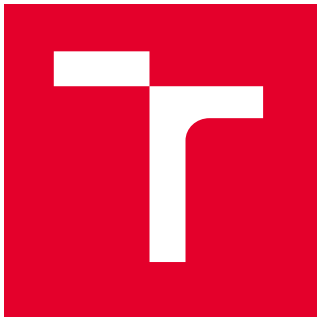


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MODELOVÁNÍ ZÁKLADNÍCH PRINCIPŮ KOMUNIKAČNÍCH TECHNOLOGIÍ

MODELING THE BASIC PRINCIPLES OF COMMUNICATION TECHNOLOGIES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Michal Ruiner

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. Jiří Hošek, Ph.D.

SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Michal Ruiner ***ID:*** 220825

Ročník: 3 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Modelování základních principů komunikačních technologií

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je návrh a implementace několika simulačních scénářů, které budou názorně demonstrovat základní principy moderních komunikačních technologií jako je směrování v paketových sítích, mechanizmy fungování transportních protokolů či řízení kvality služeb. Simulační modely budou popsány formou detailních návodů (psaných v anglickém jazyce), které budou sloužit pro výukové účely. Každý návod bude obsahovat teoretický úvod do dané problematiky, bodově popsaný postup tvorby simulačního scénáře a následné vyhodnocení dosažených výsledků formou grafů a tabulek. Závěr každého návodu bude tvořen několika otázkami, které ověří, zda student po dokončení laboratorní úlohy porozuměl problematice. Veškeré výsledky dosažené při řešení bakalářské práce budou shrnuty v podobě závěrečné práce.

DOPORUČENÁ LITERATURA:

[1] LEE, Bob. Network Lab Scenarios II [IP & Routing]: Becoming the Network Expert with Packet Tracer. ISBN: 978-1798782019. 2019.

[2] KAUFMANN, Morgan. Computer Networks: A Systems Approach. ISBN: 978-0128182000. 2021.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 31.5.2022

Vedoucí práce: doc. Ing. Jiří Hošek, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce se věnuje návrhu 5 laboratorních úloh, v nichž se modelují základní principy komunikačních technologií. Čtenář je nejdříve seznámen s dostupným softwarem pro simulaci síťových technologií. Porovnávají se zde simulátory a emulátory. Pro laboratorní úlohy je zvolen Packet Tracer, software vyvinutý společností Cisco, a síťový analyzátor Wireshark pro zachytávání provozu na lokálním počítači. Dále se popisuje referenční model ISO/OSI pro počáteční uvedení do principu síťové komunikace. Kapitola Směrování vysvětluje základní problematiku vyhledávání síťových cest, obsah směrovacích tabulek a dělení podle typu směrování a skupin směrovacích protokolů. V následující kapitole se podrobně rozebírají jednotlivé komunikační protokoly využité v laboratorních úlohách. Poslední část krátce popisuje realizaci laboratorních úloh. Laboratorní návody jsou uvedeny v přílohách.

KLÍČOVÁ SLOVA

Simulátor, Emulátor, Packet Tracer, Wireshark, ISO/OSI, směrování, ARP, RIP, OSPF, UDP, TCP, DHCP, komunikační technologie

ABSTRACT

This bachelor thesis deals with the design of 5 laboratory tutorials where the basic principles of communication technologies are modelled. A reader is apprised with the available software first for the simulation of network technologies. The simulators and emulators are compared. Packet Tracer, a software developed by the Cisco company, is chosen for the lab exercises and Wireshark, a network protocol analyzer, for the traffic capturing on a local PC. Then the reference model ISO/OSI is described for the basic introduction to the principle of network communication. The Chapter 3 explains the basic problematics of network paths searching, the routing table contents and division according to the type of routing and routing protocols groups. In the next chapter, particular communication protocols used in the labs are analyzed in detail. Final part briefly describes the implementation of all laboratory exercises. Laboratory tutorials are provided in the appendices.

KEYWORDS

Simulator, Emulator, Packet Tracer, Wireshark, ISO/OSI, routing, ARP, RIP, OSPF, UDP, TCP, DHCP, communication technologies

Vysázeno pomocí balíčku `thesis` verze 4.07; `http://latex.feec.vutbr.cz`

RUINER, Michal. *Modelování základních principů komunikačních technologií*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 212 s. Bakalářská práce. Vedoucí práce: doc. Ing. Jiří Hošek, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Michal Ruiner
VUT ID autora:	220825
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Modelování základních principů komunikačních technologií

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 bakalářské práce panu doc. Ing. Jiřímu Hoškovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Komunikační technologie se v dnešní době staly klíčovými pro fungování moderního světa. Usnadňují každodenní práci podniků, počínaje malými firmami až po nadnárodní korporace (mezi něž patří např. Microsoft, Google a IBM). Jejich důležitost nelze opomenout ani na úrovni mezinárodní komunikace vládních institucí a spolupráce ve formě varování před možnými kybernetickými útoky, které se objevují stále častěji. Pro umožnění komunikace mimo lokální síť, obecně autonomní systém, je třeba zřídit připojení do celosvětové sítě internet. Mezi známé poskytovatele internetového připojení, též známých pod zkratkou ISP (Internet Service Provider), patří CETIN, AT&T, GiTy, O2 a další.

Pro správný průběh komunikace se musí dodržovat pravidla, která určují, jakým způsobem se data přenáší, zajišťují stejný význam pro obě komunikující strany atp. Tato pravidla se nazývají protokoly. Firmy si mohou vytvářet proprietární řešení pro svá zařízení nebo využívat veřejné standardy definované mezinárodními organizacemi (ISO, IETF, IEEE, ITU atd.).

Cílem práce je navrhnout 5 níže uvedených laboratorních úloh a následně je realizovat ve zvoleném simulátoru.

- Laboratorní úloha č. 1 se zabývá analýzou protokolu ARP (Address Resolution Protocol) v lokální síti.
- Laboratorní úloha č. 2 srovnává statické a dynamické směrování v Packet Traceru. Jako dynamický směrovací protokol byl zvolen protokol RIP (Routing Information Protocol) verze 1 (RIPv1).
- Laboratorní úloha č. 3 se zaměřuje na 2 základní skupiny dynamických směrovacích protokolů – *Distance Vector* a *Link State*. Pro demonstraci vlastností daných skupin se simulují protokoly RIPv2 a OSPF (Open Shortest Path First).
- Laboratorní úloha č. 4 vysvětluje rozdíly transportních protokolů UDP (User Datagram Protocol) a TCP (Transmission Control Protocol) s využitím aplikačních protokolů HTTP (Hypertext Transfer Protocol) a DNS (Domain Name System).
- Laboratorní úloha č. 5 přibližuje činnost aplikačního protokolu DHCP (Dynamic Host Configuration Protocol).

Práce se člení do kapitol podle diskutovaných témat. První kapitola se zabývá analýzou dostupného softwaru pro simulaci a výběrem nejvhodnějšího z nich. Na závěr se srovnají vlastnosti programů formou tabulky. Druhá kapitola popisuje referenční model ISO/OSI pro představení základů síťové komunikace. Třetí kapitola se věnuje problematice směrování. Čtvrtá kapitola podrobně rozebírá jednotlivé komunikační protokoly, jimiž se laboratoře zabývají. Poslední kapitola krátce shrnuje

scénář jednotlivých laboratorních úloh. Součástí práce jsou přílohy s návody k vypracovaným laboratorním úlohám.

1 Software pro simulaci síťových technologií

Firmy i státní orgány se staly natolik závislými na internetové komunikaci, že i výpadek v řádu minut může způsobit fatální následky pro fungování společnosti. Není proto výjimkou, že síťoví inženýři před zavedením nových nebo upgradu stávajících technologií využívají dostupných simulátorů a emulátorů.

1.1 Rozdíl mezi simulátory a emulátory

Síťové simulátory a emulátory se společně využívají pro testovací účely, v jádru se však jedná o zcela odlišné nástroje. Již na počátku se uživatelé musí zamyslet, co očekávají od výsledku práce. Jedním z možných scénářů je tvorba teoretického modelu, kde se vytvoří topologie sítě, testuje se konfigurace směrovačů, přepínačů a dalších aktivních prvků sítě, případně se zkoumá funkce protokolů. Pro tento případ lze výhodně použít některý z dostupných simulátorů, mezi něž se řadí Packet Tracer, ns-3 (Network Simulator 3) a NetSim. Většina simulátorů tohoto typu využívá DES (Discrete-event simulation – diskrétní simulace), kde určitá událost (např. příchod paketu na směrovač a jeho následné předání na odchozí rozhraní) vyvolá specifickou reakci v následujícím okamžiku (např. přijetí paketu druhým koncem linky v případě *point-to-point* spoje). Během dvou po sobě následujících událostí se předpokládá, že nenastane žádná změna v systému.

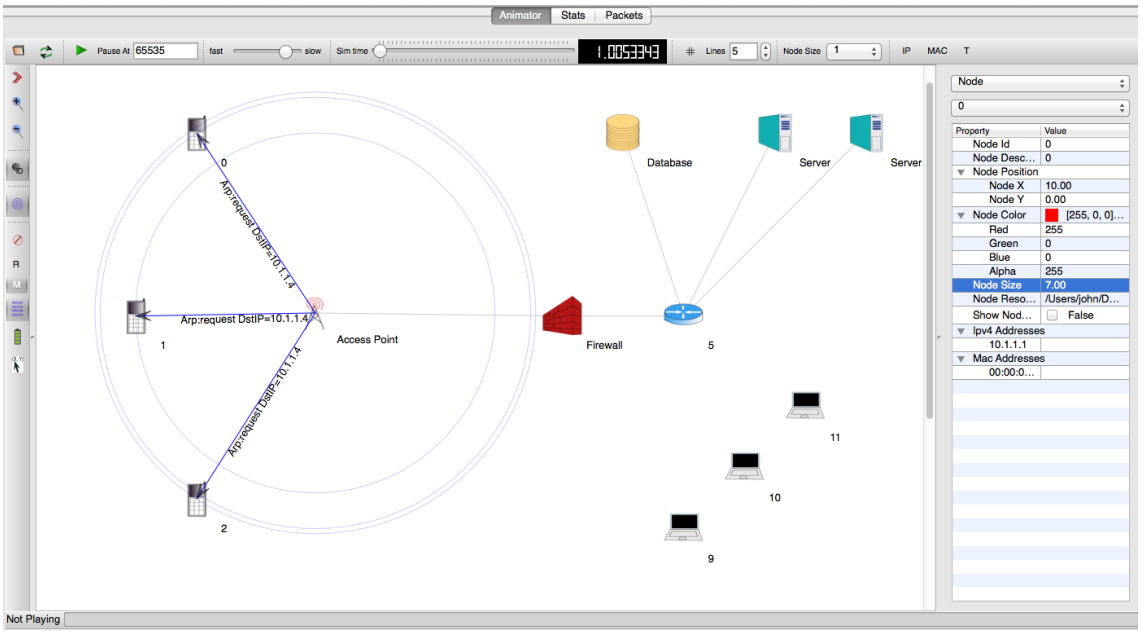
Rozdílný přístup vyžaduje testování, které zasahuje do skutečné sítě v provozu (emulace). Do prostředí emulátoru se přidají podporované síťové prvky (dáno výrobcem a verzí OS), které vytvoří virtuální síť. Zde se následně zpracovává živý provoz připojených reálných nebo virtuálních počítačů a aplikací. Důležité je zmínit, že virtuální síť pouze napodobuje chování skutečné sítě. Typicky se analyzuje provoz aplikací, jejichž data odchází na zařízení, kde je emulátor spuštěn. Prostřednictvím virtuální sítě lze do trasy uměle vložit ztrátovost paketů, zpoždění, chybovost na linkách aj. Pomocí těchto veličin se vyhodnocuje kvalita z pohledu koncového uživatele, případně výkon samotné aplikace. Příkladem známých síťových emulátorů jsou GNS3 (Graphical Network Simulator-3) a EVE-NG (Emulated Virtual Environment – Next Generation).

Lze tedy shrnout, že simulátory jsou vhodné pro výukové účely, pochopení problematiky komunikačních protokolů a návrhu sítě od úplného počátku včetně testování konfigurací. Emulátory se s výhodou využijí v situacích, kdy je třeba monitorovat výkon sítě, reakce aplikací v nestandardních stavech a chování sítě po připojení nového zařízení [1].

1.2 Porovnání dostupných softwarů

1.2.1 Network Simulator 3

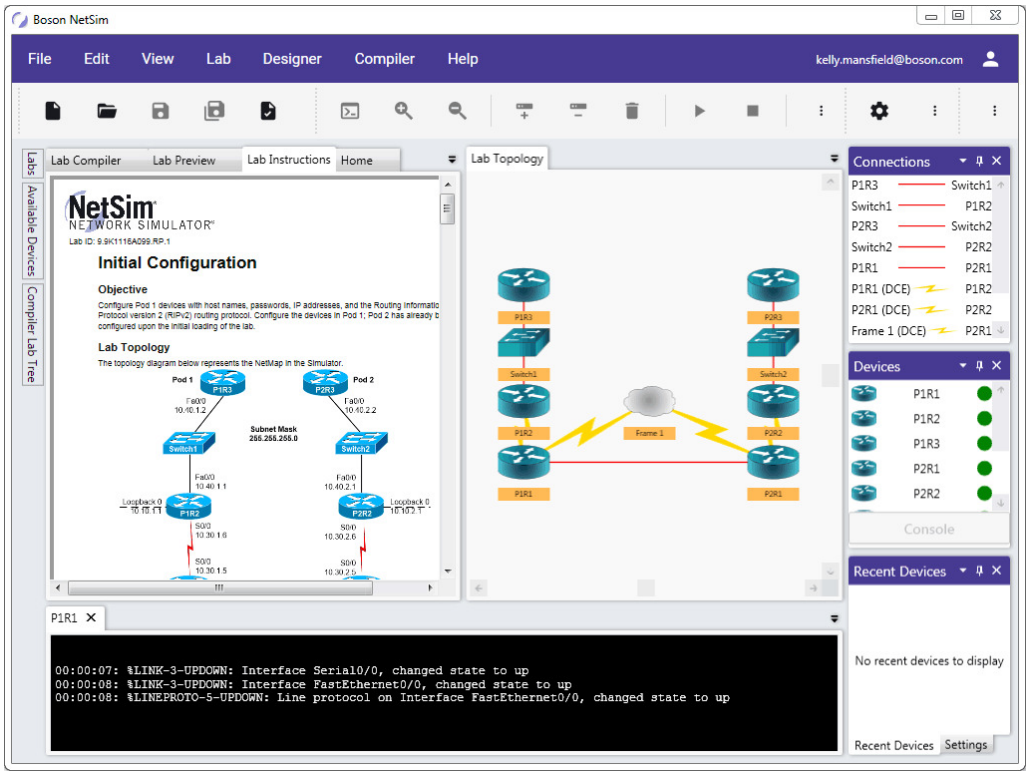
Network Simulator 3 (ns-3) patří do skupiny open source síťových simulátorů. Předpokládá práci v CLI (Command Line Interface – příkazový řádek). Software je napsán v jazyce C++, uživatel tedy získá zdrojový kód, v němž naprogramuje požadovaný scénář a ten následně zkompiluje v C++ kompilátoru (existuje možnost psaní skriptů v jazyce Python). Způsob realizace dává uživateli značnou kontrolu nad chováním simulace, vyžaduje ale jistou míru znalosti programování. Do zdrojového kódu lze zakomponovat část, která vygeneruje PCAP soubor, jenž obsahuje kompletní přehled komunikace z proběhlé simulace. Ten lze následně otevřít v analyzátoru provozu (např. Wireshark nebo NetworkMiner), který se využije např. pro vyhodnocení statistiky provozu. Pro grafickou animaci vytvořila skupina vývojářů program NetAnim, který zobrazuje Obr. 1.1. Network Simulator 3 je primárně určen pro Linux a macOS, Windows není v základu podporován. Alternativně se dá přistoupit k virtuálnímu stroji, na němž se spustí některá z distribucí Linuxu, případně prostředí, které umožňuje zkompilovat kód určený pro unixové systémy ve Windows (např. Cygwin) [2].



Obr. 1.1: Ukázka prostředí NetAnim [3].

1.2.2 NetSim

NetSim simulátor společnosti Boson se zaměřuje na Cisco certifikace CCNA (Cisco Certified Network Associate) a CCNP (Cisco Certified Network Professional), konkrétně CCNP ENCOR (Implementing and Operating Cisco Enterprise Network Core Technologies) a CCNP ENARSI (Implementing Cisco Enterprise Advanced Routing and Services). Software přichází ve dvou variantách: *Demo* a *Full*. Demo verze je dostupná zdarma, platí zde ale omezení dostupných příkazů a předpřipravených laboratoří. Oproti tomu Full verze vyžaduje zakoupení licence a v závislosti na zvoleném produktu odemyká rozšiřující technologie a laboratorní scénáře. NetSim neomezuje pouze na předem vytvořené scénáře. Uživatel má možnost vytvářet vlastní topologie a nakonfigurovat zařízení podle svých požadavků. Okna programu se dělí na sekce, v nichž se nachází všechny laboratorní návody (dostupnost se liší na základě zakoupené licence), logická topologie, scénář a tabulka implementovaných zařízení a jejich vzájemných propojení. Samotná konfigurace zařízení se provádí prostřednictvím CLI [4]. Prostředí programu je možné vidět na Obr. 1.2.

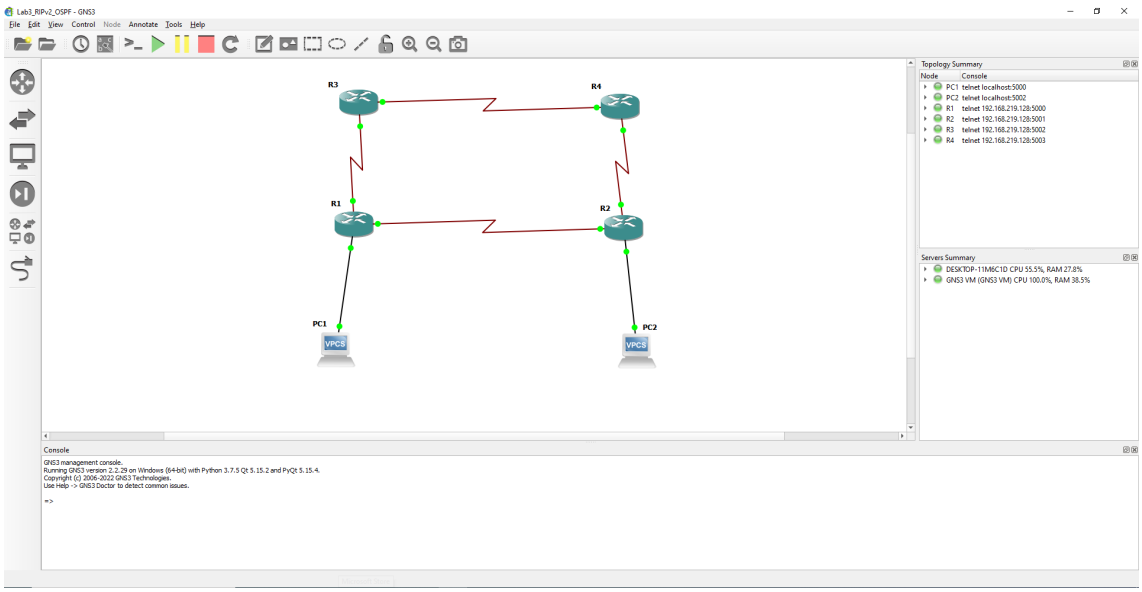


Obr. 1.2: Boson NetSim [5].

1.2.3 Graphical Network Simulator-3

Graphical Network Simulator-3 (GNS3) se řadí mezi open source síťové emulátory. Využívají jej síťoví inženýři po celém světě k simulování a testování skutečných a virtuálních sítí [6]. Skládá se ze dvou částí: GUI (Graphic User Interface – grafické uživatelské rozhraní) a VM (Virtual Machine – virtuální stroj). V grafickém rozhraní si uživatel sestaví vlastní logickou topologii přidáním aktivních síťových prvků, koncových zařízení a následně jejich vzájemným propojením. K základním prvkům, které jsou dostupné již po instalaci, patří Ethernet rozbočovač, Ethernet přepínač, ATM (Asynchronous Transfer Mode) přepínač, Frame Relay přepínač a virtuální PC. Pro přidání dalších prvků je možné obrátit se na GNS3 Marketplace, kde se vyskytují podporovaná zařízení, operační systém však musí být nahrán dodatečně. Dostupnost OS závisí na výrobcích, kteří na svých oficiálních stránkách mohou zpřístupnit image (soubor s OS) zdarma či placeně, nebo je nutné zakoupit si fyzické zařízení s již nahraným operačním systémem. Kupříkladu MikroTik poskytuje speciálně do virtuálního prostředí image v sekci Cloud Hosted Router (dále jen CHR). CHR zahrnuje 4 typy licencí. Využít lze licenci zdarma, která však omezuje rychlost na 1 Mbit na interface. Zbylé 3 licence jsou placené, P1 poskytuje omezení rychlosti 1 Gbit na interface, P10 10 Gbit na interface a P-Unlimited zahrnuje neomezenou rychlost. Na druhé straně např. pro využívání OS společnosti Cisco (IOS) je nutné si všechny licence zakoupit. Existuje zde ale možnost překopírování IOS z reálného hardwaru do emulátoru. VM vyžaduje přítomnost hypervizoru, např. VMware Workstation Pro/Player nebo Oracle VM VirtualBox. Jeho instalace se doporučuje pro spouštění virtuálních prvků sítě, např. Cisco VIRL (Virtual Internet Routing Lab). Na počet zařízení v topologii se neklade omezení, rozhodující je výkon PC, na němž běží emulátor. Obr. 1.3 ukazuje grafické rozhraní programu. Výhody GNS3:

- Software je open source.
- Podpora zařízení různých společností.
- Možnost sledování provozu pomocí síťových analyzátorů, např. Wireshark.



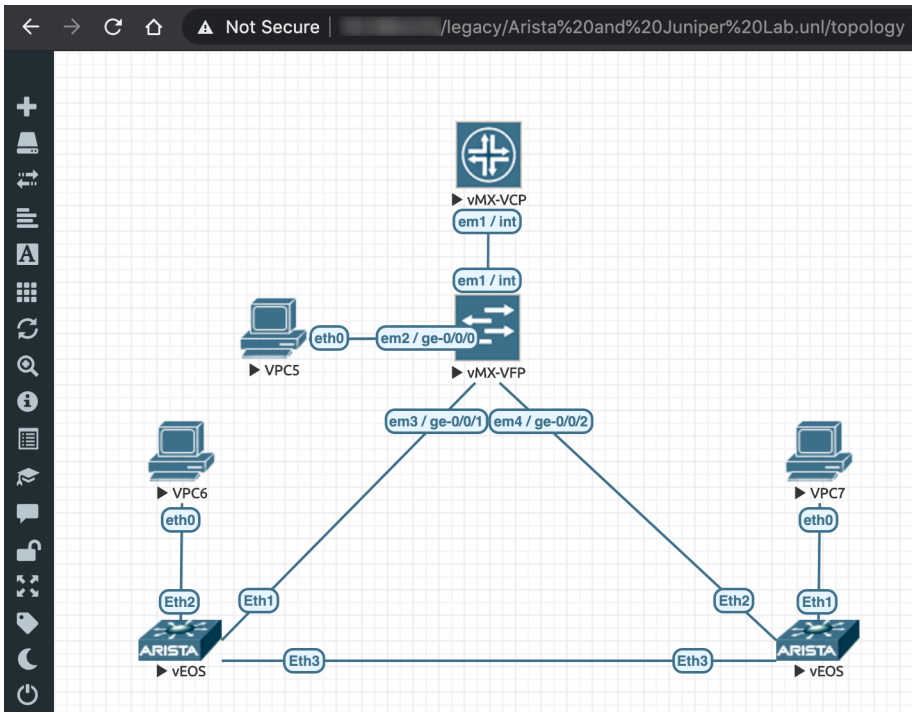
Obr. 1.3: GNS3 GUI.

