Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Siete, Internet a telekomunikácie
Sytém DNS

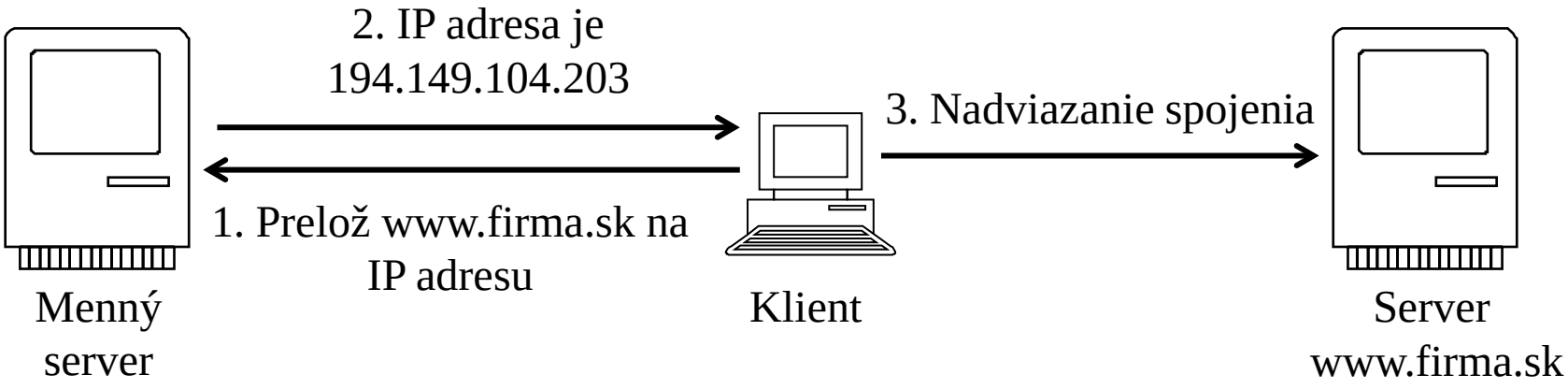
Ladislav Hudec

2013

KPMG
cutting through complexity™

DNS – Domain Name System

- Všetky aplikácie, ktoré zaisťujú komunikáciu medzi počítačmi používajú k identifikácii komunikujúcich uzlov IP adresu. Pre človeka ako používateľa sú však IP adresy ťažko zapamätateľné. Preto sa používá namiesto adresy IP názov sieťového rozhrania. Pre každú IP adresu máme zavedené meno sieťového rozhrania (počítača), presnejšie povedané **doménové meno (domain name)**. Toto doménové meno môžeme používať vo všetkých príkazoch, kde je možné použiť adresu IP. Výnimkou, kde sa musí použiť adresa IP je identifikácia samotného **menného servera** (name server). Jedna adresa IP môže mať priradených aj niekoľko doménových mien.
- Väzba medzi menom počítača a IP adresou je definovaná v databáze DNS. DNS (*Domain Name System*) je celosvetovo distribuovaná databáza. Jednotlivé časti tejto databáze sú umiestnené na tzv. **name (menných) serveroch**.

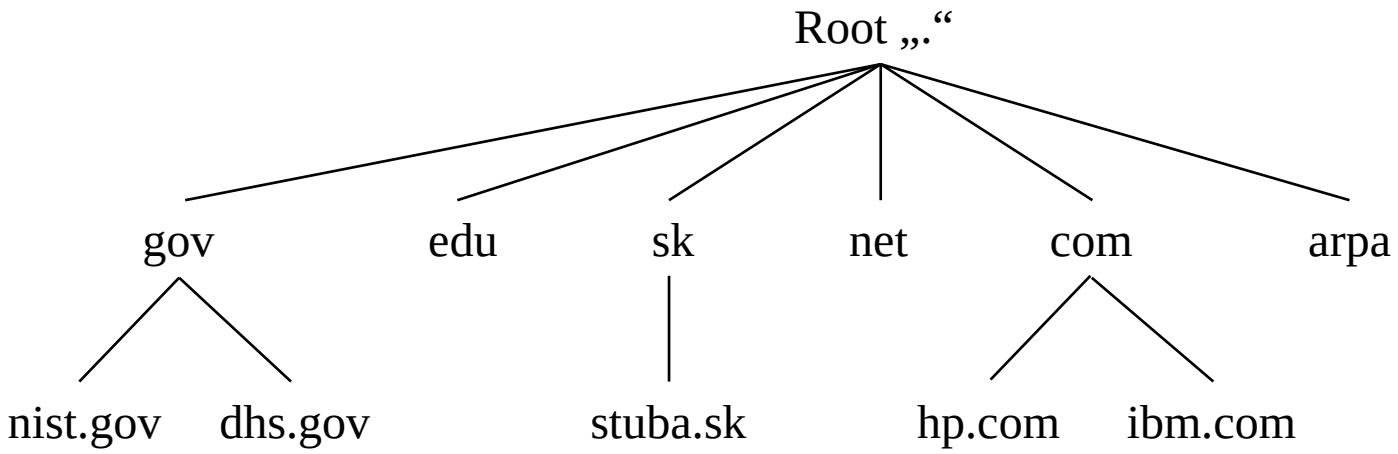


Domény a subdomény

- Internet je rozdelený do tzv. **domén**, tj. skupín mien, ktoré k sebe logicky patria. Domény špecifikujú, či patria mená jednej firme, jednej krajine apod. V rámci domény je možné vytvárať podskupiny tzv. subdomény napr. v doméne firmy vytvoriť subdomény pre oddelenia. Doménové meno odráža príslušnosť uzlu ku skupine a ku podskupine. Každá skupina má priradené meno. Z jednotlivých mien skupín je potom zložené doménové meno uzla.
- Meno je uvádzané v bodkovej notáci. Napr. *cisco.fiit.stuba.sk*. Meno má všeobecnú syntax: *reťazec. reťazec. reťazec.... reťazec.,* kde prvý reťazec je meno počítača (rozhrania), ďalší meno najnižšej vnorenej domény, ďalší vyššej domény atď. Pre jednoznačnosť sa na konci uvádza tiež bodka, vyjadrujúca **root (koreňovú) doménu**.
- Celé meno môže mať maximálne 255 znakov, reťazec potom maximálne 63 znakov. Reťazec sa môže skladať z písmen, číslic a pomlčky. Pomlčka nesmie byť na začiatku ani na konci reťazca. Existujú i rozšírenia špecifikujúce bohatší repertoár znakov použiteľných na tvorbu mien. Zásadne sa však týmto ďalším znakom vyhýbame, pretože iba niektoré aplikácie toto rozšírenie podporujú.
- V zásade možno použiť veľké aj malé písmená. **Z hľadiska uloženia a spracovania v databáze mien (databáza DNS) sa veľké a malé písmená nerozlišujú.** Tj. meno *newyork.com* bude uložené v databáze na rovnaké miesto ako *NewYork.com* alebo *NEWYORK.com* atď. Teda pri preklade mena na IP adresu je jedno kde používateľ zadá velké a kde malé písmená. Avšak v databáze je meno uložené s veľkými a malými písmenami, tj. ak tam bolo uložené napr. *NewYork.com*, potom pri dopyte databáza vráti *NewYork.com*.

Domény a subdomény

- V niektorých prípadoch sa môže časť mena sprava vynechať. Temer vždy môžeme koncovú časť doménového mena vynechať v aplikačných programoch. V databázach opisujúcich domény je však situácia zložitejšia.
- Je možné vynechať:
  - Poslednú bodku takmer vždy.
  - Na počítačoch vnútri domény sa spravidla môže vynechať koniec mena, ktorý je zhodný s názvom domény. Napr. vnútri domény stuba.sk, je možné napísať namiesto počítač.fiit.stuba.sk iba počítač.fiit (nesmie se ale uviesť bodka na konci!). Do ktorých domén počítač patrí sa definuje príkazmi *domain* a *search* v konfiguračnom súbore resolvera.



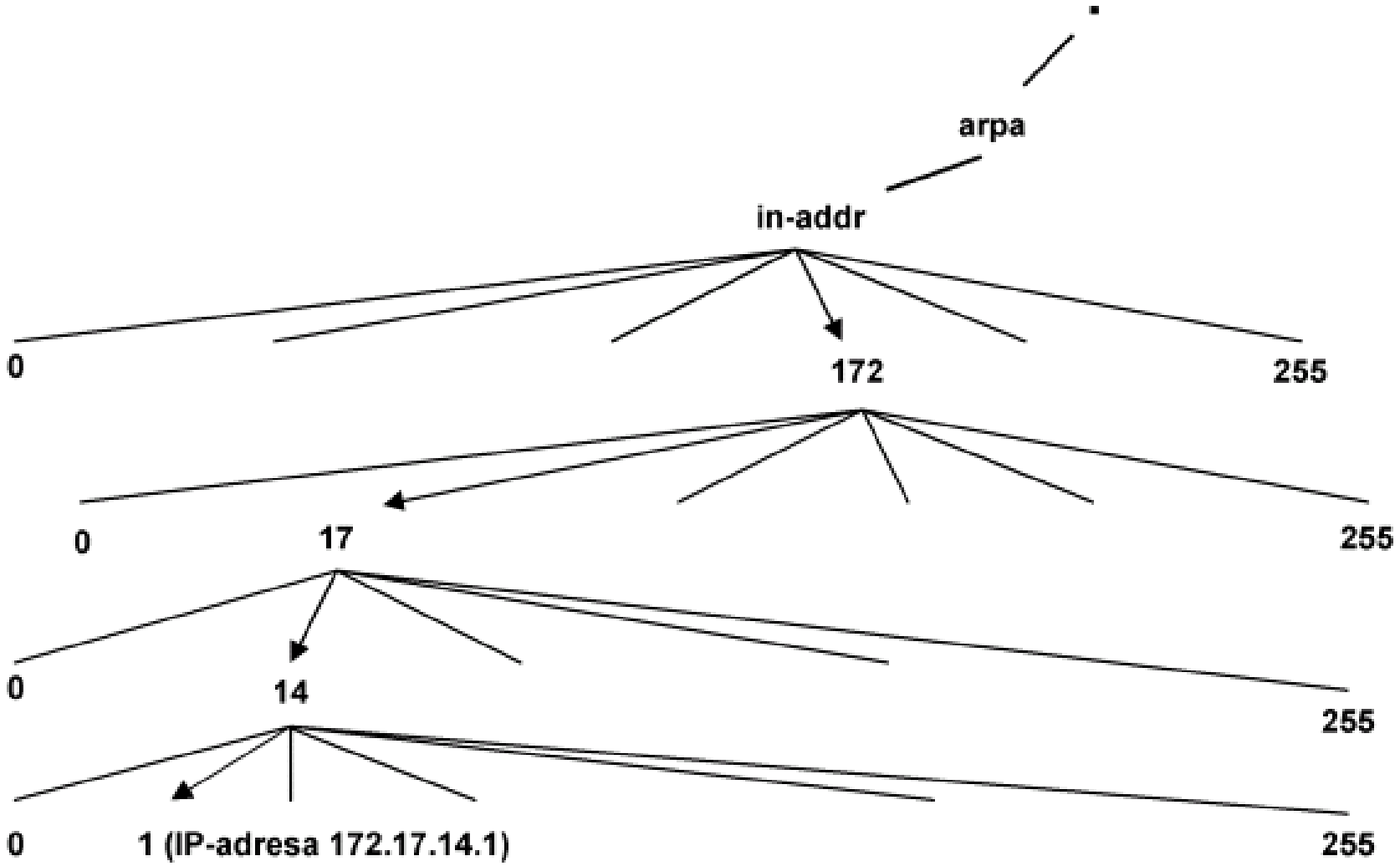
Top Level Domain-**TLD**

Enterprise Level Domain -**ELD**

Reverzné domény

- Už bolo uvedené, že komunikácia medzi uzlami v sieti prebieha na základe adries IP a nie doménových mien. Niektoré aplikácie naopak potrebujú k adrese IP nájsť meno, tj. nájsť tzv. reverzný záznam. Ide teda o preklad IP adresy na doménové meno. Tento preklad sa často nazýva **spätným** (reverzným) **prekladom**.
- Podobne ako domény, tvoria aj adresy IP stromovú štruktúru. Domény tvorené adresami IP sa potom často nazývajú *reverzné domény*. Pre účely reverzného prekladu bola definovaná pseudo doména *"in-addr.arpa"*. Meno tejto pseudo domény má historický pôvod, ide o zkratku z "*inverse addresses in the Arpanet".*
- Pod doménou *in-addr.arpa* sú domény menujúce sa ako prvé číslo z adresy IP siete. Napr. sieť 194.149.101.0 patrí do domény 194.in-addr.arpa. Sieť 172.17 patrí do domény 17.172.in-addr.arpa. Ďalej doména 172.in-addr.arpa sa delí na subdomény, takže sieť 172.17 tvorí subdoménu 17.172.in-addr.arpa. Ak je sieť 172.17 rozdelená pomocou sieťovej masky na subsiete, potom každá subsieť tvorí ešte vlastnú subdoménu. Všimnite si, že domény sú tu tvorené akoby adresami IP sietí písanými ale opačne.

