

1-2

Q: Jaké požadavky (na funkčnost a architekturu) se uplatnily při vzniku protokolů TCP/IP?

A: Podpora propojování sítí (napojení dalších sítí na ARPANET), plus aby toto propojení bylo jednoduché, univerzální a efektivní a aby nebránilo využití žádné konkrétní technologie. Decentralizace. Robustnost (tolerance vůči chybám ostatních, "ale sám chyby nedělej".) End-to end, tj. řešit, co nejvíce funkcí až v koncových uzlech. Možnost distribuované správy. Podpora různých druhů komunikačních služeb – NEPOVEDLO SE!

Q: Proč při návrhu protokolů TCP/IP došlo k rozdělení původně navrhovaného protokolu TCP na dva protokoly: TCP a IP?

A: Jon Postel kritizuje původní TCP, že porušuje princip vrstevnatých modelů, řeší dvě různé věci současně: (1) end-to-end komunikaci mezi HOSTy (to by měl být protokol na transportní vrstvě) a (2) vkládání bloků (segmentů) do linkových rámců (to by měl být protokol na síťové vrstvě) → měli bychom zavést nový síťový protokol a TCP ponechat jako transportní end-to-end protokol. Další důvody: fungování pouze spolehlivě a spojovaně

Q: Proč neexistuje protokol IPv5 a jaké vlastnosti má ten protokol, který je tomu na vině?

A: Protože verzi 5 si v hlavičce paketů zabral Internet Stream Protocol (SP) (aby se odlišil od IPv4), který zůstal pouze experimentální. Proto vznikl až IPv6. SP měl fungovat na síťové vrstvě, paralelně k IP. Měl vycházet vstříc potřebám přenosu hlasu – packetized voice (dnes voip).

Q: Jaký je rozdíl mezi orgány IETF (Internet Engineering Task Force) a IRTF (Internet Research Task Force)? Jak jsou vnitřně strukturovány?

A: IETF řeší aktuální úkoly a problémy. IRTF řeší "výzkumné" úkoly a perspektivní záležitosti. Jsou velmi neformální – spíše otevřená platforma pro odbornou diskusi nad standardy. Přístupné komukoliv.

Q: Jaká je role organizace ICANN a jaký je její vztah k dalším orgánům (IANA, ISOC)?

A: Internet Corporation for Assigned Names and Numbers.

ICANN řeší "Internet governance". Rozhoduje koncepční otázky, spory, řeší politiku, etc .

Teoreticky reprezentuje celou internetovou komunitu. Vytvořen na žádost americké vlády.

Zastřešuje pod sebe orgán

IANA (má na starosti adresy, identifikátory a doménová jména). Zajišťuje její právní kritiku a její aktivity.

ISOC řeší spíše standardizaci a osvětu. Zastřešuje standardizační proces, zajišťuje financování.

Spolupracuje s IANA . Stará se o fungování internetu po technické stránce.

Q: Jaké jsou úrovně zralosti (maturity levels) dokumentů RFC, dříve i nyní, a do jakých větví (tracks) se řadí?

A: Tři větve podle úrovně zralosti

(1) standards track

maturity levels dnes: Proposed Standard, Internet Standard

maturity levels dříve: Proposed, Draft, Full

(2) non-standards track

maturity levels: Experimental, Informational, Historic (neměnili se)

(3) "almost standard"

maturity levels: Best current Practice (neměnili se)

Q: Co jsou dokumenty "Internet drafts", jak vznikají a kdo je vytváří?

A: Forma dokumentů vedle RFC. Určená pro pracovní dokumenty, které mají (ale nemusí mít) ambici stát se RFC. Jde o popis rozdělané práce. Dříve vznikaly na akademické půdě nebo přímo v pracovních skupinách IETF. Dnes je může tvořit kdokoliv. Vydává je IETF. Dnes vznikají

automaticky, bez posuzování a schvalování obsahu. Mají časově omezenou platnost.

Q: Jaký je postup při vydávání dokumentů RFC? Odkud návrhy prochází a kdo se k nim vyjadřuje?

A:

Od IETF:

Úspěšný pokus vydání: Internet draft submit -> IESG (dozorčí rada) <--> IETF (posouzení v komunitě) → vrací se požipomínkami, poznámkami -> RFC Production (editace) -> RFC Publisher

Neúspěšný pokus o vydání: Internet draft submit -> IESG (dozorčí rada) <--> IETF (posouzení v komunitě) → zpět, pokud jsou pochybnosti.

Od nezávislého zdroje:

Úspěšný pokus: Internet draft submit → independent stream editor → RFC Production → RFC Publisher

Neúspěšný pokus: Internet draft submit → Independent stream editor → zpět pokud jsou pochybnosti

3

Q: Co je "princip robustnosti" (Postel's Law)? Co požaduje a jak se i dnes projevuje v praxi?

A: Postel's law: „Be liberal in what you accept, be conservative in what you send.“ Princip robustnosti - říká, aby síť tolerovala chyby na vstupech, ale nedělala chyby na výstupech. Má zaručovat schopnost sítě vyrovnat se s ne zcela ideálními podmínkami – chyby, výpadky. Projevuje se v preference nespolehlivého, nespojovaného přenosu.

Q: Jak se protokoly TCP/IP staví k vrstvě síťového rozhraní a jakými výjimkami?

A: TCP/IP síťovou vrstvu nijak nepokrývá. Nic nedefinuje, předpokládá, že budou použity "takové technologie, které existují" (Ethernet). Výjimkou jsou protokoly SLIP a PPP (dvoubodové spojení, Ethernet je overkill).

Q: Jaké jsou výhody a nevýhody preference nespojovaných přenosů v TCP/IP?

A:

Výhody:

- bezestavové
- bez nutnosti reagovat na změny v přenosové infrastruktuře, rušením a novým navazováním spojení
- zajišťují adaptivní mechanismy směrování
- vhodné pro "řídce" přenosy (menší objem dat, hodně rozložený v čase)

Nevýhody:

- není vhodné pro intenzivní přenosy (větší objem dat v krátkém čase)
- různé pakety mohou být přenášeny různými cestami a být doručeny v jiném pořadí
- nevhodné pro multimediální přenosy

Spojovaný způsob přenosu může zaručit vyšší vrstva (skrze volbu TCP).

Q: Jaké jsou výhody a nevýhody preference nespolehlivých přenosů v TCP/IP?

A: Výhody:

- plynulost přenosu
- žádná režie s potvrzováním, to chtějí multimediální aplikace

Nevýhody:

- data se mohou poškodit či ztratit

Spolehlivost si mohou zajistit aplikace samy na vyšší vrstvě (skrze volbu TCP)

Q: Jaké jsou výhody a nevýhody preference principu best effort v TCP/IP?

A: Se všemi daty nakládá stejně. Při nedostatku zdrojů, "bere" všem stejně. Alternativa QoS.

Výhody:

- nevadí to "počítačovým" aplikacím (email, přenos souborů, ...)
- přenosové protokoly mohou být jednodušší
- přenosová síť může být levnější a rychlejší

Nevýhody:

- vadí to multimediálním aplikacím
- přenosové protokoly musí být složitější
- přenosová síť je dražší a pomalejší

Dodatečné přidání QoS moc nefunguje.

Q: Jaké jsou příčiny a důsledky konceptů "IP over Everything" a "Everything over IP"?

A: IP over Everything: slogan, zdůrazňuje, že protokol IP může dnes běžet nad prakticky jakoukoli linkovou technologií. IP protokol byl portován na všechny dostupné linkové technologie - IP pakety lze balit do všech linkových rámců, buněk, atd. IP pakety lze přenášet například po Ethernet, Token Ring (linkové technologie), ISDN, xDSL (pevné technologie), GSM, GPRS (mobilní technologie). Everything over IP: slogan, zdůrazňuje, že prakticky všechny aplikace dokáží fungovat nad protokolem IP. Aplikace jsou portovány nad protokol IP (ikdyž tento protokol nemusely předpokládat a nemusí pro ně být příliš vhodný). Nad IP lze portovat například přenos hlasu (VOIP), přenos obrazu (IPTV).

Q: Jaké jsou rozdíly mezi hostitelskými počítači a směrovači v IPv4, co je multihomed host a jaká doporučení se ho týkají?

A: Hostitelské počítače: připojený pouze k jedné síti, slouží k potřebám uživatelé, hostí zdroje. Směrovače: připojeny do dvou či více sítí, slouží pouze k potřebám směrování - přestup z jedné sítě do druhé. Multihomed host: obecně se nedoporučuje. Uzel, který by současně fungoval jako směrovač i jako hostitelský počítač. Dnes se připouští připojení hostitelských počítačů do více sítí kvůli rozkladu zátěže či zálohování.