1.2.4 Emulated Virtual Environment – Next Generation

Emulated Virtual Environment – Next Generation (EVE-NG) spadá do kategorie síťových emulátorů. Vývojáři poskytují 2 licence: zdarma (*Community Edition*) a placenou (*Professional/Learning Center*). Placená licence obsahuje ve srovnání s licencí zdarma navíc určité mechanismy, které značně prohlubují možnosti emulací. Zde jsou uvedeny pouze některé z nich:

- Možnost zachytávání provozu na sériových linkách.
- Import a export konfigurací do virtuální laboratoře.
- Pokročilé designování objektů.
- Úpravy parametrů linek – ztrátovost paketů, jitter, zpoždění.
- Paralelní připojení administrátora na zařízení jiného uživatele.
- Integrovaný Wireshark.

EVE-NG je srovnatelný s GNS3 po stránce přístupu k síťovým OS. V základu software obsahuje pouze VPC (Virtual PC – virtuální počítač), operační systémy jiných společností je nutné dokoupit, případně stáhnout z oficiálních stránek. Protože lze stáhnout instalační soubor ve formátu ISO, odpadá nutnost použití VM, ale při stažení souboru OVF se VM stále vyžaduje. Při virtualizaci je nutné dbát na podporované nástroje, jelikož v době psaní této práce není podporován VirtualBox. Veškerá tvorba topologií probíhá ve webovém prostředí, pro správu zařízení lze využít HTML5 konzoli, případně SSH/telnet klienta (např. Putty) [7]. Webové prostředí ukazuje Obr. 1.4.



Obr. 1.4: EVE-NG webové rozhraní [8].

1.2.5 Packet Tracer

Packet Tracer je proprietární nekomerční software společnosti Cisco, který se dnes hojně využívá k výukovým a testovacím účelům po celém světě. Program primárně vznikl pro Cisco akademie CNNA, ale s výhodou se dá využít i pro širokou škálu zájemců o komunikační technologie, kteří si zde vyzkouší tvorbu sítí většinou na úrovni logické topologie, ovšem lze přejít i do topologie fyzické. Ta se upotřebí např. při návrhu domácností s implementací IoT (Internet of Things – internet věcí) nebo při realizaci propojení poboček či měst. Software je tedy vhodný pro:

- Návrh sítě od úplného počátku.
- Testování konfigurací před samotnou implementací do reálného provozu.
- Softwarové zabezpečení prvků a následné ověření funkčnosti.
- Sledování interních procesů během síťové komunikace.
- Návrh IoT systémů v domácnostech.
- Zkoumání změn ve struktuře dat při průchodu jednotlivými vrstvami RM (Reference Model – referenční model) ISO/OSI v závislosti na použitých protokolech.

Uživatelé si zde vybírají z rozmanité nabídky **Cisco zařízení**. První velkou skupinou jsou síťová zařízení, kde se dále nachází podskupiny směrovačů, přepínačů,

bezdrátových zařízení, hubů, firewallů a emulací WAN (Wide Area Network). Skupina koncová zařízení obsahuje PC, laptopy, servery, IP telefony atd. Podskupiny se týkají IoT, chytrých měst, průmyslu a elektrické sítě. Další důležitou skupinou jsou propojení. Zde se nachází různorodé kabely, kterými se propojují zařízení v závislosti na jejich typu. Řadí se sem sériové linky DCE/DTE (Data Communications Equipment / Data Terminal Equipment), konzolový kabel, koaxiální kabel, optický kabel aj. Mezi nejpoužívanější (v lokálních sítích) patří přímý a křížený kabel. Křížený kabel se v minulosti využíval k propojení zařízení se stejnou funkčností, tedy např. PC-PC, směrovač-směrovač a přepínač-přepínač. Přímým kabelem se propojovala zařízení navzájem různá, tedy např. PC-směrovač, PC-přepínač a přepínač-směrovač. V dnešní době již existuje technologie Auto MDI-X, která dokáže automaticky detekovat, zda propojení vyžaduje přímý nebo křížený kabel a přizpůsobí podle toho konfiguraci rozhraní.

Režimy simulace

Program disponuje dvěma módy: *Realtime* a *Simulation*. V režimu Realtime probíhá tok dat v reálném čase, tedy úspěch či neúspěch přenosu lze zjistit pouze na základě výsledku komunikace (např. správné zobrazení webové stránky pomocí protokolu HTTP). Odhalení případného chybného článku se provádí síťovými utilitami ping, traceroute a další. Režim Simulation umožňuje sledovat komunikaci od vzniku zprávy, přes průchod jednotlivými zařízeními až po doručení k cíli. Ve všech těchto krocích lze detailně zobrazit obsah rámců (na úrovni linkové vrstvy) nebo paketů (na úrovni síťové vrstvy) a vyčíst adresy. Koncová zařízení, která pracují se všemi vrstvami ISO/OSI, umožňují zobrazit i použité porty, nastavené příznaky, aplikační protokol aj. Při nesprávné konfiguraci je tedy jednoduše zjistitelné místo poruchy.

Konfigurace zařízení

Konfigurace zařízení probíhá prostřednictvím: *CLI*, *Config*, *Desktop* nebo *GUI*. Zpočátku lze pro jednoduchost využívat okno Config, které intuitivně poskytuje jednoduchý způsob konfigurace základních parametrů, např. statické a dynamické směrování (protokol RIP), přidání/odebrání VLAN a nastavení rozhraní (MAC (Media Access Control) a IP (Internet Protocol) adresa, režim Duplex). Tento způsob je uživatelsky přívětivý, omezuje ale možnosti pouze na úplný základ. V praxi se konfiguruje téměř výhradně prostřednictvím CLI, proto i v Packet Traceru by se uživatelé měli uchýlit tímto směrem. Kladou se zde vyšší nároky na znalosti (nutnost znát příkazy). Za každým příkazem lze použít otazník, který zobrazí nabídku dostupných možností rozšiřujících příkaz a jejich krátký popis. Prostřednictvím GUI se

nastavují parametry např. bezdrátového směrovače, což odpovídá běžné implementaci v domácnostech. Uživatel je ochráněn před složitostmi CLI a vyplňuje parametry do předpřipravených kolonek. Koncová zařízení (PC, server atd.) obsahují okno Desktop, kde se jednotlivé položky shodují s aplikacemi běžně dostupnými na reálných zařízeních včetně IP konfiguračního okna.

Osazování modulů

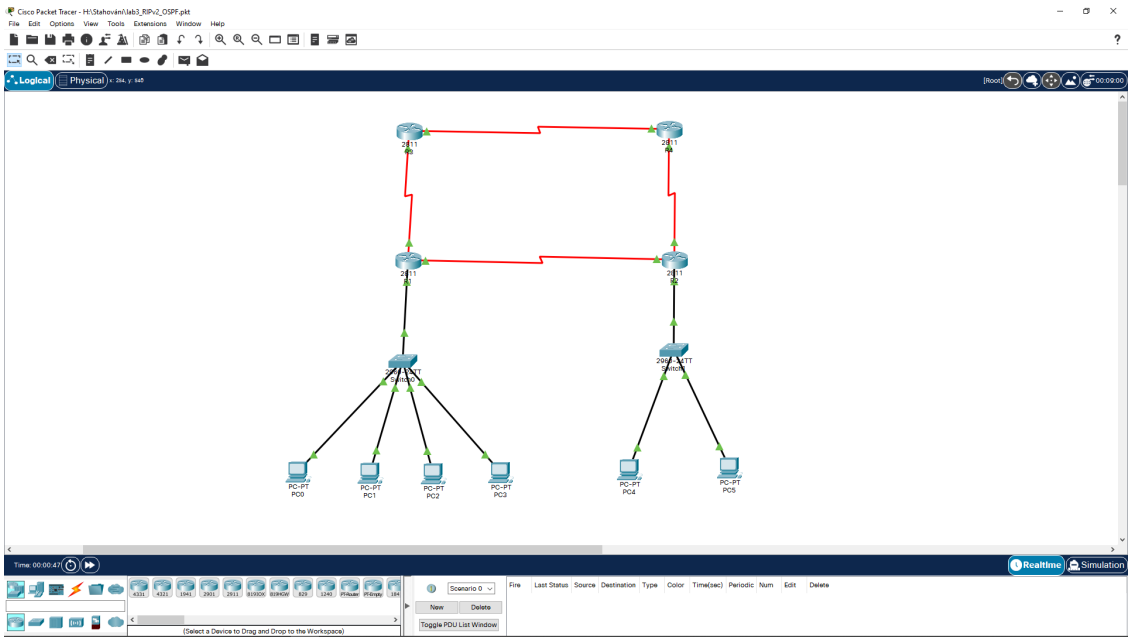
Není výjimkou, že se v praxi k směrovačům pořizují moduly a k počítačům síťové karty (např. s rozhraním pro optiku). V Packet Traceru lze tyto úpravy provádět v okně **Physical**. Zde se zobrazuje fyzické zařízení, na němž se nachází tlačítko napájení, jednotlivé moduly a síťové karty. V závislosti na typu zařízení se mění nabídka dostupných modulů. Pozor, upravovat moduly je možné pouze s vypnutým zařízením. Zařízení v programu již obsahují předpřipravené moduly. Pokud uživatel potřebuje více fyzických portů nebo přidat jiný typ rozhraní (např. sériové nebo optické), stačí si vybrat odpovídající modul a vložit jej do zařízení.

Hierarchie příkazové řádky

Cisco IOS poskytuje více úrovní výzev (prompt), kde každá zpřístupňuje jiné příkazy. Nejnižší úroveň *User EXEC mode* se identifikuje výzvou >, která dává přístup k základním monitorovacím utilitám (např. ping) a umožňuje zobrazit stav zařízení. Na vyšší úrovni je postaven *Privileged EXEC mode* značený výzvou #. Tento mód by měl být vždy chráněn heslem, protože umožňuje přístup ke všem příkazům a módům, zobrazit konfiguraci zařízení a informace o hardwaru. Odtud se přechází do globálního konfiguračního módu, kde probíhají veškeré konfigurační úkony [9]. Obr. 1.5 zachycuje prostředí programu.

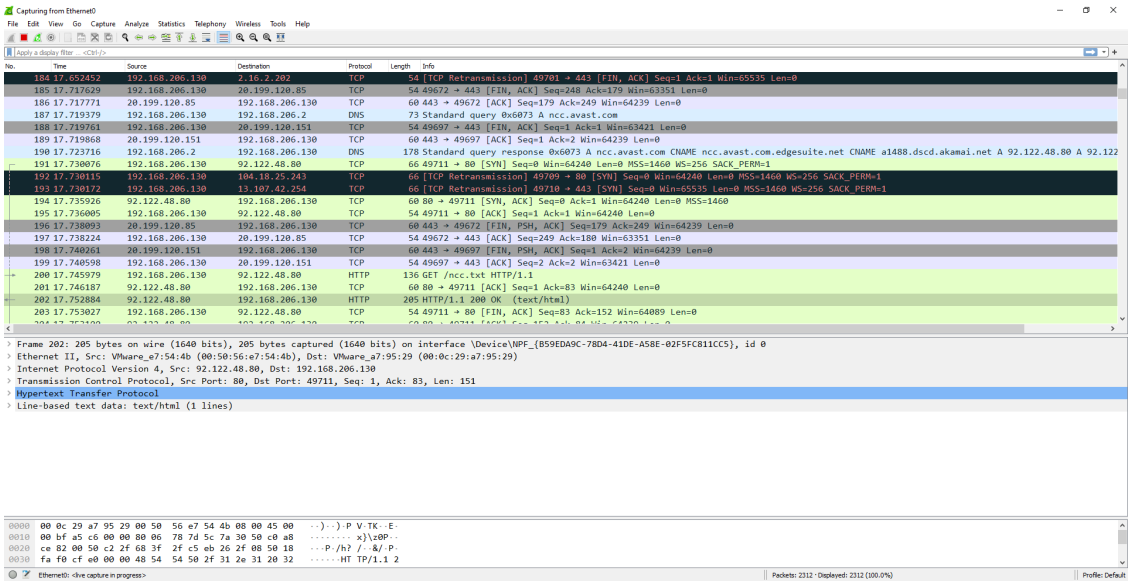
1.3 Wireshark

Program Wireshark nespadá do kategorie simulátorů ani emulátorů. Jedná se o open source síťový analyzátor, kterým lze zachytávat komunikaci na fyzické nebo virtuální síťové kartě. Při spuštění uživatel zvolí rozhraní, na němž chce sledovat průběh komunikace a následně se zobrazují jednotlivé příchozí a odchozí pakety. Hodnoty jednotlivých paketů se řadí do následujících sloupců: číslo zachyceného paketu, časové razítko paketu, zdrojová a cílová adresa, protokol, délka paketu a bližší informace. Po rozkliknutí konkrétního paketu lze v dolní polovině okna vidět detaily o použitých protokolech, které se podobají struktuře vrstvového modelu TCP/IP. Konkrétně se jedná o přenosovou technologii na vrstvě síťového rozhraní (často Ethernet a s ním spojené MAC adresy) a protokol na síťové vrstvě (IPv4, IPv6), kde se zobrazuje



Obr. 1.5: Packet Tracer.

zdrojová a cílová logická adresa. V závislosti na protokolu, který se má přenést, se dále popisuje protokol na úrovni transportní vrstvy (UDP/TCP), pokud je třeba k přenosu, a použité porty. V poslední části se zobrazuje přenášený protokol, tedy aplikační protokol (např. DNS, HTTP atd.) nebo protokol jiné vrstvy (např. na síťové vrstvě ICMP, IGMP atd.). Jednotlivé řádky (vrstvy) je možné rozbalit a zobrazit tak podrobné informace uvnitř záhlaví (např. příznaky) a data [10]. Grafické rozhraní je možné vidět na Obr. 1.6.



Obr. 1.6: Hlavní okno programu Wireshark.

1.4 Shrnutí

Pro simulaci počítačových sítí je možné využít 2 druhy softwaru: simulátory a emulátory. Simulátory představují matematickou analýzu, která názorně zobrazuje komunikaci v sítích, reakci na nestandardní stavy apod. Síť vytvořenou tímto stylem nelze propojit se skutečnou sítí. Simulátory se tedy s výhodou využijí při návrhu sítě od úplného počátku, k testování konfigurací atp. Emulátory vytváří virtuální síť, jež může být připojena do reálné sítě. Prostřednictvím koncových stanic a mezilehlých uzlů lze vygenerovat provoz, který se odesílá na internet nebo k fyzickým zařízením připojeným do existující sítě. S pomocí emulátorů lze testovat zavedení nových technologií do skutečné sítě a sledovat, jaký dopad to bude mít na vytížení sítě.

Pro výukové účely a realizaci laboratorních úloh této práce je výhodný program Packet Tracer. Studenti si zde mohou vyzkoušet konfiguraci Cisco zařízení, často využívaných vzhledem ke své kvalitě, výkonu a možnostem na páteřních spojích poskytovatelů a ve firmách všech rozsahů pro svou bezpečnost a spolehlivost. Jednoduchost práce spočívá v rozdílných režimech, kde jsou uživatelé schopni sledovat přenos v reálném čase, ale i postupné kroky komunikace včetně zobrazení obsahu přenášených dat.

Tab. 1.1 shrnuje vybrané příklady známých dostupných simulátorů a emulátorů. U každého příkladu se uvádí, zda se jedná o simulátor či emulátor, jestli spadá pod open source licenci a zda je kromě verze zdarma dostupná i placená verze, která rozšiřuje funkcionalitu softwaru. Dále se vyznačuje možnost externí analýzy za pomoci síťových analyzátorů.

Tab. 1.1: Přehled dostupných simulátorů a emulátorů.

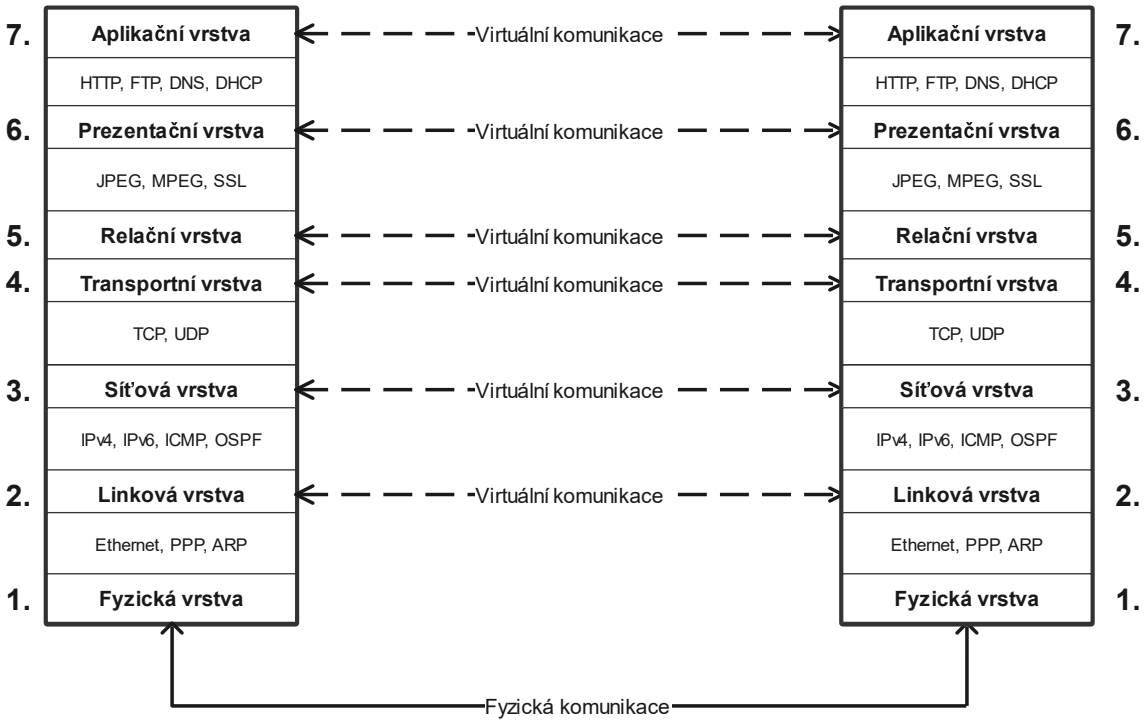
Software	**Open source**	**Placená verze**	**Simulátor**	**Emulátor**	**Externí analýza**
ns-3	X		X		X
EVE-NG		X		X	X
NetSim		X	X		X
GNS3	X			X	X
Packet Tracer			X		

2 Referenční model ISO/OSI

Na samotném počátku sítí spolu vzájemně mohla komunikovat pouze zařízení stejných výrobců. Na základě tohoto poznatku se začaly utvářet ideje o vytvoření síťové architektury, jejíž implementace by umožnila komunikaci nezávislou na výrobních podnicích.

V 70. letech 20. století zahájila organizace ISO (International Organization for Standardization – Mezinárodní organizace pro normalizaci) práci na referenčním modelu OSI (Open Systems Interconnection), který byl přijat jako mezinárodní norma ISO 7498 v roce 1984. Obecně lze referenční model ISO/OSI chápat jako systém síťových protokolů, který představuje soustavu názorů na fungování počítačových sítí. Skládá se ze 7 vrstev, přičemž každá vrstva má definovanou vlastní funkčnost a sadu protokolů [11].

2.1 Princip vrstvového modelu

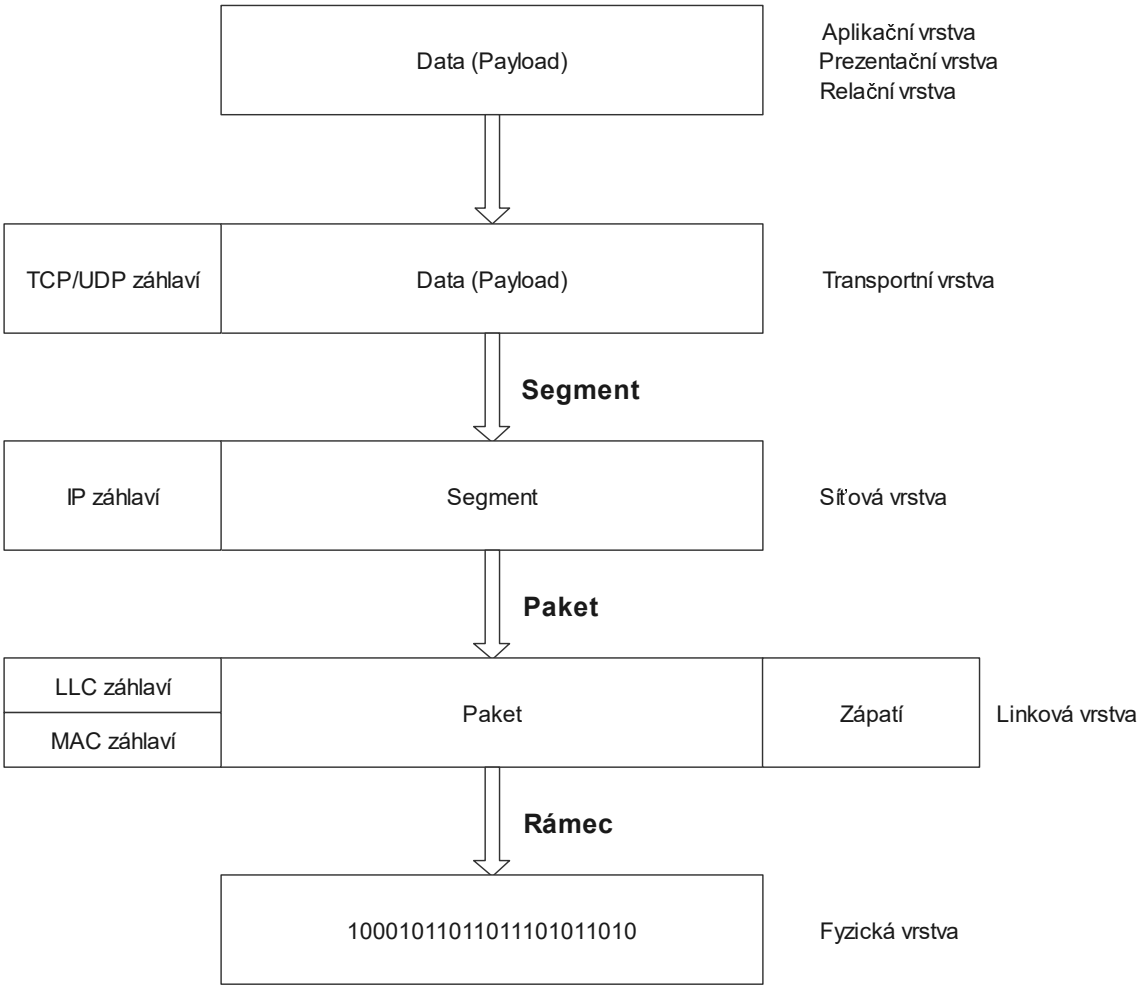


Obr. 2.1: Referenční model OSI/OSI.

