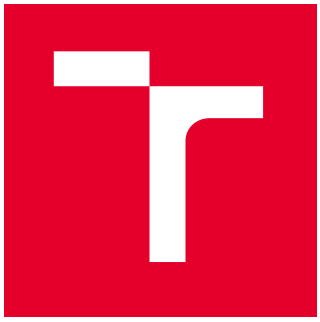


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

NOVÉ SCÉNÁŘE S PROGRAMEM WIRESHARK V KOMUNIKAČNÍCH TECHNOLOGIÍCH

NEW SCENARIOS WITH WIRESHARK IN COMMUNICATION TECHNOLOGIES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Jan Šíma**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Jeřábek, Ph.D.**
SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Jan Šíma ***ID:*** 203717

Ročník: 2 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Nové scénáře s programem Wireshark v komunikačních technologiích

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte fungování a možnosti využití programu Wireshark a také stávající laboratorní scénáře předmětů Komunikační technologie a Pokročilé komunikační technologie. Dále se zaměřte na problematiku základních komunikačních protokolů používaných v paketových sítích. Prostudujte dostupné laboratorní scénáře zaměřené na komunikační protokoly a následně navrhněte a popište čtyři vlastní komplexní laboratorní scénáře. Cílem prvního scénáře bude seznámit vhodným způsobem studenty s problematikou překladu síťových adres (NAT, resp. NAPT) s využitím programu Wireshark. U druhého a třetího scénáře půjde především o protokol DNS, popř. další příbuzné protokoly, a využití nástroje Klient DNS. V posledním scénáři se pak zaměřte na některé další téma z oblasti komunikačních technik. Výstupem práce budou čtyři kompletní scénáře včetně podrobných návodů pro studenty v českém jazyce, prokonzultovaných s vedoucím práce, předpřipravených výchozích situací či souborů, doplňujících úkolů pro studenty a vzorového řešení. Předpokládá se, že délka realizace jedné úlohy bude pro studenta přibližně 2 hodiny času.

DOPORUČENÁ LITERATURA:

[1] KUROSE, J. F., ROSS, K. W., Computer networking: a top-down approach. 7th global ed. Essex: Pearson, 2017, 852 s. ISBN 978-1-292-15359-9.

[2] JEŘÁBEK, J. Pokročilé komunikační techniky. Skriptum FEKT Vysoké učení technické v Brně, 2021. s. 1-180.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 24.5.2022

Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Cílem této diplomové práce je navrhnout a vytvořit scénáře, které budou sloužit studentům jako podklady pro jednotlivá laboratorní cvičení v rámci předmětu, který se věnuje komunikačním technologiím. K těmto scénářům je také potřeba doplnit teoretický úvod, který studentům přiblíží konkrétní probírané téma. Vytvořené scénáře obsahují doplňující úkoly, které jsou určeny pro samostatnou práci studentů. Pro vyučující jsou k těmto úkolům připraveny samostatné soubory, které shrnují jejich správné řešení.
Práce začíná teoretickým popisem použitých síťových protokolů. Nejdříve je popsán model TCP/IP a následně pod něj spadající protokoly, a to hlavně ty, které se vyskytují ve vícero vytvořených scénářích, jako například protokoly TCP, UDP nebo DNS. Jsou zde také popsány programy a aplikace, které jsou využity při vypracovávání scénářů. Mezi ty hlavní patří zejména program Wireshark a aplikace Klient DNS. Následuje kapitola, která se zabývá návrhy jednotlivých scénářů. Ty nastíní, čemu se dané scénáře věnují, co je jejich cílem a obsahují také soupis využitých protokolů, programů a utilit, které jsou v dané úloze využity. První scénář obsahuje úkoly založené na práci s překladem adres NAT a jeho současné spolupráci s dalšími protokoly, jako například již zmíněné TCP či UDP a nebo také protokoly ICMP a FTP. Druhý vytvořený scénář pracuje a analyzuje protokoly, které se snaží zabezpečit překlad doménových jmen. Konkrétně jde o rozšíření DNSSEC a o protokol DoH. Třetí scénář se zaměřuje podrobněji na funkci rekurzivních serverů při DNS a DNSSEC komunikaci. Poslední vytvořený scénář vysvětluje plánování a přidělování adresního prostoru a soustřeďuje se na směrovací tabulky a NAT překlad.

KLÍČOVÁ SLOVA

DNS, DNSSEC, DoH, IPv4, IPv6, NAT, přidělování adresního prostoru, směrovací tabulka, TCP/IP, Wireshark

ABSTRACT

The aim of this thesis is to design and create scenarios that will be used by students as a basis for laboratory exercises in a course that focuses on communication technologies. These scenarios also need to be accompanied by a theoretical introduction that will introduce the topic under discussion to the students. The created scenarios contain additional tasks that are intended for individual work of students. For the teachers there are prepared separate files summarising the correct solution of these tasks.

This thesis starts with a theoretical description of used network protocols. First the TCP/IP model is described and then the protocols that belong to this model. The main ones are those that occur in multiple created scenarios, such as TCP, UDP or DNS. The programs and applications that are used in the scenarios are also described. The main program is Wireshark and also the application called Klient DNS. This is followed by a chapter that deals with the drafting of each scenario. The drafts outline what the scenarios goals are. The drafts also list the protocols, programs, and utilities that are used in the exercise. The first scenario contains tasks based on Network Address Translation and its interaction with other protocols, such as the previously mentioned TCP, UDP, and also ICMP and FTP protocols. The second created scenario is dealing and analyzing protocols that try to secure the translation of domain names. Specifically, the DNSSEC extension and also the DoH protocol. The third scenario is focusing more in detail at the function of recursive servers in DNS and DNSSEC communication. The last created scenario explains the planning and allocation of the address space, where again the NAT topic is encountered and the function of routing tables is also explained here.

KEYWORDS

DNS, DNSSEC, DoH, IPv4, IPv6, NAT, address space allocation, routing table, TCP/IP, Wireshark

ŠÍMA, Jan. *Nové scénáře s programem Wireshark v komunikačních technologiích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 158 s. Diplomová práce. Vedoucí práce: doc. Ing. Jan Jeřábek, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Jan Šíma
VUT ID autora:	203717
Typ práce:	Diplomová práce
Akademický rok:	2021/22
Téma závěrečné práce:	Nové scénáře s programem Wireshark v komunikačních technologiích

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Janu Jeřábkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod

Tato práce se věnuje oblasti komunikačních protokolů, které spadají do síťového modelu TCP/IP (Transmission Control Protocol/Internet Protocol). Práce se snaží tyto protokoly teoreticky popsat, vysvětlit a následně na jejich základě vytvořit několik samostatných scénářů. Ty by měly být v budoucnu využity v rámci předmětů, které se věnují právě problematice komunikačních protokolů.

Tato oblast prochází neustálým vývojem. Nastupují protokoly úplně nové a stejně jako u aktualizovaných verzí již známých protokolů je potřeba znát základní principy jejich fungování. Kromě nových protokolů vznikají také nové nástroje, které mohou studentům pomoci při představení a objasnění práce komunikačních protokolů. V dnešní době je také důležité dbát na zabezpečení jednotlivých protokolů, kde se zaměřujeme na zajištění a ověření autenticity, integrity a důvěrnosti přenášených dat. Současná orientace na zabezpečení veškerého síťového provozu zvyšuje riziko útoků na některý z nezabezpečených komunikačních kanálů. Je tedy potřeba vědět, které protokoly komunikují zabezpečenou formou a které nikoliv, případně v jaké míře a zda je zabezpečení například autenticity řešeno na všech komunikujících úrovních daného protokolu.

Pro získaní dat, které uživatel požaduje stáhnout, například z webových serverů zadáním několika písmen do adresního řádku, je ve většině případů nutná komplexní komunikace několika na sebe navzájem navazujících síťových protokolů. Pro studenty komunikačních technologií je tedy nutné mít základní povědomí o těchto provázaných procesech, mezi které patří například zjišťování a překlad adres nebo portů, přenos souborů skrze přístup ke vzdáleným serverům a také celková kompatibilita použitých protokolů při současném důrazu na zabezpečenou komunikaci.

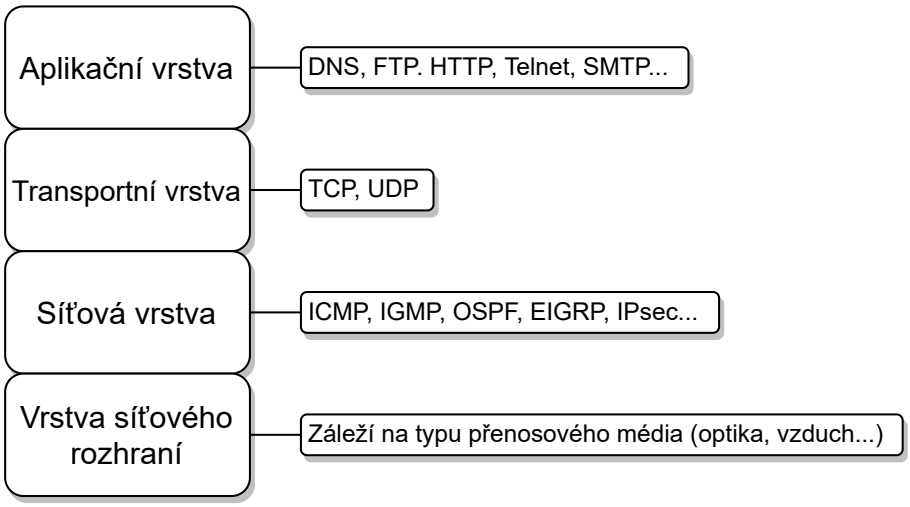
1 Protokoly a funkce využité v této práci

Vytvořené a navržené scénáře pracují s komunikačními protokoly z rodiny modelu TCP/IP. Je tak potřeba vědět, jak dané protokoly fungují, znát jejich výhody, nevýhody, možnosti a důvody použití v určitých situacích. Tomuto teoretickému popisu se věnuje následující kapitola.

1.1 Model TCP/IP

Síťová architektura TCP/IP (Transmission Control Protocol/Internet Protocol) definuje 4 vrstvy, tak jak je ukázáno na obr. 1.1.1. Každá z těchto vrstev má v tomto modelu svůj úkol směrem k přenosu paketů, rychlosti, kvalitě a spolehlivosti síťové komunikace. Dále jsou na každé vrstvě definovány protokoly, které modelu poskytují různé služby. Tyto služby jsou poté využívány v podobě získaných informací a jsou předávány sousedním vrstvám modelu.

Každý z protokolů této síťové architektury je detailně popsán ve vlastním RFC (Request for Comments) dokumentu. Můžeme zde najít například schémata paketů, podrobně popsanou funkčnost protokolu, popis jednotlivých bitů, položek v záhlaví atd. [1]

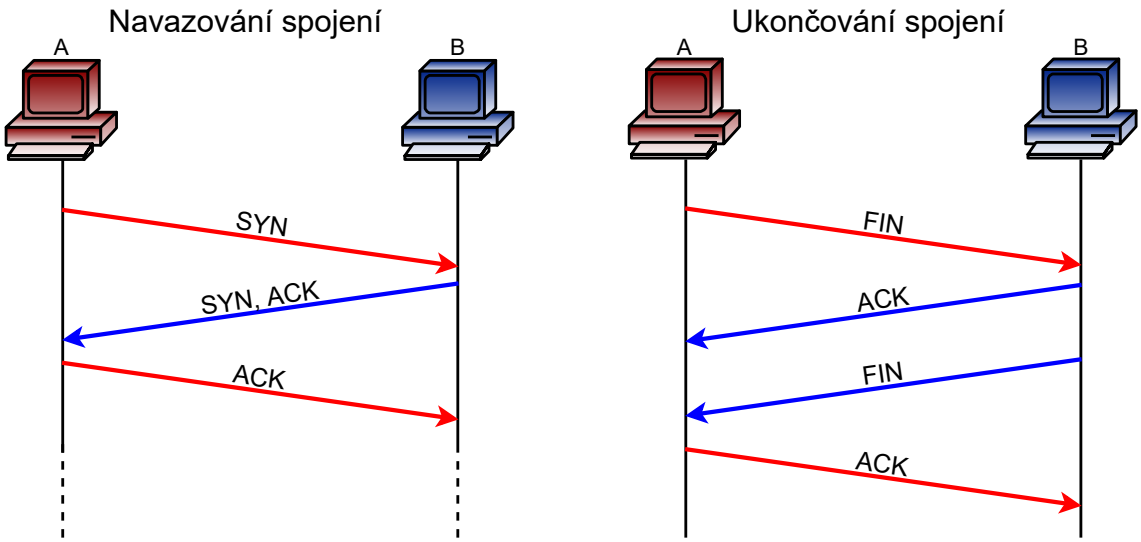


Obr. 1.1.1: Čtyři vrstvy síťové architektury TCP/IP spolu s protokoly, které se na jednotlivých vrstvách mohou vyskytovat.

1.2 Transmission Control Protocol (TCP)

TCP je jeden ze základních protokolů síťové architektury TCP/IP pracující na transportní vrstvě modelu. Jde o spojově orientovaný protokol, což znamená že před samotnou výměnou požadovaných dat je potřeba provést navázání spojení pomocí tzv. „three-way handshake“. Stejně pak při ukončování TCP komunikace je potřeba ukončit spojení, pomocí předem definovaných typů zpráv.

TCP se snaží zajistit spolehlivý přenos všech dat mezi klientem a serverem tím způsobem, že všechny přenášené segmenty jsou po přijetí potvrzovány. Pokud nedojde k potvrzení přijetí segmentu, tak se segment posílá opakovaně. Tím se zaručí přenos veškerých dat, což požadujeme například při stahování instalačních souborů z internetu, kdy je potřeba po dokončení stahování sestavit instalační soubor přesně tak, jak vypadal na straně odesílatele. Daní za toto potvrzování všech segmentů je vyšší režie, která způsobí vyšší vytížení komunikačního kanálu. Kromě potvrzování přijatých segmentů se TCP protokol snaží doručovat segmenty v pořadí, v jakém byly odesílány.



Obr. 1.2.1: Jednotlivé kroky navazování a ukončování TCP spojení pomocí definovaných typů zpráv.

Jak bylo zmíněno výše, TCP protokol vyžívá definované typy zpráv v různých situacích. Jednotlivé zprávy se identifikují podle tzv. příznaků, respektive příznakových bitů se kterými pracují počítače. Pro navázání spojení mezi klientem a serverem jsou potřeba 3 zprávy, jak ukazuje obr. 1.2.1. První z nich nese požadavek na vytvoření spojení s příznakem SYN (synchronize sequence number). Tuto zprávu iniciuje klient, který se snaží navázat spojení se serverem. Server mu následně odpoví

zprávou s příznaky SYN a ACK (acknowledgment – potvrzení) v jednom segmentu. Příznakem ACK server potvrzuje přijetí zprávy SYN od klienta a spojení je tak v jednom směru vytvořeno. A právě pro vytvoření spojení i ve druhém směru je serverem zaslána také zpráva s příznakem SYN. Klient poté jen potvrdí přijetí tohoto požadavku na synchronizaci zprávou s příznakem ACK a spojení je úspěšně navázáno. Podobně je realizováno ukončení spojení, kde je využit příznak FIN (No more data from sender) a opět dochází k potvrzení ukončení spojení pomocí příznaku ACK. [1]

1.3 User Datagram Protocol (UDP)

Protokol UDP poskytuje služby také na transportní vrstvě modelu, ale volí jiný přístup týkající se spolehlivosti doručení všech dat než protokol TCP. Nedochází zde k potvrzování doručených paketů a před začátkem UDP komunikace není nutné vytvářet spojení. UDP je tedy nespojovaný protokol, který často bývá označován jako „nespolehlivý“, protože neposkytuje zaručené doručení všech dat v podobě, v jaké byla data odeslána. S tím však počítají aplikace, které tento protokol využívají. Jde o řešení, kdy není nutné přenést všechna data v původní podobě, ale klade se zde důraz na rychlost přenosu bez zbytečných prodlev. Jde tedy o služby zprostředkovávající přenos multimediálního obsahu, jako například on-line video streamy nebo přenos hlasu pomocí VoIP (Voice over IP). Zobrazené video nebo přehraná audio stopa totiž budou pochopitelné i když některý z UDP paketů z komunikace vypadnou. UDP tak snižuje režii, opakovaně nepřenáší ztracené pakety a také má zjednodušené záhlaví oproti protokolu TCP.

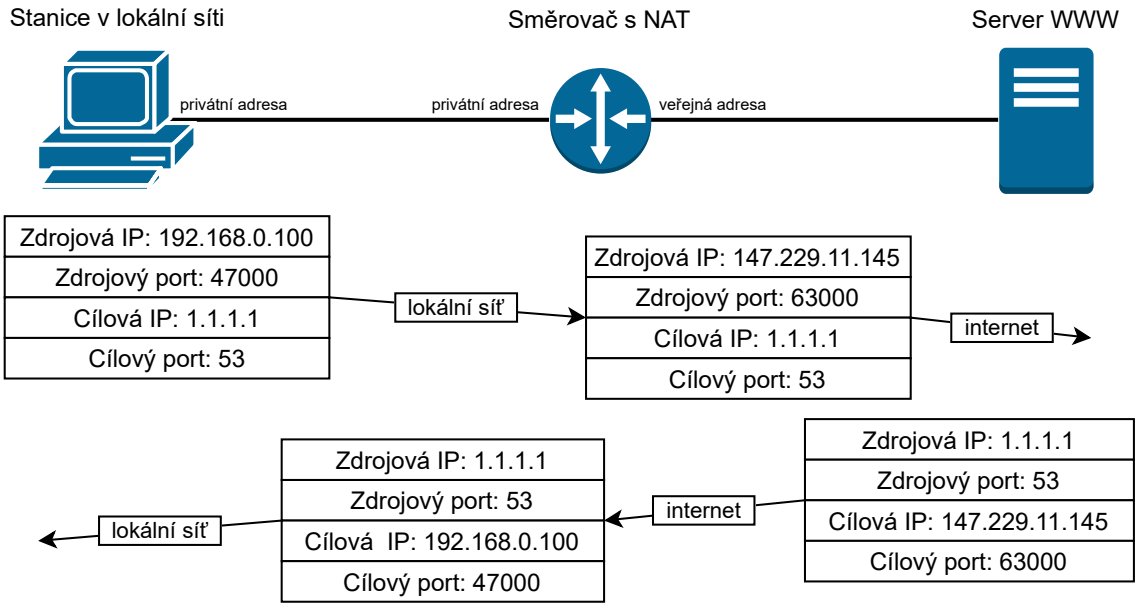
Nad UDP protokolem lze sestavit služby, které změní některé ze zmíněných vlastností. Například protokol QUIC nabízí navazování spojení, potvrzování přijatých paketů nebo zabezpečení přenosu, které samo o sobě nenabízí UDP ani TCP. Zároveň má za cíl snížit čas a množství režie potřebné pro navázání spojení a sestavení zabezpečeného přenosového kanálu.

Protokoly na transportní vrstvě pracují a komunikují s aplikacemi na základě portů. Síťová vrstva nám předá data na základě IP adresy a transportní vrstva tato data dále rozdělí a přidělí konkrétním aplikačním protokolům (HTTP, HTTPS, FTP), respektive aplikacím (webový prohlížeč, FTP klient) na základě portů. [3]

1.4 Překlad síťových adres (NAT)

Překlad síťových adres může probíhat na směrovačích, respektive v paketech, které těmito směrovači procházejí. Překládat se mohou síťové (IP) adresy, pak jde o klasický NAT. Další možností je překlad portů PAT (Port Address Translation). Další rozdělení se odvíjí od směru, který překlad probíhá. Pokud jde o záměnu zdrojové adresy, tak jde o tzv. SNAT (Source NAT). K tomuto překladu může dojít např. při komunikaci směrem ven z vnitřní sítě za směrovačem. Druhou variantou je překlad cílové adresy, tedy DNAT (Destination NAT). Tento překlad se provádí při návratu paketu z internetu zpět do vnitřní sítě. V takovém případě má paket jako cílovou adresu nastavenou IP adresu směrovače. Na tom je poté rozhodnutí, kterému zařízení v jeho podsíti paket odešle, respektive na jakou adresu upraví cílovou IP adresu daného paketu. Toto rozhodnutí směrovači usnadňuje překladová tabulka, kam se při prvním překladu daných adres ukládá záznam o záměně adres a při dalším překladu se směrovač tímto záznamem řídí. Překlad IP adres i portů v obou směrech mezi klientem a serverem ukazuje obr. 1.4.1.

Toto překládání umožňuje šetřit IPv4 adresní prostor, protože pod jedním směrovačem s veřejnou adresou se mohou skrýt další zařízení, která v rámci lokální sítě vystupují pod IP adresou z neveřejného rozsahu. [4]

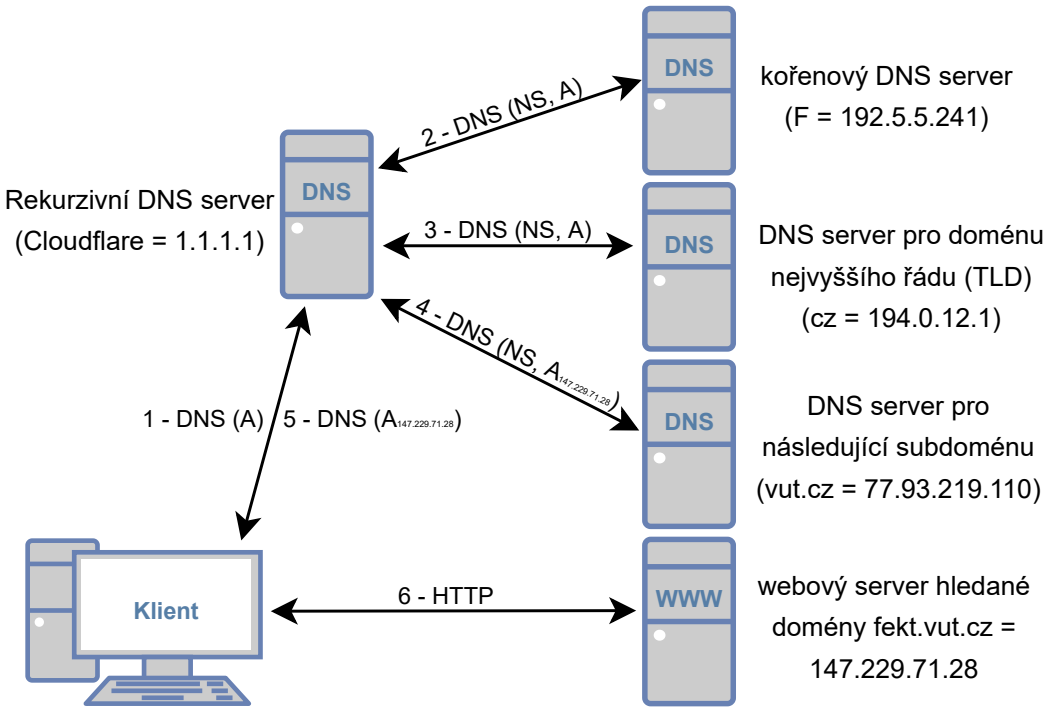


Obr. 1.4.1: Ukázkový příklad zobrazující NAT překlad IP adres i portů v obou směrech při komunikaci mezi klientem a serverem.

1.5 Systém doménových jmen (DNS)

Systém doménových jmen je protokol pracující na aplikační vrstvě, který má na starosti překlad doménových jmen na IP adresy. Uživatel zadává do adresního řádku webového prohlížeče adresu serveru v podobě doménového jména. Toto zadané doménové jméno je nutné přeložit, aby počítač a zařízení v síti věděli, který server mají kontaktovat pro správné načtení webové stránky.

Překlad probíhá s pomocí rekurzivního DNS serveru, na který klient odesílá DNS dotazy, které obsahují dotazované doménové jméno. Tento rekurzivní server se následně dotazuje dalších DNS serverů v hierarchii až se dostane k odpovědi v podobě IP adresy. Tu následně přepošle klientovi, který na zjištěnou adresu odesílá požadavky spojené s dalšími protokoly, jako je například HTTP. Schéma komunikace je ukázáno na obr. 1.5.1.



Obr. 1.5.1: Schéma DNS zobrazující komunikaci mezi klientem, rekurzivním serverem a navazujícími DNS servery.

Výše popsaný postup využívá DNS dotazy typu A pro komunikaci pomocí IPv4m případně dotazy typu AAAA pro IPv6 komunikaci. DNS protokol však umožňuje využití i dalších typů dotazů, jako je například typ NS, PTR, MX nebo CNAME. Zmíněné záznamy typu NS (Name Server) se používají při komunikaci hlavně mezi DNS servery, kdy se nadřazenější servery ptají na jména a IP adresy

DNS serverů v dotazovaných podřazených doménách. Mějme například doménu fekt.vut.cz. Rekurzivní DNS server se bude nejdříve dotazovat kořenového (root) DNS serveru, jestli zná doménu .cz a to právě dotazem typu NS. Kořenový server poskytne IP adresu a rekurzivní DNS server se následně doptává DNS serveru pro doménu .cz na IP adresu DNS serveru pro subdoménu vut.cz. Pokud mu bude IP adresa poskytnuta, tak se již dále nevyužije záznam typu NS, ale rekurzivní server zvolí klasický dotaz typu A a od DNS serveru domény vut.cz si zjistí přímo IP adresu serveru, ze kterého bude dostupná hledaná doména fekt.vut.cz. Další zmíněný typ DNS dotazu je PTR, který slouží naopak k překladu, kdy známe IP adresu a hledáme k ní doménové jméno.

Celá tato DNS komunikace není zabezpečená, takže v případě zachycení síťové komunikace může případný útočník data pozměnit nebo pouze zobrazit, takže by věděl, na které webové stránky, respektive doménová jména jsme se dotazovali. [5]

1.5.1 Domain Name System Security Extensions (DNSSEC)

Autentizaci přijatých dat mezi rekurzivním a dotazovaným DNS serverem řeší rozšíření DNSSEC. Rekurzivní resolver přijatá data ověřuje pomocí přiložených digitálních podpisů v RRSIG (Resource Record Signature) záznamech a také pomocí veřejných klíčů, které si může vyžádat skrze DNSKEY (DNS Public Key) záznamy. Toto řešení však stále spoléhá na to, že mezi rekurzivním resolverem a klientem probíhá komunikace důvěryhodným kanálem, protože v této části není zabezpečena ani autentizace ani důvěrnost přenášených informací. Pro zajištění autentizace až ke klientovi, který DNS komunikaci inicioval, by se muselo ověřování RRSIG záznamů provádět až na koncovém zařízení klienta, což by více zatěžovalo jak jeho síť, tak i samotný DNS systém. Kromě již zmíněných záznamů RRSIG a DNSKEY můžeme při DNSSEC komunikaci ještě narazit na záznamy typu DS (Delegation Signer) a NSEC (Next Secure). DS slouží k ověřování veřejného klíče v DNSKEY u nadřazené domény. Prakticky pak může být například DNSKEY domény seznam.cz ověřen u nadřazené domény cz. Záznamy NSEC jsou využity v případě, kdy se snažíme dotazovat na neexistující doménu. Je nutné i tuto informaci zaobalit v samostatném záznamu, protože jinak by mohl útočník využívat neexistující nebo podobné domény ve svůj prospěch. Takto máme pomocí NSEC záznamu autentizovanou informaci o tom, že DNS server dotazovanou doménu nezná.

Pokud klient požaduje provedení autentizace pomocí rozšíření DNSSEC, pak je nutné tuto volbu specifikovat již v DNS dotazu. Tam se tato informace dokládá v tzv. příznakových bitech AD (Authenticate Data) a CD (Checking Disabled). Bit AD rekurzivnímu resolveru říká, zda požadujeme provedení autentizace a bit CD

oznamuje, zda chceme přijmout i neautentizovaná data, nebo striktně požadujeme příjem pouze autentizovaných dat. Dále si pak klient může vyžádat i zaslání RRSIG záznamu spolu s hlavním záznamem typu A, který mu doručí přeložené doménové jméno. Aby se k odpovědi přiložil RRSIG záznam, tak je potřeba nastavit příznakový bit DO (DNSSEC OK). V případě, že se bit DO nenastaví, tak stále může být rekurzivním resolverem provedena autentizace, ale mezi klientem a resolverem se přenesou pouze záznamy typu A nebo AAAA bez RRSIG. [6]

Samotná práce rekurzivního resolver je v první části obdobná jako u klasického DNS bez využití DNSSEC, kromě nastavení zmíněných příznakových bitů. Resolver začíná komunikaci s kořenovým DNS serverem a ptá se jich na názvy serverů, které obsluhují doménu (například cz) pomocí NS záznamů. K odpovědi, je ale již připojen kromě NS záznamu, také záznam typu DS, který nám sděluje otisk veřejného klíče právě pro podřazenou doménu (cz). Záznam DS je poté ještě podepsán digitálním podpisem, který je zaslán v navazujícím RRSIG záznamu.

Pro provedení autentizace je dále potřeba všem použitým DNS serverům odeslat DNS požadavek o zaslání veřejného klíče záznamem DNSKEY. Nejdříve se dotaz na záznam DNSKEY odesílá na kořenový DNS server a poté na navazující servery. V odpovědi dostaneme dva veřejné klíče označované jako Key Signing Key (KSK) a Zone Signing Key (ZSK). Následně se porovnává hash KSK z DNSKEY záznamu s otiskem hashe z DS záznamu pro kořenovou (root) zónu. Veřejný klíč ZSK se použije k ověření RRSIG zázánamů v dané zóně. Poté se proces přesouvá na komunikaci s DNS serverem domény .cz a jsou odesílány DNS dotazy na záznamy DNSKEY podobně jako v předchozím případě. Na konci řetězce dojde k ověření RRSIG podpisu záznamu typu A a v případě, že všechna ověření proběhla úspěšně, tak se tento A záznam přepošle od rekurzivního serveru ke klientovi. [7]

1.6 Internet Control Message Protocol (ICMP)

ICMP je protokol pracující na síťové vrstvě komunikačního modelu TCP/IP, který má za úkol přenos kontrolních zpráv. Tyto zprávy slouží k informování například o nedostupnosti cílové sítě, o vypršení hodnoty TTL (Time-to-live) během přenosu, o nemožnosti poskládat fragmenty zpět do původní podoby atd. Kromě toho také protokol ICMP zajišťuje zasílání zpráv při vykonávání příkazu ping nebo traceroute v příkazové řádce pomocí zpráv echo request a echo reply. Echo request zpráva je odesílána klientem směrem k požadovanému serveru. Pokud je tento server dostupný, tak tuto ICMP zprávu přijme a vygeneruje odpověď v podobě zprávy echo reply, kterou odešle směrem ke klientovi. Pokud paket projde sítí v pořádku i ve druhém směru, tak nám příkazová řádka oznámí, že je dotazované zařízení dostupné a také nám zobrazí dobu odezvy.

Tab. 1.1: Označení pomocí typu a kódu a význam kontrolních zpráv protokolu ICMP.

typ	kód	popis chyby nebo oznámení
0	0	echo reply – odpověď používaná u příkazu ping
8	0	echo request – požadavek používaný při použití příkazu ping
3	0	cílová síť není dostupná
3	4	je vyžadována fragmentace, ale je nastaven bit „Don't fragment“
11	0	běh5em přenosu vypršela hodnota TTL (Time-to-live)
11	1	fragmenty nelze poskládat zpět do původní podoby

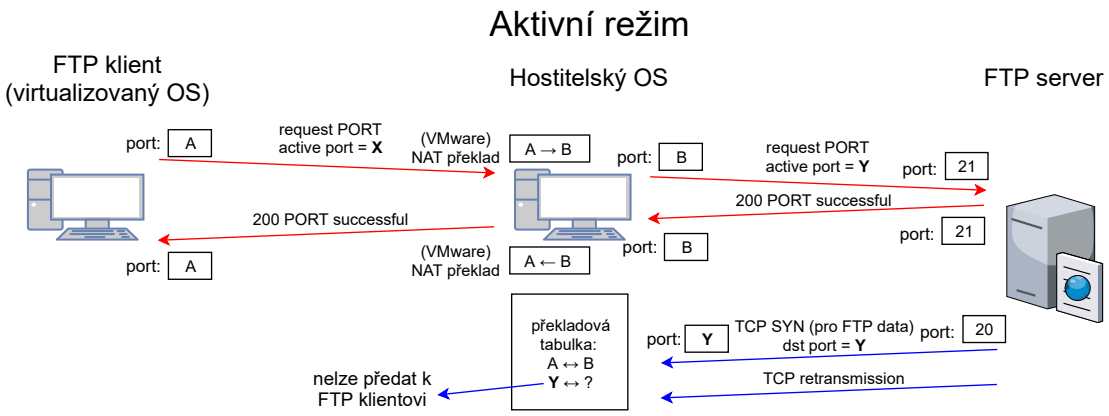
Jednotlivé typy zpráv jsou identifikovány podle typu a kódu, což je číselné označení uváděné v záhlaví ICMP protokolu. Některé z těchto zpráv jsou spolu označením typu a kódu zobrazeny v tab. 1.1. Kromě těchto dvou hodnot se v záhlaví nachází také hodnota kontrolního součtu a také proměnná část, ve které se může uchovávat například hodnota pořadového čísla nebo dalších identifikátorů. Toto pořadové číslo (sequence number) je využíváno k rozlišení jednotlivých dotazů v případě, že se některý z paketů při přenosu zpozdí v síti nebo je současně odesíláno několik ICMP požadavků z jednoho zařízení. V případě, že v operačním systému Windows ihned po zavedení OS odešleme příkaz ping v příkazové řádce, tak by měl první ICMP paket se zprávou echo request obsahovat pořadové číslo 1, stejně jako zpráva echo reply odpovídající na tento paket. Druhá zpráva echo request by měla mít pořadové číslo inkrementováno o jedna a stejně tak všechny navazující zprávy oproti jejich předchozím zprávám. [8]

1.7 File Transfer Protocol (FTP)

FTP (File Transfer Protocol) je protokol umožňující vzdálený přenos souborů prostřednictvím Internetu. Komunikujícími stranami jsou klient a FTP server. Klientem může být webový prohlížeč nebo specifická aplikace podporující připojení k FTP serveru (např. Total Commander nebo FileZilla). Na druhé straně FTP server vyčkává na požadavky a připojení klienta. Tyto strany využívají pro FTP komunikaci transportní protokol TCP, protože požadujeme spolehlivý přenos všech dat a následné sestavení dat do původní podoby a to i za cenu delší doby přenosu a vyšší režie. Klient se připojuje na server skrze port 21, který obsluhuje tzv. řídící spojení. Toto spojení zajišťuje identifikaci klienta, přenos uživatelského jména a hesla a také další požadavky, které slouží k vyžádání určitých souborů nebo přesunu mezi adresáři. Pro samotný přenos dat se na straně FTP serveru využívá port číslo 20 (tzv. datové spojení). Na straně klienta je pro každý soubor vytvořeno nové TCP spojení a je tedy využit i odlišný port. [2]

Kromě známých bezpečnostních problémů, kdy klasická verze FTP umožňuje po zachycení paketů získat přihlašovací údaje klienta, může nastat problém i při použití FTP spolu s NAT překladem. Záleží však na použitém způsobu překladu adres a portů. FTP umožňuje vytvoření spojení dvěma způsoby, kdy každé z nich se chová odlišně při současně aktivním NAT překladu kdekoliv na trase mezi klientem a serverem. U FTP je možnost zvolit aktivní nebo pasivní režim. Volba režimu se týká pouze datového spojení (port 20 na straně serveru), nikoliv řídícího spojení (port 21).

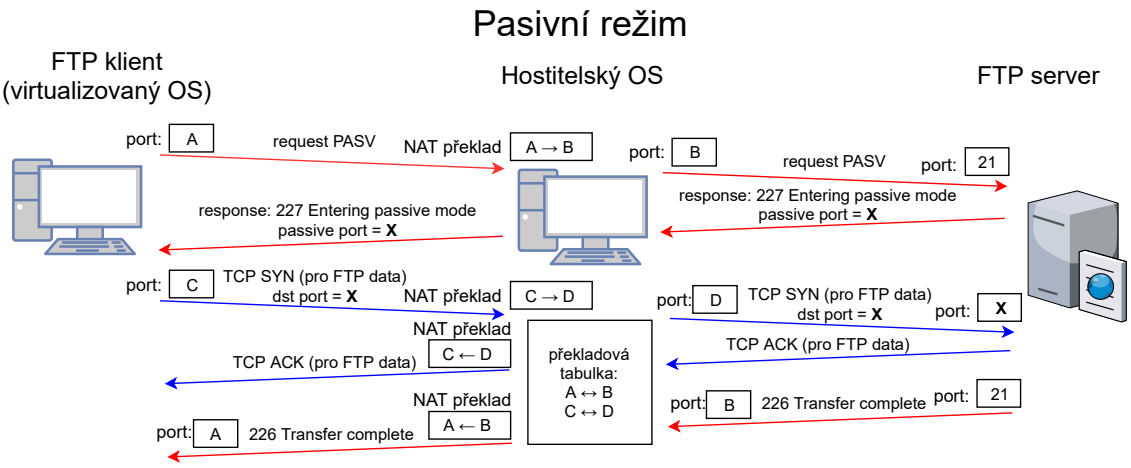
U aktivního režimu si sám klient určuje náhodné číslo portu v paketu s textovou zprávou PORT, ke kterému se má server připojit a následně server inicializuje komunikaci směrem k tomuto portu. To způsobí problém při NAT překladu, protože tabulka pro překlad neví, který port byl zvolen pro příjem na straně klienta. Vzhledem k principu fungování Source NATu a tomu, že datová komunikace začíná na straně serveru je jasné, že datové spojení nebude vytvořeno. Source NAT je totiž založen na tom, že první paket musí být přenesen z vnitřní sítě NAT, nikoliv ze strany serveru směrem k síti s NAT překladem. Zároveň komunikace na tento port začíná ze strany serveru, takže NAT při přijetí paketu od serveru neví, na který port a adresu jej má přeložit. FTP komunikace tak selže a připojení k severu se nezdaří. Schéma FTP komunikace při zvoleném aktivním režimu ukazuje obr. 1.7.1.



Obr. 1.7.1: Schéma FTP komunikace při použití aktivního režimu a současném využití NAT překladu mezi klientem a serverem.

Oproti tomu u pasivního režimu si klient zažádá o port pro pasivní režim v paketu s textovou zprávou PASV a server mu v odpovědi zašle číslo portu, na které se klient může připojit. Následně sám klient komunikaci směrem k serveru na zvolený port inicializuje. Tabulka pro NAT překlad tedy uloží číslo zvoleného portu a následnou odpověď ze strany serveru již bude umět přeložit a FTP komunikace tedy bude úspěšná. Schéma FTP komunikace při zvoleném pasivním režimu zobrazuje

obr. 1.7.2. Z obrázku je zřejmé, že všechna TCP spojení začínají zprávou od klienta, což v případě Source NAT není žádný problém. [9]



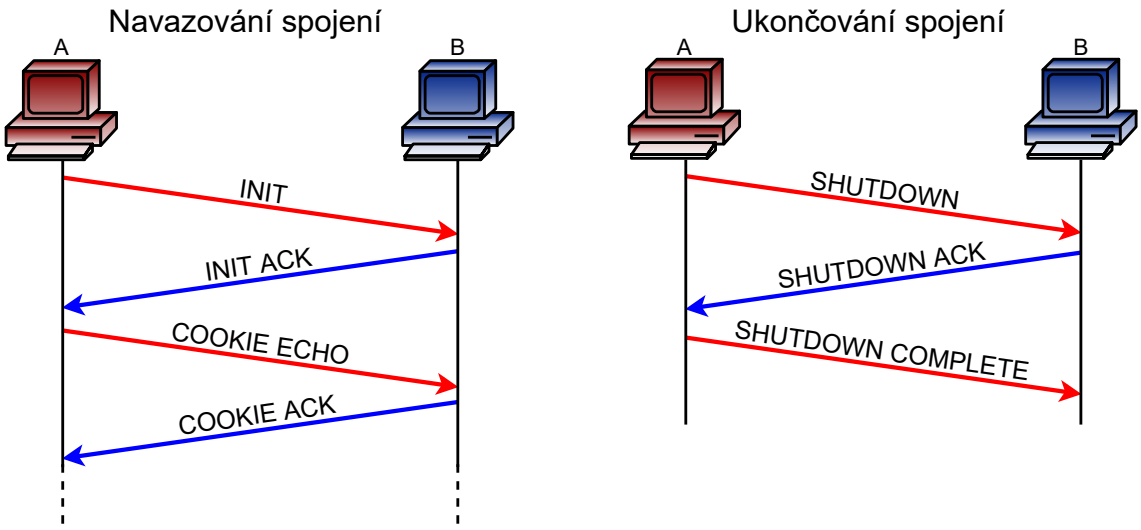
Obr. 1.7.2: Schéma FTP komunikace při použití pasivního režimu a současném využití NAT překladu.

1.8 Stream Control Transmission Protocol (SCTP)

SCTP protokol je protokol transportní vrstvy, který je stejně jako protokol TCP spojově orientovaný a také je označován za spolehlivý. Takže podobně jako u TCP protokolu i u SCTP dojde k přeneseních všech dat, které můžeme poskládat do původní podoby. Existuje však jedna výjimka a to v případě, že nastavíme parametr „Message Time-to-Live“, kterým můžeme měnit dobu života pro danou zprávu. Pokud tato doba vyprší před doručením zprávy k příjemci, tak můžeme celou zprávu zahodit a nemusíme ji doručovat. Dalším typem zprávy jsou zprávy SCTP Heartbeat (Keep-alive), které jsou posílány v okamžicích, kdy není spojení využito a nejsou posílány žádné jiné zprávy. Tento typ zprávy slouží k testování dostupnosti obou stran a úpravě hodnoty Round Trip Time (RTT). Zprávy jsou zasílány v intervalech (Heartbeat interval), pokud nebyla přenesena žádná jiná SCTP zpráva v daném spojení po dobu jednoho intervalu, jehož časové rozmezí lze přizpůsobit.

Kromě těchto typů zpráv přináší SCTP ještě tzv. multihoming a multistreaming. Multihoming představuje situaci, kdy jednomu koncovému bodu komunikace přiřadíme více IP adres. Tento bod následně využívá hlavní IP adresu a v případě poruchy může využít jinou IP adresu. Multistreaming umožňuje přes jedno sestavené SCTP spojení přenášet několik proudů dat nezávisle na sobě, což je u protokolu TCP možné pouze v případě vytvoření několika spojení zároveň. [1]

Jak bylo zmíněno dříve, tak je protokol SCTP spojově orientovaný. To stejně jako u protokolu TCP značí, že před samotnou výměnou dat mezi komunikujícími stranami je potřeba sestavit spojení. U SCTP jsou pro navázání spojení přenášeny 4 zprávy (4-way handshake) a pro ukončení spojení v obou směrech 3 zprávy, což je striktně dané a nelze spojení ukončit jen v jednom směru, jako je tomu u TCP. Rozdílné jsou také názvy příznaků, které jsou při navazování a ukončování spojení zasílány ve zprávách mezi oběma stranami, jak je znázorněno na obr. 1.8.1. Jako první je přenášena zpráva INIT, která zahajuje SCTP navázání spojení. Protistrana potvrzuje zahájení navazování komunikace zprávou INIT ACK, ve které je tzv. ověřovací značka (state cookie), která obsahuje Message Authentication Code (MAC. První strana tuto zprávu přijme včetně ověřovací značky, kterou následně používá po celou dobu komunikace ve všech jejích zprávách. Zároveň tato strana přijatou ověřovací značku vloží do zprávy COOKIE ECHO, kterou opět posílá protistraně. Protistrana zprávu přijme a následně ji ověřuje pomocí dříve zmíněné hodnoty MAC. Pokud proběhne ověření v pořádku, tak je spojení navázáno, což je stvrzeno zprávou COOKIE ACK, která je poslána iniciátorovi spojení. Ukončení spojení začíná zprávou SHUTDOWN směrem od klienta k serveru. Pokud server nechce udržovat toto spojení dále, tak pošle klientovi zprávu SHUTDOWN ACK a klient ukončení potvrdí zprávou SHUTDOWN COMPLETE. [10]



Obr. 1.8.1: Schéma navazování a ukončování spojení u protokolu SCTP.

1.9 DNS over HTTPS (DoH)

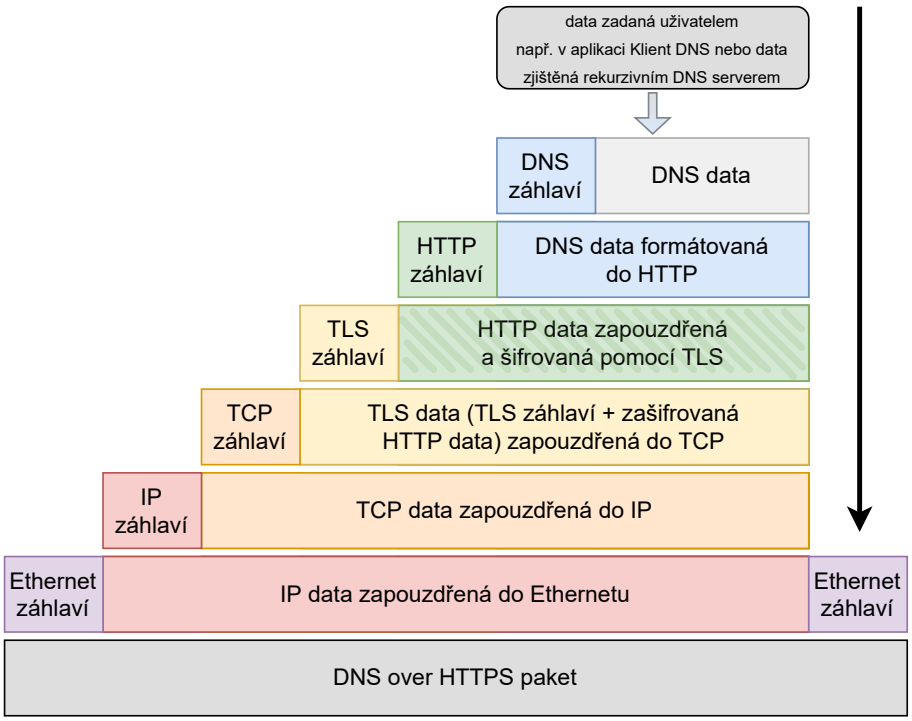
Protokol DNS ani jeho rozšíření v podobě DNSSEC nezajišťují služby bezpečnosti při komunikaci rekurzivního serveru a klienta. Takže i když použijeme rozšíření DNSSEC, tak stále může dojít k narušení autenticity, integrity nebo důvěrnosti při přenosu na tzv. poslední míli, což je označení právě pro komunikaci na začátku a konci DNS komunikace, kdy se klient dotazuje nebo dostává odpověď od rekurzivního serveru.

Protokol DNS over HTTPS tedy umožňuje zajistit autenticitu, která je stejně jako u DNSSEC rozšíření zajištěna pomocí typu zprávy RRSIG. Tato zpráva uchovává informace o digitálním podpisu, který může být použit pro ověření autenticity, ale samotné připojení této zprávy nám autenticitu ani integritu nezařídí. K tomu slouží použité šifrování, které je hlavní změnou oproti DNSSEC. Celá DNS zpráva je včetně zprávy typu RRSIG zašifrována a díky tomu je zajištěna nejen autenticita a integrita, ale také důvěrnost, protože oproti klasickému DNS protokolu zde není možnost při zachycení paketů na poslední míli zobrazit a případně upravovat překládanou doménu a ani již přeloženou IP adresu. Jde tak o zabezpečení proti útokům Man in the middle, jako je například DNS spoofing.

Nejvíce rozšířenou a implementovanou variantou šifrovaného DNS protokolu je právě DNS over HTTPS, u kterého je obsah DNS zprávy zapouzdřen do hlavičky HTTP (Hypertext Transfer Protocol) protokolu a tato data jsou následně šifrována protokolem TLS (Transport Layer Security), kterému je věnována podkapitola 1.10. Zapouzdření dat protokolu DNS over HTTPS je zobrazeno na obr. 1.9.1. Kromě této varianty, která využívá síťový port 443, stejně jako klasická HTTPS komunikace, existují i další varianty šifrované DNS komunikace. Například DNS over TLS a DNS over DTLS (Datagram Transport Layer Security), které se nespoléhají na zapouzdření HTTP záhlaví, ale pouze na šifrování pomocí TLS. Tyto šifrované protokoly jsou používány méně oproti DoH hlavně z důvodu možného blokování například firewallem, protože využívají speciální čísla portů. Protokol DoT využívá port číslo 853, jak pro verzi, která využívá transportní protokol TCP (DNS over TLS), tak i UDP (DNS over DTLS). Je také možné využít variantu DNS over QUIC. [11]

Podpora protokolu DoH je implementována v nejznámějších webových prohlížečích jako je Chrome, Edge nebo Firefox. Kromě samotné podpory je ale pro bezpečnost také důležité to, aby byla tato možnost využívána již ve výchozím nastavení těchto prohlížečů, což ještě v minulých letech nebylo zcela zavedeno a mnoho běžných uživatelů se tak k této funkci nedostalo. Kromě webových prohlížečů je protokol DoH podporován také v operačním systému Windows 11 a jeho využití je nejspíše plánováno i ve Windows 10, kde je protokol prozatím

součástí pouze u testovacích verzí operačního systému. [12] [13]



Obr. 1.9.1: Zapouzdření dat u protokolu DNS over HTTPS.

1.10 Transport Layer Security (TLS)

Transport Layer Security (TLS) je protokol, který byl navržen tak, aby zajišťoval zabezpečený přenos dat při komunikaci mezi klientem a serverem. Protokol TLS je implementován například při komunikaci pomocí webového prohlížeče, kde požadujeme maximální možnou ochranu dat, kterými mohou být osobní údaje nebo například údaje o platebních kartách. V takovém případě vyžadujeme, aby se z klasické HTTP komunikace, která je zasílána v otevřené podobě, stala komunikace zabezpečená, která využívá šifrování a je označována jako HTTPS. Kromě tohoto nejběžnějšího příkladu můžeme na protokol TLS narazit také u zabezpečené mailové komunikace pomocí SMTPS (Simple Mail Transfer Protocol Secure) nebo u zabezpečené hlasové VoIP (Voice over IP) komunikace. V této práci se protokol TLS zmiňuje především u zabezpečené DNS komunikace, která může být zajišťována protokoly DNS over HTTPS nebo také DNS over TLS.

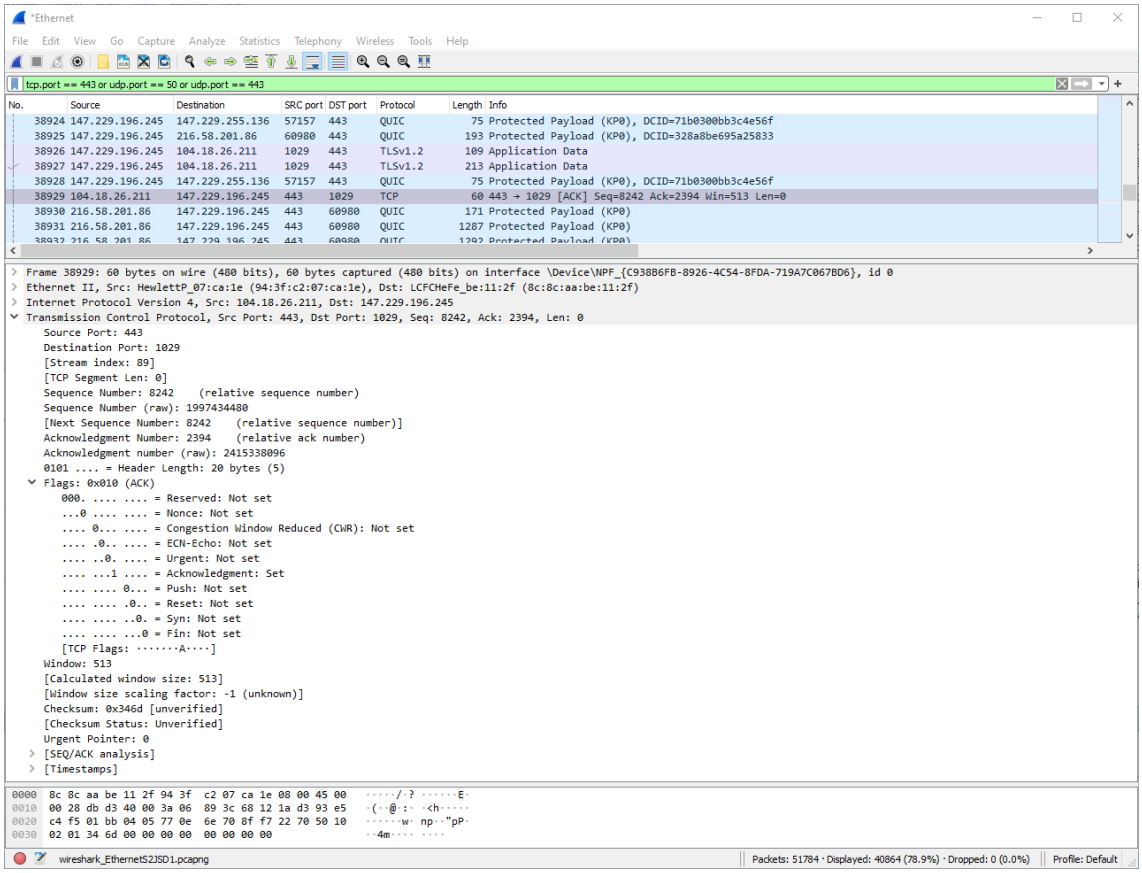
TLS nám u těchto aplikací zajišťuje důvěrnost, autentizaci i integritu. Důvěrnost je zajištěna pomocí šifrování, které zabrání možným útočníkům získat citlivé údaje v čitelné podobě. Pro ustanovení šifrované komunikace je potřeba provést navázání

spojení (TLS handshake), ve kterém se specifikuje, kterou verzi protokolu TLS bude klient se serverem využívat. Můžeme narazit na protokol TLS ve verzí 1.0, 1.2 a dnes již především na verzi 1.3. Dále při navazování TLS spojení dojde k ověření autenticity serveru. Server je klientem ověřen pomocí veřejného klíče a TLS certifikátu. A jsou také ustanoveny klíče pro danou relaci (session keys), které jsou vytvořeny z náhodných hodnot (pro klienta i server) a z tajného klíče (premaster secret), který je mezi stranami vyměněn pomocí asymetrické kryptografie. Takto vytvořené klíče (session keys) jsou po úspěšném sestavení TLS spojení využity pro šifrování a dešifrování komunikace (HTTP, DNS...) symetrickou kryptografií. [14]

2 Použité programy a aplikace

2.1 Wireshark

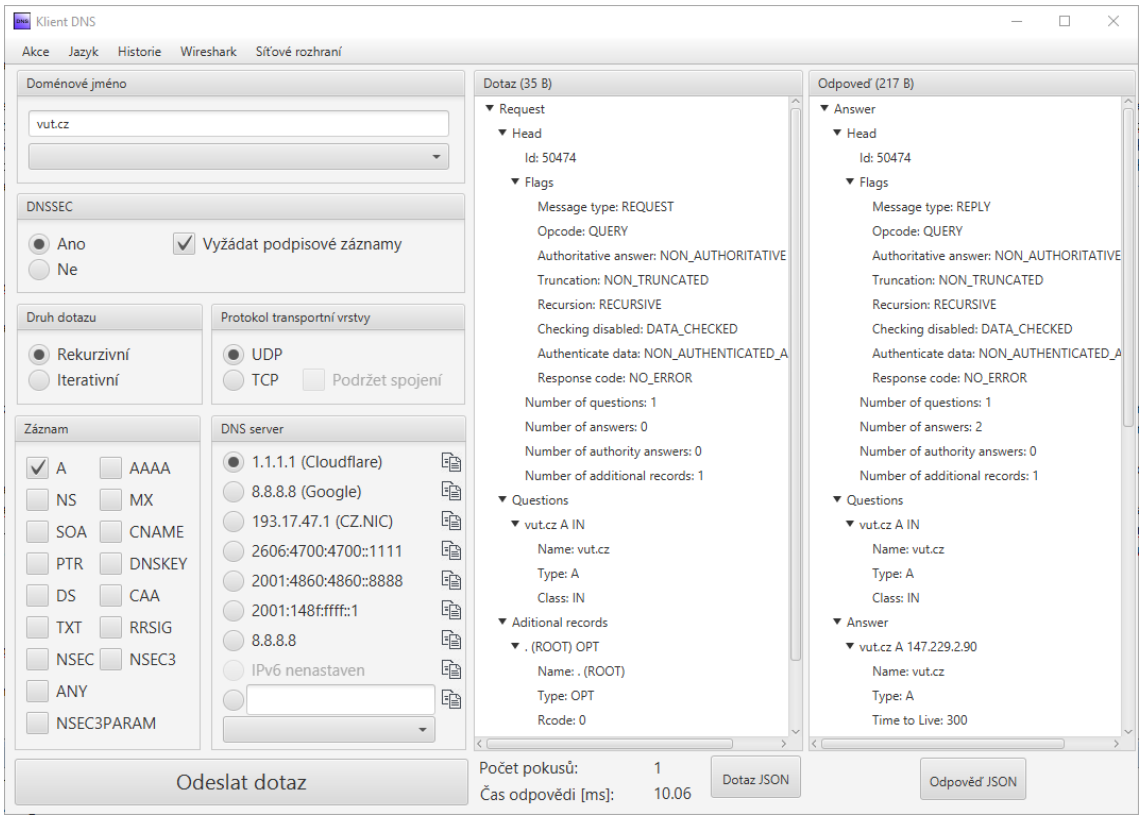
Wireshark je jedním ze základních programů, který bude využit pro analýzu paketů ve všech vytvořených scénářích. Wireshark dokáže zachytit komunikaci, která prochází přes síťovou kartu. Následně tuto komunikaci v podobě paketů dokáže přehledně zobrazit. Uživatel poté může jednotlivé pakety sledovat a také filtrovat podle použitého protokolu, IP adres, portů a dalších podrobností. Získaná data lze využít pro další analýzu pomocí grafů nebo schémat komunikace a zachycený síťový provoz lze také uložit pro pozdější zpracování v mnoha různých formátech. Program je vydavatelem stále aktualizován a podporován a je dostupný zdarma pro nejpoužívanější operační systémy, jako je Windows, Linux i macOS. Ukázku hlavní obrazovky programu Wireshark můžeme vidět na obr. 2.1.1. [15]



Obr. 2.1.1: Úvodní obrazovka programu Wireshark se zachyceným síťovým provozem.

2.2 Klient DNS

Klient DNS je studentsky vytvořená a stále vyvíjená aplikace, která umožňuje generovat DNS dotazy v různých formách, které následně můžeme analyzovat buď pomocí programu Wireshark nebo přímo v prostředí aplikace Klient DNS, kde můžeme zobrazit základní strukturu DNS dotazu i odpovědi. U klasických DNS dotazů umožňuje aplikace nastavit několik parametrů, jako například použitý transportní protokol, využitý DNS server, typ DNS záznamu a hlavně můžeme definovat dotazované doménové jméno. Kromě toho umožňuje aplikace využít i rozšíření DNSSEC, DNS over HTTP nebo mDNS, čehož bude využito při tvorbě a řešení scénářů. Aplikace umožňuje volbu síťového rozhraní a také můžeme zvolit preferovaný jazyk, na výběr máme češtinu a angličtinu. Hlavní obrazovka aplikace jsou zobrazeny na obr. 2.2.1. [16]



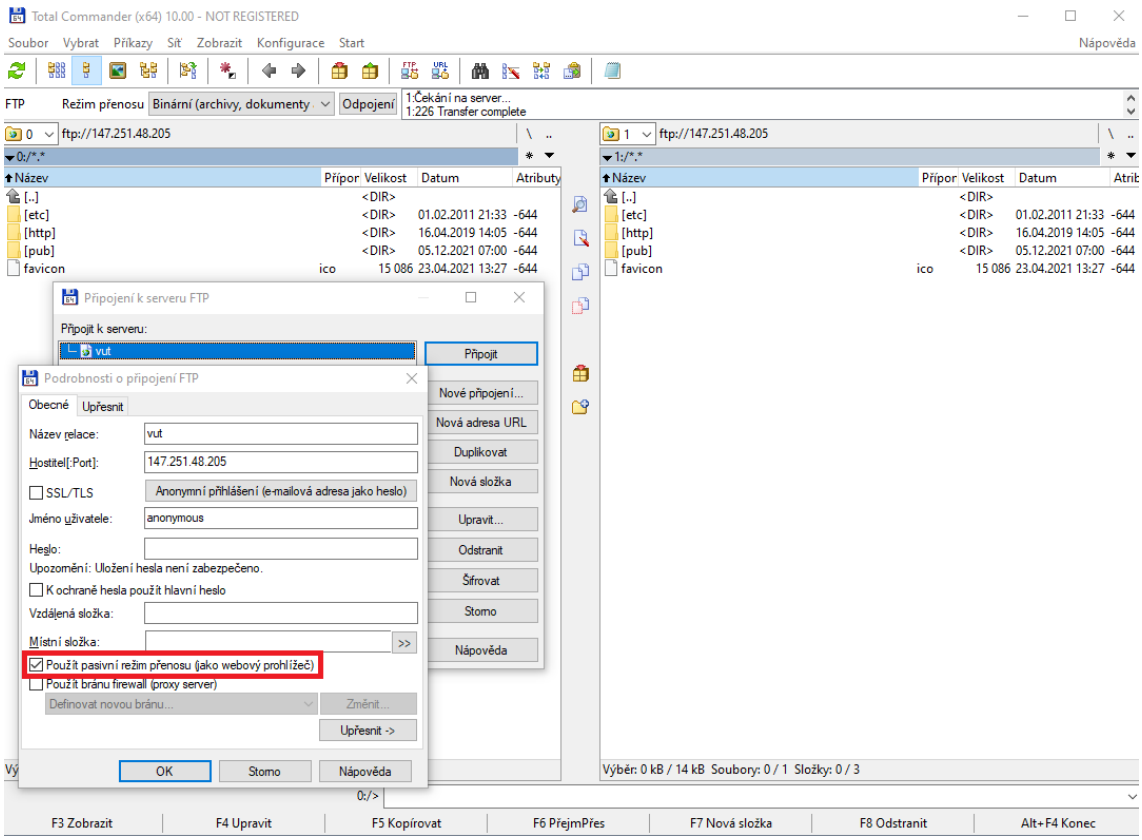
Obr. 2.2.1: Hlavní obrazovka aplikace Klient DNS, která vlevo ukazuje parametry, které můžeme nastavovat a v pravé části můžeme vidět základní strukturu odeslaného DNS dotazu a přijaté DNS odpovědi.