Reverzné domény – doména 1.14.17.172.in-addr.arpa



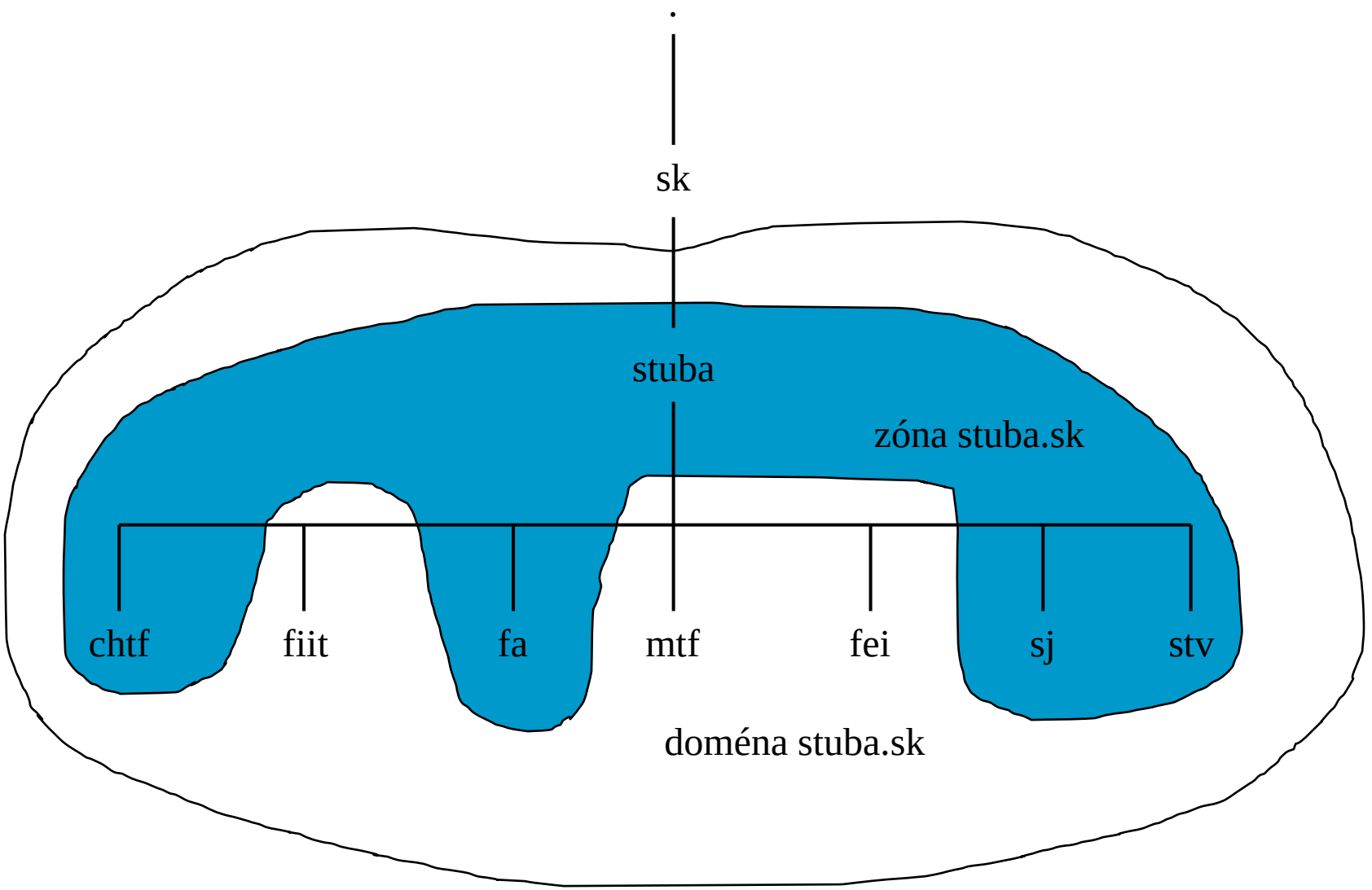
Reverzné domény - doména 0.0.127.in-addr.arpa

- Istou komplikáciou (zvláštnosťou) je adresa siete 127.0.0.1. Sieť 127 je totiž určená pre *loopback*, tj. softvérovú slučku na každom počítači.
- Zatiaľ čo ostatné IP adresy sú v Internete jednoznačné, adresa 127.0.0.1 sa vyskytuje na každom počítači.
- Každý menný server je autoritou nielen "obyčajných" domén ale ešte autoritou (primárnym menným serverom) k doméne **0.0.127.in-adr.arpa**. V ďalšom texte budeme tento fakt považovať za samozrejmosť a v tabuľkách ho pre prehľadnosť nebudeme uvádzať, ale nikdy na neho nesmieme zabudnúť.

Zóna

- Čo je zóna a aký má vzťah k doméne? Ukážeme na príklade domény sk.
- Ako sme už uviedli doména je skupina počítačov, ktoré majú spoločnú pravú časť svojho doménového mena. Doména je napríklad skupina počítačov, ktorých meno končí sk. Doména sk je však veľká. Delí se ďalej na subdomény napr. *stuba.sk*, *tuke.sk*, a tisíce ďalších. **Každú z domén druhej úrovne si väčšinou spravuje na svojich menných serveroch majiteľ domény alebo jeho poskytovateľ Internetu.** Dáta pre doménu druhej úrovne napr. *stuba.sk* nie sú na rovnakom mennom serveri ako doména sk. Sú rozložené na mnoho menných serverov. Dáta o doméne uložené na mennom serveri sú nazývané **zónou (zone file)**. Zóna teda obsahuje iba časť domény. **Zóna je časť priestoru mien, ktorú obhospodaruje jeden menný server.**
- Na nasledujúcom obrázku je znázornené, ako môže byť (hypoteticky) v doméne *stuba.sk* pomocou viet typu NS decentralizovaná kompetencia (delegovanie) na nižšie správne celky. Takže doména *stuba.sk* obsahuje v sebe všetky subdomény, ale zóna *stuba.sk* delegovala na iné menné servery právomoci na zóny *fiit.stuba.sk*, *mtf.stuba.sk* a *fei.stuba.sk*. Takže zóna *stuba.sk* obsahuje doménu *stuba.sk* až na tri uvedené výnimky.

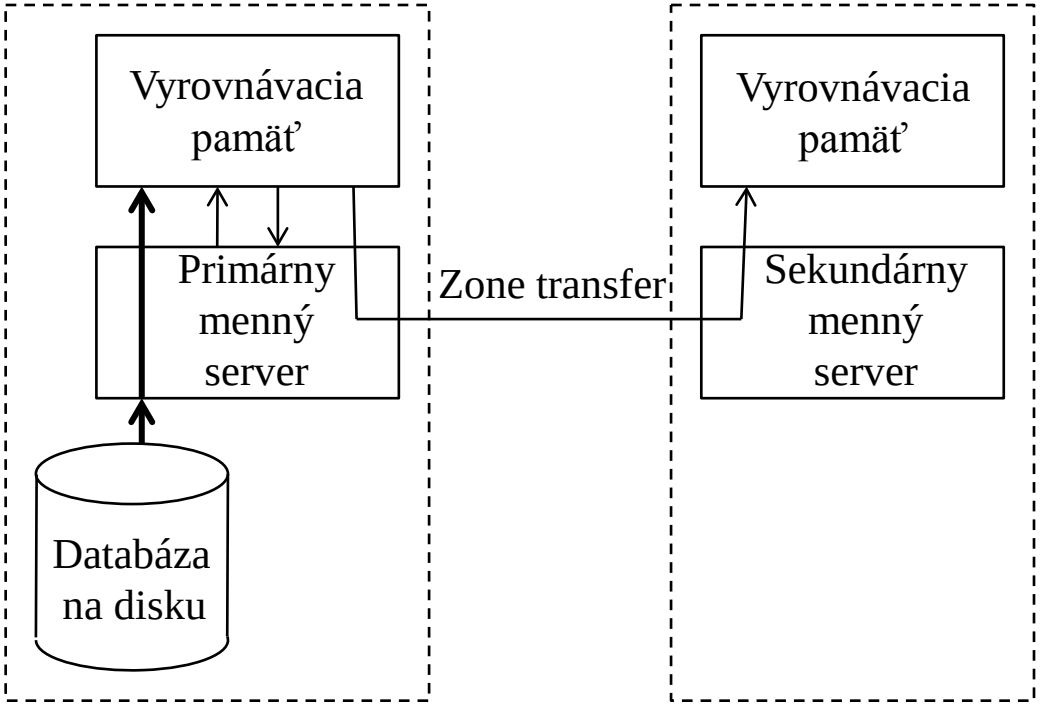
Príklad domény a zóny



Dopyty (Preklady)

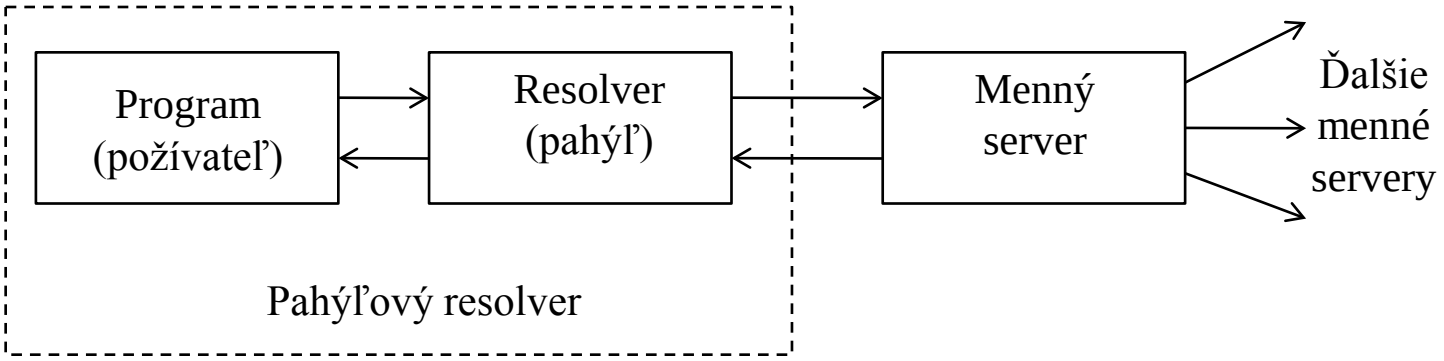
- Preloženie mena na adresu IP sprostredkuje tzv. **resolver** (komponent operačného systému). Resolver je klient, ktorý sa dopytuje menného servera. Pretože je databáza celosvetovo distribuovaná, nemusí najbližší menný server poznať odpoveď, preto môže tento menný server požiadať o pomoc ďalšie menné servery. Získaný preklad potom menný server vráti ako odpoveď resolveru. Všetka komunikácia sa skladá z **dopytov a odpovedí**.
- Menný server po svojom štarte natiahne do pamäti dáta pre zónu, ktorú obhospodaruje. Primárný menný server natiahne dáta z lokálneho disku, sekundárny menný server dopytom *zone transfer* získa pre obhospodarované zóny dáta z primárneho menného servera a takisto ich uloží do pamäti. Tieto dáta primárneho a sekundárneho menného servera sa označujú ako **autoritatívne** (nezvratné). Ďalej menný server natiahne z lokálneho disku do pamäti dáta, ktoré nie sú súčasťou dát jeho obhospodarovanej zóny, ale umožní mu spojenie s koreňovými mennými servermi a prípadne s mennými servermi, ktorým delegoval právomoc pre obhospodarovanie subdomén.Tieto dáta sa označujú ako **neautoritatívne**.
- Menný server i resolver spoločne zdieľajú vyrovnávaciu pamäť. Behom práce do nej ukladajú kladné odpovede na dopyty, ktoré vykonali na iné menné servery, t.j. ku ktorým sú iné menné servery autority. Ale z hľadiska nášho menného servera sú tieto dáta opäť neautoritatívne – iba šetria čas pri opätovných dopytoch.
- Do pamäti sa ukladajú iba kladné odpovede. Prevádzka by bola podstatne zrýchlená, keby sa tam ukladali i negatívne odpovede (negatívny caching), avšak to je podstatne zložitejší problém. Podpora negatívného cachingu je záležitosť posledných niekoľko rokov.

Primárny menný server a sekundárny menný server

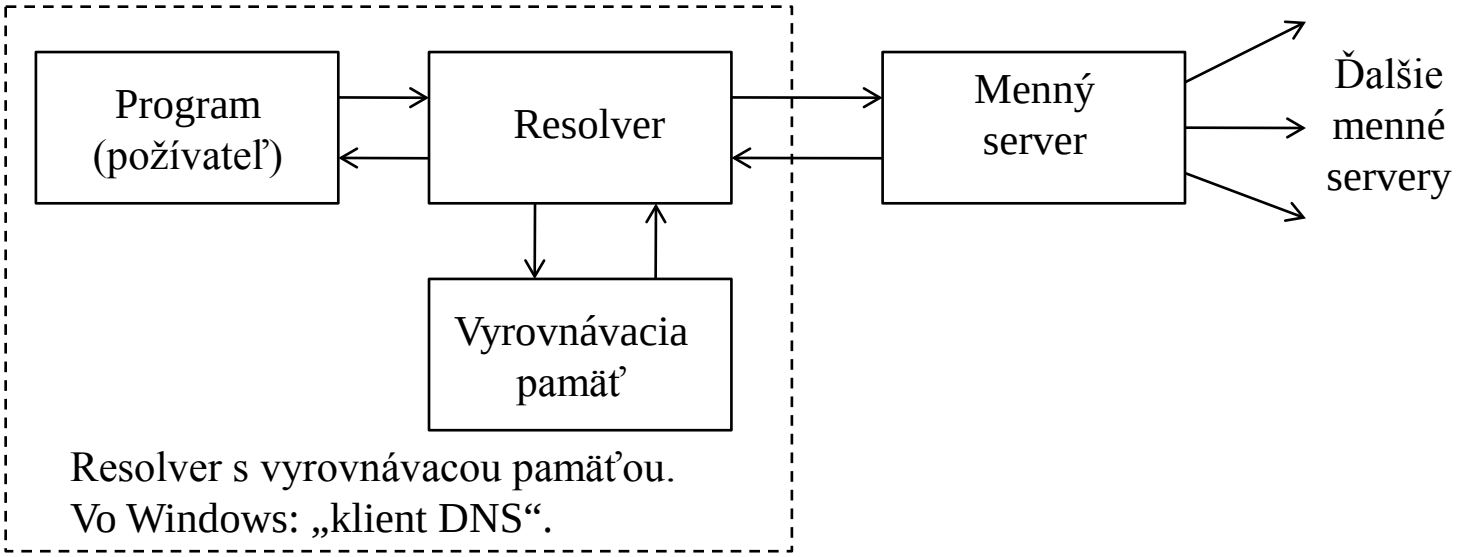


- Primárny menný server načíta dáta z disku, sekundárny menný server získava dáta dopytom ***zone transfer***.