Na straně vysílače probíhá komunikace od nejvyšší vrstvy (aplikační) po nejnižší vrstvu (fyzickou), následně se data odesílají přes fyzické médium a příjemce data zpracovává od nejnižší vrstvy po nejvyšší. Tímto způsobem se realizuje fyzická

komunikace (fyzický tok dat). Virtuální komunikace (logický tok dat) se uskutečňuje mezi stejnolehlými vrstvami na straně příjemce a odesílatele (viz Obr. 2.1). Pravidla pro virtuální komunikaci se definují v protokolech. Jednotlivé vrstvy jsou poskládány hierarchicky a mohou komunikovat pouze se sousední vyšší či nižší vrstvou, přičemž vrstva nižší poskytuje služby vyšší vrstvě. Pravidla pro komunikaci sousedních vrstev zahrnuje definice rozhraní mezi danými vrstvami.

2.2 Popis jednotlivých vrstev ISO/OSI



Obr. 2.2: Datové jednotky vrstev ISO/OSI (PDU) [12].

Nejvyšší 3 vrstvy (*aplikační, prezentační, relační*) se zabývají daty aplikací na koncových zařízeních. *Transportní vrstva* přizpůsobuje komunikaci dle požadavků vyšších vrstev. Nejnižší 3 vrstvy (*síťová, linková, fyzická*) se orientují na reálný přenos dat přes síť a implementují se do mezilehlých prvků. Každá vrstva zapouzdřuje data vyšší vrstvy do své definované datové jednotky, tzv. PDU (Protocol Data Unit).

PDU transportní vrstvy se nazývá segment, PDU síťové vrstvy paket, PDU linkové vrstvy rámec a na fyzické vrstvě se přenáší samotné bity (viz Obr. 2.2).

Aplikační vrstva představuje rozhraní mezi aplikacemi a nižšími vrstvami. Zajišťuje správné formátování předávaných a přebíraných dat aplikačních programů [13]. Vrstva řeší komunikaci aplikací přes datové sítě, neřeší tedy samotný přenos. Jednotkou přenosu je zpráva.

Prezentační vrstva zodpovídá za správný význam dat na straně příjemce i odesílatele. Řeší tedy komprimaci, dekomprimaci, zabezpečení (šifrování, zajištění integrity dat, digitální podpis) a kódování ve smyslu prezentace symbolů v abecedách.

Relační vrstva se stará o zřizování, udržování a ukončování relací mezi dvěma komunikujícími stranami. Vrstva logicky odděluje aplikační data jednotlivých relací, tedy např. při přístupu na více webových stránek dochází k separaci požadavků pro každou stránku. Jednotkou přenosu je relační paket.

Transportní vrstva zajišťuje spolehlivý a spojovaný, nebo nespolehlivý a nespojovaný charakter služeb. V prvním případě se používá protokol TCP (vytváří se logické spojení mezi komunikujícími koncovými uzly), v druhém případě protokol UDP. Protokol se volí v závislosti na komunikující aplikaci. Pro rozlišení jednotlivých procesů se využívají čísla portů. Porty se dělí do 3 kategorií v rozmezí čísel 0–65535:

- Známé porty (*Well-known ports*) – 0–1023 – Porty pro zpracování požadavků široce používaných aplikačních služeb na serverové straně.
- Registrované porty (*Registered ports*) – 1024–49151 – Porty registrované pro komunikaci klientů i serverů.
- Dynamické porty (*Dynamic ports*) – 49152–65535 – Dynamicky přiřazované porty na klientské straně.

Kompletní seznam využití portů lze nalézt v [14]. Data vyšších vrstev se zapoudřují do tzv. segmentů.

Síťová vrstva vyhledává nejvhodnější cestu k cíli. Výstupem může být síť připojená k jinému rozhraní na daném zařízení, případně údaje k dosažení následujícího uzlu, přes nějž se předpokládá dostupnost sítě. Proces vyhledávání se nazývá směrování (*routing*). Existují 2 způsoby směrování: *statické* a *dynamické*. K adresaci se využívají logické IP adresy. V dnešní době se lze setkat s IPv4 adresami (32 bitů) a s IPv6 adresami (128 bitů). IPv6 adresy se zavedly z důvodu vyčerpání IPv4 adres, protože se objevuje stále více zařízení a technologií vyžadujících internetovou

komunikaci. Síťová vrstva formuje data vyšších vrstev do tzv. paketů.

Linková vrstva formátuje data do bloků nazývaných rámce (*frames*). Slouží pro komunikaci zařízení, která spolu přímo sousedí (*point-to-point* nebo *point-to-multipoint*), tedy nachází se v jedné síti a nevyžadují použití prvku na úrovni síťové vrstvy. K adresaci se využívá 48bitová fyzická MAC adresa (technologie Ethernet), jejíž hodnota se mění při každém přechodu přes směrovač. Linková vrstva se dále dělí na 2 podvrstvy: MAC (*Media Access Control*) a LLC (*Logical Link Control*). MAC podvrstva definuje, jakým způsobem jsou data vkládána na médium (přístupové metody, např. CSMA/CD), zajišťuje adresování a vytváří rámce zapouzdřením paketů a přidáním záhlaví a zápatí. Dále se stará o detekci chyb vzniklých při přenosu. LLC podvrstva identifikuje protokol vyšší vrstvy (síťové) a řídí tok dat.

Fyzická vrstva zajišťuje vlastní přenos dat ve formě bitů. Zabývá se vlastnostmi signálu v závislosti na použitém fyzickém médiu (elektrické, optické nebo bezdrátové). Definuje tedy napěťové úrovně přenášených signálů, kódování, modulace, reálnou přenosovou rychlost aj. Vrstva se nestará o význam jednotlivých bitů ani bitových sekvencí.

2.3 ISO/OSI v praxi

V praxi se model ISO/OSI neuplatnil pro svou složitost implementace, slouží spíše jako teoretický model pro pochopení principu komunikace a pro programátory. Model předpokládá zejména spolehlivý a spojovaný charakter služeb, což dnešním požadavkům příliš neodpovídá. Místo něj se používá model TCP/IP, který oproti modelu OSI/OSI obsahuje pouze 4 vrstvy. *Aplikační, prezentační* a *relační vrstvu* slučuje do aplikační vrstvy, *transportní* a *síťová* vrstva zůstávají samostatné. *Linkovou* a *fyzickou* vrstvu model TCP/IP slučuje do *vrstvy síťového rozhraní*. Model dále předpokládá nespojovaný charakter služeb a spolehlivost vnímá jako problém komunikujících stran [11].

3 Směrování

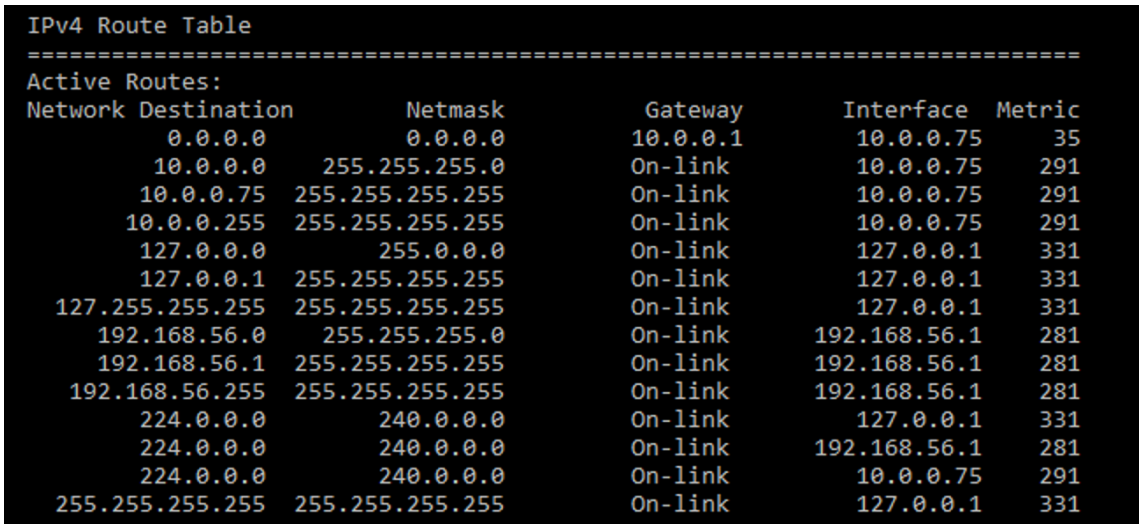
Směrování označuje proces vyhledávání nejlepší (nejefektivnější) cesty od zdroje k cíli, který probíhá na síťové vrstvě referenčního modelu ISO/OSI (TCP/IP). Využívají se tedy **logické adresy** (IP). Dominantním se dnes stalo směrování v oblasti přepojování paketů, vyskytuje se ale i ve všech ostatních způsobech přenosů (přepojování okruhů, buněk a zpráv). Službu vyhledávání cesty provádí mezilehlá zařízení pracující na síťové vrstvě. Od počátků komunikačních sítí to byly směrovače, vývojem technologií se přidaly přepínače vrstvy 3. [15].

3.1 Směrovací tabulka

Při rozhodování o nejlepší cestě se využívá směrovací tabulka, která obsahuje statické a dynamické záznamy. Každé zařízení udržuje vlastní tabulku, do níž nahlíží při komunikaci. Záznamy se prezentují ve formě matice, kde řádek představuje jeden záznam a sloupce konkrétní hodnoty. Ve standardních systémech koncových zařízení (např. Windows) se záznam skládá z následujících informací (viz Obr. 3.1):

- **IP adresa cílové sítě** – Logická adresa sítě, do jejíhož rozsahu (určeno podle síťové masky) spadá cílová adresa obsažená v IP paketu.
- **Síťová maska** – Rozhodující parametr při výpočtech rozsahu, do kterého spadá adresa cílové stanice. Síťová maska obsahuje zleva nepřetržitý řetězec bitů rovných hodnotě log. 1 pro část adresy sítě. Masku lze vyjádřit prefixem, jehož hodnota vyjadřuje počet bitů s hodnotou jedna. Zbylé bity (část pro stanice) se rovnají hodnotě log. 0. Záznamy se řadí podle hodnoty masky od nejobecnějších (nejmenší prefix) po nejkonkrétnější (největší prefix).
- **Výchozí brána** – Logická adresa směrovače, který je prvním hopem (označení pro směrovač v terminologii směrování) na cestě k cíli.
- **Síťové rozhraní** – Lokální rozhraní, přes nějž se paket odesílá pro dosažení cíle.
- **Metrika** – Rozhodující parametr při volbě nejlepší cesty, pokud lze cíle dosáhnout více cestami. Jedná se o dekadickou hodnotu vyjadřující zhodnocení cesty na základě příslušných algoritmů (liší se pro jednotlivé směrovací protokoly). Nižší hodnota metriky vyjadřuje lepší cestu.

Specifickým záznamem ve směrovací tabulce je **výchozí cesta**. Skládá se z totožných hodnot pro adresu sítě i masku. Všechny bity se rovnají hodnotě log. 0, tedy dekadicky vyjádřeno zápisem *0.0.0.0*. Záznamy se v tabulce prohledávají od nejkonkrétnějších po nejobecnější. V případě, kdy není nalezen odpovídající záznam,



```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         10.0.0.1        10.0.0.75     35
         10.0.0.0    255.255.255.0         On-link         10.0.0.75    291
        10.0.0.75  255.255.255.255         On-link         10.0.0.75    291
       10.0.0.255  255.255.255.255         On-link         10.0.0.75    291
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
     192.168.56.0    255.255.255.0         On-link       192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link       192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link       192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link       192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         10.0.0.75    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
```

Obr. 3.1: Směrovací tabulka systému Windows [16].

využije se tzv. výchozí cesta[1]. Ta slouží pro odeslání všech paketů, o jejichž cílové adrese neexistuje v tabulce záznam. Výchozí cesta nalezne využití i v případě snížení počtu záznamů ve směrovací tabulce. Typickým příkladem je tzv. **stub network** (koncová síť, která se propojuje s jinými sítěmi právě jedním spojem). Hraniční směrovač (přes nějž se propojení realizuje – typicky domácí směrovač) nepotřebuje znát adresy všech cílových sítí. Všechna data odesílá na sousední směrovač (typicky směrovač poskytovatele internetového připojení), tudíž stačí jediný záznam – záznam o výchozí cestě. Odeslání dat správným směrem následně zajišťuje poskytovatel. Tímto způsobem dojde k výraznému snížení počtu záznamů ve směrovací tabulce a zároveň se urychlí proces směrování (prohledává se méně záznamů).

Poněkud odlišně vypadá směrovací tabulka na síťových prvcích. Dále se popisuje struktura na Cisco směrovačích (viz Obr. 3.2):

- **Kód** – Určuje, jakým způsobem se vytvořil záznam ve směrovací tabulce. Následující příklady ukazují některé z dostupných kódů:
  - **C** – Přímo připojená síť.
  - **S** – Statický záznam zadaný administrátorem.
  - **R** – Dynamický záznam vytvořený na základě protokolu RIP.
  - **O** – Dynamický záznam vytvořený na základě protokolu OSPF.
  - **D** – Dynamický záznam vytvořený na základě protokolu EIGRP.
- **Cílová síť s prefixem**
- **Údaje o cestě** – Údaje se zobrazují ve formátu [**X/Y**], kde X značí administrativní vzdálenost a Y představuje metriku. Příklady administrativních

[1]Záznam o výchozí cestě nemusí být přítomen, v takovém případě se paket zahodí.

vzdáleností jsou uvedeny v Tab. 3.1.

Tab. 3.1: Administrativní vzdálenosti.

Administrativní vzdálenost	Směrovací protokol
0	Přímo připojená síť
1	Staticky nakonfigurovaná cesta
90	EIGRP
110	OSPF
120	RIP

- **Následující hop** – Udává adresu sousedního směrovače, přes nějž se data vysílají k cíli.
- **Doba existence záznamu** – Čas [s] od poslední aktualizace záznamu.
- **Výstupní rozhraní** – Označení lokálního výstupního rozhraní, ze kterého se data odesílají na sousední směrovač (směr k cíli).

```
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 6 subnets, 2 masks
R       172.20.2.0/30 [120/1] via 172.20.2.5, 00:00:20, Serial0/3/1
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.6/32 is directly connected, Serial0/3/1
C       172.20.2.8/30 is directly connected, Serial0/3/0
L       172.20.2.10/32 is directly connected, Serial0/3/0
R       172.20.2.12/30 [120/1] via 172.20.2.9, 00:00:21, Serial0/3/0
```

Obr. 3.2: Cisco směrovací tabulka.

3.2 Typy směrování

Z pohledu konfigurace se směrování dělí do dvou základních skupin – statické a dynamické. Záznamy o přímo připojených sítích se do směrovací tabulky přidávají automaticky.

3.2.1 Statické směrování

Principem statického směrování je manuální zadávání cest do směrovací tabulky. Administrátor tedy musí znát celou topologii sítě a zvolit nejvhodnější trasy pro pakety. Mezi výhody se řadí jednoduchost konfigurace v případě malých sítí. Dále nezatěžuje výpočetní prostředky směrovacích prvků (statický záznam se bez zásahu administrátora nemění a samotné zařízení nepřepočítává cesty) a nevyužívá část šířky pásma pro zasílání směrovacích informací. Nevýhody se objevují s rostoucí topologií sítě. Počet záznamů ve směrovacích tabulkách se zvětšuje[2] a vzhledem ke skutečnosti, že se trasy nastavují na všech směrovačích (obecně zařízeních provádějících operaci směrování), existuje zde jistá pravděpodobnost vzniku chyby lidského faktoru. Velká nevýhoda spočívá v potřebě upravit, při jakékoli změně v topologii/adresování, všechny záznamy, jichž se změna týká. U větších sítí se může jednat o časově velmi náročnou operaci. Významným příkladem statického záznamu je výchozí cesta.

3.2.2 Dynamické směrování

Dynamické směrování přináší značné množství výhod nad statickým směrováním, zároveň se objevuje i řada nevýhod. Základem je zprovoznění **dynamického směrovacího protokolu** na všech směrovačích, které se účastní procesu směrování uvnitř směrovací domény. Mezi přednosti dynamického směrování se řadí minimalizace zásahu administrátora do konfigurace po správném nastavení všech prvků. Směrovací protokoly se starají o šíření informací (adresa sítě, maska sítě, metrika) a reagují odpovídajícím způsobem na změnu v síti (objevení nové sítě, výpadek sítě/linky). Dynamické směrování se využívá ve větších sítích. Mezi nevýhody lze zařadit nároky na výpočetní výkon (využití procesoru k výpočtům nejlepší cesty, přepočítávání v případě výpadku a zpracování směrovacích informací), které se projevují výrazněji u výkonově slabších zařízení, a spotřeba části šířky pásma pro šíření směrovacích informací. V porovnání se statickým směrováním je dynamické směrování méně bezpečné, protože při výpadku může být zvolena nechtěná alternativní cesta, případně lze podvrhnout falešné směrovací informace, což může vést až k výpadku sítě. V současnosti se často aplikuje kombinace statického a dynamického směrování [17].

[2] V případě vhodně rozděleného adresního prostoru lze využít tzv. **sumarizaci**, která nahrazuje určitý počet záznamů ve směrovací tabulce jediným záznamem. K tomu se využije nejnižší adresa sítě a vhodně zvolená maska zahrnující prostor všech sumarizovaných sítí.

3.3 Směrovací protokoly

Dynamické směrovací protokoly se využívají ve větších sítích pro komunikaci mezi směrovači. Ty si předávají směrovací informace a vytváří z nich záznamy ve svých směrovacích tabulkách. Informace se přenáší v pravidelných intervalech, u novějších protokolů se upřednostňuje zasílání v případě změny v topologii sítě. Základní operace směrovacích protokolů lze rozdělit do následujících bodů:

- Odesílají se informace o přímo připojených sítích směrovačů. Prostřednictvím této operace se směrovače dozvídají o sítích, s nimiž nejsou fyzicky propojeny.
- Výpočet nejlepší cesty k cíli, jež je určena **nejnižší hodnotou metriky**.
- V případě změny v topologii sítě, což představuje připojení nové sítě, odpojení stávající sítě, přerušení spoje, výpadek směrovacího prvku atp., se automaticky přepočítá nová nejlepší cesta k cíli.

K účelu zpracování směrovacích informací a výpočtu nejlepší cesty k cíli slouží **směrovací algoritmy**. Každý směrovací protokol obsahuje sadu zpráv, pomocí kterých se komunikuje se sousedními směrovači (na nichž je aktivní stejný směrovací protokol), odesílají se směrovací informace a informuje se o změně topologie [17].

3.3.1 Autonomní systém

V současné době internetu, kdy existuje extrémní počet sítí, není pro směrovače možné udržovat informace o všech sítích na světě. Z tohoto důvodu se využívá koncept **autonomních systémů** (zkratka AS (Autonomous System)). Jedná se o oblast směrovačů a sítí, které se nachází pod společnou správou organizace (např. ISP, vysoká škola, rozsáhlá společnost atd.). V rámci autonomního systému se využívá jednotná směrovací politika (implementace společného směrovacího protokolu[3]). Dělí se do tří základních kategorií:

- **Stub AS** – Propojený s právě jedním AS.
- **Transit AS** – Slouží k propojení více AS a zároveň k přenosu dat mezi nimi.
- **Multihomed AS** – Propojený s více AS, ovšem odesílá pouze svá data, neslouží k přesunu dat mezi ostatními AS [18].

Autonomním systémům se přiřazují globálně unikátní šestnáctibitová[4] čísla organizací IANA (Internet Assigned Numbers Authority). Tato čísla se označují jako ASN (Autonomous System Number). Nabývají tedy hodnot 0–65535, kde hodnoty 1–64511 se přiřazují pro směrování na internetu a hodnoty 64512–65535 se využívají k privátním účelům [19], [20].

[3]Existují situace, kdy lze uvnitř autonomního systému využít více směrovacích protokolů pro různé části sítě [17].

[4]Od roku 2007 se přešlo na 32-bitová čísla z důvodu neustálého nárustu počtu autonomních systémů [22].

S konceptem autonomních systémů se směrovací protokoly rozdělily do dvou hlavních skupin – **interní směrovací protokoly** a **externí směrovací protokoly**.

3.3.2 Interní směrovací protokoly

Slouží ke směrování uvnitř autonomních systémů [21]. Organizace si zvolí protokol dle svého uvážení a ten následně implementuje na směrovače. Po dosažení stavu konvergence[5] existují ve směrovací tabulce každého směrovače záznamy, které umožňují komunikaci s libovolnou sítí v rámci autonomního systému. Interní směrovací protokoly se dělí do dvou podskupin – **protokoly vektoru vzdáleností** a **protokoly stavu linek**.

Protokoly vektoru vzdáleností

Jedná se o prvotní návrh principu směrovacích protokolů. Již z názvu lze odvodit, že klíčovým prvkem je vektor, který se fyzikálně definuje **velikostí** a **směrem**. Velikost zde představuje počet přeskoků (hopů), tedy směrovačů, přes něž paket projde, než se dostane do cíle. Směr určuje sousední směrovač (*next hop*), ke kterému se paket odesílá pro dosažení cíle.

Tato skupina protokolů se zakládá na **Bellman-Fordově algoritmu**. Princip spočívá v distribuovaném výpočtu, kde směrovače přijímají informace o vzdálených sítích a počtu přeskoků k jejich dosažení. Příchozí rozhraní se současně stává odchozím rozhraním pro komunikaci s danými sítěmi. Nejlepší cesta se volí jako cesta s nejnižší metrikou, tedy nejmenším počtem přeskoků. Pokud směrovač přijme informace o síti, k níž už má stanovenou cestu, ovšem nově přijatá metrika je nižší, nahradí se záznam o počtu přeskoků a odchozí rozhraní aktuálnějšími informacemi. Tento proces se opakuje, dokud všechny směrovače v síti nedosáhnou stavu konvergence.

I když směrovače získají potřebné informace pro sestavení kompletní topologie, nevyužívají tuto možnost. Udržují pouze informace o přímo připojených sítích, vzdálených sítích s počtem přeskoků a směru pro dosažení těchto sítí (sousední směrovač / odchozí rozhraní). Mezi směrovací protokoly, které využívají princip vektoru vzdáleností, patří RIPv1 (novější RIPv2) a IGRP (novější EIGRP).

Protokoly stavu linek

Protokoly stavu linek představují novější přístup k fungování směrovacích protokolů. Směrovače si na základě přijatých informací vytváří kompletní mapu síťové topologie

[5]Konvergence vyjadřuje stav, kdy mají všechny směrovače v autonomním systému úplné a správné (nejaktuálnější) informace o topologii sítě.

a z ní vypočítají nejkratší cestu k cíli. Informace (IP adresa a síťová maska, typ sítě, cena linky a ID souseda) se přenáší pomocí LSP (Link State Packet). Každý směrovač vygeneruje informace o všech svých aktivních rozhraních (účastnících se procesu směrování) a odesílá je všem sousedům. Na rozdíl od protokolů vektoru vzdáleností (přepočítají informace – zvýší počet přeskoků o jedna – a ty odesílají dále), když směrovač přijme LSP, zaznamená nové informace do **databáze stavu linek** a ihned paket přeposílá ke všem svým sousedům (kromě souseda, od nějž paket přijal).

Protokoly využívají **Dijkstrův algoritmus**, označovaný též jako SPF (Shortest Path First) algoritmus. Poté, co se dosáhne stavu konvergence[6], se spustí proces výpočtu SPF stromu. Kořen stromu představuje směrovač, na němž se výpočet odehrává a listy stromu tvoří ostatní směrovače. Výsledkem je určení nejlepší cesty k jednotlivým sítím na základě metriky, kterou reprezentuje cena spojů. Tento proces se spouští nezávisle na každém směrovači.

Ve srovnání s protokoly vektoru vzdáleností dosahuje tato skupina rychlejší konvergence (jedno z hlavních kritérií pro výběr směrovacího protokolu) a využívá se v rozsáhlejších sítích. Nevýhodou jsou vyšší nároky na systémové prostředky, zejména paměť (každý směrovač si udržuje informace o celé topologii sítě).