2.3 VMware Workstation

VMware Workstation je program, který umožňuje virtualizovat několik virtuálních strojů na jednom fyzickém zařízení. Můžeme tak vytvořit a používat virtualizovaný stroj s vlastním operačním systémem, nastavením a konfigurací. Toho lze využít například při výuce, kdy při zpracování úloh studenty může docházet k různým úpravám systému a nastavení programů a to vše při poskytnutí administrátorského oprávnění, takže studenti mohou se systémem pracovat bez jakýchkoliv omezení. Hlavní výhodou je to, že po dokončení úlohy lze virtualizovaný stroj vrátit do původního stavu a mohou na něm bez problému pracovat další studenti. Další výhodou je, že v tomto prostředí můžeme simulovat různé situace, například se síťovou kartou. Můžeme simulovat zpoždění sítě nebo úplně změnit režim připojení virtuálního síťového adaptéru například na NAT. Toho využívá i jeden z vytvořených scénářů, kde se testuje chování jednotlivých protokolů při zvoleném NAT překladu na síťové kartě. Na připravovaném virtualizovaném stroji bude zprovozněn operační systém Windows 10 a budou zde dostupné všechny programy, které jsou nutné pro úspěšné zvládnutí všech vytvořených scénářů a tyto scénáře zde budou také otestovány. [17]

2.4 Total Commander

Program Total Commander je jedním z nejvyužívanějších správců souborů a složek zejména pro operační systém Windows. Lze jej ale spustit také na systémech Linux nebo Android. Program podporuje širokou škálu různých typů souborů a také obsahuje funkci FPT klienta. A to jak v klasické variantě (FTP), tak i ve variantě šifrované (FTPS). Ve vytvářeném scénáři bude využit program Total Commander jako FTP Klient v klasické nešifrované variantě FTP. Bude zde ukázka chování protokolu FTP při současně použitém NAT překladu. FTP protokol a stejně tak i program Total Commander nabízejí volbu jednoho ze dvou režimů připojení k FTP serveru, přičemž se každý chová odlišně při překladu adres a portů NAT. Jedná se o aktivní a pasivní režim, kdy aktivní režim neumožňuje úspěšné připojení k FTP serveru při současně běžícím NAT překladu. Naopak při zvoleném pasivním režimu by připojení k serveru mělo fungovat bez problémů i při použitém NAT překladu. Oba případy budou zahrnuty v prvním vytvořeném scénáři a přepínání těchto režimů a generování FTP paketů nám zajistí právě program Total Commander, jak můžeme vidět na obr. 2.4.1. [18]



Obr. 2.4.1: Hlavní obrazovka programu Total Commander a okno pro nastavení FTP připojení, včetně zaškrtnuté položky pro použití pasivního režimu.

3 Návrhy scénářů

V rámci této práce byly navrženy čtyři scénáře a z těchto návrhů se poté vycházelo při jejich vytváření a realizaci. Každý scénář obsahuje krátký teoretický úvod, na který pak naváže praktická část s řešením scénáře a samostatné úkoly. Správné řešení v podobě odpovědí nebo vyplněných schémat k těmto úkolům je poté shrnuto v samostatných souborech mimo scénáře, které mohou být předloženy studentům.

První scénář obsahuje postupy a úkoly, které studentům objasní princip překladu síťových adres pomocí NAT. Téma druhého a třetího scénáře se částečně prolíná, protože v obou těchto scénářích dochází k využití rozšíření DNSSEC při překladu doménových jmen. Ve druhém scénáři je popisováno a analyzováno nejen DNSSEC rozšíření, ale také protokol DoH. Cílem třetího scénáře je objasnění funkcí rekurzivních serverů u DNSSEC komunikace. A poslední vytvořený scénář je zaměřen na problematiku plánování a přidělování adresního prostoru a s tím spojená témata, jako jsou směrovací tabulky nebo překlad adres NAT.

V prvních třech scénářích se využívá virtualizované prostředí operačního systému Windows 10, kdy virtualizace probíhá pomocí programu VMware a v těchto scénářích je také využit paketový analyzer Wireshark. Pro čtvrtý scénář není žádné speciální prostředí potřeba.

3.1 Návrh prvního scénáře

První scénář se zaměří na problematiku překladu síťových adres protokolem NAT. Princip bude vysvětlen pomocí překladu u protokolu UDP, jehož pakety budou generovány pomocí aplikace Klient DNS. Součástí úlohy bude i popis této aplikace, která umí vytvářet širokou škálu DNS dotazů s různými parametry. Dále se zde vysvětlí princip překladu IP adres i portů a následně scénář ukáže odlišnosti při NAT překladu u různých protokolů, jako například ICMP, FTP nebo SCTP.

Začátek úlohy bude věnován správnému nastavení virtualizovaného systému, konkrétně síťového adaptéru, bez jehož nastavení by k NAT překladu nemuselo vůbec docházet. Po kontrole připojení k internetu bude vysvětlena obsluha aplikace Klient DNS a také se nastaví jednotlivé položky, jako je doménové jméno, IP adresa DNS serveru a další důležité parametry pro úvodní zachycení paketů programem Wireshark. Následně dojde k základní analýze těchto paketů před a po NAT překladu. U této základní DNS komunikace, která bude probíhat pomocí transportního protokolu UDP, se před a po překladu u DNS dotazu změní zdrojová IP adresa, zdrojový port a hodnota kontrolního součtu. U DNS odpovědi pak půjde o překlad cílové adresy a cílového portu. Tím bude studentům ukázán základní princip NAT překladu, jehož pochopení bude nutné pro následující podkapitoly

a úkoly. Hned v další podkapitole bude zadána samostatná práce, kde budou studenti analyzovat TCP komunikaci. Následně bude vytvořen úkol, který ověří pochopení této problematiky. Při něm budou muset studenti doplnit jednotlivé adresy a porty před a po NAT překladu do názorného obrázku, který bude součástí úkolu. Následující část bude zaměřena hlavně na překlad portů při několika po sobě jdoucích DNS paketech. Číslo zdrojového portu pro jednotlivé DNS dotazy se totiž liší, a to jak před NAT překladem, tak i po tomto překladu.

V dalších podkapitolách dojde k zachycení paketů nesoucích protokoly ICMP, FTP a SCTP. U protokolu ICMP se kromě adres a kontrolních součtů změní také hodnota TTL. V části s protokolem FTP bude vysvětleno, jakým způsobem se připojit na FTP server skrze program Total Commander a budou popsány dva režimy, ve kterých je možné se k FTP serveru připojit. Každý z nich posílá jiný typ zpráv při navazování komunikace se serverem. A hlavně jsou mezi režimy rozdíly při současně použitém NAT překladu. Poslední část scénáře se bude zabývat protokolem SCTP, jehož pakety nebudou studenti zachytávat přímo, ale pro analýzu použijí předem vytvořený soubor z linuxového operačního systému. Hlavním úkolem této části s protokolem SCTP bude ukázat, že ne všechny protokoly při současně použitém NAT překladu fungují. Zástupcem z této kategorie bude právě protokol SCTP, který není s NAT překladem kompatibilní, o čemž nás informuje jeden ze zachycených ICMP paketů se zprávou, že zvolený transportní protokol není podporován.

3.2 Návrh druhého scénáře

Druhý scénář bude ve své první části vysvětlovat funkce rozšíření DNSSEC, které je možné implementovat u DNS protokolu. Budou ukázány výhody rozšíření DNSSEC a také rozdíly v jeho záhlaví oproti běžným DNS paketům. Podobným způsobem bude ve druhé části vysvětlen protokol DNS over HTTPS, včetně popisu a analýzy jeho komunikace a také definice jeho přínosu oproti klasickému protokolu DNS nebo rozšíření DNSSEC. Využita zde bude opět aplikace Klient DNS, program Wireshark a také webová aplikace DNSSEC Analyzer.

Scénář bude začínat základní analýzou DNSSECu s vysvětlením principu jednotlivých bitů, které se u běžných DNS paketů nevyskytují nebo jsou nastaveny na jiné hodnoty, jako jsou AD, CD nebo DO bity. Dále v průběhu úlohy budou vysvětleny jednotlivé typy záznamů, které se u DNSSEC dají použít. Zejména půjde o záznamy typu NS, RRSIG, NSEC, DNSKEY nebo DS a opět se tedy jedná hlavně o záznamy, které se u klasického DNS nevyskytují. Následující podkapitoly se tak budou zaměřovat vždy na jeden typ záznamu nebo na specifickou situaci, která může při komunikaci s různými servery nastat.

Jako první bude nastíněna situace, kdy se budeme dotazovat na neexistující doménu. V této části by se mělo odpovědět na otázku, jak je daná situace vyřešena, jaký typ záznamu je v takové situaci využit a také jak je daný záznam zabezpečen v porovnání s běžným DNSSEC dotazem, kdy se dotazujeme na existující doménu. Dále bude scénář pokračovat dotazy na doménu, která bude v odpovědi vracet chybně podepsaný záznam. Nejdříve bude vyzkoušen dotaz na překlad doménového jména pomocí běžného DNS dotazu a poté pomocí DNS dotazu s rozšířením DNSSEC. Tím bude objasněna hlavní výhoda, kterou DNSSEC přináší. Následovat budou dotazy na domény, které vůbec nepodporují rozšíření DNSSEC. V této části bude také využit webový nástroj DNSSEC Analyzer. Studenti také budou moci zjistit, jak je na tom DNSSEC z hlediska jeho rozšíření v jednotlivých státech světa.

Poslední část tohoto scénáře se zaměří na protokol DNS over HTTPS (DoH), jehož pakety lze také generovat v aplikaci Klient DNS, podobně jako u předchozího DNSSEC. Nejdříve bude vysvětleno, jak probíhá komunikace pomocí DoH a co všechno můžeme při této komunikaci zobrazit v programu Wireshark. V dalších částech bude porovnána velikost a množství paketů potřebné při použití DNSSEC se situací bez jeho využití a nakonec bude otestována implementace protokolu DoH přímo ve webovém prohlížeči.

3.3 Návrh třetího scénáře

Třetí scénář bude úzce spjatý s druhým scénářem, protože se bude stále věnovat problematice rozšíření DNSSEC. Obsahem scénáře bude popis práce rekurzivních serverů, které zprostředkovávají DNS komunikaci mezi klientem, který požaduje překlad doménového jména na IP adresu, a mezi DNS servery, které tento překlad realizují na několika úrovních. Bude tak plně objasněn proces kontaktování jednotlivých DNS serverů. Nejdříve se začíná od kořenových DNS serverů, následně se dotazujeme DNS serverů, které komunikují na úrovni TLD (Top Level Domain – doména nejvyššího řádu) a poté postupujeme k dalším DNS serverům v subdoménách. Pro generování DNSSEC dotazů bude i v tomto scénáři opět využita aplikace Klient DNS.

Studentům bude vysvětlen princip na ukázkovém příkladu a následně si sami vyzkouší simulovat práci rekurzivního serveru tak, že se budou dotazovat jednotlivých DNS serverů a budou zjišťovat IP adresy pro navázání komunikace s následujícími subdoménami. Výsledkem by měla být přeložená IP adresa webového serveru, která odpovídá doménovému jménu, které na začátku procesu definuje klient. Tento proces bude obsahovat klasickou DNS komunikaci rekurzivního serveru s dotazy na záznamy typu NS, ale dojde také na využití záznamů určených pro komunikaci při využití služeb DNSSEC rozšíření. Studenti tak budou muset pracovat

i se záznamy typu RRSIG nebo DNSKEY, které slouží k ověřování dříve získaných záznamů. V následující části dojde k porovnání zachycené komunikace při simulaci rekurzivního serveru s komunikací mezi klientem a rekurzivním serverem. Toto porovnání bude založeno na časové odezvě jednotlivých částí DNSSEC komunikace a budou k tomu využity funkce programu Wireshark. Půjde například o úpravu zobrazovaných sloupců s hodnotami, kde si studenti zobrazí zmiňovanou časovou odezvu DNSSEC odpovědí. Dále také studenti využijí nástroj pro tvorbu grafu ze zachycených paketů. Tyto pakety budou poté ještě analyzovány pomocí další integrované utility programu Wireshark, která dokáže zobrazit podrobné statistiky například o počtu jednotlivých záznamů ve vyfiltrovaných paketech. V poslední části tohoto scénáře dojde k využití webového nástroje DNSViz. Ten vizuálně objasní návaznost jednotlivých zón a záznamů u DNS komunikace.

Studenti si tak v tomto scénáři ujasní, kolik paketů v DNSSEC komunikaci obstarává rekurzivní DNS resolver bez vědomí uživatele, kterému přijde pouze jediný paket s odpovědí a také si zopakují, jak na sebe jednotlivé zóny u DNS komunikace navazují.

3.4 Návrh čtvrtého scénáře

Poslední z vytvořených scénářů se bude věnovat plánování a přidělování adresního prostoru. Plánování zahrnuje zejména rozhodování o tom, jaký typ adres požadujeme nebo můžeme využít pro konkrétní sítě. Jde tedy o volbu, zda využijeme veřejné nebo privátní rozsahy IPv4 adres, případně také IPv6 rozsahy. Následně dojde k přidělování adresního prostoru, kdy se rozhodujeme, které poskytnuté rozsahy IP adres využijeme pro kterou síť. K této problematice se také váže využití NAT překladu a také směrovací tabulky.

Tento scénář vždy studentům přidělí určité rozsahy IP adres a také schéma, které bude zobrazovat jednotlivé sítě. Tyto sítě budou v každé podkapitole disponovat jiným počtem koncových zařízení. Zařízení bude vždy buď server nebo osobní počítač, který chceme propojit s ostatními zařízeními v dané síti, případně i mezi ostatními sítěmi. Proto každé zařízení musí na konci úlohy disponovat vlastní IP adresou z nabídnutých rozsahů. Scénář bude rozdělen do čtyř kapitol a každá kapitola bude disponovat nejdříve částí, která popíše řešenou situaci s určitým přiděleným typem adres a následovat bude samostatná část, kde budou studenti sami přidělovat jednotlivé rozsahy do připravených šablon.

V první kapitole při realizaci tohoto scénáře budou k dispozici pouze veřejné rozsahy adres a všechna zařízení by tak měla mít možnost komunikovat mezi sebou i bez použití NAT překladu. V navazující kapitole bude možné přiřadit pouze adresy z privátních rozsahů a zařízení tak budou moci komunikovat pouze v rámci jejich

interní sítě. Třetí kapitola zkombinuje dva předchozí přístupy a ukáže tak jeden z dnes nejběžnějších postupů v IPv4 prostředí, kdy se využívají jak veřejné, tak privátní adresy. Bude zde tedy také zopakován i prakticky ukázán princip NAT překladu IP adres. Kromě toho dojde také na tvorbu směrovacích tabulek, bez kterých by komunikace v reálném prostředí internetu nebyla možná. První tři kapitoly se věnují čistě IPv4 adresám. To se změní u čtvrté kapitoly, která využije adresy IPv6. I pro tento typ IP adres budou sestaveny směrovací tabulky, a to nejdříve v názorné ukázce a následně půjde o jeden ze samostatných úkolů pro studenty.

Tento scénář tedy zopakuje základní znalosti o IP adresách a jejich typech. Studenti si ujasní, které adresy z přidělených rozsahů je možné přiřadit koncovým zařízením, bude zde vysvětlen princip a funkce směrovacích tabulek a tím i základní princip vzájemné komunikace síťových a koncových zařízení v prostředí internetu.

3.5 Kompletní znění vytvořených scénářů

Vzhledem k rozsáhlosti vytvořených scénářů jsou kompletní soubory se scénáři z důvodu přehlednosti uvedeny v příloze této práce. Nejdříve jsou řazeny všechny čtyři vytvořené scénáře, kdy první scénář obsahující mimo jiné 12 samostatných úkolů pro studenty je uveden v příloze A. Druhý scénář, jehož obsahem je i 11 samostatných úkolů, se nachází v příloze B. Ve třetím scénáři, který je uveden v příloze C, se nachází celkem 10 samostatných úkolů. A čtvrtý scénář, nacházející se v příloze D, zadává studentům 16 samostatných úkolů.

Následně jsou v příloze uvedeny soubory, které shrnují samostatné úkoly a odpovědi na ně. Konkrétně se jedná o přílohy E, F, G a H. Nakonec je v kapitole I uvedena struktura a také popis elektronické přílohy práce. V odevzdané příloze jsou nahrány všechny vytvořené scénáře, včetně souborů pro jejich případnou editaci pro potřeby výuky.

Závěr

Tato práce ve své teoretické části popisuje základní protokoly síťového modelu TCP/IP využité ve čtyřech scénářích, které byly navrženy a následně vytvořeny. Návrhy popisují, kterými protokoly se budou dané scénáře zabývat a také jaké programy a aplikace budou využívat ke svému řešení. Dále je zde nastíněn postup a návaznost úkolů v jednotlivých scénářích.

Scénáře byly vytvářeny a konzultovány s vedoucím práce tak, aby mohly být využity v rámci předmětů, které se věnují komunikačním technologiím. Co se týká využitých a také teoreticky popsaných protokolů, tak jde především o transportní protokoly TCP, UDP a v části prvního scénáře se využívá i protokol SCTP. Následně byl popsán i překlad síťových adres a portů NAT, kterému se věnuje celý první scénář, a to při současném použití protokolů například ICMP nebo FTP. Z pohledu druhého a třetího scénáře je důležitou součástí této práce také protokol DNS a jeho rozšíření DNSSEC. Druhý scénář popisuje a analyzuje základní komunikaci s tímto rozšířením a také se na něj dále dívá z hlediska jeho využití v různých situacích, které mohou nastat. Oproti tomu třetí scénář je zaměřen pouze na obecnou základní komunikaci, kterou ale i obyčejný uživatel využije několikrát za den. O to důležitější je pak znát a porozumět těmto základním procesům, které na sebe navazují, ať už v běžném DNS, nebo i v rozšíření DNSSEC. Jde zde zejména o to, co všechno zajišťuje samotný rekurzivní server, aniž by o tom koncový uživatel musel mít nějaké povědomí. Kromě rozšíření DNSSEC je ve třetím scénáři popsána a analyzována komunikace protokolu DoH. Poslední scénář se od ostatních liší tím, že se přímo nezaměřuje na nějaký konkrétní protokol aplikační, případně transportní vrstvy modelu TCP/IP, který by dopodrobna analyzoval. Čtvrtý scénář se totiž zabývá zejména přiřazováním IP adres z přidělených rozsahů a je zde také využit překlad adres pomocí NAT. I tento scénář a v něm probíraná problematika se tak stávají jedním ze základních pilířů znalostí, které je potřeba v oboru komunikačních technologií znát a rozumět jim.

Z hlediska využitých programů byl nejdůležitější program Wireshark. V něm probíhalo zachycení síťové komunikace, následná analýza paketů a především protokolů, které byly v těchto paketech přenášeny. Wireshark je jedním z nejznámějších programů ve svém oboru a od studentů komunikačních technologií se tak očekává, že budou seznámení s jeho prostředím a funkcemi. Ke generování DNS, DNSSEC a DoH dotazů byla využita aplikace Klient DNS. Tyto dva zmíněné nástroje jsou využity v prvních třech scénářích. Dále se v práci také využil virtualizační software VMware Workstation, program Total Commander a webové nástroje DNSViz a DNSSEC Analyzer. Tyto webové nástroje byly využity hlavně kvůli jejich přehlednému prostředí a schopnosti dobře vizualizovat teoreticky

popsanou problematiku.

V rámci diplomové práce mělo dojít k návrhu a vytvoření čtyř scénářů, což bylo splněno. Scénáře obsahují krátký teoretický úvod k probíranému tématu a také samostatné úkoly pro studenty. Ty jsou shrnuty ve speciálních souborech mimo vytvořené scénáře a obsahují také odpovědi na položené otázky. Všechny vytvořené soubory byly předány vedoucímu práce, aby bylo možné je v budoucnu využít a případně také upravovat pro potřeby výuky.

Literatura

[1] JEŘÁBEK, Jan. *Pokročilé komunikační techniky* [online]. Brno, 2021, 30.6.2021 [cit. 2021-10-23]. Skriptum. FEKT Vysoké učení technické v Brně. Dostupné z: <https://www.vut.cz/studenti/predmety/detail/241987>.

[2] KUROSE, James F. a Keith W. ROSS. *Computer networking: a top-down approach.* Seventh edition. Essex: Pearson, [2017]. ISBN 978-1-292-15359-9.

[3] ROSENCRANCE, Linda a George LAWTON, MOOZAKIS, Chuck, ed. What is User Datagram Protocol (UDP)? *TechTarget* [online]. Newton, Massachusetts, USA, October 2021 [cit. 2021-11-01]. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol>.

[4] Network Address Translation. *Avi Networks* [online]. Santa Clara, California, © 2021 [cit. 2021-10-29]. Dostupné z: <https://avinetworks.com/glossary/network-address-translation/>.

[5] What is DNS? | How DNS works. *Cloudflare* [online]. San Francisco, Kalifornie, USA, © 2021 [cit. 2021-11-02]. Dostupné z: <https://www.cloudflare.com/learning/dns>.

[6] Overview of DNSSEC. *Microsoft* [online]. Redmond, Washington, USA, © 2021, 08/31/2016 [cit. 2021-11-08]. Dostupné z: <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/jj200221(v=ws.11)>.

[7] How DNSSEC Works. *Cloudflare* [online]. San Francisco, Kalifornie, USA, © 2021 [cit. 2021-11-08]. Dostupné z: <https://www.cloudflare.com/dns/dnssec/how-dnssec-works/>.

[8] RFC 792 - Internet Control Message Protocol. *IETF Datatracker* [online]. Fremont, Kalifornie, USA, 2021 [cit. 2021-11-15]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc792>.

[9] VILLANUEVA, John Carl. Active vs. Passive FTP Simplified: Understanding FTP Ports. *JSCAPE* [online]. Jan 04, 2021 [cit. 2021-11-15]. Dostupné z: <https://bit.ly/jscape_FTP>.

[10] SCTP association startup and shutdown. *IBM* [online]. Armonk, New York, USA [cit. 2021-11-18]. Dostupné z: <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-sctp-association-startup-shutdown>.

[11] POSEY, Brien. DNS over HTTPS (DoH). *TechTarget* [online]. Newton, Massachusetts, USA, May 2020 [cit. 2022-01-28]. Dostupné z:<https://www.techtarget.com/searchsecurity/definition/DNS-over-HTTPS-DoH>.

[12] PARMAR, Mayank. How to enable DNS-over-HTTPS (DoH) in Windows 10. *Bleeping Computer* [online]. September 13, 2020 [cit. 2022-01-28]. Dostupné z: <https://www.bleepingcomputer.com/news/microsoft/how-to-enable-dns-over-https-doh-in-windows-10/>.

[13] ANWAR, Kamil. How to Configure and Use DNS-Over-HTTPS (DoH) in Windows 11. *Appuals* [online]. October 6, 2021 [cit. 2022-01-28]. Dostupné z: <https://appuals.com/configure-doh-windows-11/>.

[14] What happens in a TLS handshake? | SSL handshake. *Cloudflare* [online]. San Francisco, © 2022 [cit. 2022-02-14]. Dostupné z: <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/>.

[15] *Wireshark* [online]. 2021 [cit. 2021-12-02]. Dostupné z: <https://www.wireshark.org/>.

[16] BIOLEK, Martin. *Klientská aplikace protokolu DNS s grafickým rozhraním pro účely výuky* [online]. Brno, 2021 [cit. 2021-12-02]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/133569>. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. Jan Jeřábek, Ph.D.

[17] VMware Workstation 15.1 Pro Release Notes. *VMware Docs* [online]. Palo Alto, Kalifornie, USA, © 2021, 14 May 2019 [cit. 2021-12-02]. Dostupné z: <https://docs.vmware.com/en/VMware-Workstation-Pro/15/rn/VMware-Workstation-151-Pro-Release-Notes.html>.

[18] GHISLER, Christian. *Total Commander* [online]. © 1995-2021 [cit. 2021-12-02]. Dostupné z: <https://www.ghisler.com/

Seznam symbolů a zkratek

AD	Authenticate Data
CD	Checking Disabled
DHCP	Dynamic Host Configuration Protocol
DNAT	Destination NAT
DNS	Systém doménových jmen – Domain Name System
DNSKEY	DNS Public Key
DNSSEC	Domain Name System Security Extensions
DO	DNSSEC OK
DoH	DNS over HTTPS
DS	Delegation Signer
DTLS	Datagram Transport Layer Security
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IoT	Internet of Things
ISP	Internet service provider – Poskytovatel internetového připojení
KSK	key signing key
MAC	Message Authentication Code
NAT	Network Address Translation
NS	Name Server
NSEC	Next Secure
OS	operační systém
PAT	Port Address Translation
RFC	Request for Comments

RRSIG Resource Record Signature

RTT Round Trip Time

SCTP Stream Control Transmission Protocol

SLAAC StateLess Address AutoConfiguration

SMTPS Simple Mail Transfer Protocol Secure

SNAT Source NAT

SOA Start of authority

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

TLD Top Level Domain – doména nejvyššího řádu

TLS Transport Layer Security

UDP User Datagram Protocol

VoIP Voice over IP

ZSK zone signing key

Seznam příloh

A Kompletní návod pro první vytvořený simulační scénář

ÚLOHA č. 1

Překlad adres a dalších parametrů pomocí NAT a jejich analýza u protokolů TCP, UDP, ICMP a SCTP.

A.1 Teoretický úvod

V tomto cvičení se budeme věnovat překladu adres při použití **NAT** (Network Address Translation). K tomu bude využit virtuální stroj VMware, který bude komunikovat s hostitelským operačním systémem a internetem právě přes NAT. Následně budeme porovnávat pakety zachycené v obou systémech a z rozdílů vyplynou vlastnosti při překladu pomocí NAT u různých protokolů.

A.1.1 Princip překladu síťových adres NAT

V tomto cvičení se budou nejdříve porovnávat pakety protokolů **TCP** (Transmission Control Protocol) a **UDP** (User Datagram Protocol) při použití s překladem adres (NAT). A právě u těchto nejpoužívanějších transportních protokolů, můžeme ukázat princip a výhody překladu adres. Překlad adres se vykonává na směrovačích, které nejčastěji zaměňují původní privátní zdrojové adresy paketů od stanic z lokální sítě za jednu jedinou veřejnou adresu, pod kterou vystupuje celá síť za směrovačem. Tento proces platí pro pakety odchozí z dané (pod)sítě. U příchozích paketů na směrovač z Internetu, se překlad provádí opačně. NAT, respektive tabulka překladu se neřídí pouze IP adresami, ale využívá také porty, pro určení konkrétních aplikací na koncových zařízeních. Na směrovačích se původní porty překládají na čísla portů z vyhrazeného rozsahu.

Jednou z hlavních výhod je úspora veřejných IP adres. Toho lze využít například v domácnostech, kde se nepředpokládá vyšší počet připojených zařízení než 254. Nevýhodou NATu může být časová prodleva oproti komunikaci bez použití překladu adres. Nejde jen o zpoždění způsobené načtením hodnot z tabulky pro překlad, ale také opětovný výpočet kontrolních součtů, který se ukládá do záhlaví TCP a UDP paketů. Další nevýhodou je častá nemožnost využití protokolů jako třeba SCTP (Stream Control Transmission Protocol) při současném použití NAT.

A.1.2 File Transfer Protocol

FTP (File Transfer Protocol) je protokol umožňující vzdálený přenos souborů prostřednictvím Internetu. Komunikujícími stranami jsou klient a FTP server. Tyto strany využívají pro FTP komunikaci transportní protokol TCP. Klientem může být webový prohlížeč nebo specifická aplikace podporující připojení k FTP serveru (např. Total Commander nebo FileZilla). Na druhé straně FTP server vyčkává na požadavky a připojení klienta. Toto připojení probíhá skrze port 21, který obsluhuje tzv. řídící spojení. Toto spojení zajišťuje identifikaci klienta, přenos uživatelského jména a hesla a také další požadavky, které slouží k vyžádání určitých souborů nebo přesunu mezi adresáři. Pro samotný přenos dat se na straně FTP serveru využívá

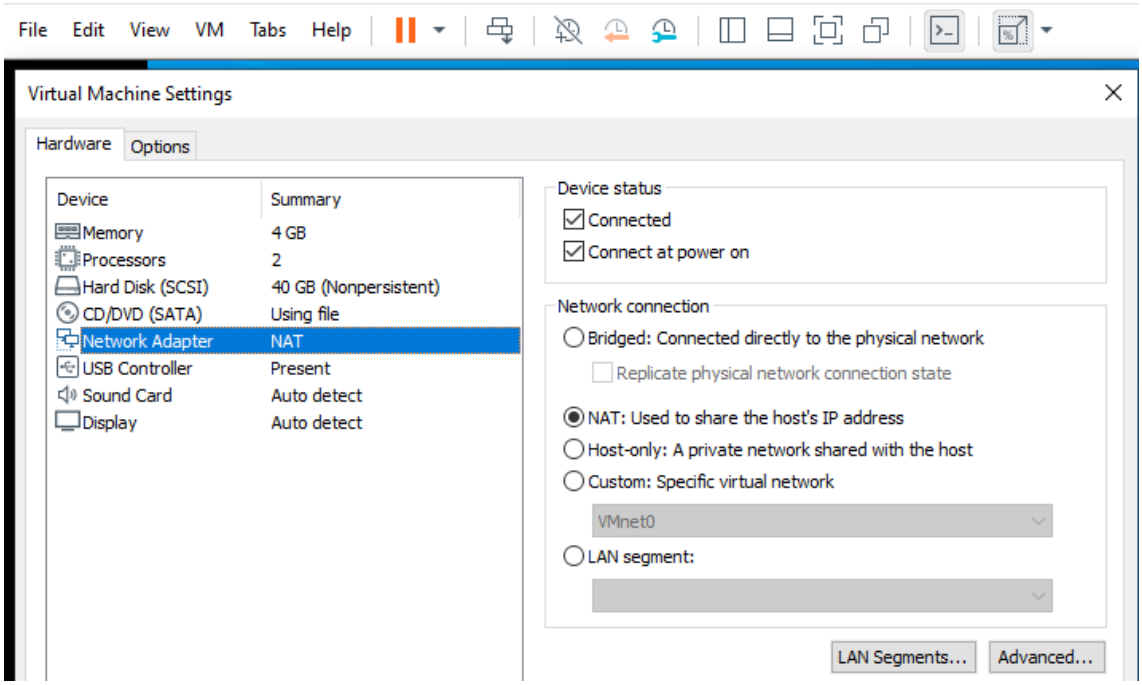
port číslo 20 (tzv. datové spojení). Na straně klienta je pro každý soubor vytvořeno nové TCP spojení a je tedy využit i odlišný port. FTP umožňuje vytvoření spojení dvěma způsoby, kdy každé z nich se chová odlišně. Jde o aktivní a pasivní režim. Popisu těchto režimů se věnuje jedna z podkapitol v následujícím scénáři.

A.2 Realizace scénáře

Tento scénář se bude zabývat srovnáním protokolů při použití metody NAT pro překlad síťových adres a dalších parametrů. Překlad obvykle probíhá na směrovačích. V našem případě budeme překlad realizovat mezi virtualizovaným a hostitelským operačním systémem z důvodu jednoduchosti tohoto řešení. Nebude tedy nutné konfigurovat směrovač a můžeme se více zaměřit na samotný překlad adres a rozdíly tohoto překladu u jednotlivých protokolů.

A.2.1 Spuštění systému a kontrola parametrů

Spusťte virtuální operační systém. Po úspěšném spuštění zkontrolujte správné nastavení virtuálního síťového adaptéru v nastavení programu VMware. Otevřete záložku v horní části obrazovky `VM > Settings...` a následně položka `Network adapter`. Je zde několik možností pro připojení virtualizovaného systému k Internetu. My chceme vyzkoušet jak funguje překlad adres a proto zde ponechte možnost připojení přes NAT, viz obr. A.2.1. Případnou změnu v nastavení potvrďte tlačítkem `OK`.



Obr. A.2.1: Nastavení komunikace přes NAT u síťového adaptéru v programu VMware.

Úkoly:

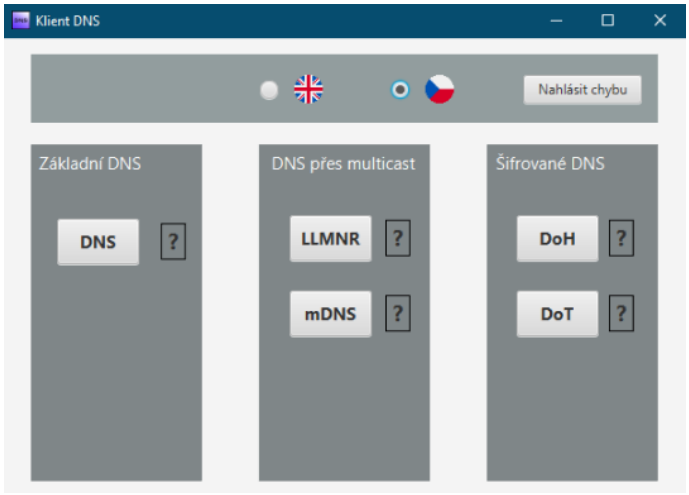
(1) Zkontrolujte konektivitu připojení k Internetu pomocí příkazu ping na libovolný webový server v příkazové řádce.

(2) Pomocí příkazu ipconfig z virtuálního OS zkontrolujte IP adresu síťového adaptéru. O jaký typ adresy se jedná a do kterého rozsahu daná adresa spadá?

A.2.2 Aplikace DNS Klient a její nastavení

Ve virtualizovaném i hostovském operačním systému spusťte aplikaci DNS Klient, která bude generovat TCP a UDP provoz. Ikona aplikace se nachází na ploše.

Aplikace DNS Klient umožňuje zasílat DNS dotazy a přijímat DNS odpovědi. Také je možné zobrazit základní DNS údaje a dále jsou zde k dispozici i pokročilejší funkce DNS, které nebudou součástí tohoto cvičení. Na úvodní obrazovce, kterou lze vidět na obr. A.2.2, tedy zvolíme `DNS` v sekci `Základní DNS`.

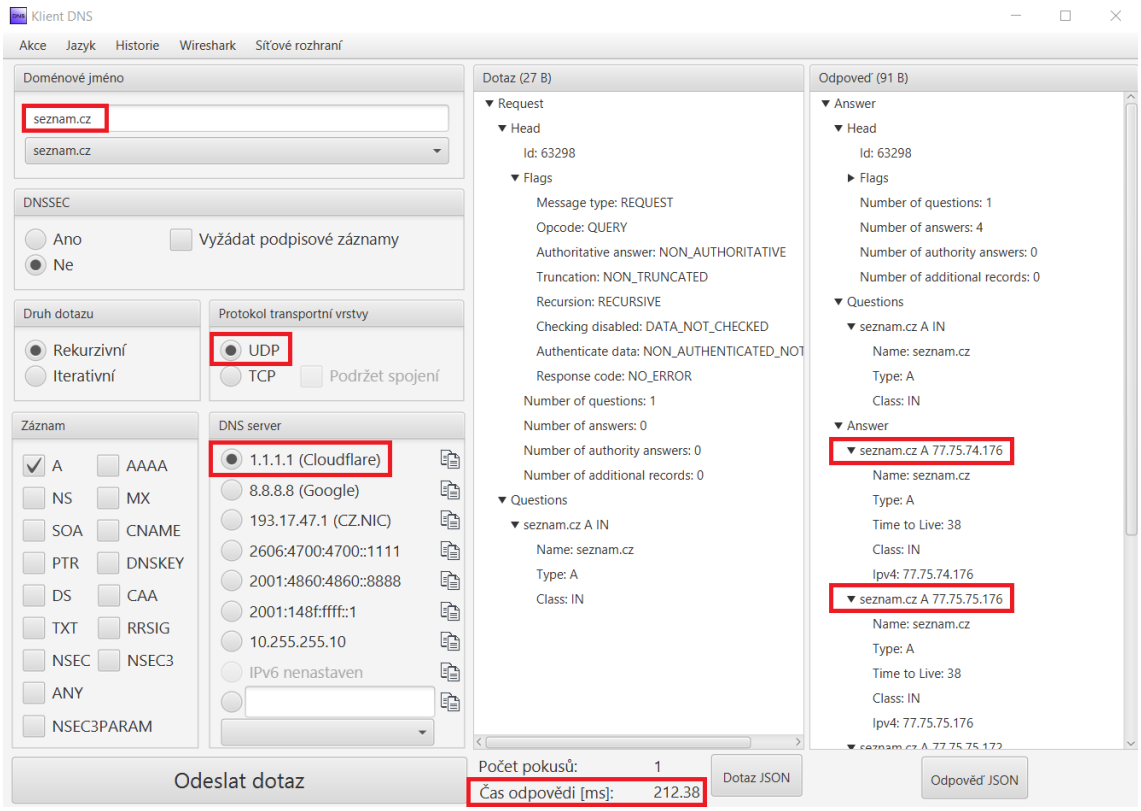


Obr. A.2.2: Úvodní obrazovka aplikace DNS Klient s různými režimy spuštění.

Zobrazí se nám okno aplikace jako na obr. A.2.3. V levém horním rohu můžeme vyplnit doménové jméno do prázdného pole, případně můžeme doménové jméno zvolit z menu pod tímto polem, kam se nám ukládají dříve zadané hodnoty. Dále můžeme zvolit, zda chceme využít DNSSEC, můžeme vybrat druh dotazu, protokol transportní vrstvy, typ záznamu a také adresu DNS serveru.

Pro toto cvičení zvolíme doménové jméno `seznam.cz` a nastavení adresy DNS serveru na server Cloudflare, tedy `1.1.1.1`. Zbytek parametrů ponecháme dle výchozího nastavení, včetně transportního protokolu UDP. Nastavení zkontrolujte podle obr. A.2.3. Pomocí tlačítka `Odeslat dotaz` vyzkoušejte, zda komunikace s DNS serverem funguje. V pravé části by se měl zobrazit jak DNS dotaz, tak

odpověď se zjištěnou IP adresou pro zvolené doménové jméno a v dolní části také čas odpovědi v milisekundách určující prodlevu mezi odesláním dotazu a přijetím odpovědi.



Obr. A.2.3: Hlavní obrazovka aplikace DNS Klient s výchozím nastavením a požadovaným doménovým jménem.

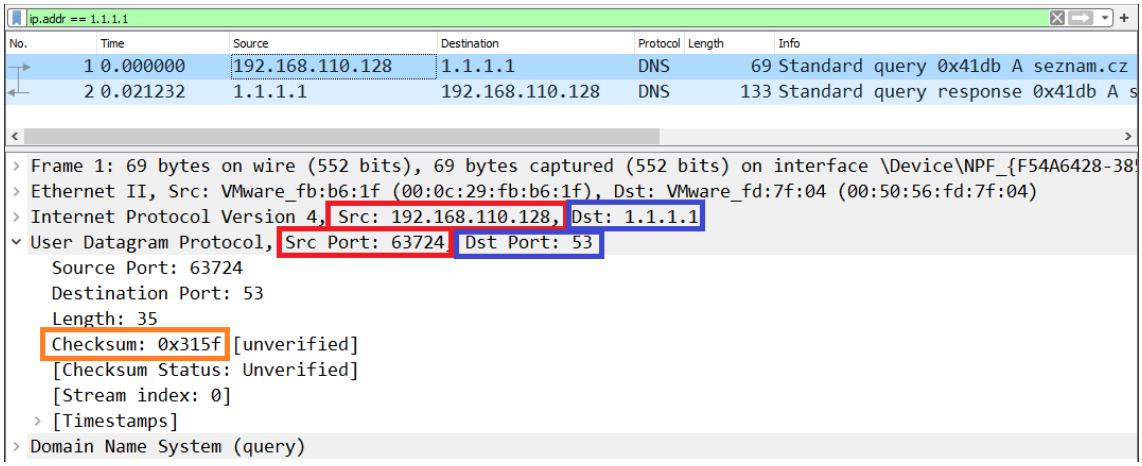
A.2.3 Zachycení UDP paketů s překladem NAT

Pokud vše funguje tak, jak bylo popsáno výše, můžeme se přesunout na samotné zachytávání paketů. To spusťte ve Wiresharku jak ve virtualizovaném, tak v hostitelském operačním systému. Následně odešlete DNS požadavek v aplikaci DNS Klient opět tlačítkem `Odeslat dotaz` a poté zastavte zachytávání paketů v obou instancích programu Wireshark.

Zejména v hostitelském OS pravděpodobně bude vyšší množství rušivé komunikace v podobě paketů jiných protokolů. Ty z výsledků skryjeme vyfiltrováním pouze požadovaných DNS paketů (filtr `dns` je nutné zadat malými písmeny), případně lze využít filtr IP adresy DNS serveru (`ip.addr == 1.1.1.1`). Po odfiltrování u obou relací programu Wireshark, by se ve výsledcích měly objevit

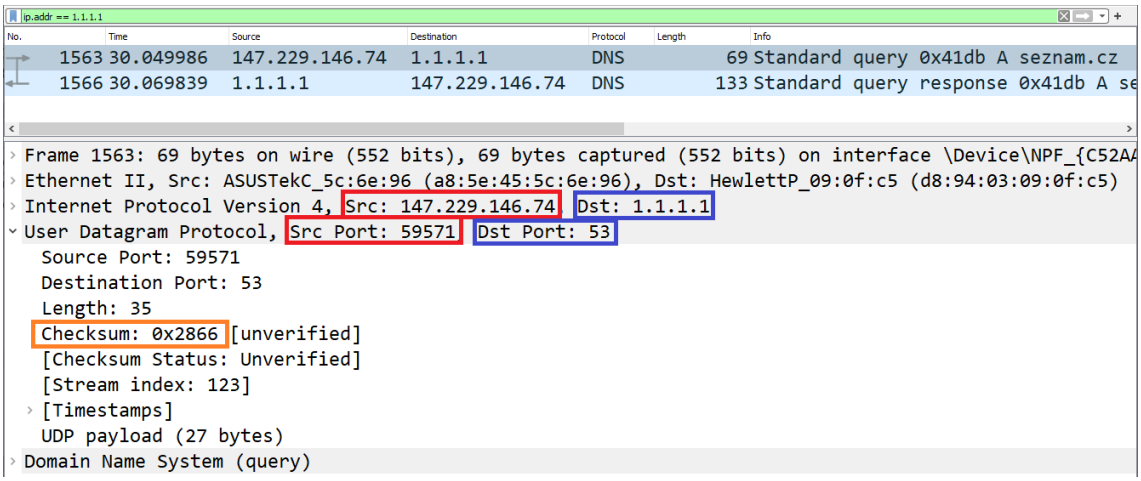
pouze 2 pakety. Konkrétně DNS dotaz a odpověď na námi požadované doménové jméno `seznam.cz`.

Ve virtuálním OS vidíme zdrojovou IP adresu (192.168.110.128) přidělenou z rozsahu pro překlad adres NAT, obdobně jako na obr. A.2.4 (zdrojová IP adresa se může lišit). Cílová adresa se v tomto případě nemění a zůstává po celou dobu komunikace ve směru k serveru stejná, tedy námi požadovaná adresa Cloudflare serveru 1.1.1.1. To stejné platí pro porty. Zdrojový port 63724 odpovídá číslu z určeného rozsahu a cílový port pro DNS služby je port číslo 53. Tento paket s DNS požadavkem byl následně předán na výchozí bránu, což je adaptér hostitelského operačního systému, kde byl proveden překlad zdrojové adresy a zdrojového portu (překlad proběhl v rámci VMware). Zdrojová adresa 192.168.110.128 tedy byla přeložena na veřejnou IP adresu 147.229.146.74, pod kterou vystupuje hostitelský počítač v rámci Internetu. Dříve zmíněný zdrojový port 63724 byl také přeložen a to na číslo portu 59571.[1] Tyto údaje se uložily do překladové tabulky a budou využity při zpětném překladu pro DNS odpověď. Takto upravený paket s přeloženými adresami a porty, jako na obr. A.2.5 se následně směruje do Internetu. Dále si všimněte pozměněné hodnoty checksum, tedy kontrolního součtu, jehož hodnota se musí přepočítat podle nové hodnoty čísla portu. Hodnota součtu se v případě uvedeném na obrázcích níže změnila z původních 0x315f na hodnotu 0x2866.



Obr. A.2.4: Zachycení původního paketu před překladem pomocí NAT na virtualizovaném OS.

[1] Ve vašem případě se adresy IP i portů budou lišit.



Obr. A.2.5: Zachycení paketu po překladu pomocí NAT na hostitelském OS.

Úkoly:

(3) Překládají se tedy IP adresy a porty, což vede na změnu i u příslušného kontrolního součtu. Které parametry naopak zůstávají stejné i po překladu NAT? O kterou vrstvu se jedná?

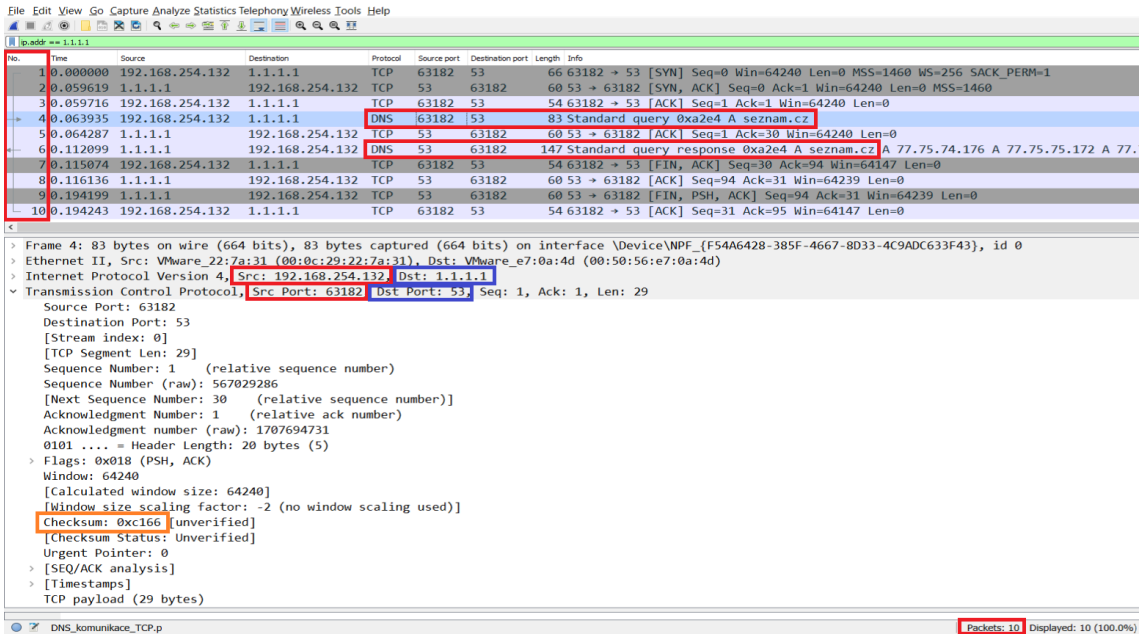
A.2.4 Zachycení TCP paketů s překladem NAT

Nyní postup opakujte, ale v aplikaci DNS Klient změňte používaný transportní protokol z UDP na TCP. Opět zapněte zachytávání v obou relacích programu Wireshark a proveďte odeslání DNS požadavků v aplikaci. Po zastavení zachytávání vyfiltrujte požadované pakety pomocí IP adresy DNS serveru, aby kromě DNS komunikace bylo vidět i navázání spojení protokolu TCP.

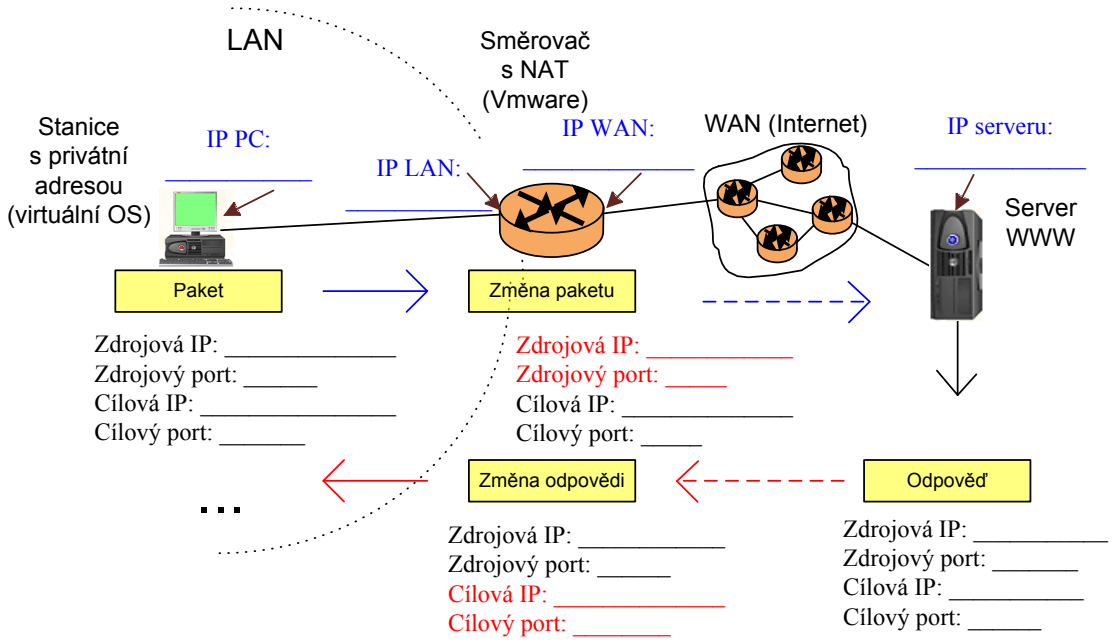
Parametry jako IP adresa a zdrojové porty se při překladu mění podobně jako u předchozí UDP komunikace. Zde je ovšem potřeba přeložit více paketů, než jen DNS dotaz a odpověď (jako v případě UDP). Překládají se totiž i adresy a porty u paketů, které slouží k navázání a ukončení TCP spojení, viz obr. A.2.6.

Úkoly:

(4) Vyplňte zachycené IP adresy a porty do šablony na obr. A.2.7 pro oba směry DNS komunikace v případě použití TCP protokolu. (Směrovači odpovídá hostitelský OS a PC v LAN je v našem případě virtualizovaný OS.)



Obr. A.2.6: Zachycená DNS komunikace realizovaná pomocí transportního protokolu TCP.

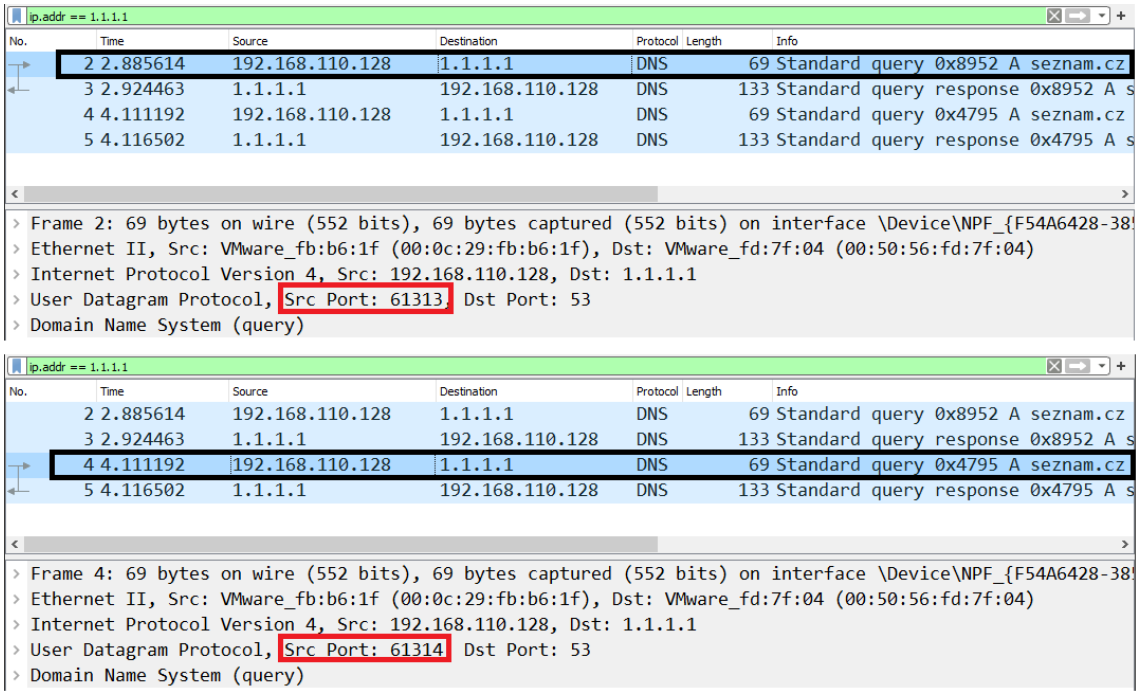


Obr. A.2.7: Šablona pro úkol č. (4) znázorňující překlad NAT při DNS komunikaci.

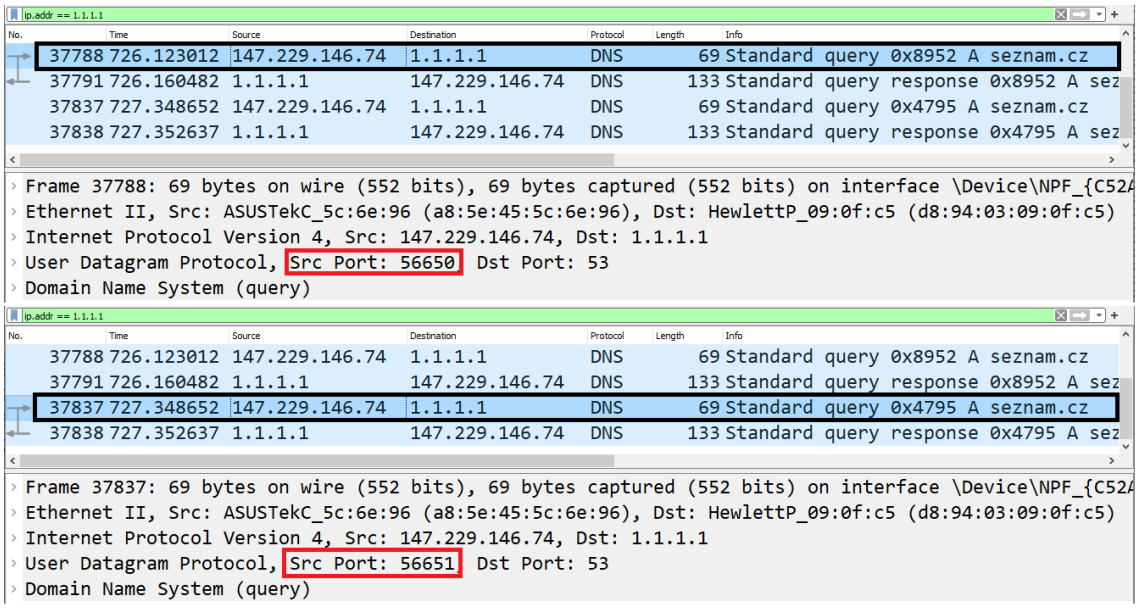
A.2.5 NAT překlad při více DNS požadavcích

Nyní budeme zjišťovat jak bude vypadat komunikace v případě více DNS požadavků jdoucích po sobě a jak se při NAT překladu mezi sebou odliší. V aplikaci DNS klient přepněte transportní protokol zpět na UDP a ostatní parametry ponechejte. Zapněte zachytávání paketů síťové komunikace v obou instancích programu Wireshark a následně odešlete alespoň dva DNS požadavky v aplikaci DNS klient. Zastavte zachytávání paketů a zamyslete se nad tím, jaká čísla zdrojových portů očekáváme u jednotlivých DNS dotazů, ať už ve virtualizovaném nebo hostitelském OS. Mohou být zdrojové porty stejné u více DNS dotazů?

Podobně jako na obr. A.2.8 by mělo být vidět, že se čísla portů u jednotlivých dotazů liší. Pro každý jeden DNS dotaz se musí rezervovat nové číslo zdrojového portu, aby nedošlo k záměně při následném příjmu DNS odpovědí.



Obr. A.2.8: Dva DNS dotazy s odlišnými čísly zdrojových portů ve virtualizovaném OS.

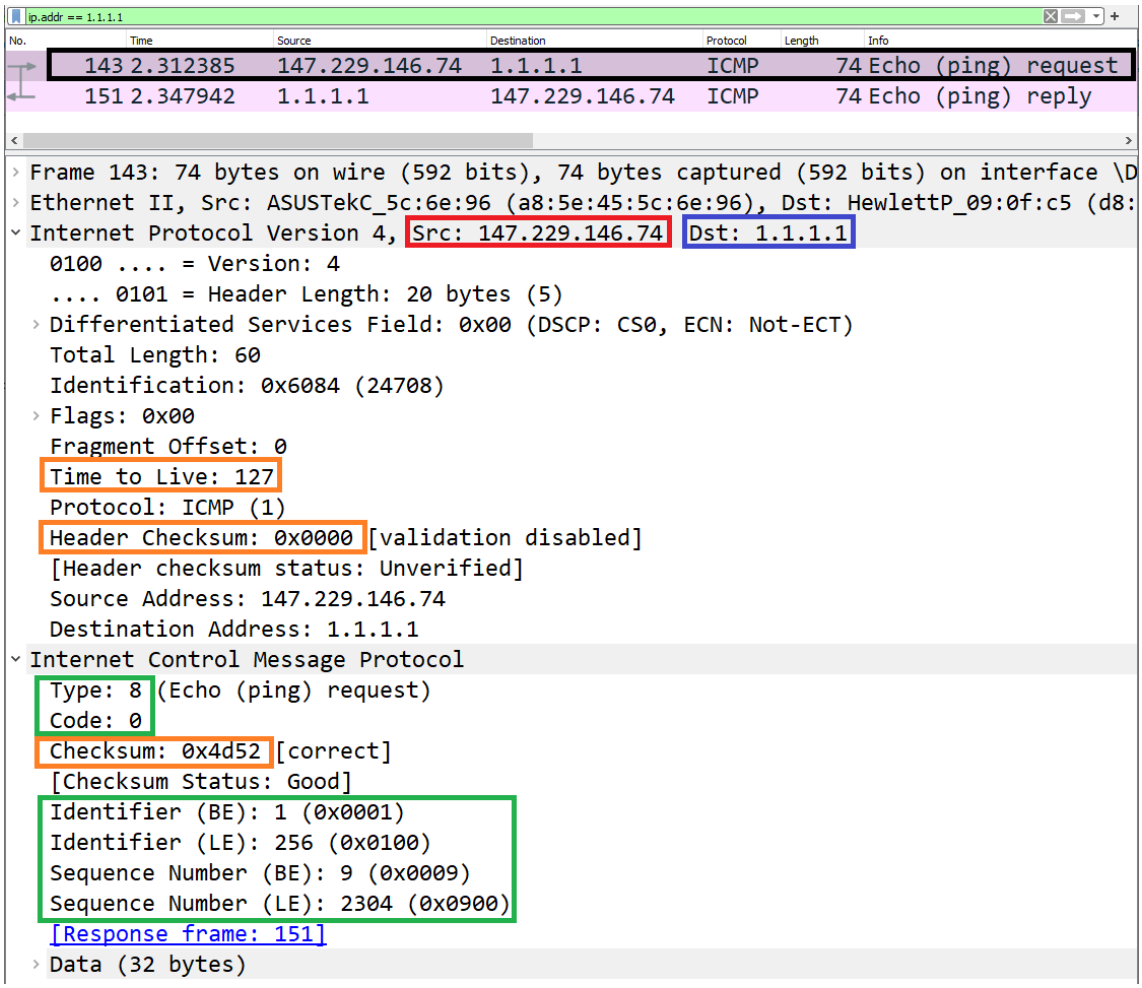


Obr. A.2.9: Dva DNS dotazy s odlišnými čísly zdrojových portů v hostitelském OS.