Pahýľový (stub) resolver a resolver s vyrovnávacou pamäťou



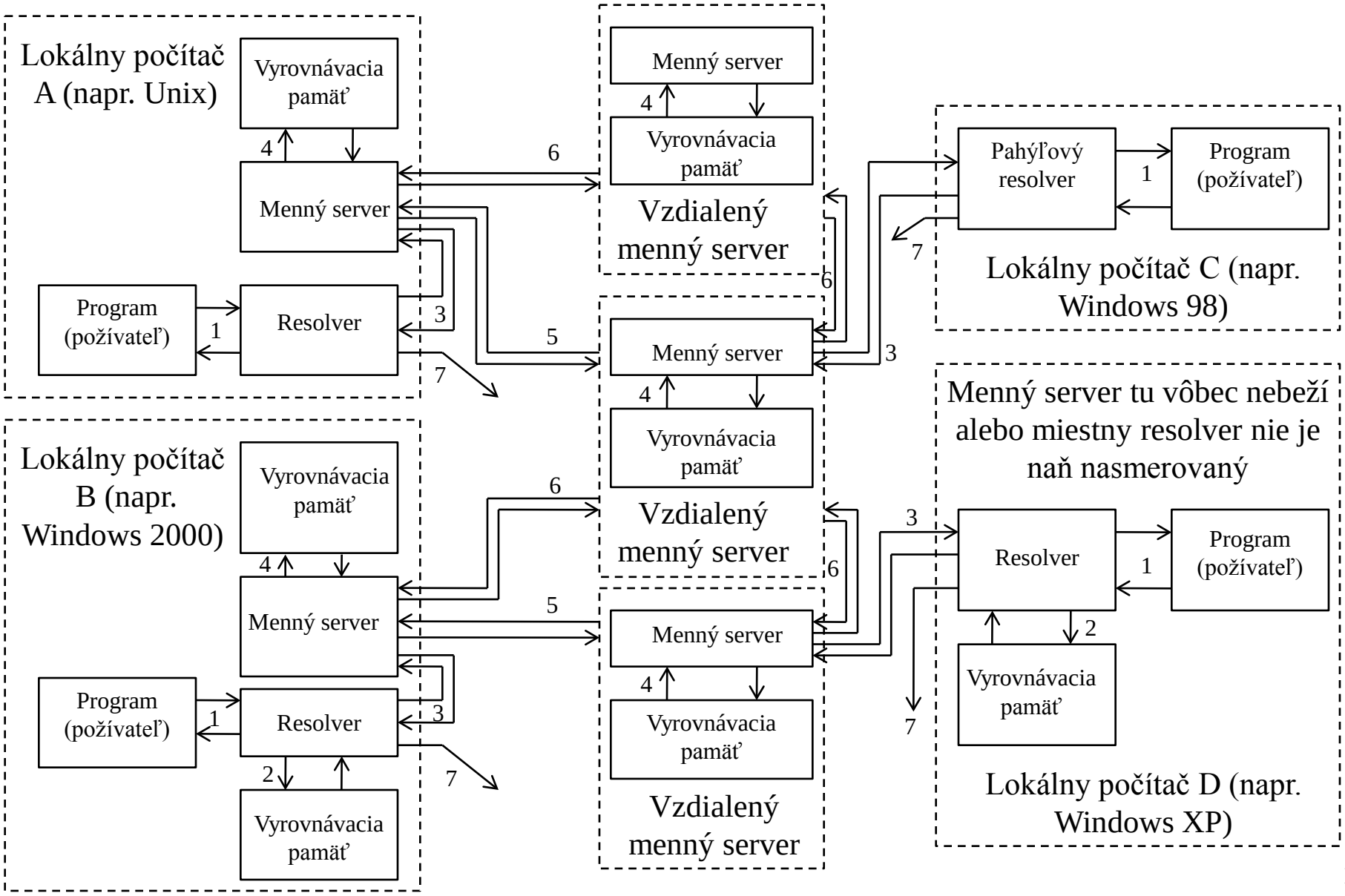
- Resolver bez vyrovnávacej pamäti sa nazýva pahýľový (stub) resolver.
- Pre Windows 2000/XP je možné pre resolver zriadiť vyrovnávaciu pamäť. Táto služba sa vo Windows označuje ako **klient DNS**.



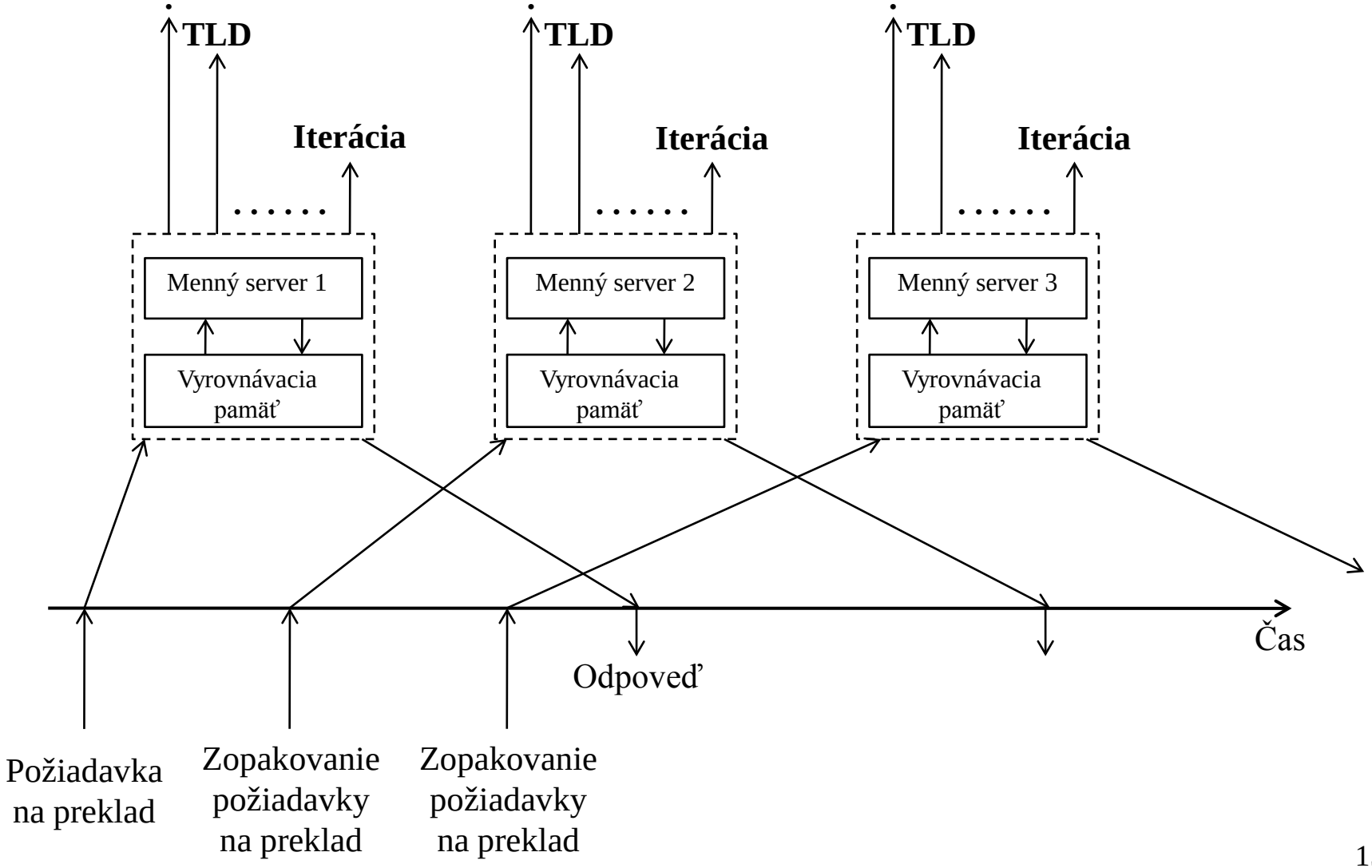
Menný server a resolver

- Na niektorých počítačoch pobeží **len resolver** (či už pahýľový alebo s vyr. pamäťou), na iných počítačoch pobeží **tak resolver ako aj menný server**. Teraz je možný celý rad rôznych kombinácií (na nasledujúcom obrázku). Princíp však zostáva rovnaký:
  1. Používateľ zadá príkaz, ktorý napríklad vyžaduje preložiť meno info.firma.sk na adresu IP (krok číslo 1 na obrázku).
  2. Pokiaľ má resolver vlastnú vyrovnávaciu pamäť, tak sa pokúsi nájsť výsledok (hľadanú adresu IP) priamo v nej.
  3. Pokiaľ sa odpoveď vo vyrovnávacej pamäti resolvera nenájde, tak resolver odovzdá požiadavku mennému serveru.
  4. Menný server hľadá odpoveď na požiadavku vo svojej vyrovnávacej pamäti.
  5. Pokiaľ menný server nenájde odpoveď vo svojej vyrovnávacej pamäti, tak hľadá pomoc u iných menných serverov (a tí prípadne u iných - rekurzia).
  6. Menný server môže kontaktovať viacero menných serverov procesom, ktorý sa nazýva **iterácia**. Iteráciou (začína sa dopytovať na koreňovom mennom serveri, potom menného servera na úrovni TLD, potom menného servera na druhej úrovni, atď) sa server môže dostať až na menný server, ktorý je autoritou pre zadaný dopyt. **Autoritatívny menný server** tak s konečnou platnosťou odpovie (napríklad i záporne, že k zadanému menu v DNS nie sú žiadne informácie).
  7. Ale pokiaľ proces opísaný v predchádzajúcich bodoch dostatočne rýchlo nevráti výsledok, tak resolver **opakuje svoj dopyt** (nasledujúci obrázok). Ak je v konfigurácii resolvera uvedených viacero menných serverov, tak svoj ďalší dopyt pošle resolver na ďalší menný server uvedený v zozname (t.j. iný menný server). Zoznam menných serverov sa prechádza cyklicky. Cyklus sa pre daný dopyt štartuje od menného servera, ktorý je v zozname uvedený ako prvý.

Menný server a resolver



Riešenie požiadavky na preklad



DNS protokol

- DNS používa ako protokol UDP, tak aj protokol TCP. Pre oba protokoly používajú port 53 (tj. porty 53/udp a 53/tcp). Bežné dopyty ako je preklad mena na adresu IP a naopak sa vykonáva prostredníctvom protokolu UDP. Dĺžka prenášaných dát protokolom UDP je implicitne obmedzená na 512 B (príznakom *truncation* môže byť signalizované, že sa odpoveď nevošla do 512 B a pre pridanú kompletnú odpoveď je navyhnutné použiť protokol TCP). Dĺžka paketu UDP je obmezená na 512 B pretože u väčších datagramov IP by mohlo prísť k fragmentácii. Fragmentácia datagramu UDP nepovažuje DNS za rozumnú.
- Dopyty, ktorými sa prenášajú **dáta o zóne** (*zone transfer*) napr. medzi primárnym a sekundárnym name serverom **sa prenášajú protokolom TCP**.
- Bežné dopyty (napr. preklad mena na adresu IP a naopak) sa vykonávajú pomocou datagramov protokolu UDP. Preklad požaduje klient (resolver) na mennom serveri. Ak si nevie menný server rady, môže požiadať o preklad (o pomoc) iný menný server prostredníctvom **koreňového menného servera**.
- V Internete platí pravidlo, že databáza s dátami nevyhnutnými pre preklad sú vždy uložené **aspoň na dvoch nezávislých počítačoch** (nezávislých menných serveroch). Ak je jeden nedostupný, tak se preklad môže vykonať na druhom počítači.
- Všobecne sa nepredpokladá, že by boli všetky name servery dostupné. V prípade, že by sa na preklad použil protokol TCP, tak by nadväzovanie spojenia na nedostupný počítač znamenalo prečkať časové intervaly protokolu TCP pre nadviazanie spojenia a až potom by bolo možno sa pokúsiť nadviazať spojenie s ďalším name serverom.

DNS protokol

- Riešenie pomocou protokolu UDP je elegantnejšie: Datagramom sa vyšle žiadosť prvému mennému serveru, ak sa nedostane odpoveď do krátkeho časového intervalu, tak sa pošle datagramom žiadosť ďalšiemu (záložnému mennému serveru), ak sa nedostane opäť odpoveď, tak sa pošle ďalšiemu atď. V prípade, že sa vyčerpajú všetky možné menné servery, tak sa opäť začne prvým a celý kolotoč sa zopakuje, pokiaľ nepríde odpoveď alebo nevyprší stanovený časový interval.

Resolver

- Resolver je komponent systému zaoberajúci sa prekladom adresy IP. Resolver je klient. Resolver nie je konkrétny program. Je to **sústava knižničných funkcií**, ktorá sa zostavuje (linkuje) s aplikačnými programami, požadujúcimi tieto služby (napr. telnet, ftp, WWW prehliadač atď.). T.j. ak potrebuje napr. telnet preložiť meno počítača na jeho IP adresu, tak zavolá príslušné knižničné funkcie.
- Klient (napr. zmienený telnet) zavolá knižničné funkcie, ktoré sformulujú dopyt a vyšlú ho na server. Server je v UNIX realizovaný programom ***named***. Server buď preklad vykoná sám, alebo si sám vyžiada pomoc od ďalších serverov, alebo zistí, že preklad nie je možný.
- Do hry ešte vstupujú časové obmedzenia. Môže sa totiž stať, že na položený dopyt nedostane resolver odpoveď, ale ďalší rovnaký dopyt už bude korektne zodpovedaný (serveru sa medzitým podarilo získať odpoveď a prvý dopyt nebol zodpovedaný preto, že odpoveď z iného menného servera dlho neprichádzala). Z hľadiska používateľa sa to javí tak, že napoprvé sa preklad nepodarí a pri ďalšom zadaní toho istého príkazu už áno.
- Podobný efekt spôsobuje aj použitie protokolu UDP. Môže sa totiž tiež stať, že server vôbec žiadosť o preklad neobdrží, pretože je sieť preťažená a UDP datagram sa proste niekde stratil.
- Klient môže síce mať v **konfiguračnom súbore** uvedených viacero menných serverov, ale použije sa vždy iba odpoveď, ktorá prišla prvá. Tj. Keď ako prvá príde negatívna odpoveď (napr., že k danému menu neexistuje adresa IP), nepokúsi sa resolver kontaktovať ďalší menný server, ktorý by meno snáď preložil (ako si mnohí predstavujú), ale oznámi, že preklad k danému menu neexistuje.