Do této skupiny se řadí protokoly OSPF a IS-IS [21].

3.3.3 Externí směrovací protokoly

Využívají se ke směrování mezi autonomními systémy, tedy systémy pod odlišnou správou. Původně se implementoval dnes již zastaralý protokol EGP (Exterior Gateway Protocol), který v současnosti nahradil protokol BGP (Border Gateway Protocol). Oba protokoly jsou určeny pro směrování na internetu, např. mezi ISP a rozsáhlými organizacemi. Zakládají se na algoritmu **vektoru cest**. Oproti předchozím algoritmům se zde vychází z kompletní analýzy cesty k cíli. V případě, že směrovač přijme záznam o destinaci, v jehož cestě k dosažení není zahrnut, přidá sebe sama a odesílá informace sousedům. Pokud směrovač přijme záznam, v jehož cestě se již vyskytuje, zamítne jej z důvodu předejití směrovacích smyček [23].

[6]Stavu konvergence se dosáhne poté, co směrovač obdrží LSP od všech ostatních směrovačů. V té chvíli má každý směrovač podrobný přehled o topologii sítě a každé zařízení disponuje identickými informacemi pro výpočet nejlepší (nejrychlejší) cesty.

4 Teoretický rozbor použitých protokolů

Tato kapitola se věnuje podrobnému popisu komunikačních protokolů, které jsou použity v navržených laboratorních úlohách.

4.1 Protokol ARP

ARP (Address Resolution Protocol), definovaný v dokumentu RFC 826 [24], funguje na pomezí 2. a 3. vrstvy modelu ISO/OSI, mezi něž lze rozdělit komunikaci.

Komunikuje-li se v lokálních sítích, IP adresa je většinou známá a slouží k jedinečnému určení koncového uživatele, jemuž jsou data určena. Přidáním IP záhlaví (se specifikovanou zdrojovou a cílovou IP adresou) k datům se vytvaří IP paket. Pro samotný přenos ale musí být paket zapouzdřen do rámce v závislosti na použité technologii. V lokálních sítích se jedná převážně o **Ethernet**. Rámce do záhlaví (mimo jiné) přidávají zdrojovou MAC adresu a cílovou MAC adresu, která typicky není známá, pokud se se zařízením dříve nekomunikovalo. Zde přichází na řadu ARP protokol, jehož cílem je namapovat MAC adresu k příslušné IP adrese cílového zařízení a umožnit tak komunikaci v rámci lokální sítě.

4.1.1 ARP tabulka

ARP tabulka představuje místo ve vyrovnávací paměti, kde se ukládají veškeré záznamy získané z ARP komunikace. Tabulku lze na unixových a Windows systémech vypsat příkazem `arp -a`. Příklad výstupu na systému Windows 10 je ukázán na Obr. 4.1.

Tabulka se skládá z následujících položek:

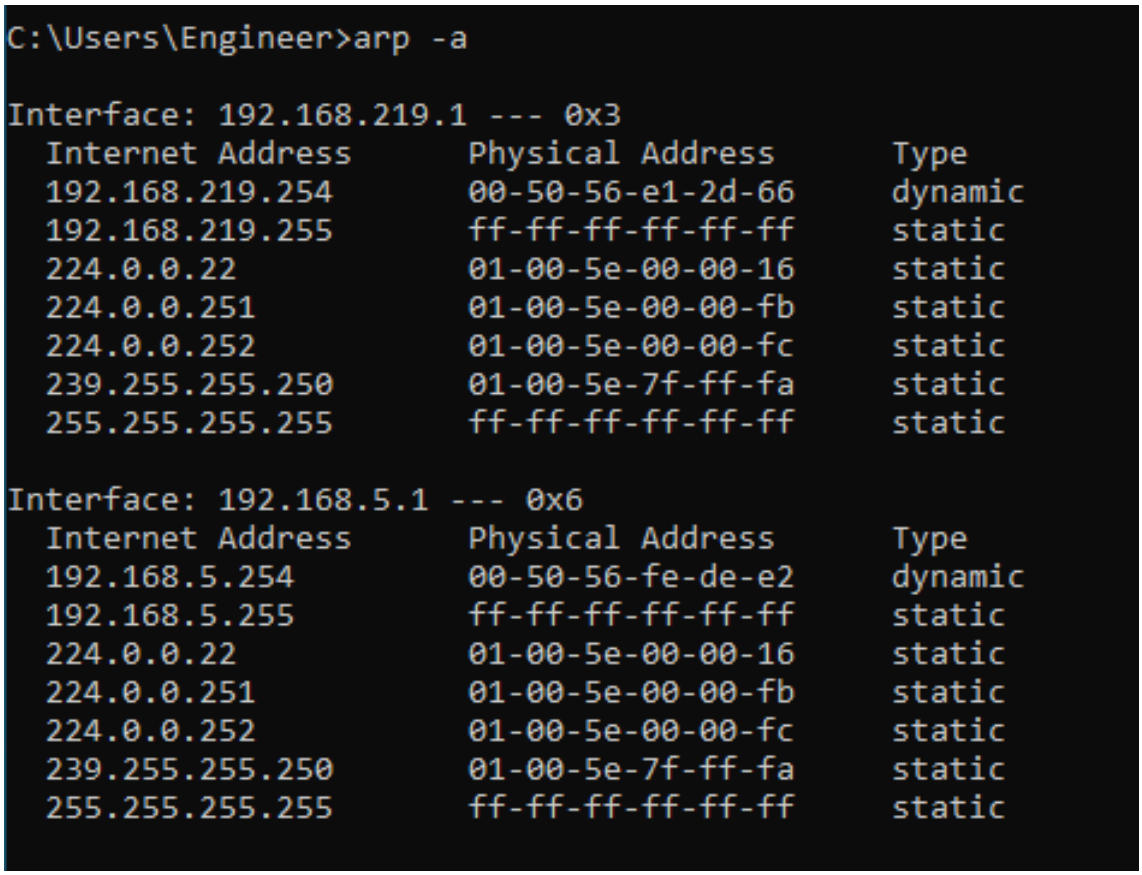
- internetová adresa (*Internet Address*),
- fyzická adresa (*Physical Address*),
- typ záznamu (*Type*).

Různé operační systémy (včetně síťových) mohou zobrazovat i další informace, např. rozhraní, na němž byl ARP paket zpracován nebo tzv. timeout, což označuje dobu platnosti, po jejímž vypršení dojde ke smazání záznamu z paměti.

Internetová adresa

Na úrovni 3. vrstvy (síťové) probíhá adresace pomocí logických IP adres verze 4 (32 bitů) nebo 6[1] (128 bitů). IPv4 adresy se zapisují dekadicky, IPv6 adresy hexadecimálně. Tyto adresy slouží jako jedinečný identifikátor koncových uživatelů

[1] ARP se využívá při IPv4 adresaci. Protokol IPv6 využívá protokol NDP (Neighbor Discovery Protocol), jehož funkce je obdobná.



Obr. 4.1: Obsah ARP tabulky.

a skládají se ze dvou částí: *síťová* a *uživatelská*. Rozlišují se 3 typy zpráv podle kritéria, komu jsou určeny:

- **Unicast** – Zpráva určená jedné koncové stanici.
- **Multicast** – Zpráva určená skupině koncových stanic, které se identifikují konkrétní adresou z multicastového rozsahu.
- **Broadcast** – Zpráva určená všem koncovým stanicím, které se nachází ve stejné síti jako zdroj.

Adresy se přidělují manuálně (tzv. statické adresy) nebo dynamicky. Statické adresy se využívají u směrovačů na hraně lokální sítě, pro virtuální rozhraní přepínačů, tiskárny a servery, tedy u zařízení, u nichž se předpokládá, že nemění svou polohu a klienti k nim přistupují. Pokud by se jejich adresa v průběhu času měnila, mohlo by dojít ke značným potížím z hlediska nedostupnosti služeb. Dynamické adresy se naopak přidělují zařízením, které nevyžadují trvalou adresu a v průběhu času se mohou odpojovat ze sítě, případně cestovat mezi různými lokacemi (mobil, notebook). Dynamické adresy se přidělují z definovaného rozsahu pomocí **DHCP** (Dynamic Host Configuration Protocol), jemuž se věnuje lab. úloha č. 5. Dále se rozlišují privátní a veřejné adresy. U IPv4 se adresy dělí do 5 tříd (viz Tab. 4.1), z nichž první

3 třídy obsahují prostor privátních adres. V lokálních sítích se komunikuje prostřednictvím privátních adres (viz Tab. 4.2), v prostředí internetu pomocí veřejných adres. Veřejné adresy musí být na celém internetu unikátní. Unikátnost privátních adres platí v rámci samostatné lokální sítě, ovšem různé lokální sítě mohou využívat stejné rozsahy.

Tab. 4.1: Rozdělení adres do jednotlivých tříd.

Třída	**Prefix**	**Maska sítě**	**Počáteční adresa**	**Koncová adresa**
Třída A	8	255.0.0.0	0.0.0.0	127.255.255.255
Třída B	16	255.255.0.0	128.0.0.0	191.255.255.255
Třída C	24	255.255.255.0	192.0.0.0	223.255.255.255
Třída D	X	X	224.0.0.0	239.255.255.255
Třída E	X	X	240.0.0.0	255.255.255.255

Tab. 4.2: Privátní adresy jednotlivých tříd.

Třída	**Prefix**	**Maska sítě**	**Počáteční adresa**	**Koncová adresa**
Třída A	8	255.0.0.0	10.0.0.0	10.255.255.255
Třída B	12	255.240.0.0	172.16.0.0	172.31.255.255
Třída C	16	255.255.0.0	192.168.0.0	192.168.255.255

Fyzická adresa

Na linkové vrstvě se adresuje pomocí MAC adres. Jedná se o fyzické adresy unikátní pro všechna zařízení, které se síťovým kartám přiřazují již při výrobě. Adresa se skládá z celkového počtu 48 bitů, kde prvních 24 bitů reprezentuje konkrétního výrobce, tzv. OUI (Organizationally Unique Identifier – jedinečný identifikátor výrobce), zbylé bity identifikují samotné zařízení. Standardní zápis se provádí ve formátu 6 dvojic hexadecimálních čísel oddělených pomlčkou, dvojtečkou nebo jiným symbolem (nemusí být žádný oddělovač). Prostřednictvím fyzických adres se komunikuje uvnitř lokálních sítí, které využívají standardy IEEE 802.X, kde X značí číslo konkrétního standardu (např. IEEE 802.3 – Ethernet, IEEE 802.11 – WiFi). Speciálním případem je broadcastová MAC adresa (*ff-ff-ff-ff-ff-ff*), která se využívá u protokolu ARP, aby rámec obdržela všechna zařízení na lokální síti. Komunikuje-li stanice s uzlem mimo lokální síť, MAC adresa cíle se nastaví na fyzickou adresu směrovače (obecně zařízení, které pracuje na síťové vrstvě a zajišťuje funkci výchozí brány).

Typ záznamu

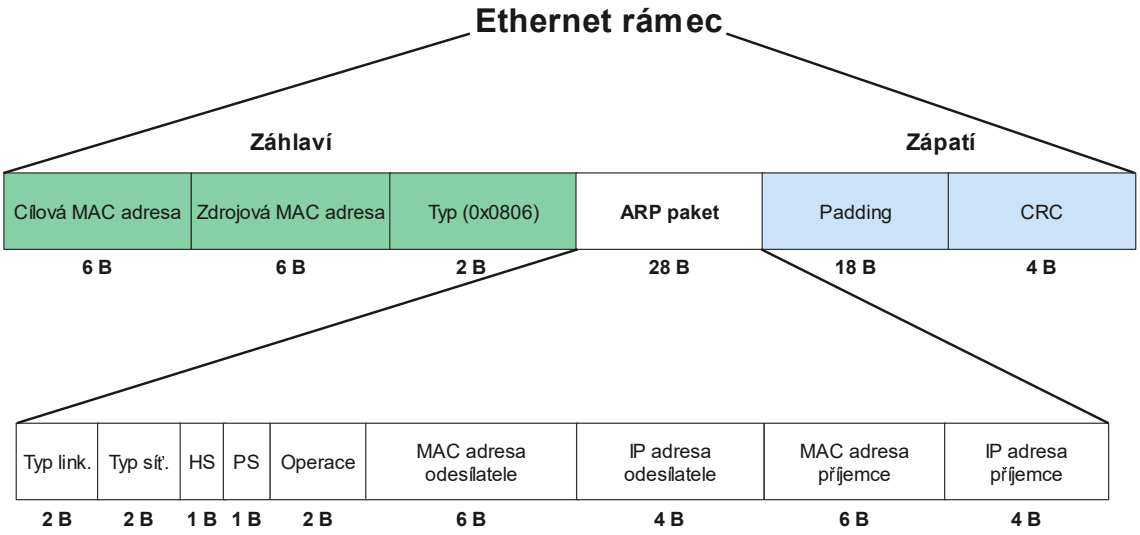
Existují dva typy záznamů: *statické* a *dynamické*.

Dynamické záznamy se vytváří na základě ARP žádostí a odpovědí. Z žádosti (zdrojová IP a MAC adresa) si vytvoří záznam dotazované zařízení, z odpovědi dotazující se zařízení. Dynamické záznamy mají svou životnost v cache paměti (*timeout*), která se liší u jednotlivých výrobců. Často se jedná o dobu dvě minuty, u níž pokud se záznam nevyužije ke komunikaci, automaticky se vymaže z tabulky. V případě aktivity dojde k resetování časovače. Aktivní záznamy mají maximální životnost 10 minut. Mazání záznamů probíhá z důvodu bezpečnosti, protože v průběhu času může dojít ke změně logické adresy jednoho z komunikujících zařízení a původní adresa se stane neaktivní nebo se přidělí jinému zařízení.

Statické záznamy vkládá uživatel ručně v následujícím formátu: `arp -s IP MAC` (např. `arp -s 192.168.0.1 ab-17-cd-18-ee-23`). U statických záznamů se nepočítá timeout, v tabulce zůstanou, dokud nedojde k ručnímu vymazání tabulky (příkaz `arp -d *`) nebo k restartu celého systému. Statický záznam se výhodně využije např. při komunikaci se serverem, jehož logická adresa zůstává v průběhu času neměnná. Přesto, že ARP provoz není z pohledu sítě příliš náročný, dojde ke snížení režie ARP dotazů a odpovědí a ušetřená kapacita se může využít pro jiný typ provozu.

4.1.2 Struktura ARP paketu

Vznikne-li požadavek na zjištění cílové MAC adresy, vytvoří se ARP paket, který se zapouzdří do ethernetového rámce (viz Obr. 4.2).



Obr. 4.2: ARP paket zapouzdřený v ethernetovém rámci [13], [25].

V ethernetovém rámci se vyskytují následující položky:

- **Cílová MAC adresa** – Obsahuje MAC adresu cíle. V případě ARP žádosti se nastaví na broadcastovou MAC adresu, kde jsou všechny bity rovny hodnotě log. 1 (*ff:ff:ff:ff:ff:ff*) z důvodu neznámé konkrétní MAC adresy cíle. Rámec tak obdrží všechna zařízení v lokální síti. V případě ARP odpovědi se nastaví na MAC adresu původního zdroje.
- **Zdrojová MAC adresa** – Obsahuje MAC adresu zdroje, tedy zařízení, které vysílá rámec.
- **Typ** – Hodnota, která vyjadřuje, jaký protokol je do rámce zapouzdřen. V případě ARP protokolu se jedná o hexadecimální hodnotu *0x0806*. Další hodnoty mohou být např:
  - *0x0800* – protokol IPv4,
  - *0x86DD* – protokol IPv6,
  - *0x8100* – VLAN rámec s tagem.
- **ARP** – Vnořený ARP paket.
- **Padding** – Výplň dat pro dosažení minimální velikosti ethernetového rámce (64 bytů).
- **CRC** – Kontrolní součet pro ověření bezchybného přenosu.

Vnořený ARP paket obsahuje následující položky:

- **Typ linkového protokolu** – Specifikuje technologii na úrovni linkové vrstvy pro odeslání ARP paketu – převážně Ethernet (hodnota 1).
- **Typ síťového protokolu** – Využívají se stejné hodnoty jako u pole **Typ** v ethernetovém rámci. Protokol IPv4 má přidělenou hodnotu *0x0800*.
- **HS** (*Hardware Size*) – Délka hardwarové (MAC) adresy.
- **PS** (*Protocol Size*) – Délka síťové (IP) adresy.
- **Operace** - Vyjadřuje, zda se jedná o žádost (*ARP request*) nebo odpověď (*ARP reply*). V případě žádosti se nastaví hodnota 1, v případě odpovědi hodnota 2.
- **MAC adresa odesílatele** – Fyzická adresa odesílatele.
- **IP adresa odesílatele** – Logická adresa odesílatele.
- **MAC adresa příjemce** – Fyzická adresa příjemce.
- **IP adresa příjemce** – Logická adresa příjemce [13], [26].

4.1.3 Komunikace ARP protokolu

Následující popis vychází z případu, kdy se síť poprvé uvede do provozu a zařízení neznají své sousedy.

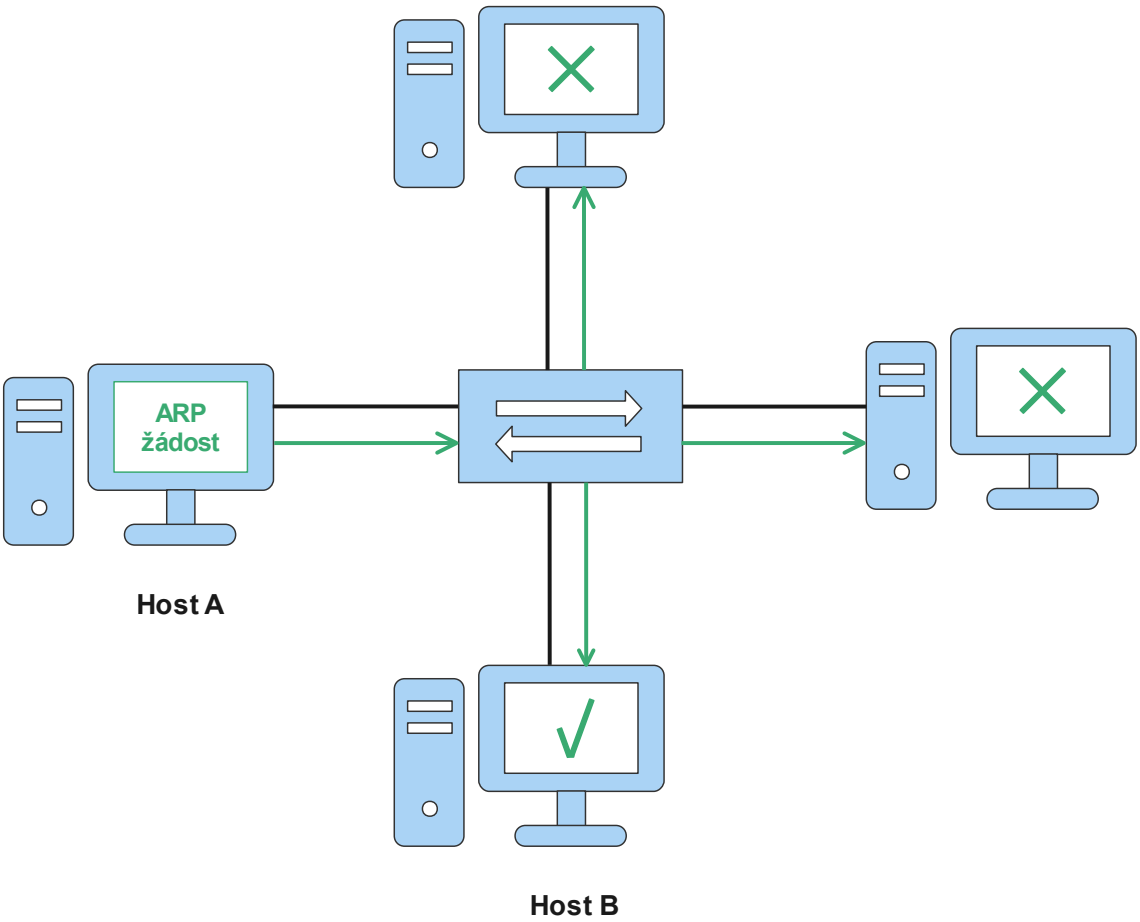
1. Host A, který iniciuje komunikaci s hostem B, prohledá svou ARP cache paměť a zjistí, zda se v ní nachází záznam skládající se z logické adresy hosta B a jeho fyzické adresy. V případě, že takový záznam neexistuje, vytvoří se *ARP žádost* a zapouzdří se do ethernetového rámce.
2. Rámec má nastavenou cílovou MAC adresu na broadcast, všechna zařízení v lokální síti tedy tento paket obdrží a zpracují. Zdrojová MAC adresa se nastaví na fyzickou adresu hosta A.
3. Přepínač obdrží rámec a vytvoří ve své MAC tabulce (*forwarding table*) záznam, který se skládá z příchozího portu a zdrojové MAC adresy v rámci. Následně jej rozešle na všechny porty kromě příchozího.
4. Zařízení, které rámec přijme, extrahuje ARP paket a provede srovnání IP adresy příjemce se svou vlastní. Shoda IP adres nastane pouze u hosta B, který s paketem dále pracuje, ostatní zařízení paket zahodí.
5. Host B prohledá vlastní ARP tabulku a určí, zda v ní existuje záznam se zdrojovu MAC adresou a IP adresou odesílatele rámce. Pokud ne, vytvoří se dynamický záznam.
6. Host B následně vytvoří *ARP odpověď*, kde nastaví MAC adresu a IP adresu odesílatele na svou vlastní. MAC adresu a IP adresu příjemce nastaví na hodnoty hosta A. Dojde tedy k záměně zdrojových a cílových adres z ARP žádosti. ARP paket se zapouzdří do ethernetového rámce, kde se nastaví cílová MAC adresa hosta A a zdrojová MAC adresa hosta B. ARP odpověď se tedy vysílá konkrétnímu zařízení.
7. Přepínač přijme rámec a vytvoří ve své MAC tabulce záznam složený z příchozího portu a zdrojové MAC adresy hosta B. Následně porovná cílovou MAC adresu se záznamy ve své MAC tabulce. Protože adresa hosta A je zde již obsažena, přepínač odesílá rámec pouze na port, který je s adresou provázán.
8. Host A přijme rámec, extrahuje ARP paket a uloží si do své ARP cache paměti MAC adresu odesílatele spolu s IP adresou odesílatele.
9. V tuto chvíli obě komunikující zařízení znají všechny potřebné údaje a výměna zpráv započne [13], [26], [27].

ARP žádost

ARP žádost se zabalí do ethernetového rámce, který se odesílá všesměrově. Všechny stanice v lokální síti rámec obdrží a zpracují. Zařízení pracující na úrovni síťové vrstvy (směrovač, přepínač vrstvy 3) paket obdrží a zpracují, nepřeposílají ho však dále do jiných sítí. Princip komunikace lze vidět na Obr. 4.3.

Hodnoty v ARP paketu měnící se během komunikace:

- **Operace** – Pro žádost nastavena na hodnotu 1.
- **MAC adresa odesílatele** – Shodná hodnota se zdrojovou MAC adresou v záhlaví ethernetového rámce.
- **IP adresa odesílatele** – Logická adresa odesílatele.
- **MAC adresa příjemce** – Všechny bity nastaveny na hodnotu log. 0 (*00:00:00:00:00:00*), protože tato položka musí být teprve zjištěna protokolem ARP.
- **IP adresa příjemce** – Logická adresa příjemce.