Q: Jaká je celková koncepce síťové vrstvy v TCP/IP?

A: Využívá se modelu "pokličky". Zakrýt všechna specifika nižších vrstev, vytvořit jednotné prostředí pro všechny vyšší vrstvy.

Výhodou je, že se vyšší vrstvy nemusí přizpůsobovat.

Nevýhodou je, že nelezeme využít výhody specifické technologie.

Ale nemá smysl zakrývat velikost linkového rámce --> existuje výjimka – vyšší vrstvy vidí parametr MTU (Maximum Transmission Unit). Udává kolik bajtů se vejde do "nákladové" části linkového rámce. Podle toho vyšší vrstvy porcují data na menší části.

Q: Co všechno "patří" do síťové vrstvy (protokoly a další koncepty)?

A:

Používá jednotné adresování – abstraktní adresy IPv4(32 bit) – síťová a relativní část.

Koncepty subnettingu a supernettingu, privátní IP adresy

Pomocní síťové protokoly:

ICMP: sdělování nestandardních situací. Utility ping a Traceroute.

(R)ARP: převod IP adres a linkových adres (oba směry).

NAT: překlad IP adres mezi veřejnými a privátními.

RIP, OSPF: směrovací protokoly

IPSec: zabezpečení

Mobile IP: podpora mobility

Q: Jaká je celková koncepce transportní vrstvy v TCP/UP a jak se vyvíjela?

A: Implementována až v koncových uzlech. Zajišťuje end-to-end komunikaci (mezi koncovými

uzly).

Transportní vrstva je první a zároveň poslední možnost kde lze změnit způsob fungování přenosových služeb.

Dvě možnosti (1) neměnit nic - UDP transportní protokol (nespojovaný, nespolehlivý, stejné jako IP protokol),

(2) změnit vše - TCP transportní protokol (spojoovaný a spolehlivý, složitý a komplexní).

UDP: řeší jen práci s porty, datagramy se musí vejít do IP datagramu

TCP: spolehlivost (potvrzování) a spojoovanost, práce s porty, řízení toku a zahlcení příjemce, přenáší segmenty podle MTU

Vývoj:

- Původně dvě extrémní možnosti (neměnit nic vs. měnit vše)

- Dnes snaha zavést jemnější škálu než 2 extrémní protokoly

- SCTP: spolehlivě (princip stejný jako TCP), spojoované (princip se liší od TCP) - data přenáší po blocích (podobně jako UDP). Podporuje více proudů současně.

Podporuje multihoming. Předchází zahlcení.

- DCCP: přenáší datagramy (UDP), spojoované (TCP), nespolehlivě (UDP), předchází zahlcení (TCP), multihoming (SCTP), neřídí tok (UDP).

- SCPS-TP: do budoucna. Space Communications Protocol Standard Transport Protocol. Pro meziplanetární komunikaci, kde je extrémně velké přenosové zpoždění.

Q: Jak se vyvíjel repertoár aplikací v TCP/IP?

A: Původně malý rozsah: SMTP, FTP, TELNET, rlogin.

Později: síťové noviny (news, netnews, USENET), sdílení souborů (NFS), Gopher, WWW (HTML, HTTP, ...), on-line komunikace (chat, IRC, ICQ, messengery, ...).

Dochází k platformizaci - přežívají jen některé aplikace, ostatní ztrácí svou identitu a stávají se "nadstavbami" těch co zůstaly.

Pro všechny tyto „počítačové“ aplikace je způsob fungování TCP/IP (hlavně princip best effort, bez podpory QoS) stále ještě akceptovatelný - byť ne ideální.

Q: Co je podstatou principu OTT (Over the Top) a jaké výhody a nevýhody přináší?

A: Aplikace je provozována přes veřejný internet (v souběhu s dalším provozem): nemá granatovanou kapacitu a podmínky (Hulu, Skype, Youtube, Netflix). Opak k privátnímu poskytování multimediálních služeb, kde je pro ně vyhrazena kapacita – dokáží fungovat garantovaně (UPC Telefon, O2 TV).

4-1

Q: Jaké jsou principiální možnosti převodu mezi HW (linkovými) a síťovými adresami v TCP/IP?

A: přes protokol ARP (Address Resolution Protocol): IP -> HW. Kdo zna IP adresu rozešle broadcastem dotaz "kdo má tuto adresu?", ten kdo ji používá odpoví.

Opačný převod je přidělování IP adresy na základě HW adresy, přes protokoly: RARP, BootP, DHCP.

Bez broadcastu: ATARP - přes tabulku (vhodný server spravuje převodní tabulku) nebo převod přímým výpočtem .

Q: Jaká je vnitřní struktura ethernetové adresy, kolik bitů mají, a jak se jednotlivé varianty označují?

A: MAC-48 a EUI-48: 3 byty OUI (Organizationally Unique Identifier) + 3 byty sériové číslo.

EUI-64: 3 byty OUI + 5 bytů sériové číslo .

Q: Jaká jsou obecná pravidla adresování na úrovni síťové vrstvy v TCP/IP? Kdy musí mít uzly stejné Network ID a kdy odlišné? Jaké volit jejich Host ID?

A: Adresují se celé uzly (představa více vzájemně propojených sítí), síťové adresy musí vyjadřovat (1) příslušnost ke konkrétní síti (network ID) a (2) relativní adresu uzlu v rámci dané sítě (host ID).
Stejná síť -> stejné network IDs, různé host IDs
Odlišné sítě -> různé network IDs, "libovolné" host IDs
Koncové uzly mají 1 IP adresu.
Směrovače mají více síťových rozhraní (a pro každé 1 IP adresu).

Q: Jak a čím je definována velikost bloku IP adres, které je nutné přidělit najednou, a proč obecně nelze vrátit nevyužitou část bloku a přidělit ji někomu jinému?

A: Počtem IP adres se stejnou síťovou částí. Vytržením by vznikly dvě sítě obsahující adresu se stejnou síťovou částí. Problém, protože se směruje jen podle síťové části, teprve uvnitř sítě berou v úvahu relativní část IP adresy. Mátlo by to celkově směrovací algoritmus TCP/IP.

Q: Jaký je princip adresování na transportní vrstvě?

A: Rozlišují se entity v rámci uzlu. Stačí relativní adresy - porty - 16b. Porty jsou abstraktní (logické) adresy a "fyzické" entity se asociují s konkrétními porty. Existuje konvence o "dobře známých portech" (IANA) – porty 0 až 1023 mají pevně daný význam.

Q: Co jsou URI/URL schémata? Popište podrobněji jedno konkrétní schéma.

A: URI=URLC → slouží k adresování různých typů objektů (obrázky, videa, ...), které se mohou nacházet na různých místech v síti – obvykle ve formě souboru. Součástí adresy je i umístění objektu. Obecný tvar je "<schéma>:<specifická část>".

`http://<user>:<password>@<host>:<port>/<url-path>?<query>#<bookmark>`

`http://jmeno:heslo@blabla.cz:8080/index.html?q=něco#sem`

4-2

Q: Popište původní koncept tříd IPv4 adres. Proč byl zaveden a jaké měl výhody a nevýhody?

A: Nemají přímou vazbu na HW adresy -> nutný překlad. 32b, "rozděleno" do bloků, po kterých se IP adresy přidělují. Celkem tři předěly:

A - velké sítě: nejvyšší bit 0, 7b síť, 3B uzly

B - střední sítě: nejvyšší bity 10, 14b síť, 2B uzlů

C - malé sítě: nejvyšší bity 110, 21b síť, 1B uzlů

D - multicast: nejvyšší bity 1110, 28b - 1 složka

E – budoucí rozšíření

To vede na velké plýtvání

Q: Jaké jsou IPv4 adresy se speciálním významem (se samými 0 a 1 v některé složce)?

A:

Samé 0 – this

Samé 1 - all

127.x.x.x loopback

0.0.0.0 já

255.255.255.255 broadcast, neprojde přes směrovač síť 0, uzel => konkrétní uzel v této síti
síť, uzel 0 => tato síť

síť, uzel 1 => cílený broadcast do všech uzlů té sítě

0 síť, uzel x – konkrétní uzel v této síti.

Q: Jaké jsou IPv4 adresy pro multicast? Podle čeho je lze dělit? Jaká je konkrétní hodnota adres "all hosts" a

"all routers"?