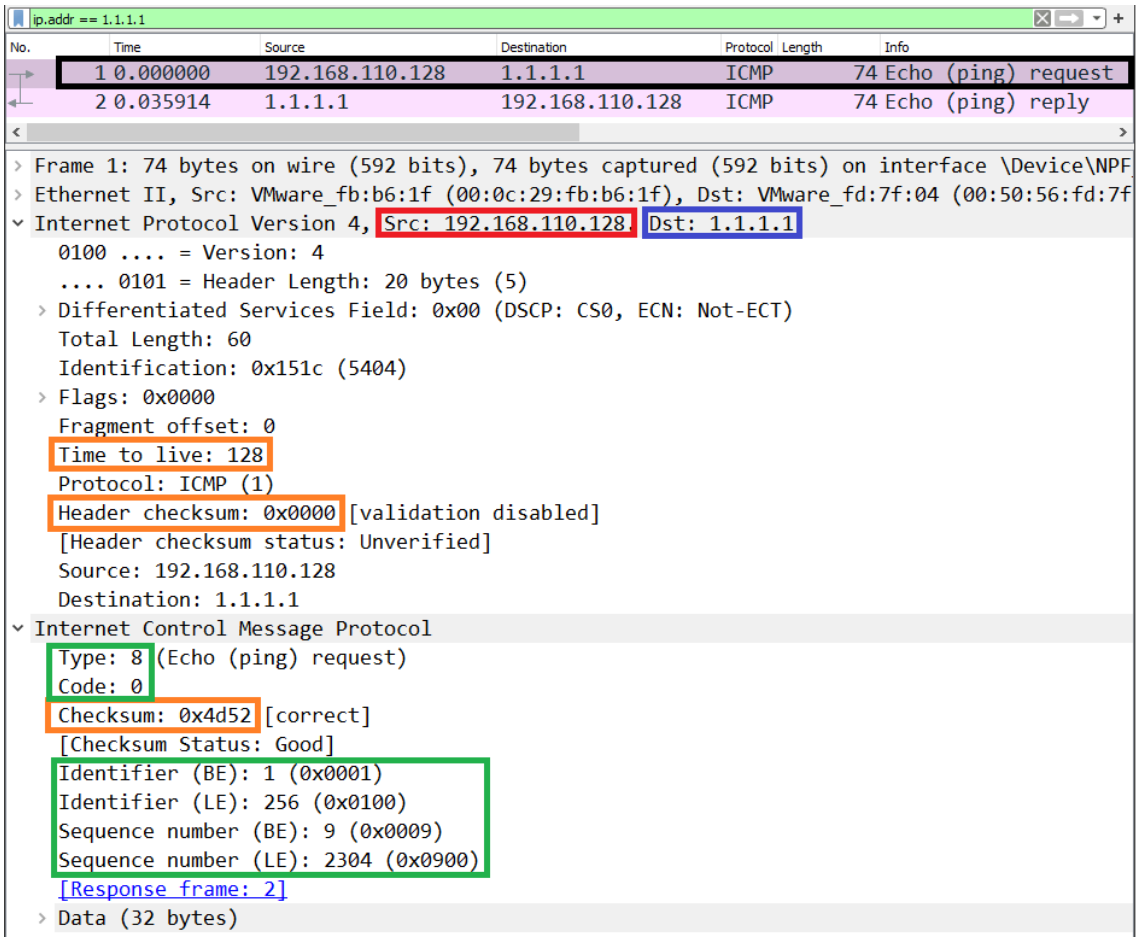
A.2.6 Zachycení ICMP paketů s překladem NAT

Nyní se od DNS paketů přesuneme k paketům nesoucím data protokolu ICMP. Pakety tohoto protokolu můžeme generovat např. pomocí příkazové řádky a příkazu `ping`. Cílovým uzlem zůstane Cloudflare DNS server a celý příkaz pro generování ICMP dotazů bude tedy vypadat takto: `ping 1.1.1.1`. Zapněte zachytávání paketů v obou instancích programu Wireshark, podobně jako v předchozích případech u DNS dotazů a následně zadejte příkaz do příkazové řádky ve virtualizovaném OS. Pro zobrazení pouze požadované komunikace použijte filtr (`ip.addr == 1.1.1.1`).

Zachycené pakety by měly vypadat podobně jako na obr. A.2.10 a A.2.11. Vidíme zde klasické parametry ICMP protokolu jako je identifikátor, sekvenční číslo a hodnoty typu a kódu chyby přenášené ICMP informace. Tyto hodnoty se při překladu nemění a jejich popisu se věnuje dřívější samostatná úloha. Co je důležité vzhledem k NAT překladu je, že se opět mění sekvenční číslo (checksum) a zdrojová IP adresa, podobně jako v případě DNS dotazu. Kromě těchto hodnot se také mění hodnoty IP záhlaví jako je `Header checksum` a `TTL`. Tyto hodnoty se mění i v předchozích případech u protokolů UDP a TCP, ale podrobněji se na tyto parametry podíváme až zde, ve spojení s protokolem ICMP.

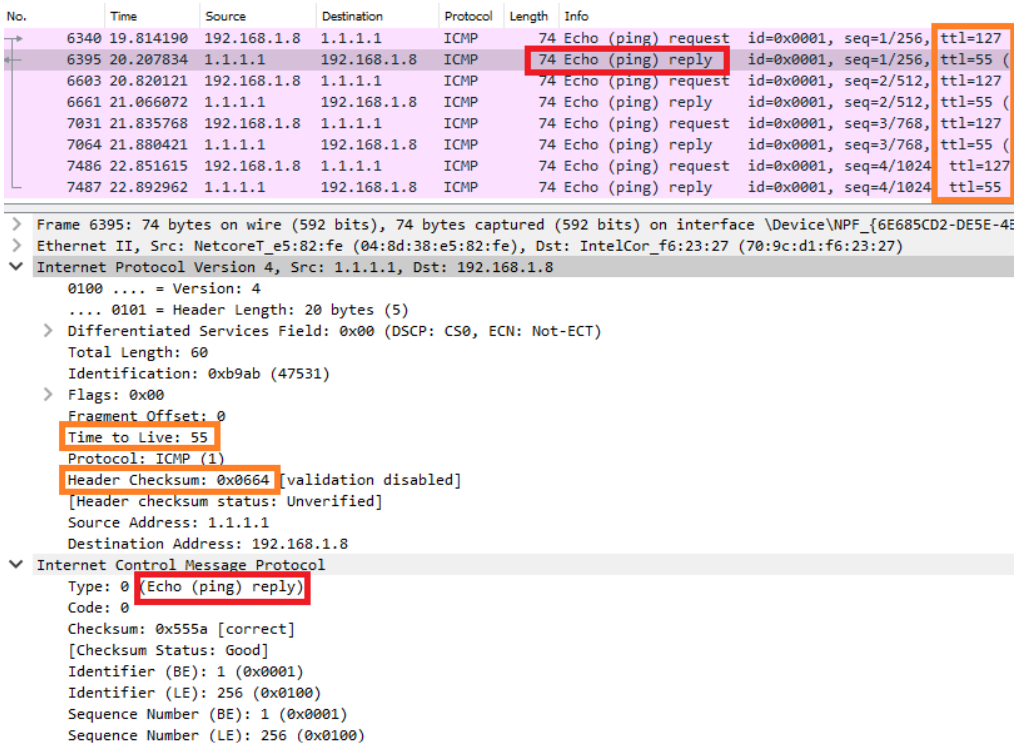


Obr. A.2.10: ICMP dotaz zachycený ve virtualizovaném OS před překladem NAT.

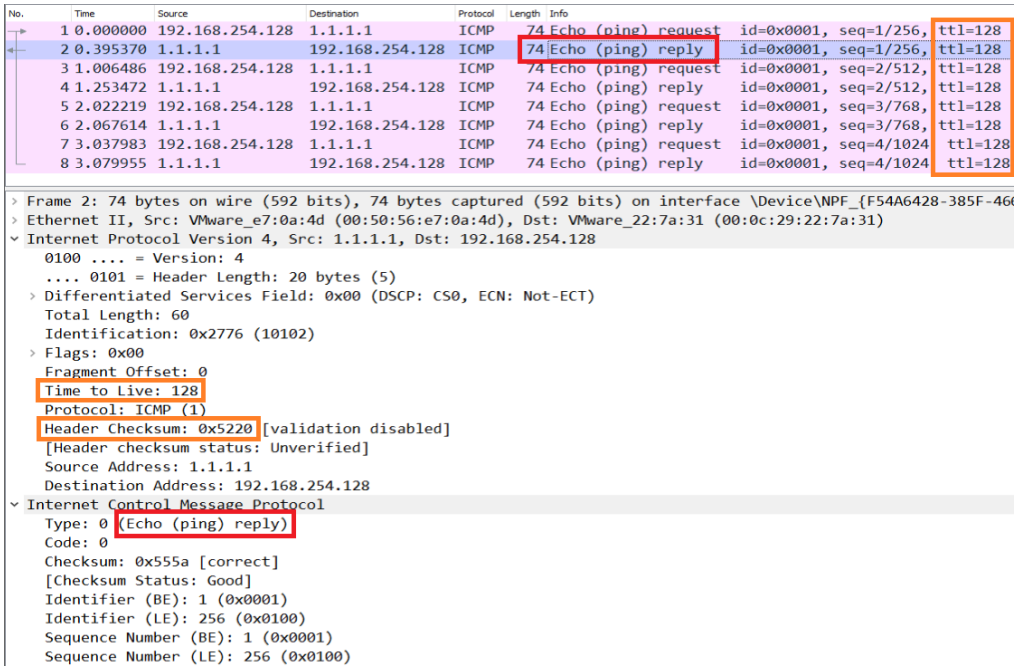


Obr. A.2.11: ICMP dotaz zachycený ve hostovském OS po překladu NAT.

Zkontrolujte hodnotu `Header checksum` v IP záhlaví u paketů s ICMP dotazem i odpovědí v obou instancích programu Wireshark. Podobně jako na obr. A.2.10 a A.2.11 by mělo být vidět, že u ICMP dotazů je hodnota kontrolního součtu záhlaví nulová. Je to tím, že tento součet je sestavován až při odesílání paketu na síťové kartě a ve Wiresharku se tedy hodnota zatím neukazuje. U ICMP odpovědí už by hodnota kontrolního součtu záhlaví neměla být nulová. V našem případě na obr. A.2.12 je tato hodnota v hostovském OS 0x0664. Po NAT překladu se hodnota součtu změní, protože se změní i IP adresa, která je obsažena v záhlaví. Ve virtualizovaném OS je tedy hodnota kontrolního součtu záhlaví 0x5220, viz obr. A.2.13.



Obr. A.2.12: ICMP odpověď zachycená v hostovském OS před překladem NAT a měnící se hodnoty TTL pro jednotlivé ICMP dotazy a odpovědi.



Obr. A.2.13: ICMP odpověď zachycená ve virtualizovaném OS po překladu NAT.

Hodnota TTL je ve virtualizovaném OS u všech ICMP paketů stejná a to většinou 128. U ICMP dotazů je tato hodnota jasná, protože paket byl ve virtualizovaném OS vytvořen a tato hodnota mu byla přidělena jako výchozí. Následně je paket odeslán do hostitelského OS, kde je proveden NAT překlad a hodnota TTL je v našem případě na obr. A.2.12 snížena na 127. U ICMP odpovědi je hodnota TTL v hostitelském OS 55. Výchozí hodnota na straně Cloudflare serveru byla s největší pravděpodobností 64. Ovšem hodnota TTL 55 není následně snížena na 54 a odeslána do virtualizovaného OS, ale je nahrazena novou výchozí hodnotou 128. Toto nahrazení ovšem nesouvisí s klasickým NAT překladem tak, jak ho známe. Ten překládá pouze IP adresy a porty. Jde o změnu hodnoty TTL v rámci virtualizačního programu VMware. Porty se u ICMP nepřekládají, jelikož je tento protokol nevyužívá.

Úkoly:

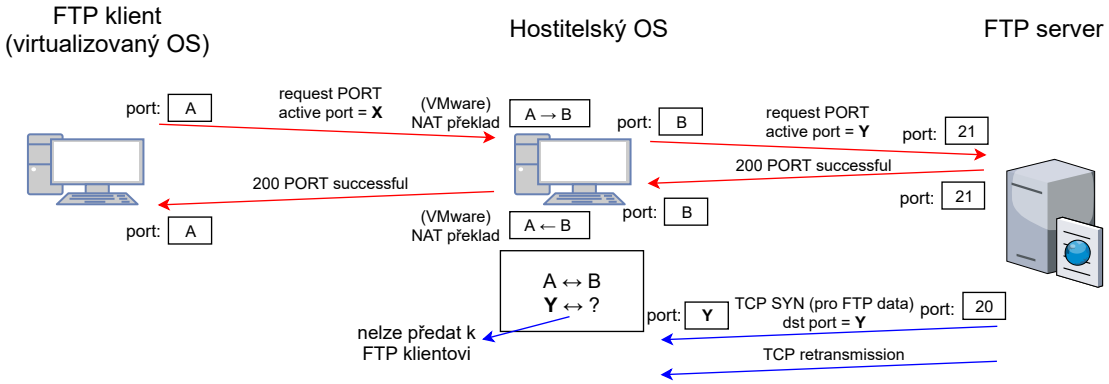
(5) Který parametr využívá protokol ICMP k rozlišení několika současných ICMP komunikací místo portů?

A.2.7 Připojení k FTP serveru a analýza FTP komunikace

Tato část se bude zabývat komunikací mezi FTP klientem, který využívá NAT překlad adres a FTP serverem. FTP (File Transfer Protocol) je protokol využívaný pro vzdálený přenos souborů přes počítačovou síť. Kromě známých bezpečnostních problémů, kdy klasická verze FTP umožňuje po zachycení paketů získat přihlašovací údaje klienta, může nastat problém i při použití NAT překladu. Záleží však na použitém způsobu překladu adres a portů. U FTP je možnost zvolit aktivní nebo pasivní režim. Každý z těchto režimů se chová odlišně při současně aktivním NAT překladu kdekoliv na trase mezi klientem a serverem. Volba režimu se týká pouze datového spojení (port 20 na straně serveru), nikoliv řídícího spojení (port 21).

U aktivního režimu si sám klient určuje náhodné číslo portu v paketu s textovou zprávou PORT, ke kterému se má server připojit a následně server inicializuje komunikaci směrem k tomuto portu. Což způsobí problém při NAT překladu, protože tabulka pro překlad neví, který port byl zvolen pro příjem na straně klienta. Vzhledem k principu fungování Source NATu a tomu, že datová komunikace začíná na straně serveru je jasné, že datové spojení nebude vytvořeno. Source NAT je totiž založen na tom, že první paket musí být přenesen z vnitřní sítě NAT, nikoliv ze strany serveru směrem k síti s NAT překladem.

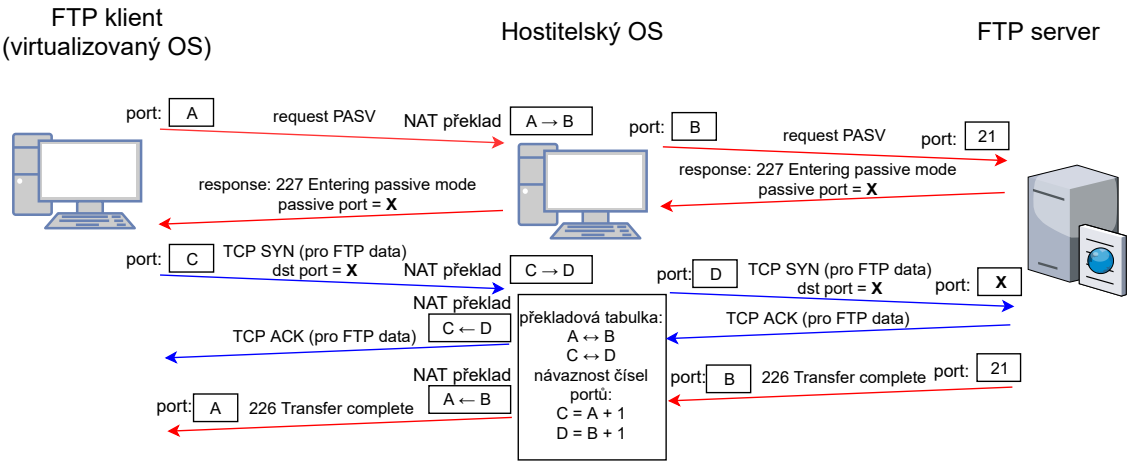
Zároveň komunikace na tento port začíná ze strany serveru, takže NAT při přijetí paketu od serveru neví, na který port a adresu jej má přeložit. FTP komunikace tak selže a připojení k severu se nezdaří. Schéma FTP komunikace při zvoleném aktivním režimu zobrazuje obr. A.2.14.



Obr. A.2.14: Schéma FTP komunikace při použití aktivního režimu a současném využití NAT překladu mezi klientem a serverem.

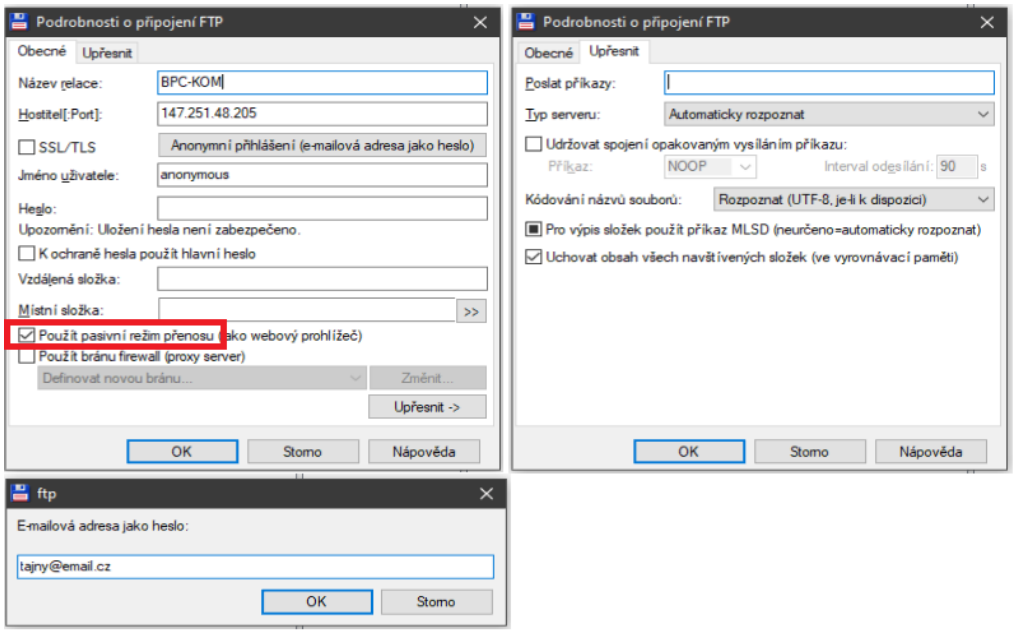
Oproti tomu u pasivního režimu si klient zažádá o port pro pasivní režim v paketu s textovou zprávou PASV a server mu v odpovědi zašle číslo portu, na které se klient může připojit. Následně sám klient komunikaci směrem k serveru na zvolený port inicializuje. Tabulka pro NAT překlad tedy uloží číslo zvoleného portu a následnou odpověď ze strany serveru již bude umět přeložit a FTP komunikace tedy bude úspěšná. Schéma FTP komunikace při zvoleném pasivním režimu zobrazuje obr. A.2.15. Z obrázku je zřejmé, že všechna TCP spojení začínají zprávou od klienta, což v případě Source NAT není žádný problém. V následujících krocích oba režimy vyzkoušíte a také proběhne analýza rozdílů při NAT překladu u FTP a např. DNS.

Ve virtualizovaném OS spusťte program Total Commander, jeho ikona se nachází na hlavním panelu. Po spuštění programu se zobrazí hlavní obrazovka programu. Tento program umožňuje spravovat soubory a adresáře a mimo jiné umí pracovat i se vzdálenými soubory přes FTP protokol, čehož bude využito v této úloze.



Obr. A.2.15: Schéma FTP komunikace při použití pasivního režimu a současném využití NAT překladu.

Pomocí volby menu `Síť > Protokol FTP - Připojit k serveru` budeme chtít zobrazit soubory uložené na vzdáleném FTP serveru. V nově otevřeném okně vytvoříme nové připojení k FTP serveru s adresou `147.251.48.205` pomocí tlačítka `Nové připojení...` Název relace zvolte `BPC-KOM`, hostitel bude zmiňovaná IP adresa FTP serveru 147.251.48.205 (ftp.muni.cz). Následně pomocí tlačítka `Anonymní přihlášení` zvolte jako heslo e-mailovou adresu `tajny@email.cz`. Zbytek parametrů ponechejte ve výchozím nastavení tak, jak je zobrazeno na obr. A.2.16.



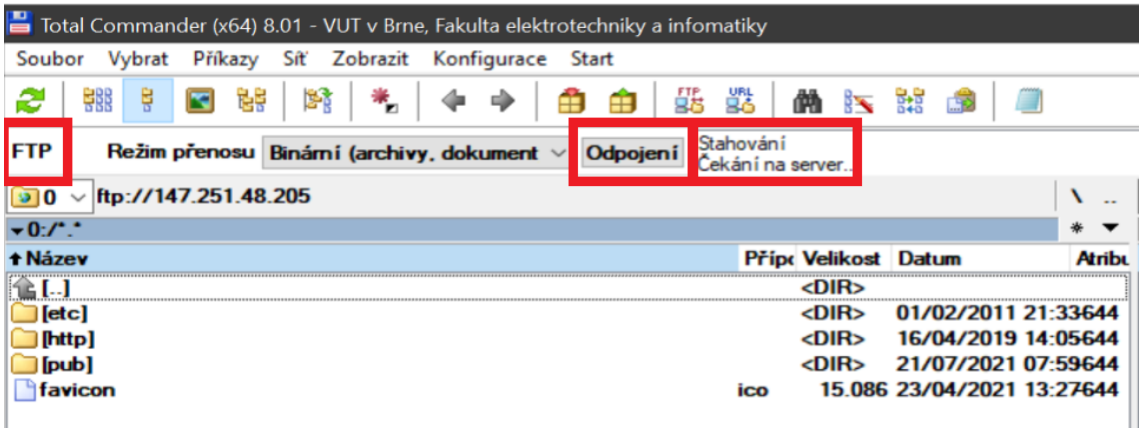
Obr. A.2.16: Připojení k FTP serveru v programu Total Commander.

Zkontrolujte hlavně zvolenou položku `Použit pasivní režim` a následně nastavení potvrďte tlačítkem OK. V okně Připojení k serveru FTP se vytvořila položka k připojení k FTP serveru s názvem BPC-KOM. Položku označte, zvolte možnost `Připojit` a otestujte funkčnost spojení s FTP serverem.

Úkoly:

(6) Jaký číselný kód má FTP zpráva s označením „Transfer complete“? Zprávy o stavu připojení k FTP serveru se zobrazují v horní části programu Total Commander, viz obr. A.2.17.

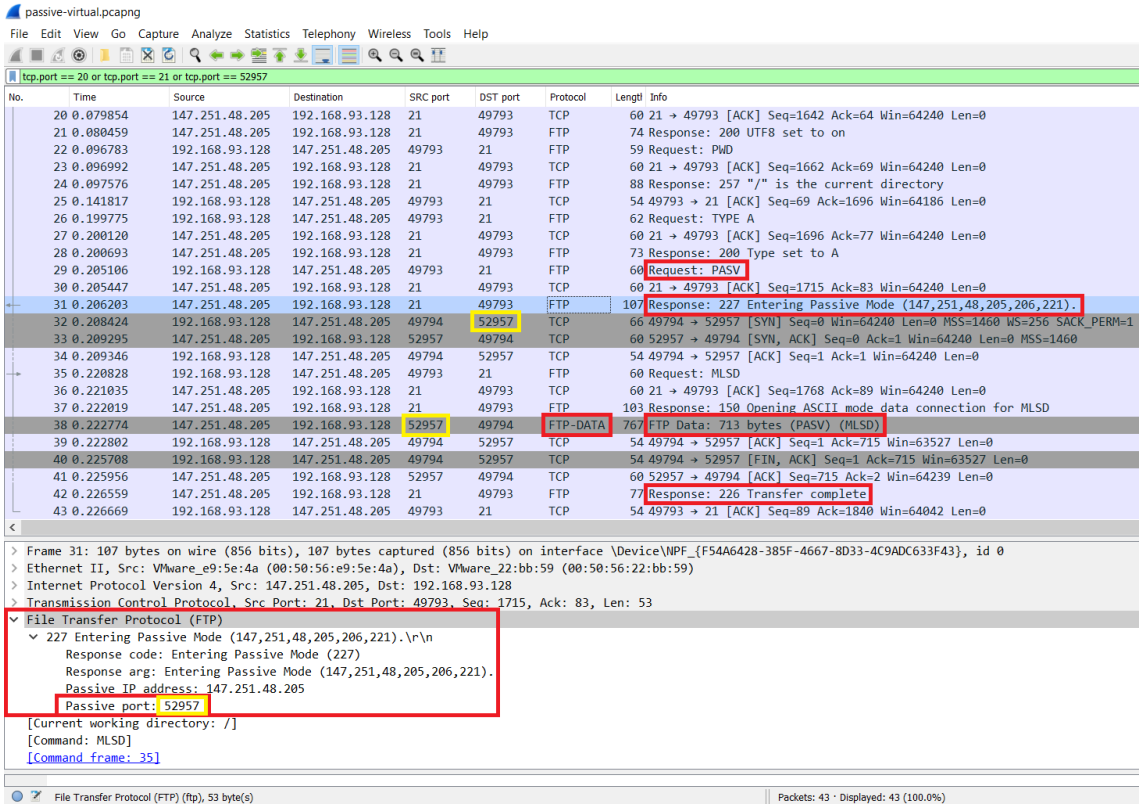
Nyní se od FTP serveru odpojte přes tlačítko `Odpojení`, které je zvýrazněno na obr. A.2.17, případně funguje také klávesová zkratka `Ctrl+Shift+F`. Zapněte zachytávání paketů ve virtualizovaném i hostitelském OS a následně se opět připojte k FTP serveru přes `Síť > Protokol FTP - Připojit k serveru` nebo klávesovou zkratku `Ctrl+F` a následně tlačíítko `Připojit`. Po úspěšném připojení k serveru zastavte zachytávání paketů a v obou instancích vyfiltrujte TCP pakety využívající porty určené protokolu FTP, tedy porty číslo 20 a 21. Jak v hlavním, tak ve virtuálním OS se nám zobrazí FTP komunikace, včetně potvrzujících TCP paketů s příznaky ACK.



Obr. A.2.17: Informace o stavu připojení k FTP serveru a možnost odpojení se od FTP serveru v programu Total Commander.

Důležitým poznatkem je, že FTP komunikace funguje a v programu Total Commander se nám zobrazí obsah kořenové složky FTP serveru. A to i přesto, že komunikujeme pomocí NAT překladu adres. Je to právě díky zvolenému pasivnímu režimu, který nechává inicializaci datové komunikace na klientovi, čímž dojde ke správnému přeložení portů při NAT překladu. V příkladě uvedeném na obr. A.2.18 (virtualizovaný OS) a A.2.19 (hostitelský OS) si tedy klient zažádá o pasivní režim.

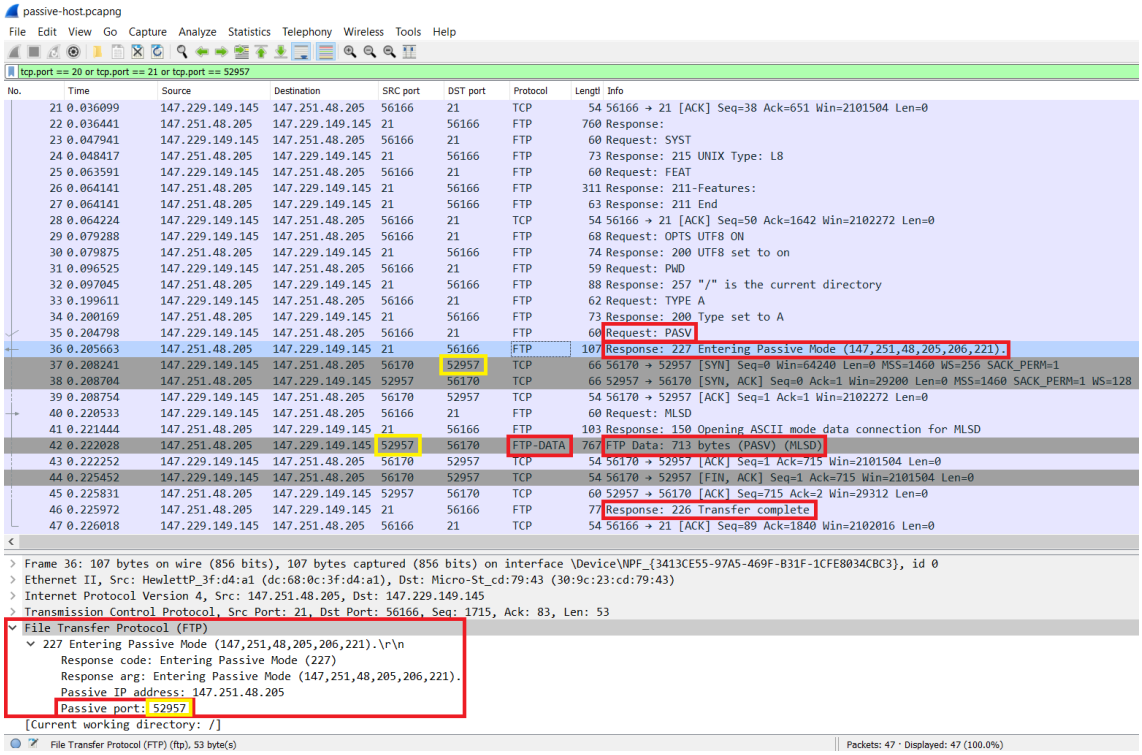
V ukázaném případě jde ve virtualizovaném OS o paket číslo 29, který má zdrojový port 49793 a jako cílový port je uveden port číslo 21, který je na FTP serveru určen k řízení FTP komunikace. V hostitelském OS je zdrojový port přeložen na hodnotu 56166 (paket č. 35). Server tuto žádost obdrží a následně umožní vstup do pasivního režimu a zašle klientovi paket s IP a číslem portu (paket č. 36), kam má směrovat následující datovou FTP komunikaci. Konkrétně v případě uvedeném na obr. A.2.19 jde o pasivní port číslo 52957. Zdrojový a cílový port tohoto paketu jsou shodné jako u paketu s žádostí, pouze v opačném pořadí.



Obr. A.2.18: Zachycené FTP pakety ve virtualizovaném OS při využití pasivního režimu.

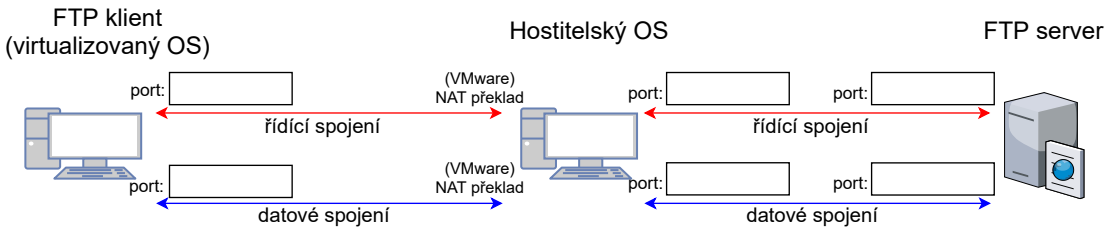
Ve Wiresharku v hostitelském i virtualizovaném OS při zvoleném filtru na porty 20 a 21 nevidíme kompletní pasivní FTP komunikaci, přes tento zjištěný pasivní port. Přidejte tedy do filtru také tento pasivní port, který slouží k přenosu dat ze vzdáleného serveru, podobně jako na obr. A.2.18. Díky úpravě filtru již vidíme celou FTP komunikaci, včetně datového přenosu (paket č. 38 v ukázce z virtualizovaného OS). Tento paket přenáší informace o obsahu kořenové složky FTP serveru a také podrobnosti o jednotlivých souborech v této složce. Ještě před tímto paketem dochází k navazování TCP spojení ze strany klienta a to již za pomoci pasivního portu (v ukázce port č. 52957), který nám FTP server zpřístupnil a jeho číslo zaslal

v paketu označeném jako `Response: 227 Entering Passive Mode`. S pasivním portem komunikuje na straně klienta port číslo 49794 (virtualizovaný OS), který je následně překládán na port číslo 56170 (hostitelský OS). Po ukončení TCP spojení a tím i datového FTP spojení je ještě přenesen paket se zprávou `Transfer complete`, která informuje o úspěšném datovém přenosu a to již opět pomocí řídícího FTP spojení s portem číslo 21 na straně serveru.



Obr. A.2.19: Zachycené FTP pakety v hostitelském OS při využití pasivního režimu.

Při dalším porovnávání zachycených paketů ve virtualizovaném a hostitelském OS zjistíme, že v hostitelském OS bylo na začátku FTP komunikace přijato několik paketů, které nesou uvítací zprávu s textem „Vítejte na FTP serveru Fakulty informatiky...“ a také informace o lokálním čase a datu. V hostitelském OS je tento text rozdělen zhruba do 8 paketů, ale ve virtualizovaném OS jsou všechny tyto informace shromážděny v jednom paketu, který nese zprávu `Response: 230-Hello...` Celý text lze nalézt po rozbalení podrobností o přenášených FTP datech v tomto paketu ve Wiresharku. Číslo pasivního portu je v případě využití pasivního režimu stejné jak ve virtualizovaném, tak v hostitelském OS.



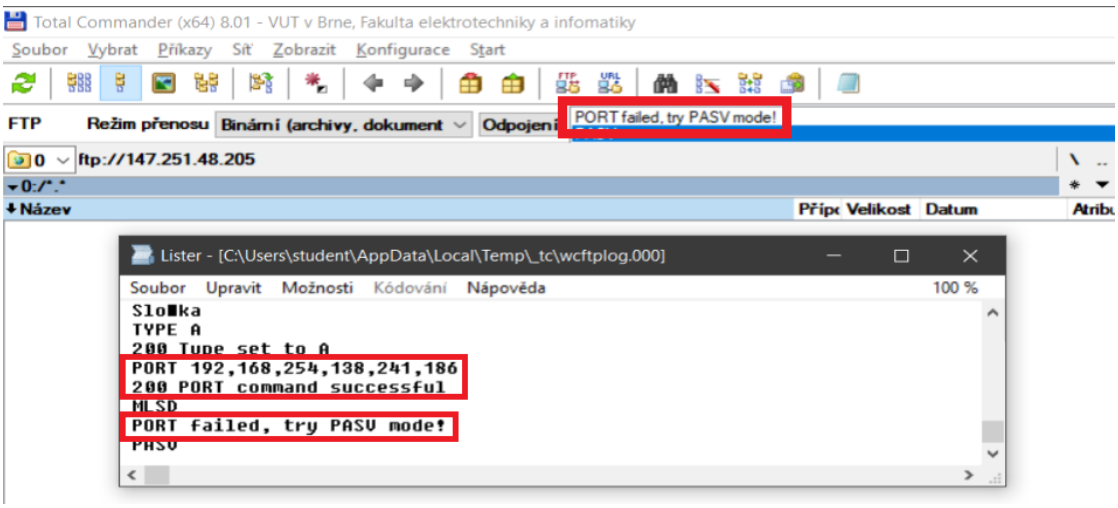
Obr. A.2.20: Šablona pro úkol č. (7) znázorňující FTP komunikaci s překladem NAT.

Úkoly:

(7) Vyplňte čísla portů ze zachycené FTP komunikace při zvoleném pasivním režimu do šablony na obr. A.2.20.

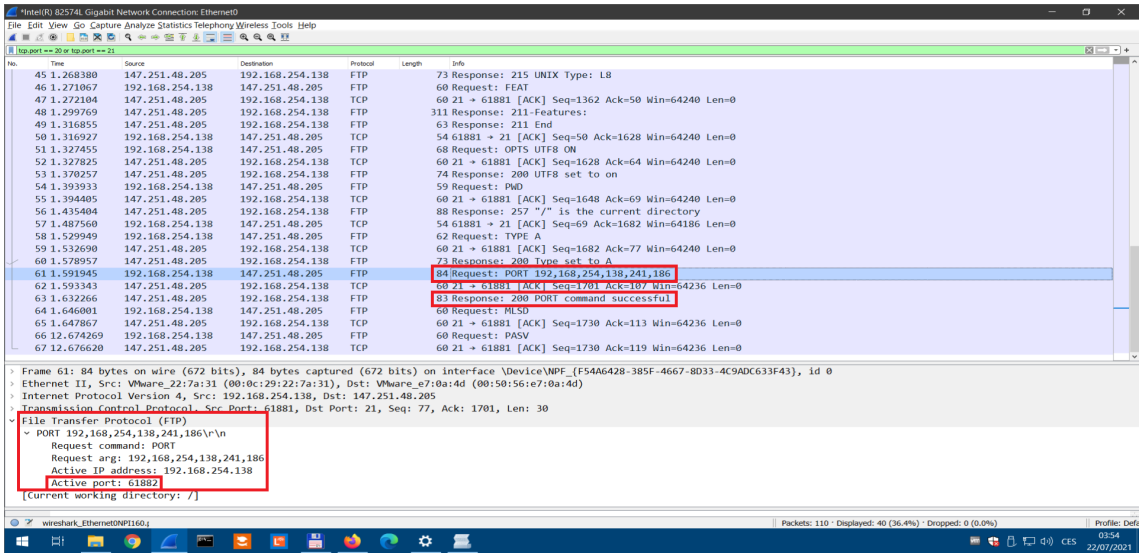
(8) Jaké podrobnosti o souborech jsou přenášeny v paketu označeném jako FTP-DATA. Uveďte alespoň 3 parametry. Nápověda: hledejte v části aplikační vrstvy pojmenované jako Line-based text data.

Nyní budeme postup opakovat, ale zvolíme aktivní režim FTP, respektive nepovolíme režim pasivní. V programu Total Commander tedy přerušíme spojení s FTP serverem tlačítkem `Odpojení`. Následně v nastavení připojení k FTP serveru přes možnost `Upravit` odškrtněte položku `Použít pasivní režim`. Nastavení uložte a zapněte zachytávání paketů v obou instancích programu Wireshark. Dále proveďte připojení k FTP serveru, stejně jako v případě použití pasivního režimu v předešlé části tohoto cvičení. Připojení k serveru se nezdaří, což je ohlášeno chybovou zprávou `PORT failed, try PASV`, viz obr. A.2.21.



Obr. A.2.21: Informace o chybě při připojování k FTP serveru v programu Total Commander.

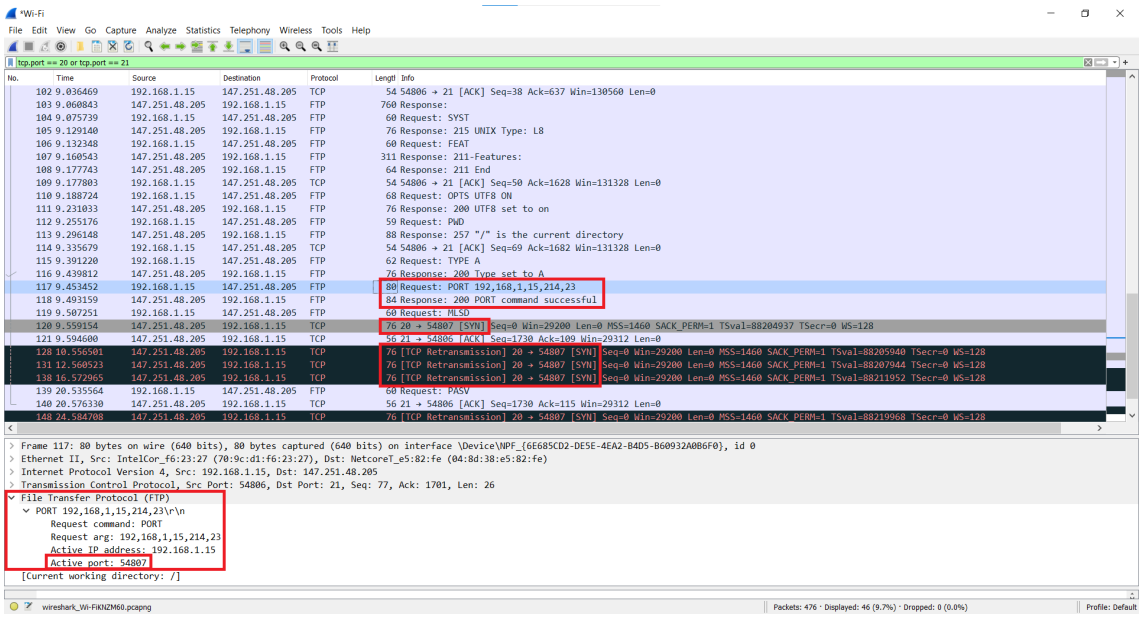
Zastavte zachytávání paketů ve Wiresharku a zadejte stejný filtr jako v předchozí části úlohy. Tedy filtr, který zobrazí pouze pakety využívající porty číslo 20 a 21, podobně jako na obr. A.2.22 a A.2.23. Vidíme, že u aktivního režimu je zpráva PASV nahrazena textovou zprávou PORT, kterou odesílá klient spolu s číslem aktivního portu. Na uvedených obrázcích si můžeme všimnout dalšího rozdílu oproti pasivnímu režimu, kde zůstalo číslo pasivního portu stejné i po NAT překladu, ale u aktivního režimu je číslo aktivního portu také přeloženo. Konkrétně z čísla aktivního portu 61882 u virtualizovaného OS (obr. A.2.22) je hodnota přeložena na číslo aktivního portu 54807 v hostitelském OS (obr. A.2.23).[2]



Obr. A.2.22: Zachycené FTP pakety ve virtualizovaném OS při využití aktivního režimu.

Při detailnějším pohledu na pakety zachycené v hostitelském OS vidíme, že klient odeslal žádost se zprávou PORT obsahující číslo aktivního portu směrem k serveru (paket 117). Tato zpráva byla odeslána z portu 54806 směrem k portu 21 na straně serveru. V paketu číslo 118 následuje odpověď se zprávou `200 PORT command successful`. Zpráva tedy byla serverem přijata. Následně server začíná FTP datovou komunikaci (paket 120) z portu číslo 20 na port, který byl uveden klientem jako aktivní port v paketu č. 117, tedy port 54807. Komunikaci se server snaží iniciovat TCP paketem s příznakem SYN, který začíná navazování spojení. Tento paket se dostane do hostitelského OS, ale následně NAT zjistí, že komunikace na novém portu nepochází z vnitřní sítě NATu, což odporuje pravidlům komunikace přes Source NAT (SNAT).

[2]Ve vašem případě se mohou čísla portů lišit.



Obr. A.2.23: Zachycené FTP pakety v hostitelském OS při využití aktivního režimu.

Komunikace dvou stran totiž u SNATu musí začínat ve vnitřní síti, nikoliv ve vnější síti ze které paket putuje od FTP serveru. Překladová NAT tabulka by ani nevěděla, na který port má být přeložen právě onen aktivní port č. 54807. Komunikace tak nijak nepokračuje směrem do virtualizovaného OS a pakety jsou NATem zahozeny bez jakékoliv další akce. Jelikož server nedostane potvrzení o úspěšném navázání TCP komunikace v podobě TCP paketu s příznakem ACK, tak server posílá paket s příznakem SYN opakovaně (pakety 128,131 a 138). V Total Commanderu je tato situace po chvíli řešena právě vypsáním chybového hlášení `PORT failed, try PASV`, jako na obr. A.2.21. Je tedy ohlášena chyba a zároveň nám program navrhuje, abychom použili pasivní režim FTP datového spojení.

Oproti předchozím případům s protokoly DNS nebo ICMP je tedy u FTP protokolu rozdíl v tom, že v případě využití NAT překladu a aktivního režimu FTP dojde i k překladu parametrů v rámci aplikační vrstvy. Konkrétně u FTP se přeložila hodnota aktivního portu. U protokolů DNS nebo ICMP k tomuto zásahu v rámci aplikační vrstvy nedochází.

Úkoly:

(9) Jak je volena hodnota čísla FTP datového portu v aktivním režimu vzhledem ke zdrojovému portu na straně klienta?

(10) Je možné v zachycené komunikaci vyhledat přihlašovací údaje pro přístup k FTP serveru? Je použití klasického FTP bezpečné?

A.2.8 Analýza SCTP paketů při NAT komunikaci

V této části se bude krátce analyzovat předem zachycená komunikace obsahující pakety protokolu SCTP (Stream Control Transmission Protocol). Ve Wiresharku otevřete soubor `sctp_komunikace.pcapng` přes menu `File > Open`. Soubor se nachází na ploše virtualizovaného OS.[3]

Komunikace je inicializována zařízením, které se nachází v síti s NAT překladem. První pakety obsahují protokol TCP, který je bez problémů přenesen. Je provedeno jak obousměrné navázání spojení, tak samotná TCP komunikace a na závěr i ukončení spojení. V případě použití filtru `sctp` se nám zobrazí pouze dva pakety, kdy první z nich je jediný „skutečný“ SCTP paket s příznakem `INIT`, kterým začíná navazování spojení u SCTP protokolu. Následná komunikace obsahuje pouze chybovou zprávu protokolu ICMP s informací, že zvolený transportní protokol není podporován. (Destination unreachable - Protocol unreachable). SCTP komunikace je touto chybou ukončena a dále nepokračuje. Jedná se o jednu z nevýhod použití NAT překladu, který běžně podporuje pouze použití s transportními protokoly TCP, UDP a částečně také ICMP.

Úkoly:

(11) Jaký typ a kód chyby je uveden u ICMP paketu?

Samotný SCTP paket obsahuje SCTP záhlaví, které je podobné tomu u TCP nebo UDP protokolu. Je zde tedy uvedeno číslo zdrojového a cílového portu (source a destination port), kontrolní součet (checksum) a také tzv. ověřovací značka (verification tag). Dále již paket obsahuje pouze hlavní datovou část, která se nazývá `chunk`. Podrobnější objasnění SCTP protokolu je nad rámec tohoto předmětu, takže se mu již nadále věnovat nebudeme.

Úkoly:

(12) Jaký zdrojový a cílový port je uveden u jediného SCTP paketu v předem zachyceném souboru?

[3] Protože Windows SCTP běžně nepodporují, použijeme předpřipravený soubor zachycený v jiném (linuxovém) OS.

B Kompletní návod pro druhý vytvořený simulační scénář

ÚLOHA č. 2

Překlad doménových jmen pomocí DNSSEC
a DNS over HTTPS a jejich analýza.

B.1 Teoretický úvod

Běžný protokol DNS zajišťuje pouze překlad doménového jména na IP adresu a případně naopak. Nic však nebrání podvržení poskytovaných informací a proto se v tomto cvičení budeme věnovat zabezpečené variantě DNS protokolu v podobě rozšíření **DNSSEC** (Domain Name System Security Extensions) a také šifrované variantě pro DNS komunikaci v podobě protokolu **DoH** (DNS over HTTPS). K tomu bude využit virtuální stroj VMware, který bude komunikovat skrze aplikaci DNS Klient s DNS servery. Úloha obsahuje návod a popis komunikace při dotazech na klasickou, ale i na neexistující doménu a také na doménu s podvrženým podpisovým záznamem RRSIG. Po zachycení síťové komunikace se budou jednotlivé záznamy a pakety analyzovat a to jak v části s rozšířením DNSSEC, tak v části, která se věnuje protokolu DNS over HTTPS.

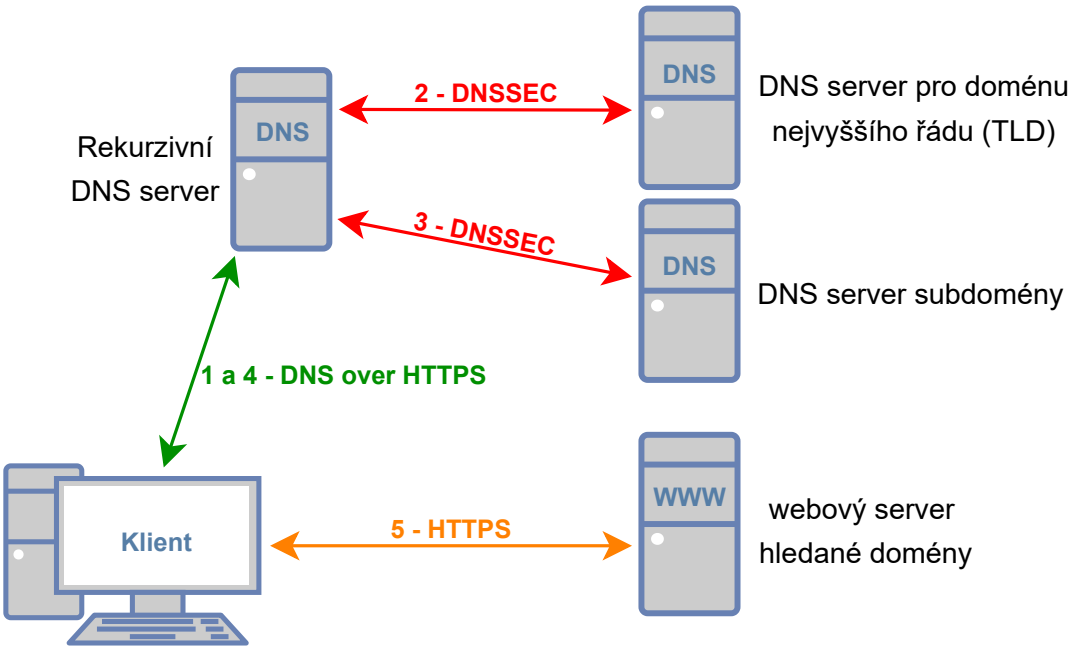
B.1.1 Princip zabezpečení překladu doménových jmen pomocí DNSSEC

DNSSEC je rozšířením pro nám již dobře známý protokol DNS. Rozšíření DNSSEC přidává k informacím v běžném DNS paketu informace o digitálním podpisu. Ověřením tohoto podpisu se zajišťuje autentičnost poskytnutých informací, kterým tak můžeme důvěřovat. Je ale potřeba zdůraznit, že zabezpečení autentičnosti dat je pomocí DNSSEC zajišťováno pouze mezi rekurzivním resolverem a dotazovaným DNS serverem, jak je znázorněno na obr. B.1.1. Rozšíření DNSSEC nezajišťuje komunikaci klienta (stub resolveru) s rekurzivním serverem. Ke klasickým DNS záznamům jako jsou záznamy A, AAAA, NS nebo PTR se tak přidávají další jako např. RRSIG (Resource Record Signature), DNSKEY (DNS Public Key) nebo NSEC (Next Secure) a taktéž v samotném DNS záhlaví se mění hodnoty některých příznakových bitů. Význam těchto záznamů, příznakových bitů a jejich obsah bude rozebrán při samotném řešení tohoto scénáře.

B.1.2 Zabezpečení komunikace klienta s rekurzivním serverem pomocí DNS over HTTPS

DoH (DNS over HTTPS) je protokol, který si bere na starosti zabezpečení komunikace z pohledu integrity a autenticity mezi klienty a rekurzivními servery, viz obr. B.1.1. Na rozdíl od rozšíření DNSSEC protokol DNS over HTTPS zajišťuje i služby důvěrnosti, tedy že přenášené informace může zobrazit pouze jejich ověřený odesílatel a adresát. Důvěrnosti je dosaženo pomocí šifrování, které je zajištěno protokolem TLS (Transport Layer Security). V případě DoH není šifrován přímo

DNS paket, ale DNS paket zapouzdřený pomocí HTTP (Hypertext Transfer Protocol) záhlaví. Zapouzdřením paketů s protokoly DNS, HTTP a TLS tedy vzniká šifrovaný DNS over HTTPS paket.

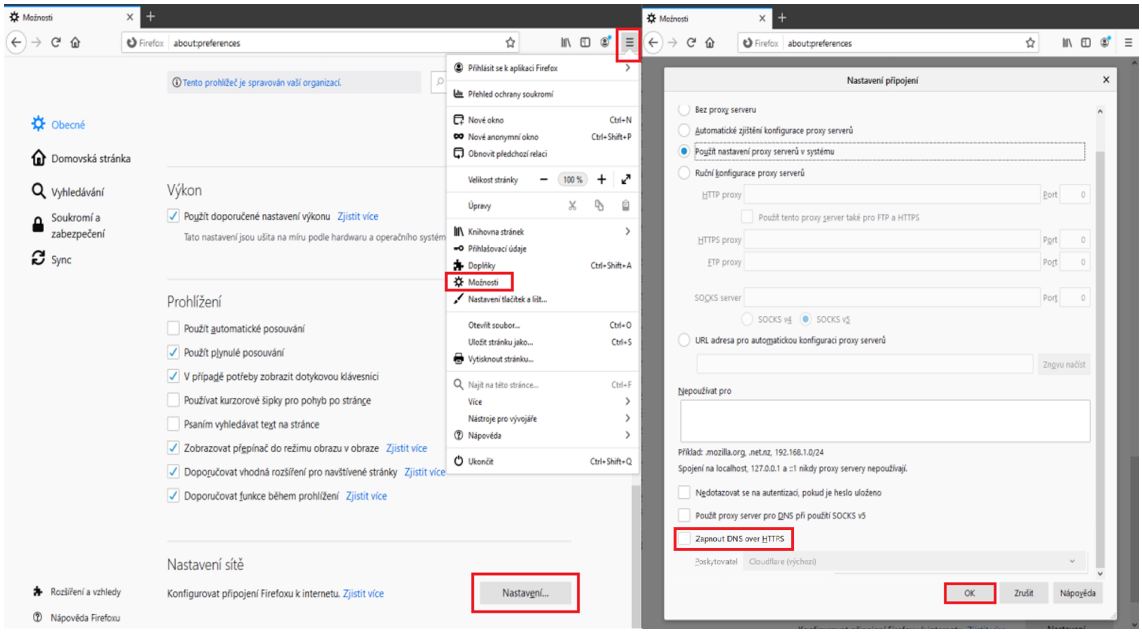


Obr. B.1.1: Schéma zabezpečené DNS komunikace pomocí rozšíření DNSSEC a protokolu DNS over HTTPS a následná zabezpečená komunikace s webovým serverem, která je zajištěna protokolem HTTPS.

B.2 Realizace scénáře

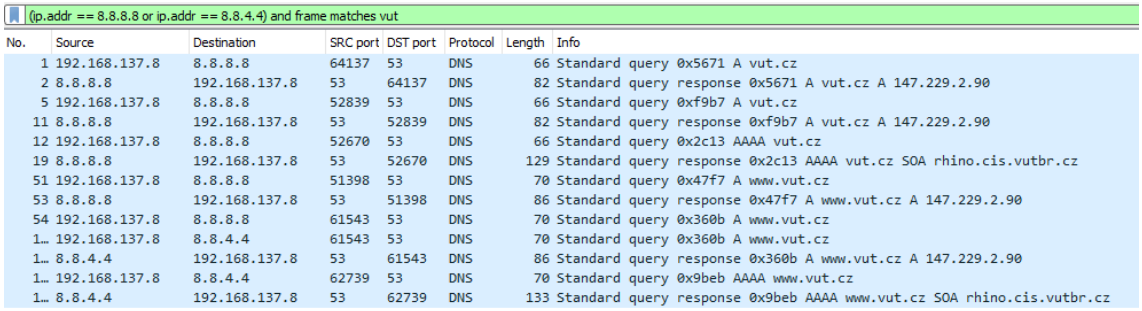
B.2.1 Základní DNSSEC komunikace a její analýza

Nejdříve se pokusíme zachytit DNSSEC komunikaci s využitím webového prohlížeče Firefox. V něm musíme nejdříve zkontrolovat a případně deaktivovat možnost, která umožňuje skrýt DNS komunikaci pomocí protokolu DNS over HTTPS. Ve virtualizovaném operačním systému otevřete webový prohlížeč Firefox a přejděte do nastavení, obdobně jako na obr. B.2.1, přes `Menu (3 čárky v pravém horním rohu) > Možnosti > Nastavení sítě > Nastavení...` Zde zkontrolujte, že položka „zapnout DNS over HTTPS“ je `deaktivovaná`. Pokud bylo využití DoH zapnuto, tak položku odškrtněte, potvrďte tlačítkem „OK“ a restartujte webový prohlížeč.



Obr. B.2.1: Deaktivace protokolu DNS over HTTPS ve webovém prohlížeči.

Dále si ve Wiresharku přichystejte filtr pro Google DNS IP adresy (8.8.8.8 a 8.8.4.4) a zapněte zachytávání síťového provozu. Ve webovém prohlížeči načtěte webovou stránku `vut.cz` a poté zastavte zachytávání paketů. Pokud je výpis paketů ve Wiresharku nepřehledný, upravte filtr ještě pomocí řetězce `frame matches vut`. Po přidání tohoto textu do pole pro vyfiltrování paketů, by měl výpis paketů vypadat podobně jako na obr. B.2.2. Vidíme zde pouze DNS dotazy a odpovědi související s doménovým jménem vut.cz



(ip.addr == 8.8.8.8 or ip.addr == 8.8.4.4) and frame matches vut

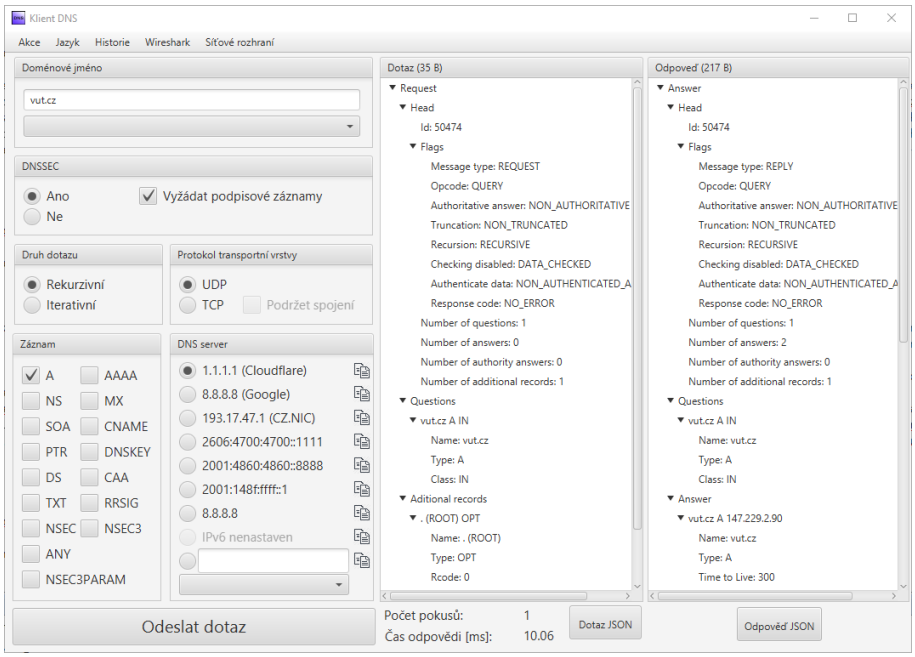
No.	Source	Destination	SRC port	DST port	Protocol	Length	Info
1	192.168.137.8	8.8.8.8	64137	53	DNS	66	Standard query 0x5671 A vut.cz
2	8.8.8.8	192.168.137.8	53	64137	DNS	82	Standard query response 0x5671 A vut.cz A 147.229.2.90
5	192.168.137.8	8.8.8.8	52839	53	DNS	66	Standard query 0xf9b7 A vut.cz
11	8.8.8.8	192.168.137.8	53	52839	DNS	82	Standard query response 0xf9b7 A vut.cz A 147.229.2.90
12	192.168.137.8	8.8.8.8	52670	53	DNS	66	Standard query 0x2c13 AAAA vut.cz
19	8.8.8.8	192.168.137.8	53	52670	DNS	129	Standard query response 0x2c13 AAAA vut.cz SOA rhino.cis.vutbr.cz
51	192.168.137.8	8.8.8.8	51398	53	DNS	70	Standard query 0x47f7 A www.vut.cz
53	8.8.8.8	192.168.137.8	53	51398	DNS	86	Standard query response 0x47f7 A www.vut.cz A 147.229.2.90
54	192.168.137.8	8.8.8.8	61543	53	DNS	70	Standard query 0x360b A www.vut.cz
1…	192.168.137.8	8.8.4.4	61543	53	DNS	70	Standard query 0x360b A www.vut.cz
1…	8.8.4.4	192.168.137.8	53	61543	DNS	86	Standard query response 0x360b A www.vut.cz A 147.229.2.90
1…	192.168.137.8	8.8.4.4	62739	53	DNS	70	Standard query 0x9beb AAAA www.vut.cz
1…	8.8.4.4	192.168.137.8	53	62739	DNS	133	Standard query response 0x9beb AAAA www.vut.cz SOA rhino.cis.vutbr.cz

Obr. B.2.2: Zachycené DNS pakety v programu Wireshark bez využití protokolu DoH.

Úkoly:
(1) Je možné zobrazit podrobnosti u jednotlivých DNS dotazů a odpovědí?

Nyní do filtru přidejte zjištěnou adresu pro doménové jméno vut.cz. Kromě DNS paketů by se měl objevit také klasický šifrovaný HTTPS provoz, skrytý v TCP a TLS paketech. Avšak DNSSEC komunikaci při použití prohlížeče nevidíme. Jsou využity pouze klasické DNS dotazy a odpovědi a na využití DNSSEC komunikace se spoléhá až mezi rekurzivním serverem a DNS servery. Vkládáme tak důvěru v rekurzivní server a jeho schopnosti ověřit původ zjištěných informací. Při cestě od rekurzivního serveru k našemu prohlížeči již žádná forma zabezpečení neprobíhá. Abychom mohli analyzovat DNSSEC komunikaci, budeme muset využít aplikaci Klient DNS.