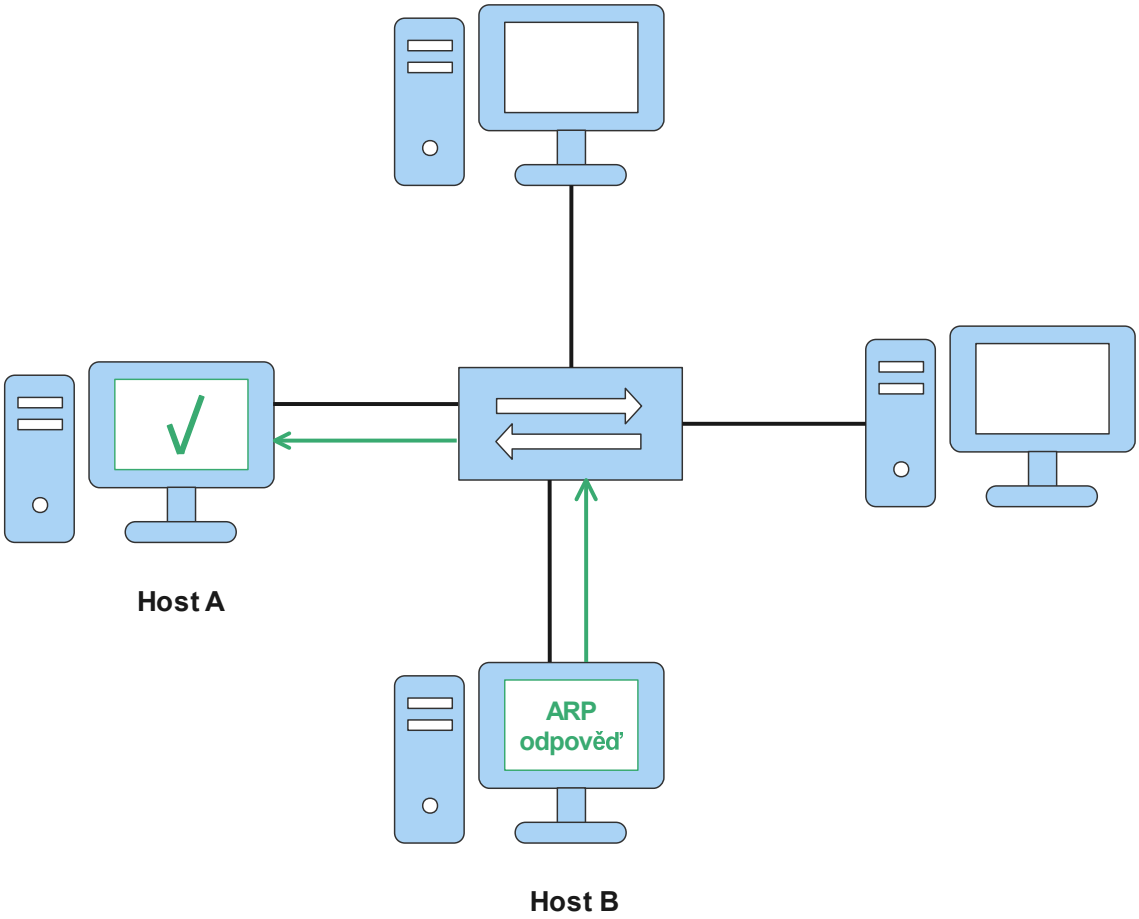
Obr. 4.3: ARP žádost.

ARP odpověď

ARP odpověď se zabalí do ethernetového rámce, jenž se odesílá pouze stanici, která vyslala ARP žádost. Princip komunikace lze vidět na Obr. 4.4.

Hodnoty v ARP paketu měnící se během komunikace:

- **Operace** – Pro odpověď nastavena na hodnotu 2.
- **MAC adresa odesílatele** – Shodná hodnota se zdrojovou MAC adresou v záhlaví ethernetového rámce.
- **IP adresa odesílatele** – Logická adresa odesílatele.
- **MAC adresa příjemce** – Fyzická adresa nastavená na hodnotu odesílatele ARP žádosti.
- **IP adresa příjemce** – Logická adresa nastavená na hodnotu odesílatele ARP žádosti.



Obr. 4.4: ARP odpověď.

4.2 Protokol RIP

Protokol RIP (Routing Information Protocol) patří do skupiny interních směrovacích protokolů a využívá algoritmus vektoru vzdáleností. Pracuje na úrovni síťové vrstvy referenčního modelu ISO/OSI. Existují celkem 3 verze: RIP verze 1 (RIPv1), RIP verze 2 (RIPv2) a RIP nové generace (RIPng). I přes fakt, že každá verze přináší jisté obohacení oproti předchozí verzi, v jádru stále vychází ze společných principů:

- Algoritmus vektoru vzdáleností značí, že se pro zhodnocení cest (metriku) využívá počet směrovačů (přeskoků/hopů), přes něž pakety musí projít pro doručení do destinace. Ve výchozím stavu se jedním přeskokem inkrementuje metrika o 1.
- Maximální počet přeskoků odpovídá hodnotě 15. V případě vyšší hodnoty (běžně se nastavuje 16) se cíl považuje za **nedosažitelný**. V praxi tedy uvnitř autonomního systému může paket projít maximálně přes 15 směrovačů.
- Na základě časovačů se periodicky odesílají údaje o směrovacích tabulkách na všechna aktivovaná rozhraní (která patří do procesu směrování). Ve výchozím stavu se informace vysílají každých 30 vteřin.
- Administrativní vzdálenost protokolu se na Cisco zařízeních rovná hodnotě 120.

Protokol podporuje tzv. **rozložení zátěže**. Pokud do cílové sítě existuje více cest s **totožnou** metrikou, do směrovací tabulky se instalují všechny tyto cesty. Následně se při komunikaci s touto sítí odesílají pakety rovnoměrně všemi zaznamenanými cestami. Výchozí hodnota dovoluje až 4 paralelní cesty, maximálně lze využít až 6 cest [17].

4.2.1 Mechanismy pro zabránění vzniku směrovacích smyček

RIP podléhá problému pomalé konvergence, důsledkem čehož může dojít ke vzniku **směrovacích smyček**. Jedná se o stav, kdy se vyslaný paket stále odesílá v uzavřeném kruhu (smyčce) mezi směrovači a nikdy nedosáhne cíle. Tuto situaci mohou způsobit různé příčiny, např. nesprávné manuální nastavení statického směrování, nedostatečná rychlost automatického předání směrovacích informací mezi směrovači v důsledku pomalé konvergence atp. Z tohoto důvodu se přistoupilo k zavedení mechanismů, jež tomuto problému předchází:

- **Omezený počet přeskoků** (*Maximum Hop Count*) – Zcela prvním indikátorem vzniku směrovací smyčky je postupně se zvyšující metrika. Proto se zavedl maximální počet přeskoků, který limituje cestování paketu na 15 přeskoků.
- **Časovače zadržení** – Viz sekce 4.2.2.
- **Rozložení horizontu** (*Split Horizon*) – Směrovač, který přijme směrovací informace na určitém rozhraní, neodesílá tyto informace zpět přes stejné roz-

hraní. V praxi se tedy sousedům neodesílají (postupem času) celé směrovací tabulky, ale pouze vybrané záznamy, o nichž se směrovač od daných sousedů nedozvěděl[2]. Soused se neinformuje ani o síti, přes niž je s vysílajícím směrovačem přímo propojen.

- **Spouštěná aktualizace** (*Triggered Update*) – V případě změny v síti, kterou představuje např. připojení nové sítě, výpadek linky (sítě) atp., směrovač nevyčkává na vypršení časovače aktualizace, ale obratem vysílá informaci o této změně všem svým sousedům. Výpadek sítě se značí hodnotou nedosažitelnosti, tzv. „nekonečno“, tedy 16.
- **Otrávení cesty** (*Route Poisoning*) – Spočívá v označení cesty při výpadku za nedosažitelnou a následném odeslání spuštěné aktualizace.

4.2.2 Časovače

RIP využívá časovače ke svým operacím se směrovacími informacemi. Každý směrovač lokálně odpočítává čas pro konkrétní úkony:

- **Časovač aktualizace** (*Update Timer*) – Ve výchozím stavu směrovač odesílá každých 30 vteřin svou kompletní směrovací tabulku na všechna rozhraní, která se účastní procesu směrování (s ohledem na pravidlo rozdělení horizontu). Když směrovač přijme směrovací tabulku souseda, aktualizuje čas všech přijatých záznamů ve své lokální směrovací tabulce.
- **Časovač neplatnosti** (*Invalid Timer*) – Pokud směrovač nepřijme aktualizaci záznamu do 180 sekund (výchozí hodnota), označí cestu za neplatnou. Metrika této cesty se automaticky nastaví na hodnotu 16.
- **Časovač zahození** (*Flush Timer*) – I přes nedosažitelnost se záznam, označený jako neplatný, udržuje ve směrovací tabulce, dokud nevyprší časovač zahození. Ve výchozím stavu se jedná o čas 240 vteřin, tedy 60 vteřin po vypršení časovače neplatnosti. Po vypršení tohoto času se záznam vymaže ze směrovací tabulky.
- **Časovač zadržení** (*Holddown Timer*) – Časovač zavedený z důvodu zabránění vzniku směrovacích smyček. Směrovač po označení sítě za nedosažitelnou spustí časovač a po určitou dobu nepřijímá (nezpracovává) aktualizace na danou síť. Dále se časovač spouští při obdržení záznamu s metrikou vyšší než metrika v lokální směrovací tabulce a při obdržení záznamu o nedosažitelné síti. Děje se tak z důvodu pomalé konvergence u protokolu RIP, aby všechny

[2]Existuje speciální případ, tzv. rozložení horizontu se zpětnou otravou. Princip spočívá v odeslání celé směrovací tabulky sousednímu směrovači, ovšem sítě, o nichž se směrovač od souseda dozvěděl, se označí za nedosažitelné, tedy hodnotou metriky 16.

směrovače uvnitř autonomního systému měly čas na adaptaci. Pokud by časovač nebyl zaveden, směrovač by při nejbližší aktualizaci mohl obdržet záznam o alternativní cestě (neplatné) s vyšší nebo stejnou metrikou. Pokud dojde k přijetí informace o cestě k nedosažitelné síti s nižší metrikou, cesta se stane aktivní. Časovač není součástí standardů, implementuje se pouze na zařízeních Cisco. Ve výchozím stavu se odpočítává 180 vteřin.

V rámci celého autonomního systému musí být časovače nastaveny identicky na všech směrovačích. Pokud by tato podmínka nebyla splněna, mohlo by dojít k fatálnímu chaosu mezi směrovači a k nedosažení stavu konvergence [17], [28], [29].

4.2.3 Typy zpráv

RIP definuje dva typy zpráv – **žádost** a **odpověď**. Žádost odesílají nově připojené (aktivované) směrovače, využívající tento protokol, o celou směrovací tabulku (na všech nově aktivovaných rozhraních protokolu RIP). Odpověď se vyskytuje ve třech různých formátech:

- Odpověď na žádost.
- Aktualizace vysílaná v pravidelných třicetivteřinových intervalech.
- Spuštěná aktualizace [30].

4.2.4 Pasivní rozhraní

Je-li rozhraní součástí směrovacího procesu[3], automaticky odesílá a přijímá směrovací zprávy. Do tohoto procesu patří i rozhraní vedoucí do lokálních sítí, tedy sítí bez přítomnosti dalších směrovačů. Vzhledem k povaze všesměrového vysílání každé zařízení v síti zprávy přijme a zpracuje. Zde přichází řada nevýhod, tedy bezpečnostní riziko (stanice může podvrhnout falešné směrovací informace), využití šířky pásma pro přenos zbytečných informací (koncové stanice nakonec zahodí pakety, protože na základě čísla portu zjistí, že takový proces u nich neexistuje) a využití výpočetních kapacit zařízení. Lze tedy využít techniku pasivních rozhraní, která zabraňuje vysílání směrovacích informací do lokální sítě, ovšem tuto síť stále zahrnuje ve zprávách pro jiné směrovače.

4.2.5 RIP verze 1

RIP verze 1 (RIPv1) se stala zcela první implementací směrovacího protokolu RIP. Definuje se v dokumentu RFC 1058 [30]. Vzhledem ke své brzké standardizaci (v roce 1988) se zakládá na **třídním adresování**. Pro adresaci zařízení se využívají

[3]Definováno specifikací sítě, z jejíhož rozsahu se určí všechna rozhraní na základě nastavených IP adres.

první 3 třídy A, B a C (viz Tab. 4.1). První verze protokolu ve zprávách nepřenáší informace o masce sítě. Směrovač určuje třídu adresy na základě prvního oktetu adresy (konkrétně prvních bitů oktetu, které jsou jednoznačně určeny pro každou třídu). Když směrovač přijme informace o síti, určí masku na základě jednoho ze dvou kritérií:

1. Přijatá adresa patří do stejné třídy adres jako aplikovaná IP adresa na příchozím rozhraní. V takovém případě se zvolí stejná maska sítě, která je nastavena pro dané rozhraní.
2. IP adresa patří do jiné třídy než rozhraní, které zprávu přijalo. V tomto případě se nastaví maska sítě specifická pro danou třídu.

Obsah směrovacích zpráv

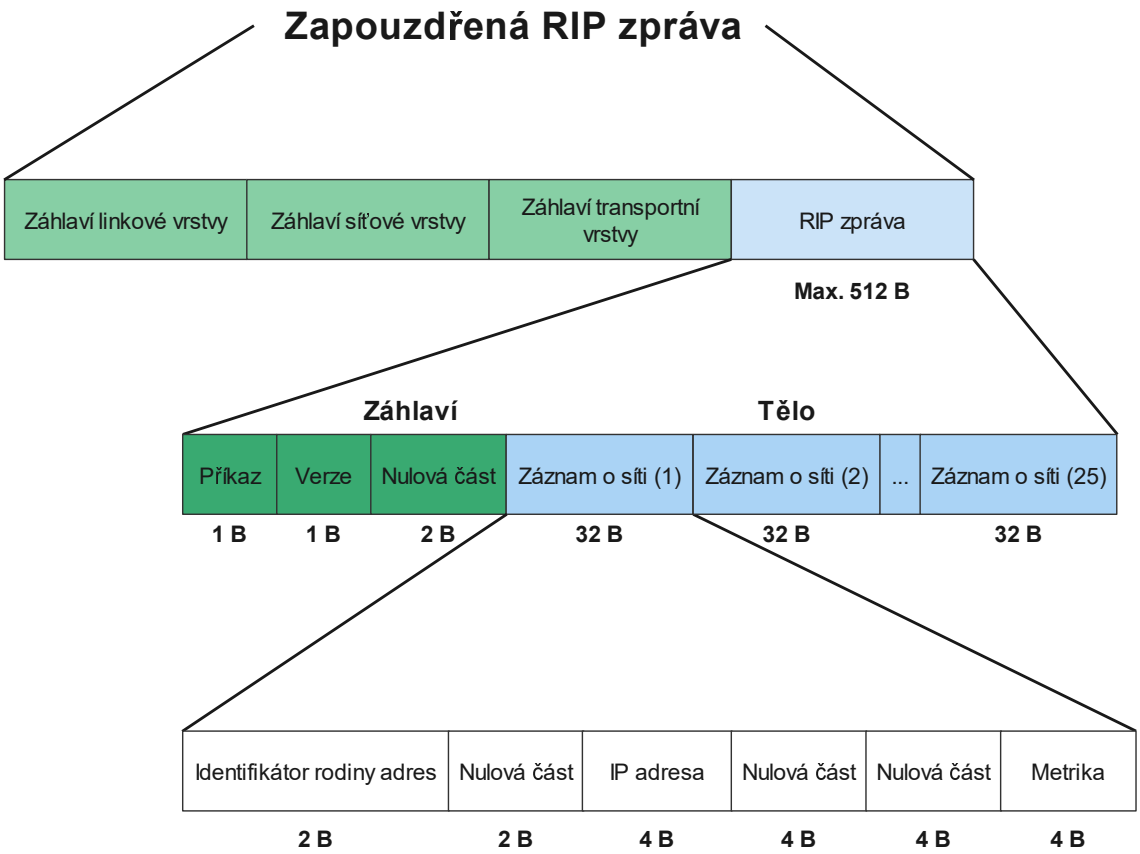
Pro výměnu směrovacích informací definuje protokol speciální formát zpráv, pomocí nějž si směrovače tyto informace předávají. RIPv1 využívá nespolehlivý transportní protokol UDP (User Datagram Protocol), kde se pro zdrojový i cílový port využívá hodnota *520*. Zpráva se odesílá všesměrově (broadcast), tedy cílová IP adresa se rovná hodnotě *255.255.255.255*[4]. Zdrojová adresa se nastavuje na adresu rozhraní, přes nějž se paket odesílá. Obdobně se na úrovni linkové vrstvy cílová MAC adresa nastavuje pro všesměrové vysílání (*ff-ff-ff-ff-ff-ff*) a zdrojová MAC adresa na fyzickou adresu zdrojového rozhraní.

Obsah zprávy lze vidět na Obr. 4.5. Samotná zpráva bez záhlaví jednotlivých vrstev může dosahovat velikosti až 512 bytů. Záhlaví se skládá ze 4 bytů a obsahuje následující položky:

- **Příkaz** – Vyjadřuje, za jakým účelem se datagram odesílá. Možné hodnoty jsou:
  - 1 – Specifikuje žádost.
  - 2 – Specifikuje odpověď.
- **Verze** – Určuje verzi protokolu RIP. Podporované verze:
  - 1 – Protokol RIP verze 1.
  - 2 – Protokol RIP verze 2.
- **Nulová část** – Poskytuje prostor budoucím vylepšením protokolu. Význam tedy spočívá v rezervě.

Tělo zprávy obsahuje informace o jednotlivých sítích. Každá síť je zahrnuta v jednom ze záznamů. Celkově zpráva může obsahovat až 25 záznamů. Skládají se z následujících položek:

[4]Tato adresa zajišťuje, že každé zařízení, které přijme paket, zpracuje data až na úroveň transportní vrstvy



Obr. 4.5: Obsah zapouzdřené RIP zprávy [17].

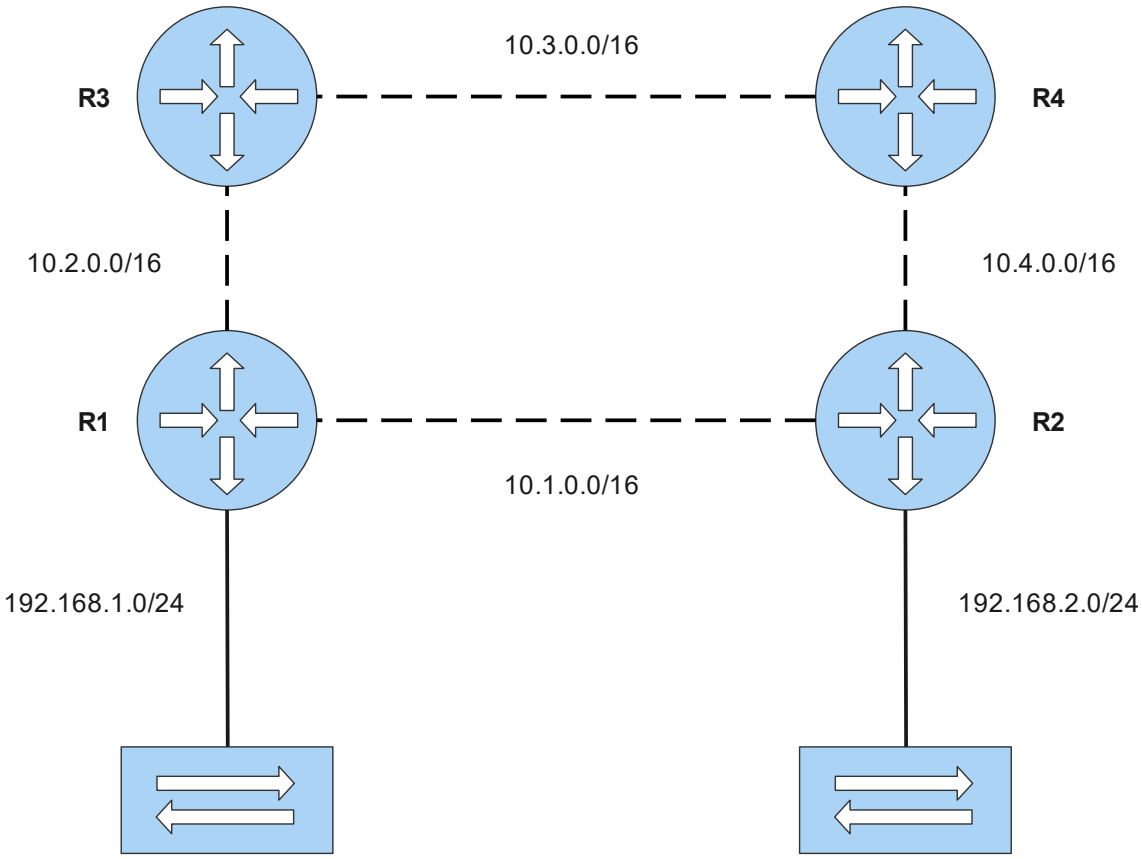
- **Identifikátor rodiny adres** – Pro IP protokol se využívá hodnota 2. Jedná se o nejčastější hodnotu. V případě žádosti o informace se nastavuje na hodnotu 0[5].
- **Nulová část** – Vyskytuje se třikrát.
- **IP adresa** – Do této položky se zapisuje adresa sítě, podsítě nebo konkrétní stanice. Ve většině případů se nevyužívají adresy hostů a adresy sítí/podsítí závisí na schématu adresování.
- **Metrika** – Obsahuje hodnotu mezi 1–16 v závislosti na počtu přeskoků. Hodnoty 1–15 vyjadřují dostupnou síť, hodnota 16 nedostupnou síť [17], [30].

Výměna směrovacích informací

V následujících bodech se popisuje výměna směrovacích informací od počátku (aktivace protokolu na směrovačích) až po dosažení úplného stavu konvergence. Popis vychází ze stavu, kdy se protokol aktivuje na všech směrovačích současně. Vychází se z topologie viz Obr. 4.6. U lokálních sítí (v topologii znázorněné přepínači) se

[5]V kombinaci s metrikou 16 se jedná o žádost o zaslání kompletní směrovací tabulky

uvažuje připojení přes pasivní rozhraní. Pro jednoduchost se uvádí stavy směrovače R1, odpovídající operace lze odvodit i na ostatních směrovačích.



Obr. 4.6: Logická topologie pro znázornění komunikace RIPv1.

Tab. 4.3: Směrovací informace odeslané směrovačem R1.

(a) Pro směrovač R2.

Adresa sítě	**Metrika**
192.168.1.0	1
10.2.0.0	1

(b) Pro směrovač R3.

Adresa sítě	**Metrika**
192.168.1.0	1
10.1.0.0	1

Tab. 4.4: Informace obdržené směrovačem R1 o přímo připojených sítích.

(a) Od směrovače R2.

Adresa sítě	**Metrika**
10.4.0.0	1
192.168.2.0	1

(b) Od směrovače R3.