A: Prvních 256 adres (224.0.0.x) je vyhrazeno pro „dobře známé“ skupiny:

224.0.0.1 („all hosts“, neboli: všechny uzly v dané síti)
224.0.0.2 („all routers“, neboli: všechny směrovače v dané síti)
posledních 2^{24} adres (239.x.x.x) je určeno pro lokální multicastové skupiny
ostatní adresy (224.0.1.0 až 238.255.255.255) jsou určeny pro globální multicastové skupiny:
dobře známé: 224.0.0.x
globální: od 224.0.1.0 do 1110 1110.255.255.255 - uzly z různých sítí
lokální multicast: 1110 1111.x.x.x - ve stejné síti
Glob. Multicast – i pro uzly z různých sítí
Lok Multicast – jen pro uzly ze stejné sítě

Q: Jak se vyvíjel způsob přidělování IP adres v době používání tříd IP adres? V čem byl problém a jaká (dočasná i trvalá) opatření byla navržena pro jeho řešení?

A: Prvotní způsob přidělování:

- zájemce s potřebou X adres dostal „nejbližší vyšší blok“ (velké množství adres často zůstalo nevyužito)
- když potřeboval 1000 adres, dostal 1x B-čko (tj. 65536 IPv4 adres)

Přešlo se na princip "více nejbližší menších bloků"

- při potřebě 1000 adres dostal zájemce 4 nebo 8 C-ček (4-8x 256 IPv4 adres)

Hrozba rychlého vyčerpání IPv4 adres -> opatření:

- dočasná: subnetting (rozdělit blok IPv4 adres na menší části pro více sítí), CIDR (zrušení tříd, přidělovat libovolné bloky), privátní IPv4 adresy
- trvalá: hledal se protokol IPng (IP next generation) -> IPv6

Q: Jak funguje subnetting? Jaké má výhody a jaká má omezení?

A: Jde o podrozdělení sítě, za účelem rozdělení bloku IP adres. Problém je, že ostatní neví, jak to udělal. Rozdělení lze udělat jen v soustavě (pod)sítí, které má pouze 1 vstupní bod (-> rozdělení nesmí to být patrné z pohledu ostatní sítí).

Používá síťové masky – popírá rozdělení adres do tříd daný prvními bity.

Posouvá předěl mezi síťovou a relativní částí IPv4 doprava.

Práci s maskami musí podporovat všechny síťové prvky i SW.

Q: Jak funguje supernetting? Jaké má výhody a přínosy?

A: Opak subnettingu. Jak z několika menších bloků IP adres udělat jeden větší blok. Jde o posun předělu směrem doleva, k vyšším bitům. Podmínka: musí být vyčerpány všechny možné kombinace v těch bitech, přes které se předěl posouvá. Supernetting umožňuje řešit problém nárůstu směrovacích tabulek.

Q: Jak funguje a co přináší mechanismus CIDR? Mění nějak závislost IP adres na poskytovateli připojení?

A: Využívá subnetting i supernetting. Ruší koncept tříd u IPv4 adres. Umožňuje přidělovat libovolně velké bloky IPv4 adres: tzv. CIDR bloky. Libovolný předěl síť/relativní (cidr prefix – číslo, které udává počet bitů síťové části)

- hierarchie přidělovatelů šetří adresy
- lepší struktura směrovacích tabulek, jen poskytovatel musí vědět adresu.

Q: Popište způsob přidělování IPv4 adres na úrovni IANA, RIR, NIR a LIR, včetně dostupnosti/vyčerpání na úrovni IANA a RIPE.

A:

IANA → RegionalIR → NationalIR → LocalIR(ISP) (IR=Internet Registry).

Nebo: IANA → RegionalIR → LocalIR(ISP) (IR=Internet Registry).

Vyčerpáno: IANA-2/2011,

Ripe-9/2012.

Q: Popište koncept privátní IPv4 adres. Které rozsahy adresy jsou vyhrazeny pro roli privátních a proč?

A: Dočasné řešení proti úbytku IP adres. Stejně IP adresy lze využít opakovaně, ale jen v takových sítích, které „nejsou zvenku vidět“. Podmínka: privátní síť musí být „schována“ za něčím, co brání šíření informací o dostupnosti privátních adres -> překlad adres (NAT) nebo firewall (proxy brána).

Vyhrazeny:

1 síťová adresa třídy A (10.x.x.x), resp. CIDR blok 10/8

16 síťových adres třídy B (172.16.x.x až 172.31.x.x), resp. CIDR blok 172.16/12

256 síť. adres třídy C (192.168.0.x až 192.168.255.x), resp. CIDR blok 192.168/16

Proč?

Směrovač (proxy brána) nesmí propustit „ven“ informaci o privátní IP adrese. Ale co když dojde k nějaké chybě a informace se (omylem) dostane ven? --> pokud byla použita vyhrazená privátní IPv4 adresa, pak nejbližší další směrovač či proxy brána chybu napraví (a informaci zastaví / nepustí dále).

Q*: Jak funguje mechanismus NAT, k čemu slouží a jaké má výhody a nevýhody? Jaké jsou varianty NATu?

A: Umožňuje používat privátní IP adresy. Překlad vnějších a vnitřních adres na síťové vrstvě. Vznikl kvůli potřebě šetřit IPv4 adresami + slouží i jako jakási forma firewallu (brání nevyžádanému přístupu k uzlům „za NAT-em“).

Problémy: obecně s dostupností uzlů „za NAT-em“, zvyšuje složitost a spotřebu zdrojů, některé aplikace s NAT-em nefungují vůbec (ty, co pracují přímo s IP adresami).

Princip fungování: v hlavičkách datagramů přepisuje vnitřní IP adresy na vnější (u odesílatele resp. příjemce).

-Musí přepočítat is pseudohlavičky UDP datagramů a TCP segmentů (jsou počítány z IP adres odesílatele a příjemce).

-Varianty: dynamický NAT (převodní tabulka dle potřeby) a statický NAT (převodní tabulka sestavena dopředu)

Q*: Jaký problém (a jak) řeší varianty Full Cone, Restricted Cone a Symmetric NAT?

A: Řeší problém, když máme jen 1 vnější adresu a více vnitřních adres sítě. Mezi vnitřní a vnější IP vzniká mapping, což „otevřít bránu“ pro komunikaci.

Stejný = stejná vnitřní IP adresa a port se překládají na stejnou vnější IP adresu a port (ale ne identickou).

- Full Cone – Stejný. Přenášet data směrem donivtř může kterýkoliv vnější uzel z kteréhokoliv portu.

- Restricted Cone - Stejný. Odpovídat mohou jen oslovené vnější uzly, ale z libovolného portu. Pro ostatní vnější uzly je komunikace neprůchozí.

- Port Restricted Cone - Stejný. Odpovídat mohou jen oslovené uzly z oslovených portů.

-Symmetric NAT - stejná vnitřní IP adresa a port se překládají na stejnou vnější IP adresu ale pokaždé jiný port. Odpovídat mohou jen oslovené vnější uzly a jen z oslovených portů. A jen na tu kombinaci IP:port, která pro ně byla otevřená.

Q*: K dispozici máte CIDR blok 192.168.1.0/24. Navrhněte způsob jeho využití (pomocí subnettingu) pro 3 IP sítě, které mají po řadě X, Y a Z uzlů (v konkrétním testu budou uvedeny konkrétní hodnoty). Jaké další předpoklady musí být splněny?

A:

Q: Jaký je význam položek IHL, ToS, TTL a Protocol v hlavičce IPv4 datagramu?

A: IHL: Internet Header Length - udává velikost hlavičky v jednotkách 32-bitů. ToS: Type of Service – původní význam dnes již není znám, využívá QoS, dnes ignorováno. TTL: Time To Live - původně časový údaj, dnes počet přeskoků.

Q: Jak je u IPv4 řešena fragmentace a jaké položky pro ni vyhrazeny v hlavičce IPv4 datagramu ?

A: Jde o překlad původního datagramu. Každý fragment má v položce identification stejnou hodnotu jako původní datagram. Položka fragmentation offset udává offset začátku datové části fragmentu oproti datové části původního datagramu. Položka "Flags" značí, jestli má být datagram fragmentován nebo zda jde o poslední fragment. Fragmentovat může kdokoli, defragmentovat můžeme jen na konci.

Q: O čem vypovídá parametr MTU, jaká je jeho minimální hodnota, k čemu slouží a jak funguje postup MTU Path discovery?

A: Maximum Transmission Unit. Udává velikost nákladové části rámce. Minimum je 68B, v praxi se používá 576B.