Ve virtualizovaném operačním systému zapněte aplikaci Klient DNS. Tato aplikace bude generovat DNS dotazy včetně žádostí o podpisové záznamy, které zajišťují bezpečnost DNSSEC. V hlavním menu aplikace zvolte možnost `DNS`. Objeví se obrazovka jako na obr. B.2.3. Do pole v levém horním rohu vyplňte dotazované doménové jméno `vut.cz`. Oproti běžné DNS komunikaci chceme využít DNSSEC nadstavbu, je tedy potřeba v sekci DNSSEC zvolit určitou konfiguraci bitů, kterými se DNSSEC odlišuje od klasického DNS. Ponechejte odškrtnutou možnost `CD` (Checking Disabled, 0 = Non-authenticated data: unacceptable), čímž řekneme, že DNSSEC odpověď musí být ověřená a dále musíme zaškrtnout položku bitu `AD` (Authenticate Data), aby bylo v DNS dotazu jasné, že chceme využít DNSSEC a u DNSSEC odpovědi byla u dat zkontrolována autenticita pomocí přiloženého RRSIG záznamu. A je potřeba také zaškrtnout bit `DO` („DNSSEC OK“). Bez této volby by došlo ke kontrole podpisu na úrovni místního DNS serveru, ale k samotnému klientovi by se data s podpisem nedostala a neměli bychom co analyzovat. Dále zvolte DNS server s IP adresou 1.1.1.1 (Cloudflare) a ponechte zvolený záznam typu A, podobně jako na obr. B.2.3. Zapněte program Wireshark a v něm zachytávání aktuální síťové komunikace. V aplikaci Klient DNS odešlete

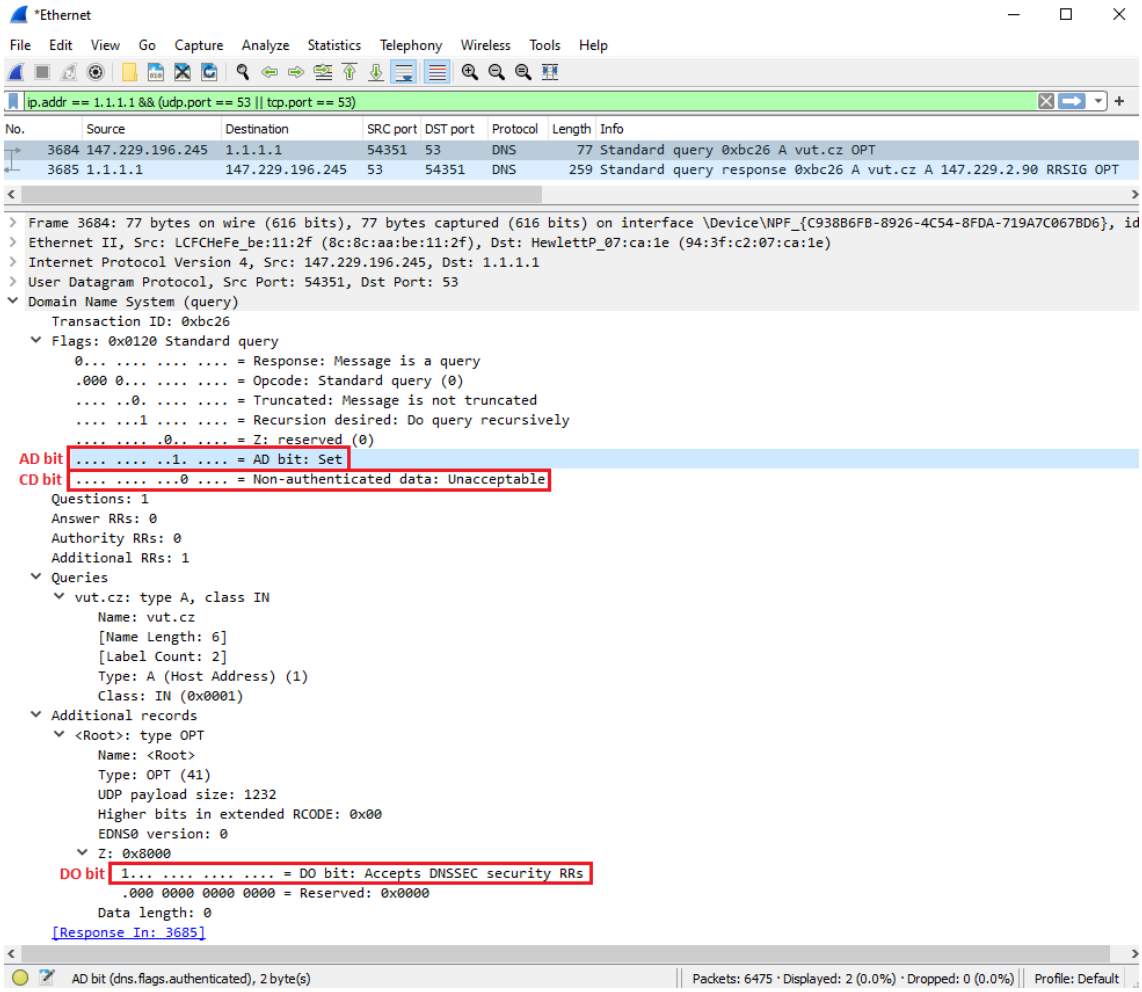


Obr. B.2.3: Nastavení aplikace Klient DNS pro odeslání DNSSEC dotazu.

DNS požadavek tlačítkem „Odeslat dotaz“.

Po zachycení DNS paketů v programu Wireshark zastavte zaznamenávání komunikace a využijte vhodný filtr pro odstranění nepotřebné komunikace. Můžete také využít funkci aplikace Klient DNS, kdy v horním menu zvolíte možnost `Wireshark > IP adresa s UDP nebo TCP jako filtr` a následně kliknutím na ikonu vedle IP adresy DNS serveru filtr zkopírujete do schránky.

Po vyfiltrování DNS paketů, by se ve výpisu měly objevit pouze dva pakety, podobně jako na obr. B.2.4. Prvním paketem je DNS dotaz, který se oproti běžnému DNS paketu odlišuje pouze v pár bitech. První dva bity které určují, zda chceme provést autentizaci či nikoliv, jsou bity v záhlaví paketu v části `DNS > Flags > AD bit` (Authenticate Data) a také `CD bit` (Checking Disabled - Non-authenticated data: unacceptable). V případě, že požadujeme použití DNSSEC, musí být AD bit a tedy autentizace dat nastavena na hodnotu 1 a CD bit na hodnotu 0. Poslední bit, který může být přidán oproti klasickému DNS je DO („DNSSEC OK“) bit, který se nachází v rozšiřujících záznamech. Ve Wiresharku konkrétně v sekci `DNS > Additional records > <Root>: type OPT > Z: 0x8000 > DO bit`. Pokud je tento bit uveden, tak je nastaven na hodnotu 1 což znamená, že požadujeme aby se k DNS odpovědi (např. záznam typu A) připojil i RRSIG záznam (Resource Record SIGnature), který obsahuje digitální podpis a informace o něm. Pokud tento záznam nepožadujeme, tak se DO bit, respektive celá sekce přidaných záznamů vůbec nevyskytuje.



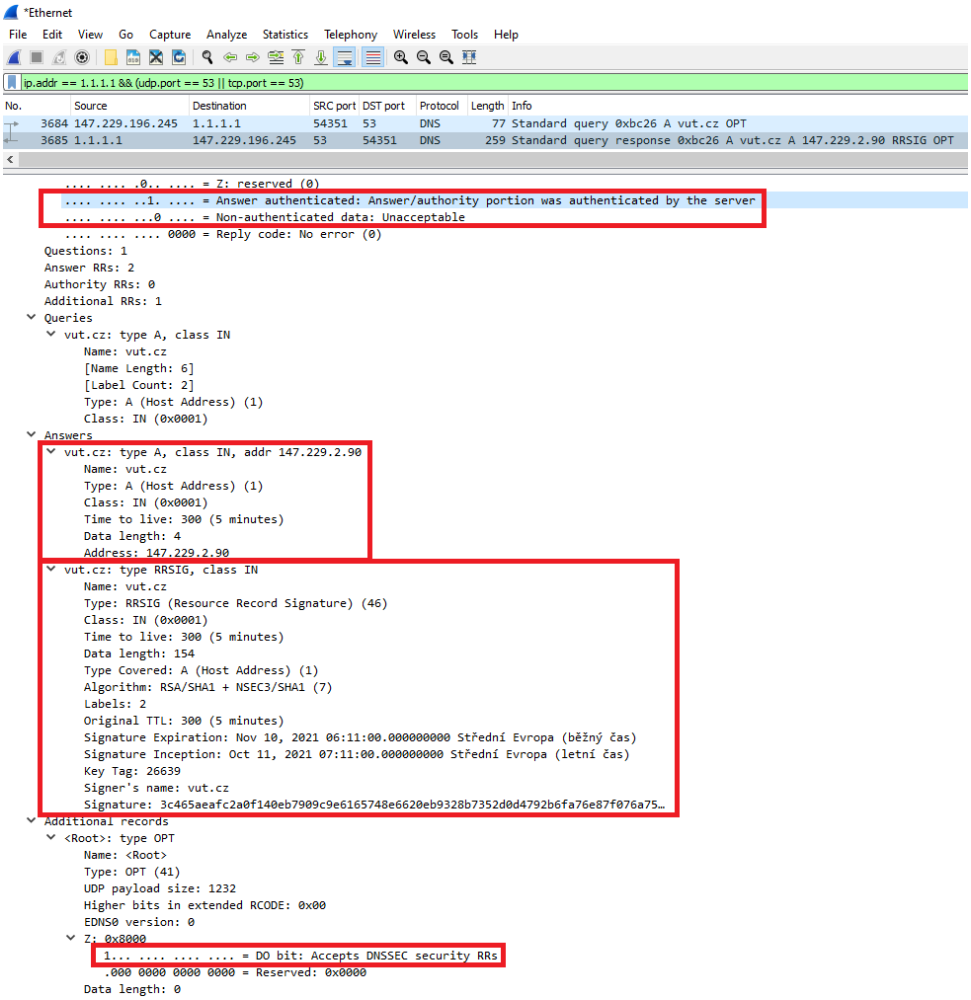
Obr. B.2.4: Zachycená DNS komunikace při využití DNSSECu a zobrazení bitů u DNS dotazu, které zajišťují provedení autentizace a zaslání RRSIG záznamu v DNS odpovědi.

Úkoly:

(2) Jaké hodnoty o digitálním podpisu jsou uvedeny v RRSIG záznamu?

V DNS odpovědi je situace obdobná, s tím že v části `DNS > Answers` je k běžnému záznamu typu A přidán i záznam RRSIG, podobně jako na obr. B.2.5.

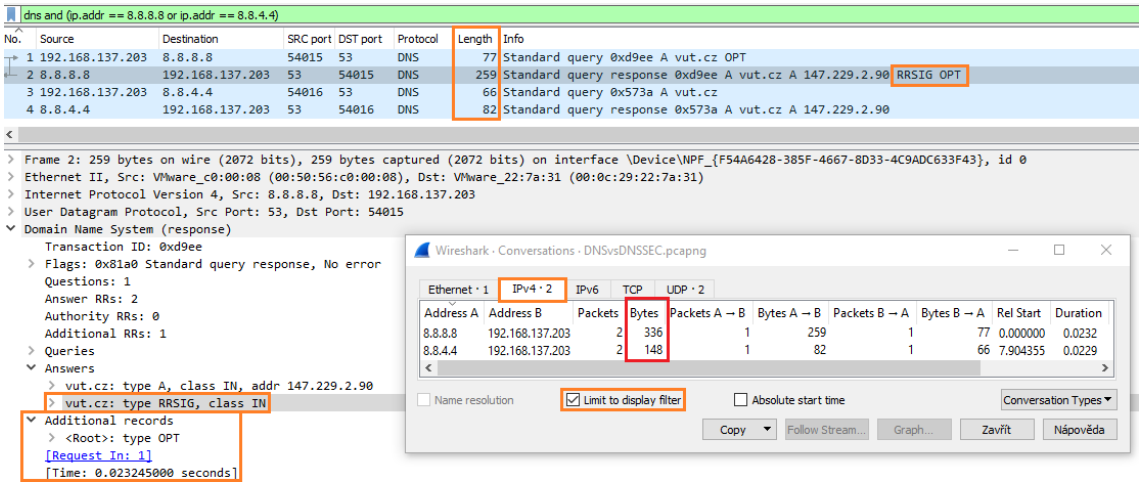
Je důležité zdůraznit, že DNSSEC komunikaci nešifruje, pouze přidává možnost ověřit si jednotlivé záznamy. Můžeme tak získaným informacím o překladu doménového jména důvěřovat, ale stále je možné, že případný útočník komunikaci zachytí a bude například vědět na kterou doménu jsme se dotazovali. Díky tomu, že komunikace není šifrovaná, je také možné si ve Wiresharku zobrazit podrobnosti o jednotlivých záznamech.



Obr. B.2.5: Zachycená DNS komunikace při využití DNSSECu a zobrazení záznamů typu A a RRSIG u DNS odpovědi.

Nyní si srovnáme komunikaci bez použití a s použitím rozšíření DNSSEC a to z pohledu velikosti celkového objemu přenesených bajtů při DNS komunikaci, tedy dohromady DNS dotazu i odpovědi. V aplikaci Klient DNS ponechejte aktivované DNSSEC, jako v předchozím případě (tedy CD=☐, AD=✓, DO=✓) a DNS server změňte na server Google s IP adresou 8.8.8.8. Doménové jméno ponechejte vut.cz. V programu Wireshark si připravte filtr pro dns komunikaci a odešlete DNSSEC dotaz. Následně zachytávání síťové komunikace nezastavujte a v aplikaci Klient DNS deaktivujte DNSSEC rozšíření (tedy CD=☐, AD=☐, DO=☐) a DNS server změňte na IP adresu 8.8.4.4. Odešlete DNS dotaz a zastavte zachytávání paketu ve Wiresharku. Měly by být vyfiltrovány celkem 4 pakety, podobně jako na obr. B.2.6. Pokud se ve vašem výpisu zobrazují další DNS pakety, tak použijte filtr který ponechá pouze DNS pakety s IP adresami 8.8.8.8 a 8.8.4.4. První dva pakety tedy

odpovídají DNSSEC komunikaci, která pro ověření autenticity přece jen potřebuje více přenesených dat, než klasická DNS komunikace, která proběhla v paketech 3 a 4.



Obr. B.2.6: Zachycení a srovnání DNSSEC a DNS komunikace.

Pro srovnání využijeme nástroj pro analýzu `Statistics > Conversations > IPv4`. Pro zobrazení pouze dříve vyfiltrovaných paketů zaškrtněte položku v dolní části okna „Limit to display filter“. Jak můžete vidět i na obr. B.2.6, tak DNSSEC komunikace vyžaduje pro dotaz i odpověď více než dvojnásobný počet bajtů. Konkrétně v ukázkovém případě jde o 336 bajtů při DNSSEC komunikaci a 148 bajtů u klasické DNS komunikace. Je to především kvůli části, která u DNSSEC odpovědi nese data o zprávě typu RRSIG (`Domain Name System (response) > Answers > type RRSIG`). Tato část má velikost 166 bajtů. A dále je rozdíl také v přidané části „Additional records“, která se vyskytuje navíc jak u DNSSEC dotazu, tak DNSSEC odpovědi. V obou paketech má velikost 11 bajtů.

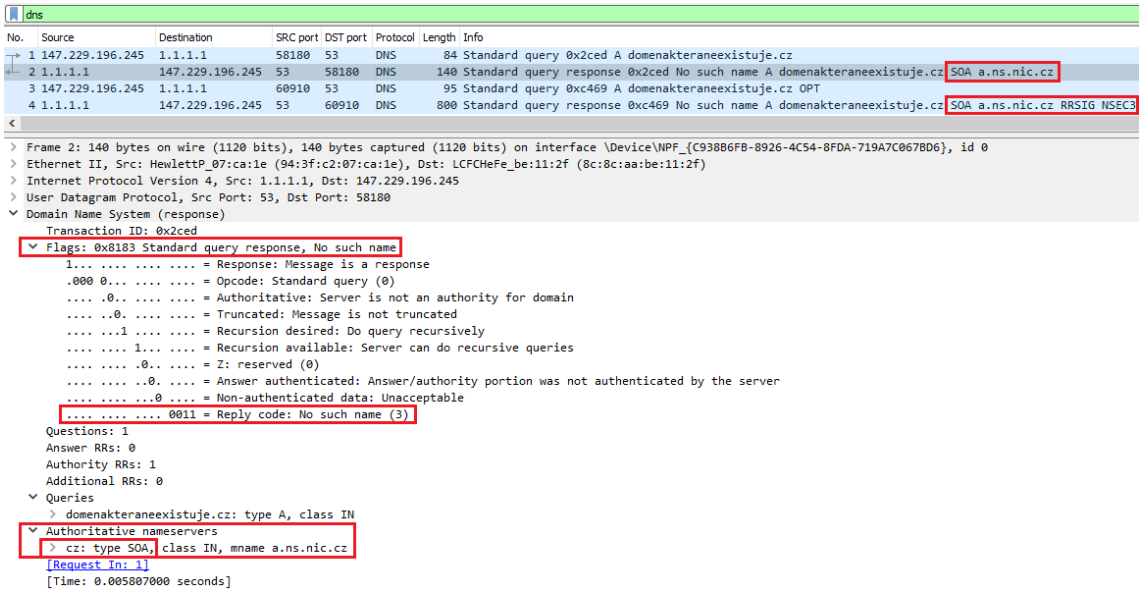
Úkoly:

(3) Který konkrétní prvek nejvíce navyšuje velikost DNSSEC komunikace oproti DNS komunikaci? Jakou velikost v bajtech má tento prvek? Nápověda: Hledejte v části zprávy typu RRSIG.

B.2.2 DNSSEC dotaz na neexistující doménu

Nyní vyzkoušíme dotaz přes aplikaci Klient DNS na neexistující doménu. Jak dojde k ověření autenticity v tomto případě? Nejdříve vyzkoušíme dotaz pomocí klasického DNS a následně s pomocí rozšíření DNSSEC.

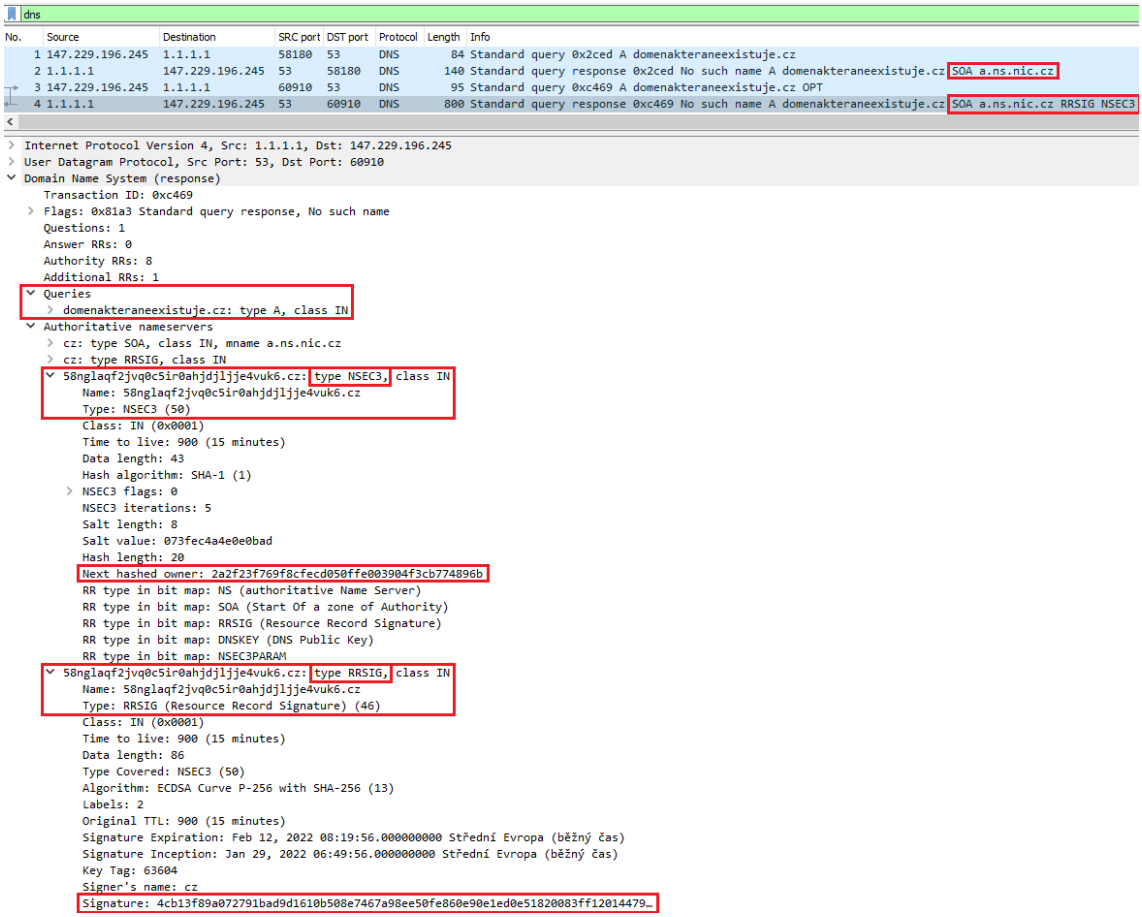
Zadejte do aplikace Klient DNS doménové jméno `domenakteraneexistuje.cz` a DNS server změňte na Cloudflare 1.1.1.1. Ostatní položky ponechejte stejné, jako v předchozí části, tedy deaktivované DNSSEC (CD=☐, AD=☐, DO=☐). Zapněte zachytávání paketů v programu Wireshark, odešlete DNS požadavek a nezastavujte zachytávání síťového provozu. DNS odpověď by měla dorazit, přestože zvolená doména neexistuje, ale samozřejmě nedostaneme žádnou IP adresu. Nyní ponechejte stejná nastavení, ale aktivujte rozšíření DNSSEC (tedy CD=☐, AD=✓, DO=✓) a odešlete dotaz. Stejně jako v předchozím případě se nám vrátí DNS odpověď bez IP adresy, která pro tuto doménu samozřejmě ani neexistuje. Zastavte zachytávání paketů. Ve Wiresharku by měly být vidět čtyři pakety, podobně jako na obr. B.2.7.



Obr. B.2.7: DNS komunikace při dotazu na neexistující doménu pomocí DNS a DNSSEC dotazu. U odpovědi na klasický DNS dotaz je vidět příznak No such name, kterým server oznamuje, že nezná hledanou doménu.

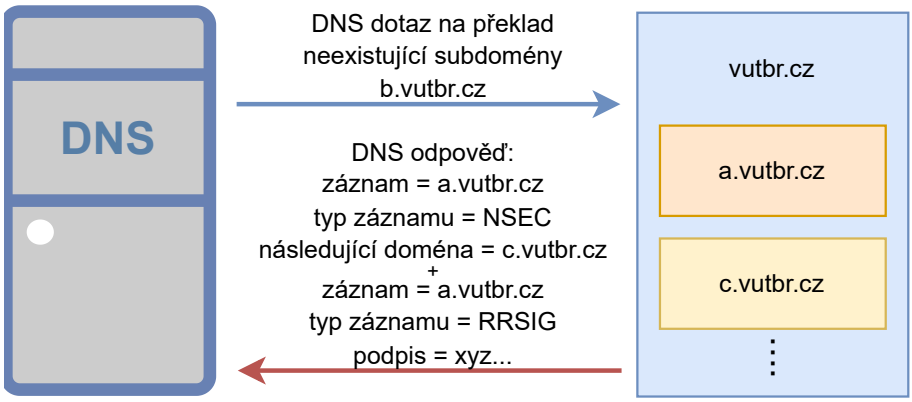
První a druhý paket obsahuje klasickou DNS komunikaci, až na dvě části, které můžeme vidět na obr. B.2.7. První část je v `Domain Name System (response) > Flags > Reply code`, kde místo kódu 0 (No error) vidíme kód 3 (No such name), což znamená, že žádný DNS server nezná hledanou doménu. Druhou odlišnou částí je část „Authoritative nameservers“, která je zde místo části „Answers“, která by se zde objevila v případě, že by nám některý z DNS serverů poskytl přeložené doménové jméno v podobě IP adresy. To se však nestalo, takže zde máme tuto část s typem zprávy SOA (Start of authority). Tato zpráva nám předává informace o nadřazené doméně (cz), jako je doménové jméno nameserveru této domény nebo emailová adresa, která je přiřazena k této doméně.

Třetí a čtvrtý paket obsahují komunikaci s DNSSEC rozšířením. Zde je DNS dotaz naprosto stejný, jako u klasického DNSSEC dotazu a změna nastává u DNS odpovědi, jejíž podrobnosti můžeme vidět na obr. B.2.8. I informaci, že daný DNS server hledanou doménu nezná, je nutné přenést a následně ověřit pravost této informace. K tomu je určen další nový typ záznamu, který se u běžného DNS nevyskytuje a tím je záznam NSEC nebo NSEC3. Každý tento záznam je podepsán standardním způsobem pomocí RRSIG záznamu. V našem případě při hledaní domény `domenakteraneexistuje.cz` je v záznamu typu NSEC3 obsažen hash následující existující domény. Kromě toho také obsahuje informace o použitém hashovacím algoritmu a také o hodnotě TTL, která určuje dobu platnosti daného záznamu. Po přijetí NSEC/NSEC3 záznamu je nutné jej ověřit pomocí podpisu, který je zaslán v RRSIG záznamu. Seznamte se s parametry, které jsou součástí NSEC3 záznamu ve čtvrtém zachyceném paketu.



Obr. B.2.8: DNS komunikace při dotazu na neexistující doménu pomocí DNS a DNSSEC dotazu. U odpovědi s využitým DNSSEC rozšířením můžeme vidět záznamy typu NSEC3 a RRSIG.

Záznamy NSEC/NSEC3 jsou využity v případě, kdy se snažíme dotazovat na neexistující doménu. Je nutné i tuto informaci sdělit v samostatném záznamu, který je podepsán, jako všechny ostatní DNSSEC záznamy, protože jinak by mohl útočník využívat neexistující nebo podobné domény ve svůj prospěch. Například při překlepu a neexistenci NSEC/NSEC3 záznamu, by útočník mohl poskytovat falešný překlad IP adresy pro doménové jméno `seynam.cy` (častá záměna písmen z a y při různém nastavení jazykové sady na klávesnici). Případně může útočník také odpovědět rychleji na DNS dotaz uživatele než DNS server s odpovědí, která uživateli řekne, že hledaná doména neexistuje (i když jde o legitimní doménu, která existuje) a tím dojde k znepřístupnění služby (DoS). V případě útoku Man in the middle útočník ani nemusí odpovídat rychleji. Stačí když odpověď DNS serveru nahradí vlastní odpovědí o neexistenci domény nebo odpovědí s falešnou IP adresou. V případě využití rozšíření DNSSEC respektive NSEC/NSEC3 záznamu máme autentizovanou informaci o tom, že DNS server dotazovanou doménu nezná a nejde o zprávu odeslanou útočníkem.



Obr. B.2.9: Schéma DNSSEC komunikace na neexistující doménu b.vutbr.cz

Nyní se ještě podíváme na rozdíly mezi záznamy NSEC a NSEC3. Oba se používají k autentizovanému informování o tom, že daný server nezná doménové jméno tak, jak bylo popsáno výše. Součástí NSEC záznamu je i informace o následující doméně dle abecedního pořadí. To znamená, že když se dotazujeme na neexistující doménu (pro tuto ukázku například b.vutbr.cz, podobně jako na obr. B.2.9), tak nám od rekurzivního DNS serveru dorazí NSEC záznam z předchozí reálně existující domény (a.vutbr.cz). A v tomto NSEC záznamu bude uvedeno, jaká je následující reálně existující doména (c.vutbr.cz). Tyto záznamy jsou opět podepsány a tak můžeme poskytnutým informacím důvěřovat a tedy pokud nás A informuje o tom, že následující existující doména je C, tak je jasné, že doména B skutečně neexistuje. Ve vymyšlených ukázkových doménách je záměrně uvedena

doména `vutbr.cz`, protože ta poskytuje záznamy typu NSEC. Tento typ záznamu ale umožňuje nežádoucí proces procházení všech subdomén. Tímto procesem, který je také nazýván jako tzv. `zone walking`, lze zjistit např. názvy subdomén, jejich poskytované typy záznamů a další podrobnosti o subdoménách v určité doméně. Řešením zone walkingu je již zmíněná varianta NSEC3 záznam. Na NSEC3 záznam můžeme narazit u většiny domén nejvyššího řádu (TLD), včetně cz domény a také u dalších běžných domén. Tento typ záznamu místo názvu následující domény poskytuje pouze hash následující domény, a tak tuto informaci nemůžeme zneužít k procházení celé domény.

Úkoly:

(4) Vyzkoušejte v aplikaci Klient DNS dotaz na záznam typu A na doménu `113let.vutbr.cz`. Existuje daná doména?

(5) Jaký název nesou předchozí a následující existující subdomény v DNS odpovědi při dotazu na doménu `113let.vutbr.cz`?

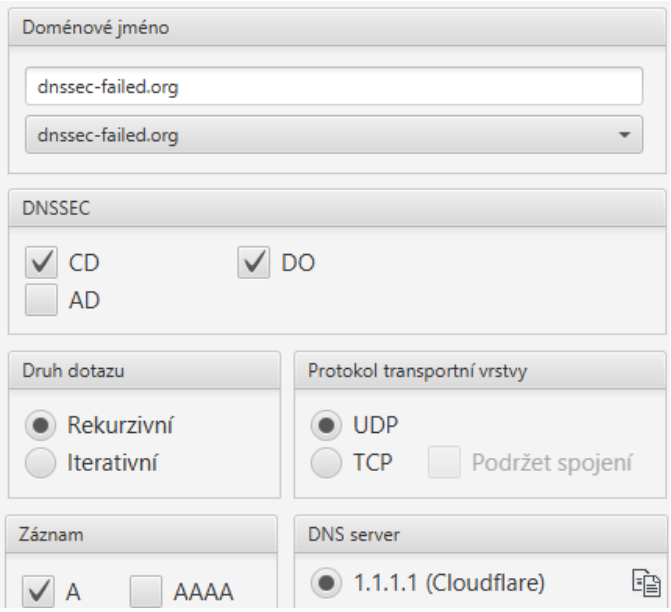
(6) Lze zobrazit název předchozí a následující existující subdomény v DNS odpovědi při dotazu na doménu `domenakteraneexistuje.cz`? Pokud jej nelze zobrazit, tak objasněte proč.

B.2.3 DNSSEC odpověď s podvrženým záznamem

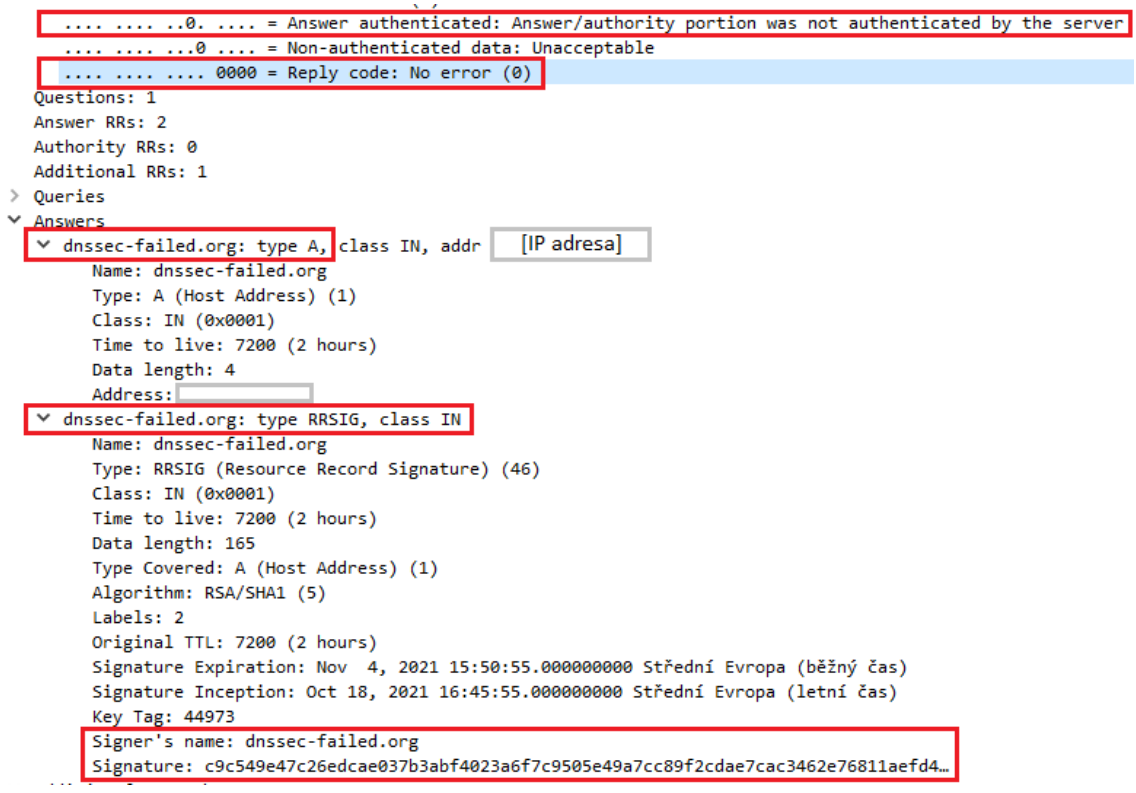
V této části si vyzkoušíme dotázat se na IP adresu doménového jména `dnssec-failed.org`. Změnou oproti předchozím částem je to, že tato doména je záměrně podepsaná tak, aby se při ověřování zjistila chyba autentizace. Tato situace může nastat právě v případě, když se někdo bude snažit narušit DNSSEC komunikaci a upraví záznam, u kterého pak není možné úspěšně ověřit jeho hodnověrnost.

Jako první vyzkoušejte zachytit dotaz a následnou odpověď na doménové jméno `dnssec-failed.org` bez použití DNSSECu. Toho dosáhnete tak, že odškrtnete v aplikaci Klient DNS položku AD bitu a naopak zaškrtnete položku CD bitu, čímž zakážeme kontrolu přijatých dat a vlastně deaktivujeme DNSSEC rozšíření. Možnost DO bitu, kterou si žádáme o podpisové záznamy ponechejte zaškrtnutou. Nastavení bitů a dalších parametrů pro tuto úlohu je zobrazeno i na obr. B.2.10. Proveďte odeslání dotazu a zachyťte tuto komunikaci ve Wiresharku.

Jak je vidět na obr. B.2.11, tak v přijaté DNS odpovědi vidíme IP adresy přeložené z požadovaného doménového jména. Může se tak zdát, že je vše v pořádku, protože klient získal požadované IP adresy a může si tak zobrazit hledanou webovou stránku. O opaku by nás měla přesvědčit následující situace kdy, budeme požadovat ověření autentičnosti těchto získaných dat pomocí DNSSEC.



Obr. B.2.10: Nastavení jednotlivých bitů pro deaktivaci DNSSEC rozšíření.



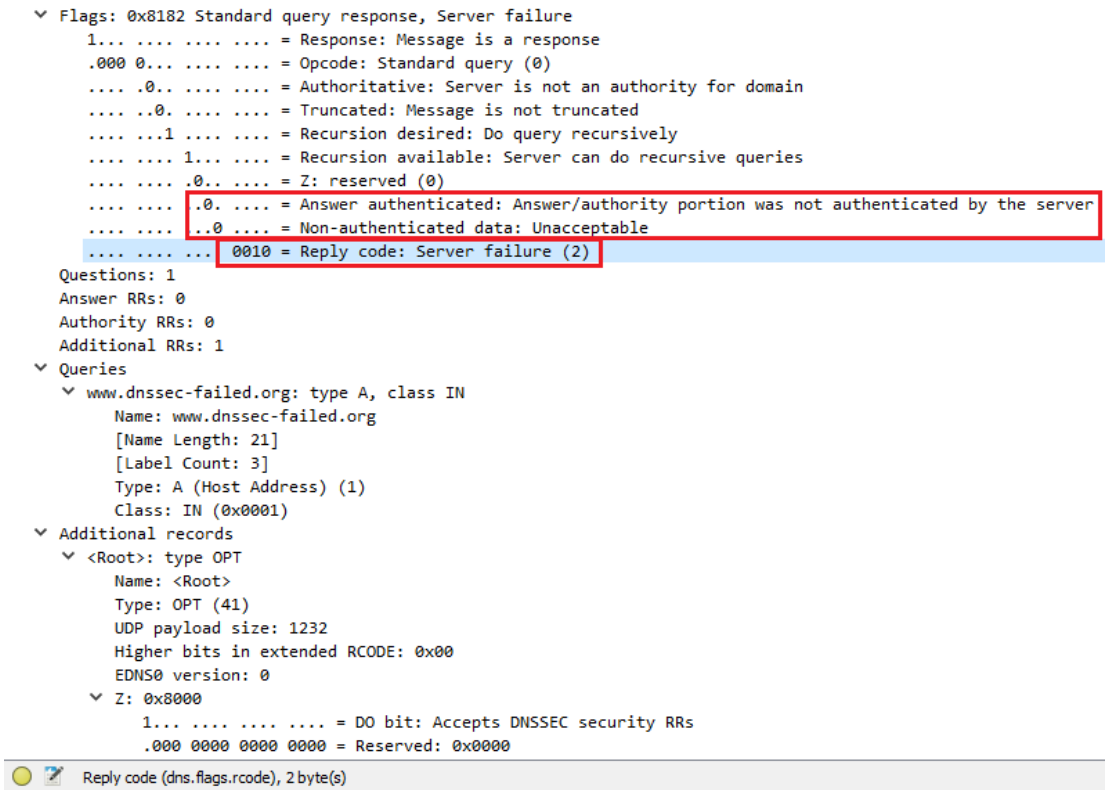
Obr. B.2.11: DNS dotaz a odpověď bez chybové zprávy při zakázaném ověření pomocí DNSSEC.

Úkoly:

(7) Jaké IP adresy byly zjištěny z posledního DNS dotazu na doménové jméno dnssec-failed.org?

Nyní zapněte zachytávání paketů v programu Wireshark a v aplikaci Klient DNS opět aktivujte DNSSEC (CD=☐, AD=✓, DO=✓). Zadejte stejné doménové jméno `dnssec-failed.org` a odešlete DNS dotaz.

Zastavte zachytávání síťové komunikace a prohlédněte si zachycené pakety. Opět je zde standardní DNS dotaz s využitím DNSSEC, ale místo odpovědi s RRSIG záznamem nebo záznamem typu A s přeloženou IP adresou je zde odpověď s chybovou zprávou „Server failure“, podobně jako na obr. B.2.12. Kód chyby je určen posledními čtyřmi bity z celkových šestnácti, které jsou určeny pro část `Domain Name System (response) > Flags`. U těchto příznakových bitů si také můžeme všimnout, že u DNS dotazu je AD bit (Authenticate Data) nastaven na hodnotu 1 (požadujeme provést ověření odpovědi), ale u DNS odpovědi je tento bit nastaven na 0 a autentizace tedy nebyla úspěšně provedena a z toho následně vyplývá chybová zpráva.



Obr. B.2.12: Zachycené DNS pakety včetně DNSSEC odpovědi s chybovou zprávou „Server failure“.

Může se tak zdát, že první varianta se zakázaným DNSSEC rozšířením je lepší, protože jsme získali užitečnou odpověď v podobě IP adres. Je nutné si ale uvědomit, že jsme se zde vzdali všech výhod, které přináší DNSSEC a poskytnuté IP adresy tak nejsou ověřené, respektive mohou být podvržené.

Nyní ještě zachyťte ve Wiresharku situaci, kdy ponecháme zapnuté DNSSEC, ale také v aplikaci Klient DNS zaškrtneme CD bit (= kontrola vypnuta). Všechny bity tedy budou aktivovány (CD=✓, AD=✓, DO=✓). CD bit (Checking Disabled) bude tedy nastaven na hodnotu 1 a data tak budou přijata i přesto, že nebyla autentizována. A to i když požadujeme využití rozšíření DNSSEC pomocí AD bitu (Authenticate Data) nastaveného na hodnotu 1 v DNS dotazu. Jak můžete vidět v zachycené komunikaci, tak jeden jediný CD bit, má vliv na přijatou DNS odpověď, protože stejně jako v případě, kdy jsme DNSSEC vůbec nevyužili, tak i zde byly přijaty přeložené (a nedůvěryhodné) IP adresy.

Úkoly:

(8) Proveďte dotaz na další záměrně špatně podepsanou doménu „rhybar.cz“ nejdříve s pomocí DNSSEC (CD=☐,AD=✓,DO=✓) a následně s možností příjmu i pro neautentizovaná data (CD=✓,AD=✓,DO=✓). Jaká mailová doména je uvedena v podrobnostech SOA záznamu?

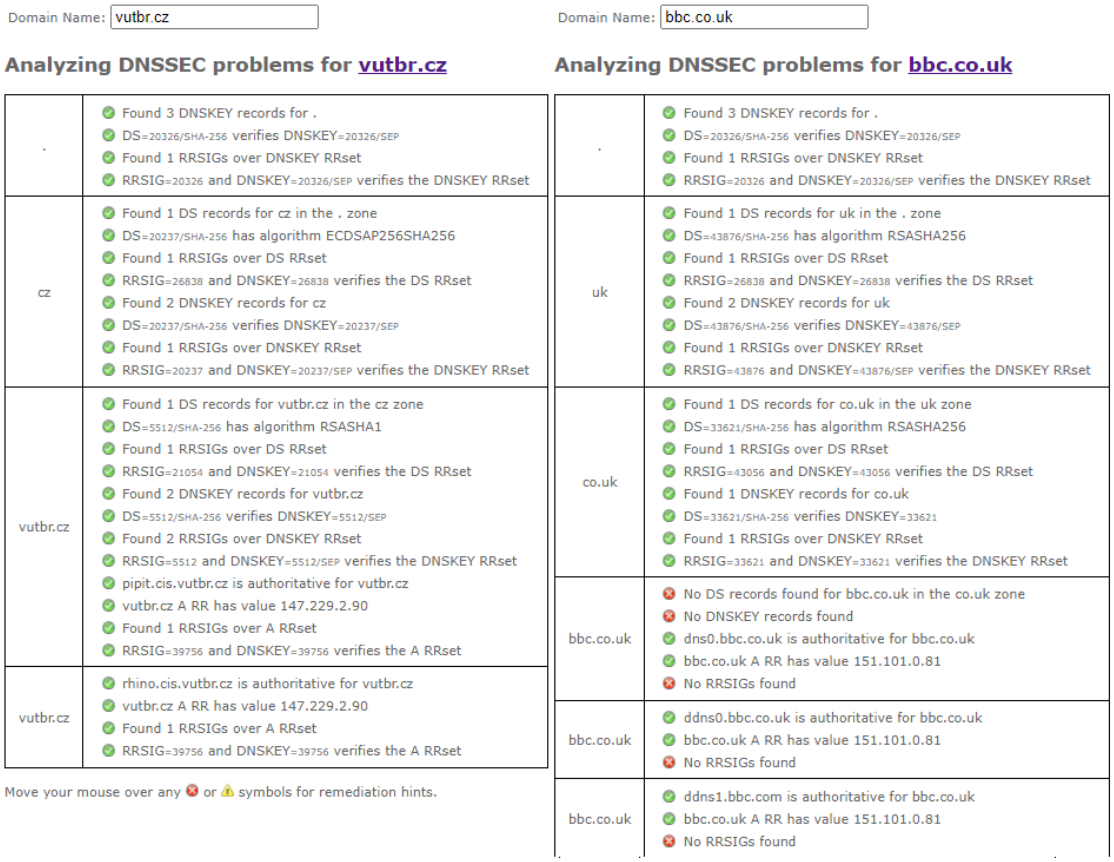
B.2.4 Dotaz na doménu nepodporující DNSSEC

Posledním typem DNSSEC komunikace, kterou budeme zachytávat, bude dotaz na doménu nepodporující DNSSEC. Přestože je DNSSEC značně rozšířen, tak zejména v zahraničí lze narazit na domény, které DNSSEC nepodporují a DNS odpovědi těchto serverů nejsou podepsány, respektive k nim není připojen RRSIG záznam i když o něj požádáme. Na internetu můžeme najít například tuto mapu[1], zobrazující kolik procent serverů podporuje ověření a komunikaci přes DNSSEC. Konkrétní domény lze ověřit například v této webové utilitě DNSSEC Analyzer.[2] Po zadání názvu domény do textového pole se zobrazí zda server na dotaz odpoví s RRSIG záznamem a tedy jestli podporuje DNSSEC. Srovnání domén vutbr.cz a bbc.co.uk lze vidět na obr. B.2.13. V případě, že by doména bbc.co.uk již v době vypracovávání scénáře DNSSEC podporovala, můžete vyzkoušet další domény, jako např. galaxus.ch nebo medium.com. Na straně domény vutbr.cz je z hlediska DNSSEC vše v pořádku, protože všechny úrovně poskytují záznamy typu DNSKEY a RRSIG a je tak možné ověřit jejich autenticitu. Naopak u domény bbc.co.uk jsou tyto typy záznamů poskytovány pouze od kořenového DNS serveru po server domény

[1] https://stats.labs.apnic.net/dnssec

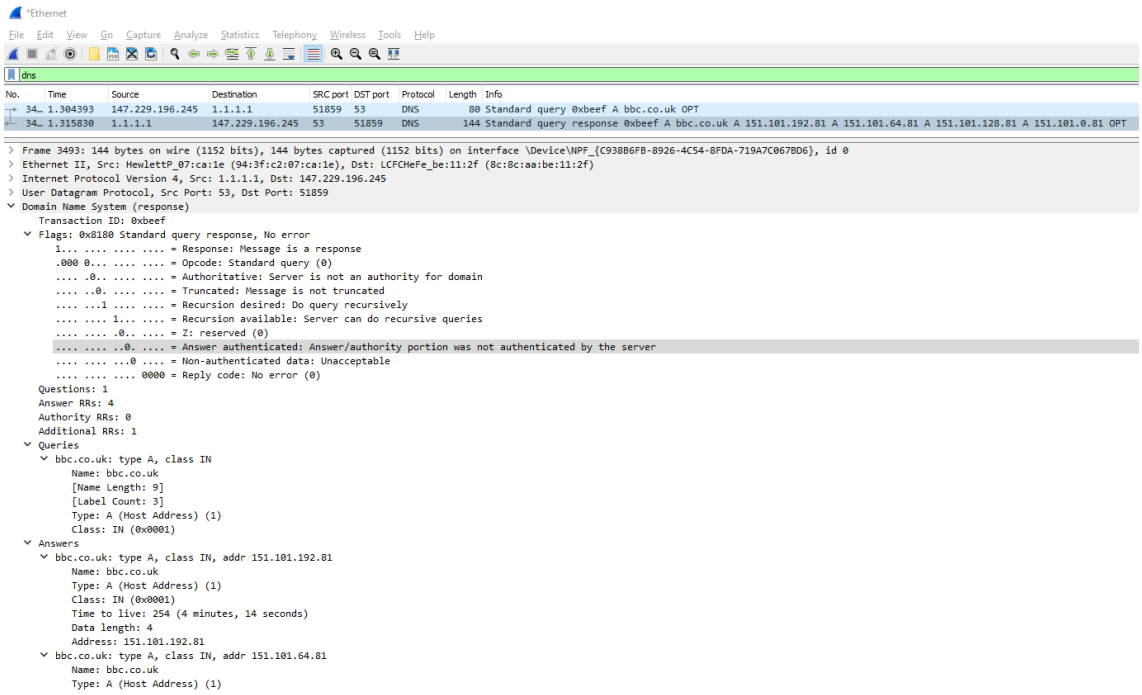
[2] https://dnssec-analyzer.verisignlabs.com/

co.uk. Samotná doména bbc.co.uk odpoví pouze na dotazy typu A, ale neposkytne záznamy DNSKEY ani RRSIG, takže zaslané informace v podobě IP adres nelze ověřit.



Obr. B.2.13: Srovnání domény vutbr.cz a bbc.co.uk z hlediska využití DNSSEC pomocí webového nástroje DNSSEC Analyzer.

Nyní zadejte do aplikace Klient DNS doménové jméno `bbc.co.uk`, u kterého jste ověřili, že nepodporuje rozšíření DNSSEC. Také aktivujte DNSSEC rozšíření tím, že zvolíte možnosti bitů takto CD=☐,AD=✓,DO=✓. Zapněte zachytávání ve Wiresharku a odešlete DNS dotaz. V zachycených paketech, by podobně jako na obr. B.2.14, mělo být vidět, že DNS požadavek byl vyřízen i přesto, že jsme požadovali provedení autentizace a tedy DNSSEC komunikaci. Můžeme tak vidět zjištěné IP adresy v záznamech typu A, ale RRSIG záznamy s podpisem, kterým by se tyto informace ověřily se v DNS odpovědi nenachází. To odpovídá i výsledkům analýzy na obr. B.2.13, kde vidíme, že doména `co.uk` obsahuje RRSIG záznam, ale subdoména `bbc.co.uk` již tento typ záznamu neobsahuje.



Obr. B.2.14: DNS komunikace s doménou, která nemá implementovaný DNSSEC.

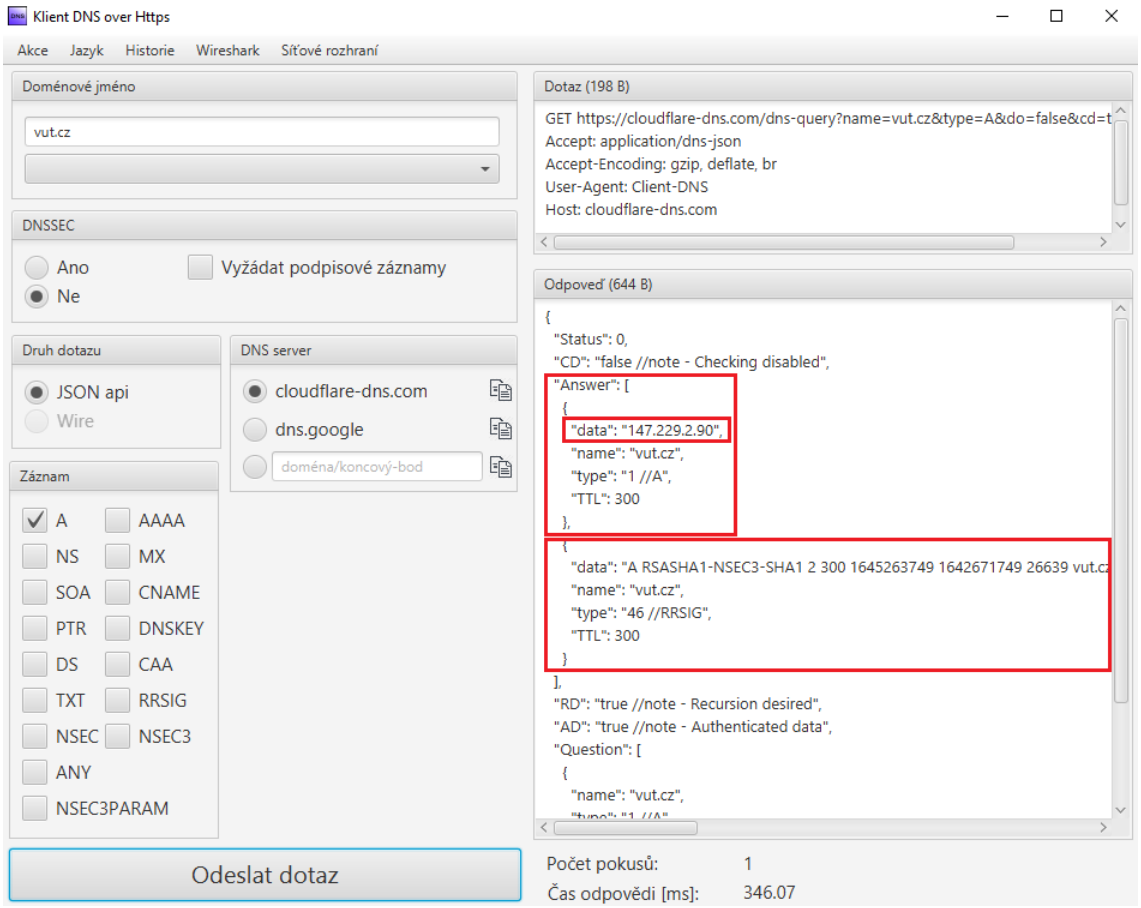
B.2.5 Základní DNS over HTTPS komunikace

Rozšířením DNSSEC jsme tedy zajistili autenticitu při komunikaci rekurzivního serveru s DNS servery. V této části budeme zachytávat provoz, který využívá protokol DNS over HTTPS (DoH). Tento protokol si bere na starosti zabezpečení komunikace z pohledu důvěrnosti, integrity a autenticity mezi klienty a rekurzivními servery, kde právě zmiňované rozšíření DNSSEC žádnou takovou službu nezajišťovalo. Zajištění služeb protokolem DoH je realizováno pomocí zapouzdření DNS paketu do HTTP (Hypertext Transfer Protocol) hlavičky a následně šifrování protokolem TLS (Transport Layer Security).

V nastavení operačního systému Windows zkontrolujte a případně upravte nastavení DNS serverů pro používaný Ethernet adaptér přes `Nastavení > Síť a internet > Ethernet`. Zde vyberte používané připojení a v sekci `Nastavení protokolu IP` ověřte, že IPv4 adresy pro DNS servery jsou `8.8.8.8` a `8.8.4.4`. Tím jsme ověřili, že Google DNS servery jsou nastaveny jako výchozí pro naše odchozí DNS dotazy, což nám pomůže v orientaci při analýze v následujících krocích.

V hlavním menu aplikace DNS Klient zvolte možnost `DoH`. Objeví se obrazovka, ze které můžeme odesílat DoH dotazy, podobně jako na obr. B.2.15. Do pole v levém horním rohu vyplňte dotazované doménové jméno `vut.cz`. Všechny další položky ponechejte ve výchozím stavu. Prozatím tedy nevyužijeme rozšíření DNSSEC,

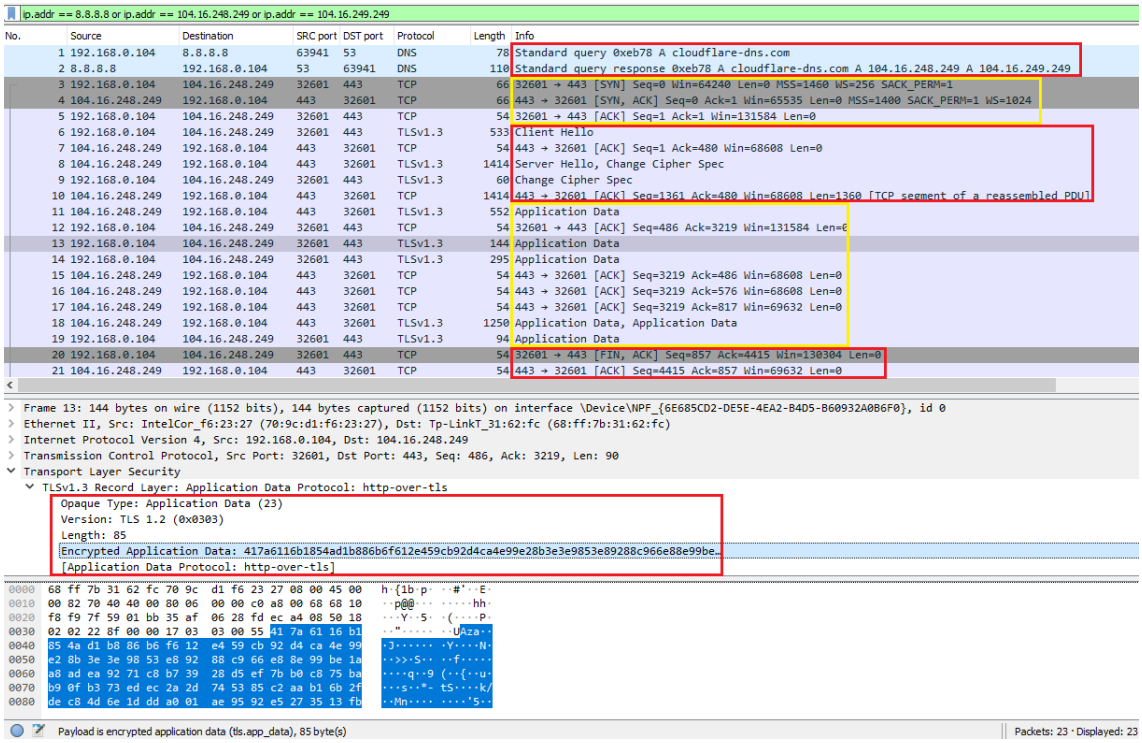
použijeme Cloudflare DNS server a požadujeme odeslat dotaz se záznamem typu A, podobně jak je ukázáno na obr. B.2.15. Zapněte program Wireshark a v něm zachytávání aktuální síťové komunikace. V aplikaci Klient DNS odešlete DoH požadavek tlačítkem „Odeslat dotaz“.



Obr. B.2.15: Nastavení aplikace Klient DNS pro odeslání DNS over HTTPS dotazu v levé části a v pravé části lze vidět doručenou DNS odpověď v podobě IP adresy.

Po přijetí DNS odpovědi zastavte zachytávání síťového provozu v programu Wireshark a zadejte filtr pouze na zobrazení komunikace s IP adresou `8.8.8.8`. Zobrazí se vám pakety, které odpovídají paketům 1 a 2 na obrázku B.2.16. Těmito pakety se ptáme nastaveného Google DNS serveru (nastavený ve Windows) na IP adresu Cloudflare DNS rekurzivního serveru (nastavený v aplikaci Klient DNS), který podporuje protokol DNS over HTTPS. IP adresa zvoleného serveru není v aplikaci pevně definována pro případ, že by byla změněna, a tak se před dotazem na zvolené doménové jméno musíme nejdříve dostat k IP adrese požadovaného rekurzivního serveru Cloudflare. Tento dotaz je komunikován pomocí klasického nezabezpečeného DNS protokolu, a tak můžeme i na obr. B.2.16 v DNS odpovědi

vidět IP adresu, kterou Cloudflare používá pro obsluhu DoH dotazů. V případě, že odešleme více DoH požadavků, tak se již nebude DNS dotaz na IP adresy Cloudflare serveru opakovat, protože se přeložené doménové jméno v podobě IP adresy uloží do cache paměti operačního systému. Přidejte zjištěné IP adresy Cloudflare DNS serveru do filtru v programu Wireshark.



Obr. B.2.16: Zachycené DNS a DoH pakety po dotazu na doménu vut.cz.

V navazujících paketech se nám po zadání vhodného filtru zobrazí klasická TLS komunikace. Začíná se navázáním TCP spojení (pakety číslo 3 až 5) a poté navázání TLS spojení (pakety 6-10), kde se komunikující strany shodnou na použitém šifrovacím algoritmu. Následně kromě potvrzovacích paketů s příznakem ACK vidíme i pakety obsahující šifrovaná data aplikací (Application data – pakety 11-20). Záhlaví tohoto paketu lze vidět na obr. B.2.16. Kromě hodnoty o velikosti zašifrovaných dat zde vidíme pouze použitou verzi protokolu TLS a právě zašifrovaná data. V případě, že bychom dokázali data dešifrovat, tak bychom zde viděli záhlaví HTTP protokolu a v něm zapouzdřená data protokolu DNS. Wireshark nám na posledním řádku v TLS záhlaví ještě zobrazuje informaci, že na aplikační vrstvě je v rámci zašifrovaného TLS paketu použit protokol HTTP (http-over-tls). Tato infromace je uvedena v hranaých závorchách, a tak se jedná o informaci uváděnou pouze Wiresharkem, v samotném paketu tato informace přenášena není. Jak bylo zmíněno výše, tak v HTTP hlavičce je následně zapouzdřen DNS paket.

Tuto informaci ale Wireshark nemá šanci zjistit právě díky použitému šifrování. V aplikaci Klient DNS však můžeme vidět část dešifrovaných dat. Konkrétně jde o část DNS odpovědi, takže podobně jako na obr. B.2.15 můžete vidět dešifrovanou odpověď v podobě záznamu typu A, což je tedy přeložená IP adresa. A dále je zde také záznam typu RRSIG, který stejně jako u rozšíření DNSSEC uchovává digitální podpis. Co v aplikaci Klient DNS z původního paketu zachyceného ve Wiresharku neuvidíme je právě záhlaví HTTP, protože se aplikace zaměřuje čistě jen na výsledky vycházející z protokolu DNS. Poslední pakety, které vidíme ve Wiresharku samozřejmě slouží k ukončení TCP spojení (pakety 20-24).

Úkoly:

(9) Ve Wiresharku porovnejte a následně popište potřebný počet bajtů pro:
1) klasický DNS dotaz a odpověď na IP adresu Cloudflare serveru
2) DoH dotaz a odpověď na překlad doménového jména vut.cz včetně navazování a ukončování spojení.
Nápověda: Využít můžete například nástroj pro analýzu `Statistics > Conversations`. Pro jednodušší orientaci zaškrtněte položku v dolní části okna „Limit to display filter“.

(10) Jaká hlavní výhoda a nevýhoda tedy vyplývá z předchozí analýzy?

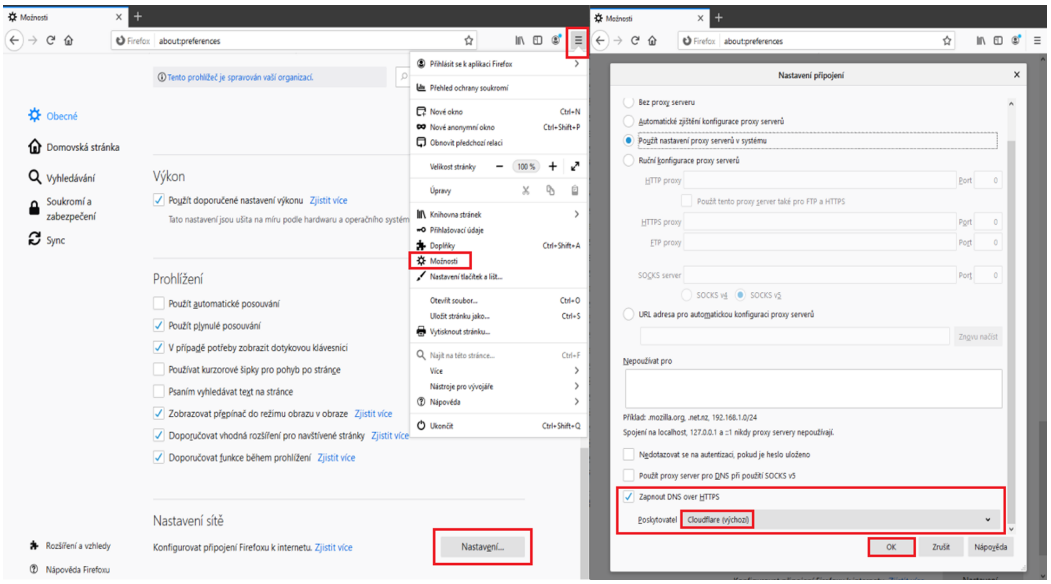
(11) Identifikujte slabé místo proběhlé DoH komunikace.

B.2.6 Implementace DoH ve webovém prohlížeči

Nyní si ověříme, jak je DNS over HTTPS implementováno v praxi v rámci webového prohlížeče. Nejznámější prohlížeče již nějakou dobu protokol DoH podporují. My si nyní vyzkoušíme vygenerovat webový provoz skrze webový prohlížeč Firefox s využitím protokolu DoH.