Adresa sítě	**Metrika**
10.3.0.0	1

1. Po aktivaci směrovacího protokolu směrovač R1 vysílá RIP žádost přes všechna rozhraní, která se účastní směrovacího procesu. Odesílá se tedy ke směrovačům R2 a R3. Zároveň posílá svoji směrovací tabulku (viz Tab. 4.3) v podobě periodické aktualizace.
2. Směrovač R1 obdrží žádosti od směrovačů R2 a R3 a vygeneruje RIP odpovědi. Odpovědi se vygenerují i přes fakt, že totožný obsah byl již odeslán v předchozím kroku v podobě pravidelné aktualizace. Dále směrovač přijme pravidelné aktualizace od sousedních směrovačů (viz Tab. 4.4), z nichž si získané informace o sítích uloží do své lokální směrovací tabulky (viz Tab. 4.5). Současně tedy každý směrovač v síti disponuje znalostmi o přímo připojených sítích svých sousedů. Aktuálně se tedy jedná o stav konvergence, výměna informací však dále pokračuje. Směrovač ještě obdrží odpovědi na své žádosti, tedy opět duplicitní informace, jež se už dozvěděl.

Tab. 4.5: Směrovací tabulka směrovače R1.

Kód	**Adresa sítě**	**Prefix**	**Metrika**	**Next hop**
C	192.168.1.0	/24	0	-
C	10.1.0.0	/16	0	-
C	10.2.0.0	/16	0	-
R	10.3.0.0	/16	1	R3
R	10.4.0.0	/16	1	R2
R	192.168.2.0	/24	1	R2

3. V následujícím kroku směrovač R1 obdrží záznamy o sítích, o nichž již udržuje záznamy, ovšem dostupných jinou cestou (přes jiný směrovač). Jedná se o aktualizace, které přeposílají sousední směrovače. Obsah zpráv viz Tab. 4.6. Stejným způsobem vysílá informace získané od sousedů se zvýšením metriky (Tab. 4.7). Vzhledem k očividnému navyšování metriky se tyto cesty ignorují, protože směrovače udržují lepší cesty k cíli. Přeposílají se pouze cesty, o nichž se směrovač nedozvěděl přímo od sousedů, jimž cesty zasílá.

Tab. 4.6: Informace obdržené směrovačem R1 o nepřímo připojených sítích.

(a) Od směrovače R2.

Adresa sítě	Metrika
10.3.0.0	2

(b) Od směrovače R3.

Adresa sítě	Metrika
10.4.0.0	2
192.168.2.0	3

Tab. 4.7: Informace odeslané směrovačem R1 o nepřímo připojených sítích.

(a) Pro směrovač R2.

Adresa sítě	Metrika
10.3.0.0	2

(b) Pro směrovač R3.

Adresa sítě	Metrika
192.168.2.0	2
10.4.0.0	2

4. Nyní se již všechny potřebné informace rozeslaly a směrovače dosáhly úplného stavu konvergence. Na základě časovačů se dále (za předpokladu stálé aktivity směrovačů a žádných změn v síti) odesílají pouze pravidelné aktualizace (viz Tab. 4.8). Směrovač naopak přijímá pravidelné aktualizace (viz Tab. 4.9).

Při pohledu na topologii lze odvodit, že po dosažení stavu konvergence využívají směrovače R3 a R4 mechanismu rozložení zátěže. Směrovač R3 se dozvídá o síti *192.168.2.0* ze dvou směrů – od směrovačů R1 a R4. Oba údaje přichází se stejnou hodnotou metriky 2. Směrovač mezi nimi nerozlišuje a využije obě cesty pro dosažení cíle. Stejný postup platí pro směrovač R4 ve vztahu k síti *192.168.1.0*.

Agregace adres

RIP verze 1 podporuje sítě a podsítě, ovšem pouze se **stejnou** maskou sítě. Jak již bylo zmíněno, směrovače přiřazují masky sítím na základě masky příchozího roz-

Tab. 4.8: Pravidelná aktualizace zasílaná od směrovače R1.

(a) Pro směrovač R2.

Adresa sítě	Metrika
10.2.0.0	1
10.3.0.0	2
192.168.1.0	1

(b) Pro směrovač R3.

Adresa sítě	Metrika
10.1.0.0	1
10.4.0.0	2
192.168.1.0	1
192.168.2.0	2

Tab. 4.9: Pravidelná aktualizace vysílaná pro směrovač R1.

(a) Od směrovače R2.

Adresa sítě	Metrika
10.3.0.0	2
10.4.0.0	1
192.168.2.0	1

(b) Od směrovače R3.

Adresa sítě	Metrika
10.3.0.0	1
10.4.0.0	2

hraní nebo třídního rozsahu. Pokud směrovač propojuje sítě patřící do různých tříd, nazývá se **hraničním směrovačem**. V jedné části autonomního systému může existovat několik podsítí (se stejnou síťovou maskou) patřících do jedné třídy. Hraniční směrovač, který odesílá informace o těchto podsítích přes síť jiné třídy, neodesílá jednotlivé záznamy, ale vyjádří je jedinou třídní adresou. Na protější straně se adrese přiřadí maska třídy a všechny podsítě se stanou dostupnými skrze jeden záznam ve směrovací tabulce. Tento proces sloučení podsítí do jedné sítě se nazývá agregace adres nebo též **sumarizace**. Výhodou procesu agregace je zmenšení počtu záznamů ve směrovacích tabulkách a tím dosažení rychlejšího vyhledávání cest pro pakety. Na druhé straně u třídního adresování, pokud se navrhne nespojité adresovací schéma, může se vyskytnout problém s nesprávným směrováním, rozložením zátěže, případně kompletní nedostupností sítí.

4.2.6 RIP verze 2

Verze 1 protokolu RIP se postupně stala neefektivní vzhledem k principu třídního adresování. S nárustem zařízení využívajících internetové připojení se velmi rychle začal vyčerpávat adresní rozsah. Z tohoto důvodu se zavedly nové mechanismy **CIDR** (Classless Inter-Domain Routing) a **VLSM** (Variable-Length Subnet Mask) pro podporu beztřídního adresování:

- CIDR upouští od třídních rozsahů s pevně stanovenými síťovými maskami. Zavádí prefixovou notaci, kde se maska vyjadřuje počtem bitů s log. hodnotou 1 (např. maska *255.255.255.0* se vyjádří notací */24*). S tímto mechanismem lze přiřadit libovolný prefix sítím bez ohledu na jejich třídu. CIDR využívá VLSM. Vytváří i nový přístup k sumarizaci adres. U třídního adresování se agregovaly rozdělené podsítě do třídní sítě. CIDR umožňuje sumarizovat rozsah třídních adres do jedné adresy s nižší maskou než třídní. Tato adresa se označuje jako nadsíť (supernet).
- VLSM dává možnost nerovnoměrného podsíťování. U protokolů založených na třídním adresování bylo možné rozdělit rovnoměrně síť na podsítě se stejnou

délkou masky. Mechanismus VLSM umožňuje tyto podsítě dělit na další podsítě s libovolnou maskou (vyšší než originální maska podsítě) bez ohledu na rozsah jiných podsítí. Dochází tak ke značné úspoře adres [31].

Se zavedením beztřídního adresování muselo dojít ke zcela novému přístupu ve směrování. Začal tedy vývoj nových směrovacích protokolů. Patří mezi ně i nová verze protokolu RIP. RIP verze 2 (RIPv2), definovaná v dokumentu RFC 1723 [32], rozšiřuje funkce verze 1 a přidává určité novinky vylepšující bezpečnost a rychlost protokolu. V jádru se tedy jedná o stejný protokol s vylepšenými vlastnostmi.

Obsah směrovacích zpráv

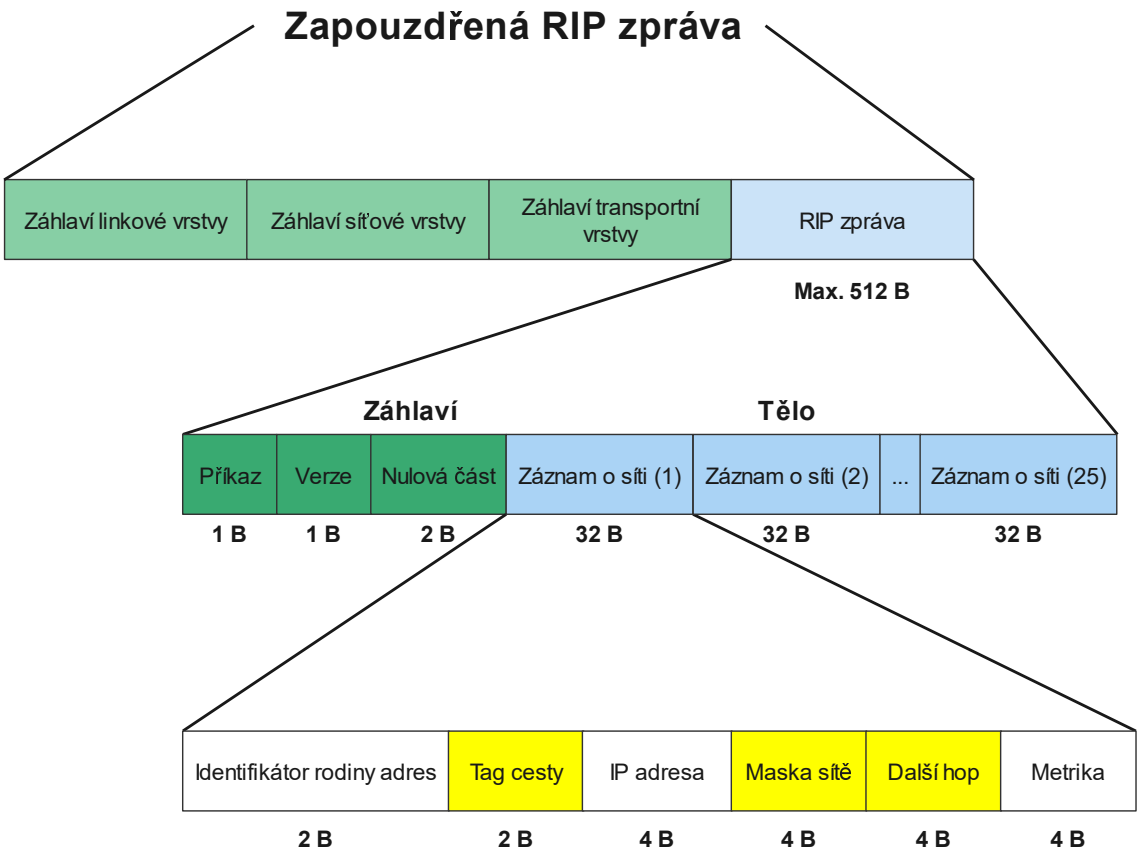
Struktura zapouzdření RIP zpráv zůstává shodná s verzí 1. Transportní protokol i porty zůstávají stejné. Ke změně dochází na síťové a linkové vrstvě. Pro vysílání se využívá skupinová adresa místo všesměrové, konkrétně *224.0.0.9* [31]. Výhoda spočívá v urychlení zpracování dat, kdy koncové stanice, které paket přijmou, zjistí již na úrovni linkové vrstvy, že nejsou cílovou destinací zprávy[6]. Ke změnám dochází i v samotném obsahu zprávy. Nevyužitý prostor v podobě nulových částí (rezerv) u RIPv1 se nahrazuje užitečnými informacemi u verze 2 (viz Obr. 4.7, položky vyznačené žlutě). Dochází tak k plnému využití všech dostupných částí uvnitř záznamu o síti.

Vzhledem k totožnému obsahu zprávy jsou dále vysvětleny pouze nově přidané položky:

- **Tag cesty** – Slouží k odlišení cest zjištěných protokolem RIP od cest naučených jiným směrovacím protokolem.
- **Maska sítě** – Směrovače přenáší informace o masce sítě, odpadá tedy nutnost určit masku na základě masky rozhraní či třídní adresy. Jedná se o výraznou změnu oproti původní koncepci třídního adresování, protože se nově otevírá možnost využití libovolně rozdělených podsítí s variabilní maskou sítě (CIDR a VLSM).
- **Další hop** – Specifikuje adresu směrovače (bezprostředně sousedního ve stejné podsíti), k němuž mají být data odeslána pro dosažení cíle. Pokud pole obsahuje adresu *0.0.0.0*, směrovač, od nějž zpráva přišla, se volí jako další přeskok. Jiná adresa označuje „vhodnější“ směrovač pro dosažení cíle, dochází tedy k přesměrování [17], [32].

Verze 2 je zpětně kompatibilní s verzí 1 za předpokladu dodržení podmínky, že žádná z rezervovaných položek se nerovná jiné hodnotě než nulové.

[6] Cílová MAC adresa se nastavuje na hodnotu *01:00:5e:00:00:09*, tedy všechna zařízení, na nichž RIP není aktivní, pakety zahazují.



Obr. 4.7: Nové položky v RIPv2 zprávě.

Výměna směrovacích informací

Pro srovnání obou verzí se využívá identická topologie, avšak s odlišným adresovacím schématem (viz Obr. 4.8). Komunikace prostřednictvím protokolu RIP verze 2 probíhá ve své podstatě stejným způsobem jako komunikace u verze 1. Stále se dodržuje systém výzva-odpověď a pravidelné zasílání aktualizací. Postup komunikace se opět rovná verzi 1, pouze s odlišnými adresami. Rozdíl přichází v obsahu směrovacích informací. Jak již bylo zmíněno, RIPv2 přenáší kromě adresy sítě tag cesty, masku sítě a informaci o dalším přeskoku.

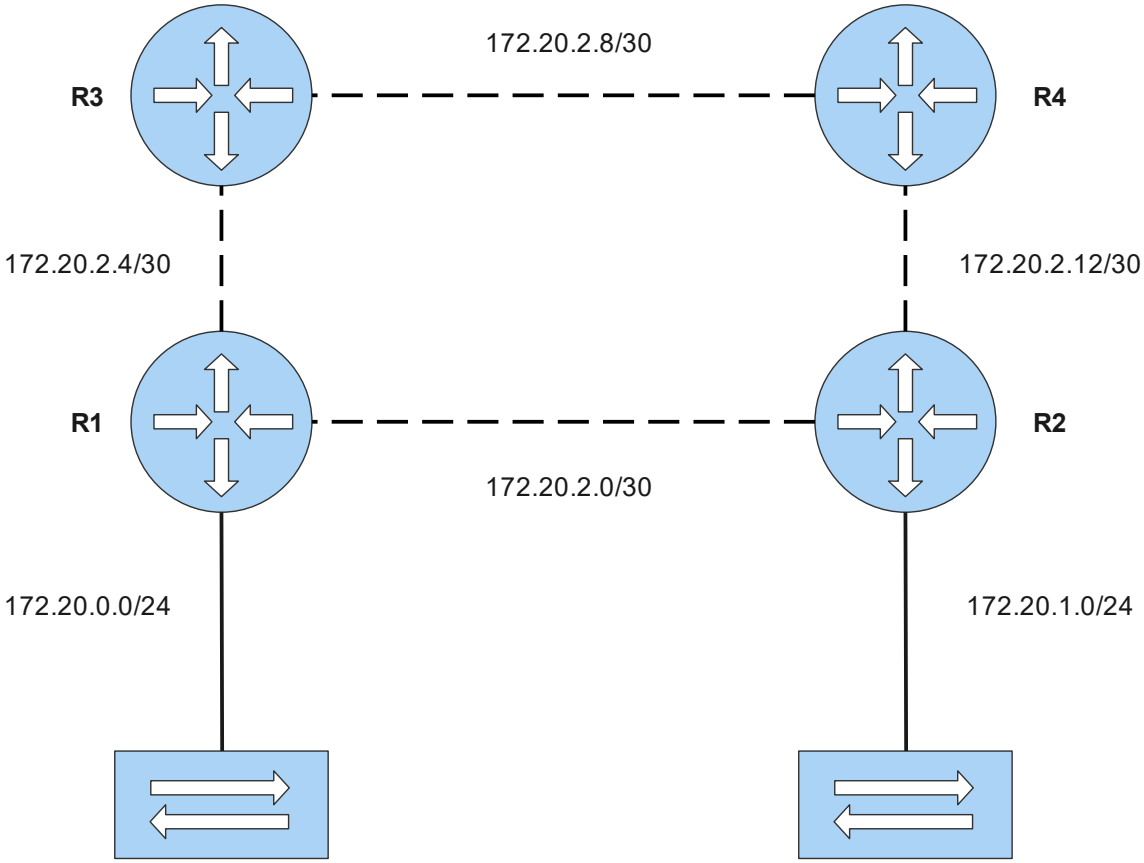
Současné adresovací schéma není možné použít pro směrování pomocí verze 1. Směrovač přiřazuje masku sítě na základě adresy příchozího rozhraní nebo třídní adresy. Z topologie vyplývá, že všechny adresní rozsahy spadají pod jednu adresu třídy B – *172.20.0.0/16*. Směrovače R1 a R2 tedy nemohou informovat své sousední směrovače o připojených lokálních sítích, protože by nedokázaly správně určit masku sítě[7].

Tento problém zaniká s verzí 2, která v záznamu o síti přenáší masku sítě. Směrovače tedy mohou informovat své sousedy o všech sítích, protože si masku nedopo-

[7]Místo prefixu /24 by se přiřadil prefix /30, který nepokrývá celý rozsah lokální sítě.

čítávají sami, ale získají tyto informace od vysílacích směrovačů.

Příklad pravidelné aktualizace se uvádí pro směrovač R1, konkrétně směrovací informace určené směrovačům R2 (viz Tab. 4.10) a R3 (viz Tab. 4.11).



Obr. 4.8: Logická topologie pro znázornění komunikace RIPv2.

Tab. 4.10: Pravidelná aktualizace směrovače R1 určená pro R2.

Adresa sítě	**Prefix**	**Next hop**	**Metrika**	**Tag**
172.20.0.0	/24	0.0.0.0	1	0
172.20.2.4	/30	0.0.0.0	1	0
172.20.2.8	/30	0.0.0.0	2	0

Tab. 4.11: Pravidelná aktualizace směrovače R1 určená pro R3.

Adresa sítě	Prefix	Next hop	Metrika	Tag
172.20.0.0	/24	0.0.0.0	1	0
172.20.1.0	/24	0.0.0.0	2	0
172.20.2.0	/30	0.0.0.0	1	0
172.20.2.12	/30	0.0.0.0	2	0

Autentizace

RIP verze 2 podporuje mechanismus autentizace pro další zvýšení bezpečnosti při výměně směrovacích informací. Autentizace představuje volitelnou službu pro ověření sousedů. Vychází z nakonfigurování klíčů na směrovačích napříč celou sítí. Ty se mohou vyskytovat v textové podobě nebo ve formě hashe, výstupu hashovací funkce MD5 (Message-Digest algorithm). Klíče musí být souhlasné na všech zařízeních.

Uvnitř RIP zprávy zabírá autentizace prostor jednoho záznamu o síti, počet maximálního počtu záznamů se tak snižuje na 24. Uvnitř autentizačního záznamu se identifikátor rodiny adres nastavuje na hodnotu *0xffff*. Tag cesty se nahrazuje za **typ autentizace** (2 značí textovou formu klíče, 3 značí MD5 hash). Zbylý prostor se vyhrazuje pro samotný klíč [32], [33], [34].

4.2.7 RIP nové generace

RIP nové generace (RIPng) definuje standard RFC 2080 [35]. Oproti předchozím verzím se nová generace zaměřuje na sítě s adresováním IPv6. Základní principy (Bellman-Fordův algoritmus, maximálně 15 přeskoků, pravidelné aktualizace) zůstávají zachovány. Na transportní vrstvě využívá protokol UDP, port 521 [36].

4.3 Protokol OSPF

S rozvojem internetu a inovací v oblasti informačních a komunikačních technologií se princip vektoru vzdáleností, založený čistě na počtu přeskoků k cíli, ukázal jako nedostatečný. Proto se začal vyvíjet zcela nový přístup ke směrování. Na výstupu byl představen princip stavu linek, jehož zřejmě nejznámějším představitelem je protokol OSPF (Open Shortest Path First).

Protokol zahrnuje dohromady 3 verze [17], z nichž první (OSPFv1) se v praxi nevyužila, druhá verze (OSPFv2), standardizovaná v dokumentu RFC 2328 [37] (dříve RFC 1247 [38]), slouží pro směrování v sítích IPv4 a třetí verze (OSPFv3), vycházející z dokumentu RFC 2740 [39], se využívá uvnitř sítí IPv6.

Směrovače s aktivovaným OSPF si oproti protokolu RIP uchovávají informace o kompletní topologii sítě, z nichž následně pomocí Dijkstrova algoritmu nejkratších cest, označovaného jako SPF (Shortest Path First), vypočítají nejoptimálnější cesty do všech destinací. Informace o všech sítích se uchovávají v **tabulce topologie**, označované též jako LSDB (Link State Database). Každý směrovač má po výměně informací identickou tabulku. Nachází se zde směrovače a jim náležící informace o spojích (rozhraních účastnících se procesu směrování), tedy druh sítě, adresa sítě, prefix, propustnost, cena spoje atp. Tyto parametry představují stav spoje. Dále směrovače udržují **tabulku sousedů**, kde se ukládají informace o sousedních směrovačích (každý směrovač udržuje vlastní). Lze zde najít např. ID souseda, adresu, stav spojení mezi sousedy aj. Poslední je **směrovací tabulka**, kam se umisťují nejrychlejší cesty vypočtené SPF algoritmem.

OSPF patří mezi protokoly podporující beztřídní adresování, využívá tedy mechanismy CIDR a VLSM. Ve srovnání s protokolem RIP dosahuje rychlejší konvergence, neboť příchozí směrovací informace uchovává v lokální databázi (tabulce) a ihned je odesílá všem svým sousedům (kromě souseda, od nějž informace obdržel). Až po rozeslání všech informací, kdy každý směrovač v síti udržuje totožnou tabulku topologie, spustí každý lokálně SPF algoritmus, který vyhodnocuje nejlepší cesty z informací uchovávaných v tabulce topologie. Kompletní směrovací informace se vysílají na počátku a každých 30 minut[8]. V mezidobí se vysílá pouze při změně v topologii a přenáší se informace týkající se pouze této změny. Metrika protokolu se odvíjí od souhrnné ceny cest. Cena cesty se určuje číselnou hodnotou reprezentující lokální odchozí rozhraní na směrovači. Nižší hodnota značí lepší cestu. Na Cisco zařízeních se cena vypočte podle propustnosti rozhraní.

Samotné směrovací informace se přenáší prostřednictvím LSA (Link State Advertisement) zpráv. Směrovače, které přijmou informace od jiných směrovačů, tyto zprávy posílají beze změny dále.

[8]Směrovače vysílají pouze vlastní záznamy, o nichž zpravily sousedy.

4.3.1 OSPF oblasti

Směrovače implementující protokol OSPF udržují kompletní mapu topologie, tedy informace o všech směrovačích a jejich připojených sítích (které se účastní směrování), což představuje určité nároky na paměť zařízení podle velikosti topologie. U rozsáhlých sítí by požadavky na paměť byly značné. Proto se využívá rozdělení sítě do jednotlivých oblastí. Do paměti jednotlivých zařízení se tedy ukládají informace pouze o sítích a zařízeních spadajících do stejné oblasti (vyjma hraničních směrovačů). Zároveň se uvolní šířka pásma pro přenos uživatelských dat místo směrovacích informací a výpočetní kapacita CPU.

