$
D	1110	224–239	*multicast*				
E	1111	240–255	*vyhrazeno jako rezerva*				

Rozsahy IP adres a masky sítě

Třída	1. bajt	minimum	maximum	maska podsítě
A	0–127	0.0.0.0	127.255.255.255	255.0.0.0
B	128–191	128.0.0.0	191.255.255.255	255.255.0.0
C	192–223	192.0.0.0	223.255.255.255	255.255.255.0
D	224–239	224.0.0.0	239.255.255.255	255.255.255.255
E	240–255	240.0.0.0	255.255.255.255	—

(Například síť třídy A je taková síť, kde první číslo čtyřčíselné IP adresy označuje síť a zbylá tři čísla označují adresu hostitele. Třída B používá první dvě pro označení sítové adresy a zbývající dvě pro hostitele a síť třídy C používá první tři čísla pro označení sítě a poslední pro označení hostitele.)

Postupem času se však i toto rozdělení ukazovalo jako velice nepružné a s rostoucím nedostatkem adres se hledaly způsoby na vylepšení tohoto systému. Od roku 1993 se pak začal používat tzv. Classless Inter-Domain Routing (CIDR, *beztřídní mezidoménové směrování*), ve kterém je možno předěl mezi adresou sítě a lokální částí adresy umisťovat libovolně. Daná adresa se pak značí kombinací prefixu a délky ve formě

192.168.24.0/21

což znamená, že takto určená síť je určena prvními 21 bity adresy (maska by byla 255.255.248.0), zbytek je adresa stanice (případně podsítě), takže tato síť používá rozsah adres 192.168.24.0–192.168.31.255.

Vyhrazené adresy

Nejnižší adresa v síti (s nulovou adresou stanice) slouží jako označení celé sítě (např. „síť 192.168.24.0“), nejvyšší adresa v síti (adresa stanice obsahuje samé binární jedničky) slouží jako adresa pro všesměrové vysílání (broadcast), takové adresy tedy nelze použít pro normální účely.

Adresy 127.x.x.x (tzv. *localhost*, nejčastěji se používá adresa 127.0.0.1) jsou rezervovány pro tzv. loopback, logickou smyčku umožňující posílat pakety sám sobě.

Dále jsou vyčleněny rozsahy tzv. interních (neveřejných) IP adres (tzv. *privátní IP adresy*), které se používají pouze pro adresování vnitřních sítí (např. lokálních), na Internetu se nikdy nemohou objevit. Jako neveřejné jsou určeny adresy:

ve třídě A: 10.0.0.0 až 10.255.255.255 (celkem 256krát 65 536 adres; tj. 16 777 216 adres, z nichž je použitelných jen 16 646 144)

ve třídě B: 172.16.0.0 až 172.31.255.255 (celkem 16krát 65 536 adres; tj. 1 048 576 adres, z nichž je použitelných jen 1 040 384)

ve třídě C: 192.168.0.0 až 192.168.255.255 (celkem 256krát 256 adres; tj. 65 536 adres, z nichž je použitelných jen 65 024)

Adresy v IPv6

Trvalejším řešením problémů s nedostatkem adres by měla být nová verze protokolu, označovaná IPv6, která se ovšem zatím rozšiřuje jen velice pozvolna. V IPv6 adresa má délku 128 bitů, což znamená, že počet možných adres je $2^{128} \approx 3\times10^{38}$. To je astronomicky velké číslo; pro představu: teoreticky se jedná o 6×10^{23} IP adres na 1 m² zemského povrchu. I pokud započítáme, že i v IPv6 je potřeba velkou část adres rezervovat a adresní prostor opět nelze dokonale využít, je těchto adres dostatek na to, aby každé zařízení připojitelné do internetu dostalo svou vlastní jedinečnou adresu.