Přejdeme k aktivaci protokolu DoH v nastavení webového prohlížeče. Takže obdobně jako v podkapitole B.2.1 najděte potřebné nastavení přes `Menu (3 čárky v pravém horním rohu) > Možnosti > Nastavení sítě > Nastavení...` Zde aktivujte položku „zapnout DNS over HTTPS“, stejně jako je ukázáno na obr. B.2.17 a v rozbalovacím menu u položky „Poskytovatel“ zvolte `Cloudflare`. Provedená nastavení uložte kliknutím na tlačítko „OK“ a restartujte prohlížeč.

Ve Wiresharku si přichystejte filtr pro Google DNS IP adresy (8.8.8.8 a 8.8.4.4) a zapněte zachytávání síťového provozu. Ve webovém prohlížeči načtěte webovou stránku `nic.cz` a poté zastavte zachytávání paketů. Při použití filtru by mělo být vidět, jak se dotazujeme Google DNS adresy na IP adresu Cloudflare serveru, podobně jako na obr. B.2.18. Dále přidejte tyto zjištěné Cloudflare adresy do filtru ve Wiresharku. Měl by být vidět DoH provoz v podobě šifrovaných TLS



Obr. B.2.17: Nastavení webového prohlížeče pro zapnutí podpory DoH a nastavení požadovaného serveru.

ip.addr == 8.8.8.8 or ip.addr == 8.8.8.8 or ip.addr == 8.8.4.4 or ip.addr == 104.16.248.249 or ip.addr == 104.16.249.249

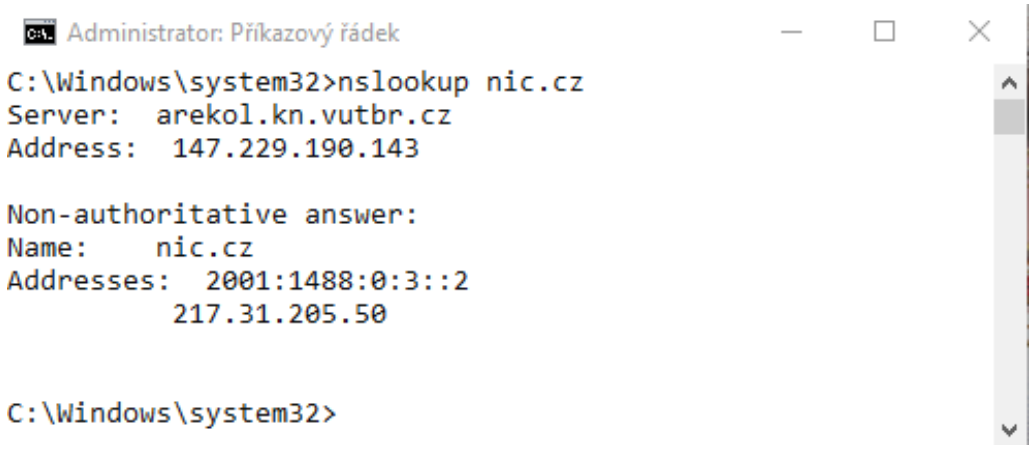
No.	Source	Destination	SRC port	DST port	Protocol	Length	Info
1	192.168.137.203	8.8.8.8	58434	53	DNS	86	Standard query 0x0c0e A mozilla.cloudflare-dns.com
2	8.8.8.8	192.168.137.203	53	58434	DNS	118	Standard query response 0x0c0e A mozilla.cloudflare-dns.com A 104.16.249.249 A 104.16.248.249
3	192.168.137.203	8.8.8.8	51510	53	DNS	86	Standard query 0xd996 AAAA mozilla.cloudflare-dns.com
4	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	110	Application Data
5	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	135	Application Data
6	8.8.8.8	192.168.137.203	53	51510	DNS	142	Standard query response 0xd996 AAAA mozilla.cloudflare-dns.com AAAA 2606:4700::6810:f9f9 AAAA 2606:4700::6810:f8f9
7	104.16.248.249	192.168.137.203	443	50259	TLSv1.2	89	Application Data
8	104.16.248.249	192.168.137.203	443	50259	TLSv1.2	211	Application Data
9	192.168.137.203	104.16.248.249	50259	443	TCP	54	50259 → 443 [ACK] Seq=138 Ack=193 Win=1025 Len=0
10	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	110	Application Data
11	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	132	Application Data
12	104.16.248.249	192.168.137.203	443	50259	TCP	60	443 → 50259 [ACK] Seq=193 Ack=272 Win=69 Len=0
13	104.16.248.249	192.168.137.203	443	50259	TLSv1.2	89	Application Data
14	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	114	Application Data
15	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	138	Application Data
16	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	114	Application Data
17	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	140	Application Data
18	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	110	Application Data
19	192.168.137.203	104.16.248.249	50259	443	TLSv1.2	134	Application Data
20	104.16.248.249	192.168.137.203	443	50259	TLSv1.2	196	Application Data
22	104.16.248.249	192.168.137.203	443	50259	TLSv1.2	115	Application Data
23	192.168.137.203	104.16.248.249	50259	443	TCP	54	50259 → 443 [ACK] Seq=698 Ack=431 Win=1024 Len=0

Obr. B.2.18: Zachycená DNS a DoH komunikace při použití webového prohlížeče.

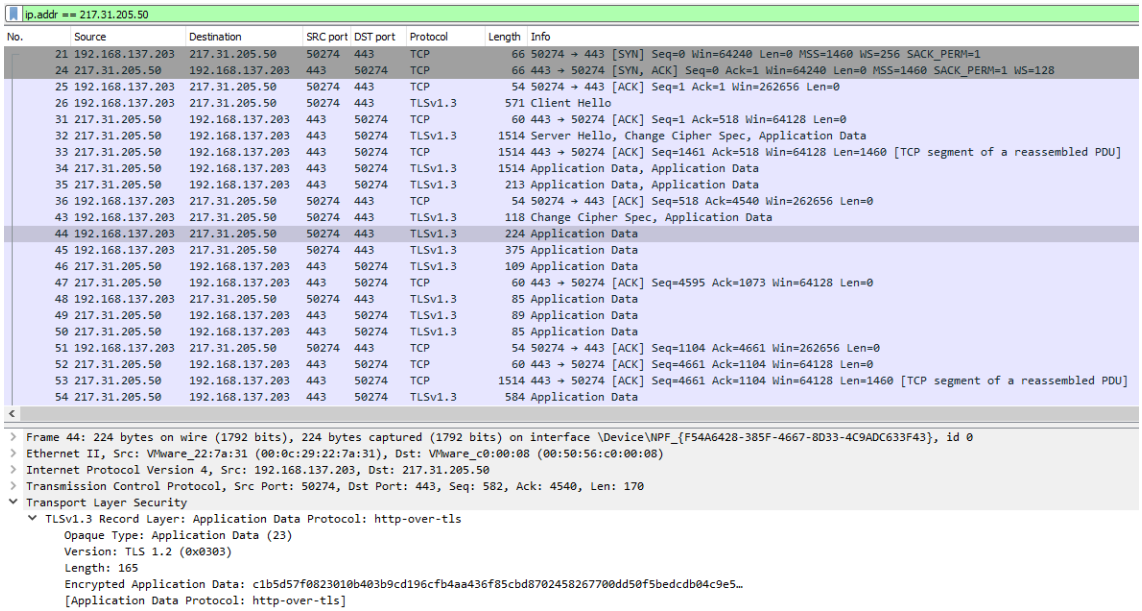
paketů. V těchto paketech se šifrovaně dotazujeme Cloudflare serveru na IP adresy doménového jména nic.cz. To je hlavní rozdíl oproti klasické nešifrované verzi DNS protokolu, jejíž zachycení proběhlo při komunikaci s prohlížečem v podkapitole B.2.1.

Otevřete příkazovou řádku a přes příkaz `nslookup nic.cz` zjistěte IP adresu domény `nic.cz`, kterou jsme načítali ve webovém prohlížeči. Okno příkazové řádky s příkazem je zobrazeno na obr. B.2.19. Následně tuto zjištěnou IPv4 adresu také vyfiltrujte ve Wiresharku. Měl by se vyfiltrovat další šifrovaný provoz, který obsahuje HTTP komunikaci, opět šifrovanou TLS protokolem, podobně jako na obr. B.2.20. Tento provoz je již obsáhlejší než samotná DNS komunikace, protože se načítá celá

webová stránka. Důležité ale je si uvědomit, že i při použití DNS over HTTPS je možné při sledování komunikujících IP adres zjistit, na které webové stránky přistupujeme nebo na které IP adresy jsme se dotazovali.



Obr. B.2.19: Příkazová řádka s příkazem nslookup pro zjištění IP adresy domény nic.cz.



Obr. B.2.20: Zachycená HTTPS komunikace po vyfiltrování IP adresy domény nic.cz.

C Kompletní návod pro třetí vytvořený simulační scénář

ÚLOHA č. 3

Simulace komunikace rekurzivního serveru při překladu doménových jmen pomocí DNS a DNSSEC.

C.1 Teoretický úvod

Běžný protokol DNS zajišťuje pouze překlad doménového jména na IP adresu a případně naopak. Nic však nebrání podvržení poskytovaných informací a proto se v tomto cvičení budeme věnovat i zabezpečené variantě DNS protokolu v podobě rozšíření **DNSSEC** (Domain Name System Security Extensions). Hlavním úkolem tohoto scénáře bude simulace komunikace rekurzivního serveru, který pro klienta zajišťuje překlad doménových jmen, a to jak v zabezpečené (DNSSEC), tak nezabezpečené (DNS) formě. K tomu bude využit virtuální stroj VMware, který bude simulovat rekurzivní server pomocí aplikace DNS Klient s DNS servery. Úloha obsahuje návod a popis komunikace rekurzivního serveru při využití klasického protokolu DNS a následně je komunikace opakována při využití rozšíření DNSSEC. U této komunikace je potřeba využít i další typy záznamů, které slouží k zajištění autenticity získaných dat. Studenti tak zjistí, jaké množství komunikace se skrývá za jedním dotazem od klienta, který na rekurzivní server pošle jeden paket s DNS dotazem a během chvíle mu dorazí paket s DNS odpovědí.

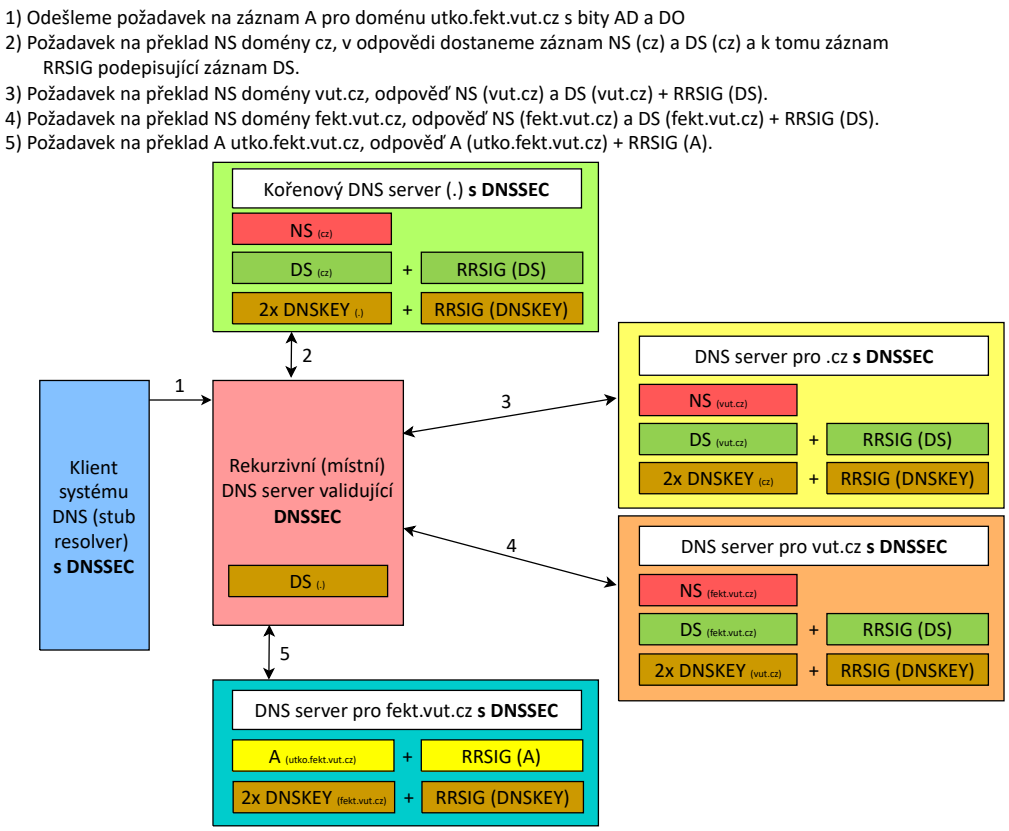
C.1.1 Princip zabezpečení překladu doménových jmen pomocí DNSSEC

DNSSEC je rozšířením pro nám již dobře známý protokol DNS. Rozšíření DNSSEC přidává k informacím v běžném DNS paketu informace o digitálním podpisu. Ověřením tohoto podpisu se zajišťuje autentičnost poskytnutých informací, kterým tak můžeme důvěřovat. Je ale potřeba zdůraznit, že zabezpečení autentičnosti dat je pomocí DNSSEC zajišťováno pouze mezi rekurzivním resolverem a dotazovaným DNS serverem, nikoliv při komunikaci klienta (stub resolveru) s rekurzivním serverem. Ke klasickým DNS záznamům jako jsou záznamy A, AAAA, NS nebo PTR se tak přidávají další jako např. RRSIG (Resource Record Signature), DNSKEY (DNS Public Key) nebo NSEC (Next Secure), popř. NSEC3 a taktéž v samotném DNS záhlaví se mění hodnoty některých příznakových bitů.

C.2 Realizace scénáře

C.2.1 Ukázka komunikace rekurzivního serveru

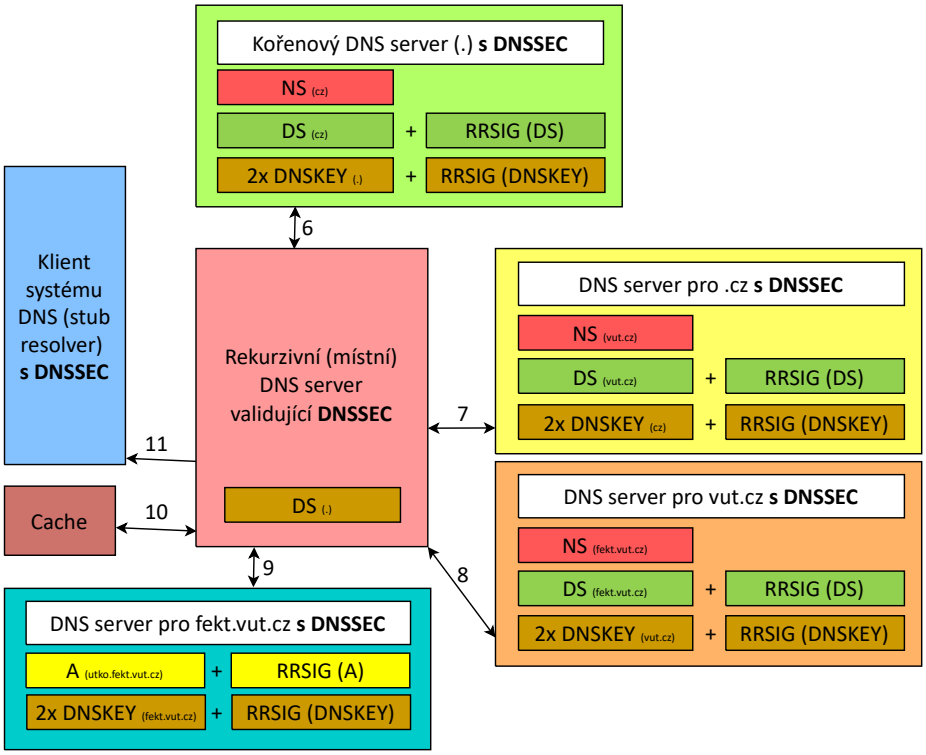
V této části bude vysvětlena a popsána práce rekurzivního resolveru, jehož fungování při dotazu na doménu `utko.fekt.vut.cz` je zobrazeno na schématech na obrázcích C.2.1 a C.2.2 a také na příkladech s konkrétními IP adresami na obrázcích C.2.3 a C.2.4. Poté si sami vyzkoušíte simulovat dotazy rekurzivního serveru na doménu `obcan.portal.gov.cz`.



Obr. C.2.1: První část fungování rekurzivního resolveru při dotazu na doménu utko.fekt.vut.cz, která ukazuje zejména záznamy typu NS.

Rekurzivní server je server, kterému klient odesílá DNS dotaz a následně je klientovi serverem poskytnut finální překlad domény na IP adresu. Mezi tímto dotazem a odpovědí je však bez povšimnutí klienta provedeno několik dalších DNS dotazů, které iniciuje právě rekurzivní DNS server. Dotazuje se postupně na všechny domény, kdy první je tzv. doména nejvyššího řádu (TLD - Top Level Domain – doména nejvyššího řádu). V případě, že klient požaduje nalezení IP adresy, která odpovídá doménovému jménu `utko.fekt.vut.cz`, tak rekurzivní DNS

6) Odešleme požadavek na záznam DNSKEY směrem ke kořenovému DNS serveru (.), v odpovědi dostaneme 2x záznam typu DNSKEY (.) a RRSIG záznam podepisující záznam DNSKEY
7) Požadavek na DNSKEY (cz), odpověď 2x DNSKEY (cz) + RRSIG (DNSKEY)
8) Požadavek na DNSKEY (vut.cz), odpověď 2x DNSKEY (vut.cz) + RRSIG (DNSKEY)
9) Požadavek na DNSKEY (fekt.vut.cz), odpověď 2x DNSKEY (fekt.vut.cz) + RRSIG (DNSKEY)
10) Ověření záznamu typu A (utko.fekt.vut.cz) a uložení všech získaných záznamů do cache
11) Odpověď typu A pro hledanou doménu utko.fekt.vut.cz + RRSIG (A)

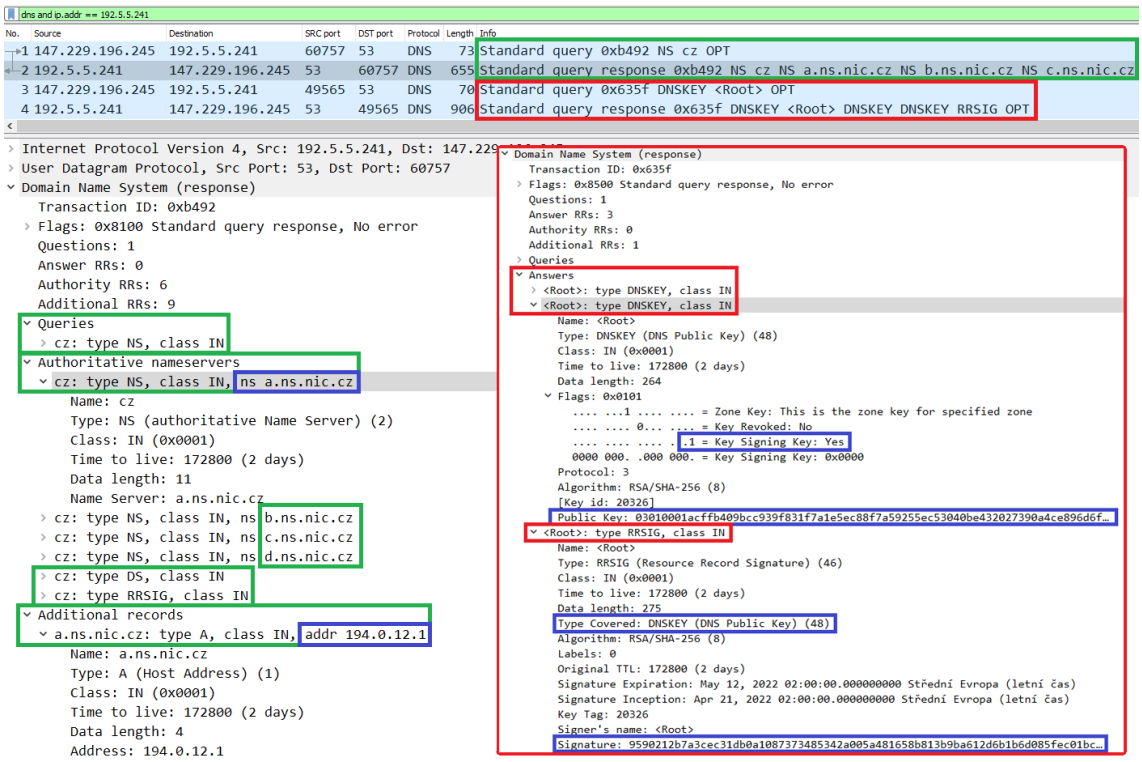


Obr. C.2.2: Druhá část fungování rekurzivního resolveru při dotazu na doménu utko.fekt.vut.cz, která ukazuje zejména získávání DNS záznamů typu DNSKEY.

server začne s dotazem na doménu `.cz` a svůj dotaz odešle na jeden z kořenových DNS serverů. Adresy těchto serverů jsou veřejně dostupné například na webových stránkách organizace IANA.[1] V našem případě vybereme například kořenový server F s IP adresou 192.5.5.241 a odešleme svůj požadavek, ve kterém specifikujeme, že požadujeme adresu serveru, který nám poskytne adresy v doméně cz. Kořenový server nám poskytne (většinou hned několik) IP adres a rekurzivní DNS server na jednu z nich poté směřuje další požadavky na další doménu v pořadí.

DNS servery pro zjištění adres dalších DNS serverů nevyužívají záznamy typu A, které již známe, ale využívají NS záznamy (Name Server). V případě použití DNSSEC je odpověď mezi servery doručena v paketu obsahující kromě NS záznamů také záznamy typu DS (Delegation Signer) a RRSIG (Resource Record Signature), které zajišťují právě služby rozšíření DNSSEC. NS záznamy poskytnou rekurzivnímu

[1] https://www.iana.org/domains/root/servers

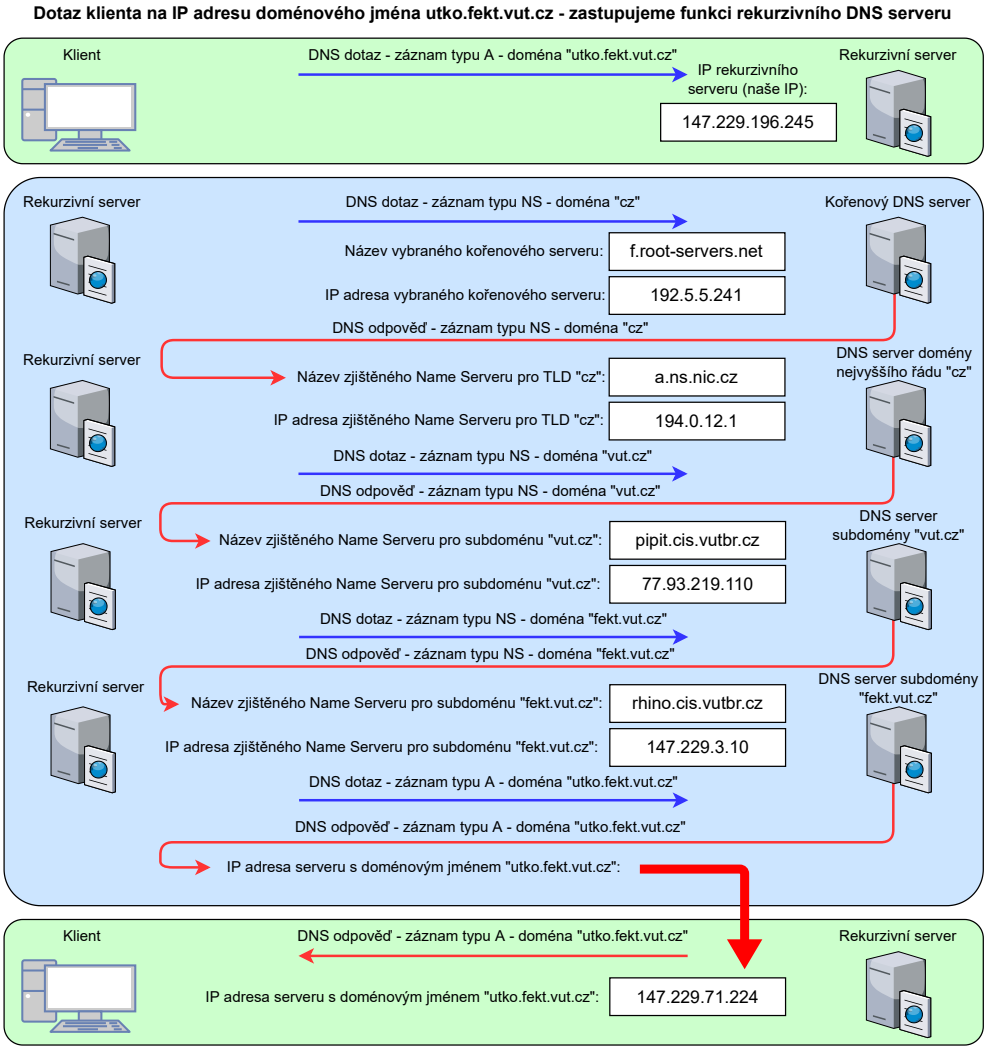


Obr. C.2.3: Simulované dotazy rekurzivního DNS serveru na kořenový DNS server ohledně NS záznamů pro doménu nejvyššího řádu cz. (levá část) a DNSKEY a RRSIG záznamů kořenového serveru (pravá část).

serveru názvy serverů, které mu následně mohou zjistit adresy v další doméně. NS záznam ale neposkytuje IP adresu těchto serverů a proto jsou většinou součástí paketů také klasické záznamy typu A, které tyto IP adresy poskytnou a rekurzivní DNS server tak nemusí generovat další DNS požadavky. Záznamy typu A jsou ve Wiresharku zobrazovány v části „Adidtional records“, jak je vidět např. na obrázku C.2.3 (levá část). Záznam DS poskytuje hodnotu hashe DNSKEY záznamu pro následující doménu `.cz`. RRSIG záznam poté obsahuje digitální podpis tohoto DS záznamu.

Dále rekurzivní server komunikuje s následujícím serverem dle DNS hierarchie, jehož název (a případně i IP adresu) zjistil pomocí předcházejících NS záznamů. V našem ukázkovém případě tedy bude následovat dotaz na název serveru pro doménu `vut.cz` a komunikovat budeme s DNS serverem domény cz s IP adresou 192.5.5.241, viz obr. C.2.3 a C.2.4. Situace je zde poté obdobná jako v předchozím případě a pokud zjistíme IP adresu DNS serveru pro doménu vut.cz, tak pokračujeme dále na subdoménu `fekt.vut.cz`. Pokud zjistíme adresu DNS serveru pro tuto subdoménu, tak již můžeme využít DNS dotaz se záznamem typu A a dotážeme

se přímo na IP adresu serveru s hledanou doménou `utko.fekt.vut.cz`. V odpovědi bude kromě A záznamu také RRSIG podpis tohoto A záznamu.

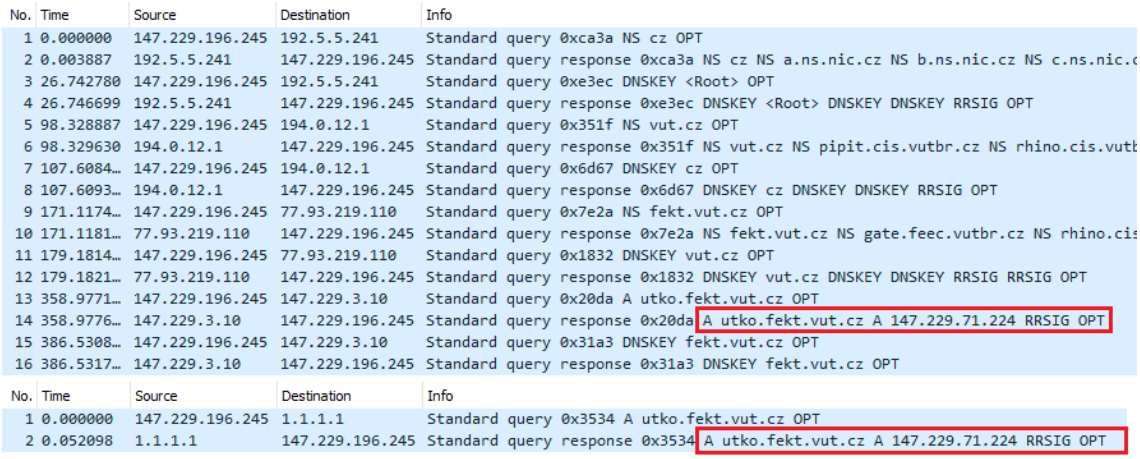


Obr. C.2.4: Schéma simulované komunikace rekurzivního serveru s dotazem na doménu utko.fekt.vut.cz s dotazy na jednotlivé DNS servery, včetně zapsaných názvů a IP adres těchto serverů.

Dále je nutné ověřit podpisy obdržených RRSIG záznamů. Toto ověření se provádí pomocí veřejných klíčů, které rekurzivní server obdrží při odeslání DNS dotazu se záznamem typu DNSKEY. V DNS odpovědi jsou uvedeny dva záznamy typu DNSKEY a k tomu odpovídající RRSIG záznam, podobně jako na obr. C.2.3 (pravá část). Jeden z DNSKEY záznamů obsahuje tzv. zone signing key (ZSK) a druhý obsahuje key signing key (KSK). Nejdříve se dotaz na záznam DNSKEY odesílá na kořenový DNS server a poté na servery nižší úrovně. Jako první se porovnává hash RRSIG záznamu k DNSKEY záznamu s hodnotou hashe z DS

záznamu pro kořenovou (root) zónu. Následně se ještě ověřuje RRSIG podpis u DS záznamu pro navazující doménu cz. V záznamu RRSIG můžeme narazit na položku „Type covered“, která pomocí číselné hodnoty definuje, ke kterému typu záznamu se daný podpis vztahuje (A = 1, DNSKEY = 48...). Poté se proces přesouvá právě na komunikaci s DNS serverem domény cz a jsou odesílány DNS dotazy na záznamy DNSKEY podobně jako v předchozím případě. Na konci řetězce dojde k ověření RRSIG podpisu záznamu typu A a v případě, že všechna ověření proběhla úspěšně, tak se tento A záznam přepošle od rekurzivního serveru ke klientovi.

Na obrázku C.2.5 lze vidět srovnání množství potřebných paketů při DNS dotazu s využitím DNSSEC na doménové jméno utko.fekt.vut.cz. Horní část obrázku ukazuje postup, který obstarává rekurzivní server. Ten začíná komunikaci s kořenovým DNS serverem a následně postupuje dále, až se dostane k požadované adrese. Pro zjednodušení se DNS resolver vždy ptá rovnou i na DNSKEY záznamy. Dolní část obrázku ukazuje dva pakety, které jako jediné projdou mezi klientem a rekurzivním DNS serverem a jejich výsledkem je opět požadovaná adresa. Klient tedy odešle DNS požadavek, ten je zpracován rekurzivním DNS serverem tak, jak je ukázáno v horní části a nakonec rekurzivní server odešle zjištěnou adresu klientovi, jak ukazuje poslední paket na tomto obrázku.



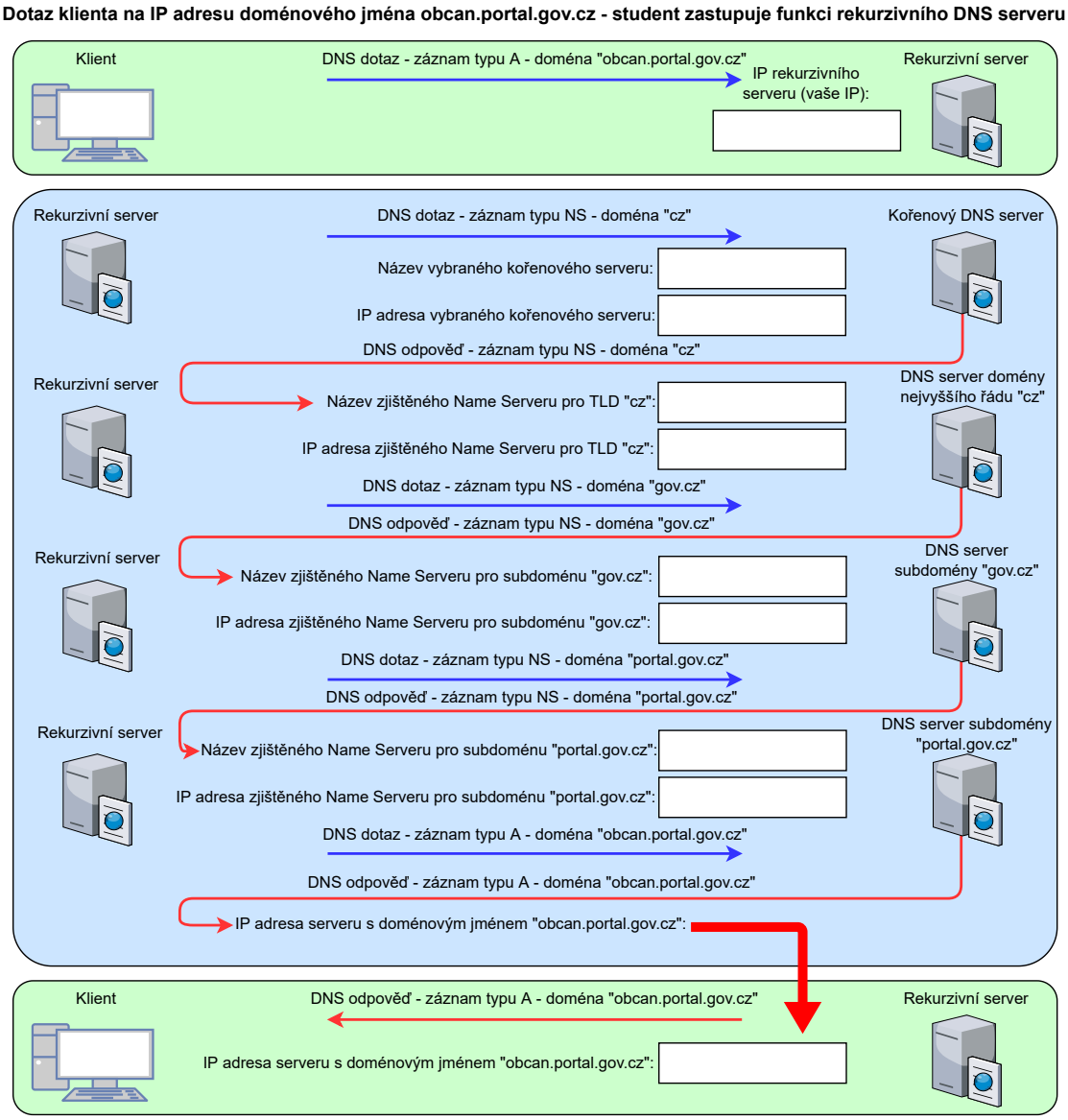
No.	Time	Source	Destination	Info
1	0.000000	147.229.196.245	192.5.5.241	Standard query 0xca3a NS cz OPT
2	0.003887	192.5.5.241	147.229.196.245	Standard query response 0xca3a NS cz NS a.ns.nic.cz NS b.ns.nic.cz NS c.ns.nic.c
3	26.742780	147.229.196.245	192.5.5.241	Standard query 0xe3ec DNSKEY <Root> OPT
4	26.746699	192.5.5.241	147.229.196.245	Standard query response 0xe3ec DNSKEY <Root> DNSKEY DNSKEY RRSIG OPT
5	98.328887	147.229.196.245	194.0.12.1	Standard query 0x351f NS vut.cz OPT
6	98.329630	194.0.12.1	147.229.196.245	Standard query response 0x351f NS vut.cz NS pipit.cis.vutbr.cz NS rhino.cis.vutb
7	107.6084…	147.229.196.245	194.0.12.1	Standard query 0x6d67 DNSKEY cz OPT
8	107.6093…	194.0.12.1	147.229.196.245	Standard query response 0x6d67 DNSKEY cz DNSKEY DNSKEY RRSIG OPT
9	171.1174…	147.229.196.245	77.93.219.110	Standard query 0x7e2a NS fekt.vut.cz OPT
10	171.1181…	77.93.219.110	147.229.196.245	Standard query response 0x7e2a NS fekt.vut.cz NS gate.feec.vutbr.cz NS rhino.cis
11	179.1814…	147.229.196.245	77.93.219.110	Standard query 0x1832 DNSKEY vut.cz OPT
12	179.1821…	77.93.219.110	147.229.196.245	Standard query response 0x1832 DNSKEY vut.cz DNSKEY DNSKEY RRSIG RRSIG OPT
13	358.9771…	147.229.196.245	147.229.3.10	Standard query 0x20da A utko.fekt.vut.cz OPT
14	358.9776…	147.229.3.10	147.229.196.245	Standard query response 0x20da A utko.fekt.vut.cz A 147.229.71.224 RRSIG OPT
15	386.5308…	147.229.196.245	147.229.3.10	Standard query 0x31a3 DNSKEY fekt.vut.cz OPT
16	386.5317…	147.229.3.10	147.229.196.245	Standard query response 0x31a3 DNSKEY fekt.vut.cz OPT

No.	Time	Source	Destination	Info
1	0.000000	147.229.196.245	1.1.1.1	Standard query 0x3534 A utko.fekt.vut.cz OPT
2	0.052098	1.1.1.1	147.229.196.245	Standard query response 0x3534 A utko.fekt.vut.cz A 147.229.71.224 RRSIG OPT

Obr. C.2.5: Porovnání množství potřebných paketů u DNS komunikace z pohledu rekurzivního serveru a koncové stanice, při použití DNSSECu pro zjištění IP adresy serveru s doménovým jménem utko.fekt.vut.cz a jejího podpisu.

C.2.2 Simulace DNS dotazů rekurzivního serveru

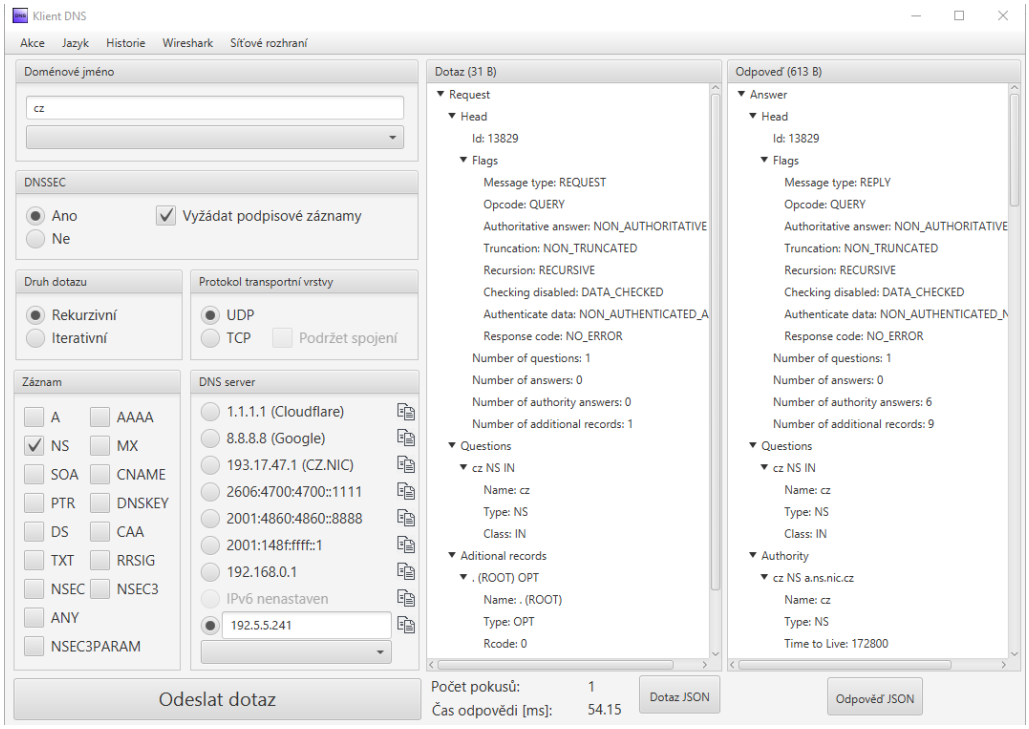
Nyní si vyzkoušíte simulovat rekurzivní DNS server při dotazu klienta na IP adresu doménového jména `obcan.portal.gov.cz`. Svůj postup včetně názvů a IP adres dotazovaných serverů zaznamenávejte průběžně do obrázku č. C.2.6, který je součástí úkolu č. (5).



Obr. C.2.6: Šablona pro úkol č. (5), která znázorňuje proces zjišťování IP adresy z doménového jména jak z pohledu klienta, tak z pohledu rekurzivního DNS serveru.

Jak bylo zmíněno výše, rekurzivní server začíná dotazem na Name Server domény nejvyššího řádu, tedy `.cz`. Jako kořenový server si vybereme kořenový server F (Internet Systems Consortium, Inc.). Jeho IP adresu si vyhledejte na tomto

odkazu.[2] Do aplikace Klient DNS vyplňte tuto IP adresu do pole `DNS server` v dolní části okna. Dále v horní části vyplňte doménu „cz“, pro kterou chceme vyhledat názvy serverů. Také povolte využití DNSSEC, podobně jako na obr. C.2.7 a zaškrtněte požadovaný typ záznamu „NS“. Zapněte zachytávání paketů ve Wiresharku a odešlete DNS dotaz na kořenový DNS server. Poté zůstane adresa DNS serveru ještě chvíli stejná a pro zjednodušení procesu odešleme ještě i požadavek na záznam typu DNSKEY pro root zónu. Do pole `Doménové jméno` vepište pouze tečku „.“, která označuje právě root doménu. Dále změňte typ záznamu z NS na DNSKEY a odešlete požadavek.

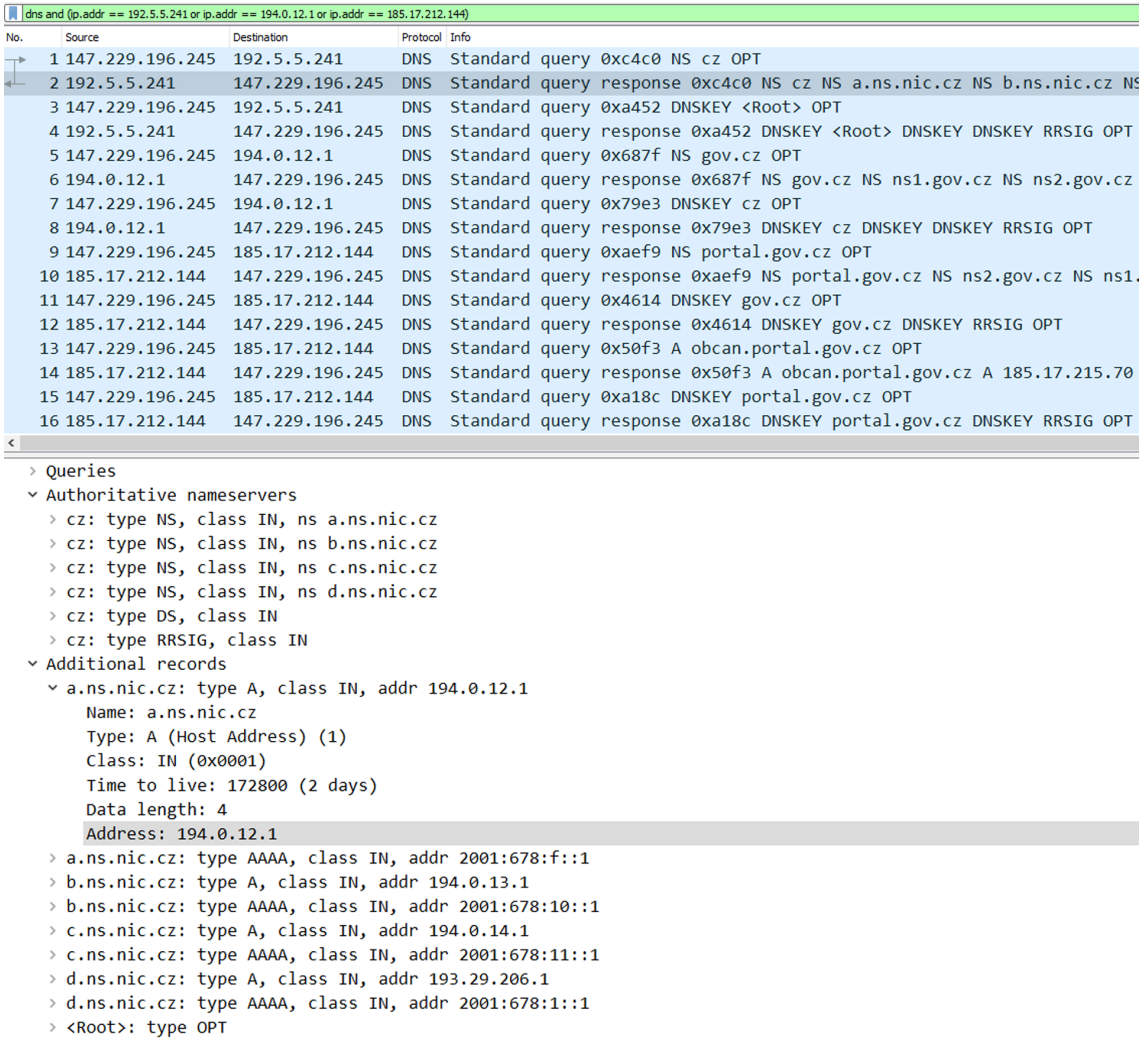


Obr. C.2.7: Nastavení aplikace Klient DNS pro dotaz se záznamem typu NS na kořenový DNS server F.

Zachytávání síťové komunikace ponechejte spuštěné a zvolte vhodný filtr pro vyfiltrování DNS komunikace. Měly by se objevit čtyři pakety, podobné paketům jedna až čtyři na na obr. C.2.8. První z nich je NS dotaz na požadovanou doménu „cz“, který kořenovému serveru říká, že očekáváme odpověď s názvem serveru, který nám poskytne DNS údaje o všech doménách v „cz“ prostoru. V odpovědi máme hned několik těchto jmenných názvů serverů (a.ns.nic.cz, b.ns.nic.cz...) a poté v sekci `Additional records` vidíme IPv4 i IPv6 adresy těchto serverů. Kromě NS záznamů si projděte také DS a RRSIG záznamy v tomto paketu. Do šablony na obrázku

[2] https://www.iana.org/domains/root/servers

C.2.6, která je součástí úkolu č. (5), vyplňte vaši IP adresu a také název a IP adresu kořenového serveru. Pakety podobné paketům číslo 3 a 4 souvisí se zjišťováním DNSKEY root zóny. Obsahují hodnoty veřejných klíčů a samozřejmě také RRSIG podpis těchto záznamů.



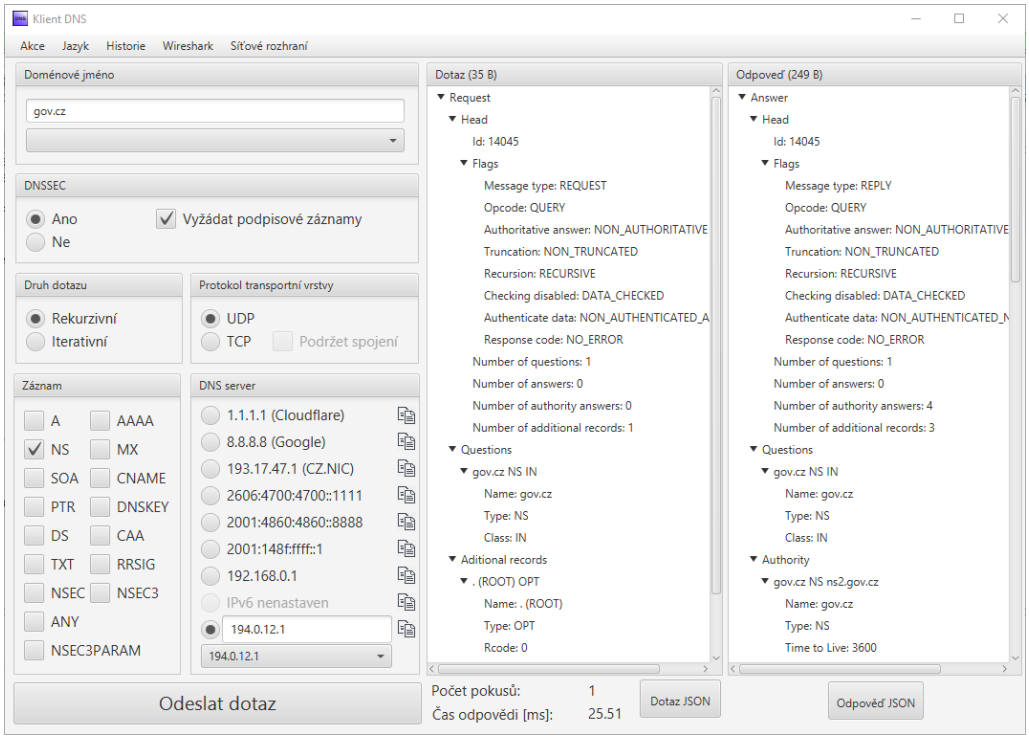
Obr. C.2.8: Komunikace mezi rekurzivním DNS serverem a ostatními servery DNS realizovaná pomocí koncového klienta.

Úkoly:

(1) Jaké položky se přenášejí v DNS záznamu typu RRSIG? Vyberte a stručně okomentujte alespoň 4.

Ze zjištěných IP adres DNS serverů v NS záznamu vyberte ten první dle abecedy a jeho název i IP adresu vyplňte do šablony úkolu. Nyní se budeme tohoto vybraného serveru dotazovat na jméno dalšího serveru, který nám řekne, kam se obrátit při komunikaci s doménou `portal.gov.cz`. Do pole v sekci `DNS server` v aplikaci

Klient DNS tedy vložte IP adresu prvního serveru dle abecedního pořadí jeho názvu. Dotazované doménové jméno změňte na `gov.cz`, typ záznamu na `NS` a zbytek parametrů ponechejte jako u předchozího odesílání požadavku, podobně jako na obr. C.2.9. Odešlete DNS dotaz a v případě, že se ve Wiresharku vyskytují více než 4 dříve popsané pakety, tak upravte filtr (např. pomocí IP adres), aby se po odeslání požadavku zobrazovalo pouze 6 DNS paketů, souvisejících s komunikací v aplikaci Klient DNS.



Obr. C.2.9: Nastavení aplikace Klient DNS pro odeslání NS dotazu na doménu gov.cz.

Dva poslední zachycené DNS pakety by měly odpovídat paketům 5 a 6 na obrázku C.2.8. Situace je zde obdobná jako v předchozím kroku, první paket je NS dotaz na doménu gov.cz a druhý paket je odpověď, která nám poskytuje názvy a IP adresy serverů. Upravte nastavení aplikace Klient DNS pro záznam typu DNSKEY na doménu cz a odešlete dotaz. V odpovědi se nám opět ukáže veřejný klíč včetně jeho podpisu v podobě RRSIG záznamu. Z šestého paketu vyberte ten server, jehož název je první dle abecedního řazení a zapište jej do obrázku č. C.2.6 a upravte nastavení aplikace Klient DNS obdobně jako v předchozím případě zpět na záznam typu NS, nyní s dotazem na doménu `portal.gov.cz`. Nové pakety ve Wiresharku by měly odpovídat paketům 9 a 10 na obrázku C.2.8. Opět proveďte i dotaz na záznam DNSKEY na doménu `gov.cz`. Jelikož jsme hledali

DNS server, který zná doménové jméno `obcan.portal.gov.cz` a již jsme jej našli v podobě DNS serveru domény `portal.gov.cz`, tak nyní se tohoto serveru můžeme zeptat přímo dotazem typu A na původní hledané doménové jméno. Přepněte tedy v aplikaci Klient DNS typ dotazu z DNSKEY na typ A a doménové jméno pozměňte na `obcan.portal.gov.cz`. Jako DNS server zvolte opět ten první dle abecedního řazení, který se nachází v předchozí DNS odpovědi záznamu typu NS a odešlete dotaz. Stejně jako v paketech 13 a 14 na obr. C.2.8 by se měla objevit DNS odpověď, která nám poskytne IP adresu pro naši hledanou doménu `obcan.portal.gov.cz`. Šestnáct zachycených paketů s použitým filtrem exportujte do souboru s názvem `„DNSSEC_simulace_rekurzivniho_serveru"` přes `File > Export Specified Packet...` Zkontrolujte zaškrtnutou položku `All packets` a také `Displayed`. Tento postup při exportu zajistí, že se do souboru uloží pouze právě vyfiltrované pakety. Tento soubor bude použit v následující kapitole pro analýzu v programu Wireshark. Reálný rekurzivní server by nyní ověřil podpisy a celý řetězec důvěry, ale v našem simulovaném příkladu s využitím aplikace Klient DNS toto ověření nedokážeme provést, i když k tomu teoreticky máme všechny potřebné informace. Dále by také rekurzivní server uložil všechny získané záznamy do cache paměti a zjištěnou IP adresu pro doménu `obcan.portal.gov.cz` by odeslal klientovi, který o ni požádal na začátku celého procesu.

Úkoly:

(2) Vyplňte všechny využité a zjištěné IP adresy a názvy DNS serverů do šablony na obrázku C.2.6, který znázorňuje postup zjišťování IP adresy pomocí rekurzivního DNS serveru.

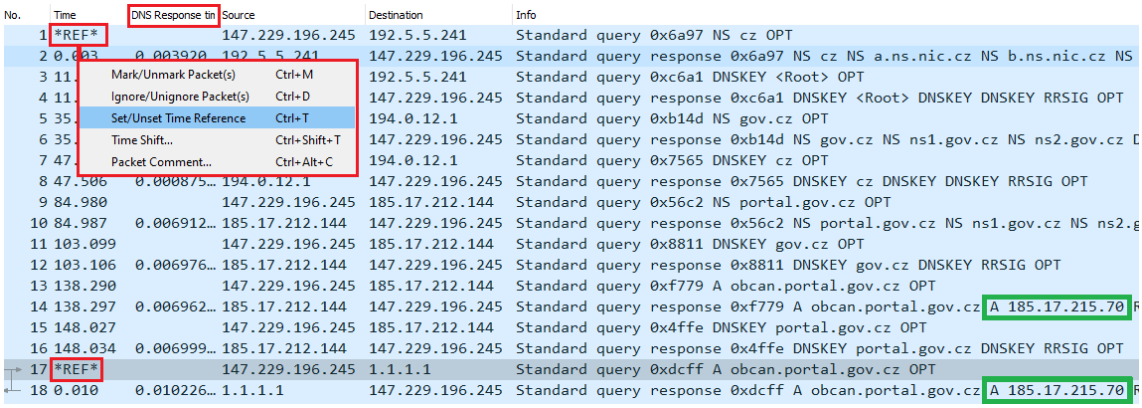
C.2.3 Wireshark analýza DNSSEC komunikace

V této části bude analyzována zachycená DNSSEC komunikace pomocí programu Wireshark. Zejména půjde o zobrazení času odezvy u jednotlivých DNS odpovědí a její zobrazení v podobě grafu a také zobrazeních dalších statistik. Dojde i na porovnání této odezvy s klasickým DNSSEC dotazem a odpovědí z pohledu klienta, který se pouze dotazuje rekurzivního serveru.

K tomu budeme potřebovat soubor se zachycenými pakety z předchozí kapitoly (DNSSEC_simulace_rekurzivního_serveru), ale také soubor s dotazem klienta a odpověď rekurzivního serveru. Tento soubor ještě nemáme připravený, takže si tyto pakety nyní zachytíme. V aplikaci Klient DNS definujte doménové jméno stejně jako v předchozí kapitole, konkrétně tedy `obcan.portal.gov.cz`. Povolte využití rozšíření DNSSEC, zvolte požadovaný záznam na typ `A` a jako DNS server vyberte Cloudflare server s IP adresou `1.1.1.1`. Ve Wiresharku rovnou definujte

filtr, který zobrazí pouze `DNS` pakety obsahující právě IP adresu `1.1.1.1`. V aplikaci Klient DNS odešlete dotaz na překlad doménového jména. Ve Wiresharku by se měl objevit DNS dotaz a k tomu odpovídající odpověď obsahující zjištěnou IP adresu. Tyto dva pakety obsahující IP adresu 1.1.1.1 exportujte pomocí `File > Export Specified Packet... > All packets + Displayed`. Soubor pojmenujte jako „`DNSSEC_klient_server"`.

Dále propojíme dva vytvořené soubory. Přes `File > Open` otevřeme první ze souborů (DNSSEC_simulace_rekurzivniho_serveru) a následně přes `File > Merge...` připojíme druhý soubor (DNSSEC_klient_server). Aktuálně by tak mělo být zobrazeno 18 paketů, stejně jako na obr. C.2.10. Dále nastavte časovou referenci na první paket z obou souborů pomocí označení daného paketu pravým tlačítkem myši a následně `Set/Unset Time Reference`, podobně jako na obr. C.2.10. Mělo by se jednat o pakety číslo 1 a 17.

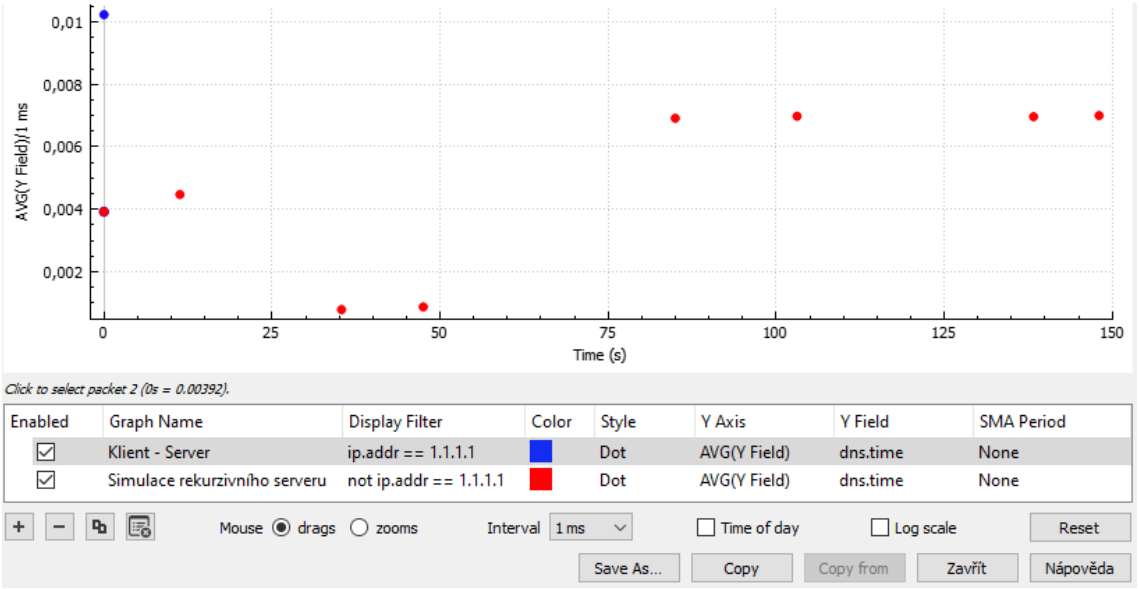


Obr. C.2.10: Propojené soubory v programu Wireshark a nastavení časové reference pro oba úvodní pakety.

Dalším krokem bude vytvoření nového sloupce s názvem `DNS Response Time`. Tento sloupec bude u DNS odpovědí zobrazovat časovou prodlevu od odeslání DNS dotazu. Pro vytvoření nového sloupce označte myší `první DNS odpověď (Standard query response)` a následně přejděte do dolní části obrazovky na detaily vybraného paketu. V detailech najděte část věnující se DNS protokolu, konkrétně `Domain Name System (response)`. Zde poté vyhledejte položku `Time` v hranatých závorkách. Pravým kliknutím myši na tuto položku otevřete nabídku a zvolte možnost `Apply as Column`. Tím se vytvoří nový sloupec s požadovaným nastavením, který bude zobrazovat časovou prodlevu DNS odpovědí, podobně jako na obr. C.2.10. Ještě nově vytvořený sloupec `Time` přejmenujte na `DNS Response Time` přes pravé tlačítko myši na záhlaví některého ze sloupců a poté `Edit Column > Title`. V poli `Fields` si povšimněte parametru `dns.time`, který vypočítává požadované hodnoty odezvy.

Hodnota v hranatých závorkách je totiž dopočítávána programem Wireshark a není reálně obsažena v jednotlivých paketech.

Ve vytvořeném sloupci tedy můžeme vidět časovou odezvu na DNS dotazy. V ukázkovém příkladu na obr. C.2.10 se časová odezva u simulace rekurzivního serveru pohybuje v rámci jednotek milisekund. U druhé situace, kdy jsme zachytili DNS odpověď od rekurzivního serveru s přeloženým doménovým jménem v podobě IP adresy, je časová odezva obdobná, konkrétně 10,2 ms.



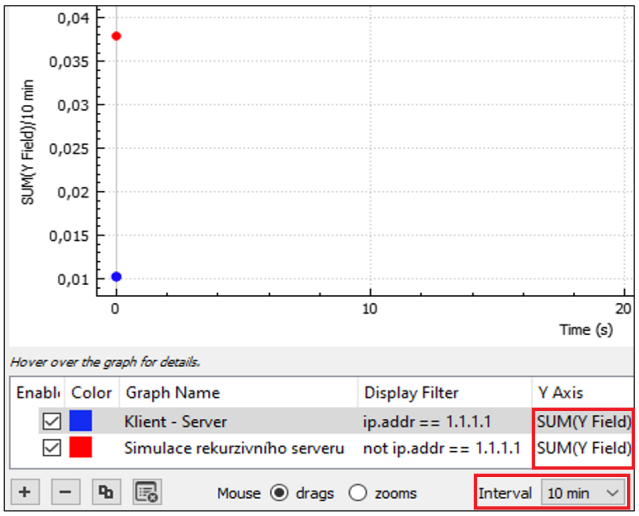
Obr. C.2.11: Propojené soubory v programu Wireshark a nastavení časové reference pro oba úvodní pakety.