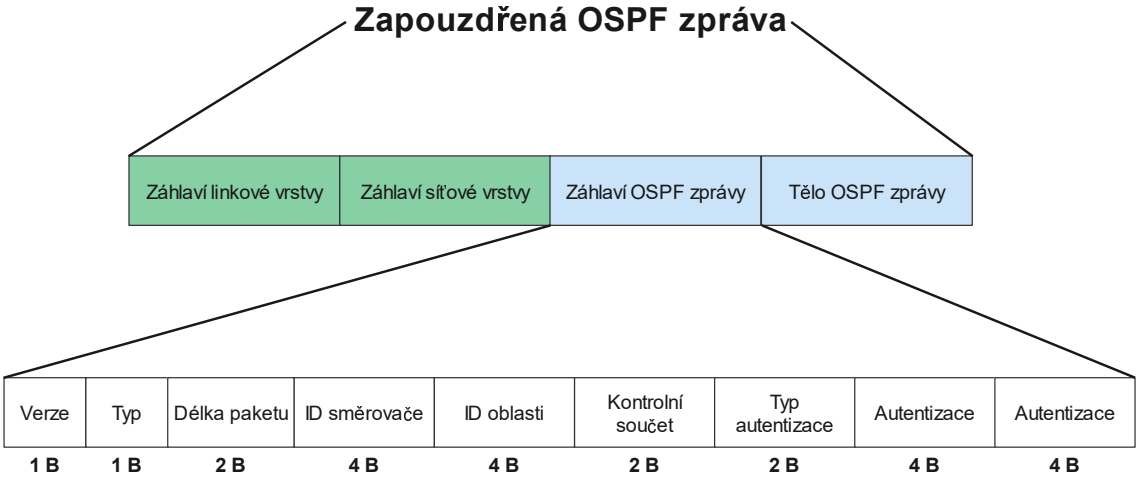
Oblasti se řadí hierarchicky a značí se číselným identifikátorem. Pravidlem při konfiguraci je počátek v **oblasti 0**. V případě jediné oblasti do ní patří všechna zařízení, při rozdělení do více oblastí představuje páteřní spojení mezi ostatními oblastmi. Směrovače se podle funkce řadí do následujících kategorií [31]:

- **Interní směrovače** – Směrovače patřící pouze do jedné oblasti.
- **Hraniční směrovače** – Hraniční směrovače, označované též jako ABR (Area Border Router), spojují jednu či více oblastí s oblastí 0. Tyto oblasti spadají pod administrativu jednoho autonomního systému.
- **Autonomní hraniční směrovače** – Jedná se o směrovače známé pod zkratkou ASBR (Autonomous System Border Router). Nachází se v oblasti 0 a distribuují informace získané jinými směrovacími protokoly (včetně informací z jiných autonomních systémů).

V situaci, kdy dojde ke změně uvnitř oblasti, se všechny směrovače (i v ostatních oblastech) dozví o této změně. Avšak pouze směrovače náležící do oblasti, v níž se změna udála, přepočítají cesty prostřednictvím SPF algoritmu. V ostatních oblastech se algoritmus nespouští [17], [40], [41].

4.3.2 Struktura OSPF zpráv

OSPF definuje celkově 5 typů zpráv. Každá z nich vychází ze společného záhlaví (viz Obr. 4.9), tělo zprávy se dále mění u jednotlivých zpráv. OSPF zpráva se zapouzdřuje do IP paketu, v modelu ISO/OSI lze tedy analyzovat linkovou a síťovou vrstvu. Na linkové vrstvě se pro zdrojovou MAC adresu využívá adresa vysílacího rozhraní, jako cílová MAC adresa se využívá speciální skupinová MAC adresa *01:00:5e:00:00:05* nebo *01:00:5e:00:00:06*. Na síťové vrstvě se pro zdrojovou IP adresu využívá opět adresa odchozího rozhraní, pro cílovou IP adresu se využívá skupinová adresa *224.0.0.5* nebo *224.0.0.6*. V IP paketu se dále nastavuje hodnota pole protokolu na 89 [17].



Obr. 4.9: OSPF záhlaví [17], [42].

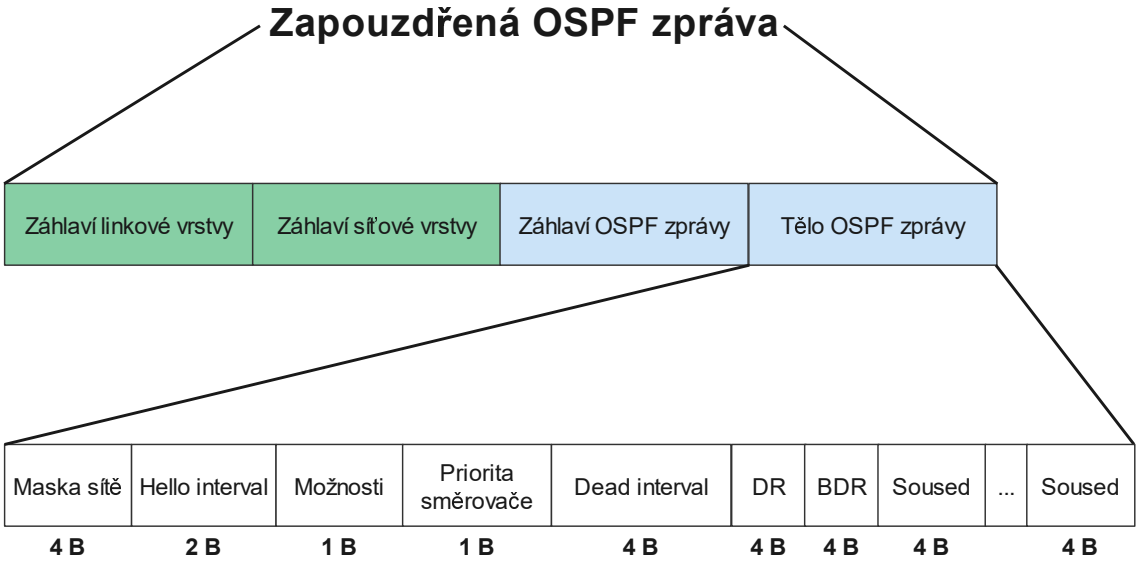
Záhlaví OSPF zpráv se vždy skládá ze stejné struktury (24 bytů). Obsahuje následující položky:

- **Verze** – Verze protokolu OSPF. Standardně 2 nebo 3.
- **Typ** – Typ OSPF zprávy, který nabývá následujících hodnot:
  - **Typ 1** – Hello paket.
  - **Typ 2** – DBD paket.
  - **Typ 3** – LSR paket.
  - **Typ 4** – LSU paket.
  - **Typ 5** – LSAck paket.
- **Délka paketu** – Celková délka OSPF zprávy včetně záhlaví.
- **ID směrovače** – Identifikátor směrovače, který zprávu odesílá.
- **ID oblasti** – Identifikátor oblasti, do níž směrovač spadá. Zapisuje se ve formě dekadického čísla, případně ve stejném formátu jako IPv4 adresa.
- **Kontrolní součet** – Kontrolní součet zajišťuje správnost přenosu celé OSPF zprávy kromě položky autentizace.
- **Typ autentizace** – Stejně jako u protokolu RIP i protokol OSPF podporuje přenos hesel prostřednictvím textu či hashe. Definují se zde 3 hodnoty:
  - **0** – Bez autentizace.
  - **1** – Autentizace prostřednictvím textového hesla v čitelné podobě.
  - **2** – Autentizace prostřednictvím hesla uloženého v podobě hashe.
- **Autentizace** – V závislosti na typu autentizace se zde přenáší informace o hesle.

Dále se rozebírá obsah jednotlivých zpráv podle typu.

Typ 1 – Hello paket

Hello pakety odesílá každý směrovač automaticky po zahrnutí rozhraní do procesu směrování. Prostřednictvím těchto zpráv se vytvoří vazby mezi sousedy, bez nichž nedochází k výměně směrovacích informací. Dále se využívají k pravidelnému testování dostupnosti sousedů a tím k udržování vazeb.



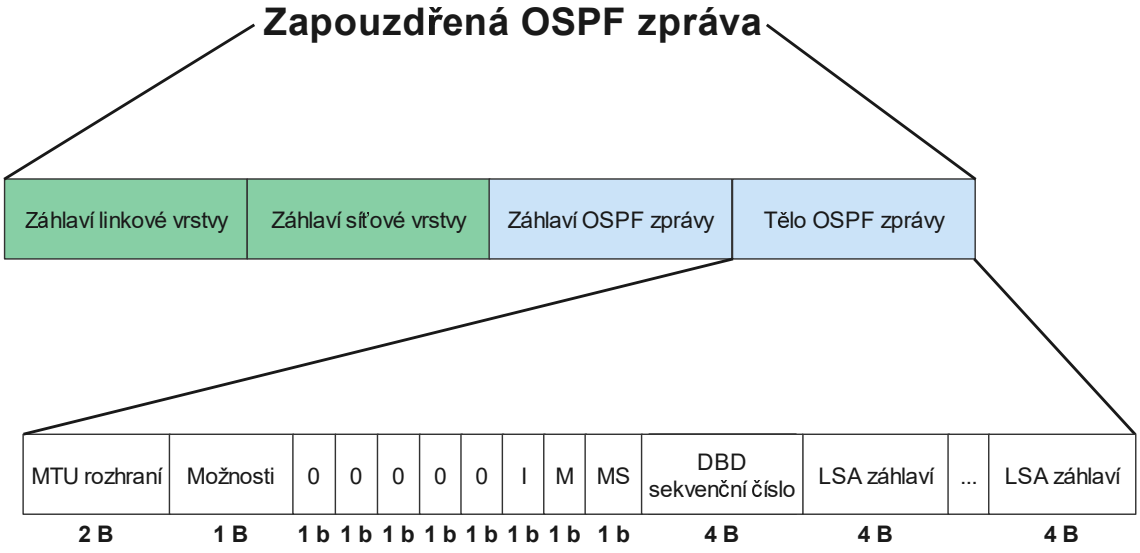
Obr. 4.10: OSPF Hello paket [17], [42].

Pakety obsahují následující položky (viz Obr. 4.10):

- **Maska sítě** – Maska sítě, do níž patří vysílací rozhraní směrovače.
- **Hello interval** – Interval, po němž se pravidelně zasílají Hello zprávy.
- **Možnosti** – Dodatečné možnosti směrovače.
- **Priorita směrovače** – Priorita směrovače, která se vyhodnocuje jako první parametr při výběru DR a BDR.
- **Dead interval** – Interval, během nějž, pokud směrovač neobdrží od souseda Hello zprávu, vyhodnotí souseda jako nedostupného.
- **DR** – DR (Designated Router) představuje identifikátor (IP adresu) vyhrazeného směrovače, který slouží pro přijímání a zasílání směrovacích zpráv ve všesměrové doméně.
- **BDR** – BDR (Backup Designated Router) plní funkci záložního směrovače pro případ selhání DR ve všesměrové doméně (opět IP adresa).
- **Soused** – Identifikátory směrovačů, od nichž vysílací směrovač obdržel Hello zprávy.

Typ 2 – DBD paket

Směrovače si po zjištění své existence vzájemně vymění informace o svých lokálních LSDB tabulkách ve formě DBD (Database Description) paketů. Přenáší se pouze zkrácené záznamy pro úsporu šířky pásma a informace se mohou přenášet prostřednictvím více paketů. Mezi sousedy se určí role *master* a *slave*. Master odesílá DBD paket jako první, slave následně odpovídá svým DBD paketem. Tyto pakety se vysílají při synchronizaci databází mezi sousedy.



Obr. 4.11: OSPF DBD paket [17], [42].

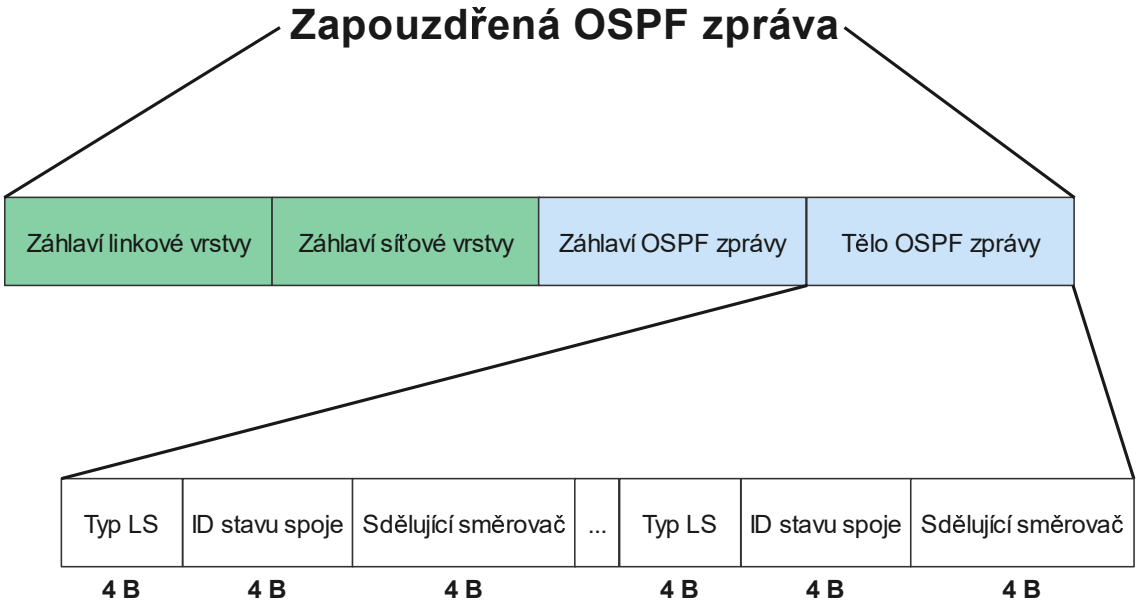
Paket se skládá z následujících položek (viz Obr. 4.11):

- **MTU rozhraní** – MTU (Maximum Transmission Unit) určuje největší možnou délku rámce, kterou lze odeslat daným rozhraním bez nutnosti fragmentace paketu (pro sítě Ethernet a Point-to-Point se jedná o délku 1500 bytů [43]).
- **Možnosti** – Dodatečné možnosti směrovače.
- **0** – Bity sloužící jako rezerva.
- **I** – Inicializační bit, jehož hodnota se rovná log. 1, pokud se jedná o zcela první DBD paket.
- **M** – Bit s hodnotou log. 1, pokud následuje více DBD paketů.
- **MS** – Bit označující směrovač s rolí master, pokud se rovná hodnotě log. 1.
- **DBD sekvenční číslo** – Sekvenční číslo se využívá pro unikátní rozlišení jednotlivých DBD paketů během výměny informací a zároveň k potvrzení přijetí. Na počátku zvolí master sekvenční číslo (inicializační bit se nastaví na hodnotu log. 1) a odesílá DBD paket k směrovači s rolí slave. Slave odpovídá svým DBD paketem se stejnou hodnotou sekvenčního čísla. Když master obdrží tento paket, inkrementuje hodnotu následujícího DBD o 1 a komunikace dále pokračuje.

- **LSA záhlaví** – V DBD paketech se přenáší pouze záhlaví LSA zpráv obsažených v databázi topologie. Přenáší se zde ID vysílacího směrovače, ID stavu spoje (informace jednoznačně popisující danou část sítě v závislosti na typu LSA), sekvenční číslo, stáří LSA záznamu aj.

Typ 3 – LSR paket

Po obdržení DBD paketů směrovače porovnají jejich obsah se svou lokální LSDB tabulkou. Pokud zjistí chybějící či zastaralé informace, vysílají LSR (Link State Request) paket pro obdržení těchto informací od daného souseda.



Obr. 4.12: OSPF LSR paket [17], [42].

V paketu lze nalézt následující informace (viz Obr. 4.12):

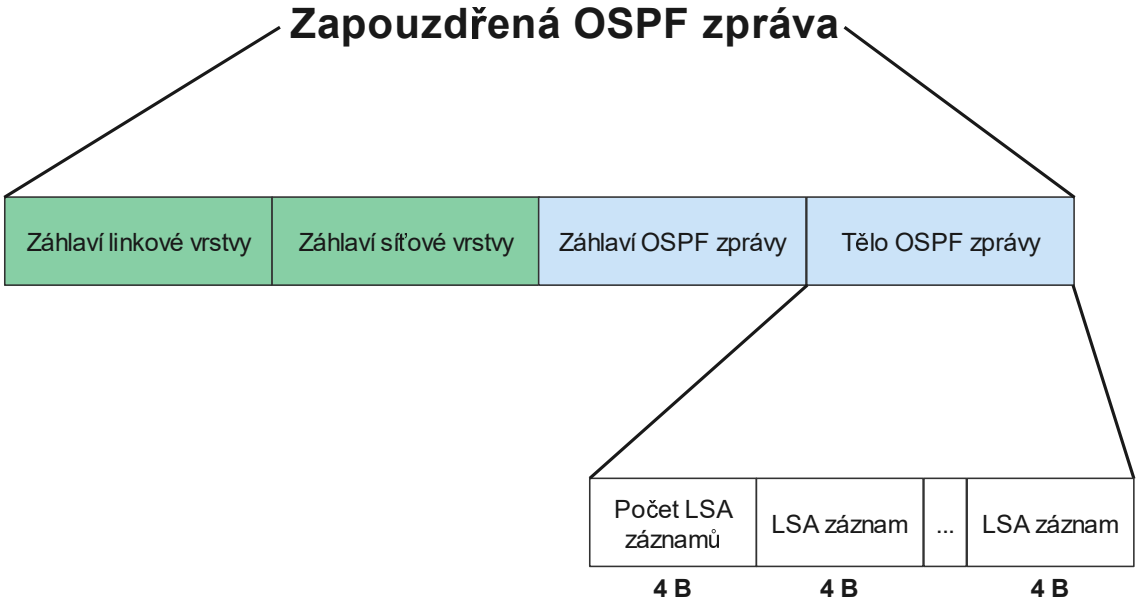
- **Typ LS** – Typ LSA zprávy, kterou směrovač požaduje. Existuje celkem 11 zpráv, níže jsou uvedeny pouze některé z nich:
  - **Typ 1 – LSA směrovače** (*Router LSA*) – Uvnitř oblasti se odesílají informace o přímo připojených sítích k směrovači.
  - **Typ 2 – LSA sítě** (*Network LSA*) – Vysílá se DR směrovačem ve všesměrových sítích (uvnitř oblasti). Obsahuje informace o síti (adresa sítě a prefix) a všech připojených směrovačích.
  - **Typ 3 – Souhrnné LSA** (*Summary LSA*) – Vytváří jej hraniční směrovače (ABR) pro informování směrovačů v jiných oblastech o dostupnosti sítí uvnitř oblasti, kterou ABR propojuje s páteřní oblastí.
  - **Typ 4 – Souhrnné LSA ASBR směrovače** – Pokud ASBR směrovač obdrží informace z jiné směrovací domény, vysílá je do OSPF oblasti

prostřednictvím této zprávy. Hraniční směrovače tuto zprávu šíří do svých oblastí prostřednictvím stejné zprávy [44], [45].

- **ID stavu spoje** – V závislosti na typu LS se odesílá identifikátor daného LSA:
  - **Typ 1** – ID směrovače, který LSA vyslal (uvnitř DBD).
  - **Typ 2** – IP adresa rozhraní uvnitř všesměrové sítě DR směrovače.
  - **Typ 3** – IP adresa cílové sítě.
  - **Typ 4** – ID ASBR směrovače [37], [44].
- **Sdělující směrovač** – ID směrovače, který uchovává aktuálnější informace (odesílatel LSA).

Typ 4 – LSU paket

Když směrovač obdrží LSR zprávu, odesílá odpověď obsahující vyžadované informace, tedy LSU (Link State Update) paket. Využívají se také pro informování o změně v topologii a rozesílání informací od jiných směrovačů.



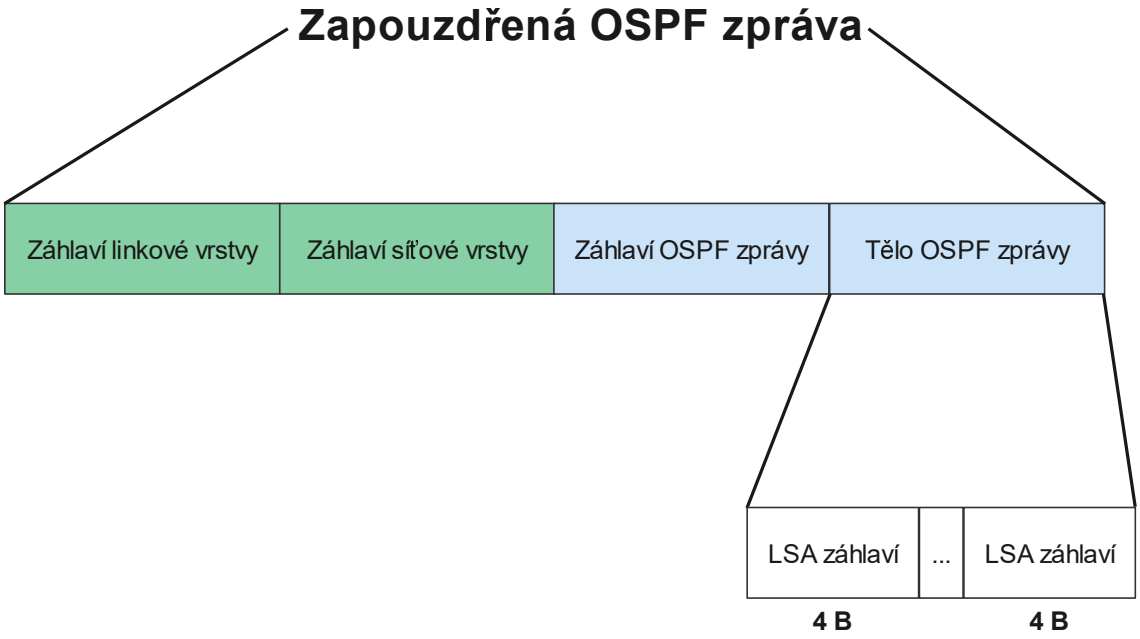
Obr. 4.13: OSPF LSU paket [17], [42].

Paket přenáší následující informace (viz Obr. 4.13):

- **Počet LSA záznamů** – Udává počet LSA záznamů přenášených v paketu.
- **LSA záznam** – Vysílají se podrobné informace, obsažené v těle LSA, k vyžádaným záznamům v závislosti na typu LSA (tedy např. údaj o metrice pro typ 1 atp.) včetně záhlaví.

Typ 5 – LSAck paket

Směrovač, k němuž LSU paket dorazí, ihned potvrzuje jeho přijetí zprávou LSAck (Link State Acknowledgment). Pokud zdroj LSU zprávy neobdrží potvrzení, opakuje přenos.



Obr. 4.14: OSPF LSAck paket [17], [42].

Paket obsahuje následující položky (viz Obr. 4.14):

- **LSA záhlaví** – Přenáší se záhlaví LSA zpráv, jejichž přijetí se potvrzuje.

4.3.3 OSPF komunikace

Na rozdíl od komunikace protokolu RIP, která spočívá v prostém schématu výzva-odpověď, využívá OSPF komplexnější mechanismus. Před samotným přenosem směrovacích informací musí být zjištěna existence směrovače (případně více směrovačů) na opačné straně prostřednictvím Hello zpráv, poté se mezi sousedy utváří vazby, které se po kompletní výměně udržují. Směrovače prochází řadou stavů, než dojde k pouhému udržování vazeb. Smyslem těchto stavů je synchronizace databází topologie. Dále se vysvětlují jednotlivé stavy:

1. **Stav bez odezvy** (*Down State*)
   Směrovač po konfiguraci protokolu OSPF nemá znalost topologie ani sousedních zařízení. Zná pouze své přímo připojené sítě. Po explicitní konfiguraci sítí, jež se účastní procesu směrování, začíná odesílat Hello zprávy přes patřičná rozhraní. Tyto zprávy se zasílají v pravidelných intervalech a bez odezvy se nepřechází do následujících stavů.