Adresa IPv6 se zapisuje jako osm skupin po čtyřech hexadecimálních číslicích, například:

2001:0718:1c01:0016:0214:22ff:fec9:0ca5

Úvodní nuly v každé skupině lze ze zápisu vynechat. Výše uvedenou adresu tedy lze psát ve tvaru
2001:718:1c01:16:214:22ff:fec9:ca5
Pokud adresa obsahuje několik po sobě jdoucích nulových skupin, lze místo nich zapsat jen „::“. Tato zkratka smí být v adrese jen jedna. Používá se často u prefixů pro nulový konec adresy či u speciálních adres, jako je loopback (smyčka), jejíž tvar ::1 je podstatně příjemnější, než 0000:0000:0000:0000:0000:0000:0000:0001.
Adresní architekturu IPv6 definuje RFC 4291. Zavádí tři typy adres:

Individuální (unicast) která identifikují právě jedno síťové rozhraní.
Skupinové (multicast) označují skupinu síťových rozhraní, jejímž členům se mají data dopravit. Skupinově adresovaný datagram se doručuje všem členům skupiny.
Výběrové (anycast) označují také skupinu síťových rozhraní, data se však doručují jen jejímu nejbližšímu členovi.

IPv6 neobsahuje všesměrové (broadcast) adresy. Byly nahrazeny obecnějším modelem skupinových adres a pro potřeby doručení dat všem zařízením připojeným k určité síti slouží speciální skupinové adresy (např. ff02::1 označuje všechny uzly na dané lince).
IPv6 zavádí také koncepci dosahu (scope) adres. Adresa je jednoznačná vždy jen v rámci svého dosahu. Nejčastější dosah je pochopitelně globální, kdy adresa je jednoznačná v celém Internetu. Kromě toho se často používá dosah linkový, definující jednoznačnou adresu v rámci jedné linky (lokální sítě, např. Ethernetu). Propracovanou strukturu dosahů mají skupinové adresy (viz níže).
Adresní prostor je rozdělen následovně:

prefix	význam
::/128	neurčená
::1/128	smyčka (loopback)
ff00::/8	skupinové
fe80::/10	individuální lokální linkové
ostatní	individuální globální

Výběrové adresy nemají rezervovánu svou vlastní část adresního prostoru. Jsou promíchány s individuálními a je otázkou lokální konfigurace, aby uzel poznal, zda se jedná o individuální či výběrovou adresu.

Individuální adresy
Strukturu globálních individuálních IPv6 adres definuje RFC 3587. Je velmi jednoduchá a de facto odpovídá (až na rozměry jednotlivých částí) výše uvedené struktuře IPv4 adresy.

n bitů	64-n bitů	64 bitů
globální směrovací prefix	adresa podsítě	adresa rozhraní

Globální směrovací prefix je de facto totéž co adresa sítě, následuje adresa podsítě a počítače (přesněji síťového rozhraní). V praxi je adresa podsítě až na výjimky 16bitová a globální prefix 48bitový. Ten je pak přidělován obvyklou hierarchií, jejíž stávající pravidla jsou:

- první dva bajty obsahují hodnotu 2001 (psáno v šestnáctkové soustavě)
- další dva bajty přiděluje regionální registrátor (RIR)
- další dva bajty přiděluje lokální registrátor (LIR)

Reálná struktura globální individuální adresy tedy vypadá následovně:

16 bitů	16 bitů	16 bitů	16 bitů	64 bitů
2001	přiděluje RIR	přiděluje LIR	adresa podsítě	adresa rozhraní

Adresa rozhraní by pak měla obsahovat modifikovaný EUI-64 identifikátor. Ten získáte z MAC adresy jednoduchým postupem: invertuje se druhý bit MAC adresy a doprostřed se vloží dva bajty obsahující hodnotu fffe. Z ethernetové adresy 00:14:22:c9:0c:a5 tak vznikne identifikátor 0214:22ff:fec9:0ca5.

Skupinové adresy
Struktura skupinové adresy je definována v RFC 4291. Vypadá takto:

8 bitů	4 bity	4 bity	112 bitů
ff	příznaky	dosah	identifikátor skupiny

Příznaky jsou definovány tři:

- **T (Transient)** rozlišuje adresy dočasné (T=1) od trvalých (T=0)
- **P (Prefix)** se používá pro skupinové adresy vycházející z individuálního prefixu definované v RFC 3306.
- **R (Rendezvous Point)** umožňuje do skupinové adresy zakódovat adresu rendezvous pointu pro protokol PIM-SM (definuje RFC 3956).