Nyní si odezvu zobrazíme v podobě grafu přes `Statistics > I/O Graphs`. K vytvoření grafu bude potřeba zvolit správný filtr, abychom oddělili dvě dříve zmíněné situace. Jelikož chceme v grafu zobrazit všechny zachycené pakety, tak jednotlivé situace v grafu oddělíme podle toho, zda pakety obsahují nebo neobsahují IP adresu `1.1.1.1`, stejně jako na obr. C.2.11. Jednotlivé hodnoty v grafu vykreslíme jako body, zvolíme tedy ve sloupci `Style` možnost `Dot`. Dále je potřeba nastavit, které hodnoty mají být zobrazovány na ose Y. My nechceme zobrazovat počet paketů, ale dobu odezvy DNS. Toho docílíme tak, že ve sloupci `Y Axis` zvolíme `AVG(Y Field)`. Tím dojde ke zprůměrování (AVG) hodnot za jednotku času, kterou si zvolíme v části `Interval`. Zvolte nejkratší možný interval, tedy `1 ms`, aby v grafu ve všech situacích byly rozeznatelné jednotlivé DNS zprávy v podobě bodů. Reálně tak vytváříme průměr pouze z jedné hodnoty, protože mezi jednotlivými DNS odpověďmi máme prodlevu v řádu jednotek sekund. Ještě musíme definovat hodnotu `Y Field` v následujícím sloupci. Zvolíme zde dříve zmíněný parametr `dns.time`.

Nakonec grafy vhodně pojmenujeme a barevně odlišíme. Výsledný graf obsahující obě situace můžeme vidět na obr. C.2.11. Na ose X máme čas v sekundách od počátku zachytávání paketů. Pro první DNS zprávy v obou situacích je počátek v čase 0 sekund, díky dříve nastavené časové referenci. U navazujících DNS zpráv se čas odvíjí od toho, jak rychle jsme odesílali DNS dotazy v aplikaci Klient DNS. Na ose Y jsou námi nastavené hodnoty doby odezvy jednotlivých DNS zpráv taktéž v sekundách.

Nyní si dvě zachycené situace porovnáme také číselně. Přes tlačítko `Save As...` v okně grafu exportujte data do CSV souboru. Vytvořený soubor poté importujte do Excelu. Po importu vyfiltrujte pouze nenulové hodnoty a následně z hodnot u simulace rekurzivního serveru vypočítejte součet a průměr hodnot, tak jako na obr. C.2.12 (levá část). Následně čas porovnejte s dobou odezvy u samostatné DNS odpovědi v případě komunikace klient - DNS server. Co z tohoto porovnání vyplývá? Zjištěné hodnoty si zapište v rámci úkolů č. (3) a (4).

Column1	Column2	Column3
Interval start	Klient - Server Doba odezvy [s]	Simulace rekurzivního serveru Doba odezvy [s]
0.003	0	0,00392
0.01	0,01023	0
11.299	0	0,00447
35.347	0	0,00079
47.506	0	0,00088
84.987	0	0,00691
103.106	0	0,00698
138.297	0	0,00696
148.034	0	0,00700
Průměr	-	0,00421
Součet	0,01023	0,03791



Obr. C.2.12: Porovnání doby odezvy při simulaci rekurzivního serveru a při komunikace klient - DNS server v Excelu a upravený graf ve Wiresharku potvrzující zjištěné hodnoty.

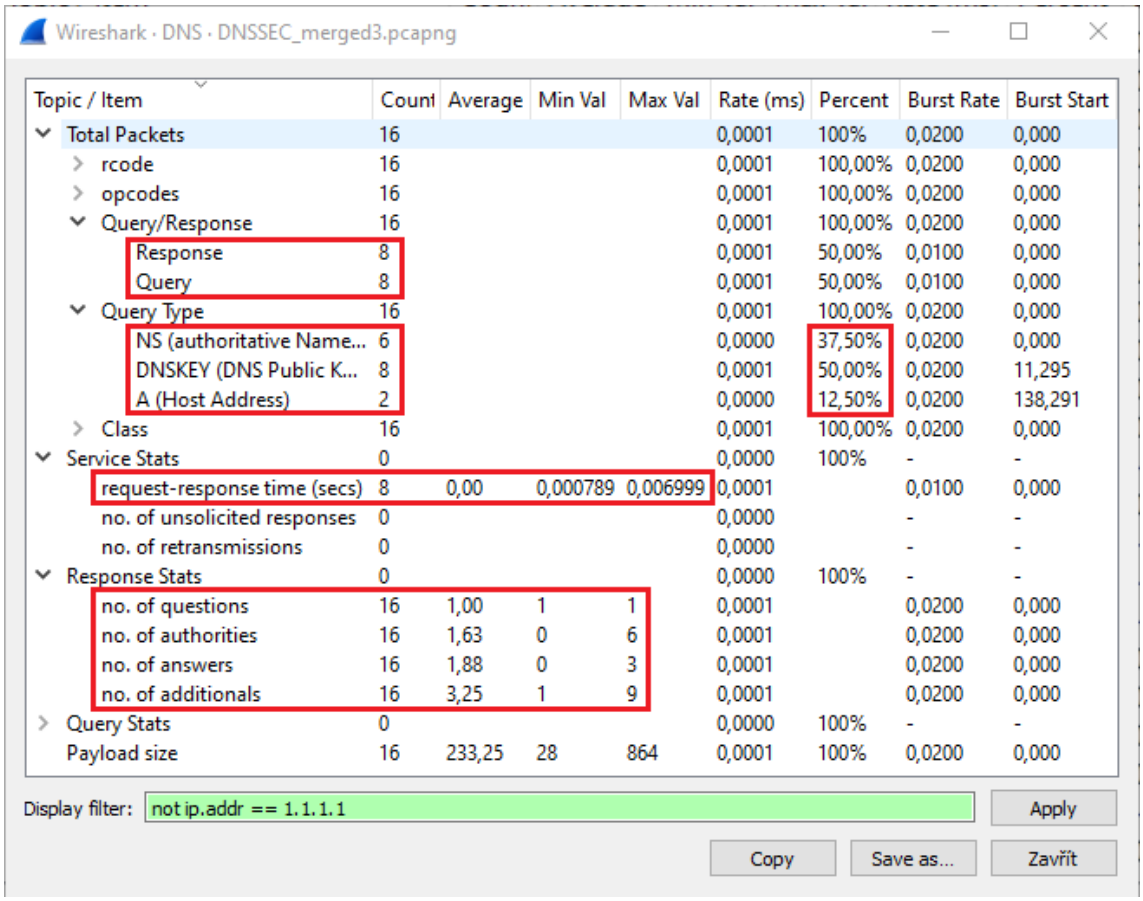
Úkoly:

(3) Jaký vám vyšel součet a průměrná hodnota doby odezvy v případě simulace rekurzivního serveru?

(4) Jakou dobu odezvy jste zjistili u DNS odpovědi v případě komunikace klient - DNS server?

V ukázkovém příkladě vyšel součet odezvy DNS odpovědí 37,91 ms. Samostatná DNS odpověď na klasický DNS dotaz při komunikaci klient - DNS server je přitom pouhých 10,23 ms. Tato hodnota je tedy menší než součet odezev u osmi DNS odpovědí při simulaci rekurzivního serveru. Je to zejména tím, že rekurzivní server má některé DNS záznamy uložené v paměti a nemusí se na ně pokaždé znovu dotazovat, čímž ušetří nejen svůj čas, ale také pásmo komunikačního kanálu a čas, respektive paměť dotazovaného serveru, který nebude muset na dotaz odpovídat. Může jít například o DNS dotazy zjišťující názvy a IP adresy DNS serverů pro domény nejvyššího řádu, jako je v našem ukázkovém případě doména `cz`. Na záznamy této národní domény se rekurzivního serveru bude nejspíše dotazovat spousta klientů a tak je nutné tento záznam uchovat v paměti po dobu jeho platnosti, kterou určuje hodnota TTL.

Nyní se vrátíme ke grafu ve Wiresharku, kde upravíme nastavení pro vykreslení grafu. Půjde o nastavení osy Y, kde změníme původní nastavení `AVG(Y Field)` na `SUM(Y Field)` a dále upravíme interval z `1 ms` na `10 min`, obdobně jako na obr. C.2.12. Díky této úpravě se nám v grafu objeví pouze 2 hodnoty, které jsou součtem (SUM) doby odezvy za 10 minut zachytávání paketů. Můžeme tak porovnat součet odezvy z Excelu s hodnotou v grafu. Odpovídá vaše odezva v grafu hodnotě vypočítané v Excelu? Doba odezvy při komunikaci klient - DNS server zůstává stejná, jako v předchozím případě, tedy 10,2 ms.



Obr. C.2.13: Podrobné statistiky vyfiltrovaných DNS paketů v programu Wireshark.

Poslední část analýzy se zaměří na obecné statistiky o DNS paketech. Statistiky zobrazíme přes `Statistics > DNS`. V dolní části okna aplikujte filtr, který odstraní pakety obsahující IP adresu `1.1.1.1` a filtr potvrďte tlačítkem `Apply`. Vidíme tak statistiky vztahující se pouze k DNS paketům v případě simulace rekurzivního serveru. Pro ukázkový příklad jsou statistiky zobrazeny na obr. C.2.13. Statistiky ukazují např. počet obsažených dotazů (query) a odpovědí (response) z celkového počtu vyfiltrovaných DNS paketů. Tento počet by měl být v ideálním případě vždy stejný, v našem případě by mělo jít o 8 dotazů a 8 odpovědí. Dále v kategorii `Query Type` můžeme vidět statistiky týkající se jednotlivých typů DNS záznamů. Statistika zobrazuje kolik paketů obsahuje NS, A nebo DNSKEY záznamy. Kromě počtu paketů je zde v pravé části i přepočet na procenta. Z dalších statistik se ještě podívejte na `request-response time (secs)` a také na `Response Stats`.

Úkoly:

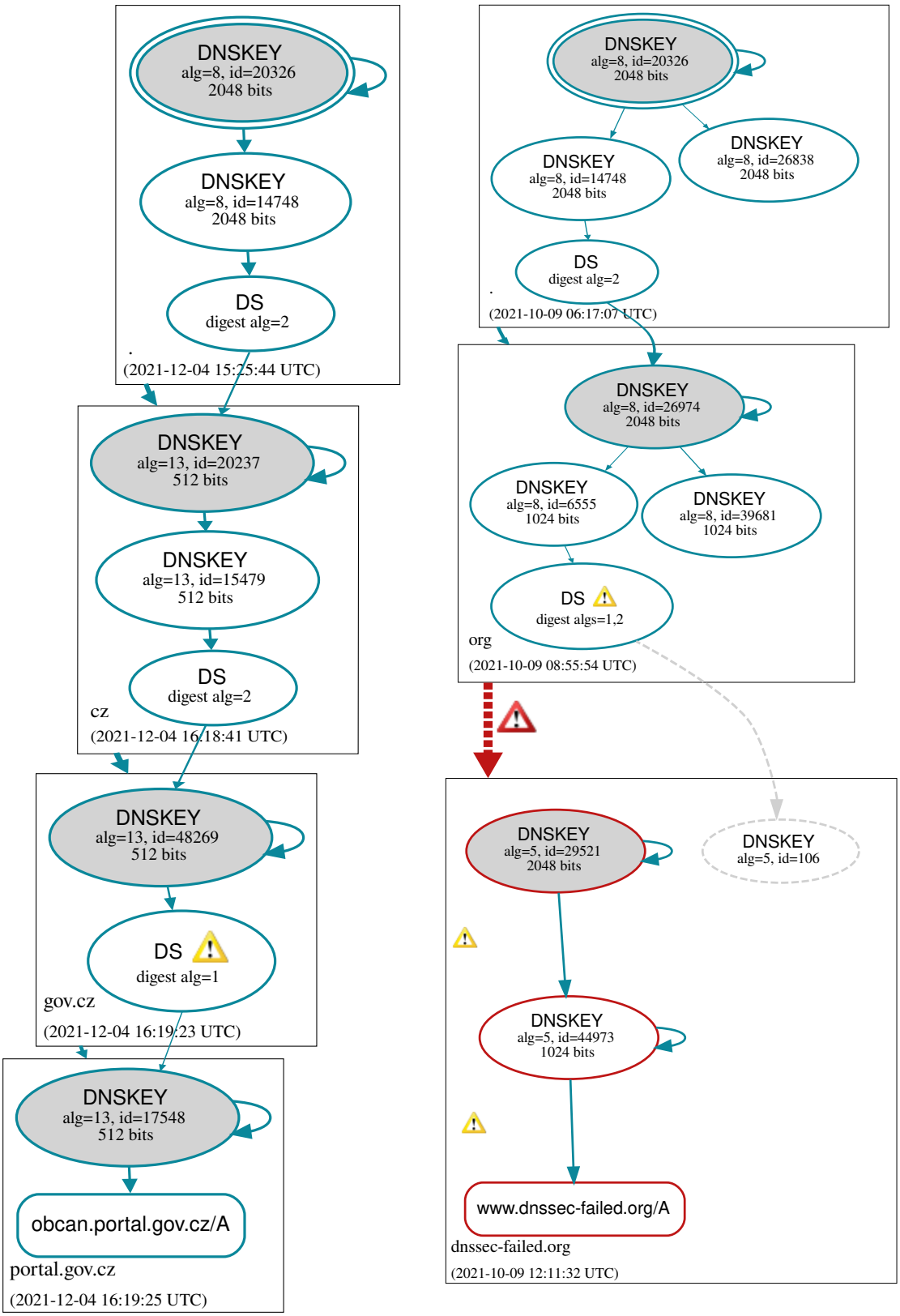
(5) Jaký je ve vašem případě nejvyšší počet doplňujících záznamů (Additional records) v rámci jednoho z vyfiltrovaných paketů?

C.2.4 Analýza pomocí aplikace DNSViz

Pro lepší orientaci v navazující DNS komunikaci ještě využijeme webovou aplikaci dnsviz.net. Na tomto webu zadejte opět doménové jméno `obcan.portal.gov.cz`. Stiskem tlačítka „Go“ proveďte analýzu a prostudujte si zobrazené výsledky. Ty by měly být podobné, jako na obr. C.2.14 (vlevo).

V prvním rámečku se zobrazují informace k dotazům na „root“ domény (tečka zobrazující se v levém dolním rohu rámečku, podobně jako „cz“ a „gov.cz“ domény v následujících krocích), které následně využívají kořenové servery. Názvy těchto serverů, jejich IP adresy a další podrobnosti se zobrazí po najetí myší na jednotlivé bubliny v diagramu. V jednotlivých krocích vidíme DNSSEC řetězec, který zajišťuje autentizaci a v posledním kroku vidíme finální dotaz typu A a po najetí na příslušné prvky diagramu se zobrazí odpovídající IP adresa. Jak si můžete povšimnout, tak ve všech krocích, se vyskytuje poznámka `Status: SECURE`. To znamená, že komunikace rekurzivního serveru s DNS servery je autentizována a získané informace jsou důvěryhodné.

Nyní v této aplikaci otestujeme dříve zmiňovanou doménu `utko.fekt.vut.cz`. Zadejte toto doménové jméno do vhodného pole na hlavní stránce aplikace DNSViz a spusťte analýzu. Zobrazí se nám 4 úrovně DNS komunikace, podobně jako v předchozím případě. Můžeme vidět, že je na všech úrovních zajištěno ověření autenticity (`Status: SECURE`).



Obr. C.2.14: Webová aplikace dnsviz.net zobrazující strukturované odpovědi na doménu obcan.portal.gov.cz (vlevo) a dnssec-failed.org (vpravo).

Jedinou změnou oproti předchozímu analyzovanému doménovému jménu je, že nyní dostáváme upozornění na doméně vut.cz a jejích subdoménách o tom, že tyto domény využívají k podpisu záznamů algoritmus, který již není doporučeno používat.

Dále vyzkoušejte do webové aplikace zadat doménové jméno `dnssec-failed.org`. Tato doména je speciálně upravena tak, aby vracela chybu při autentizaci. Jak lze vidět na diagramu (podobně jako na obr. C.2.14 vpravo), tak doména „org“ je zabezpečená a tedy komunikace s kořenovými a TLD DNS servery je autentizovaná. Ale při dotazu na celou doménu již dochází k chybě při autentizaci, což vede k tomu, že se zjištěná IP adresa nepošle klientovi, protože byla (záměrně) podepsána nevalidním podpisem.

Úkoly:

(6) Jmenujte 3 konkrétní vlastnosti, které lze najít v podrobnostech u jednotlivých prvků ve schématu ve webové aplikaci dnsviz.net?

(7) Vyhledejte v aplikaci dnsviz.net doménu csfd.cz. Je komunikace s touto doménou z hlediska DNS zabezpečená na všech úrovních?

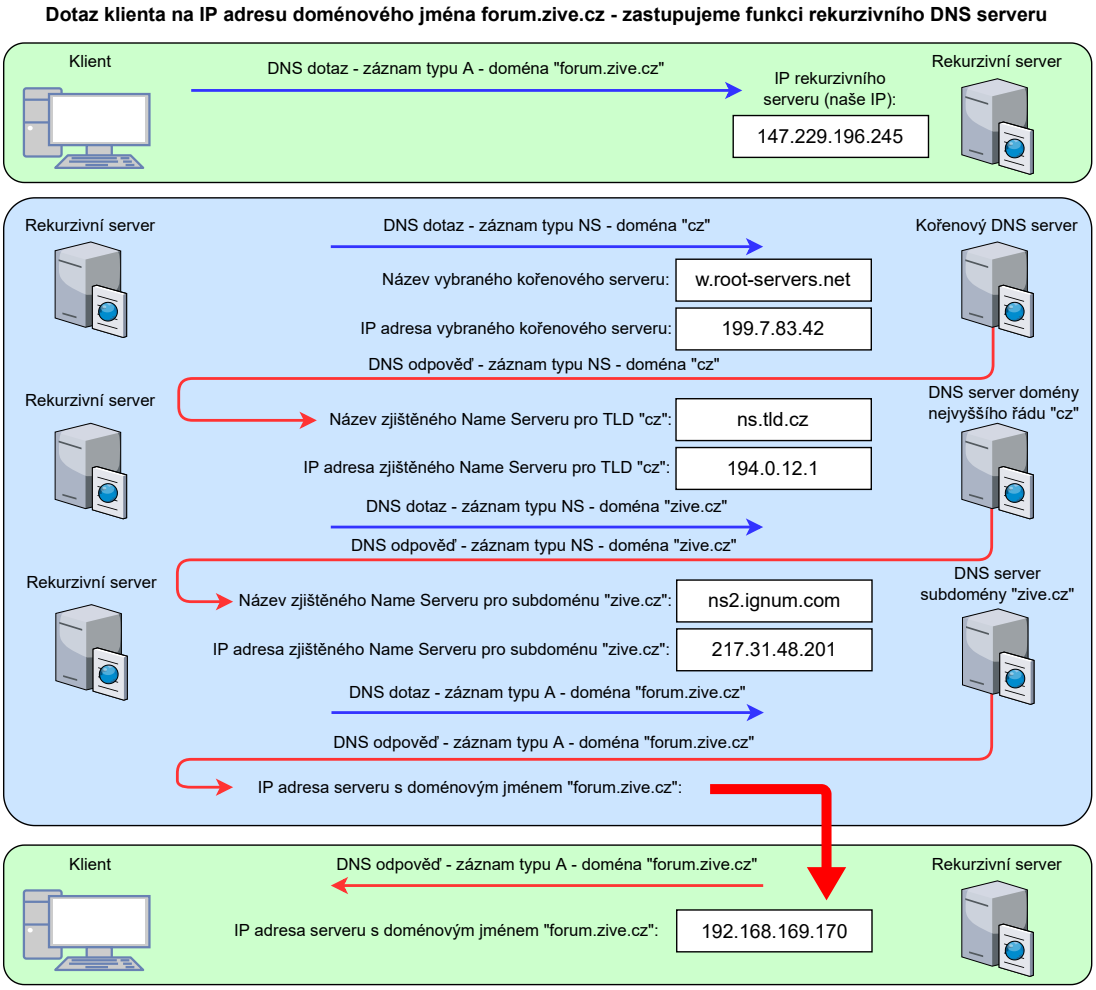
(8) Dohledejte jmenné názvy (NS záznamy) doménových serverů pro doménu google.com.

(9) Co nastane v případě, kdy neobdržíme některý z požadovaných DNSKEY nebo DS záznamů?

Nyní se budeme v rámci samostatného úkolu č. (10) věnovat schématu na obr. C.2.15. Toto schéma simuluje postup zjišťování IP adresy pomocí rekurzivního serveru a obsahuje chyby, které je potřeba opravit. Chyby mohou být ve jmenném názvu serverů nebo v uvedených IP adresách. Chyby odhalíte tak, že budete postupovat podobně jako u úkolu č. (5). Je tedy potřeba volit správné typy DNS dotazů na doménová jména a tyto dotazy odesílat na vhodné DNS servery.

Úkoly:

(10) Vyhledejte a opravte 3 chyby týkající se názvů a IP adres využitých DNS serverů ve schématu na obr. C.2.15.



Obr. C.2.15: Šablona pro samostatný úkol č. (10) zobrazující schéma DNS komunikace, které obsahuje chyby v podobě špatného jmenného názvu serverů nebo špatných IP adres.

D Kompletní návod pro čtvrtý vytvořený simulační scénář

ÚLOHA č. 4

Plánování a přidělování adresního prostoru, využití privátních adres, NAT překladu a úprava směrovacích tabulek.

D.1 Teoretický úvod

V tomto cvičení se budeme věnovat plánování adresního prostoru. Vyzkoušíme si přiřadit dostupný adresní prostor několika sítím a to jak za použití veřejných, tak i privátních IP adres. S tím se také v případě IPv4 úzce pojí využití NAT (Network Address Translation) překladu adres, který nám umožňuje pod jednu veřejnou IP adresu „skrýt“ celou podsíť s využitím privátních adres. Pro jednotlivé situace budou vytvořeny ukázkové příklady a také šablony, do kterých se budou zapisovat zjištěné informace a nebo také vypracované směrovací tabulky.

D.1.1 Adresy IPv4

Internet Protocol verze 4 definuje parametry tzv. IP adres, které slouží k identifikaci síťových a koncových zařízení v prostředí internetu. Adresa slouží k rozlišení jednotlivých zařízení a proto je vyžadováno, aby každé z těchto zařízení mělo vlastní unikátní IP adresu, která se v rámci internetu neopakuje a je tak možné přesně směrovat pakety internetem až k požadovanému a jasně identifikovatelnému cíli. IP adresy verze 4 jsou 32 bitové hodnoty, které jsou většinou zapisovány v desítkové soustavě v podobě čtyř oktetů. Oktety jsou odděleny tečkou. Vzhledem k tomu, že adresy mají velikost 32 bitů, tak je jasné že je omezený počet těchto adres. Konkrétně jde o 2^{32} adres. Tento počet je při stále rostoucím množství zařízení a sítí nedostatečný. Unikátní adresu totiž musí mít síťová zařízení, jako jsou směrovače a také všechna koncová zařízení jako osobní počítače, servery, mobilní telefony, Smart TV a další prvky IoT (Internet of Things).

Proto bylo nutné rozdělit IP adresy do veřejných a privátních rozsahů. Kdy veřejné jsou unikátní v rámci internetu (WAN) a privátní se používají pouze v rámci lokálních sítí (LAN). U lokálních sítí je nutné dodržet jedinečnost adres v rámci jedné sítě, ale v různých lokálních sítích se adresy mohou opakovat, protože nejsou směrovatelné za hranice lokální sítě. Komunikace z lokální sítě do veřejného internetu pak probíhá pomocí NAT překladu IP adres. Konkrétně překlad privátních adres koncových zařízení na veřejnou adresu výchozí brány a naopak.

Ukázka veřejné IPv4 adresy: `147.229.2.90`
Privátní rozsahy IPv4 adres: `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`

D.1.2 Adresy IPv6

Internet Protocol verze 6 pracuje na stejném principu, jako výše uvedené IPv4, akorát k identifikaci zařízení využívá 128 bitové adresy, místo 32 bitových u IPv4. Zde už by byla IP adresa zapsaná v desítkové soustavě moc dlouhá a tak se využívá hexadecimální zápis adres. Zápis je možné zkrátit vynecháním nul na začátku každé skupiny hexadecimálních znaků, které jsou do skupin zapisovány po čtyřech znacích. Skupiny jsou mezi sebou odděleny dvojtečkou a ne tečkou jako u IPv4. Pokud je celá skupina složena z nul, nebo na sebe dokonce navazuje více skupin se samými nulami, tak je možné tuto část adresy nahradit pouze dvěma dvojtečkami, toto zkrácení však lze použít pouze jednou pro jednu IPv6 adresu, viz příklad níže.

Ukázka veřejné IPv6 adresy: `2001:0db8:0001:0000:0000:0AB9:0000:01c2`
Zkrácený zápis veřejné IPv6 adresy: `2001:db8:1::ab9:0:1c2`

D.1.3 Přiřazení a NAT překlad IP adres

Adresy IPv4 i IPv6 mohou být přiřazeny staticky nebo dynamicky. Staticky znamená, že každému zařízení v síti pevně určíme IP adresu. Toto řešení je vhodné při malém počtu zařízení v případě, že požadujeme aby konkrétní zařízení vždy vystupovalo pod jednou konkrétní IP adresou. Dynamické přiřazení adresy přiřadí zařízením vždy některou z volných IP adres z přiděleného rozsahu pomocí protokolu DHCP (Dynamic Host Configuration Protocol), případně DHCPv6 pro IPv6. V IPv6 sítích se však častěji používá tzv. automatická bezestavová konfigurace SLAAC (StateLess Address AutoConfiguration), kdy si koncové stanice určují konkrétní adresu sami na základě prefixu sítě, který do sítě vysílá směrovač. Automatické (dynamické) přidělování adres je v IPv4 v dnešní době využíváno v mnoha klasických sítích. Je vhodné při vyšším počtu zařízení a v situacích kdy dochází k četnému střídání (připojování a odpojování) různých zařízení v dané síti.

NAT překlad IPv4 adres je funkce, kterou mohou disponovat směrovače. Tato funkce zaměňuje původní privátní zdrojové adresy paketů od stanic z lokální sítě za jednu jedinou veřejnou adresu, pod kterou vystupuje celá síť za směrovačem. Tento proces platí pro pakety odchozí z dané (pod)sítě. U příchozích paketů na směrovač z Internetu, se překlad provádí opačně. NAT, respektive tabulka překladu se neřídí pouze IP adresami, ale využívá také porty, pro určení konkrétních aplikací na koncových zařízeních. Na směrovačích se původní porty překládají na čísla portů z vyhrazeného rozsahu. Jednou z hlavních výhod je úspora veřejných IPv4 adres a nevýhodou může být nemožnost využití protokolů jako třeba SCTP (Stream Control Transmission Protocol) při současném použití NAT překladu.

D.2 Realizace scénáře

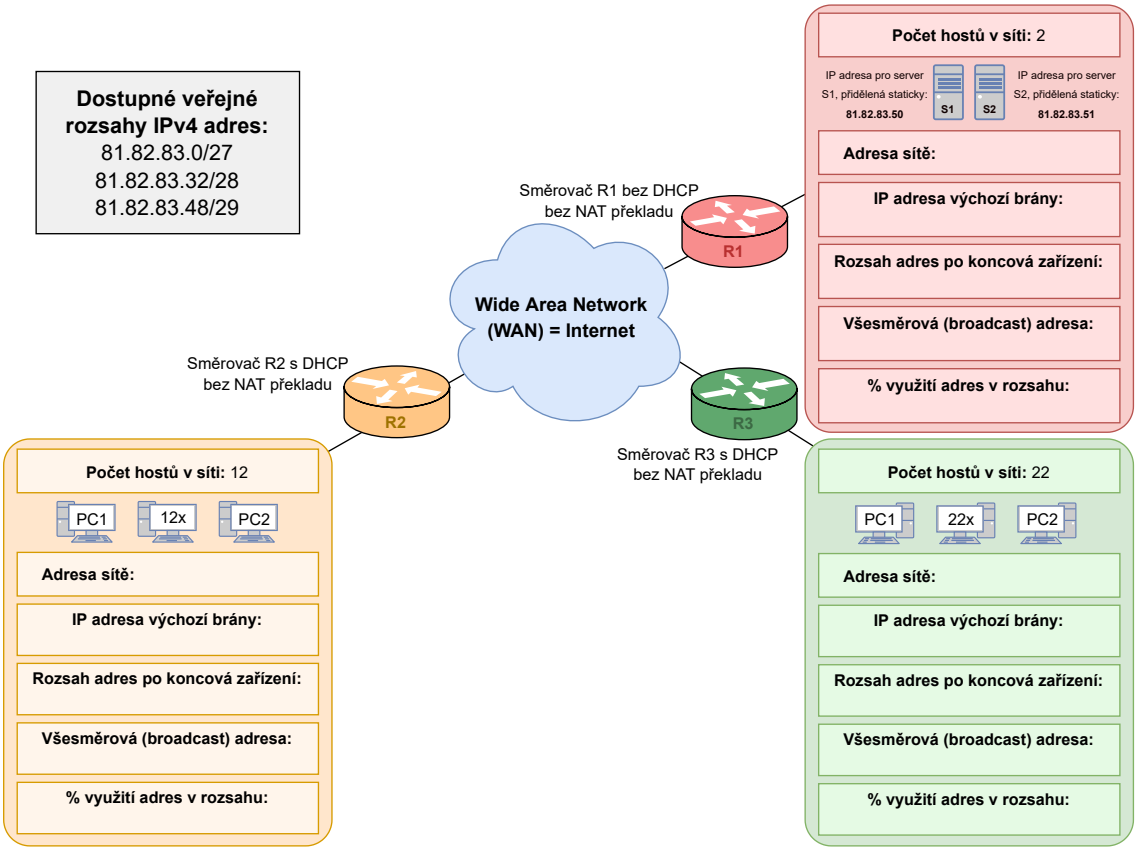
D.2.1 Veřejné IP adresy bez využití NAT překladu

V následující části budeme využívat pouze IP adresy z veřejného rozsahu. Prozatím zde nevyužijeme možnost NAT překladu, z čehož vyplývá, že každý host musí mít svoji unikátní IP adresu, která se nebude opakovat v jiných podsítích v tomto příkladu a zároveň ani v jiných veřejných sítích v rámci internetu.

Budeme pracovat se schématem sítí, které je zobrazeno na obr. D.2.1. Vidíme zde tři barevně odlišené sítě, kdy každá obsahuje různý počet stanic (LAN), kterým je potřeba přidělit IP adresy ze získaného rozsahu. Routery jsou u ukázkového příkladu propojeny v rámci internetu (WAN), což představuje modrý oblak. U této veřejné sítě předpokládáme, že je schopna přenášet pakety mezi koncovými sítěmi. Adresování v rámci této veřejné sítě pro jednoduchost v této ukázce řešit nebudeme. Dostupné veřejné rozsahy IPv4 veřejných adres se pro tuto ukázku skládají z rozsahů `81.82.83.0/27`, `81.82.83.32/28` a `81.82.83.48/29`.

První červená síť za směrovačem R1 obsahuje celkem dva servery, kterým je potřeba přiřadit IP adresy. Ve druhé síti chceme připojit 12 osobních počítačů a ve třetí síti za směrovačem R3 máme 22 počítačů bez přidělené adresy. Úkolem tedy je přiřadit jednotlivé výše uvedené rozsahy IP adres k uvedeným sítím tak, aby byl rozsah využit co nejefektivněji. Nechceme například síti přiřadit rozsah, ve kterém je k dispozici 126 IP adres, když se v síti nachází pouze jednotky počítačů nebo serverů. Docházelo by tak ke zbytečnému plýtvání veřejných IPv4 adres, kterých máme pouze omezené množství.

Dále tedy musíme pro jednotlivé rozsahy určit počet hostů, kterým dokáží přiřadit adresu. K tomu nám poslouží `maska sítě`, jejíž hodnota se uvádí za lomítkem daného rozsahu. Z hodnoty masky sítě můžeme určit hodnotu tzv. `wildcard masky`. V případě zápisu za lomítkem se jedná o rozdíl hodnoty 32 a aktuální hodnoty masky sítě. Hodnota 32 je teoreticky nejvyšší možná hodnota pro masku sítě, která vychází z toho, že maska sítě se skládá stejně jako IP adresy ze 32 bitů (4 oktety). V případě zápisu wildcard masky za lomítkem můžeme zjistit možný počet IP adres v dané síti tak, že číslo dva umocníme na tuto hodnotu wildcard masky. Pro zjištění maximálního možné počtu zařízení v síti je nutné ještě odečíst hodnotu 2, protože u IPv4 je první možná adresa z rozsahu adresa sítě a poslední adresa z rozsahu je broadcast adresa.

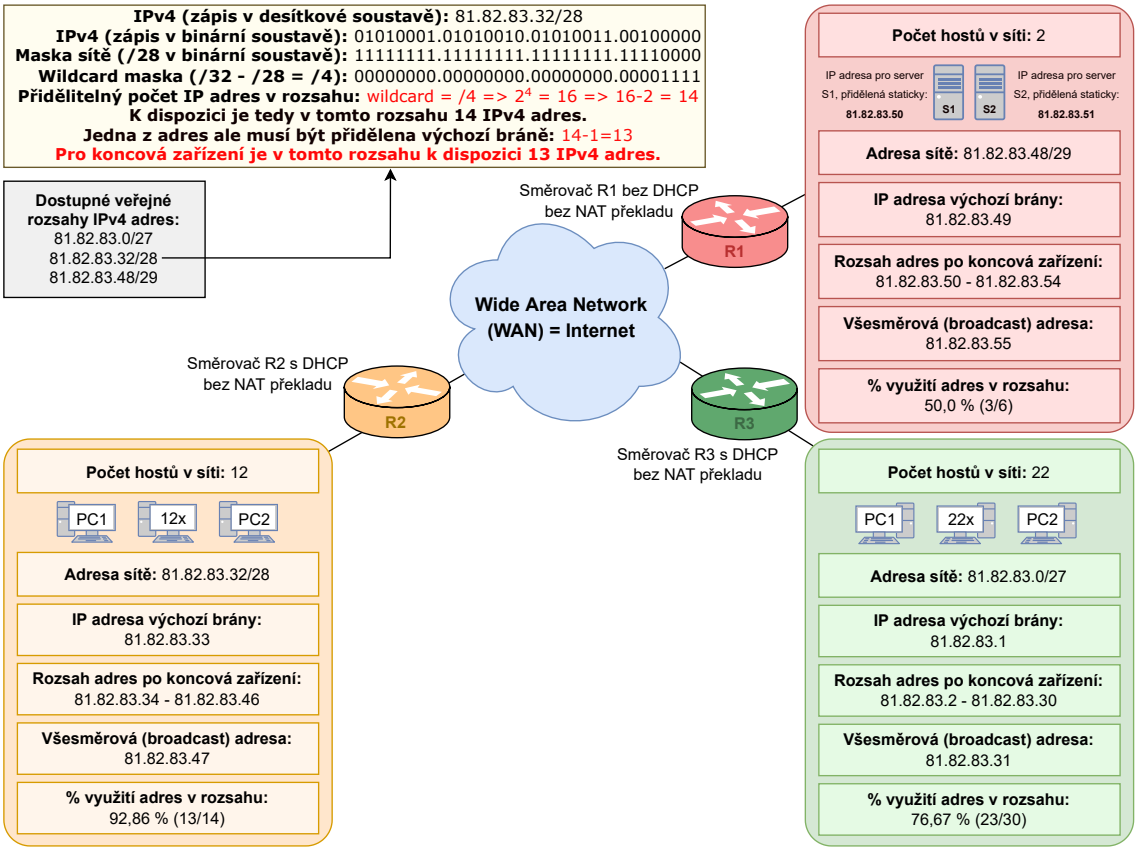


Obr. D.2.1: Šablona obsahující schéma sítí, pro které budeme hledat správný a co nejefektivnější rozsah veřejných adres.

Pro první rozsah sítě 81.82.83.0/27 bude tedy výpočet vypadat takto:
Počet bitů Wildcard masky: $32 - 27 = 5$
Celkový počet IP adres v daném rozsahu: $2^5 = 32$
Pro možný počet hostů musíme od vypočítané hodnoty odečíst IP adresu využitou jako adresu sítě a také adresu pro broadcast.
Přidělitelný počet IP adres v daném rozsahu: $32 - 2 = 30$
Adrese sítě odpovídá první možná adresa z rozsahu: 81.82.83.0
Adresa směrovače je většinou následující možná adresa: 81.82.83.1
Možný počet hostů (koncových zařízení) v daném rozsahu: $30 - 1 = 29$
Adresy využitelné pro hosty v dané síti: $81.82.83.2 - 81.82.83.30$
Broadcast adrese odpovídá poslední adresa z rozsahu: 81.82.83.31

Podobně budeme postupovat i u ostatních rozsahů. U druhého rozsahu `81.82.83.32/28` nám vyjde, že počet možných hostů, kterým lze přidělit IP adresy je 13. Pro třetí rozsah `81.82.83.48/29` je to 5 koncových zařízení. Pro náš ukázkový

příklad z obrázku D.2.1 tedy vyjde správné přidělení rozsahů veřejných IP adres tak, že nejmenší červené síti bude stačit rozsah `81.82.83.48/29`. Je zde možné přiřadit až 6 IP adres jednotlivým zařízením, my ale chceme připojit pouze dvě zařízení v podobě serverů. Pro výpočet procenta využití adresního prostoru však musíme započítat i směrovač, takže máme použité 3 adresy z 6-ti dostupných. Z toho vychází procento využití IP adres v této síti 50 %. Oranžové síti přiřadíme rozsah `81.82.83.32/28`, kam je možné umístit celkem 12 koncových zařízení a směrovač. Procento využití adres je tedy 92,86 % (13/14 = 0,9286). A největší síť, která čítá 22 koncových zařízení bude disponovat rozsahem `81.82.83.0/27`, kde je možné přiřadit až 30 IP adres. Procento využití IP adres v tomto rozsahu je zde 76,67 % (23/30 = 0,7667). Takto dojde k co nejefektivnějšímu rozdělení a přidělení jednotlivých veřejných rozsahů. Výsledky této části jsou ukázány také na obr. D.2.2.



Obr. D.2.2: Vyplněná šablona s řešením ukázkového příkladu s veřejnými IP adresami.

Přidělení konkrétních IP adres koncovým zařízením může proběhnout staticky (manuálně) nebo automaticky pomocí DHCP (Dynamic Host Configuration Protocol). V ukázkách na obr. D.2.1 a D.2.2 mají dynamické přidělování IP adres (DHCP) povolené směrovače R2 a R3. V takovém případě dojde směrem od

koncového zařízení dotaz na směrovač na automatické přidělení IP adresy. Směrovač vybere některou z volných IP adres z rozsahu a nabídne ji konkrétnímu zařízení. Takže například ve žluté síti chceme připojit nejdříve počítač PC1. Máme aktuálně volné všechny adresy pro koncová zařízení z rozsahu `81.82.83.34 - 81.82.83.46`. K výběru adres z rozsahu dochází buď náhodně a nebo může dojít k přidělování té stejné adresy dle MAC adresy daného zařízení. V našem ukázkovém příkladu tak zařízení PC1 dostane přidělenou IP adresu např. `81.82.83.34`. Obdobná situace pak nastane při připojení dalších zařízení do této sítě. Rozdíl bude pouze v tom, že druhý počítač PC2 již nemůže obdržet adresu, která byla přidělena PC1. Zařízení PC2 tak obdrží např. adresu `81.82.83.35`.

V případě červené sítě za směrovačem R1, kde se v síti nachází pouze pár serverů, požadujeme statické přiřazení IP adresy oproti dynamickému přiřazení (DHCP). Správce dané sítě tak musí mít v rámci směrovače přehled o aktuálně přidělených a volných adresách z dostupného rozsahu pro danou síť. V případě připojení nového zařízení by mu IP adresu vybral a přiřadil manuálně. Takže například prvnímu serveru S1 jsme vybrali a staticky přiřadili adresu `81.82.83.50`. Druhému serveru S2 poté adresu `81.82.83.51`.

Při použití veřejných adres je komunikace sítí mezi sebou možná jen díky správné konfiguraci směrování, kterému se budeme podrobněji věnovat až v navazujících kapitolách. Dále je také potřeba si uvědomit, že při takovémto využití veřejných IPv4 adres není potřeba využívat funkce překladu adres NAT.

Samostatná práce s veřejnými IP adresami:

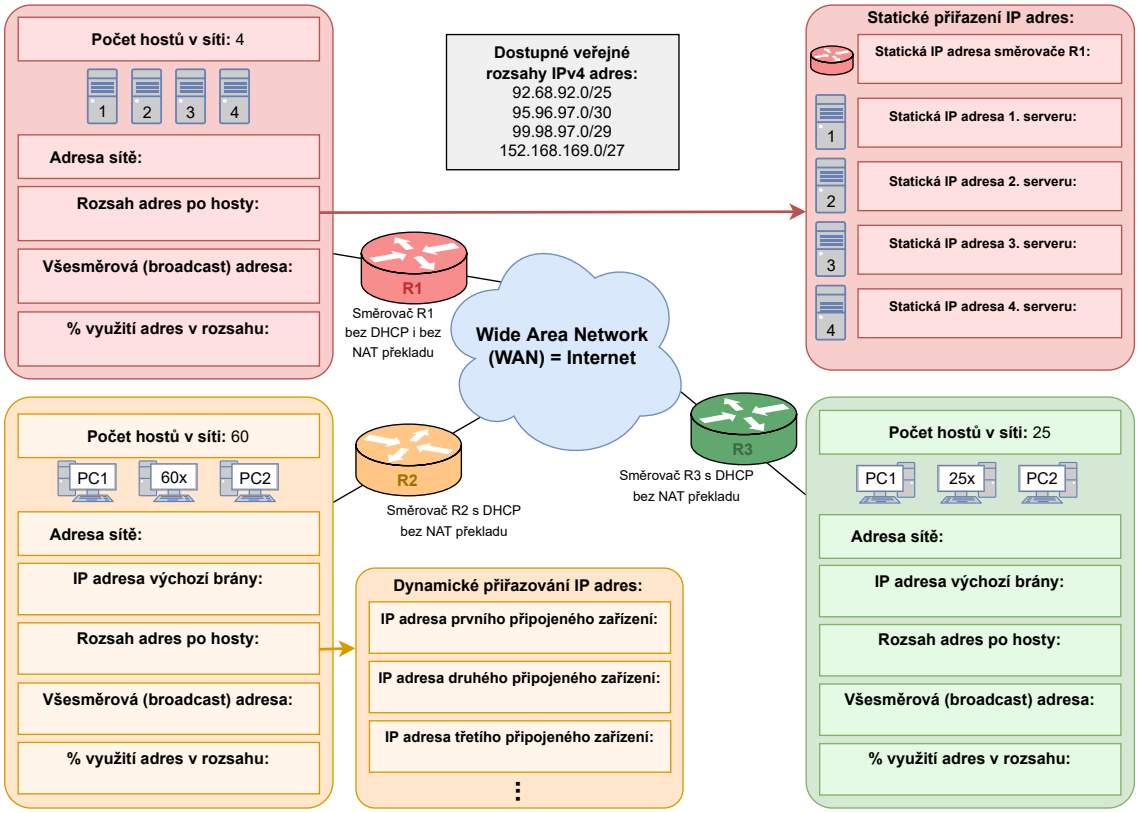
Nyní si přiřazení veřejných adres vyzkoušíte na dalším příkladě, který vychází z obr. D.2.3. V šabloně jsou uvedeny 4 dostupné veřejné rozsahy IP adres, ze kterých chceme vybrat 3 co nejefektivněji, aby nedocházelo ke zbytečnému plýtvání adres. Zmíněné procento využití IP adres vypočítáte podobně jako v ukázkovém příkladu. Můžeme tedy určit kolik IP adres bude přiřazeno a využito a tím pádem i kolik jich zůstane z přiřazeného rozsahu nevyužitých, což může být důležité při případném rozšiřování sítě v budoucnu.

Úkoly:

(1) Přiřaďte rozsahy veřejných IPv4 adres a doplňte další údaje o sítích do šablony na obr. D.2.3.

(2) Kolika zařízením lze přiřadit IP adresu ve zbývajícím nevyužitém rozsahu adres z obr. D.2.3.

(3) Stačil by některé ze sítí na obr. D.2.3 i rozsah s delší hodnotou masky sítě, než jí byl přidělen v úkolu (1)? Uveďte příklad takového rozsahu.



Obr. D.2.3: Šablona pro úlohu (1) obsahující schéma sítí, pro které budeme hledat správný a co nejefektivnější rozsah adres.

D.2.2 Privátní IP adresy bez využití NAT překladu

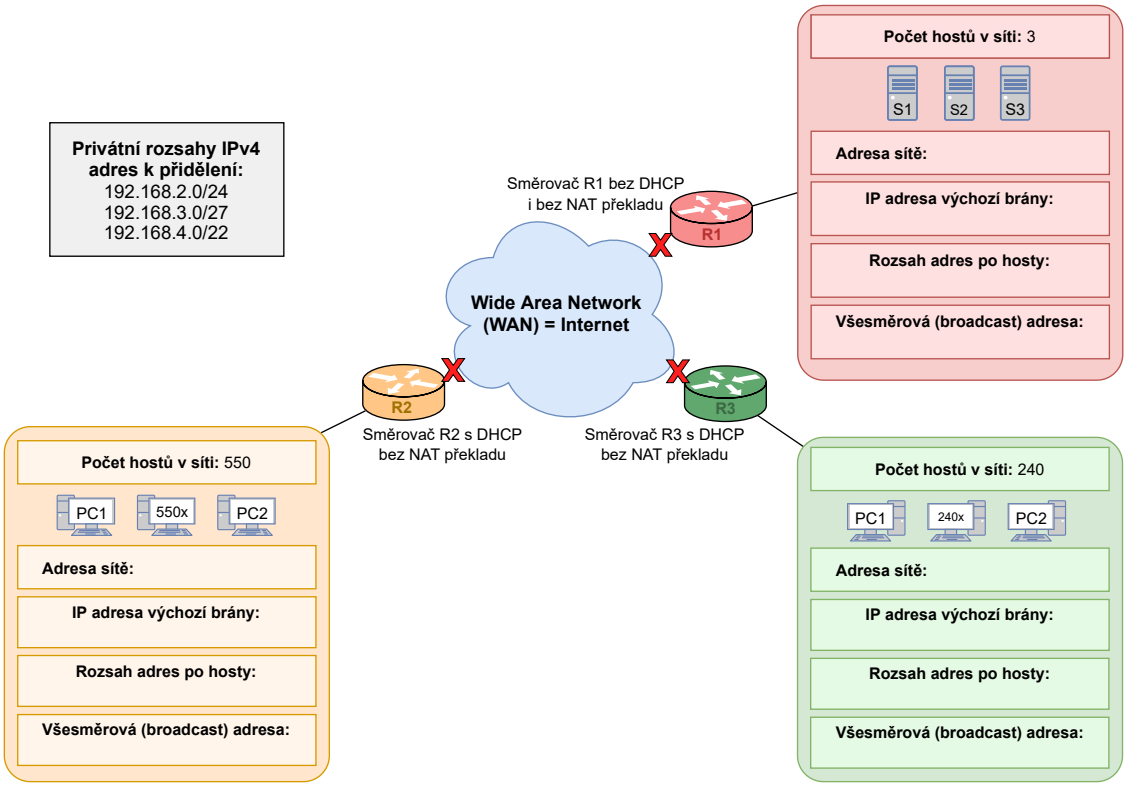
V této části budeme pracovat pouze s privátními adresami. Vzhledem k tomu, že opět ještě nevyužijeme NAT překlad na směrovačích, tak jednotlivé sítě v této části nebudou moci komunikovat mezi sebou. Vzniknou tak izolované sítě, kde bude komunikace zajištěna pouze mezi zařízeními v lokální síti. Toho lze využít například ve firemním či průmyslovém prostředí, kde není vyžadována online komunikace s okolním světem, ale pouze komunikace mezi zařízeními v dané firmě (například ve stejné budově, v jednom areálu…).

Privátní adresy mají speciálně přiřazené rozsahy, které se nevyskytují v rámci veřejných adres, nejsou směrovatelné a nejsou ani globálně unikátní. Můžeme tedy ve více lokálních sítích narazit na stejné privátní adresy.

Konkrétně jde o adresy v rozsazích:

```
10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
```

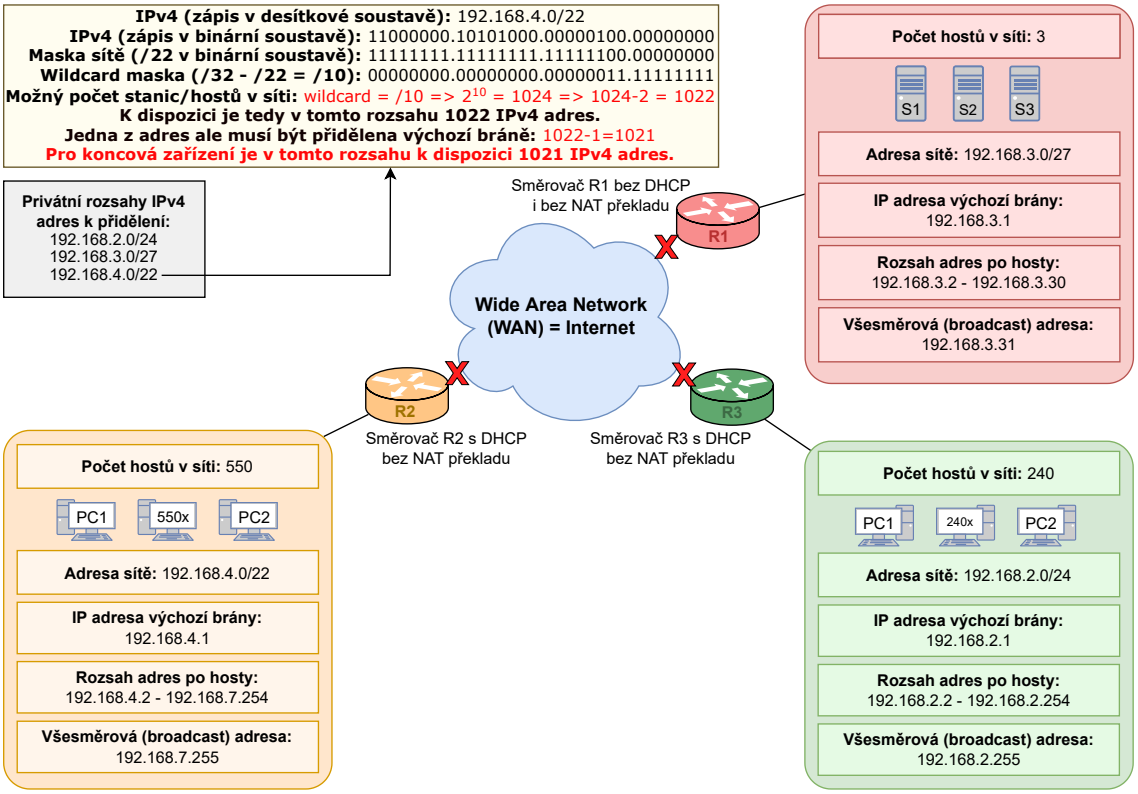
Na obr. D.2.4 máme opět schéma sítí, kterým budeme přiřazovat privátní adresy. Výpočet pro určení možného počtu IP adres, které je možno v daném rozsahu přiřadit je obdobný, jako v případě veřejných adres. Nejmenší červená síť požaduje IP adresy pro 3 koncová zařízení. Nejvhodnější rozsah je tedy ten s nejvyšší hodnotou masky sítě a tedy rozsah `192.168.3.0/27`. Oranžová síť požaduje až 550 IP adres a zelená síť 240 adres. Zelená síť tak obsahuje méně zařízení a přiřadíme jí tak rozsah IP adres `192.168.2.0/24` a největší oranžové síti zbývá rozsah s nejnižší hodnotou masky sítě a to `192.168.4.0/22`. Přidělení adres by proběhlo obdobně jako v případě v předchozí kapitole. Na směrovači v červené síti je vypnuté dynamické přiřazování IP adres (DHCP) a privátní adresy tak budou muset být přiřazeny manuálně. U oranžové a zelené sítě je možné využít funkci automatického přidělení privátních IPv4 adres pomocí DHCP.



Obr. D.2.4: Šablona obsahující schéma sítí, pro které budeme hledat správný rozsah privátních IP adres.

U privátních adres zpravidla nemusíme volit malé rozsahy IP adres tak, jak je tomu v ukázce na obr. D.2.4. Můžeme klidně využít celý rozsah, jako je např. `192.168.0.0/16`. K omezení rozsahu je v tomto případě přistoupeno z toho důvodu, aby bylo v ukázce i následném úkolu jasné, které síti se má přiřadit který rozsah IP adres. Pokud reálně adresujeme sítě s privátními adresami, můžeme použít i mnohem větší síťové rozsahy, než by bylo nezbytné a než uvádí ukázkový příklad. V našem příkladu, kde máme privátní sítě od sebe odděleny veřejnou částí, bychom mohli klidně použít úplně stejný rozsah adres pro každou z těchto izolovaných sítí, protože privátní adresy nejsou přes veřejné rozsahy směrovatelné.

Řešení pro ukázkový příklad s privátními adresami je zobrazen na obr. D.2.5. Jedinou novinkou vzhledem k zadaným hodnotám masky sítě je to, že v případě kdy máme síť s maskou sítě kratší než 24, tak je zde možné přiřadit i IP adresu končící číslem 0 některému z hostů. Například adresa 192.168.5.0 s maskou /22 tedy není adresa sítě, ale adresa, kterou lze přiřadit konkrétnímu zařízení v dané síti. Tato situace může nastat jak u privátních, tak u veřejných IP rozsahů.



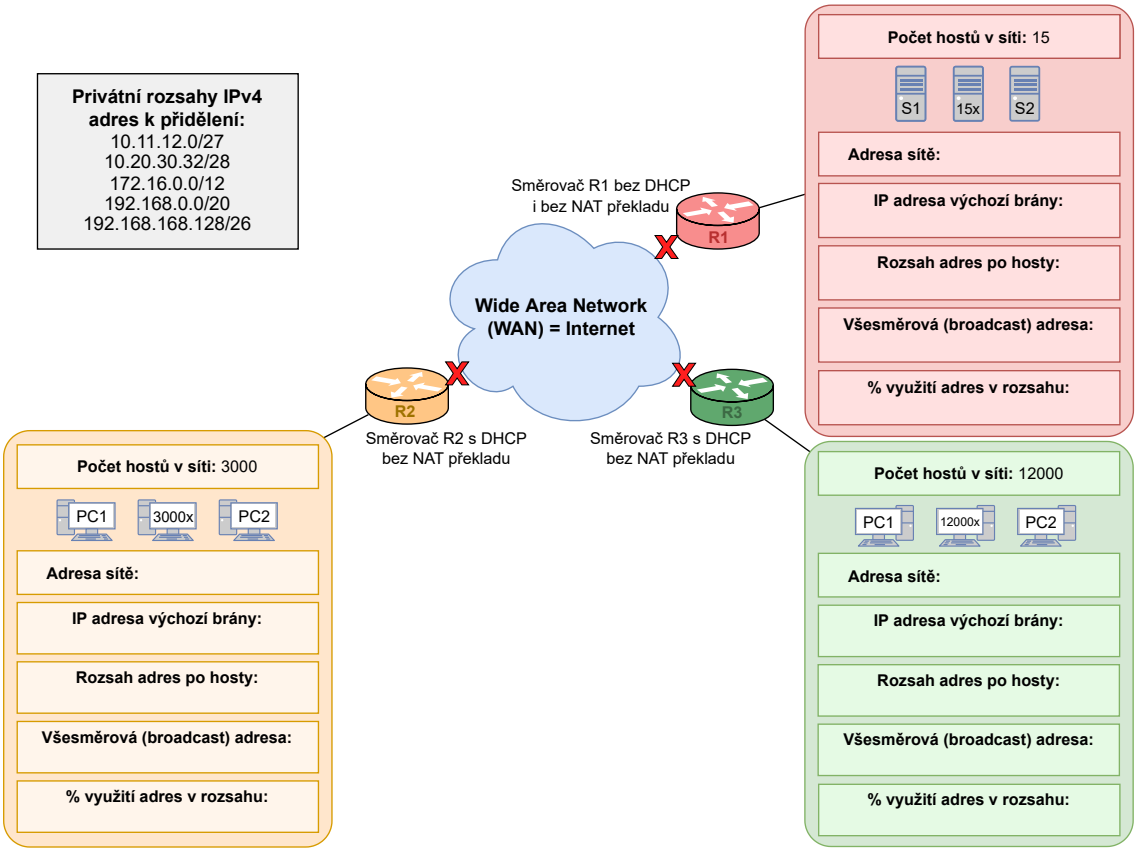
Obr. D.2.5: Vyplněná šablona s řešením ukázkového příkladu s privátními IP adresami.

U sítí s privátními adresami, podobně jako v předešlé ukázce, se směrování běžně neprovádí. Není totiž možné komunikovat s dalšími sítěmi z toho důvodu,

že privátní adresy jsou směrem z lokální sítě nesměrovatelné, respektive není možné zajistit jejich globální unikátnost. Pro vzájemnou komunikaci mezi sítěmi by muselo dojít k nahrazení privátních adres veřejnými adresami nebo by muselo dojít k NAT překladu privátních adres na veřejnou adresu směrovače, obdobně jako to bude provedeno v následující kapitole D.2.3.

Samostatná práce s privátními IP adresami:

Dále si přiřazení privátních adres vyzkoušíte na následujícím příkladu, který je zobrazen v šabloně na obr. D.2.6. Podobně jako u příkladu s veřejnými adresami přiřaďte každé síti některý z IPv4 rozsahů v rámci úkolu č. (4).



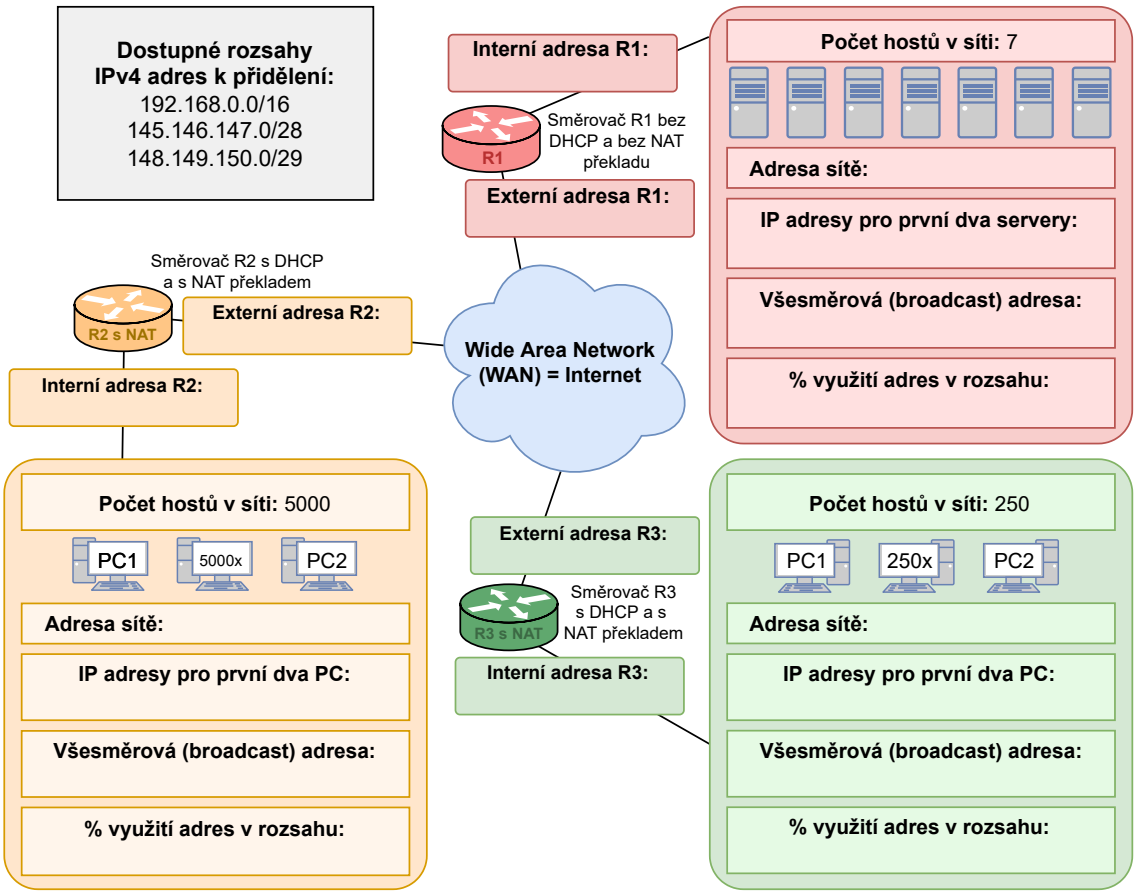
Obr. D.2.6: Šablona pro úkol č. (4) obsahující schéma sítí, pro které budeme hledat správný a co nejefektivnější rozsah privátních adres.

Úkoly:

(4) Přiřaďte rozsahy privátních IP adres a doplňte další údaje k jednotlivým sítím do šablony na obr. D.2.6.

D.2.3 Kombinace veřejných a privátních IPv4 adres

V této části budeme pracovat jak s veřejnými, tak s privátními adresami. Toto řešení je v dnešní době nejvyužívanější v klasických sítích vzhledem k omezenému počtu veřejných IPv4 adres. Některé sítě využívají veřejné adresy z přiděleného rozsahu i pro koncové stanice, jako například velká část sítě VUT. Naopak v domácích sítích narazíme převážně na IP adresy z privátního rozsahu a router nám celý tento rozsah překládá a „skrývá“ za jedinou veřejnou adresu. Pokud má i ISP (Internet service provider – Poskytovatel internetového připojení) nedostatek volných veřejných adres, tak může i on na této úrovni přistoupit k NAT překladu adres a využití privátních adres. Může tak dojít k několikanásobnému NAT překladu IP adres v případě, že komunikujeme z koncové stanice v LAN směrem do prostředí internetu (WAN).