Resolver

- Konfiguračný súbor pre resolver sa v operačnom systému UNIX menuje */etc/resolv.conf*. Spravidla obsahuje dva typy riadkov (druhý sa môže niekoľkokrát opakovať):
  - *domain meno_miestnej_domény*
  - *nameserver IP_adresa_name_servera*
- V prípade, že používateľ zadal meno bez bodky na konci, tak resolver za zadané meno pridá meno domény z príkazu domain a pokúsi sa meno odovzdať mennému serveru na preloženie. V prípade, že sa preklad nevykoná (negatívna odpoveď menného servera), tak sa resolver pokúsi ešte preložiť samotné meno, tj. bez prípony z príkazu domain.
- Niektoré resolvery umožňujú mená miestnych domén zadať príkazom *search*.
- Príkazom *nameserver* sa špecifikuje IP adresa menného servera, ktorý má resolver kontaktovať. Je možné uviesť aj ďalšie príkazy *nameserver* pre prípad, že niektoré menné servery sú nedostupné. Musí sa tu uviesť **IP adresa name serveru** – a nie doménové meno menného servera!
- V prípade konfigurácie resolvera na mennom serveri môže príkaz *nameserver* ukazovať na miestny menný server 127.0.0.1 (nemusí to však byť pravidlom).
- Ďalšie parametre resolvera (napr. maximálny počet príkazov *nameserver*) sa dá nastaviť v konfiguračnom súbore jadra. Tento súbor sa často menuje */usr/include/resolv.h*. Musí pochopiteľne potom nasledovať zostavenie jadra operačného systému.

Menný server

- Menný server:
  - udržuje informácie pre preklad mien počítačov na adresy IP (resp. pre reverzný preklad).
  - obhospodaruje nejakú časť z priestoru mien všetkých počítačov. Táto časť sa nazýva **zóna**. Zóna je tvorená doménou alebo jej časťou. Menný server totiž môže pomocou vety typu NS vo svojej konfigurácii delegovať obhospodarovanie subdomény na menný server nižšej úrovni.
  - je program, ktorý vykonáva na žiadosť resolvera preklad. V UNIX je menný server realizovaný programom *named*.
- Podľa uloženia dát rozlišujeme tieto typy menných serverov:
  - **Primárny menný server** udržuje dáta o svojej zóne v databázach na disku. Iba na primárnom mennom serveri má zmysel editovať tieto databázy.
  - **Sekundárny name server** si kopíruje databázu v pravidelných časových intervaloch z primárneho menného servera. Tieto databázy nemá zmysel na sekundárnom mennom serveri editovať, pretože budú pri ďalšom kopírovaní prepísané. Primárne a sekundárne menné servery sú tzv. autoritou pre svoje domény, t.j. ich dáta pre príslušnú zónu sa považujú za autoritatívne (nezvratné).
  - **Caching only server** nie je pre žiadnu doménu ani primárnym ani sekundárnym menným serverom (nie je žiadnou autoritou). Avšak využíva všeobecné vlastnosti menného servera, t.j. dáta, ktoré ním prechádzajú ukladá do svojej pamäti. Tieto dáta sa označujú ako **neautoritatívne**. Každý server je caching only server, ale slovami caching only zdôrazňujeme, že pre žiadnu zónu nie je ani primárnym ani sekundárnym menným serverom (Pochopiteľne aj caching only server je primárnym name serverom pre zónu 0.0.127.in-addr.arpa, ale to sa nepočíta).

Menný server

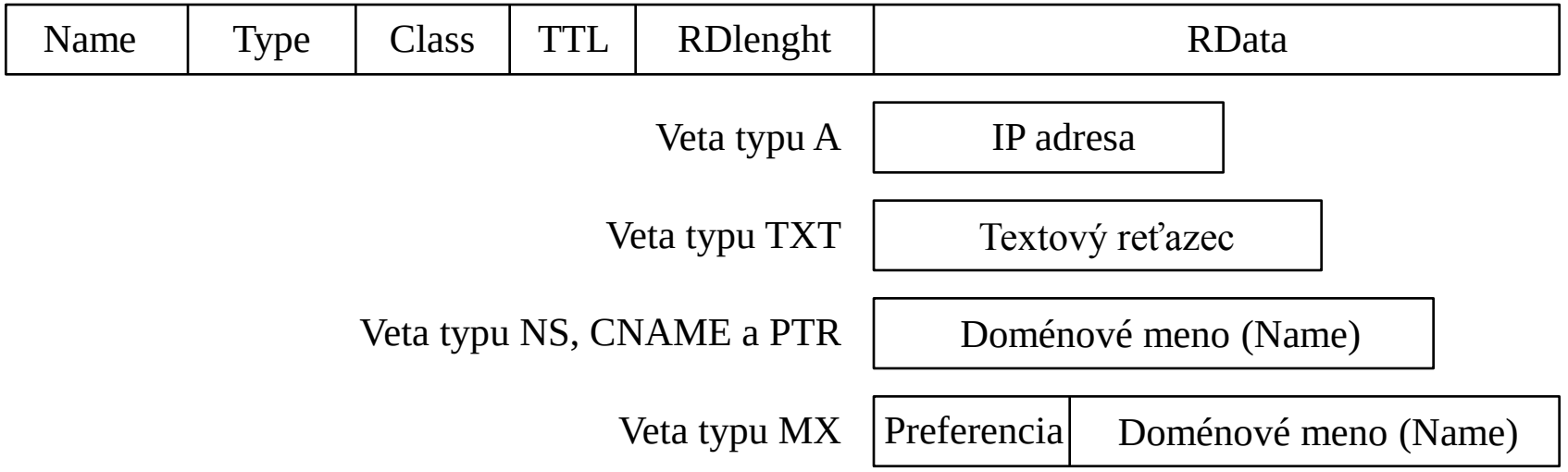
- Podľa uloženia dát rozlišujeme tieto typy menných serverov:
  - **Koreňový menný server** (Root name server) je menný server obsluhujúci koreňovú doménu. Každý koreňový menný server je **primárnym serverom**, čo ho odlišuje od ostatných menných serverov.
- Jeden menný server môže byť pre nejakú zónu primárnym serverom, pre iné sekundárnym serverom.
- Z hľadiska klienta nie je žiadny rozdiel medzi primárnym a sekundárnym menným serverom. Oba majú dáta rovnakej dôležitosti - oba sú pre danú zónu autority. Klient nemusí ani vedieť, ktorý server pre zónu je primárny a ktorý sekundárny. Naproti tomu caching only server nie je autoritou, tj. ak nedokáže vykonať preklad, tak kontaktuje autoritatívny server pre danú zónu.
- Takže ak pridá správca zóny (*hostmaster)* do databázy na primárnom mennom serveri ďalší počítač, tak po dobe stanovenej parametrom vo vete SOA sa táto databáza automaticky opraví aj na sekundárnych menných serveroch (ak by opravil ručne iba databázu na sekundárnom mennom serveri, tak by po rovnakej dobe oprava zmizla!). Problém nastane v prípade, že používateľ v dobe, keď ešte nie je sekundárny menný server aktualizovaný, dostane prvú odpoveď od sekundárneho menného serveru. Tá je negatívna, tj. taký počítač v databáze nie je.

Menný server

- Klasickou chybou je, že primárny menný server pracuje korektne, ale na sekundárnom mennom serveri z nejakého dôvodu nie sú dáta pre zónu. Klienti dostávajú autoritatívne odpovede z primárneho menného servera či zo sekundárneho menného servera. Odpovede z primárneho menného servera správne prekladajú, kdežto odpovede zo sekundárneho menného servera sú negatívne (používatelia potom hovoria: “raz to ide raz nie”).
- Autoritatívne dáta pochádzajú z databázy na disku. Je tu iba jedna výnimka. Pre správnu činnosť menného servera musí menný server poznať koreňové menné servery. Pre tie však nie je autoritou, aj tak každý menný server má na disku databázu informácií o koreňových serveroch, ktorú ale zavádza príkazom *cache* do sekundárnej pamäti (nie je k ním autorita).

Vety RR (Resource Records)

- Informácie o doménových menách a im prislúchajúcich adresách IP, tak isto ako všetky ostatné informácie distribuované pomocou DNS sú uložené v pamäti serverov DNS v tvare **zdrojových viet RR** (*Resource Records)*. Menný server naplňuje svoju pamäť niekoľkými spôsobmi. Autoritatívne dáta načíta zo súboru na disku, alebo ich získa pomocou dopytu *zone transfer* z pamäti iného servera. **Neautoritatívne dáta získavá postupne z pamäti iných serverov, tak ako vybavuje jednotlivé dopyty DNS.**
- V prípade, že klient DNS (resolver) potrebuje získať informácie z DNS tak požaduje od menného servera vety RR podľa zadaných požiadaviek. Klient môže napr. požadovať od servera vety RR typu A, ktoré obsahujú adresy IP pre dané doménové meno, apod.
- Všetky vety RR majú rovnakú štruktúru. Štruktúra RR vety je na obrázku.



Vety RR (Resource Records)

- Jednotlivé položky formátu RR vety predstavujú:
  - **Name** (premenlivá dĺžka) – doménové meno
  - **Type** (2B) – typ vety
  - **Class** (2B) – trieda vety
  - **TTL** (4B) - Time to live, udáva dobu, počas ktorej môže byť tento RR udržovaný vo vyrovnávacej pamäti servera ako platný. Po vypršaní tejto doby musí byť veta považovaná za neplatnú. Hodnota 0 zabraňuje neautoritatívnym serverom uložit RR vetu do vyrovnávacej pamäti.
  - **RDlenght** (2B) - špecifikuje dĺžku poľa RData.
  - **RData** (premenlivá dĺžka) - vlastné dáta v tvare reťazca. Formát tohto poľa závisí na type a triede RR.
- Najčastejšie typy viet:
  - **A** (A host address) – 32 bitová IP adresa
  - **NS** (Authoritative Name Server) – Doménové meno menného servera, ktorý je autoritou pre danú doménu.
  - **CNAME** (Canonical name for an alias) – Doménové meno špecifikujúce synonymum k NAME
  - **SOA** (Start Of Authority) – Práve jedna veta SOA je na začiatku každej zóny. Obsahuje 7 polí.
  - **PTR** (Domain name pointer) – Doménové meno. Veta sa používa pre reverzný preklad.
  - **HINFO** (Host information) – Obsahuje dva znakové reťazce. Prvý obsahuje opis HW a druhý opis SW, ktoré sú používané na počítači Name.
  - **MX** (Mail exchange) – Obsahuje dve polia. Prvé 16 bitové pole bez znamienka obsahuje preferenciu a druhé obsahuje doménové meno mailového servera.

Vety RR (Resource Records)

- Najčastejšie typy viet:
  - **TXT** (Text string) – Textový reťazec s opisom.
  - **AAAA** (IP6 address) – 128 bitová adresa IP (IP verzia 6).
  - **WKS** (Well known service description) – Opisuje obvyklé služby servera v protokoloch TCP a UDP. Obsahuje 3 časti: 32 bitovú adresu, číslo protokolu, porty služieb.
  - **SIG** (Security signature) – Podpisová veta, používaná pri autentizácii v Secure DNS.
  - **KEY** (Security key) – Verejný kľúč zóny používaný na overenie podpisu pri autentizácii.
  - **NXT** (Next domain) – Ďalšie doménové meno. Potvrdenie neexistencie doménového mena a typu.

Protokol DNS

- Doménová služba je realizovaná jednoduchým protokolom. Tento protokol pracuje spôsobom **dopyt – odpoveď**. Klient pošle dopyt serveru a server na dopyt odpovie. Istou komplikáciou je kompresia mien, ktorá sa vykonáva preto, aby boli pakety DNS čo najúspornejšie.
- Protokol DNS je protokol aplikačnej vrstvy, nerieši teda otázku vlastného prenosu paketov. Prenos svojich paketov zveruje transportnému protokolu. Na rozdiel od drvivej väčšiny ostatných aplikačných protokolov využíva DNS ako transportný protokoly UDP aj TCP. Dopyt aj odpoveď sú prenášané vždy rovnakým transportným protokolom.
- Pri dopytoch na preklad (tj. žiadosti o RR) je dávaná prednosť protokolu UDP. V prípade, že je DNS odpoveď dlhšia ako 512B, vloží sa do odpovedi iba časť informácii nepresahujúca 512B a v záhlaví sa nastaví bit TC, špecifikujúcí, že ide o neúplnú odpoveď. Klient si môže kompletnú odpoveď vyžiadať protokolom TCP.
- Pri prenose zón napr. medzi primárnym a sekundárnym menným serverom sa používa protokol TCP. Menný server štandardne očakáva dopyty ako na porte 53/udp tak i na porte 53/tcp.

Dopyt a odpoveď DNS

- Protokol DNS používa niekoľko typov operácií. Štandardnou najčastejšie používanou operáciou je **DNS QUERY** (dopyt DNS na získanie RR záznamu). Ďalšími typmi operácií sú napríklad **DNS NOTIFY** (informácia sekundárnych serverov o zmene údajov v zóne - zone file) alebo **DNS UPDATE** (dynamické aktualizácie v databáze DNS).
- **Formát paketu DNS**
  - DNS používa rovnaký formát paketu pre dopyt aj pre odpoveď. Paket sa môže skladať až z piatich sekcií, vždy musí obsahovat sekciu **HEADER** (záhlavie). Ďalšími štyrmi **sekciami** sú:
    - **QUESTION** – dopyty
    - **ANSWER** - odpovede
    - **AUTHORITY** - autoritatívne name servery
    - **ADDITIONAL** - doplňujúce informácie.
- **Záhlavie paketu** je povinné (formát na obrázku), je obsiahnuté v dopyte aj odpovedi.
  - Prvé dva bajty (16 bitov) záhlavia obsahujú identifikátor správy ID. Identifikáciu správy generuje klient a server ju kopíruje do odpovedi. Identifikácia slúži na párovanie dopytu a odpovedi. Jednoznačne určuje, ku ktorému dopytu patrí ktorá odpoveď. Identifikácia umožňuje klientovi posielať viacero dopytov súčasne, bez toho aby musel čakať na odpoveď.
  - Ďalšie dva bajty záhlavia obsahujú riadace bity.

Formát záhlavia paketu dopyt DNS

ID – identifikátor správy

QR – indikácia dopyt (0), odpoveď (1)

OPCODE – typ otázky, je rovnaký v dopyte aj odpovedi, 0-štandardná otázka QUERY, 1-inverzná otázka QUERY, 2 – otázka na STATUS, 4-otázka NOTIFY, 5 – otázka UPDATE

AA – odpoveď nie je autoritatívna (0), odpoveď je autoritatívna (1)

TC – 1-odpoveď bola skrátená na 512B. Ak má klient záujem o celú odpoveď, musí dopyt zopakovať TCP protokolom.