Velmi podstatná je hodnota *dosahu*, která omezuje šíření skupinově adresovaných dat. Pokud je dosah řekněme 2, je adresa omezena na jednu linku (lokální síť) a nebude šířena za její hranice. Čili datagram se předá řekněme v rámci jednoho Ethernetu, ale žádné z připojených zařízení je už nebude šířit nikam dál. Definované dosahy jsou:

hodnota	význam
0	rezervováno
1	**jediné rozhraní (interface-local)**
2	**linka, síť linkové vrstvy (link-local)**
3	rezervováno
4	**administrativně definovaná síť (admin-local)**
5	**místo, koncová síť (site-local)**
6	nepřiřazeno
7	nepřiřazeno
8	**organizace (organization-local)**
9	nepřiřazeno
A	nepřiřazeno
B	nepřiřazeno
C	nepřiřazeno
D	nepřiřazeno
E	**globální**
F	rezervováno

Programové rozhraní

Připojení počítače k místní síti je možné pouze tehdy, jsou-li k dispozici následující technické (HW) a programové (SW) součásti:

- přenosové médium (kabel)
- síťová karta (tzv. adaptér)
- obslužný program pro adaptér (tzv. driver)
- síťový operační systém

Většina současných výrobců sítí má snahu o to, aby jejich síť mohla spolupracovat s co největším množstvím ostatních (konkurenčních) sítí. Z těchto důvodů dochází ke snaze vytvořit síť (v rámci možností) modulární, aby byla možná co největší zaměnitelnost jednotlivých komponent od více výrobců. V praxi se tento trend dostal až tak daleko, že ve více sítích je možné použití síťových karet různých výrobců – příkladem je např. NOVELL. Výrobci sítí proto mají snahu své výrobky navrhovat tak, aby vyhovovaly tzv. VRSTVOVÉ STRUKTUŘE KOMUNIKACE V SÍTI podle doporučení organizace ISO.

Programové vybavení

Každé komunikující zařízení musí být vystrojeno příslušným programovým vybavením (rozprostřeným přes síť) To na nižší úrovni řídí tok dat mezi koncovými zařízeními prostřednictvím pasivních prostředků sítě, při spolupůsobení aktivních prostředků sítě. Na vyšší úrovni zajišťuje větší zabezpečenost přenosu (protokoly) a umožňuje vstup do síťových aplikací.

Programové vybavení na nižší úrovni

zajišťuje spolupráci koncových zařízení s komunikačními zařízeními, prostřednictvím ”protokolů” zajišťuje výměnu dat mezi koncovými zařízeními a aktivními prostředky sítě. Je většinou závislé na typu komunikačního zařízení a počítačové sítě.

Příklady: Ovladače síťových karet, ARP protokol, CSMA/CD, PPP.

Programové vybavení na vyšší úrovni

je tvořeno řadou protokolů vloženou na programové vybavení koncového zařízení. Slouží k zajištění cílové služby nebo k zpřístupnění síťového prostředku. Obvykle je tato část programového vybavení nezávislá na vybavení nižší úrovně, typu komunikačního zařízení nebo počítačové sítě.

Příklady: protokoly TCP/IP, UDP, SNMP, TELNET, RSA, DCE.

Propojování sítí a funkce propojovacích prvků

V současném světě sítí se využívají při budování sítí především následující prvky jako prvky propojovací

Repeater (Opakovač)

Při šíření signálu mediem dochází k jeho zeslabení vlivem útlumu, rušení atd. Proto je při přenosu signálu na dlouhou vzdálenost (co je dlouhá vzdálenost závisí na použitém přenosovém mediu) potřeba signál vhodným způsobem posilovat. K tomuto účelu slouží lze použít opakovač neboli repeater. Jedná se o zařízení, které pracuje přímo se signálem (druh signálu a tedy i technologie zesílení záleží na použitém mediu) pracuje na první (fyzické) vrstvě OSI/ISO modelu. Smyslem tohoto zařízení není nic jiného než přijmout signál, posílit ho poslat (zopakovat) dál. Opakovač musí šířit kolize, z tohoto důvodu vhodné jejich použití vícenásobně za sebou. V Ethernetu se mohou použít maximálně dva opakovače za sebou. Opakovač je pasivní prvek v síti.