MTU Path discovery:

1. X nastaví na velikost "místního" MTU
2. uzel A připraví IP datagram velikosti X, nastaví mu příznak Don't Fragment a odešle jej uzlu B
3. pokud A dostane zpět ICMP zprávu Destination Unreachable (Type 3), s podtypem (Code=4, tj. Fragmentation Needed), sníží X a jde zpět na bod 2
4. Path MTU má hodnotu X

Q: K čemu slouží protokol ICMPv4 a jak se přenáší jeho zprávy? Co je na tomto způsobu přenosu nestandardního?

A: Signalizace chyb a nestandardních situací. Vkládán do IP datagramu (jako by byl z transportní vrstvy, která není ve směrovačích), porušuje tím vrstevnatý model, správně by měl být ICMP protokol vedle IP.

Q: Jaký je formát ICMPv4 zprávy Destination Unreachable a co konkrétně signalizuje (v rozlišení dle položky Code)?

A: Formát: Type (0.-7. bit), Code (8.-15. bit), Checksum (16. - 30. bit), další 4B nevyužity, 20+8B hlavička IP datagramu + prvních 8B payloadu.

Informuje o tom, že uzel nemohl pokračovat v požadovaném zpracování IP datagramu a musel ho zahodit. Typ 3 (Type = 3). Různé důvody, proč se datagram zahodil -> rozlišuje se dle hodnoty v Code:

Code=0: Network Unreachable

Code=1: Host Unreachable

Code=2: Protocol Unreachable

Code=3: Port Unreachable

Code=4: Fragmentation Needed and DF Set (= potřeba fragmentace, ale je nastaveno, ať se nefragmentuje)

Q: Jaký je formát ICMPv4 zprávy Time Exceeded a jak se využívá v utilitě Traceroute, jaké jsou varianty Traceroute?

A: Formát: Type (0.-7. bit), Code (8.-15. bit), Checksum (16. - 30. bit), další 4B nevyužity, 20+8B hlavička IP datagramu + prvních 8B payloadu.

Type obsahuje hodnotu 11, Code je buď 0 nebo 1. Zpráva je genrována ve dvou situacích:

1. Když dojde k vynulování TTL -> Code=0

Význam: zacyklení

2. Když při sestavování fragmentů vyprší časový limit a fragmenty nejsou všechny -> Code=1

Význam: nelze sestavit celý původní datagram

Využití v Traceroute:

Pošle se blok dat na adresu uzlu B, TTL se nastaví 1

Nejbližší router nastaví TTL na 0, datagram zahodí a vrátí zpět ICMP Time Exceeded

A se dozví adresu prvního routeru po cestě

Posílají se další bloky dat, TTL se zvyšuje o 1, vrací se Time Exceeded

Odhalí se cesta z A do B

Varianty Traceroute:

původní: ICMP echorequest;

Van Jacobson: UDP datagram a vysoký port

Q: K čemu slouží a jak funguje ICMPv4 zpráva Source Quench? Jak se využívá v praxi?

A: Důvodem zahození datagramu může být i přehlcení --> příjemce může generovat ICMP zprávu Source Quench (Type=4, Code=1). Pošle se tomu, o kom si myslí, že způsobuje zhlcení. Jde o "jednostranný výkřik" ve smyslu : zpomal. Neříká jak moc zpomalit a reakce na tuto zprávu není definovaná. Neexistuje zpráva o konci zhlcení. Dnes se použití nedoporučuje, zhlcení se řeší jinak.

Q: Jaký je formát ICMPv4 zpráv Echo Request a Echo Reply, jakým způsobem se využívají v rámci utility Ping?

A: Slouží k testování dostupnosti síťových uzlů.

Echo Request: Výzva protistraně. Type=8.

Echo Reply: Reakce protistrany na výzvu. Type=0.

Formát: Type (0.-7. bit), Code (8.-15. bit)=0, Checksum (16. - 30. bit), další 4B identifier (páruje výzvy a reakce) + sequence number (pořadové číslo), zbytek vycpávka, kvůli zvětšení objemu zprávy.

Využití v rámci utility Ping:

Testuje dostupnost a reakce síťových uzlů

Uzel, kde se Ping spustí posílá série zpráv Echo cílovému uzlu. Cílový uzel odpovídá Echo Reply

Vyhodnocuje se ztrátovost odpovědi a jejich zpoždění (min, avg, max)

Q: K čemu slouží a jak funguje protokol ARP? Jaký je formát jeho zprávy a jak se přenáší?

A: Převod IP adres na linkové adresy. Broadcast po linkové vrstvě, ozve se ten čí je IP. Pracuje jen v dané síti, nepřekračuje hranice – kvůli tomu by měl patřit do linkové vrstvy. Vkládá ale ARP zprávy do linkových rámců, tedy patří do síťové vrstvy.

1) uzel A zná IP adresu uzlu B, a potřebuje znát jeho HW adresu

2) sestaví ARP zprávu, ve které uvede IP adresu uzlu B a svou IP a HW adresu

3) tuto ARP zprávu vloží do linkového rámce a pomocí (linkového) broadcastu rozešle jako dotaz po celé síti, ve které se nachází

4) uzel B rozpozná svou IP adresu a odpoví sestaví ARP zprávu s odpovědí, ve které uvede svou HW adresu, pošle unicastem

Formát*: typ HW adresy, typ protokol adresy, délka HW adresy a délka IP adresy, operace(dotaz/odpověď), kdo se ptá (HW i IP adresa), cílová (HW-vyplní na dotaz, IP)

Q: Jaká je funkce ARP cache a jak probíhá zpracování ARP dotazu a odpovědi na různých uzlech v dané síti?

A: Protokol ARP je drahý --> co nejvíce využívat ARP cache: vyrovnávací paměť, ve které si ARP pamatuje výsledky převodů IP->HW.

Položky v tabulce mohou být statické nebo dynamicke (častější). Dynamicke se musí pravidelně zapomínat a reflektovat změny v síti a musí být osvěžovány.

Situace: uzel A zná IP adresu uzlu B, a potřebuje znát jeho HW adresu

Postup:

uzel A se podívá do své ARP cache

pokud zde najde HW adresu k IP adrese uzlu B, končí

uzel A sestaví a rozešle (linkovým broadcastem) ARP zprávu s dotazem

vyplní v ní svou IP a HW adresu, a IP adresu uzlu B

každý uzel v síti zachytí ARP zprávu (vysílanou broadcastem), a:

vyjme ze zprávy vazbu (binding) mezi IP a HW adresu uzlu A a pokud ji už má ve své ARP cache, tak ji osvěží (prodlouží její platnost)

zjistí, zda je uzlem B (zda IP adresa uzlu B je jeho IP adresou) • pokud ne, ARP zprávu zahodí a končí

uzel B:

si do své ARP cache zanesou vazbu (binding) mezi IP a HW adresou uzlu A

nebo ji osvěží, pokud ji ve své ARP cache již měl

sestaví ARP zprávu s odpovědí a odešle ji cíleně (unicast-em) uzlu A

v dotazu přehodí Sender/Target, příznak Dotaz/Odpověď a vyplní svou HW adresu

Q: Co je smyslem Proxy ARP, k čemu se využívá a jak funguje?

A: Řešit přímou nedosažitelnost uzlu. Cíl je například mobilním agentem, za směrovačem, co spojuje dvě části sítě, za extrémní latencí, atd.

Vše vyřizuje jiný uzel jako prostředník (ARP odpovědi, přijímá zprávy a přeposílá od cíle), udá svou HW adresu.

Q: K čemu slouží, jak fungují a v čem se liší protokoly RARP a BOOTP?

A: Přidělení IP na základě HW adresy.

RARP - uzel zná svou HW adresu, a chce znát svou IP adresu. Opačně vyplněný ARP, linkový broadcast, uzel, který funguje jako RARP server se ozve.

BOOTP - dotaz na broadcastovou adresu: server odpoví unicastem, broadcastem, nebo na IP (jiné dotazy), pokud je server v jiné síti, vyřizují mu to směrovače

Rozdíly: síťová (RARP) x aplikační (BOOTP), linkový x IP broadcast, server jen ve stejné síti x i jiné, jen IP x i další údaje

Q: Jak funguje protokol DHCP a jaké možnosti přidělování IP adres nabízí? Jaké další údaje může poskytovat svým klientům?

A: DHCP přiděluje IP adresy

Ručně - správce sítě

Trvale - IP adresu určí DHCP server sám a přidělí ji (výjimečně – lepší už je ručně)

Dočasně - IP adresu určí DHCP server sám, ale přidělí ji pouze na omezenou dobu (nejčastější)

Chování DHCP klienta:

Alokace - klient ještě nemá adresu a žádá o pronájem

Realokace - klient již má adresu a snaží se tento pronájem obnovit (ještě trvá doba pronájmu)

Obnova - před koncem pronájmu se klient snaží o jeho prodloužení

Rebinding - snaží se získat pronájem stejné adresy od jiného DHCP serveru

Uvolnění - klient vrací pronajatou adresu ještě před koncem jejího pronájmu

Další údaje, které DHCP server může poskytovat:

Masku sítě

Místní časové pásmo

Seznam směrovačů vedoucí ven ze sítě

DNS jméno uzlu a doménu

Jméno a velikost souboru s boot image

....