Obr. D.2.7: Šablona obsahující schéma sítí, pro které budeme hledat správnou kombinaci rozsahů veřejných a privátních IP adres.

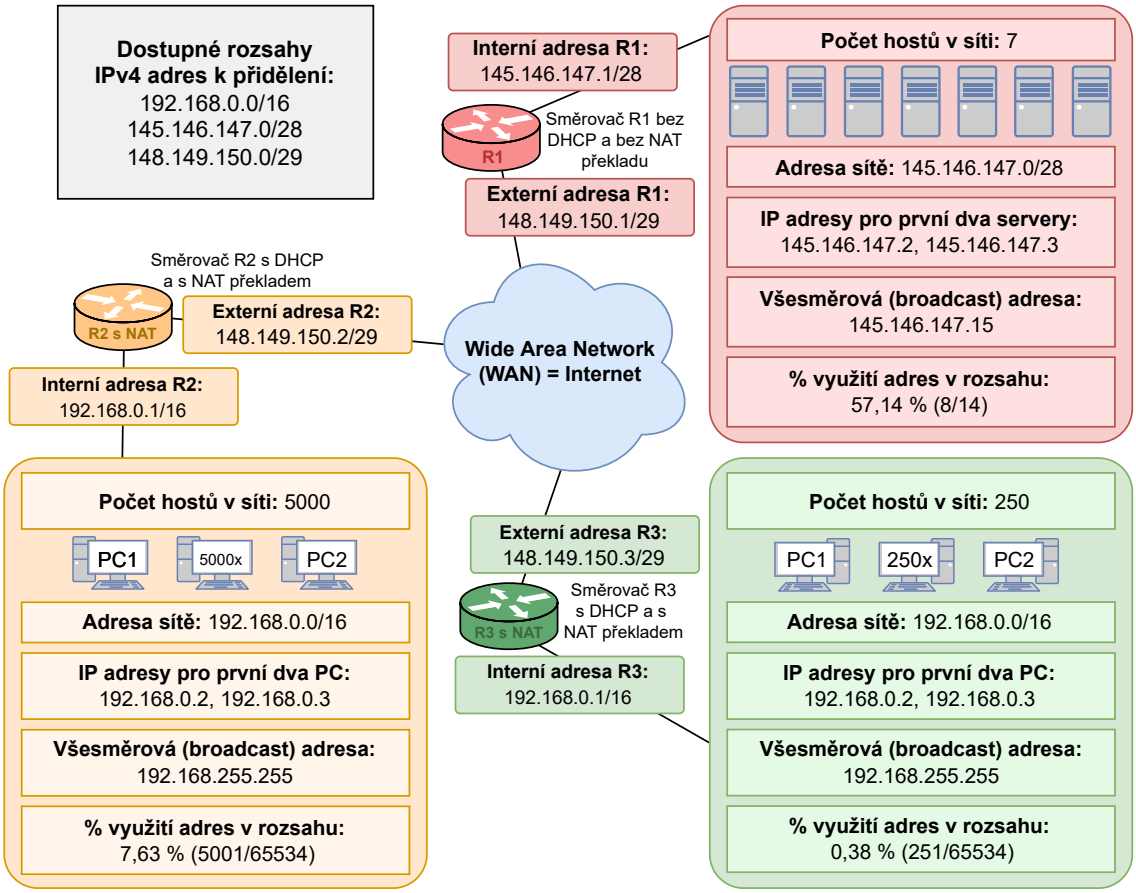
Nyní budeme pracovat s ukázkovým případem, který je zobrazen na obr. D.2.7. Vidíme zde modrý oblak, který představuje prostředí veřejného internetu. Také zde opět máme dostupné rozsahy IPv4 adres a to jak veřejných, tak privátních.

Dále jsou ve schématu zapojeny směrovače. U některých plánujeme využít NAT překlad, u některých ne, jak je uvedeno v obrázku. Takže je potřeba zvážit, který rozsah se přiřadí které síti. Směrovačům se zapnutým NAT překladem budeme chtít přiřadit privátní rozsah pro koncová zařízení v lokální síti a také adresu z tohoto rozsahu, jako privátní (interní) adresu směrovače a také veřejnou adresu z některého z rozsahů. Pro jednoduchost budeme předpokládat, že externí (veřejné) adresy všech tří směrovačů jsou v rámci jednoho IPv4 subnetu. Vnitřním sítím za směrovači bez NATu musíme poskytnout rozsahy s veřejnými adresami, kde každé koncové zařízení bude mít vlastní veřejnou adresu, stejně jako interní rozhraní směrovače. Veřejné adresy musíme také poskytnout vnějším rozhraním směrem do internetu na všech směrovačích. Jak již bylo uvedeno, tyto vnější rozhraní budou na všech směrovačích disponovat IP adresou z jednoho veřejného rozsahu.[1] Těmito postupy se zajistí to, že každé koncové zařízení bude mít možnost komunikovat v rámci internetu.

Ze tří dostupných rozsahů `145.146.147.0/28`, `148.149.150.0/29` a `192.168.0.0/16` spadají první dva do veřejných rozsahů a poslední rozsah je privátní. Červená síť potřebuje adresu přiřadit sedmi koncovým zařízením. Zároveň směrovač R1 nepoužívá NAT překlad, protože koncová zařízení připojená v jeho interní síti jsou servery u nichž není vhodné využívat tento překlad adres. Všechna zařízení v síti tak potřebují veřejnou adresu. Proto pro tuto síť je možné vybrat pouze rozsah `145.146.0.0/28`, který dokáže přiřadit až třináct IP adres. Směrovače R2 a R3 v oranžové a zelené síti podporují NAT překlad adres a tudíž zařízením v těchto sítích můžeme přiřadit IP adresy z rozsahu `192.168.0.0/16`. V obou sítích se mohou přiřadit libovolné (i shodné) adresy z tohoto rozsahu, protože nejsou směrovatelné za směrovač lokální sítě a musí tak být na příslušném směrovači přeloženy pomocí NAT.

Aby tedy koncová zařízení v těchto sítích mohla komunikovat s okolními sítěmi, tak je potřeba směrovačům přiřadit také veřejnou (externí) adresu. Pro to využijeme poslední zbývající rozsah veřejných adres `148.149.150.0/29`. Externí adresa směrovače R1 tak bude například `148.149.150.1/29`. Privátní adresy v oranžové síti budou na směrovači R2 překládány na veřejnou adresu `148.149.150.2/29` a to stejné bude platit pro zelenou síť a směrovač R3 s adresou `148.149.150.3/29`. Řešení tohoto ukázkového příkladu je zobrazeno na obr. D.2.8. V rámci tohoto příkladu vidíme, že u sítí za směrovači R2 a R3 byly využity stejné rozsahy privátních adres. Vzhledem k tomu, že je mezi těmito sítěmi 2x překlad NAT a subnet s veřejnými adresami, tak to ničemu nevadí. Tyto privátní sítě na sebe přímo nevidí a nejedná se o kolizi adres.

[1] Externí adresy směrovačů jsme z důvodu jednoduchosti u předchozích příkladů neřešili.

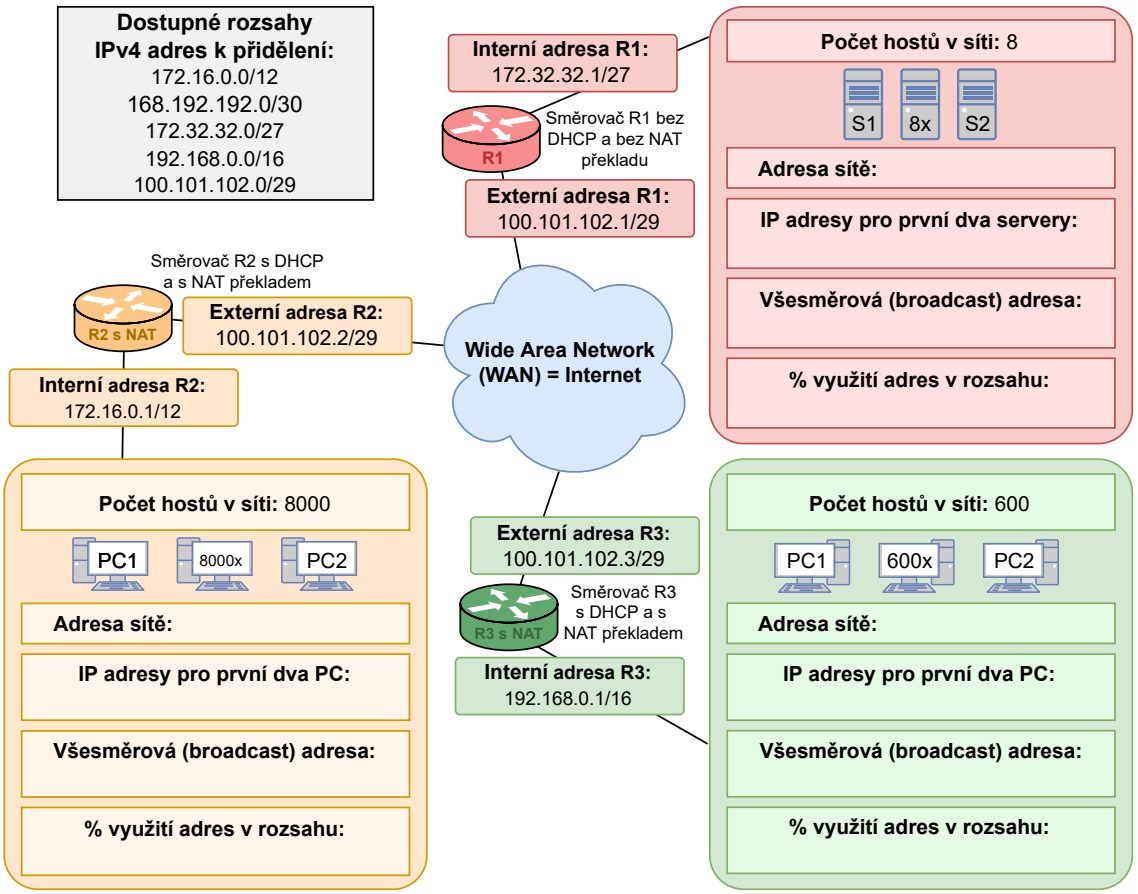


Obr. D.2.8: Vyplněná šablona s řešením ukázkového příkladu s kombinovanými (privátními i veřejnými) IP adresami.

Oproti předchozím příkladům s výhradně privátními nebo výhradně veřejnými rozsahy je tedy potřeba přemýšlet, ve kterých sítích je možné využít určitý typ adres. K rozhodnutí přispěje hlavně to, zda u daného směrovače je vhodné použít NAT překlad a také informace o tom, jaké aplikace a protokoly chceme využívat na zařízeních v lokální síti při komunikaci přes internet.

Samostatná práce s kombinací veřejných a privátních IP adres:

Nyní si sami vyzkoušíte přiřadit kombinaci privátních i veřejných IPv4 adres na následujícím příkladě, který je zobrazen v šabloně na obr. D.2.9. Podobně jako v předcházejícím ukázkovém příkladě s kombinovanými adresami přiřaďte každé síti některý z IPv4 rozsahů v rámci úkolu č. (5) tak, aby všechna zařízení mohla komunikovat i s ostatními sítěmi.



Obr. D.2.9: Šablona pro úkol č. (5) obsahující schéma sítí, pro které budeme hledat správný a co nejefektivnější rozsah při kombinaci veřejných a privátních adres.

Úkoly:

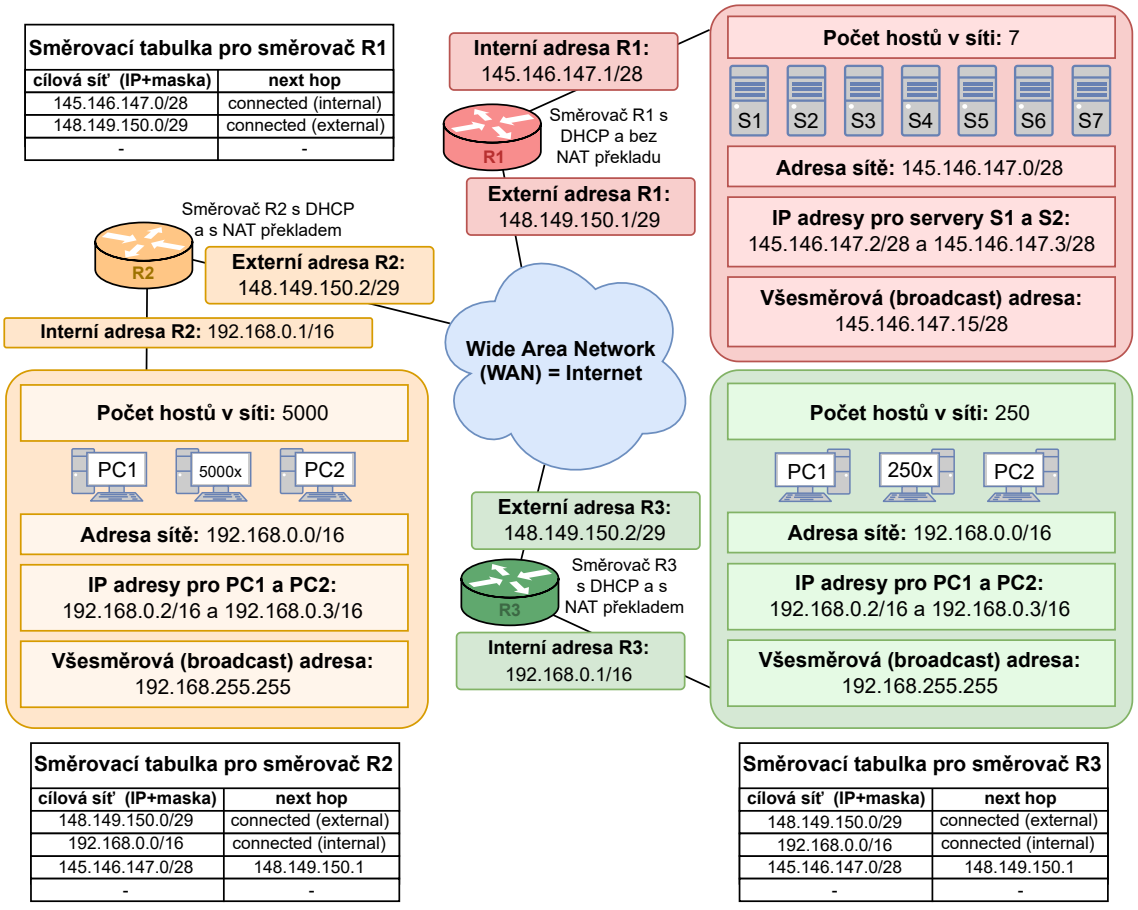
(5) Přiřaďte rozsahy veřejných a privátních IP adres a doplňte další údaje k jednotlivým sítím do šablony na obr. D.2.9.

(6) Jaká bude zdrojová a cílová adresa paketu, který zachytíme na vnějším rozhraní (veřejná adresa) směrem do internetu na směrovači R2? Tento paket byl vytvořen na prvním PC v rámci interní (žluté) sítě a chceme ho doručit prvnímu serveru v červené síti za směrovačem R1.

(7) Jaká bude zdrojová a cílová adresa paketu, který zachytíme na vnitřním rozhraní (privátní adresa) směrovače R2? Tento paket byl vytvořen na prvním PC v rámci interní (žluté) sítě a chceme ho doručit prvnímu PC v zelené síti za směrovačem R3.

D.2.4 Směrovací tabulky pro směrovače s IPv4

Pro správné fungování sítě je potřeba zajistit všem hostům IP adresy z přiděleného rozsahu a je také potřeba, aby každý směrovač disponoval vlastní směrovací tabulkou, která směrovači řekne, přes jaký další skok a pomocí kterého síťového rozhraní má směrovač odeslat paket směrem k cílové adrese. Na ukázkovém příkladu na obr. D.2.10 jsou zobrazeny zjednodušené směrovací tabulky pro směrovače R1, R2 a R3. Tyto tabulky se skládají ze sloupců `cílová síť` a `next hop`. V reálné směrovací tabulce by bylo informací více, např. sloupec síťové rozhraní a metrika. Označení síťového rozhraní by určovalo, které rozhraní směrovače má být využito pro doručení paketu na next hop adresu. Hodnota metriky by se mohla skládat z více parametrů, jako je například počet přeskoků mezi sítěmi, časové zpoždění linek nebo jejich aktuální vytížení. Schéma sítě a přidělené rozsahy vychází z ukázkového příkladu na obr. D.2.8.



Obr. D.2.10: Ukázkový příklad zobrazující směrovací tabulky pro směrovače R1, R2 a R3.

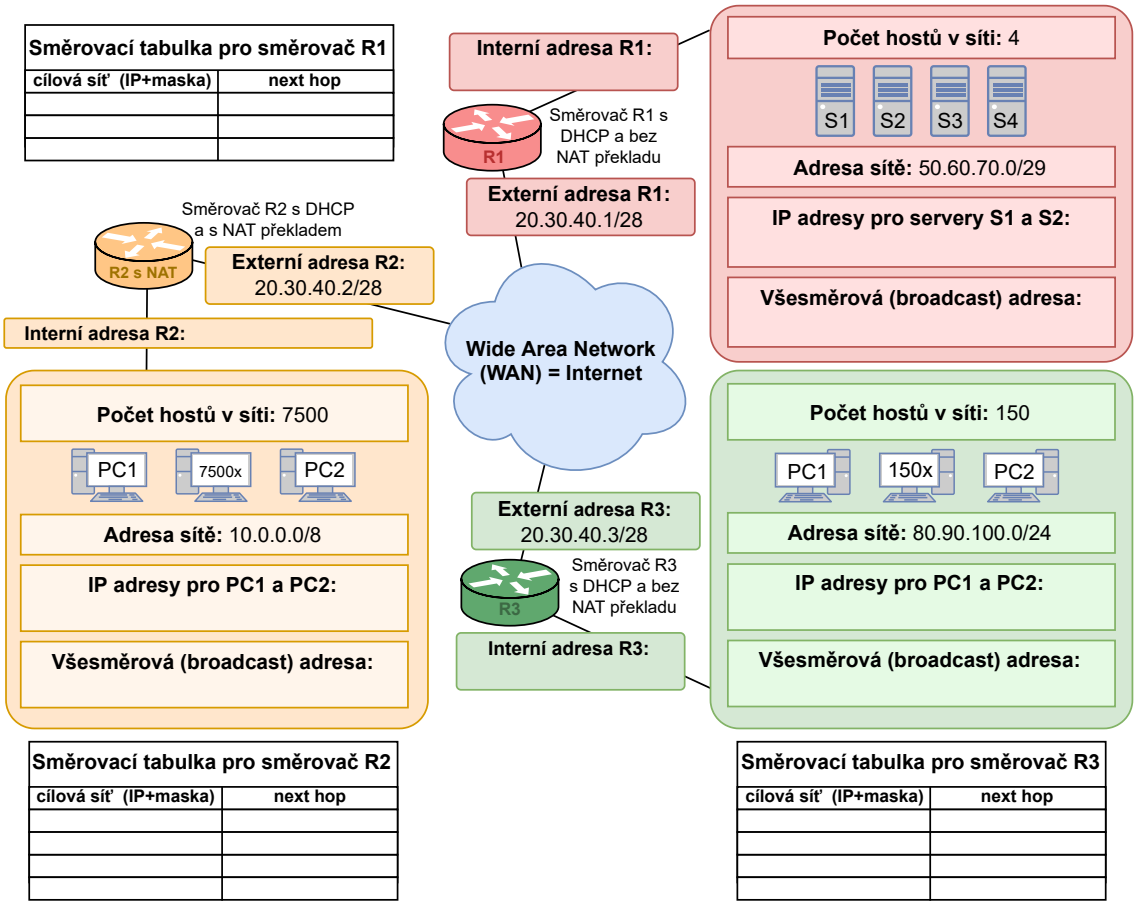
Pro jednoduchost je samotné schéma zapojení tří směrovačů realizováno tak, že jsou tyto směrovače umístěny v rámci jedné sítě s jedním IP rozsahem (`148.149.150.0/29`). Za směrovači se nachází lokální sítě a to se zařízeními s privátními, ale i veřejnými IP adresami.

Jak již bylo zmíněno výše, tak směrovací tabulky obsahují v tomto příkladě dva sloupce. Adresa ve sloupci `cílová síť` obsahuje adresu sítě včetně masky sítě, pro kterou daný záznam na daném řádku platí. Jako první tedy ve směrovací tabulce pro směrovač R1 uvedeme adresu lokální sítě, která se nachází za směrovačem R1. Jde tedy o záznam pro síť `145.146.147.0/28`. Jako next hop adresu pro tuto lokální síť uvedeme pouze textový záznam „connected (internal)“. Záznam „connected“ znamená že zmíněná síť, je přímo spojená se směrovačem R1 a není tak již potřeba směrovat provoz do dalších sítí přes další směrovače. Doplněk „internal“ naznačuje, že síť pro kterou záznam platí, se nachází v lokální síti za směrovačem. Druhý záznam, který může být v daném schématu důležitý, je záznam pro síť `148.149.150.0/29`. Tato síť je opět přímo spojená se směrovačem R1, prostřednictvím vnější IP adresy směrovače `148.149.150.1`. A proto pro next hop záznam opět použijeme výraz „connected“, nyní ale s doplňkem „external“, protože tato síť směřuje směrem k WAN síti.

Obdobně jsou vytvořeny záznamy pro směrovače R2 a R3. Jedinou změnou je, že z těchto směrovačů může dojít i ke směrování paketu do sítě `145.146.147.0/28`, protože tato síť disponuje veřejně směrovatelnými adresami na rozdíl od lokálních sítí za směrovači R2 a R3. Proto je například ve směrovací tabulce směrovače R2 uveden záznam pro zmiňovanou cílovou síť `145.146.147.0/28`. Zde je již next hop záznam uveden ve formě IP adresy, protože abychom se ze směrovače R2 dostali do sítě `145.146.147.0/28`, tak je potřeba přeskočit na směrovač R1 a to konkrétně přes jeho externí IP adresu `148.149.150.1`. Tuto adresu tedy uvedeme jako třetí záznam směrovací tabulky a obdobně přidáme záznam do tabulky pro směrovač R3.

Samostatná práce se směrovacími tabulkami pro směrovače s IPv4:

Nyní si sami vyzkoušíte doplnit jednotlivé záznamy do směrovacích tabulek a to pro schéma, které je uvedeno na obr. D.2.11. Dále popíšete trasu a změny adres konkrétního paketu a také zkusíte odhalit chyby v předdefinované ukázce na obr. H.3.1.



Obr. D.2.11: Šablona pro úkol č. (8) a (7) obsahující schéma sítí se směrovači R1, R2 a R3, pro které budeme doplňovat záznamy do jejich směrovacích tabulek.

Úkoly:

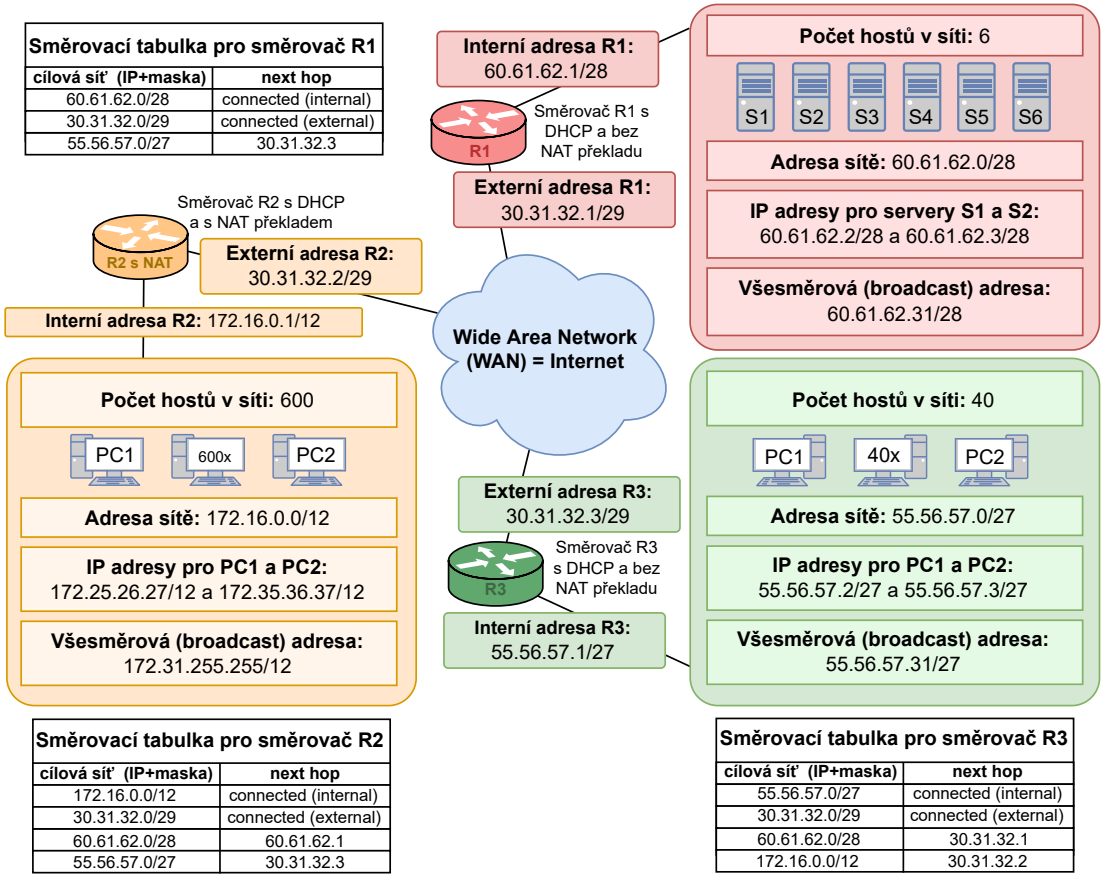
(8) Do šablony na obr. D.2.11 doplňte chybějící IP adresy pro jednotlivá zařízení v červené a zelené síti. Pozor: V tomto případě je kromě směrovače R1 NAT vypnut i na směrovači R3 a v lokální síti jsou použity veřejné adresy.

(9) Do šablony na obr. D.2.11 doplňte záznamy do směrovacích tabulek pro směrovače R1, R2 a R3.

(10) Mějme paket vytvořený koncovým zařízením PC2 v interní (žluté) síti za směrovačem R2. Tento paket chceme doručit na server S2 v červené síti. Popište, jak se bude měnit zdrojová a cílová IP adresa paketu a kudy bude paket při cestě k cílové stanici procházet.

(11) Při jaké minimální hodnotě TTL u paketu z předchozího úkolu (10) ještě dojde k jeho doručení a při které hodnotě TTL dojde k zahození paketu na směrovači R1?

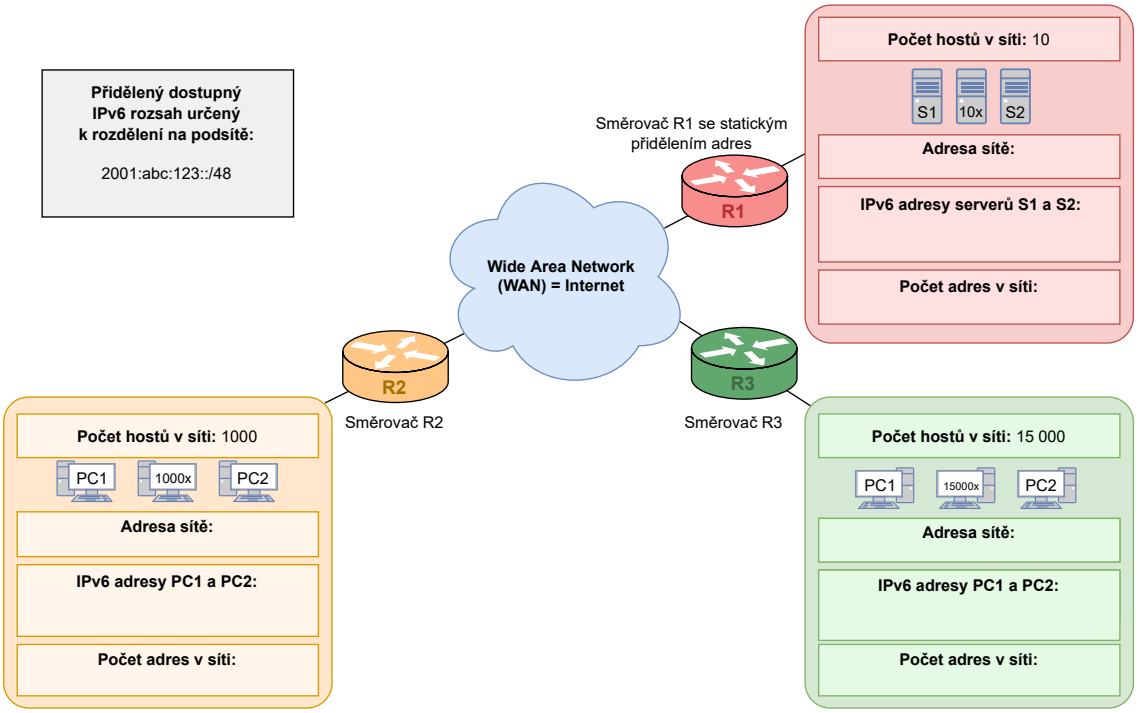
(12) V šabloně na obr. H.3.1 vyhledejte a opravte 5 chyb v konfiguraci síťových adres (IP adresy, rozsahy, masky) a ve směrovacích tabulkách.



Obr. D.2.12: Šablona pro úkol č. (12) obsahující chyby v konfiguraci síťových adres a také ve směrovacích tabulkách.

D.2.5 Plánování a přidělování IPv6 adresního prostoru

Následující část se bude věnovat problematice přiřazování IPv6 adres. IPv6 přináší 128 bitové IP adresy, které jsou většinou zapisovány v hexadecimálním tvaru. Samotná IP adresa se rozděluje na 3 hlavní části. První část o délce 48 bitů je označována jako globální prefix. Tato část adresy bývá přidělována lokálním registrátorem. Následuje 16 bitová část, která slouží koncovému uživateli pro vytváření podsítí. Teoreticky si taky může uživatel vytvořit 2^{16} podsítí a v každé podsíti následně přiřadit 2^{64} IPv6 adres, protože zbývající 64 bitová část adresy slouží právě pro přiřazení koncovým zařízením. U IPv6 protokolu se tak obejdeme bez překladu adres NAT.

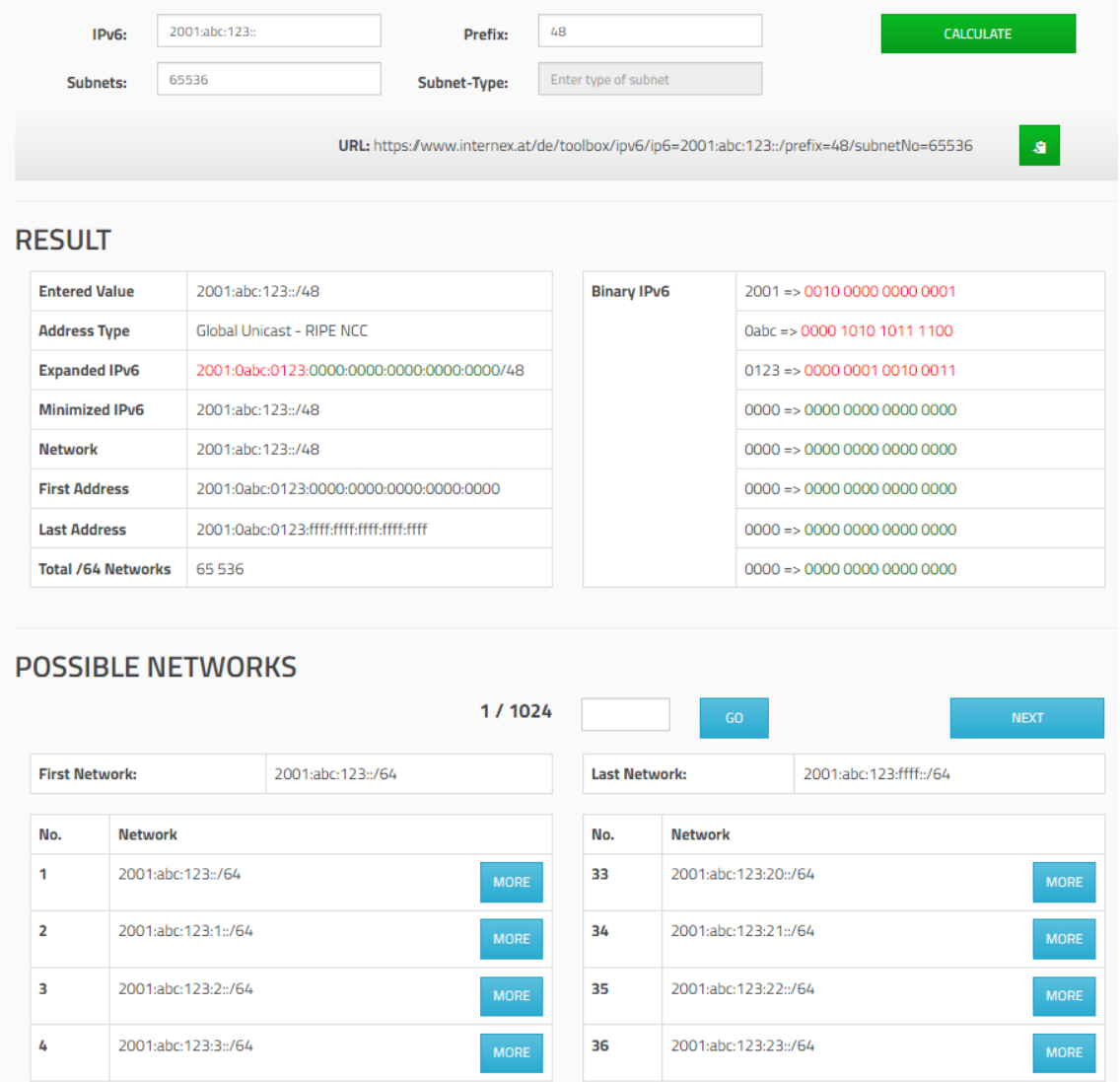


Obr. D.2.13: Šablona obsahující schéma sítí, pro které budeme hledat co nejvhodnější rozsah IPv6 adres.

Budeme tedy přidělovat rozsahy podobně jako v příkladě s veřejnými adresami u IPv4. Na obr. D.2.13 však vidíme pouze jeden dostupný IPv6 rozsah. Adresy z tohoto rozsahu chceme přiřadit třem sítím za jednotlivými směrovači R1, R2 a R3. Musíme tedy z rozsahu s maskou /48 vytvořit 3 podsítě, které budou disponovat maskou /64. Pro lepší pochopení a vizualizaci využijeme i online kalkulátor Internex IPv6 Subnet Calculator.[2] Vezmeme si tedy rozsah `2001:0abc:0123::/48` a vložíme ho do online nástroje Internex pro výpočet parametrů IPv6 podsítí, podobně jako na obr. D.2.14. Do textového pole „Subnets“ vložíme hodnotu 65536, aby vytvořené podsítě měly masku /64.

Ve výsledcích vidíme, že se jedná o globální unicastovou adresu a nejde tak o žádnou speciální lokální adresu, která by nebyla směrovatelná v rámci internetu. Dále zde vidíme barevně rozlišené části, kdy červená část označuje prefix (maska sítě), který je pro tento rozsah /48. Zbývající zelená část patří rozsahu pro podsítě (/16) a také IP adresám pro koncová zařízení (/64). Dle masky sítě můžeme tedy odvodit, že v tomto přiděleném rozsahu je možné vytvořit až 65536 podsítí. Část pro koncová zařízení tedy obsahuje 64 bitů z čehož nám vyplývá, že v každé podsíti síti je možné přiřadit 2^{64} adres. V pravé části můžeme také vidět přehledný převod

[2] https://www.internex.at/de/toolbox/ipv6/



Obr. D.2.14: Online nástroj Internex pro výpočet parametrů dostupné IPv6 adresy z ukázkového příkladu.

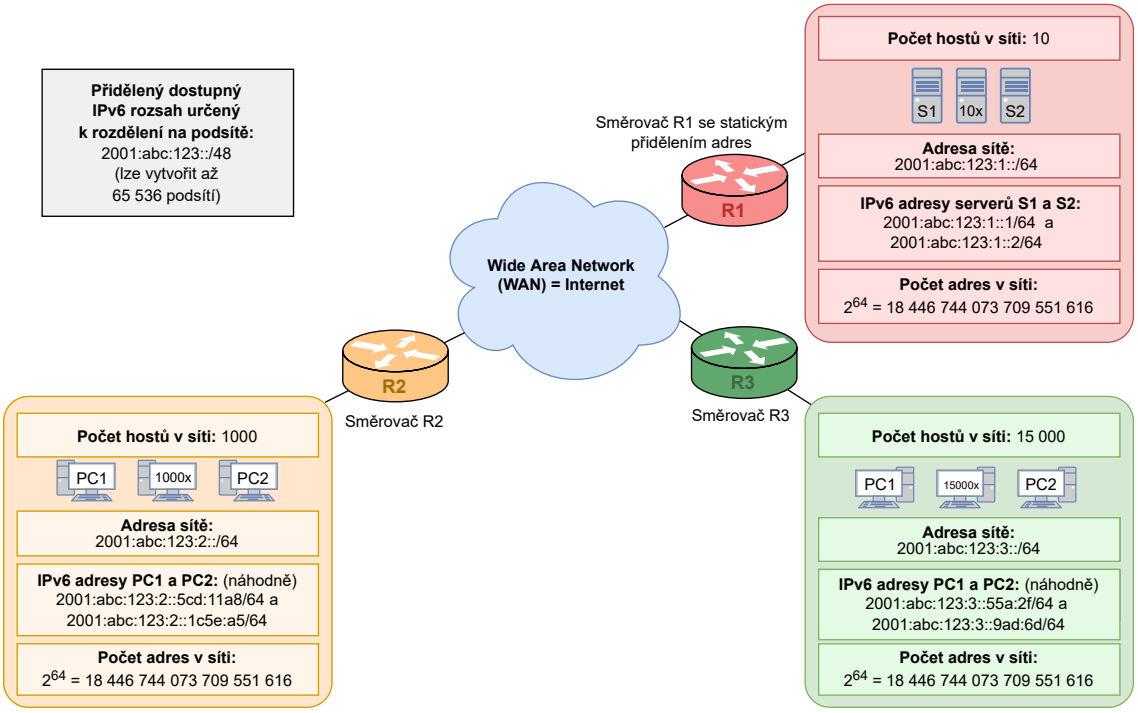
IPv6 adresy v hexadecimálním zápisu na zápis binární, který známe z IPv4. Pokud do textového pole v horní části vyplníme u položky „Subnets“ hodnotu 65536, tak nám kalkulačka sama navrhne vhodné rozsahy pro podsítě. V našem ukázkovém příkladě nám z tohoto počtu postačí 3 adresy pro 3 sítě..

V přiděleném rozsahu `2001:abc:123::/48` tedy chceme vytvořit 3 podsítě. Pro tyto podsítě máme vyčleněno 16 bitů. Těchto 16 bitů je reprezentováno čtyřmi hexadecimálními znaky, které jsou v následujícím rozsahu nahrazeny hodnotami `WXYZ`: `2001:abc:123:WXYZ::/64`. Konkrétní rozsah pro červenou podsíť tedy může vypadat takto: `2001:abc:123:1::/64`.[3] Povšimněte si, že u této IPv6 adresy jsme

[3]Klidně bychom mohli začít číslovat podsítě od nuly, tj. od subnetu 2001:abc:123:0::/64, avšak

využili možnost zkrátit zápis o 21 znaků (4 skupiny nul a také nuly na začátcích jednotlivých hexadecimálních skupin). Počet možných IPv6 adres v této červené síti je 2^{64}.

Oranžová síť požaduje 1000 IPv6 adres, ale my můžeme přidělit rozsah /64, který je pro koncové sítě u IPv6 určen. A tak můžeme přiřadit navazující rozsah na ten, který byl přidělen u červené podsítě a to `2001:abc:123:2::/64`. Zbývající zelená síť požaduje připojit až 15 000 koncových zařízení a pokud se budeme držet návaznosti, tak jí přiřadíme rozsah `2001:abc:123:3::/64`. Do šablony ještě doplníme IPv6 adresy pro první a druhé zařízení v každé z podsítí. U červené sítě využijeme statické přidělení adres, aby měl každý server pevně definovanou IPv6 adresu. První server S1 tedy může využívat adresu `2001:abc:123:1::1/64` a druhý server S2 adresu `2001:abc:123:1::2/64`. U ostatních sítí se adresy přidělí automaticky tak, že se náhodně vyberou IPv6 adresy z rozsahu. Řešení ukázkového příkladu je ukázáno na obr. D.2.15.



Obr. D.2.15: Vyplněná šablona s řešením ukázkového příkladu s IPv6 adresami.

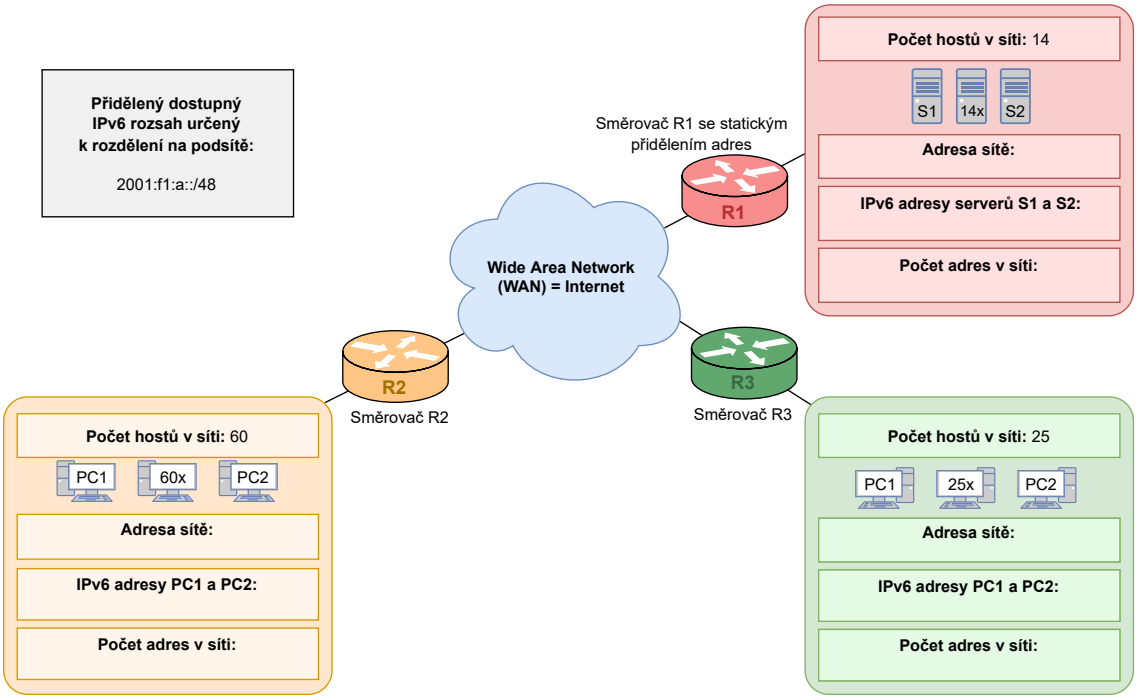
Při využití IPv6 tedy dostaneme rozsah, ve kterém můžeme vytvořit velké množství podsítí a každá z těchto podsítí disponuje velkým počtem IPv6 adres. Každé koncové zařízení v síti tak může mít přidělenou vlastní veřejnou IPv6 adresu

z důvodu názornosti používáme subnet 1 na prvním routeru atd.

a nemusí se využívat NAT překlad, na rozdíl od IPv4 adres a rozsahů, jak bylo ukázáno v předchozích ukázkových příkladech.

Samostatná práce obsahující plánování a přidělování IPv6 adresního prostoru:

V následující části si sami vyzkoušíte rozdělit přidělený IPv6 rozsah a také následně přidělíte jednotlivé rozsahy podsítím, podobně jako v ukázkovém příkladě.



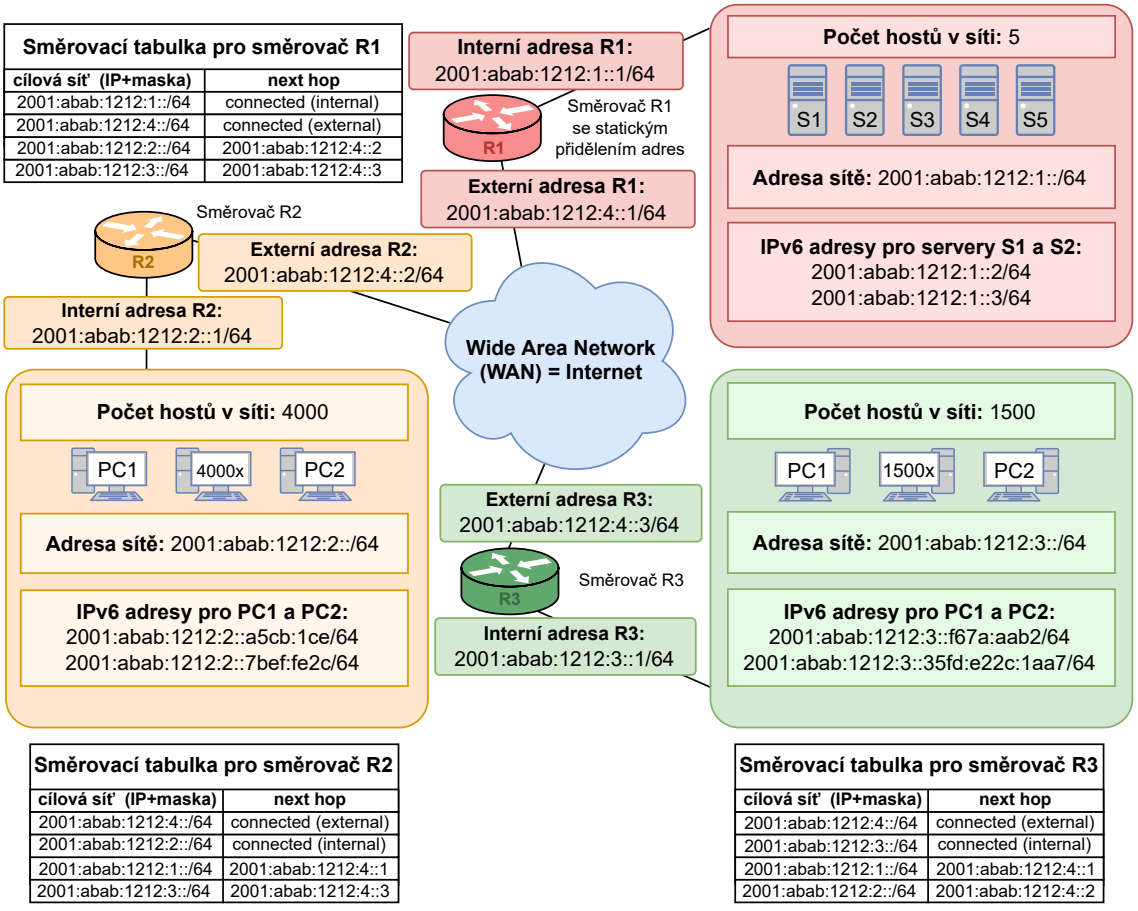
Obr. D.2.16: Šablona pro úkol č. (13) obsahující schéma sítí, pro které budeme hledat správný rozsah globálních IPv6 adres.

Úkoly:
(13) Rozdělte dostupný IPv6 rozsah a vytvořené rozsahy pro podsítě následně přiřaďte jednotlivým sítím do šablony na obr. D.2.16.

D.2.6 Směrovací tabulky pro směrovače s IPv6

Stejně jako u IPv4, tak i u IPv6 je pro fungování sítě potřeba zajistit všem zařízením v síti IP adresy z přiděleného rozsahu a je také potřeba, aby každý směrovač disponoval vlastní směrovací tabulkou. Na ukázkovém příkladu na obr. D.2.17 jsou zobrazeny směrovací tabulky pro směrovače R1, R2 a R3. Oproti předcházejícím příkladům nyní stanovíme adresní plán i na síti mezi směrovači R1, R2 a R3. Pro jednoduchost budeme uvažovat, že jsou všechny tyto směrovače na jednom

subnetu, konkrétně 2001:abab:1212:4::/64. Tabulky obsahují stejné položky, jako u IPv4. Opět zde máme záznamy, kdy next hop adresa je vyjádřena pomocí textového záznamu „Connected (internal)“ pro sítě, které jsou přímo spojeny s danými směrovači jako lokální síť a také „Connected (external)“ pro sítě, které jsou se směrovači spojeny při komunikaci směrem do sítě WAN. Jelikož všechny IPv6 adresy z tohoto příkladu jsou veřejně směrovatelné, tak je potřeba do směrovacích tabulek uvést vždy i interní sítě za zbývajícími směrovači ze schématu. Pro tyto záznamy bude next hop adresa konkrétní IP adresa směrovače. Například pro směrovací tabulku směrovače R1 bude uveden záznam cílové sítě za směrovačem R2 v podobě IPv6 rozsahu `2001:abab:1212:2::/64`. Next hop adresa bude v tomto případě externí adresa směrovače R2, konkrétně `2001:abab:1212:4::1`. I ostatní záznamy jsou obdobné těm, které byly vysvětleny u směrovacích tabulek pro IPv4 sítě. Kompletně vyplněné směrovací tabulky ukázkového příkladu s IPv6 adresami můžeme vidět na obr. D.2.17.

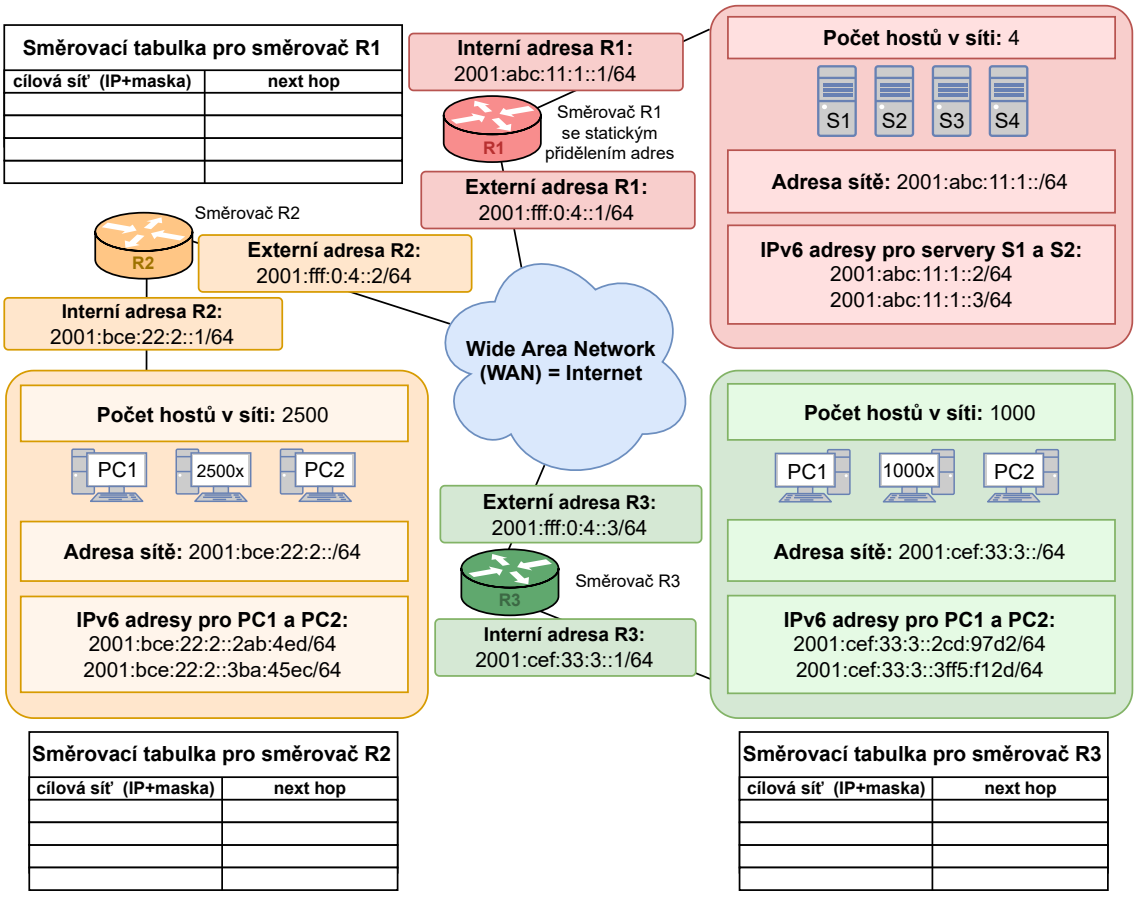


Obr. D.2.17: Ukázkový příklad zobrazující směrovací tabulky pro směrovače R1, R2 a R3 s IPv6.

Díky dostatku IPv6 adres není u IPv6 nutné přistupovat k využívání privátních IP adres v lokálních sítích a není tedy potřeba využívat ani služby NAT překladu.

Samostatná práce se směrovacími tabulkami pro směrovače s IPv6:

Dále si sami vyzkoušíte doplnit jednotlivé záznamy do směrovacích tabulek pro schéma se sítěmi s IPv6 adresami, které je uvedeno na obr. D.2.18. Povšimněte si, že nyní se rozsahy jednotlivých sítí liší i v prvních 48 bitech adresy (tzv. globální směrovací prefix). Nakonec se také pokusíte odhalit chyby v předdefinované ukázce na obr. D.2.19.



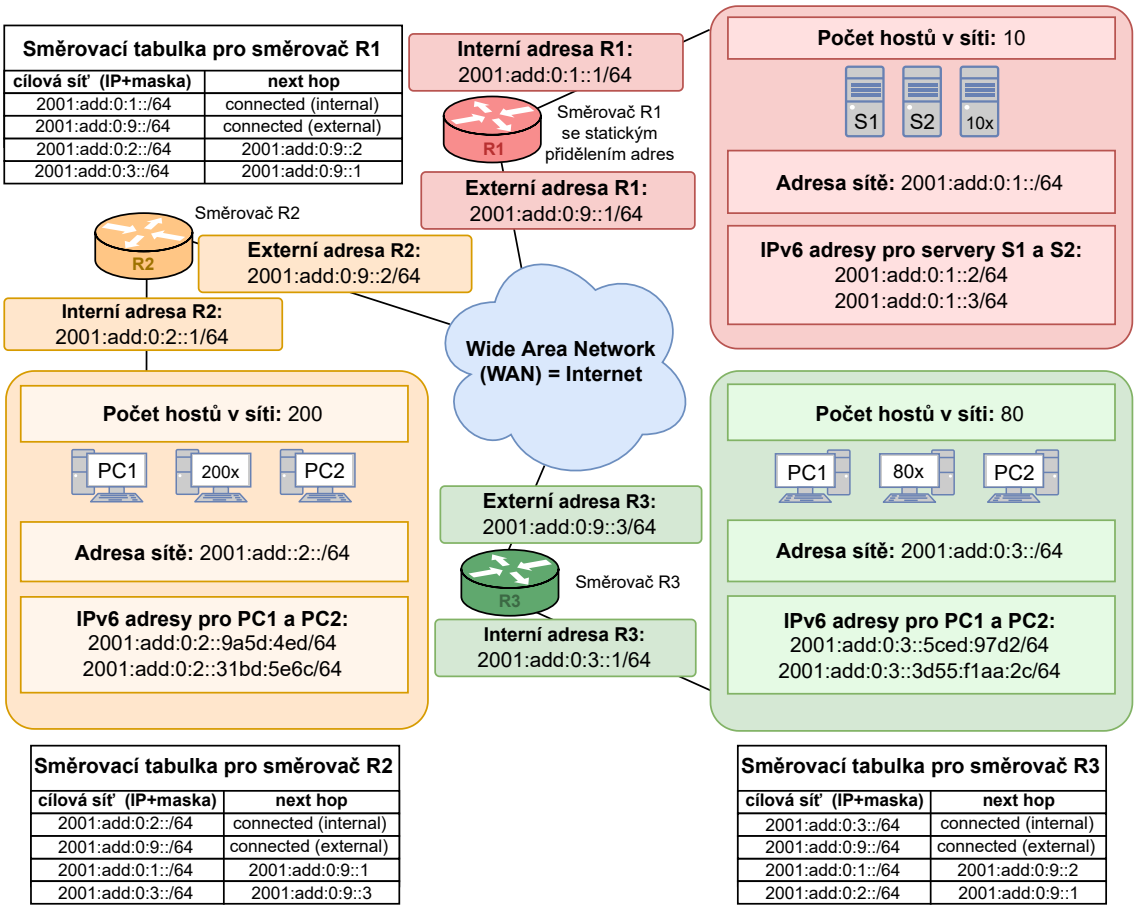
Obr. D.2.18: Šablona pro úkol č. (14) obsahující schéma sítí se směrovači R1, R2 a R3, pro které budeme doplňovat IPv6 záznamy do jejich směrovacích tabulek.

Úkoly:

(14) Do šablony na obr. D.2.18 doplňte záznamy v podobě IPv6 sítí a adres do směrovacích tabulek pro směrovače R1, R2 a R3.

(15) Popište jednotlivé kroky směrování paketu, který byl vytvořen na PC1 ve žluté síti. Cílová destinace paketu je server S2 v červené síti.

(16) V šabloně na obr. D.2.19 vyhledejte a opravte 3 chyby v konfiguraci síťových adres (IP adresy a rozsahy) a ve směrovacích tabulkách.



Obr. D.2.19: Šablona pro úkol č. (16) obsahující chyby v konfiguraci síťových adres a také ve směrovacích tabulkách.

E Řešení prvního simulačního scénáře

E.1 Spuštění systému a kontrola parametrů

(1) Zkontrolujte konektivitu připojení k Internetu pomocí příkazu ping na libovolný webový server v příkazové řádce.

(2) Pomocí příkazu ipconfig z virtuálního OS zkontrolujte IP adresu síťového adaptéru. O jaký typ adresy se jedná a do kterého rozsahu daná adresa spadá?
O: (192.168.254.128) Je to privátní adresa z rozsahu pro použití při NATu ve třídě adres C. Přidělená IP adresa by měla odpovídat adrese v rozsahu 192.168.0.0 – 192.168.255.255.

E.2 Aplikace DNS Klient a její nastavení

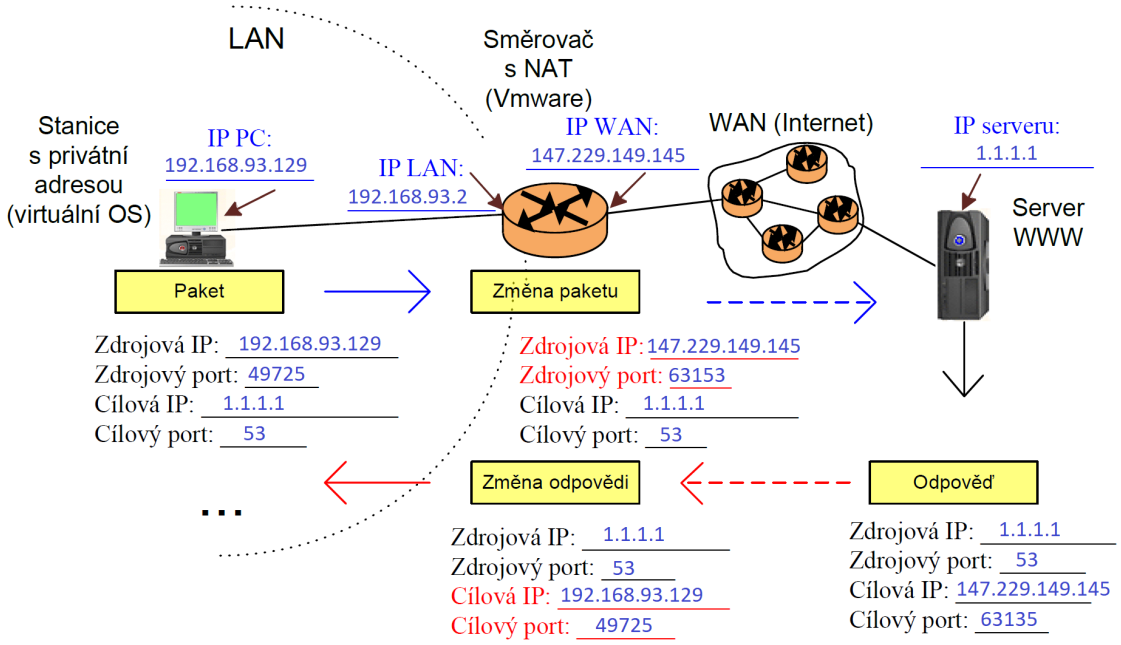
Tato kapitola neobsahuje kontrolní otázky.

E.3 Zachycení UDP paketů s překladem NAT

(3) Překládají se tedy IP adresy a porty, což vede na změnu i u příslušného kontrolního součtu. Které parametry naopak zůstávají stejné i po překladu NAT? O kterou vrstvu se jedná?
O: Beze změny zůstává celé DNS záhlaví a hlavně tedy Transaction ID, které je pro každou DNS komunikaci unikátní. Překlad NAT tedy do aplikační vrstvy v případě DNS nezasahuje.

E.4 Zachycení TCP paketů s překladem NAT

(4) Vyplňte zachycené IP adresy a porty do šablony na obr. E.4.1 pro oba směry DNS komunikace v případě použití TCP protokolu. (Směrovači odpovídá hostitelský OS a PC v LAN je v našem případě virtualizovaný OS.)
O: Čísla portů se mohou lišit až na port využitý na straně serveru, který musí odpovídat číslu 53. Stejně tak IP adresy mohou být odlišné, ale IP adresa serveru by měla odpovídat adrese 1.1.1.1.



Obr. E.4.1: Šablona pro úkol č. (4) znázorňující překlad NAT při DNS komunikaci.