Hub (Rozbočovač)

Hub funguje obdobně jako opakovač s tím rozdílem, že opakovač má za úkol posílit signál na mediu (kabel) kdežto rozbočovat umožňuje navíc rozposlání signálu do více segmentů sítě. Například 4 portový hub přijme na jednom portu vysílání, průchodem se signál posílí a pošle se do zbylých 3 portů. Stejně jako opakovač musí šířit kolize, proto jich nemůže být libovolně mnoho za sebou a platí zde stejné, co jsme uvedli u opakovače. Vzhledem k tomu že funkce Hubu a Repeateru jsou téměř totožné a asi vás nepřekvapí, že se často obě zařízení uvádějí pospolu bez bližšího rozlišení. Ani Repeater ani Hub neumožňuje propojovat segmenty sítě s rozdílnou rychlostí.

Bridge(Most)

Most si lze představit jako zařízení, které přenáší data z jedné sítě do druhé. Na rozdíl od předchozích zařízení Bridge už se zajímá o komunikaci nikoliv pouze o signál a vytváří tunel mezi dvěma mezilehlými systémy. Komunikace se tedy přeposílá pouze určenému adresátovi nikoliv ostatním účastníkům komunikace v dané síti. Je to v podstatě jednoduchý **Switch,** který ale nerozhoduje jaká data a kam pošle, ale má předem určený kanál do kterého, bude vysílat, neumí přepínat komunikaci v rámci sítě.

Switch(Přepínač)

Z názvu je zřejmé, že toto zařízení přepíná komunikaci. Switch pracuje na linkové vrstvě, zná tedy hardwarovou adresu vysílače i příjemce zprávy a umí zařídit, že zpráva pro určitého příjemce bude poslána právě do jeho segmentu sítě. K tomu aby něco takového mohl Switch dělat, potřebuje znát své okolí. Buď se může Switch nakonfigurovat ručně, to znamená, že mu ručně nastaví, které hardwarové stanice se nacházejí na kterém segmentu sítě (portu Switche). Takové řešení je ovšem nevýhodné zvláště pro rozlehlé sítě. Častější je případ, kdy má Switch implementován nějaký algoritmus vlastního učení. Princip funkce je pak takový, že po zapnutí se Switch chová stejně jako HUB, tedy posílá všechno všem, při tom je schopen zjistit jaké adresy poslouchají na jakém portu (pokud se zapojili do komunikace).

Router(Směrovač)

si již uvědomuje topologii celé sítě, a díky tomu je pak schopen rozhodnout, kudy vede cesta do nějakého vzdáleného uzlu. Přijme-li nějaký blok dat, umí se rozhodnout, kterým směrem jej poslat dál, aby se nakonec (po případném průchodu dalšími mezilehlými uzly) dostal až ke svému konečnému adresátovi. K tomu ovšem směrovač nevystačí se stejnými informacemi, jaké má k dispozici most (bridge), ale musí se do přenášených bloků dat dívat hlouběji. V praxi to znamená, že směrovač pracuje na úrovni tzv. síťové vrstvy, která je bezprostředně nad vrstvou linkovou. Na této úrovni se přenášeným blokům dat říká obvykle pakety (packets). Ty jsou opatřeny jiným druhem adres, než jaké se používají na úrovni linkové vrstvy, a jsou vkládány do rámců linkové vrstvy (proto se směrovač musí dívat do přenášených dat "hlouběji").