RD – 1-pokiaľ klient požaduje rekurzívny preklad (dôležité pre dopyt)

RA - 1-pokiaľ server umožňuje rekurzívny preklad (dôležité pre odpoveď)

Z – rezervované pre budúce použitie

RCODE – výsledkový kód odpovedi , 0-bez chyby, 1-chyba vo formáte dopytu, server ju nevie interpretovať, 2- server nevie odpovedať, 3- meno z dopytu neexistuje, túto odpoveď môžu dať iba autoritatívne menné servery, 4- server nepodporuje tento typ dopytu, 5- server odmieta odpovedať

Formát paketu DNS – sekcia QUESTION

- Pakety dopytov DNS obsahujú väčšinou iba jednu sekciu a to sekciu dopytu (QDCOUNT=1). Sekcia dopytu obsahuje tri polia:
  - **QName** - obsahuje doménové meno. Protokol DNS nepoužíva na vyjadrenie doménového mena bodkovú notáciu. Každá časť doménového mena (v bežnom zápise mezi bodkami) začína bajtom obsahujúcim dĺžku reťazca. Na konci doménového mena je nula označujúca koniec doménového mena (nulová dĺžka reťazca). Príkladom obsahu tohto poľa v dopyte na preklad doménového mena fiit.stuba.sk: 4fiit5stuba2sk0. Dĺžky reťazca sú v binárnom tvare.
  - **QType** - špecifikuje typ dopytu, t.j. požadovaný typ vety v odpovedi. Napríklad, kód 1 indikuje požiadavku na vetu typu A (požiadavka na získanie adresy IP verzie 4), kód 2 indikuje požiadavku na vetu typu NS (požiadavka na získanie autoritatívnych menných serverov), kód 5 indikuje požiadavku na získanie vety typu CNAME, kód 6 indikuje požiadavku na získanie vety typu SOA, kód 15 indikuje požiadavku na získanie vety vetu typu MX, atď.
  - **QClass** - špecifikuje triedu dopytu. Napríklad kód 1 indikuje triedu IN - Internet

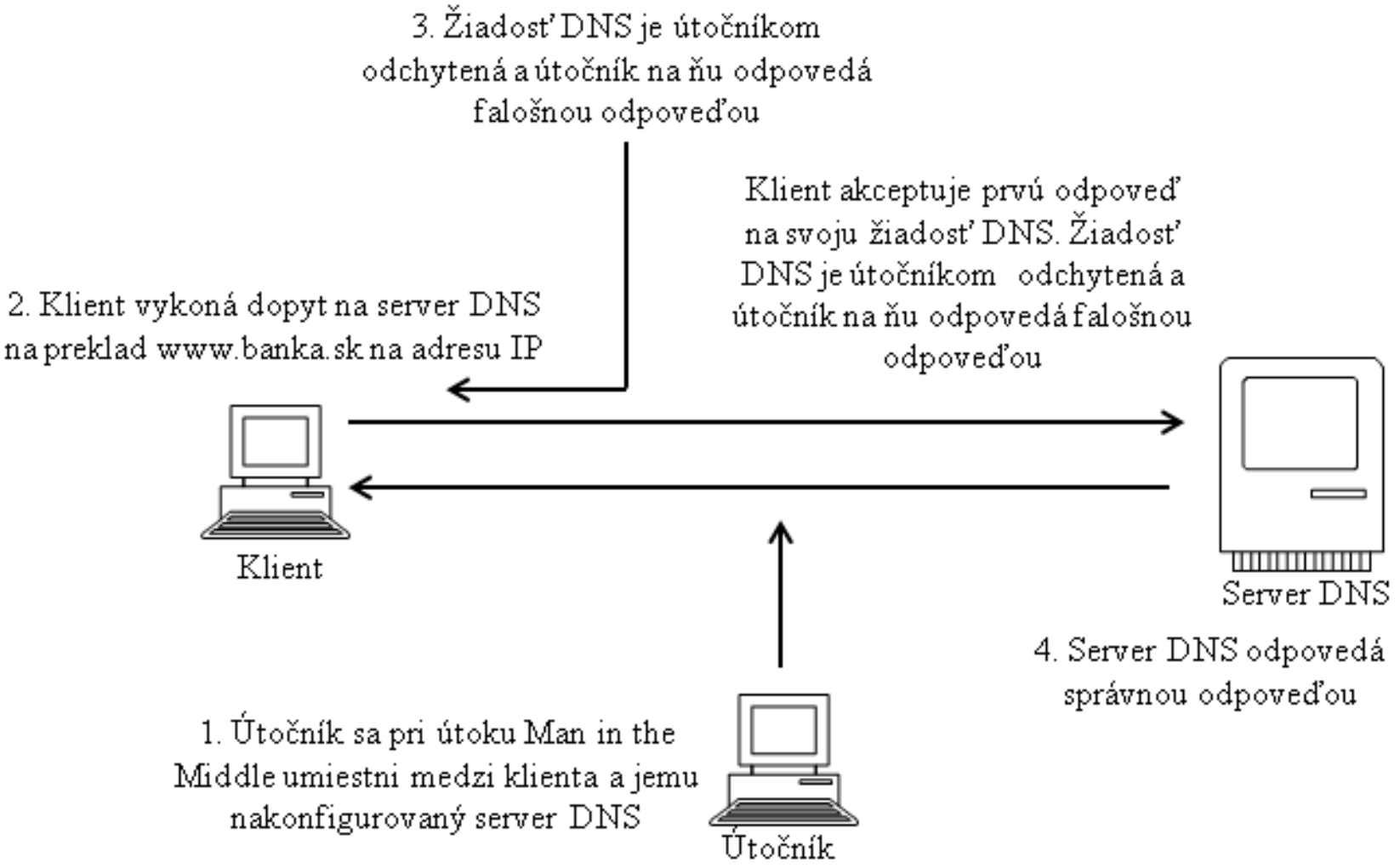
Formát paketu DNS – sekcia ANSWER

- Rovnaký formát paketu DNS majú aj sekcie AUTHORITATIVE SERVERS a ADDITIONAL INFORMATIONS
- Pakety odpovedi obsahují obvykle okrem záhlavia a zopakovanej sekcie dopytu ešte tri sekcie: sekciu odpovedi, sekciu autoritatívnych serverov a sekciu doplňujúcich informácií. Sekcia autoritativne menné servery obsahuje mená menných serverov uvedených vo vetách NS. Sekcia doplnkové údaje obsahuje obvykle IP adresy autoritatívnych menných serverov. Vety v týchto sekciách sú bežné zdrojové záznamy (RR) - podobné vetám vo vyrovnávacej pamäti menného servera a majú spoločný formát:
  - **Name** - obsahuje doménové meno, rovnaký formát ako v sekcii dopytu QName
  - **Type** - špecifikuje typ vety, rovnaký formát ako v sekci dopytu QType
  - **Class** - trieda vety, rovnaký formát ako v sekcii dopytu QClass
  - **TTL** - doba platnosti RR, t.j. ako dlho môže byť odpoveď udržovaná ako platná vo vyrovnávacej pamäti.
  - **RDlenght** - dĺžka časti RData
  - **RData** – pravá strana zdrojovej vety (IP adresa alebo doménové meno).

Útoky na DNS – Man in the Middle

- V prípade, že **útočník je schopný odchytávať komunikáciu medzi klientom a serverom DNS**, potom útočník vie tiež odchytávať dopyty klienta na resolvenciu mena a poslať klientovi (namiesto servera DNS) falošnú odpoveď, ktorá mapuje meno domény na nesprávnu IP adresu.
- Tento útok je založený **na súbehu odpovedí**, a to falošnej odpovedi od útočníka a odpovedi oprávneného servera DNS. Útočník musí na dopyt klienta na resolvenciu mena odpovedať skorej ako odpovie oprávnený server DNS. Zdržanie odpovedi (ak je to nevyhnutné) oprávneného servera DNS je možné vykonať zaslatím viacerých dopytov na resolvenciu (simulácia útoku DoS na server DNS) alebo požiadavkou klienta na rekurzívny dopyt.
- Demonštrácia útoku je na nasledujúcom obrázku a skladá sa z týchto krokov:
  1. Útočník sa umiestni v štruktúre sieti medzi klienta a menného server (sieťová kolízna doména s klientom alebo na segment, kde je umiestnený menný server)
  2. Klient vykoná dopyt DNS na resolvenciu domény www.banka.sk
  3. Dopyt je odchytený útočníkom, ktorý odpovedá falošnou informáciou
  4. Server DNS odpovedá správnou informáciou, ale túto informáciu klient neakceptuje, pretože už dostal a akceptoval informáciu od útočníka.
- Na realizáciu takéhoto útoku existujú voľne šíriteľné nástroje.

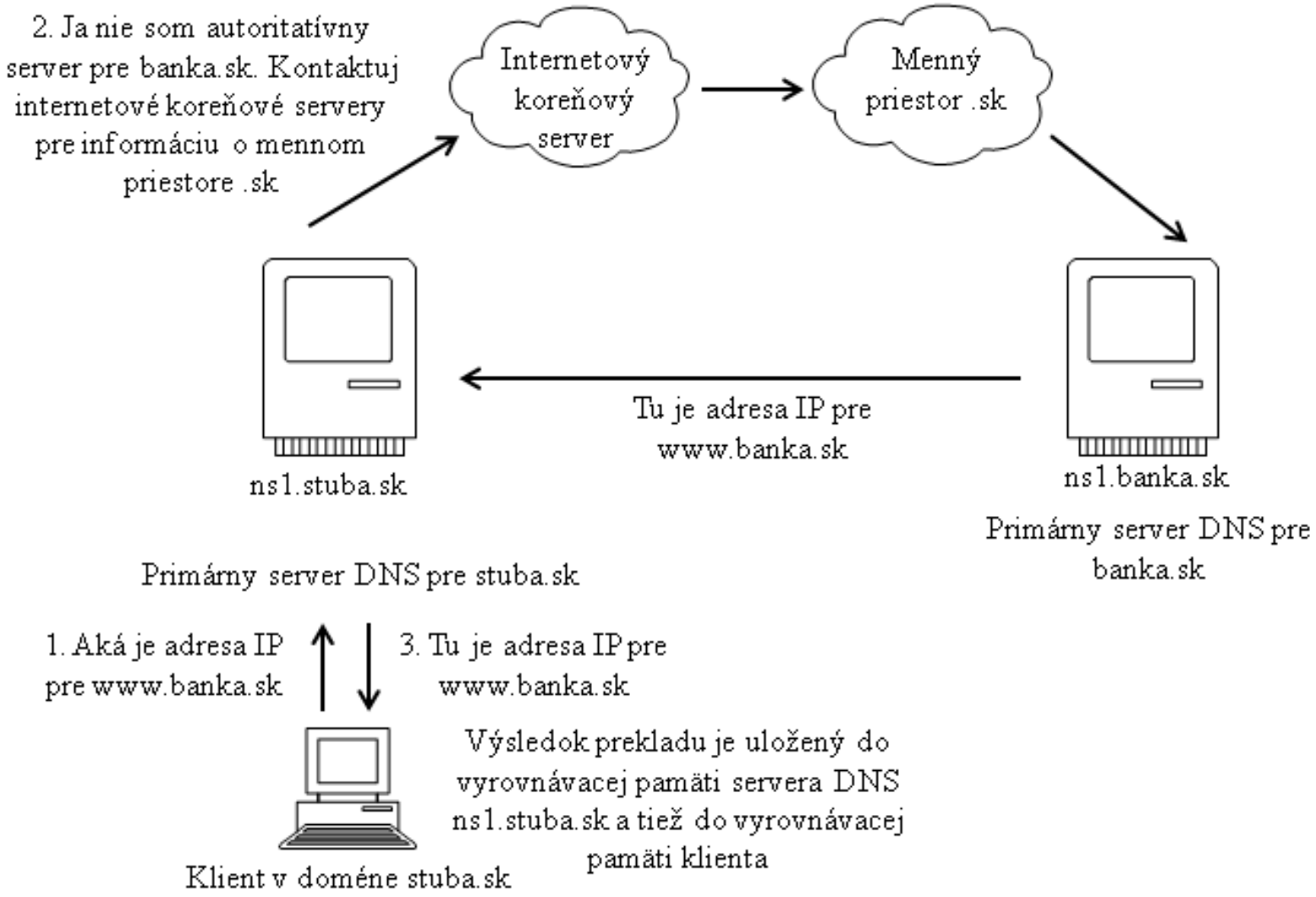
Útoky na DNS – Man in the Middle



Útoky na DNS – cache poisoning

- Ak klient v doméne **stuba.sk** vykoná dopyt na resolvenciu adresy IP pre doménu www.banka.sk, typicky sa vykoná takáto sekvencia udalostí, ktoré sú dokumentované na nasledujúcom obrázku:
  1. Klient kontaktuje jemu nakonfigurovaný server DNS a požiada ho o resolvenciu www.banka.sk. Tento dopyt bude obsahovať informáciu o klientovom čísle zdrojového portu UDP, adrese IP a ID transakcie DNS.
  2. Klientov server DNS, pretože nie je autoritatívnym serverom pre doménu banka.sk, prostredníctvom dopytov cez Internetové koreňové servery DNS kontaktuje server DNS pre .sk a získa odpoveď na dopyt.
  3. Tento úspešný dopyt potom server DNS pošle klientovi naspäť a aj server DNS aj klient si túto informáciu **uložia do vyrovnávacej pamäti**.
- Na uvedenej sekvencii udalostí si treba všimnúť tieto skutočnosti:
  - V kroku 3 klient akceptuje iba takú spätnú odpoveď od servera DNS, v ktorej server DNS použije **správne číslo zdrojového portu, adresy IP a ID transakcie**, tak ako boli použité pri dopyte v kroku 1. Tieto tri položky sú jedinou formou **autentizácie** použitej na akceptáciu odpovedí DNS.
  - Spätná odpoveď od servera DNS domény www.banka.sk je uložená do vyrovnávacej pamäti na serveri DNS ns1.stuba.com a tiež do vyrovnávacej pamäti na klientovi a to po dobu špecifikovanú parametrom TTL (time to live). Ak iný klient požiada server DNS ns1.stuba.com o resolvenciu doménového mena www.banka.sk počas platnosti tohto záznamu (daný TTL), potom server DNS na tento dopyt vráti informáciu zo svojej vyrovnávacej pamäti a nebude posielať dopyty na iné menné servery (koreňový, .sk a ns1.banka.sk).