Q: Jak je řešen multicast v IPv4? K čemu slouží protokol IGMPv4?

A: Vyžaduje adresaci (skupinové adresy), správu multicast skupin, přenos datagramů (přenos všem členům skupiny).

Přenos – musí mít informace o složení skupin, umístění uzlů skupin, buduje distribuční stromy, duplikuje datagramy.

Každý hostitelský počítač musí vědět, do kterých multicast skupin je zařazen.

IGMPv4 řeší správu multicastových skupin (vytváření a rušení skupin, přidávání a odebrání uzlů, atd) . Zprávy IGMP se vkládají do IP datagramů.

Q: Co je specifické na provozování IP protokolu na dvoubodových spojkách? Co zajišťují a jak fungují protokoly SLIP a PPP?

A: Dvoubodové spojení je jednoduché - nevznikají kolize, není potřeba adresace. Jediné, co je třeba zajistit je framing. Řešení v rámci TCP/IP je "udělat to ještě jednodušejí a levněji než je Ethernet", pomocí vlastních protokolů SLIP (Serial Line IP) a PPP (Point-to-Point Protocol). Oba protokoly jsou linkové - přenáší linkové rámce, do kterých se vkládají IP datagramy.

-SLIP

Znakově orientovaný

Asynchronní přenos po sériových linkách

Rozděluje proud do 8b znaků na jednotlivé rámce, začátek a konec vymezuje znakem END

Zajišťuje transparentci dat

Pokud se v těle rámce vyskytne END, nahradí ho dvojicí ESC a 0xDC

Pokud se v těle rámce vyskytne ESC, nahradí ho dvojicí ESC a 0xDD

-PPP

Bohatší než SLIP

Znakově orientovaný

LineControlProtocol - dokáže navazovat spojení na linkové vrstvě, ukončovat spojení, dohodnout s protistranou parametry

Podporuje autentizaci

Podporuje vkládání síťových paketů různých druhů

6

Q: Jaké jsou základní principy směrování v IP sítích? Co je princip katenetu, hop-by-hop routing, least cost routing a destination-based routing? Naznačte možné alternativy.

A: Principy směrování v IP sítích: bezstavové (další směr nezávislý na historii předchozích datagramů) a nezávislé na obsahu a zdroji

Princip katenetu: svět je tvořen soustavou sítí vzájemně propojených pomocí směrovačů

Hop-by-hop routing: v každém směrovači se rozhoduje znovu o "dalším hopu"

Least-cost routing: optimální cesta se volí podle nejnižší "ceny" (cena = používaná metrika). Nelze využít více cest se stejnou cenou.

Destination-based routing: směruje se jen na základě cílové adresy (zdrojová adresa nehraje roli), pouze v cílové síti se bere v úvahu také relativní částí IP adresy

Alternativy:

K bezstavové: koncept toků (pakety/datagramy patří k sobě), tag switching (odbočka toků),

K nezávislosti na obsahu a zdroji: content switching (v úvahu se bere charakter dat), source-based routing (v úvahu se bere odkud data pochází), policy-based routing (celá řada faktorů)

Q: Jaký je rozdíl mezi přímým a nepřímým doručováním? V čem s neliší adaptivní a neadaptivní směrování, jaké mají výhody a nevýhody?

A: Přímé: cílový uzel se nachází ve stejné síti, ke směrování nedochází. Nepřímé: cílový uzel není ve stejné síti, směrovač určí kam dál.

-Adaptivní: snaží se reagovat na změny. Vyžaduje aktualizace informací o stavu celé soustavy sítí.

Vyžaduje průběžné hledání nejkratších cest. Požaduje protokoly jako RIP, OSPF, BGP (dynamicky aktualizují obsah směrovacích tabulek). Nevýhodou je vysoká režie, zejména aktualizací.

-Neadaptivní: nereaguje na změny v soustavě propojených sítí. Není potřeba aktualizace informací ani spolupráci s ostatními uzly. Obsah směrovacích tabulek je pevně dán (statický). Výhodou je, že vyhovuje zvýšeným požadavkům na bezpečnost (nelze napadnou skrze šíření aktualizací), není režie na aktualizace, lze vyhovět speciální požadavkům na aktualizace. Nevýhodou je, že nereaguje na případné změny.

Q: Jaký je rozdíl mezi směrovací a forwardovacími tabulkami, kdo a jak zajišťuje jejich naplnění a aktualizaci?

A: Směrovací tabulky slouží k rozhodování (logické činnosti – hledání nejkratších cest a výměnu směrovacích informací), ale ne k manipulaci s IP datagramy. Pracují s nimi RIP a OSPF. Obsahují další info jako síťovou masku, next hop IP, odchozí rozhraní, ohodnocení (v metrice).

Forwardovací tabulky jsou rychlé, menší a již se u nich "nepřemýšlí" – používají se k manipulaci s IP datagramy. Forwardovací tabulka je výcucem ze směrovací - obsahuje jen ty cesty, které již byly vybrány jako optimální.

Q: Popište možnosti zmenšování počtu položek směrovacích (i forwardovacích) tabulek?

A: Lze snižovat agregací CIDR bloků - supernetting. Další možnost je zavedení implicitní cesty - explicitně se vyjmenuje jen to, co „jde jinudy“, vše ostatně se směruje implicitně přes default router (prefix 0; typicky u koncových sítí vede do Internetu, explicitně se popíší jen nižší sítě).

Q: Jak se liší role hostitelských počítačů a směrovačů při směrování? Jak funguje a k čemu slouží ICMPv4 zpráva Redirect? Jak by na ni měl reagovat hostitelský počítač?

A:

-Směrovače: zajišťují všechny činnosti, spojené se směrováním (hledání cest, aktualizace informací, atd) .

-Hostitelské počítače: neúčastní se hledání cest ani aktualizací. Pouze se chovají tak, jak jim někdo řekne. Mají forward tables a používají je, ale samy si je neaktualizují. Pokud host posílá data po špatné cestě, nejbližší směrovač ho upozorní a řekne, kudy vede lepší cesta (pošle zpět ICMP Redirect) a host si to může zapamatovat v jeho tabulce.

-ICMPv4 Redirect: odesílá ji směrovač (směrovač také zaručuje, že data, která toto způsobila správně doručí), jde o informační zprávu. Příjemce je hostitelský počítač, měl by na zprávu reagovat tím, že se poučí (zaneše si informace do směrovací tabulky) – ale nemusí např. kvůli bezpečnosti.

Q: Jak fungují ICMPv4 zprávy Router Solicitation a Router Advertisement? Jaký mají formát?

A:

-Router Solicitation: výzvza, kterou posílá hostitelský počítač ve smyslu "je zde v síti nějaký směrovač? Ozvěte se ...". Pokud takový směrovač existuje, odpoví zprávou ICMP Advertisement (může jich odpovědět více). Vysílá se: (1) když je k dispozici multicast na adresu 224.0.0.4, (2) pokud není k dispozici multicast na místní broadcast adresu (255.255.255.255). Formát: Type 0.-8.b (type=10), Code 9.-16.b (code=0), Checksum 17.-31. b + další 4B vyhrazeny (celkem 8B).

-Router Advertisement: posílá směrovač jako odpověď na Router Solicitation nebo z vlastní iniciativy (upozorňuje na svou existenci). Vysílá se: (1) když je k dispozici multicast na adresu 224.0.0.1, (2) když není k dispozici multicast pomocí broadcastu. Formát:

Type byte, Code byte, Checksum 2byty, počet směrovačů byte (Address entry size), velikost adresy byte, lifetime 2byty, poté IPv4 adresa směrovače 4 byty a preference 4 byty pro tolik směrovačů, kolik je uvedeno v počtu směrovačů.

Q: Jakými způsoby se můžete hostitelský počítač dozvědět o existenci směrovačů v jeho síti? Popište podrobněji možnosti, které dává protokol ICMPv4.

A: Nastavené, od DHCP, sám je může zjistit pomocí router solicitation, dostane info od směrovačů pomocí router advertisement, ICMP redirect .

Router solicitation – výzva k tomu, aby se mu ozvali směrovače, pokud jsou v síti. Vysílá se na multicast či broadcast.