Hlavním úkolem směrovače je tedy rozhodnout, kterým směrem posílat jednotlivé pakety tak, aby se dostaly až ke svým koncovým adresátům. Tomuto rozhodování (a jeho praktickému naplňování, tj. posílání paketů ve zvoleném směru) se ne náhodou říká směrování (routing). Směrovač přitom vychází ze znalosti topologie sítě a na ni pak v jednotlivých případech aplikuje algoritmus volby dalšího směru přenosu. Tomuto algoritmu se říká metoda směrování. Může mít mnoho různých podob - od neadaptivních statických algoritmů, které vychází z předem dané tabulky a nijak nereagují na jakékoli změny stavu sítě, až po takové algoritmy, které se dynamicky přizpůsobují změnám v topologii sítě, přetížení jednotlivých přenosových cest i jiným změnám.

Mezi opakovačem na jedné straně a mostem se směrovačem na straně druhé existuje zásadní rozdíl v tom, že opakovač přenáší data v reálném času, zatímco most a směrovač nikoli. Ty totiž musí nejprve načíst celý paket či rámec (nebo alespoň jeho část) do své vyrovnávací paměti, a teprve pak se podle jeho obsahu mohou rozhodnout, co s ním.

Mezi mostem a směrovačem pak existuje další zásadní rozdíl v tom, že most je pro ostatní uzly neviditelný, zatímco existenci směrovače si ostatní uzly musí plně uvědomovat. Most totiž funguje na principu odposlechu - odesilatel adresuje svůj rámec skutečnému příjemci, o kterém se domnívá, že s ním má přímé spojení (tj. že se nachází ve stejném segmentu jako on sám). Ve skutečnosti ale jde pouze o iluzi, kterou vytváří most. Ten totiž toto vysílání zachytí, rámec přijme sám a na základě v něm obsažených adres jej zase vyšle do toho segmentu, ve kterém se skutečný příjemce doopravdy nachází.
Naproti tomu směrovač, který pracuje na úrovni bezprostředně vyšší vrstvy, je pro odesilatele již plně viditelný. Pokud nějaký uzel potřebuje odeslat paket příjemci v jiném segmentu, opatří jej adresou tohoto příjemce a vloží jej do rámce linkové vrstvy, který ale pošle směrovači! Ten rámec "rozbalí", najde v něm paket síťové vrstvy a v něm adresu koncového příjemce. Podle ní se pak rozhodne, kudy paket poslat dál. Když tak učiní, paket znovu vloží do rámce linkové vrstvy a pošle jej ve zvoleném směru. Příjemcem tohoto rámce pak může být buď konečný adresát paketu (pokud se v příslušném segmentu již nachází), nebo opět další směrovač, ve kterém se celý proces opakuje. Na rozdíl od mostů jsou tedy směrovačům jednotlivé pakety explicitně adresovány.

Gateway (Brána)
Pro správné pochopení významu bran je dobré si uvědomit jednu významnou skutečnost: počítačové sítě nejsou všechny stejné. Právě naopak, často se mohou i velmi významně lišit. Na tom, jak dalece se liší, pak závisí i možnosti jejich propojení na úrovni jednotlivých vrstev. Obecně platí, že čím větší je odlišnost, tím vyšší je vrstva, na které je možné realizovat jejich vzájemné propojení.
Z tohoto pohledu pak brána slouží k propojování těch nejvíce odlišných sítí. Velmi často pracují brány až na úrovni nejvyšší, tj. aplikační vrstvy, kde zajišťují převod dat mezi jednotlivými aplikacemi. Příkladem mohou být různé druhy poštovních bran (mail gateways), které umožňují předávat elektronickou poštu z jedné sítě do druhé. Různé sítě totiž mohou používat různý formát jednotlivých zpráv (hlavně hlavičky), jiný způsob adresování, a například i jiný způsob kódování jednotlivých znaků v těle i hlavičce zprávy. Zajistit potřebnou konverzi pak dokáže až brána, pracující na aplikační úrovni, protože pouze na této úrovni může "vnímat" strukturu přenášených dat, a dokáže tak rozpoznat jednotlivé zprávy a správně interpretovat (a zkonvertovat) jejich obsah. Pro brány, pracující na nižší úrovni, by šlo jen o souvislý proud dat, jejichž význam by na této úrovni již nebyl znám.