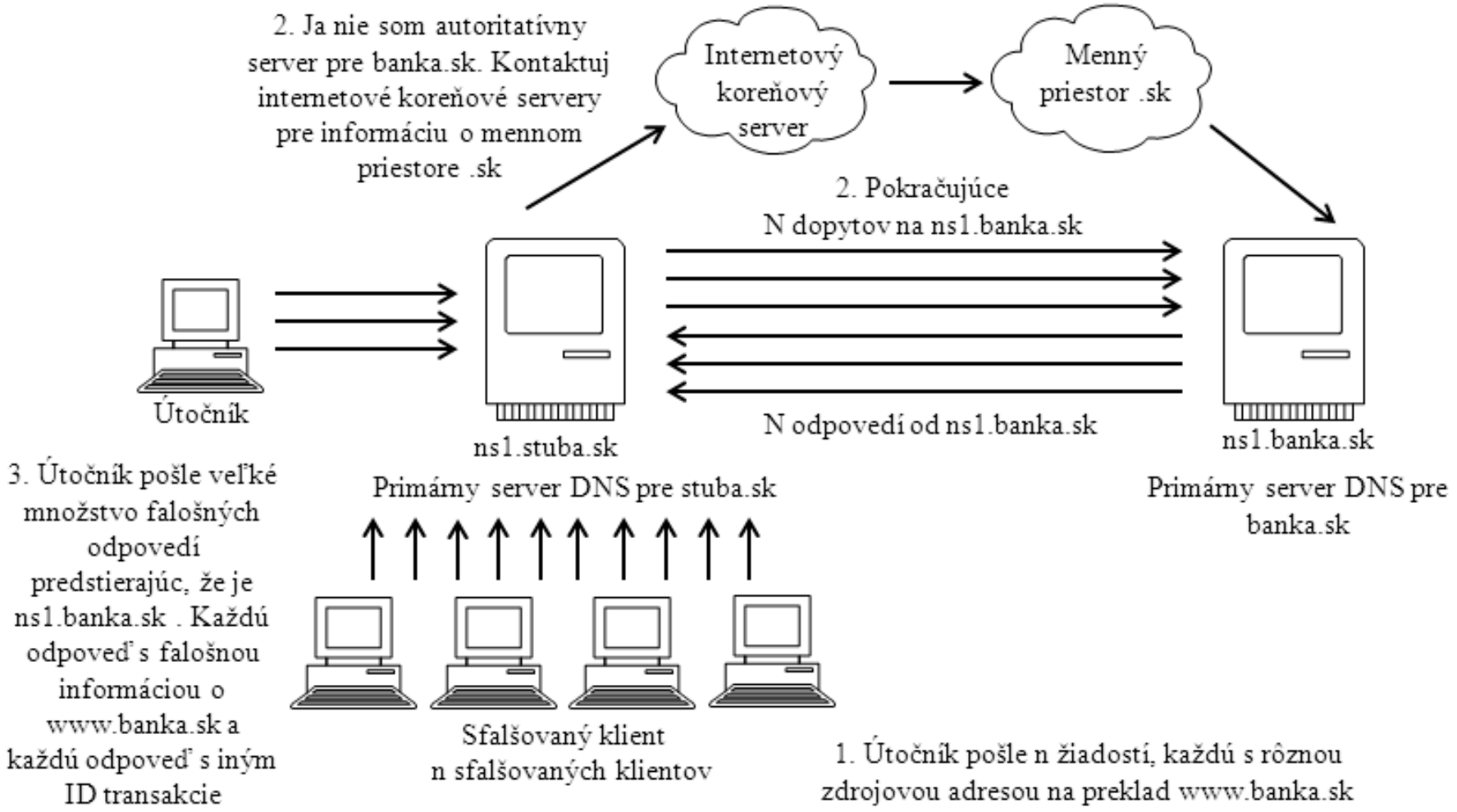
Útoky na DNS – cache poisoning



Útoky na DNS – cache poisoning

- Je potrebné rozlišovať ID medzi transakciou medzi klientom a menným serverom a medzi transakciou medzi mennými servermi. V skutočnosti ide o dve rôzne DNS transakcie, teda **ID transakcií bude samozrejme rôzne**.
- Vyššie uvedené kroky môžu byť útočníkom zneužité na umiestnenie falošnej informácie do vyrovnávacej pamäti ns1.stuba.sk. Na nasledujúcom obrázku útočník sa snaží správne uhádnuť ID transakcie (2B) použitej pri komunikácii menných serverov.
- Aby to útočník dosiahol, urobí toto:
  1. Pošle veľké množstvo žiadostí mennému serveru ns1.stuba.sk o preklad, každá žiadosť má inú falošnú zdrojovú adresu IP, mena domény www.stuba.sk na adresu IP. Dôvodom na poslatie veľkého počtu žiadostí je to, že každej žiadosti bude pridelené jedinečné ID transakcie a aj keď všetky žiadosti sú pre to isté meno domény, každá žiadosť bude spracovávaná nezávisle.
  2. Menný server ns1.stuba.sk pošle každú z týchto žiadostí na ďalšie servery DNS a eventuálne ns1.banka.sk. To znamená, že menný server ns1.stuba.sk očakáva veľké množstvo odpovedí od menného servera ns1.banka.sk.
  3. Útočník využije tento čakací interval na bombardovanie servera ns1.stuba.sk falošnými odpoveďami od servera ns1.banka.sk udávajúcimi, že doméne www.banka.sk odpovedá adresa IP, ktorá je pod kontrolou útočníka (falošná adresa, falošná infomácia). Každá falošná odpoveď má iné ID transakcie. Útočník dúfa, že uhádne správne ID transakcie, t.j. také ako bolo použité mennými servermi.
- Ak je útočník úspešný, **bude falošná informácia** (falošná adresa IP) **uložená do vyrovnávacej pamäti servera ns1.stuba.sk**. Treba poznamenať, že tento útok je viac menej útokom na menný server, ktorý má dopad na klienta používajúceho cieľový menný server s falošnými informáciami.

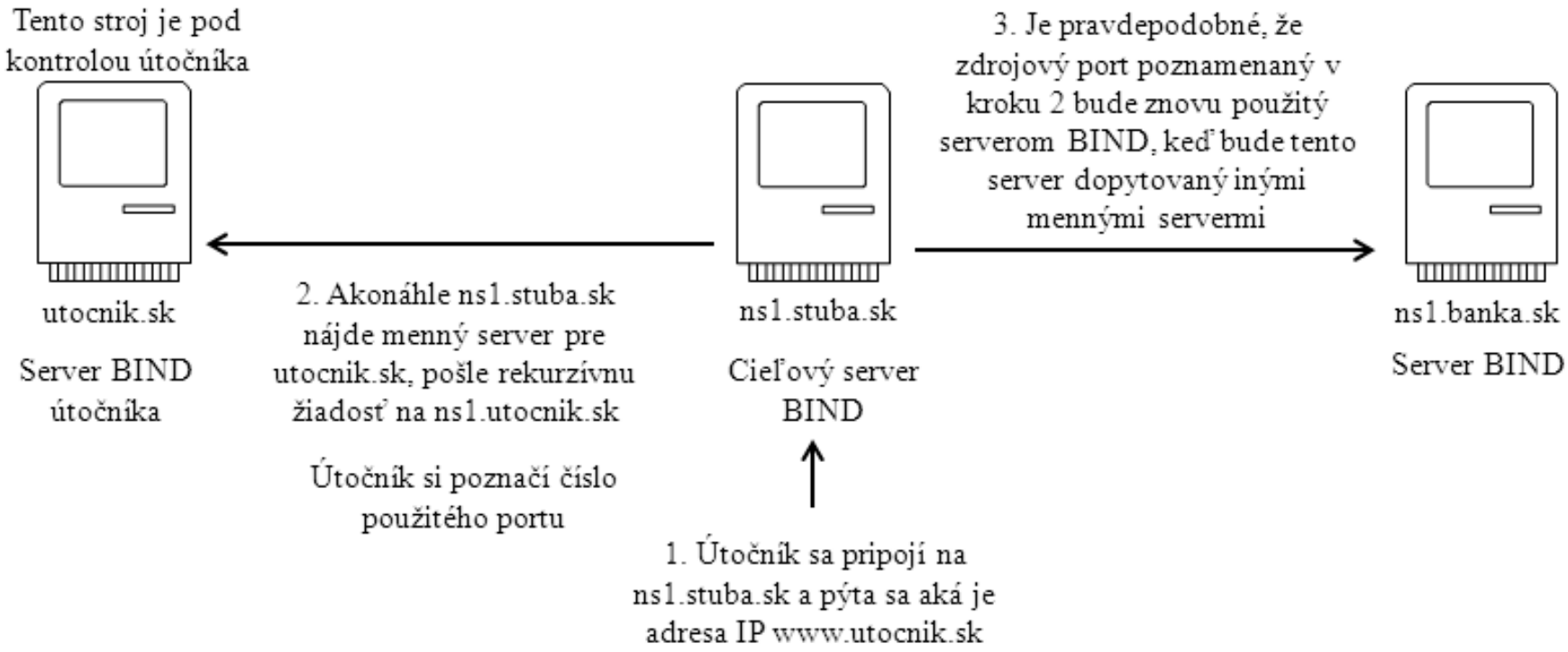
Útoky na DNS – cache poisoning



Útoky na DNS – cache poisoning

- Teraz sa opäť vráťme k trom autentizačným položkám dopytu a odpovede, t.j. ID transakcie, zdrojovej adrese IP a číslu zdrojového portu. Zistenie zdrojovej IP adresy menného servera je priamočiare, pretože poznáme adresu IP menného servera, ktorému klient posiela dopyty. Zistenie **čísla zdrojového portu je obťažnejšie**.
- Častejšie áno ako nie, softvér BIND znovupoužíva **to isté číslo zdrojového portu na dopyty toho istého klienta, t.j. menného servera BIND**. To znamená, že ak útočník má pod kontrolou nejaký BIND autoritatívny name server (ns1.utocnik.sk), môže ako prvé zadať dopyt na cieľový menný server na preklad doménového mena z útočníkovej domény (napr. www.utocnik.sk) a keď príde paket s rekurzívnym dopytom na ns1.utocnik.sk, môže útočník zistiť číslo zdrojového portu na cieľovom mennom serveri.
- Je pravdepodobné, že bude použité to isté číslo zdrojového portu aj keď obeť pošle dopyty pre doménu, ktorá bude unesená (hijacked). Odchytávaním výstupov troch po sebe idúcich dopytov pre rôzne doménové mená bolo napríklad zistené:
  - 172.16.1.2.22343 > 128.1.4.100.53
  - 172.16.1.2.22343 > 23.55.3.56.53
  - 172.16.1.2.22343 > 42.14.212.5.53
- Pri dopytoch na tri rôzne menné servery všetky tri dopyty použili číslo zdrojového portu 22343.
- Zistenie zdrojového číslo portu je dokumentované na nasledujúcom obrázku.

Útoky na DNS – cache poisoning



- BIND v4 a 8 používa sekvenčné prideľovanie ID pre transakcie. Zo znamená, že útočník môže ľahko nájsť aktuálne ID jednoducho vykonaním dopytu na server a zistením čísla ID a znalosťou, že nasledujúci dopyt BIND na ďalší name server sa vykoná s ID+1.
- BIND v9 prideľuje transakciám čísla ID náhodne a neposiela viacnásobné rekurzívne dopyty pre tie isté mená domén.

Narodeninový útok - narodeninový paradox

- Príklad (tréningový): Majme **hašovaciu funkciu H(x)**, pričom hašovacia funkcia nadobúda n rôznych hodnôt (s rovnakou pravdepodobnosťou). Koľko rôznych vstupov $x_1,x_2,x_3,.. x_k$ treba vybrať, aby pravdepodobnosť P(k), že hašovacie hodnoty H(h) a $H(x_i)$ sú rovnaké aspoň pre jedno x_i, bola 0,5?
- Riešenie príkladu:
  - Počet rôznych hašovacích hodnôt je n, preto pravdepodobnosť, že pre dané x_i bude $H(x_i)=H(h)$, je 1/n. Pravdepodobnosť, že nie sú si rovné je (1-1/n).
  - Ak vyberieme k rôznych vstupov $x_1,x_2,x_3,.. x_k$ potom pravdepodobnosť, že $H(x_i)\neq H(h)$ pre každé i=1,… k, je $(1-1/n)^k$.
  - Pravdepodobnosť P(k), že aspoň jedna hašovacia hodnota $H(x_i)$ je rovná H(h), je $P(k)=1- (1-1/n)^k$.
  - Pre veľké n (a teda malé 1/n) je možné výraz $1- (1-1/n)^k$ aproximovať na (1-1+k/n)=k/n. (Použijeme binomickú vetu a zanedbáme členy s mocninami k^2 a viac.) Teda P(k)=k/n
  - Podľa zadania je P(k)=1/2, to znamená, že riešime rovnicu ½=k/n, teda k=n/2.
- Odpoveď: Je potrebné vybrať n/2 vstupov, aby pravdepodobnosť, že hašovacie hodnoty H(h) a $H(x_i)$ sú rovnaké aspoň pre jedno x_i, bola 0,5.
- Ilustrácia: Nech napríklad $n=2^{16}$, potom k danej hašovacej hodnote H(h) je potrebné náhodne vybrať $k=n/2=2^{15}$ vstupov $x_1,x_2,x_3,.. x_k$, aby pravdepodobnosť, že aspoň pre jedno x_i je $H(x_i)=H(h)$, bola 0,5.