Router advertisement – zprávu vysílá směrovač, jako odpověď na solicitation nebo z vlastní iniciativy. Na multicast nebo broadcast.

Redirect při zjištění, že host používá neoptimální cestu ho upozorní na jiný, lepší směrovač v síti.

Q: Popište, jak je implementován a jak funguje protokol RIP. Čím a jak je jeho fungování omezeno?

A: Aplikační protokol pro tvorbu směrovacích tabulek. Funguje na principu distance vector. Každý směrovač rozesílá svou směrovací tabulku svým přímým sousedům každých 30 sekund na port 520 pomocí UDP. Krok 0: jen sousedé, v každém kroku se vymění od sousedů tabulky, spočtení lepších cest. Nejvýše 15 přeskoků, 16 je nekonečno.

Q: Popište, jak se vyvíjelo směrování v rámci celosvětového Internetu.

A:

-Na začátku: Internet byl jenou jedinou soustavou vzájemně propojených sítí. Každý směrovač měl úplnou informaci o topologii celého Internetu. To se stalo informačně neúnosné.

-Později: Internet byl rozdělen na páteř (core) a ostatní (non-core). Směrovače (core gateways) v páteři měly úplné směrovací informace. Směrovače mimo páteř (non-core gateways) měly podrobné směrovací informace jen o své oblasti - znaly cestu do svých podsítí, vše ostatní pomocí implicitní cesty. GGP (gateway to gateway protocol), EGP (Exterior Gateway Protocol).

-Nyní: Čem předchází řešení znovu neúnosné, přešlo se na systém směrovacích domén. V rámci domény jsou šířeny detailní směrovací informace – domény lze volit dostatečně malé. Mezi doménami se šíří pouze informace o dostupnosti. IGP protokoly (RIP OSPF).

Q: Popište smysl a účel směrovacích domén, princip hierarchického směrování, autonomních systémů a směrovacích politik.

A: Nahrazuje systém páteřních a nepáteřních směrovačů – tento systém nebyl dostatečně škálovatelný. Detailní informace jen lokálně (směrovací doména), jen informace o dostupnosti (interval - často jen prefix). Směrování jen podle dostupnosti, ne ceny, path vector (cesta). Směrovací doména=AS (autonomní systém) (číslo od IANA) - mohou si rozhodovat o detailním směrování uvnitř sebe sama. IGP uvnitř (RIP, OSPF)

- směrovací politika provozovatele: co za IGP uvnitř, provoz od ostatních AS, co jim inzerovat o konkrétních sítích => k tomu EGP(BorderGatewayProtocol) - definují vazby mezi AS.

Q: V jakých režimech může fungovat protokol OSPF, s jakými databázemi pracuje a jaké zprávy používá?

A:

Basic Topology: celý AS je homogenní (jedna LSDB pokrývá celý AS), každý směrovač zná celý AS, všechny směrovače jsou si rovny.

Hierarchical Topology: Celý AS rozdělen na oblasti (Areas). Každá oblast má svou LSDB. Jedna oblast je páteřní, ostatní ne-páteřní. Role směrovačů se mohou lišit na:

boundary router – propojuje vnější svět

páteřní směrovač – uvnitř páteřní oblasti

area border router – propojuje oblasti mezi sebou

vnitřní směrovač – uvnitř oblasti, pracuje jen s 1 LSDB

Pracuje s LSDB – link-state database – každý směrovač má úplnou informaci o topologii celé soustavy propojených sítí, ve které se nachází.

Používá 5 druhů zpráv, vkládá je do IP datagramů: Hello, Database description, Link State Request, Link State Update, Link State Acknowledgement.

7

Q: Způsob zápisu IPv6 adres a možnosti jeho zkracování, srovnání se způsobem zápisu IPv4 adres

A: Zapisuje se po slovech (slovo = 16b), vyjádřeno hexadecimálně -
805b:2d9d:dc28:0000:0000:fc57:d4c8:1fff.

IPv4 se zapisují po bytech, každý byte je vyjádřen dekadicky.

Možnosti zkracování zápisu IPv6

-Leading zero suppressed - nulová slova se zkrátí na jednu číslici

805b:2d9d:dc28:0:0:fc57:d4c8:1fff

-Zero-compressed - nulová slova se zcela vynechají

805b:2d9d:dc28::fc57:d4c8:1fff

lze použít nejvýše jednou v celé IPv6 adrese

-Mixed notation - pro vkládání IPv4 adres do IPv6, posledních 32 bitů se zapíše jako u IPv4

::212.200.31.255

Q: Kolik složek mají globální individuální IPv6 adresy a jaký je jejich význam? Vysvětlete koncept místa (anglicky: site)

A: Celkem tři složky - prefix, subnet ID, interface ID.

-prefix - globální směrování, 48b/64b

-subnet ID - místní topologie (identifikátor podsítě), 0b/8b/16b (0 pokud víme, že nebude potřeba žádné (pod)sítě)

-interface ID - identifikátor rozhraní, 64b

site - skupina (pod)sítí pod jednou společnou správou (majitelem, uživatelem)

Q: Naznačte původně uvažovaný způsob distribuce globálních individuálních IPv6 adres (s pevnou strukturou síťového prefixu) a srovnajte s dnešní skutečnou praxí.

A: Původně:

Koncoví držitelé vždy jen CIDR bloky /48

LIR/ISP CIDR bloky /24

Dnes:

Koncoví držitelé /48, /56, /64

/64 by měli dostat jen tehdy, pokud skutečně nepotřebují více (pod)sítí

LIR/ISP bloky /32

bloky /29 mohou dostat bez nutnosti zdůvodňovat potřebu

Q: Popište, jak z MAC-48 (resp. EUI 48 adresy) vytvořit Interface ID pro IPv6 adresu (pro potřeby autokonfigurace).

A: Doprostřed mezi 3. a 4. byte (mezi OUI a seriové číslo) vloženo FF|FE a v prvním byte převrácení předposlední hodnoty local ↔ global bit (v IPv6 to je naopak oproti EUI, z kterého vychází).

Q: Jaký je smysl lokálních linkových IPv6 adres (link local), jaký je jejich formát a možnosti využití?

A: Jde o privátní IPv6 adresy pro jednotlivé (pod)sítě. Začínají prefixem fe80::/10. Všechny směrovače vědí, že se takové adresy nemají směrovat.

Využití - uzel si může svou lokální linkovou IPv6 adresu přidělit sám v rámci autokonfigurace (zná HW adresu → z ní interface ID → zleva přidá prefix fe80::/10).

Q: Jaký je problém u lokálních místních IPv6 adres (site local) a proč se dnes už nemají používat?

A: Definice místa (site) je velmi vágní, může způsobovat problémy. Například při úniku lokálních místních adres mimo dané místo (při rozšíření site, spojení dvou site) - protože není úplně jasné, co vlastně je site, jak má být velká, etc.

Příčina problému: tyto adresy nijak neidentifikují (nerozlišují mezi sebou) jednotlivá místa.

Q: Jaký smysl a účel mají unikátní lokální IPv6 adresy (ULA) a jak řeší problém, kvůli kterému se nemají používat lokální místní (site local) IPv6 adresy?

A: Jde o privátní adresy pro všechny (pod)sítě v rámci místa (site), ale současně s identifikací celého místa - proto jsou unikátní. Pro identifikaci místa (site) je vyhrazeno 40b - global ID. Global ID se buďto náhodně vygeneruje (každé místo samo, prvních 8b je 1111 1101, tj. fd::/8), nebo se rozdělí nějak systematicky a bude se dbát na unikátnost (prvních 8b je 1111 1100, tj. fc::/8. Dnes se nepoužívá).

Q: Jaké jsou možné dosahy (scope) multicastových IPv6 adres? Jaké další příznaky je rozlišují? Ukažte na příkladu adres pro "všechny uzly" a "všechny směrovače"

A: Další příznaky: flags (rendezvous point - shromazdiste, prefix based - skupina podle prefixu, transient – zda jde o trvalou/dočasnou skupinovou adresu)
dosah (scope) - 4b číslo

1: node-local

dosahem je daný uzel

2: link-local

dosahem je celá síť (vše, co je propojeno na linkové vrstvě)

4: admin-local

dosah musí být nastaven správcem

5: site-local

dosahem je místo (site) (soustava sítí v daném místě)

8: organization-local

dosah musí být nastaven správcem

e:global

celý internet

Příklad*:

uzly: ff0 1/2::1(node/link), směrovače: ff0 1/2/5::2(node/link/site)

Q: Jaký formát mají IPv6 adresy pro vyzývaný uzel (solicited node address) a k čemu se využívají?