E.5 NAT překlad při více DNS požadavcích

Tato kapitola neobsahuje kontrolní otázky.

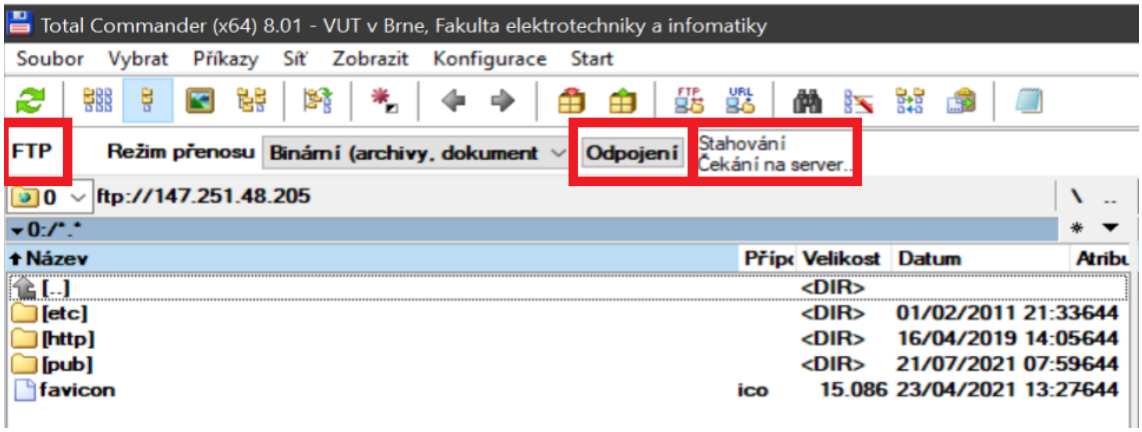
E.6 Zachycení ICMP paketů s překladem NAT

(5) Který parametr využívá protokol ICMP k rozlišení několika současných ICMP komunikací místo portů?
O: K rozlišení několika současných ICMP komunikací je místo portů využito sekvenční číslo.

E.7 Připojení k FTP serveru a analýza FTP komunikace

(6) Jaký číselný kód má FTP zpráva s označením „Transfer complete“? Zprávy o stavu připojení k FTP serveru se zobrazují v horní části programu Total Commander, viz obr. E.7.1.

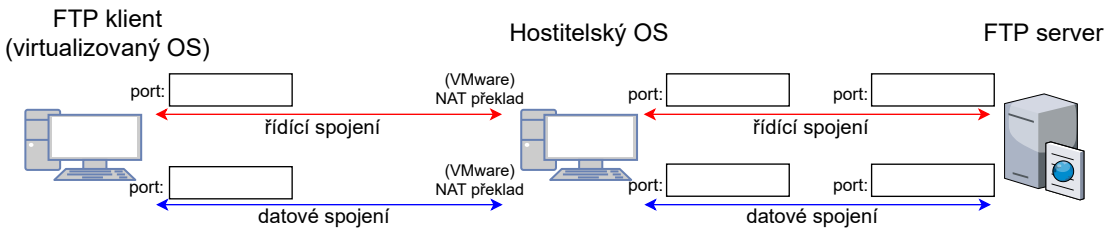
O: Zpráva má číselný kód „226 Transfer complete“.



Obr. E.7.1: Informace o stavu připojení k FTP serveru a možnost odpojení se od FTP serveru v programu Total Commander.

(7) Vyplňte čísla portů ze zachycené FTP komunikace při zvoleném pasivním režimu do šablony na obr. E.7.2.

O: Čísla portů se mohou lišit, až na port č. 21 na straně FTP serveru u řídícího spojení.



Obr. E.7.2: Šablona pro úkol č. (7) znázorňující FTP komunikaci s překladem NAT.

(8) Jaké podrobnosti o souborech jsou přenášeny v paketu označeném jako FTP-DATA. Uveďte alespoň 3 parametry. Nápověda: hledejte v části aplikační vrstvy pojmenované jako Line-based text data.

O: Jsou přenášeny podrobnosti o souborech jako např. datum poslední změny souboru, velikost souborů, mód určující možnosti přístupu k souboru jednotlivým typům uživatelů a samozřejmě také název souborů nebo složek v daném adresáři.

(9) Jak je volena hodnota čísla FTP datového portu v aktivním režimu vzhledem ke zdrojovému portu na straně klienta?

O: Číslo aktivního portu je o 1 vyšší než aktuální hodnota zdrojového portu klienta.

(10) Je možné v zachycené komunikaci vyhledat přihlašovací údaje pro přístup k FTP serveru? Je použití klasického FTP bezpečné?

O: Ano, je možné v komunikaci najít přihlašovací údaje uživatele včetně jeho hesla. Použití klasického FTP tak není z tohoto hlediska bezpečné a je vhodné zvolit některou z bezpečnějších variant (FTPS nebo FTP s SSL/TLS).

E.8 Analýza SCTP paketů při NAT komunikaci

(11) Jaký typ a kód chyby je uveden u ICMP paketu?

O: ICMP paket oznamuje informace o chybě typu 3 (Destination unreachable) a kódu 2 (Protocol unreachable).

(12) Jaký zdrojový a cílový port je uveden u jediného SCTP paketu v předem zachyceném souboru?

O: V zachyceném souboru je uveden zdrojový port 41028 a cílový port 5201.

F Řešení druhého simulačního scénáře

F.1 Základní DNSSEC komunikace a její analýza

(1) Je možné zobrazit podrobnosti u jednotlivých DNS dotazů a odpovědí?
O: Ano, je to možné, protože se využívá klasický DNS protokol, nikoliv protokol DNS over HTTPS.

(2) Jaké hodnoty o digitálním podpisu jsou uvedeny v RRSIG záznamu?
O: V RRSIG záznamu jsou o digitálním podpisu uvedeny informace, jako například algoritmus použitý pro vytvoření a ověření podpisu, datum vytvoření a expirace podpisu a také jméno podepisujícího.

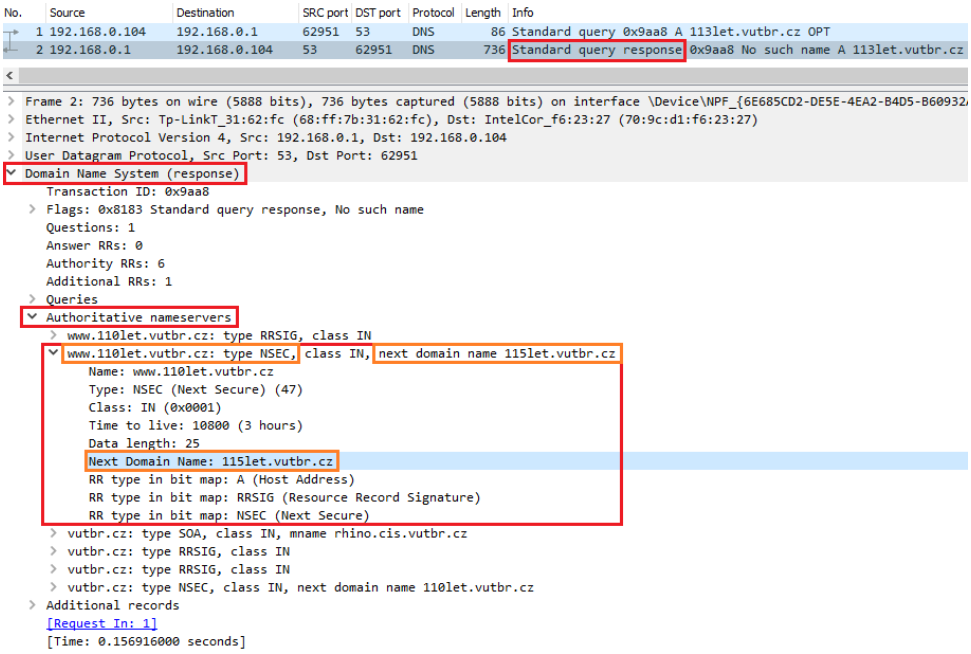
(3) Který konkrétní prvek nejvíce navyšuje velikost DNSSEC komunikace oproti DNS komunikaci? Jakou velikost v bajtech má tento prvek? Nápověda: Hledejte v části zprávy typu RRSIG.
O: Největší část z celkových 166 bajtů, které má část RRSIG zprávy nese prvek, který uchovává hodnotu podpisu (Signature). Má velikost 128 bajtů.

F.2 DNSSEC dotaz na neexistující doménu

(4) Vyzkoušejte v aplikaci Klient DNS dotaz na záznam typu A na doménu 113let.vutbr.cz. Existuje daná doména?
O: Neexistuje, protože v DNS odpovědi je záznam typu NSEC, který nás informuje o tom, že doména vutbr nezná subdoménu 113let.

(5) Jaký název nesou předchozí a následující existující subdomény v DNS odpovědi při dotazu na doménu 113let.vutbr.cz?
O: Předchozí existující doménou je 110let.vutbr.cz a následující existující doménou je 115let.vutbr.cz. Informace se nachází v NSEC záznamu v DNS odpovědi v Domain name system (response) > Authoritative nameservers > www.110letvutbr.cz > Next Domain Name: 115let.vutbr.cz, viz obr. F.2.1.

(6) Lze zobrazit název předchozí a následující existující subdomény v DNS odpovědi při dotazu na doménu domenakteraneexistuje.cz? Pokud jej nelze zobrazit, tak objasněte proč.
O: Nelze, protože doména nejvyššího řádu cz poskytuje informace o názvu předchozí a následující existující subdomény ve formě hashe.



Obr. F.2.1: Odpověď na otázku č. (5) věnující se NSEC záznamu, který zobrazuje předchozí a následující existující doménu.

F.3 DNSSEC odpověď s podvrženým záznamem

(7) Jaké IP adresy byly zjištěny z posledního DNS dotazu na doménové jméno dnssec-failed.org?

O: Byly zjištěny IP adresy 69.252.193.191 a 68.87.109.242.

(8) Proveďte dotaz na další záměrně špatně podepsanou doménu „rhybar.cz“ nejdříve s pomocí DNSSEC (CD=☐,AD=✓,DO=✓) a následně s možností příjmu i pro neautentizovaná data (CD=✓,AD=✓,DO=✓). Jaká mailová doména je uvedena v podrobnostech SOA záznamu?

O: V podrobnostech SOA záznamu je uvedena mailová doména hostmaster.nic.cz. Podrobnosti se nacházejí v DNS odpovědi v: Domain name system (response) > Authoritative nameservers > rhybar.cz: type SOA > Responsible authority's mailbox: hostmaster.nic.cz.

F.4 Dotaz na doménu nepodporující DNSSEC

Tato kapitola neobsahuje kontrolní otázky.

F.5 Základní DNS over HTTPS komunikace

(9) Ve Wiresharku porovnejte a následně popište potřebný počet bajtů pro:
1) klasický DNS dotaz a odpověď na IP adresu Cloudflare serveru
2) DoH dotaz a odpověď na překlad doménového jména vut.cz včetně navazování a ukončování spojení.
Nápověda: Využít můžete například nástroj pro analýzu Statistics > Conversations. Pro jednodušší orientaci zaškrtněte položku v dolní částí okna „Limit to display filter“.
O: Klasický DNS dotaz a odpověď mají velkost zhruba 188 bajtů ve dvou paketech. DNS over HTTPS dotaz a odpověď včetně navázání a ukončování spojení mají celkem 21 paketů o velikosti 6465. Hodnota se může mírně lišit v závislosti na počtu chyb při přenosu atp. DoH komunikace tak potřebuje zhruba 34x více bajtů než klasická DNS komunikace.

(10) Jaká hlavní výhoda a nevýhoda tedy vyplývá z předchozí analýzy?
O: Hlavní nevýhodou je potřebný počet bajtů a paketů, který je potřeba přenést po síti a z toho plynoucí zatížení sítě, vyšší paměťová náročnost na zpracování paketů (šifrování, dešifrování) a možné zpoždění DNS překladu. Výhodou je zajištění bezpečné komunikace mezi klientem a rekurzivním serverem.

(11) Identifikujte slabé místo proběhlé DoH komunikace.
O: Slabým místem je překlad názvu DNS serveru Cloudflare bez DNSSEC pouze přes klasické DNS −> neověří se tím, že se jedná o legitimní server.

F.6 Implementace DoH ve webovém prohlížeči

Tato kapitola neobsahuje kontrolní otázky.

G Řešení třetího simulačního scénáře

G.1 Ukázka komunikace rekurzivního serveru

Tato kapitola neobsahuje kontrolní otázky.

G.2 Simulace DNS dotazů rekurzivního serveru

(1) Jaké položky se přenášejí v DNS záznamu typu RRSIG? Vyberte a stručně okomentujte alespoň 4.

O: V RRSIG záznamu se přenáší název domény, pro kterou je podpis určen, informace o hodnotě TTL, informace o délce RRSIG záznamu, informace o použitém šifrovacím algoritmu a hashovací funkci. Dále jsou zde informace o podpisu, konkrétně jeho začátek a konec platnosti, jméno autora podpisu a samozřejmě samotný podpis.

(2) Vyplňte všechny využité a zjištěné IP adresy a názvy DNS serverů do šablony na obrázku 2.6, který znázorňuje postup zjišťování IP adresy pomocí rekurzivního DNS serveru.

O: Viz obrázek na následující straně.

G.3 Wireshark analýza DNSSEC komunikace

(3) Jaký vám vyšel součet a průměrná hodnota doby odezvy v případě simulace rekurzivního serveru?

O: V ukázkovém příkladě vyšel součet doby odezvy 37,91 ms a průměrná hodnota byla 4,74 ms. Hodnoty u studentů se mohou mírně lišit.

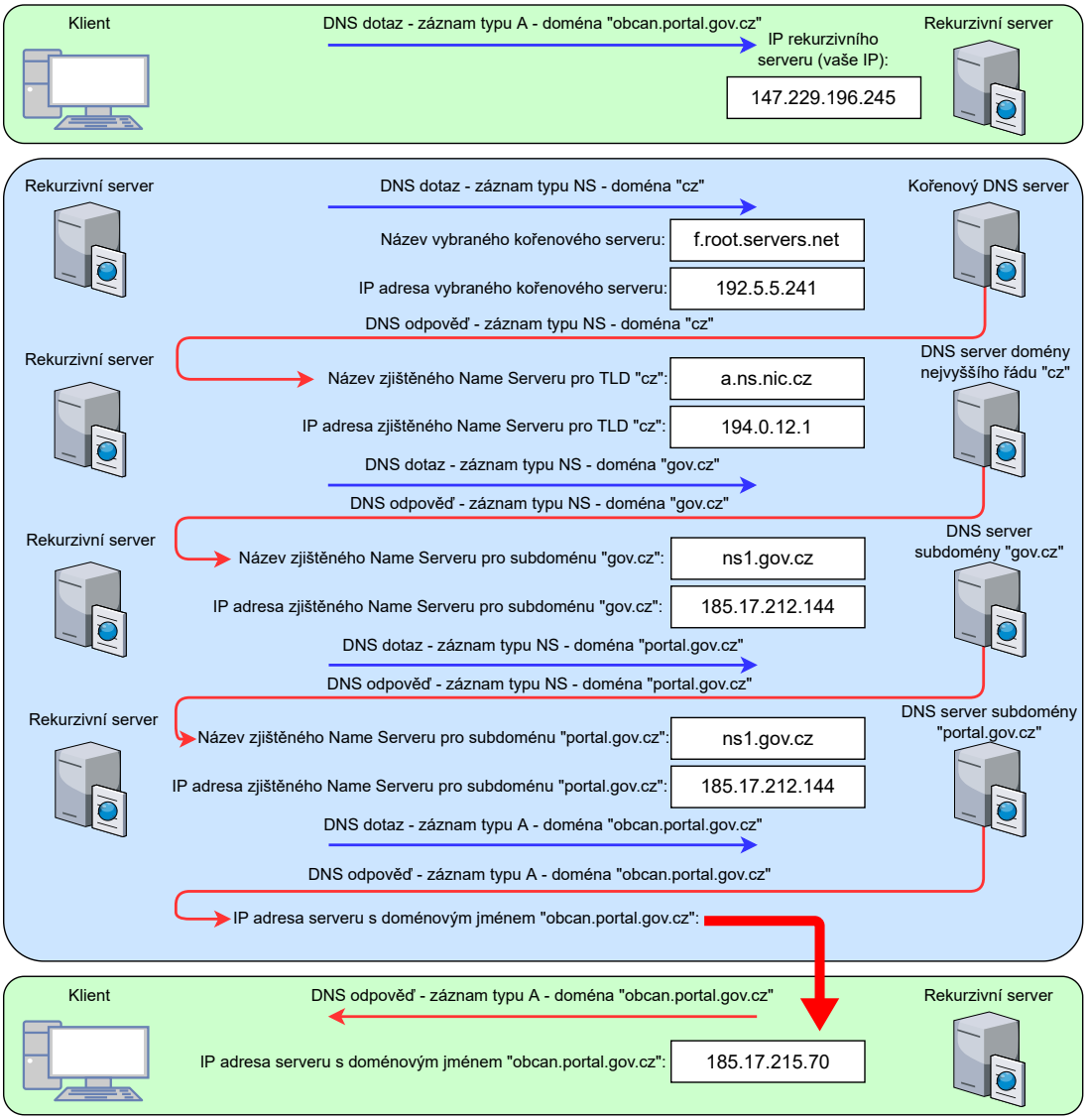
(4) Jakou dobu odezvy jste zjistili u DNS odpovědi v případě komunikace klient - DNS server?

O: V ukázkovém příkladě byla doba odezvy u komunikace klient - DNS server 10,23 ms. Hodnota u studentů se může mírně lišit, ale měla by být menší než hodnota součtu u předchozí otázky, protože by si měl server uchovávat některé záznamy v paměti a tím šetřit čas.

(5) Jaký je ve vašem případě nejvyšší počet doplňujících záznamů (Additional records) v rámci jednoho z vyfiltrovaných paketů?

O: V ukázkovém příkladě je nejvyšší počet doplňujících záznamů 9 a to v paketu číslo 2.

Dotaz klienta na IP adresu doménového jména obcan.portal.gov.cz - student zastupuje funkci rekurzivního DNS serveru



G.4 Analýza pomocí aplikace DNSViz

(6) Jmenujte 3 konkrétní vlastnosti, které lze najít v podrobnostech u jednotlivých prvků ve schématu ve webové aplikaci dnsviz.net?

O: Kromě IP adres a názvů serverů zde můžeme najít použitý šifrovací algoritmus a hashovací funkci, informace o délce klíče, key tag, hodnotu TTL a informace o stavu zabezpečení (autentizace).

(7) Vyhledejte v aplikaci dnsviz.net doménu csfd.cz. Je komunikace s touto doménou z hlediska DNS zabezpečená na všech úrovních?

O: Není, je zabezpečena pouze komunikace mezi rekurzivním serverem a servery TLD (cz) a kořenovým serverem, ale komunikace s hledanou doménou csfd.cz probíhá bez ověření autenticity DNS serveru. Viz poznámka INSECURE u položky status u šipky mezi doménami cz a csfd.cz a také u zjištěných záznamů SOA, A, MX, NS, TXT. Schéma lze zobrazit přes následující odkaz: https://dnsviz.net/d/csfd.cz/dnssec/.

(8) Dohledejte jmenné názvy (NS záznamy) doménových serverů pro doménu google.com.

O: Jmenné názvy pro doménu google.com jsou: ns1.google.com, ns2.google.com, ns3.google.com, ns4.google.com

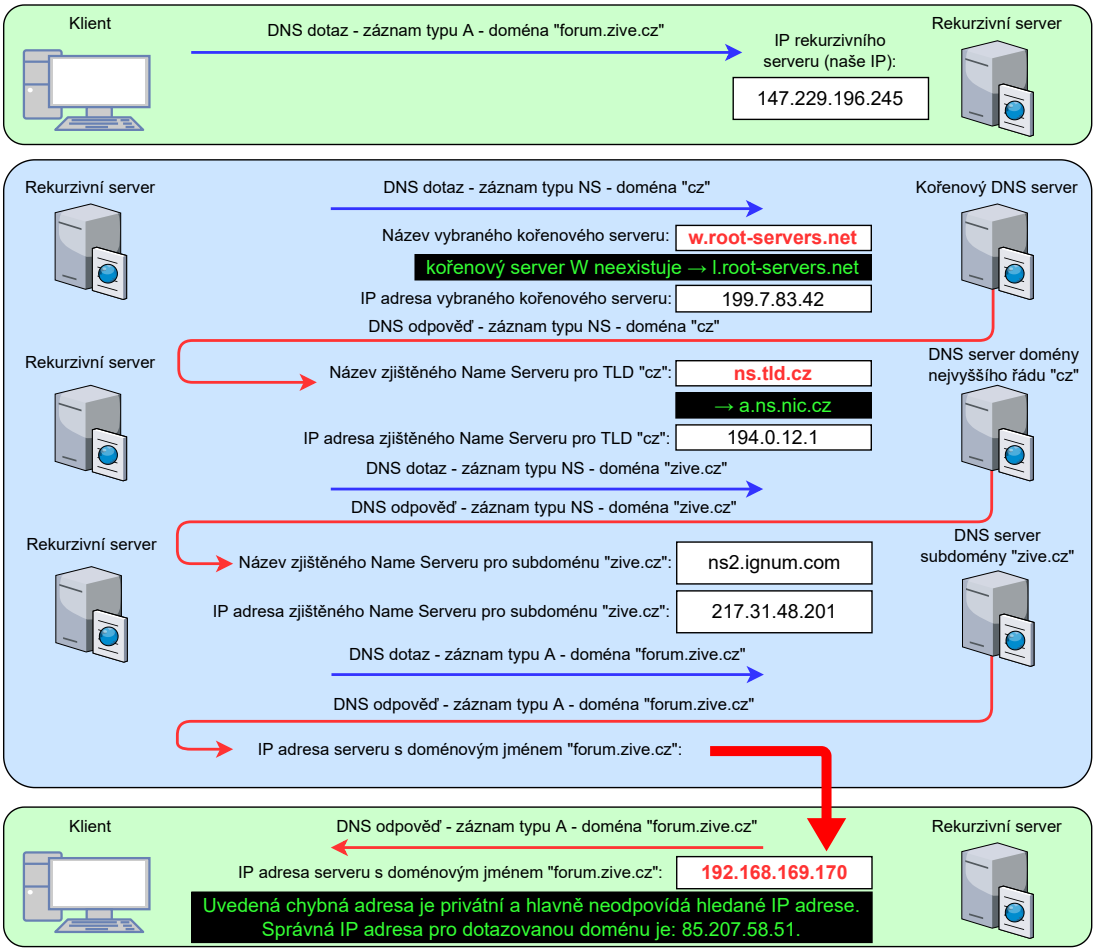
(9) Co nastane v případě, kdy neobdržíme některý z požadovaných DNSKEY nebo DS záznamů?

O: Rozpadne se řetězec důvěry a nebudeme schopni ověřit informace, ke kterým byly tyto záznamy přiřazeny.

(10) Vyhledejte a opravte 3 chyby týkající se názvů a IP adres využitých DNS serverů ve schématu na obr. 2.15.

O: Obrázek ukazující chybné i opravené údaje je zobrazen na následující straně. Chybné údaje jsou označeny červeně a správné (opravené) údaje jsou vyznačeny zeleně v černém poli.

Dotaz klienta na IP adresu doménového jména forum.zive.cz - zastupujeme funkci rekurzivního DNS serveru

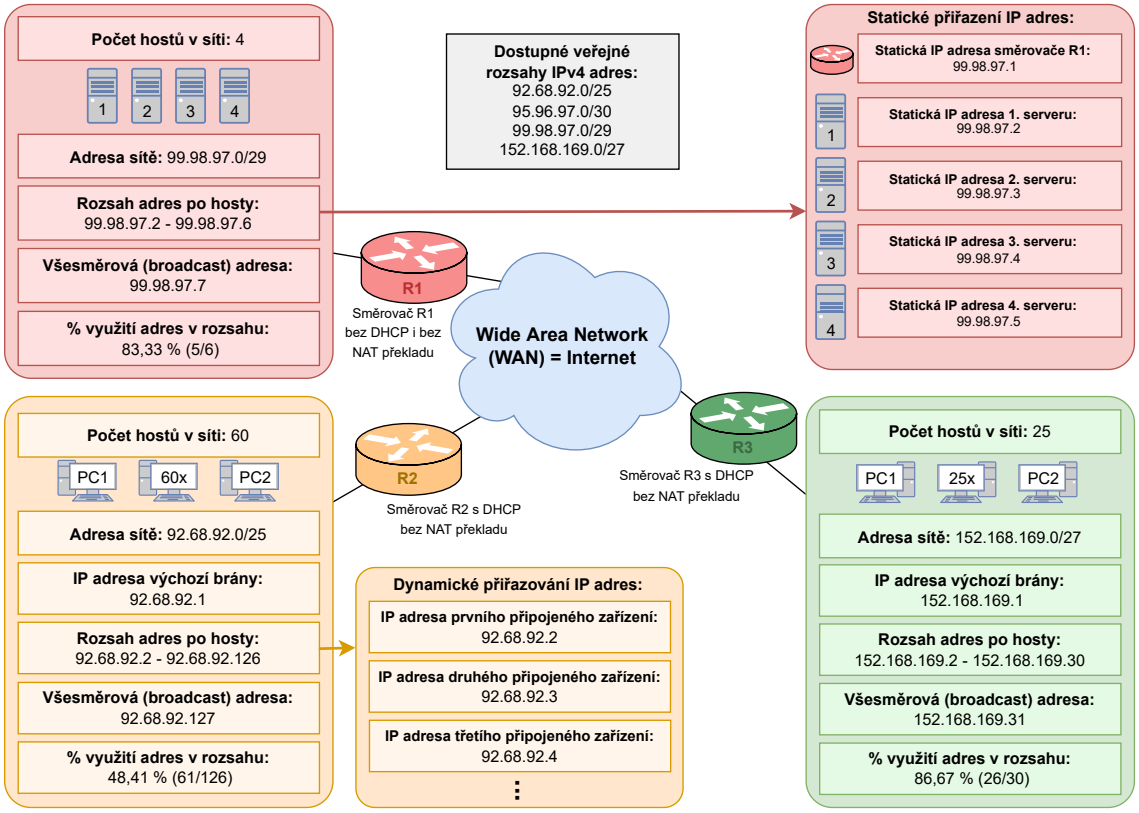


H Řešení čtvrtého simulačního scénáře

H.1 Veřejné IP adresy bez využití NAT překladu

(1) Přiřaďte rozsahy veřejných IPv4 adres a doplňte další údaje o sítích do šablony na obr. 2.3.

O:



(2) Kolika zařízením lze přiřadit IP adresu ve zbývajícím nevyužitém rozsahu adres z obr. 2.3?

O: V nevyužitém rozsahu 95.96.97.0/30 lze přiřadit pouze dvě IP adresy (95.96.97.1 a 95.96.97.2).

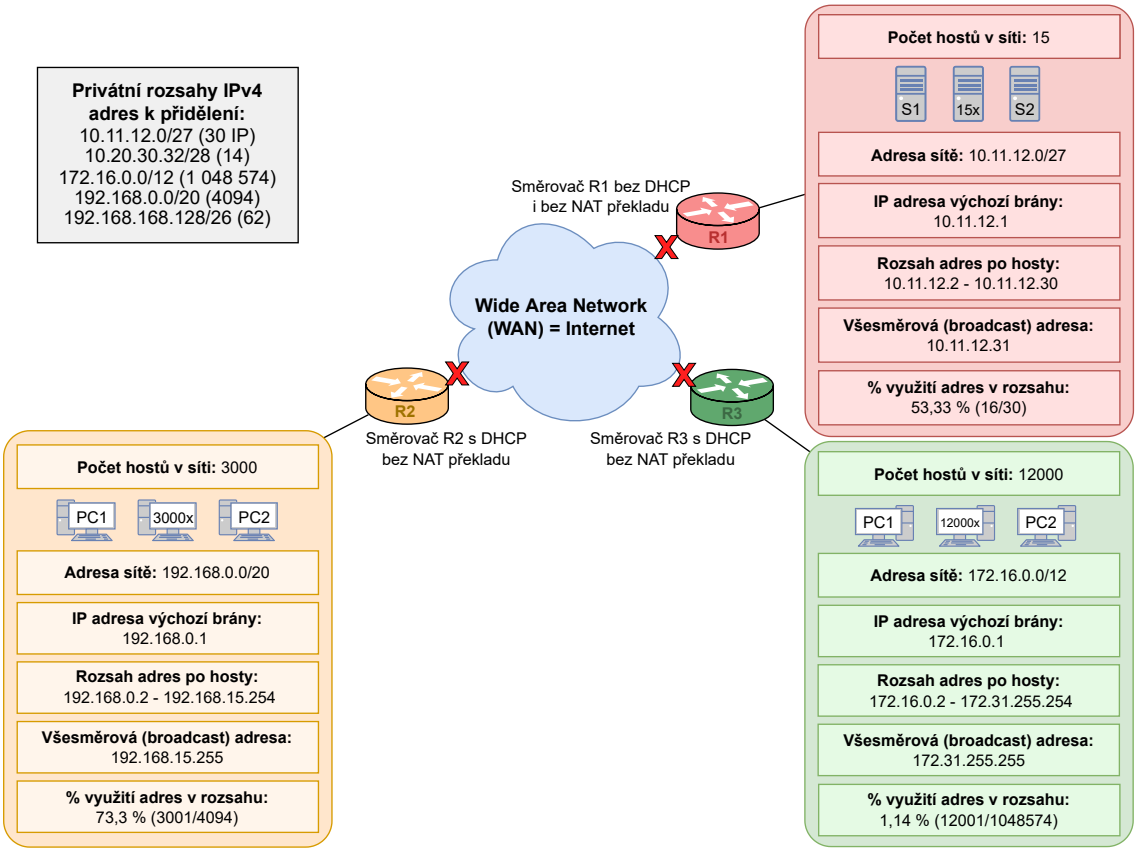
(3) Stačil by některé ze sítí na obr. 2.3 i rozsah s delší hodnotou masky sítě, než jí byl přidělen v úkolu (1)? Uveďte příklad takového rozsahu.

O: Ano, síti se směrovačem R2 (92.68.92.0/25) by stačila i maska /26 (92.68.92.0/26), kde lze přiřadit dostačujících 62 IP adres. Obecně se dá říci, že v síti která má vypočítané využití adresního prostoru pod 50 %, je možné využít vyšší hodnotu masky sítě.

H.2 Privátní IP adresy bez využití NAT překladu

(4) Přiřaďte rozsahy privátních IP adres a doplňte další údaje k jednotlivým sítím do šablony na obr. 2.6.

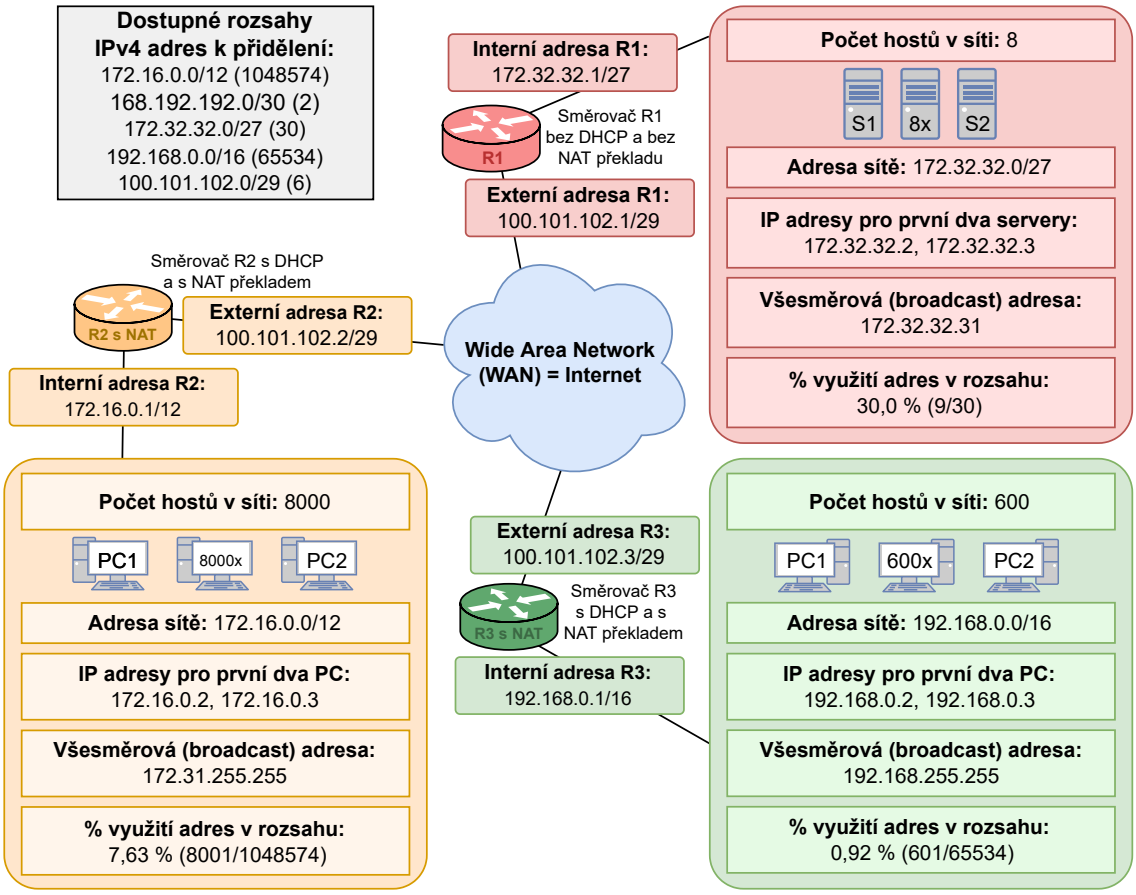
O:



H.3 Kombinace veřejných a privátních IPv4 adres

(5) Přiřaďte rozsahy veřejných a privátních IP adres a doplňte další údaje k jednotlivým sítím do šablony na obr. 2.9.

O: Viz obrázek na následující straně.



(6) Jaká bude zdrojová a cílová adresa paketu, který zachytíme na vnějším rozhraní (veřejná adresa) směrem do internetu na směrovači R2? Tento paket byl vytvořen na prvním PC v rámci interní (žluté) sítě a chceme ho doručit prvnímu serveru v červené síti za směrovačem R1.

O: Zdrojová adresa bude veřejná adresa rozhraní, na kterém jsme paket zachytili, protože již došlo k NAT překladu z původní privátní adresy (172.16.0.2). Zdrojová IP adresa tedy bude 100.101.102.2. Cílová adresa bude adresa konkrétního serveru v červené síti, protože zařízení v této síti disponují veřejnými adresami a nedochází na směrovači R1 k NAT překladu. Cílová adresa tohoto paketu tak bude 172.32.32.2.

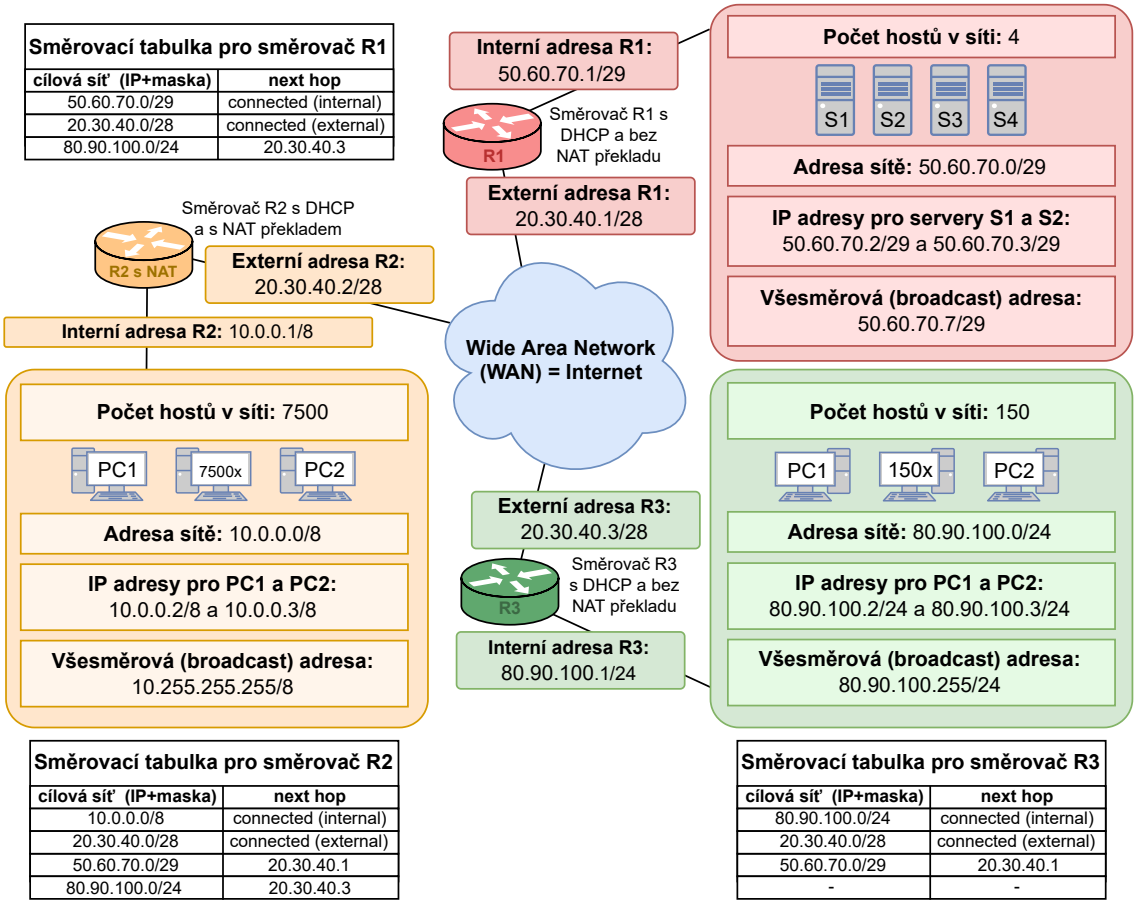
(7) Jaká bude zdrojová a cílová adresa paketu, který zachytíme na vnitřním rozhraní (privátní adresa) směrovače R2? Tento paket byl vytvořen na prvním PC v rámci interní (žluté) sítě a chceme ho doručit prvnímu PC v zelené síti za směrovačem R3.

O: Zdrojová adresa bude adresa prvního PC v interní (žluté) síti. Zdrojová adresa paketu tedy bude 172.16.0.2. Překlad zdrojové adresy bude proveden až v následujícím kroku. Cílová adresa bude adresa vnějšího rozhraní směrovače R3, protože to je poslední veřejná (a tedy i v internetu směrovatelná) IP adresa. Poté by byla adresa přeložena na adresu konkrétního zařízení v zelené síti. My však cílovou adresu zkoumáme po zachycení na směrovači R2 a tak bude cílová adresa tohoto paketu 100.101.102.3.

H.3.1 Směrovací tabulky pro směrovače s IPv4

(8) Do šablony na obr. 2.11 doplňte chybějící IP adresy pro jednotlivá zařízení v červené a zelené síti. Pozor: V tomto případě je kromě směrovače R1 NAT vypnut i na směrovači R3 a v lokální síti jsou použity veřejné adresy.

O:



(9) Do šablony na obr. 2.11 doplňte záznamy do směrovacích tabulek pro směrovače R1, R2 a R3.

O: Viz odpověď na otázku č. (8).

(10) Mějme paket vytvořený koncovým zařízením PC2 v interní (žluté) síti za směrovačem R2. Tento paket chceme doručit na server S2 v červené síti. Popište, jak se bude měnit zdrojová a cílová IP adresa paketu a kudy bude paket při cestě k cílové stanici procházet.

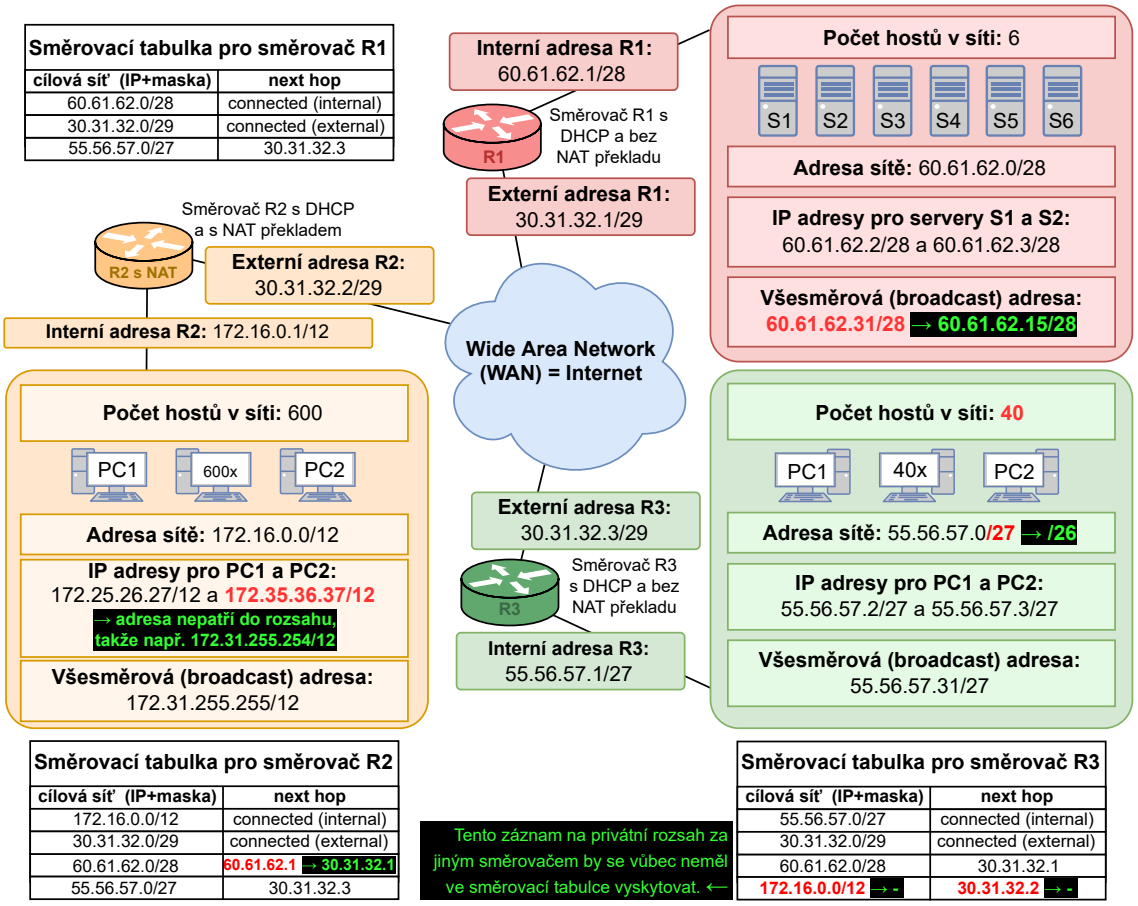
O: Paket bude vytvořen se zdrojovou IP 10.0.0.3 a cílovou adresou 50.60.70.3. z PC2 bude paket vyslán na výchozí bránu, tedy na rozhraní směrovače R2 s IP adresou 10.0.0.1. Na směrovači R2 dojde k NAT překladu zdrojové IP adresy z původní 10.0.0.3 na veřejnou adresu směrovače R2 20.30.40.2. Po nahlédnutí do směrovací tabulky na směrovači R2 zjistíme, že paket musí být vyslán směrem ke směrovači R1. Dojde tedy k předání paketu mezi směrovači R2 a R3. Paket dorazí na rozhraní směrovače R1 s veřejnou IP adresou 20.30.40.1. Interní síť za směrovačem R1 (červená) využívá veřejný rozsah adres a tak na tomto směrovači nemusí docházet k NAT překladu adres. Směrovač paket předá na interní rozhraní s IP adresou 50.60.70.1 a následně již bude paket směrován ke koncovému zařízení, tedy na server S2 s IP adresou 50.60.70.3. Cílová IP adresa paketu zůstane po celou dobu v nezměněné podobě.

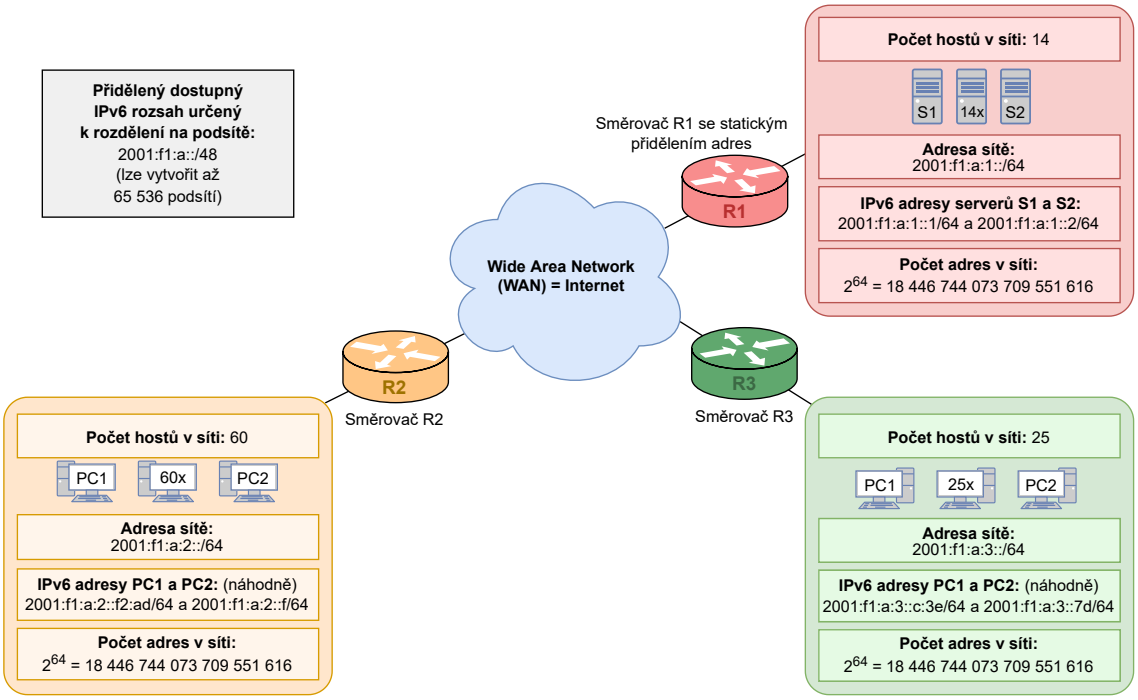
(11) Při jaké minimální hodnotě TTL u paketu z předchozího úkolu (10) ještě dojde k jeho doručení a při které hodnotě TTL dojde k zahození paketu na směrovači R1?

O: Minimální hodnota TTL při které bude ještě paket doručen na server S2 je hodnota 3. K zahození paketu by došlo pokud by byla hodnota nižší než 3. Konkrétně při výchozí hodnotě TTL 2 by došlo k zahození paketu na směrovači R1.

(12) V šabloně na obr. 2.12 vyhledejte a opravte 5 chyb v konfiguraci síťových adres (IP adresy, rozsahy, masky) a ve směrovacích tabulkách.

O: Viz horní obrázek na následující straně. Chybné údaje jsou označeny červeně a správné (opravené) údaje jsou vyznačeny zeleně v černém poli.


Směrovací tabulka pro směrovač R1
cílová síť (IP+maska) | next hop
60.61.62.0/28 | connected (internal)
30.31.32.0/29 | connected (external)
55.56.57.0/27 | 30.31.32.3
Interní adresa R1: 60.61.62.1/28
Směrovač R1 s DHCP a bez NAT překladu
Externí adresa R1: 30.31.32.1/29
Počet hostů v síti: 6
S1 S2 S3 S4 S5 S6
Adresa sítě: 60.61.62.0/28
IP adresy pro servery S1 a S2: 60.61.62.2/28 a 60.61.62.3/28
Všesměrová (broadcast) adresa: 60.61.62.31/28 → 60.61.62.15/28
Směrovač R2 s DHCP a s NAT překladem
R2 s NAT
Externí adresa R2: 30.31.32.2/29
Interní adresa R2: 172.16.0.1/12
Wide Area Network (WAN) = Internet
Počet hostů v síti: 600
PC1 600x PC2
Adresa sítě: 172.16.0.0/12
IP adresy pro PC1 a PC2: 172.25.26.27/12 a 172.35.36.37/12
→ adresa nepatří do rozsahu, takže např. 172.31.255.254/12
Všesměrová (broadcast) adresa: 172.31.255.255/12
Externí adresa R3: 30.31.32.3/29
Směrovač R3 s DHCP a bez NAT překladu
Interní adresa R3: 55.56.57.1/27
Počet hostů v síti: 40
PC1 40x PC2
Adresa sítě: 55.56.57.0/27 → /26
IP adresy pro PC1 a PC2: 55.56.57.2/27 a 55.56.57.3/27
Všesměrová (broadcast) adresa: 55.56.57.31/27
Směrovací tabulka pro směrovač R2
cílová síť (IP+maska) | next hop
172.16.0.0/12 | connected (internal)
30.31.32.0/29 | connected (external)
60.61.62.0/28 | 60.61.62.1 → 30.31.32.1
55.56.57.0/27 | 30.31.32.3
Tento záznam na privátní rozsah za jiným směrovačem by se vůbec neměl ve směrovací tabulce vyskytovat. ←
Směrovací tabulka pro směrovač R3
cílová síť (IP+maska) | next hop
55.56.57.0/27 | connected (internal)
30.31.32.0/29 | connected (external)
60.61.62.0/28 | 30.31.32.1
172.16.0.0/12 → - | 30.31.32.2 → -


Přidělený dostupný IPv6 rozsah určený k rozdělení na podsítě: 2001:f1:a::/48 (lze vytvořit až 65 536 podsítí)
Směrovač R1 se statickým přidělením adres
Počet hostů v síti: 14
S1 14x S2
Adresa sítě: 2001:f1:a:1::/64
IPv6 adresy serverů S1 a S2: 2001:f1:a:1::1/64 a 2001:f1:a:1::2/64
Počet adres v síti: 2^64 = 18 446 744 073 709 551 616
R1
Wide Area Network (WAN) = Internet
R2
Směrovač R2
R3
Směrovač R3
Počet hostů v síti: 60
PC1 60x PC2
Adresa sítě: 2001:f1:a:2::/64
IPv6 adresy PC1 a PC2: (náhodně) 2001:f1:a:2::f2:ad/64 a 2001:f1:a:2::f/64
Počet adres v síti: 2^64 = 18 446 744 073 709 551 616
Počet hostů v síti: 25
PC1 25x PC2
Adresa sítě: 2001:f1:a:3::/64
IPv6 adresy PC1 a PC2: (náhodně) 2001:f1:a:3::c:3e/64 a 2001:f1:a:3::7d/64
Počet adres v síti: 2^64 = 18 446 744 073 709 551 616

H.4 Plánování a přidělování IPv6 adresního prostoru

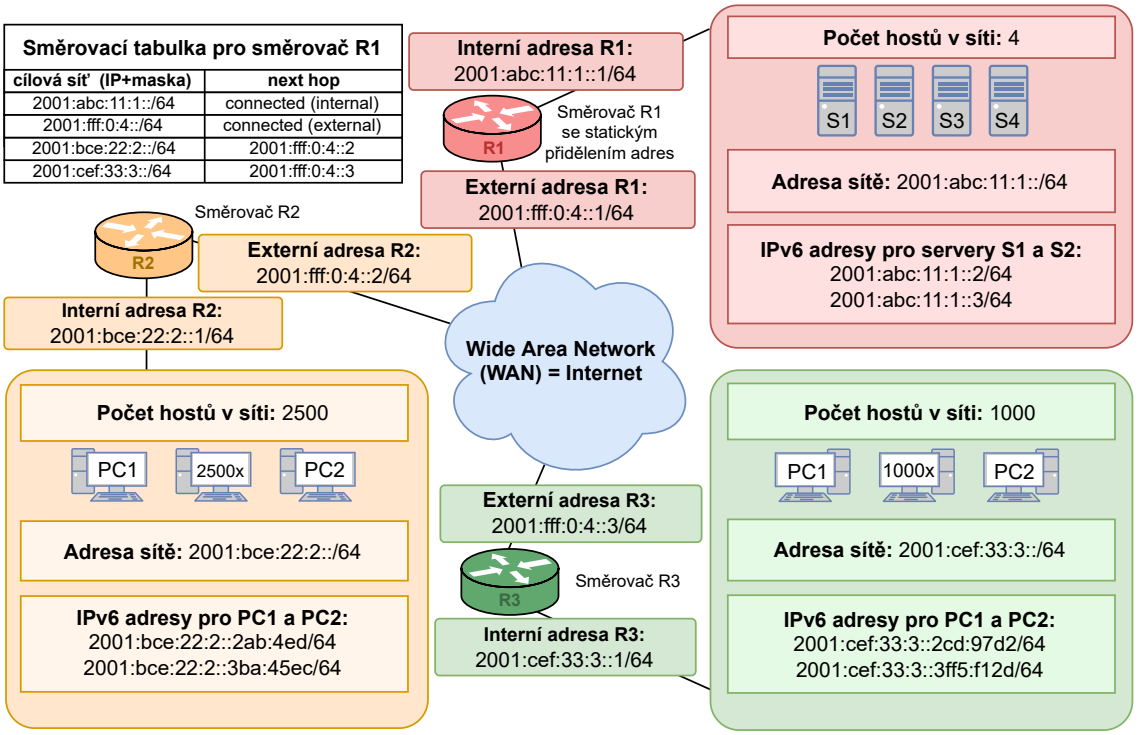
(13) Rozdělte dostupný IPv6 rozsah a vytvořené rozsahy pro podsítě následně přiřaďte jednotlivým sítím do šablony na obr. 2.16.

O: Viz spodní obrázek na předchozí straně.

H.4.1 Směrovací tabulky pro směrovače s IPv6

(14) Do šablony na obr. 2.18 doplňte záznamy v podobě IPv6 sítí a adres do směrovacích tabulek pro směrovače R1, R2 a R3.

O:

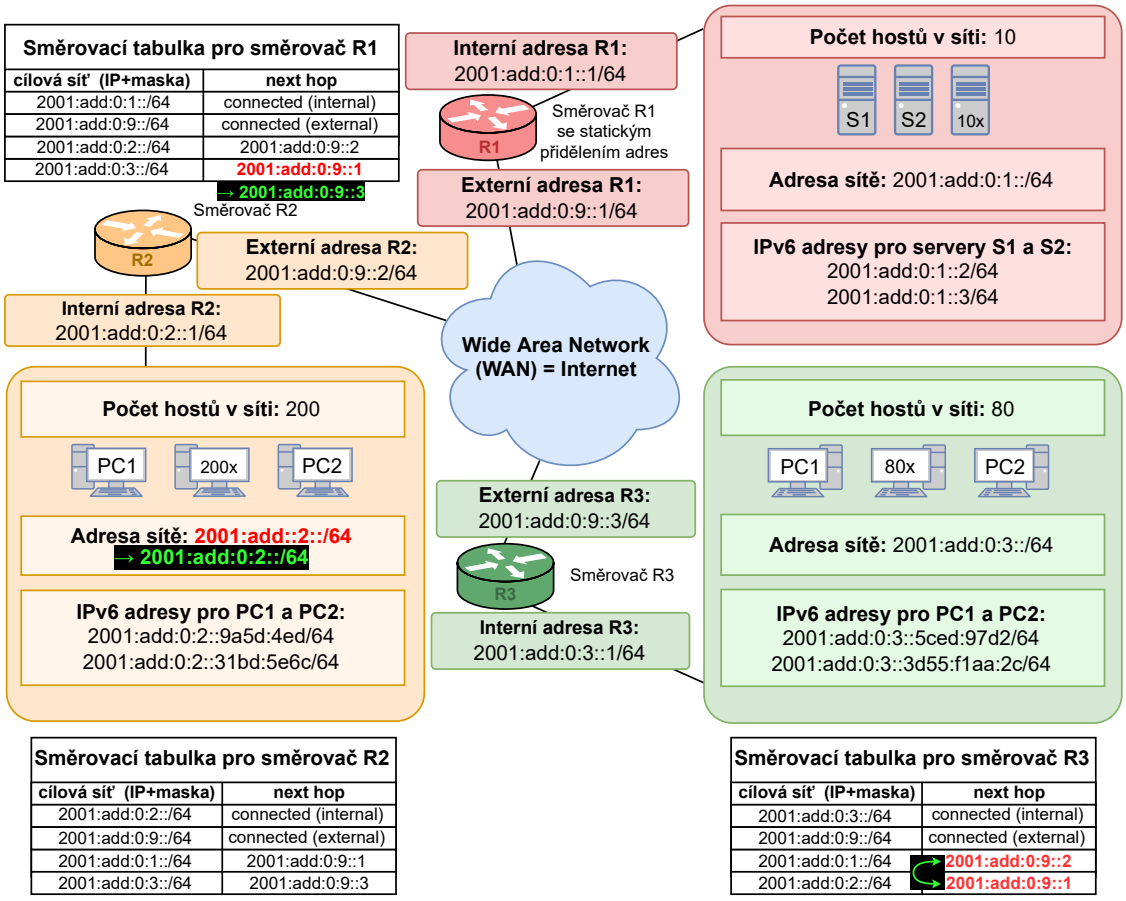


(15) Popište jednotlivé kroky směrování paketu, který byl vytvořen na PC1 ve žluté síti. Cílová destinace paketu je server S2 v červené síti.

O: Paket tedy vytvoří PC1 se zdrojovou adresou daného počítače (2001:bce:22:2::2ab:4ed/64) a cílovou adresou serveru S2 (2001:abc:11:1::3/64). Paket se ze zdrojového PC odešle na interní rozhraní směrovače R2 s IPv6 adresou 2001:bce:22:2::1. Na tomto směrovači nahlédneme do směrovací tabulky a zjistíme, že pokud se chceme dostat do sítě 2001:abc:11:1::/64 (červená), tak musíme provést skok na externí rozhraní směrovače R1 s IPv6 adresou 2001:fff:0:4::1. K tomu využijeme externí rozhraní původního směrovače R2. Na směrovači R1 opet nahlédneme do směrovací tabulky a tam zjistíme, že cílová síť je již připojena interně k tomuto směrovači. Není tak třeba využívat žádný další směrovač a paket se odešle přímo na cílový server S2 s adresou 2001:abc:11:1::3.

(16) V šabloně na obr. 2.19 vyhledejte a opravte 3 chyby v konfiguraci síťových adres (IP adresy a rozsahy) a ve směrovacích tabulkách.

O: Viz následující obrázek. Chybné údaje jsou označeny červeně a správné (opravené) údaje jsou vyznačeny zeleně v černém poli.



I Obsah odevzdané elektronické přílohy

Na této straně je zobrazen obsah odevzdané přílohy a také vzdáleného datového úložiště, které je dostupné přes následující odkaz.[1] Kořenový adresář přílohy i zmíněného úložiště obsahuje všechny soubory, které byly vytvořeny v rámci práce. Jde o čtyři vytvořené scénáře a také o soubory, které shrnují odpovědi na samostatné úkoly ze scénářů. Výsledné dokumenty jsou dostupné ve formátu pdf a pro jejich editaci pomocí programu LaTeX slouží přiložené soubory ve složkách. Kromě toho jsou zde obsaženy všechny vytvořené vektorové obrázky a schémata, které můžeme upravovat pomocí souborů s příponou .drawio a nechybí ani předpřipravené situace pro jednotlivé scénáře v podobě paketů pro program Wireshark. Soubory byly kromě informačního systému odevzdány také vedoucímu práce.

- `/` ..... Kořenový adresář poskytnutého odkazu se soubory
  - `První scénář` ..... Všechny soubory týkající se prvního scénáře
    - `Laboratorní úloha` ..... Vše pro editaci prvního scénáře
      - `sablona-prace.tcp` ..... Projekt k editaci prvního scénáře
    - `Otázky a odpovědi` ..... Vše pro editaci řešení ke scénáři
    - `Předpřipravené soubory (Wireshark)`
    - `První-scénář.pdf` ..... Vypracovaný první scénář
    - `První-scénář-řešení.pdf` ..... Otázky a odpovědi k prvnímu scénáři
  - `Druhý scénář` ..... Všechny soubory týkající se druhého scénáře
    - `Laboratorní úloha` ..... Vše pro editaci druhého scénáře
      - `sablona-prace.tcp` ..... Projekt k editaci druhého scénáře
    - `Otázky a odpovědi` ..... Vše pro editaci řešení ke scénáři
    - `Předpřipravené soubory (Wireshark)`
    - `Druhý-scénář.pdf` ..... Vypracovaný druhý scénář
    - `Druhý-scénář-řešení.pdf` ..... Otázky a odpovědi ke druhému scénáři
  - `Třetí scénář` ..... Všechny soubory týkající se třetího scénáře
    - `Laboratorní úloha` ..... Vše pro editaci třetího scénáře
      - `sablona-prace.tcp` ..... Projekt k editaci třetího scénáře
    - `Otázky a odpovědi` ..... Vše pro editaci řešení ke scénáři
    - `Předpřipravené soubory (Wireshark)`
    - `Třetí-scénář.pdf` ..... Vypracovaný třetí scénář
    - `Třetí-scénář-řešení.pdf` ..... Otázky a odpovědi ke třetímu scénáři
  - `Čtvrtý scénář` ..... Všechny soubory týkající se čtvrtého scénáře
    - `Laboratorní úloha` ..... Vše pro editaci čtvrtého scénáře
      - `sablona-prace.tcp` ..... Projekt k editaci čtvrtého scénáře
    - `Otázky a odpovědi` ..... Vše pro editaci řešení ke scénáři
    - `Čtvrtý-scénář.pdf` ..... Vypracovaný čtvrtý scénář
    - `Čtvrtý-scénář-řešení.pdf` ..... Otázky a odpovědi ke čtvrtému scénáři
  - `Editovatelné obrázky a schémata` ..... Složka s vektorovými obrázky
  - `Diplomová-práce-Šíma.pdf` ..... Kompletní diplomová práce

[1] `https://owncloud.cesnet.cz/index.php/s/OLJDN1xxMQsBBUq`