Narodeninový útok - narodeninový paradox

- Príklad (narodeninový paradox): Aká je prevdepodobnosť, že v skupine **k** ľudí budú **aspoň dvaja s rovnakým dňom narodenia v roku** (neuvažujme priestupný rok)?
- Riešenie príkladu:
  - Najprv si odvodíme pravdepodobnosť, že tam nebudú žiadni dvaja s rovnakým dňom narodenia
  - Pravdepodobnosť, že dvaja ľudia sa nenarodili v ten istý deň v roku je Q(365,2)= (počet možných výberov)/(počet všetkých výberov)=(365.(365-1))/(365.365).
  - Pravdepodobnosť, že dvaja ľudia z k ľudí sa nenarodili v ten istý deň v roku je Q(365,k)= (počet možných výberov)/(počet všetkých výberov)=(365.(365-1).(365-2)…(365-k+1))/(365^k). Teda Q(365,k)=(365!)/((365-k)!.365^k)
  - Pravdepodobnosť, že v skupine k ľudí sú aspoň dvaja s rovnakým dňom narodenia v roku potom je P(365,k)=1-Q(365,k).
  - Dá sa ľahko zistiť, že pre P(365,k)=1/2 je k=23. Môže to byť prekvapujúce, ale si treba uvedomiť, že pri k=23 je 23.(23-1))/2 dvojíc, čo je 253 rôznych dvojíc. Odtiaľ je taká vysoká pravdepodobnosť.
- Ak by sme numericky vyčíslili vyššie uvedený výraz, tak pre počet **k** ľudí v skupine dostaneme nižšie uvedené pravdepodobnosti **P(365,2)** narodenia dvoch členov skupiny ten istý deň v roku.

k	2	9	16	23	30	37	44	65	79
P(365,2)	0,0027	0,0946	0,2836	0,5073	0,7063	0,8487	0,9329	0,9977	0,9999

Narodeninový útok - narodeninový paradox

- Príklad (všeobecný prípad duplikácie): Majme prirodzené čísla 1,… n s rovnakou pravdepodobnosťou rozloženia a výber **k** inštancií (k≤n) náhodných premenných. Aká je pravdepodobnosť P(n,k), že medzi výberom k inštancií sú **aspoň dve čísla rovnaké**?
- Riešenie príkladu:
  - Podľa analógie narodeninového paradoxu je P(n,k)=1-Q(n,k)=)=1-(n!)/((n-k)!.n^k).
  - Na úpravu vyššie uvedeného výrazu použijeme túto užitočnú nerovnosť $e^{-x} = S_{i=1}^{æ} (-1)^i . x^i/i!$ (Taylorov rozvoj funkcie). Pre malé x je e^{-x}≥1-x alebo môžeme použiť e^{-x}~1–x.
  - Výraz P(n,k) možno ďalej upraviť na 1-((n-1)/n).(n-2)/n).(n-3)/n)…(n-k+1)/n), čo ďalej možno upraviť na 1-((1-1/n).(1-2/n).(1-3/n)…(1-(k+1)/n).
  - Podľa vyššie uvedenej užitočnej nerovnosti (e^{-x}~1–x) možno P(n,k) potom prepísať na tvar P(n,k)~1- $e^{-1/n}$. $e^{-2/n}$. $e^{-3/n}$… $e^{-(k-1)/n}$ n=1-$e^{-k.(k-1)/(2.n)}$
- Odpoveď: Hľadaná pravdepodobnosť je vyjadrená výrazom uvedeným vyššie.
- Ilustrácia:
  - Aká je hodnota pre P(n,k)=1/2? Dosadením do výrazu P(n,k)=1/2=1-$e^{-k.(k-1)/(2.n)}$ dostaneme ln2=k.(k-1)/(2.n). Ak sa nahradí k.(k-1)~k^2, potom k=$(2.n.ln2)^{1/2}$ =1,18.$n^{1/2}$. Pre prípad n=365 je k=1,18.$365^{1/2}$=1,18.19,105=22,54.
  - Aká je hodnota pre P(n,k)=9/10? Analogickým postupom ako vyššie dostaneme k=$(2.n.ln10)^{1/2}$ =2,15.$n^{1/2}$. Pre prípad n=2^{16} (napr. ID transakcie v DNS) dostaneme k=2,15.$(2^{16})^{1/2}$=2,15.2^8=2,15.256=550,4!!! **To je pre hackerov veľmi sľubný výsledok?!**
    - Pre k=650 je pravdepodobnosť 0,9604, pre k=750 je pravdepodobnosť 0,9865.

Bezpečné DNS

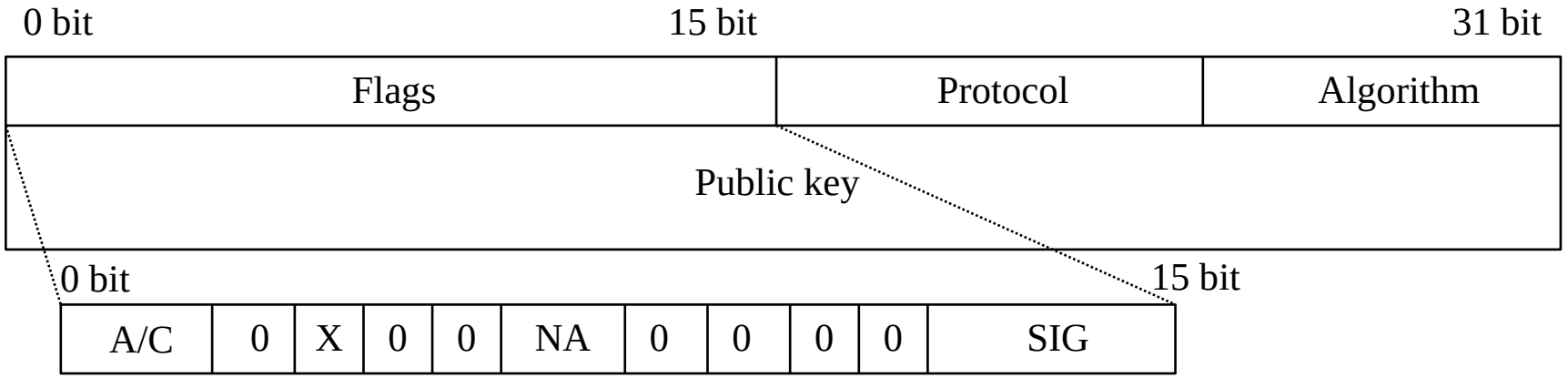
- Základnými mechanizmami zabezpečenia DNS sú mechanizmy **DNSsec** a **TSIG**. (Aktuálna verzia menného servera BIND v9 podporuje prevažnú väčšinu týchto zabezpečovacích protokolov.)
- DNSsec je rozšírenie DNS špecifikované v RFC2535:
  - Rieši základné otázky zabezpečenia DNS.
  - V strome tvorenom doménami môžeme **od určitej domény nižšie** vykonať zabezpečenie pomocou DNSsec. (Ideálne by bolo, keby zabezpečenie začínalo na root name serveroch a pokračovalo celým stromom DNS až k menám jednotlivých počítačov, poštových proxy, či iných mien vedených v DNS.)
    - Prvý problém, ktorý si musíme pre DNSsec uvedomiť je to, že z prevádzkového hľadiska nie je priestor mien DNS členený na domény ale na zóny. Pretože bezpečnosť budú zabezpečovať konkrétne menné servery spravované konkrétnymi administrátormi, budú príslušné verejné kľúče platné v rámci zóny a nie všeobecne v rámci celej domény.
  - DNSsec využíva **asymetrickú kryptografiu** tak, ako ju poznáme z PKI, ale aplikuje sa celkom odlišným spôsobom. Nevyužíva certifikáty, ale verejné kľúče ukladá do viet typu KEY. Pri vkladaní verejných kľúčov do DNS (vety typu KEY) nepriamo dochádza k certifikácii verejných kľúčov, pretože **správca nadradenej domény podpíše verejný kľúč podriadenej domény**.
    - Ak je napríklad pomocou DNSsec zabezpečená doména *stuba.sk* nižšie, uvedieme v DNS pre doménu *stuba.sk* vetu typu KEY a v nej verejný kľúč, ktorého príslušným privátnym kľúčom sú elektronicky podpisované informácie týkajúce sa zóny *stuba.sk*.
    - Ak napríklad zriadime zónu *fiit.stuba.sk*, potom v databáze name servera pre zónu *fiit.stuba.sk* uvedieme ďalšiu vetu typu KEY s verejným kľúčom, ktorý bude slúžiť na verifikáciu údajov tejto subdomény. Ide samozrejme všeobecne o iný verejný kľúč.

DNSsec

- Aby subdoména *fiit.stuba.sk* bola vidieť z Internetu, musí správca zóny *stuba.sk* vykonať delegáciu na zónu *fiit.stuba.sk*. Delegácia znamená, že do databázy pre zónu stuba.sk musia uviesť príslušné vety typu NS delegujúce právomoc smerom nadol.
- Ak sa používa DNSsec, potom sa pri delegácii neuvedú iba vety typu NS a prípadné vety typu A, **ale uvedie sa i veta typu KEY s verejným kľúčom zóny**. Správca zóny zónu elektronicky podpíše a podpis umiestni do vety typu SIG. Uvedenie vety typu KEY v zóne, z ktorej sa deleguje právomoc nižšie , je tak obdobou certifikácie verejného kľúča. Ak potom získame autorizovanú odpoveď obsahujúcu verejný kľúč zóny nižšej úrovne, je verejný kľúč pre uvedenú zónu nižšej úrovne dôveryhodný.
- Otázkou je, ako distribuovať verejné kľúče najvyššej domény, pretože tie nie sú certifikované žiadnym kľúčom vyššej úrovne. Riešenie je jednoduché – **klientom sa ručne napíšu do konfiguračného súboru resolvera.**

Veta typu KEY

- Obsahuje verejný kľúč udržovaný v systému DNS.
- Má špecifické pole RData. Ukážeme ho na obrázku. Ostatné polia sú analogické ako v ostatných RR vetách.

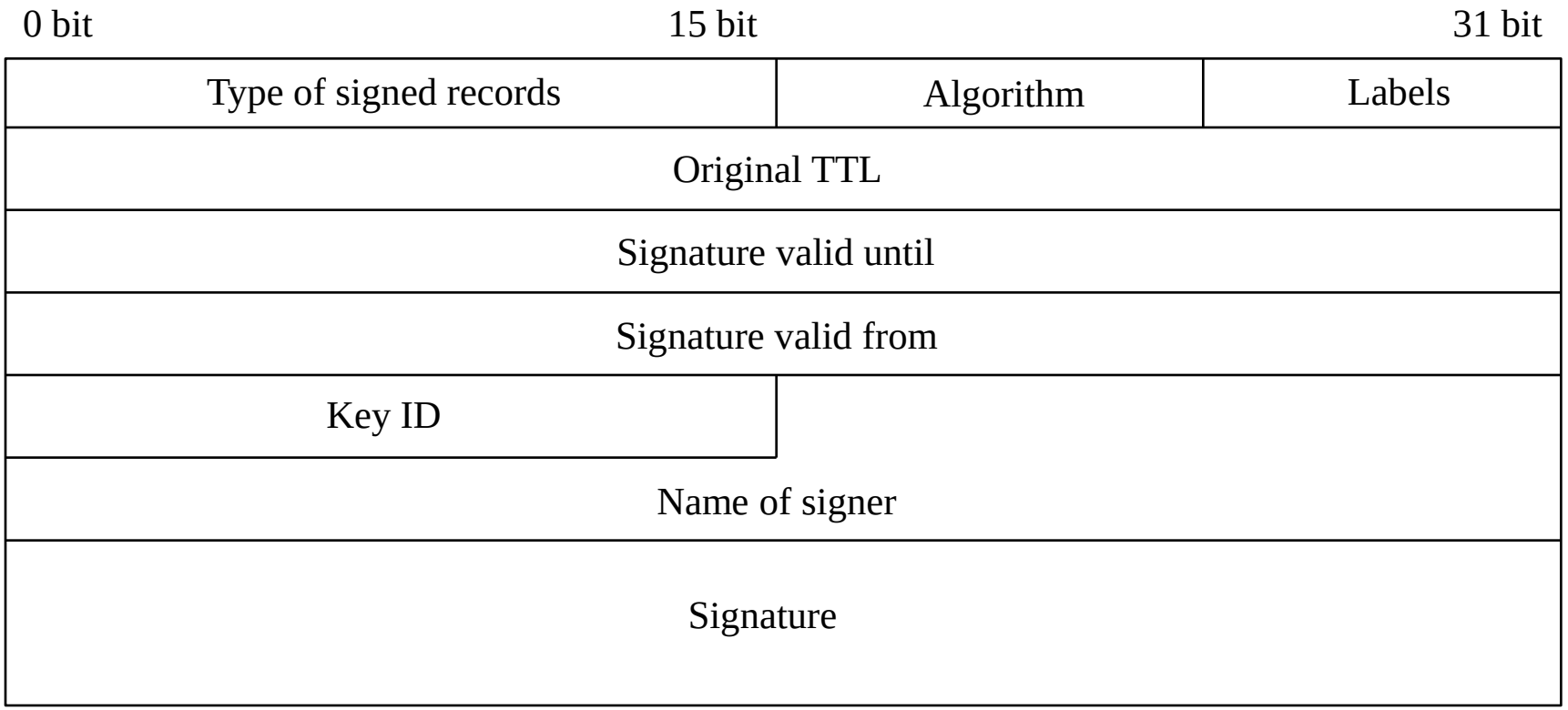


DNSsec

- Pole Flags, bity **A/C**:
  - 10 – kľúč je zakázané použiť na autentizáciu
  - 01 – kľúč je zakázané používať na šifrovanie
  - 11 – kľúč je možné použiť na autentizáciu a aj na šifrovanie
  - 00 – veta typu KEY neobsahuje žiadny kľúč.
- Pole Flags, bit **X** indikuje, že veta typu KEY obsahuje rozšírené pole Flags (ďalšie 2B za poľom Algorithm)
- Pole Flags, bity **NA** určuje, na aký účel je kľúč určený:
  - 00 – veta obsahuje kľúč používateľa, čo sa dá použiť napr. na autentizáciu v aplikačných protokoloch (napríklad telnet, ftp apod.)
  - 01 – veta obsahuje kľúč zóny. T.j. kľúč, ktorým budú primárnym DNS serverom elektronicky podpisované údaje zóny.
  - 11 – veta obsahuje kľúč zóny pre iné účely. (napríklad na zabezpečenie smerovania, správu času – protokol NTP, apod).
- Položka **Protocol** obsahuje protokol, pre ktorý je kľúč určený:
  - 1 – rezervované pre protokol TLS
  - 2 – rezervované pre pelektronickú poštu
  - 3 – DNSsec
  - 4 – rezervované pre IPsec
- Položka **Algorithm** obsahuje kryptografický algoritmus, pre ktorý je kľúč určený: 1 – RSA/ MD5, 2 – Diffie-Hellman, 3 – DSA, 4 - ECC
- Položka **SIG** slúži na označenie kľúča, ktorý je možné použiť pre DNS UPDATE.