A: Mají prefix ff02:0:0:0:0:1:ff::/104, jsou trvalé, nejsou založeny na prefixu (P=0) a mají dosah jen vpnp dané sítě.

Využívají se pro objevování sousedů, překlad IPv6 na HW adresu, detekci duplicitních adres.

Q: Popište koncept výběrových (anycast) IPv6 adres, naznačte příklad jejich využití.

A: Adresují celou skupinu uzlů v dosahu, který je dán velikostí prefixu. Ozve se vždy jeden uzel z celé skupiny (dáno implementací, často nejbližší z pohledu skoků). Využití - rozklad zátěže, zlepšení doby odezvy, lepší rezistence vůči DDOS.

Příklad využití – kořenové DNS servery.

8

Q: Jak jsou řešeny hlavičky u IPv6 paketů a v čem se liší od hlaviček IPv4 datagramů? Které položky z hlavičky IPv4 datagramů byly v IPv6 odstraněny, které přejmenovány a které nahrazeny či nově přidány?

A: IPv6 má více hlaviček místo jedné. IPv4 má jednu hlavičku proměnné velikosti (≥ 20 B). IPv6 má povinnou základní hlavičku - vždy 40B, další rozšiřující hlavičky proměnné velikosti - připojují se pouze v případě, že jsou opravdu zapotřebí. IPv6 má povinný multicast, podpora QoS, mobility, bezpečnosti, jumbo pakety. Odstraněno: length, checksum, Identification, Flags, fragmentation offset, Option field, padding Přejmenováno: ToS (Traffic class), Total length (Payload length), TTL (Hop limit), protocol (next header)

Nově: flow label

Q: Jaké jsou druhy rozšiřujících hlaviček IPv6 paketů, jak jsou řazeny a jak je rozlišen jejich druh? Jaký je obecný formát rozšiřujících hlaviček?

A: Druhy: Hop-by-hop options, destination options, routing, fragment, authentication, ESP (šifrování obsahu), destination options, mobility.

Obecný formát – byte Next Header, byte Header Length (jen u typů s proměnlivou velikostí), poté data podle typu. Druh se určí podle Next Header. Next header má 1B, délka hlavičky je uložena jen u proměnných délek.

Q: Jak se liší způsob řešení fragmentace v IPv6 od IPv4? Jaký je formát rozšiřující hlavičky pro fragmentaci?

A: V IPv6 může fragmentovat pouze odesílající uzel, směrovače uzly nefragmentují (narazí-li směrovač na paket, co by měl být fragmentován, musí ho zahodit).

Pro fragmentaci slouží samostatně rozšiřující hlavička.

Formát: Next Header, rezerva, Fragment offset, 2 nepoužité bity, identification – fragment offset a identification stejná úloha jako u IPv4.

Pokud IPv6 pošle paket do max velikosti 1280 bytů, nedojde k fragmentaci.

Q: V čem se liší protokol ICMPv6 od ICMPv4? Jak se počítá kontrolní součet hlavičky ICMPv6 zprávy

A: Stejný účel. V těle ICMPv4 zprávy je hlavička IPv4 datagramu + prvních 8 bytů jeho těla (--> pevná velikost). V těle ICMPv6 zprávy je co největší část IPv6 paketu, který způsobil chybu („co největší“ = taková, aby velikost IPv6 paketu s ICMPv6 zprávou nepřesáhla 1280B)

Kontrolní součet

U ICMPv4 se počítá (pouze) ze samotné zprávy

U ICMPv6 se počítá navíc také z tzv. pseudohlavičky

Ta je složena z adresy odesílatele, adresy příjemce, délky zprávy a next headeru

Q: Jak funguje postup Path MTU Discovery v IPv6 a v čem se liší od IPv4?

A: Uzel pošle paket o velikosti místního MTU. Pokud neprojde a vrátí se, dostane ICMPv6 zprávu Packet Too Big, z ní se dozví, jaké je MTU v úseku, kde je potřeba fragmentace. Použije novou hodnotu MTU a opakuje.

Rozdíl oproti IPv4 je, že nemusí odhadovat novou hodnotu MTU --> rychlejší.

Q: K jakým účelům se využívá protokol IPv6 Neighbor Discovery? Jaké ICMPv6 zprávy využívá a jaké databáze předpokládá u jednotlivých síťových uzlů?

A: Má několik účelů:

Address Resolution - IPv6 adresa --> HW adresa

Router Discovery- hledání směrovačů, dostupných v dané síti

Prefix Discovery - zjišťování síťového prefixu a dalších parametrů sítě

Parameter Discovery - zjištění „místního“ MTU a počáteční hodnoty, na kterou by měl nastavit Hop Count

Address Auto-Configuration - možnost, aby si uzel sám zvolil svou IPv6 adresu

Neighbor Unreachability Detection - zjišťování (ne)dostupnosti sousedních uzlů

Duplicate Address Detection - zjišťování duplicitních adres

Redirect - informování hostitelského počítače o existenci lepší cesty

Využívané ICMPv6 zprávy:

Router Solicitation

Router Advertisement

Neighbor Solicitation

Neighbor Advertisement

Redirect

Předpokládá se, že každý uzel má vlastní

Neighbor Cache - údaje o sousedech, s kterými komunikoval

Destination Cache - všechny uzly, s kterými komunikoval (nadmnožina neighbor cache)

Prefix List - seznam prefixů, které používají přímo dosažitelné uzly

Default Router List - seznam směrovače v dané síti

Q: Jak funguje převod z IP adresy na HW adresy u IPv6?

A: Multicast na adresu pro vyzývaný uzel (solicited node address) ff02:0:0:0:1:ffxx:xxxx

xx.xxxx je posledních 24 bitů hledané IPv6 adresy

Většinou jde o unicast (členem skupiny v dané síti je pouze jeden uzel)

Dotaz nemá podobu ARP zprávy, ale ICMPv6 zprávy Neighbor Solicitation, ve které se vyplní celá hledaná I/

v6 adresa. Odpověď má podobu ICMPv6 zprávy Neighbor Advertisement, s vyplněnou HW adresou se posílá

zpět.

Q: Jak funguje objevování sousedů u IPv6?

A: Používá ICMPv6 zprávy. Předpokládá, že každý uzel (směrovač i host) má vlastní:

Neighbor cache (obdoba ARP cache)

Destination cache (všech uzlů, se kterými komunikoval – nadmnožina neighbor cache)

Prefix List – seznam prefixů které přímo dostažitelných uzlů.

Default Router List – seznam směrovačů v dané síti.

Princip stejný jako s ARPem – zeptá se všech, ten kdo má danou adresu mu vrátí unicastem odpověď. Dotaz se ale pošle na multicast adresu (IPv6 nemá broadcast).

Na adresu pro vyzývaný uzel ff02:0:0:0:1:ffxx:xxxx , kde xx.xxxx je posledních 24 bitů hledané IPv6 adresy.

Dotaz i odpověď má podobu ICMPv6 zprávy.

Q: Jak funguje zjišťování duplicity IPv6 adres (Duplicate Address Detection)?

A:

1) uzel A se snaží provést Address Resolution na svou novou IPv6 adresu X

Do zprávy Neighbor Solicitation vloží adresu X jako cílovou IPv6 adresu .

Adresu X ještě nesmí používat proto pošle "od :: - kdo má X?" (:: je nespecifikovaná IPv6 adresa) ma adresu vyzývaného uzlu

2) pokud jiný uzel B (ve stejné síti) již používá stejnou IPv6 adresu X, odpoví

svou odpověď rozešle na multicastovou adresu ff02::01 , protože nemůže odpovědět přímo A.

Q: Jak funguje zjišťování přítomnosti směrovačů u IPv6?

A: Směrovače průběžně inzerují svou existenci. Opakovaně posílají Router Advertisement na ff02::01 (tj. všem

uzlům v dané síti). Hostitelské počítače se mohou chovat:

1) Pasivně

počkat, až směrovač rozešle další Router Advertisement

2) Aktivně

rozeslat zprávu Router Solicitation na ff02::02 (tj. všem směrovačům v dané síti) a routery odpoví Router Advertisement, cíleně tazateli.

Q: Jak probíhá bezstavová autokonfigurace v IPv6?

A: Koncový uzel si vygeneruje vlastní lokální linkovou adresu začínající prefixem fe80::/10. Pro Interface ID použije svou HW adresu. Uzel si otestuje jednoznačnost vygenerované lokální linkové

adresy pomocí Duplicate Address Detection (součást Neighbor Discovery). Pokud je unikátní, začne ji používat k oslovení směrovačů (pasivně - směrovače osloví jeho, aktivně - on je). Z informací/odpovědí směrovačů se koncový uzel dozví, zda je k dispozici stavovat konfigurace, jak má směrovat a v jaké síti se nachází (globální směrovací síťový prefix a subnet ID). Na základě toho si přidělí svou globální unikátní IPv6 adresu.

9

Q: Jaká je koncepce transportní vrstvy v TCP/IP, jak se s postupem času mění? Jaké jsou rozdíly mezi protokoly TCP a UDP?

A: Staví na jednotné síťové vrstvě (IP protokol - nespojovaný, nespolehlivý, best effort, žádné QoS). Nabízí dvě varianty přizpůsobení - UDP (minimální změna) vs. TCP (změnit všechno).

-UDP: nespojovaný, nespolehlivý, jednoduchý, bez řízení toku, datagramy (vše stejné jako protokol IP)

-TCP: spojovaný, spolehlivý, složitý, best effort bez QoS (jediné stejné jako protokol IP), zajišťuje řízení toku a předchází zahlcení, přenáší proud bytů

Dnes nenabízí jen TCP a UDP, ale celou paletu protokolů

Q: Popište koncept portů, konvenci o dobře známých portech, a rozdíl mezi dobře známými a registrovanými porty

A: Jde o abstraktní transportní adresy v TCP/IP. Čísla v rozsahu 16b, od 0 do 65535. Data se posílají na adresu (uzel) a port.

Dobře známé porty - konvence přiřazení účelu portům 0 až 1023, od IANA. Stejný port slouží jen jednomu účelu - je pro daný účel vyhrazen.

Registrované porty - 1024 až 49151, konvence zajišťuje unikátnost účelu. Každý port je zaregistrován jen pro jeden účel, ale může se používat i pro jiné účely. Též tzv. uživatelské porty.

Q: Jak a čím jsou jednoznačně identifikována aplikační spojení? Jak dokáže jeden server spolehlivě rozlišit více klientů?

A: Pětice <transportní protokol, IP1, port1, IP2, port2>. Vždy má klient různé alespoň ip, port, nebo protokol

Q: Jaký je rozdíl mezi porty a sockety, jak se pracuje se sockety a jak se zajišťuje jejich vazba na porty?

A: Porty jsou abstrakcí - na všech platformách stejné, musí být nějak implementovány.

Socket - nejčastější implementace portu. Jde o abstrakci souboru (Open, R/W, Close). Lze si ho představit jako bránu, která vede k souboru, nebo je koncovým bodem komunikace. Socket musí být vytvořen, pak je asociován (BIND) se zadaným portem, na všech síťových rozhraních, které uzel má. Pracuje se s ním přes socketové API.

Q: Naznačte způsob spojované komunikace (mezi klientem a serverem) pomocí socketů.

A:

Client = C, Server = S

C – bind S – bind

S – listen

C – connect

S – accept

C – send S – receive

C – receive S – send

C – close S – close

Connect: požadavek na spojení s protistranou na zadané adrese (IP + port)
Listen: čeká na požadavky (socket ve stavu "poslouchání")
Accept: přijetí požadavku na navázání, paralelní server může přijmout další send/recv data
Send: pošle data skrz spojení, navázané socketem
Recv: přijme data skrze spojené, navázané socketem
Close: konec spojení a uvolnění zdrojů

Q: Jaký je formát UDP datagramu a jak se počítá kontrolní součet jeho hlavičky? Jaký je smysl pseudohlavičky?

A: Proměnná velikost, max 2^{16} B. Hlavička pevnou velikost - 8B. Volitelný kontrolní součet. Kontrolní součet se počítá z celého datagramu, doplněného o pseudohlavičku. Ta se ale nepřenáší, je jen pro potřebu výpočtu checksum.

- Hlavička: source port, destination port, length checksum, data

- Pseudohlavička: source adress, destination adress, 0, protocol, total length

Kontrolní součet se počítá v jedničkovém doplňu (nulový kontrolní součet = samé jedničky (tzv. záporná nula), žádný kontrolní součet = samé nuly (tzv. kladná nula))

Kvůli bezpečnosti - při doručení na špatnou adresu (IP datagram) nesedí součet a datagram se zahodí, bez generování ICMP zprávy

Q: Jak funguje a co všechno zajišťuje protokol TCP? Srovnajte s protokolem UDP.

A: Spojovaný, spolehlivý, dvoubodové spojení, řízení toku, ochrana zahlcení, stream, navázání a ukončení spojení, přenáší bloky v datagramech bez featur.

Q: K čemu využívá protokol TCP pozici v bytovém proudu a jak nastavuje počáteční pozici v obou směrech?

A: Pozice v bufferu. Pozici používá k potvrzení doručení až do pozice.

Při odesílání říká: "posílám data z proudu počínaje pozicí X" (v hlavičce sequence number), při potvrzování

říká: "přijal jsem v pořádku data až do pozice Z" (v hlavičce acknowledgment number)

Obě strany se na počátečních pozicích dohodnou - 3-way handshake:

A pošle B: Navrhuji navázat spojení, navrhuji začít od pozice X

B pošle A: Souhlasím s navázáním spojení. Navrhuji začít od pozice Y. Souhlasím s X.

A pošle B: Potvrzuji navázání spojení. Souhlasím s Y.

Q: Jak je zajišťována spolehlivost přenosu dat v protokolu TCP? Jak souvisí s řízením toku?

A: Metoda okénka: odesílání dat až do velikosti okénka, při potvrzení poslat další, jinak opakovat, velikost okénka (příjemce nastaví podle schopností přijímat) tak řeší i řízení toku.

Q: Které položky v hlavičce TCP segmentu se týkají dopředného přenosu dat a které se týkají přenosu dat v opačném směru? Jak funguje piggybacking?

A: Dopředný přenos: source port, destination port, sequence number (= pozice odesílaných dat v bytovém proudu), HLEN, code bits, checksum, urgent pointer

Přenos v opačném směru: acknowledgement number (= pozice potvrzovaných dat), window (= velikost okénka)

Piggybacking: snaha "vložit něco užitečného" do TCP segmentu, který je přenášen v protisměru.

Při navazování spojení - pokud ej nastaven příznak ACK, do segmentu může být vložena a přenesena první část "užitečných dat"

Při potvrzování - potvrzení o přijetí dat lze vložit do TCP segmentu, který je přenášen „v protisměru“ s jinými „užitečnými daty“

Q: Jak probíhá navazování spojení v rámci protokolu TCP?

A: Pomocí 3 way handshake. A - navazuje spojení, B - akceptuje spojení, X, Y- počáteční pozice v

bytovém proudu

A pošle B) volí počáteční pozici v odchozím bytovém proudu, nastaví SYN

B pošle A) akceptuje počáteční pozici v příchozím bytovém proudu. Nastaví ACK. Volí počáteční pozici v odchozím bytovém proudu. Nastaví SYN

A pošle B) Pozice v odchozím bytovém proudu je domluvena. Akceptuje počáteční pozici v příchozím bytovém proudu. Nastaví ACK.

Q: Jak probíhá ukončování spojení v TCP a v čem se liší od navazování spojení?

A: Jde o dva jednostrané úkony. Nemusí na sebe navazovat Každý strana může jednostranně ukončit spojené - když v odesílaném segmentu nastaví příznak FIN - "už nebudu dále nic posílat, ale budu stále přijímat".

Druhá strana může potvrdit ukončení okamžitě pomocí FIN, ACK (první ještě přidá ACK) → 3-fázový handshake.

Nebo může potvrdit pomocí ACK, nadále posílat data a až pak ukončit spojení – FIN (první strana ještě dodá ACK).

Q: Jak protokol TCP bojuje proti zahlcení?

A: Při řízení toku je úzkým hrdlem je příjemce - u TCP jde o metodu okénka. Při zahlcení je úzkým hrdlem přenosová síť - TCP nemá podle čeho poznat, zda zahlcení způsobil někdo jiný - bere to na sebe a sám podniká opatření, jak kdyby to způsobil on. TCP má málo možností, jak poznat, že k zahlcení došlo. Orientuje se primárně podle absence potvrzení.

-TCP slow start - nejprve se odešle jeden TCP segment, pokud je včas a kladně potvrzen, mohou se poslat dva segmenty, atd., dokud není dosaženo maxima, které povoluje aktuální okénk.

-Congestion avoidance - kdykoli TCP nedostane včas kladné potvrzení odeslaného segmentu, chápe to jako jím způsobené zahlcení. Reaguje tak, jako kdyby začínal od začátku - odešle jeden segment a aplikuje pomalý start.