DNSsec

- **Veta typu SIG**
  - Slúži na uloženie elektronického podpisu do DNS. To znamená, že DNSsec pomocou viet typu SIG autentizuje svoje údaje.Obsahuje verejný kľúč udržovaný v systéme DNS.
  - Pole RData vety typu SIG je na obrázku.

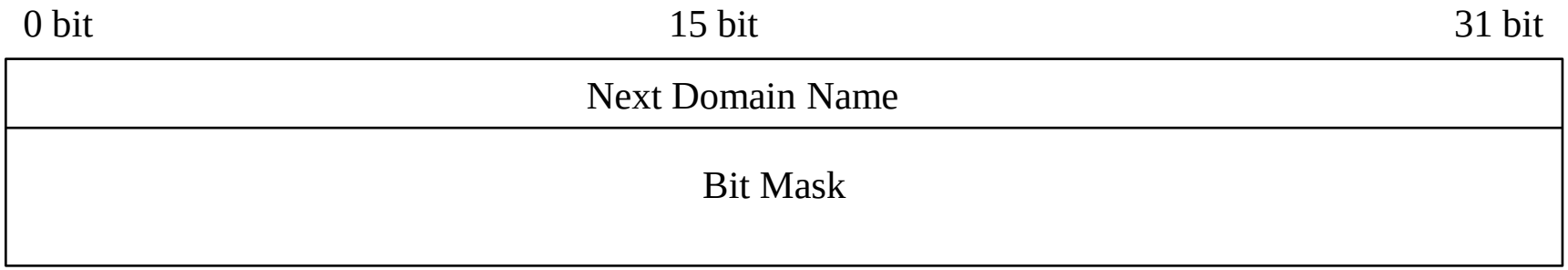


DNSsec

- Pole **Type of signed records** obsahuje číslo typu podpisovaných viet.
- Pole **Algorithm** má ten istý význam ako vo vete typu KEY.
- Pole **Labels** obsahuje počet reťazcov, z ktorých sa skladá DNS meno, napríklad pre DNS meno „.“ je Labels=0, pre sk. je Labels=1, pre stuba.sk je je Labels=2.
- Pole **Original TTL** obsahuje pôvodnú hodnotu TTL vety typu RR. Ťažkosť je totiž v tom, že hodnoty poľa TTL sa v cache jednotlivých DNS serverov automaticky znižujú. Ak je však veta RR elektronicky podpísaná, je nevyhnutné pole TTL udržovať dvakrát: raz pôvodné podpísané, ktoré nie je možné meniť (to by spôsobilo neplatnosť podpisu) a druhé aktuálne.
- Elektronický podpis je platný od **Signature valid from** do **Signatute valid until.**
- Pole **Key ID** obsahuje identifikáciu kľúča, ktorý bol použitý na elektronický podpis. Toto pole je užitočné najmä v prípade, keď na rovnaký účel máme viacero kľúčov. Viacero kľúčov môžeme v DNS mať napríklad preto, že potrebujeme používať viacero kryptografických algoritmov súčasne. Ako identifikácia kľúča sa napríklad pre algoritmus RSA/MD5 berú najnižšie 2B z modulu verejného kľúča.
- Položka **Name of signer** obsahuje meno toho, kto vytvoril podpis.
- Položka **Signature** obsahuje vlastný elektronický podpis.

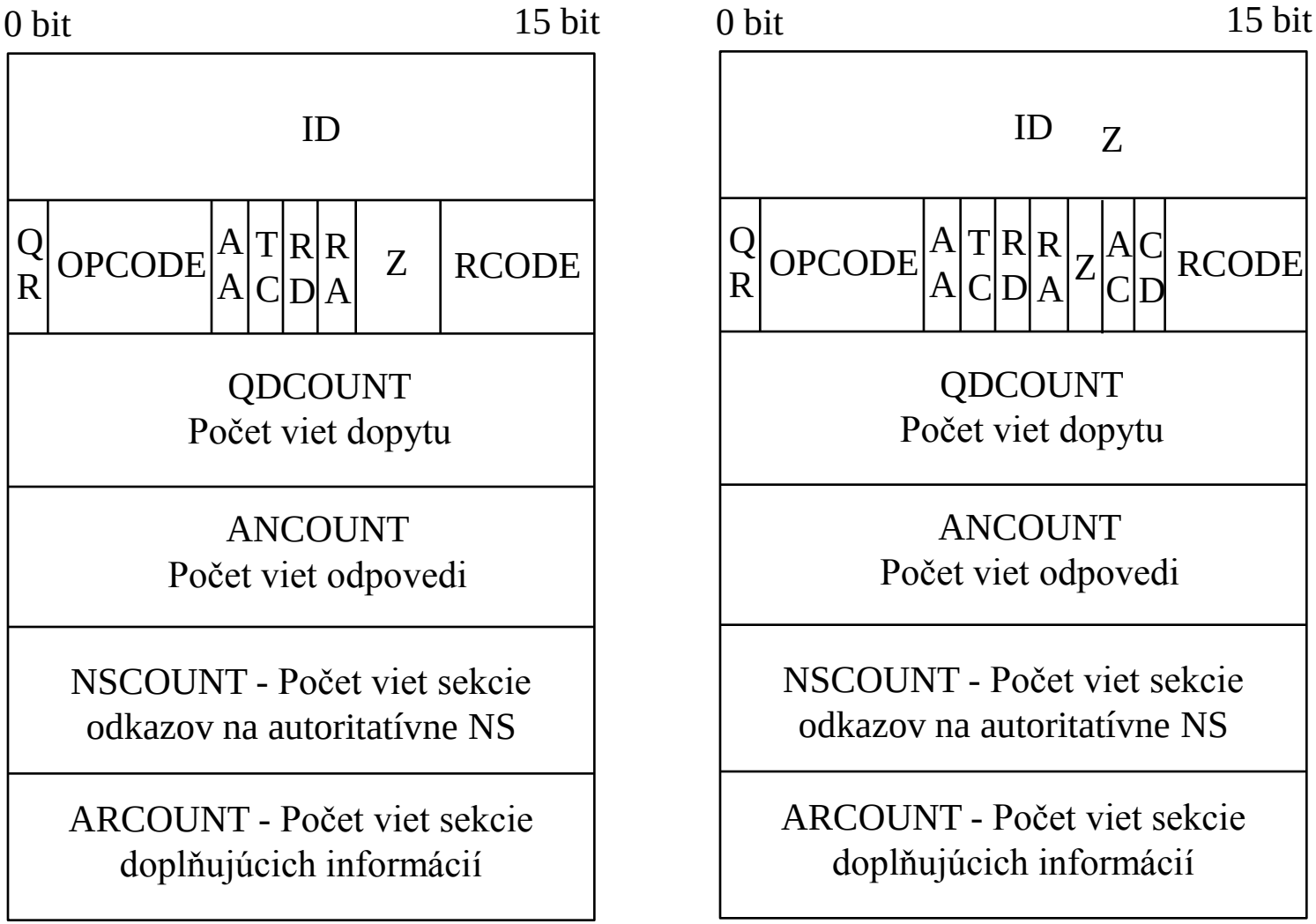
- **Veta typu NXT**
  - V DNS nie sú jednotlivé vety zviazané do postupnosti ako sú za sebou. Tento nedostatok odstraňuje veta typu NXT. Pomocou tejto vety sa špecifikuje, aký objekt nasleduje v DNS za aktuálnym objektom.
  - Majme hypotetický príklad DNS záznamu:
    - fiit.stuba.sk IN SOA
    - IN NS ns1.stuba.sk
    - ftp IN A 10.1.1.1
    - pocitac IN A 10.1.1.2
  - Potom útočník môže napríklad pri prenose zóny vypustiť vetu začínajúcu „pocitac IN A …“ a tým spôsobiť, že uzol pocitac.fiit.stuba.sk bude nedostupný.
  - Pomocou viet typu NXT je možné oznámiť, aká veta vždy nasleduje (riadky sú očíslované)
    - 1 fiit.stuba.sk IN SOA …
    - 2 IN NS ns1.stuba.sk
    - 3 IN NXT ftp.fiit.stuba.sk NS SOA NXT
    - 4 ftp IN A 10.1.1.1
    - 5 IN NXT pocitac.fiit.stuba.sk A NXT
    - 6 pocitac IN A 10.1.1.2
    - 7 IN NXT fiit.stuba.sk A NXT
  - V uvedenom príklade tak za úvodnými vetami SOA a NS (riadok 1 a 2) nasleduje záznam pre ftp.fiit.stuba.sk . Táto väzba je opísaná vetou typu NXT v riadku 3 (najprv v zozname je napísaný nasledujúci záznam a potom predchádzajúce. Posledný záznam vety typu NXT ukazuje na začiatok zónového súboru.

- **Veta typu NXT**
  - Pole RData vety typu NXT je na obrázku a skladá sa z dvoch položiek:
    - Prvá položka obsahuje meno DNS **Next Domain Name**
    - Druhá položka obsahuje bitovú masku **Bit Mask** špecifikujúcu, aké typy viet sú použité v opise aktuálneho objektu databázy. Poradové číslo bitu masky odpovedá typu vety. Bit pre vetu NXT je nastavený vždy. Jednotlivé typy viet majú pridelené čísla napríklad veta typu A má číslo 1, veta NS má 2, veta CNAME má 5, veta SOA má 6, veta MX má 15, veta SIG má 24, veta KEY má 25, veta NXT má 30, atď. To znamená, že pokiaľ objekt má vety typu NS a SOA, potom sú v maske nastavené bity 2 (NS), 6 (SOA) a 30 (NXT).



DNSsec – protokol DNS

- **Protokol DNS**
  - Na obrázku vľavo je záhlavie paketu DNS QUERY. Toto záhlavie obsahuje tri rezervné bity Z (nastavené na 0). Rozšírenie DNSsec využije dva bity AD a CD z tejto rezervy.



DNSsec – protokol DNS

- Bit AD (Authentic Data) v odpovedi name servera indikuje, že dáta uvedené v sekcii odpovedi a v sekcii autoritatívne name servery sú serverom autentizované.
- Bit CD (Checking Disabled) indikuje, že pre resolver sú akceptovateľné i neautentizované dáta.

- Ukázali sme ako elektronicky podpisovať vety RR pomocou viet typu SIG. Avšak tento mechanizmus **nezabezpečuje odpoveď DNS servera ako celku**, t.j. nezabezpečuje transakciu. Útočník môže ľahko zmeniť bity v záhlaví DNS paketu a z niektorých sekcií vypustiť nejaké vety RR alebo ich môže prehodiť medzi sekciami. Pritom môže vypustiť či prehodiť vety vrátane viet typu SIG, t.j. vrátane elektronického podpisu.
- Riešením je pridanie **špeciálnej vety typu SIG** na koniec odpovedi servera DNS. Táto veta typu SIG elektronicky podpíše odpoveď servera vrátane sekcie dopytu (dopytu resolvera). Nevýhodou podpisovania odpovedí, t.j. pridanie zmienených špeciálnych viet typu SIG na koniec sekcie doplňujúcich informácií je nepríjemné v tom, že je nevyhnutné mať stále (on-line) k dispozícii privátny kľúč, ktorým sa vykonáva elektronický podpis. Odpovedi DNS severa sú totiž tak variabilné, že nie je možné mať odpovede dopredu predpripravené a podpísané.
- Pri podpisovaní veľkých zón je zrejmé, že ide o časovo náročnú operáciu. Takže sa predpokladá, že privátny kľúč bude držaný v špeciálnom zariadení (obdoba HSM). Administrátor zóny podpíše zónu na tomto zariadení a následne podpísanú zónu prenesie prenesie na name server. Týmto opatrením sa zvýši bezpečnosť privátneho kľúča.

TSIG (Transaction SIGnatures)

- DNSsec opísaný skorej má niekoľko nevýhod:
  - Asymetrická kryptografia je pomerne náročná
  - Týmto mechanizmom je ťažko vykonávať dynamický DNS UPDATE
- **TSIG** je alternatívny mechanizmus **bezpečného DNS** špecifikovaný v RFC2845.
  - TSIG je určený na autorizáciu komunikácie medzi dvomi stranami. Obe strany si navzájomm vymenia spoločné tajomstvo. Prenášané údaje medzi stranami potom autorizujú algoritmom HMAC-MD5. To znamená, že prenášané údaje zreťazia so spoločným tajopmstvom a z výsledku sa vypočíta kontrolný súčet (hašovacia hodnota) algoritmom HMAC-MD5.
  - Tento kryptografický kontrolný súčet je prenášaný vo vete typu TSIG. Táto veta je vždy znovu vytváraná pre každé prenášané dáta, preto ju nemá zmysel uchovávať v databáze.
- Pre správnu činnosť TSIG je nevyhnutná výmena spoločného tajomstva.
  - Dynamicky je možné vymeniť spoločné tajomstvo pomocou Diffie-Hellmanovho (D-H) algoritmu. **Algoritmus TKEY** špecifikovaný v RFC2930 využíva túto možnosť. Klient za účelom výmeny D-H parametrov pošle dopyt (veta typu TKEY) obsahujúci v sekcii dodatočných informácií vetu typu TKEY s príslušným verejným D-H číslom. Server vo svojej odpovedi uvedie svoje verejné D-H číslo. Na základe oboch verejných D-H čísiel si obe strany vypočítajú spoločné tajomstvo.
  - Iný voliteľne podporovaný mechanizmus je použitie asymetrického šifrovacieho algoritmu. Resolver v tomto prípade pošle name serveru dopyt, v ktorom ho požiada, aby server vygeneroval spoločné tajomstvo. Súčasťou dopytu je i veta typu KEY s verejným kľúčom klienta. Server vygeneruje spoločné tajomstvo a zašle ho klientovi zašifrované jeho verejným kľúčom. Je možné, aby tak urobil aj klient a poslal spoločné tajomstvo serveru zašifrované verejným kľúčam servera.

Uloženie certifikátov do DNS

- RFC2538 špecifikuje uloženie certifikátov a CRL do DNS.
  - Certifikáty a CRL sa ukladajú do viet typu CERT.
  - Je podporované ukladanie certifikátov a CRL podľa X.509, certifikátov PGP a certifikátov SPKI.
  - Je treba zdôrazniť, že tu DNS slúži na distribúciu uvedených certifikátov a CRL.
  - Vety typu CRL neslúžia na zebezpečenie DNS.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Otázky a diskusia

Ďakujem za pozornosť

KPMG
cutting through complexity™