2. **Stav inicializace** (*Init State*)
   Do stavu inicializace směrovač přechází poté, co obdrží Hello zprávu na některém ze svých rozhraní (účastnících se procesu směrování). V této fázi však ID směrovače, který zprávu přijal, není obsaženo v seznamu sousedů zprávy. ID směrovače, od nějž zpráva přišla, doplní do svých následujících Hello zpráv.
3. **Stav obousměrné komunikace** (*Two-Way State*)
   Ve chvíli, kdy směrovač obdrží Hello zprávu, v níž se vyskytuje jeho ID v seznamu sousedů, dostává se do stavu ustanovené obousměrné komunikace. Po dosažení tohoto stavu na obou stranách existuje mezi sousedy plnohodnotná vazba. Další postup se odvíjí podle typu zapojení:

Spojení bod-bod

Při zapojení bod-bod se ihned přechází do stavu počátku výměny. Směrovače propojené tímto spojem vždy dosáhnou (při správné konfiguraci) stavu kompletní synchronizace.

Všesměrová síť

U všesměrových sítí (např. Ethernet) může být v topologii zapojeno více směrovačů, což by během šíření směrovacích zpráv vedlo k značnému zahlcení přenosového pásma. Z tohoto důvodu se volí směrovače s rolí DR a BDR. DR a BDR směrovače zpracovávají a shromažďují LSA zprávy všech směrovačů (jako jediné v dané všesměrové doméně), pouze však DR je vysílá prostřednictvím adresy *224.0.0.5* všem ostatním směrovačům. BDR se volí z důvodu redundance v případě selhání DR. Ostatní směrovače (označované DROTHER) vysílají LSA na adresu *224.0.0.6*, na níž naslouchají pouze DR a BDR. Směrovače s označením DROTHER dosáhnou stavu kompletní synchronizace pouze se směrovači DR a BDR, mezi sebou vzájemně udržují stav obousměrné komunikace.

Na Cisco zařízeních se DR volí podle položek v následujícím pořadí:

(a) Manuálně nastavená OSPF priorita.
(b) Explicitně nakonfigurovaný identifikátor směrovače.
(c) Nejvyšší adresa loopback rozhraní.
(d) Nejvyšší adresa aktivního rozhraní (nemusí být součástí směrování).

Aby mohly mezi směrovači vzniknout vazby, musí se shodovat autentizační parametry přenášené Hello zprávami a dále:

- oblast,
- síť,
- Hello interval,
- Dead interval [47].

4. **Stav počátku výměny** (*ExStart State*)
   Než se přistoupí k samotné výměně DBD paketů, nesoucích zkrácené záznamy o sítích, musí být zvoleny role *master* a *slave*. Do role *master* se volí směrovač s nejvyšším ID[9]. Volí se z důvodu stanovení počátečního sekvenčního čísla pro přenos DBD paketů[10].
5. **Stav výměny** (*Exchange State*)
   Směrovače si po určení, kdo převezme roli *master* a kdo roli *slave*, vyměňují DBD pakety s informacemi ze své databáze topologie. Následně se informace porovnají s lokální databází. V případě, že obsahují nové či aktuálnější informace, přechází se do stavu načítání. V opačném případě se rovnou přechází do stavu kompletní synchronizace.
6. **Stav načítání** (*Loading State*)
   Když směrovače zjistí, že potřebují informace od sousedů, vysílají požadavky prostřednictvím LSR zpráv. Sousední směrovače naopak odpovídají LSU zprávami, které obsahují podrobné informace. Veškeré LSU zprávy musí být potvrzeny prostřednictvím LSAck zpráv.
7. **Stav kompletní synchronizace** (*Full State*)
   Do stavu kompletní synchronizace se směrovače dostanou ve chvíli, kdy sousední směrovače udržují identické informace v lokálních databázích (LSDB). V této fázi mezi sousedy došlo k plnohodnotné synchronizaci.

Dále se popisuje, jak probíhá komunikace v reálném zapojení. Vychází se ze stejné topologie jako v případě protokolu RIPv2 (viz Obr. 4.8). Průběh komunikace mezi sousedy se popisuje pro směrovače R1 a R2 (viz Obr. 4.16), stejný princip lze odvodit i pro ostatní.

1. Ve výchozím stavu se protokol aktivuje na obou směrovačích, které začnou odesílat Hello zprávy. Před samotným vysíláním určí své ID. Na směrovačích není explicitně nastavené ID ani loopback rozhraní, proto se volí nejvyšší aktivní adresa. V případě R1 se jedná o *172.20.2.5* a u R2 *172.20.2.13*. R1 vysílá Hello zprávu na skupinovou adresu *224.0.0.5*.
2. Směrovač R2 přijme Hello zprávu od R1. Nenalezl své ID v položce sousedů,

[9]Vzhledem ke skutečnosti, že se DR volí primárně na základě priority, nemusí být zvolen do role master.

[10]Přijetí DBD paketu může také způsobit přechod směrovače ze stavu inicializace do stavu obousměrné komunikace [46].

proto mění svůj stav do stavu inicializace. Následně vygeneruje svoji Hello zprávu, doplní mezi sousedy ID směrovače R1 a odešle ji.

3. Směrovač R1 obdrží Hello zprávu, jež obsahuje jeho ID, a přechází do stavu obousměrné komunikace. Směrovače jsou propojeny v síti typu bod-bod, nedochází tedy k volbě DR a BDR. Vzápětí přechází směrovač R1 do stavu počátku výměny a vygeneruje prázdnou DBD zprávu, která nezahrnuje žádná LSA záhlaví. Zpočátku sebe sama prohlašuje do role master, bity I, M a MS tedy nastavuje na hodnotu log. 1. Vygeneruje náhodné sekvenční číslo a zprávu odesílá.
4. Po přijetí DBD zprávy směrovač R2 ihned přechází do stavu počátku výměny. Podle OSPF záhlaví určí, že vlastní vyšší ID a předpokládá roli master. Odesílá vlastní DBD zprávu s bity I, M a MS nastavenými na hodnotu log. 1 a opět vygeneruje vlastní sekvenční číslo.
5. Směrovač R1, který zprávu přijal, určí sebe sama do role slave (podle vyššího ID v OSPF záhlaví) a vygeneruje DBD paket obsahující LSA záhlaví (typu 1) ze své lokální databáze topologie. Nastaví pouze bit M na log. 1 a sekvenční číslo na hodnotu čísla příchozího DBD paketu, čímž zároveň potvrzuje jeho přijetí. Zprávu odešle.
6. Směrovač R2 přijme zprávu a definitivně převezme roli master. Zároveň přejde do stavu výměny a odesílá LSA záhlaví z vlastní databáze topologie. Před odesláním inkrementuje hodnotu sekvenčního čísla o 1.
7. Směrovač R1 obdrží zkrácený výpis databáze topologie na směrovači R2. Nedisponuje dalšími informacemi, které by sdělil směrovači R2, proto vysílá DBD paket čistě pro potvrzení přijetí (identické sekvenční číslo) bez LSA záhlaví. Bity I, M i MS nastaví na hodnotu log. 0.
8. Směrovač R2, jemuž je DBD zpráva doručena, také nevlastní žádné další informace ke sdělení, proto vysílá stejný formát zprávy pouze s bitem MS nastaveným na hodnotu log. 1 a inkrementovaným sekvenčním číslem.
9. Směrovač R1 z příchozí DBD zprávy určí, že nenásledují žádné další informace, proto může vyhodnotit aktuální stav a přechází do stavu načítání. Odešle potvrzovací DBD zprávu a následně LSR paket žádající o dodatečné informace k novému záznamu z předešlé DBD zprávy.
10. Směrovač R2 také přechází do stavu načítání. Na přijatou LSR zprávu odpovídá LSU paketem a vygeneruje vlastní LSR zprávu pro dodatečné informace od směrovače R1. V LSU zprávě přenáší informace o sítích *172.20.1.0/24*, *172.20.2.0/30* a *172.20.2.12/30* (ID stavu spoje typu 3).
11. Směrovač R1 zpracuje příchozí LSU zprávu, doplní informace do své LSDB a přechází do stavu kompletní synchronizace. Dále odpovídá na LSR zprávu od směrovače R2 vlastním LSU paketem se záznamy o sítích *172.20.0.0/24*,

172.20.2.0/30 a *172.20.2.4/30*.

12. Směrovače si poté ještě vymění LSU zprávy se stejným obsahem (podle vlastní databáze) s přidanou informací o propojení směrovačů mezi sebou (ID stavu spoje typu 1). Záznam tedy obsahuje ID souseda a adresu odchozího rozhraní vysílajícího směrovače (místo masky sítě).
13. Směrovače přijmou LSU pakety, doplní lokální databázi a nakonec si vzájemně potvrdí přijetí **obou** LSU (odlišeno sekvenčními čísly) prostřednictvím LSAck zpráv.
14. V této chvíli se databáze obou směrovačů plně synchronizovaly a dále se, pokud nedojde ke změně v síti, vysílají v pravidelných intervalech pouze Hello zprávy.

4.3.4 SPF algoritmus

Směrovače si předávají LSU zprávy (v nich zapouzdřené LSA), dokud nedosáhnou úplného stavu konvergence, tedy udržují identické databáze topologie. V daném momentě každý směrovač spustí lokálně Dijkstrův SPF algoritmus. Princip spočívá v uspořádání směrovačů do pomyslného stromu, nazývaného SPT (Shortest Path Tree), a následného výpočtu všech možných cest, z nichž pouze nejlepší (nejrychlejší) se vloží do směrovací tabulky. Jako kořen stromu se zvolí směrovač, na němž se výpočet odehrává, a ostatní směrovače představují listy. Spoje mezi směrovači lze označit za větve. Nejrychlejší cesta se vyznačuje nejnižší kumulativní cenou, zahrnující všechny směrovače po cestě.

Samotný algoritmus probíhá v následujících krocích:

1. Na počátku se vytvoří dva seznamy, z nichž první (dále značeno U) představuje množinu všech směrovačů v oblasti, které doposud nebyly algoritmem zpracovány, a do druhého (dále značeno V) se přesunují směrovače, které již prošly SPF výpočty [48]. Postupně se tedy všechny směrovače přesunou z množiny U do V. Směrovač, který vykonává algoritmus, označí sebe sama cenou 0 a všechny ostatní směrovače cenou ∞, protože metrika zatím není známá.
2. Vyhodnotí se všechny sousední směrovače kořene stromu a přiřadí se jim příslušná metrika (cena spojů – odchozích rozhraní). Kořen se přesouvá do seznamu V.
3. Ze sousedů (vyskytujících se v množině U) se zvolí ten s nejnižší metrikou a vyhodnotí se jeho sousední směrovače. Ke všem se spočte kumulativní cena (počínající v kořeni stromu). Pokud některá, předešle vypočtená metrika ke známému cíli dosahuje vyšší hodnoty než aktuálně spočítaná, nahradí se. V opačném případě nedochází ke změně.
4. Směrovač se přesouvá do množiny V.

5. Kroky 3 a 4 se opakují až do chvíle, kdy množina U neobsahuje žádné další položky.

Konkrétní postup algoritmu se odvíjí od topologie na Obr. 4.15, na níž jsou znázorněny ceny jednotlivých spojů. Dále se popisují jednotlivé kroky výpočtů na směrovači R1 (viz Tab. 4.12):

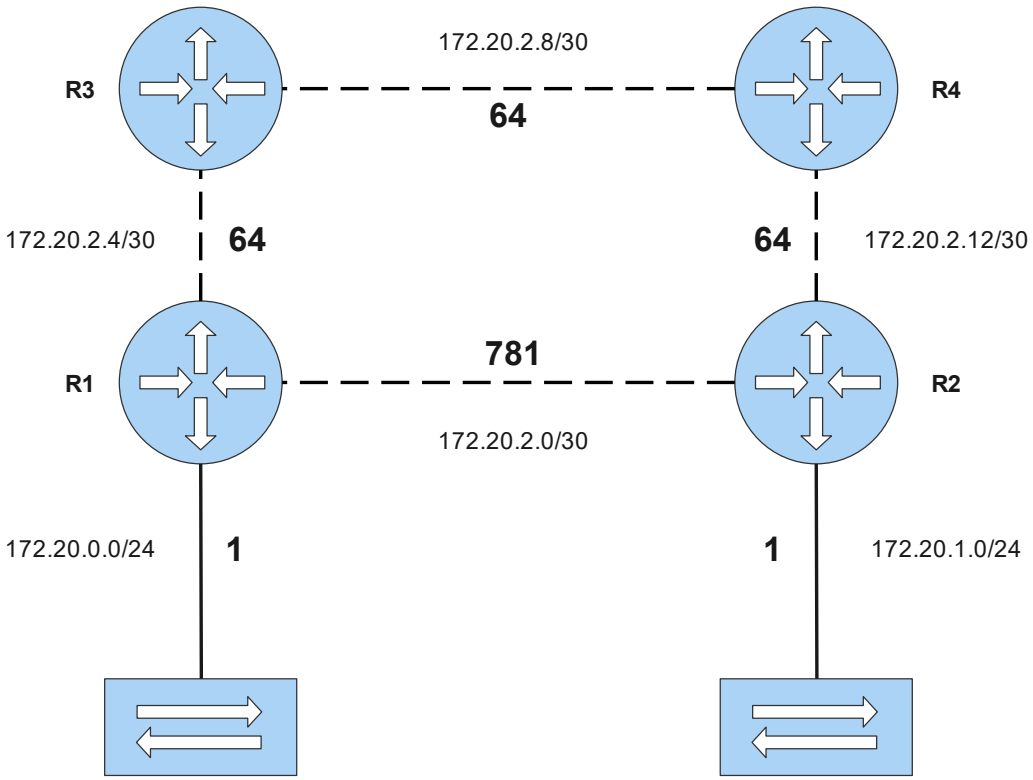
1. Směrovač R1 se považuje za kořen stromu a jeho cena se tedy nastaví na hodnotu 0. O jiných směrovačích prozatím nemá informace, metrika se tedy nastaví na hodnotu nekonečna.
   - $U = \{R1, R2, R3, R4\}$,
   - $V = \{\}$.
2. V druhém kroku se vyhodnotí metrika k přímým sousedům. Cena cesty ke směrovači R2 odpovídá hodnotě 781, cena ke směrovači R3 hodnotě 64. R1 následně zvolí souseda s nejnižší metrikou (z množiny U) a přidá se do množiny V. Nejnižší metrika se rovná hodnotě 64, zvolí se tedy směrovač R3.
   - $U = \{R2, R3, R4\}$,
   - $V = \{R1\}$.
3. Jako jediný nový soused se objevuje R4. Kumulativní cena odpovídá hodnotě 128. Tato metrika je stále menší než cena ke směrovači R2, jako další směrovač se tedy volí R4.
   - $U = \{R2, R4\}$,
   - $V = \{R1, R3\}$.
4. Směrovač R4 sousedí se směrovačem R2 a celková cena pro jeho dosažení se rovná hodnotě 192. Tato metrika je nižší než metrika doposud udržovaná, proto se nahradí (zvolí se lepší cesta).
   - $U = \{R2\}$,
   - $V = \{R1, R3, R4\}$.
5. V posledním kroku se přejde na směrovač R2. Ten již nedisponuje žádnými novými sousedy ani informacemi, proto se výpočty ukončí. Algoritmus aktuálně vyhodnotil nejrychlejší cesty ke všem směrovačům, které dosadí do směrovací tabulky (viz Tab. 4.13). Pro dosažení lokální sítě směrovače R2 se přičte k celkové metrice její cena.
   - $U = \{\}$,
   - $V = \{R1, R2, R3, R4\}$.

Tab. 4.12: SPF algoritmus na směrovači R1.

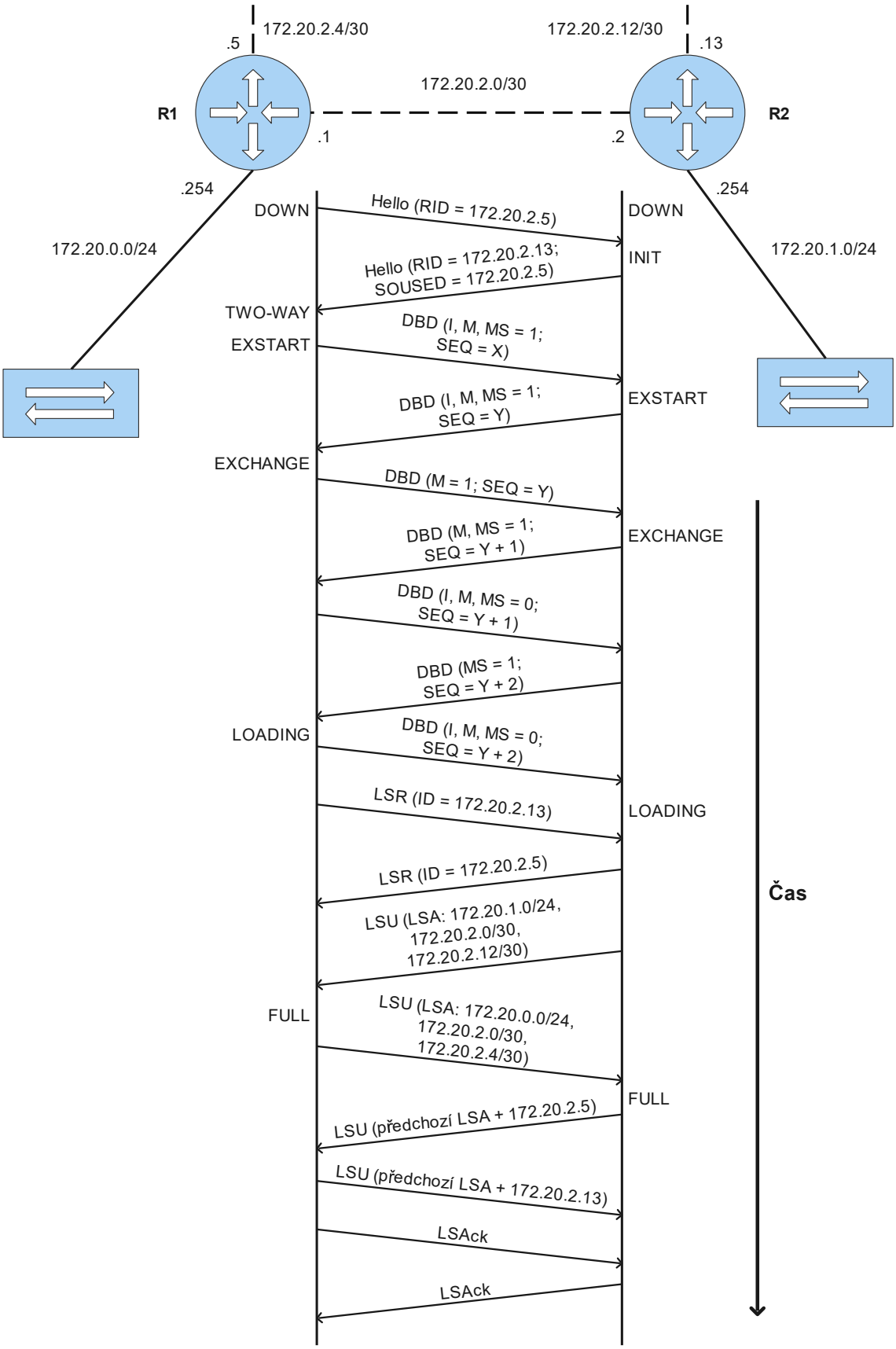
Směrovač	**Krok SPF algoritmu**				
	1	2	3	4	5
R1	0	0	0	0	0
R2	∞	781	781	192	192
R3	∞	64	64	64	64
R4	∞	∞	128	128	128

Tab. 4.13: Obsah směrovací tabulky na R1 po dokončení SPF algoritmu.

Kód	**Adresa sítě**	**Prefix**	**Metrika**	**Next hop**
C	172.20.0.0	/24	0	-
O	172.20.1.0	/24	193	R3
C	172.20.2.0	/30	0	-
C	172.20.2.4	/30	0	-
O	172.20.2.8	/30	128	R3
O	172.20.2.12	/30	192	R3



Obr. 4.15: OSPF cena spojů.



Obr. 4.16: OSPF komunikace.

4.4 Protokol UDP

UDP (User Datagram Protocol) patří mezi dva hlavní protokoly pracující na transportní vrstvě. Roku 1980 byl standardizován v dokumentu RFC 768 [49]. S výhodou jej využívají aplikace, jejichž požadavky se soustředí na nízkou režii ve formě dodatečných dat (záhlaví) a rychlost komunikace. Protokol poskytuje služby aplikacím (procesům), které vyžadují přenos informací přes síť. Přidělí jim jedinečný číselný identifikátor nazývaný **port**. Číslo portu jednoznačně určuje proces komunikující stanice. Společně s IP adresou, která identifikuje stanici v rámci sítě, dochází k jednoznačnému určení procesu napříč celou sítí. Kombinace **IP adresa:port** se označuje pojmem **socket**.

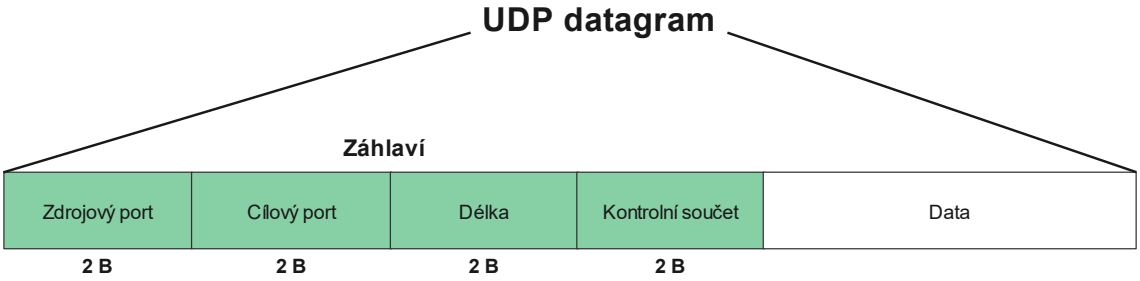
Protokol UDP nabízí procesům nespolehlivý a nespojovaný charakter služeb. Nespolehlivost představuje fakt, že veškerá data odesílaná prostřednictvím UDP nejsou zpětně potvrzována odesílateli. Zdroj tedy odesílá zprávy a nezajímá se, zda byly skutečně doručeny. Tato starost se přenáší na samotné aplikace, které mohou neúspěšný přenos detekovat např. nepřijetím patřičné odpovědi (schéma výzva-odpověď). Nespojovaný charakter znamená, že zdroj před vysíláním dat neověřuje, zda je cíl připraven k příjmu dat ani jeho samotnou existenci. Informace se jednoduše vyšlou bez jakýchkoliv mechanismů pro řízení provozu.

UDP tedy, podobně jako protokol IP, funguje na principu nejlepší snahy k doručení (*Best-effort*). Neexistuje zde garance (správného) doručení ani opětovné vyslání dat, u nichž vznikly chyby (na přijímací straně neodpovídá vypočtená hodnota kontrolního součtu).

Služby nespolehlivé a bez spojení i přes zjevné počáteční nevýhody přináší řadu značně výhodných vlastností, které využijí k provozu dnes zřejmě nejrozšířenější síťové služby – **multimédia**. S ohledem na skutečnost, že příjemci multimediálních služeb jsou lidé, kladou se velké požadavky na rychlost a jistá míra ztrátovosti se připouští (lidské smysly nejsou dokonalé a mozek se dokáže vyrovnat s určitou ztrátou obrazových a zvukových dat). Pokud by se zajistil spolehlivý přenos, opětovně přenesená data by již byla zastaralá v době jejich přijetí (video či audionahrávky pokračují v reálném čase). UDP dále umožňuje všesměrový a skupinový přenos dat, což lze považovat za základní stavební kámen multimédií. Zdroj odesílá data a za replikaci zodpovídají směrovače po cestě do destinací. Této skutečnosti využívají i směrovací protokoly, které si všesměrově nebo skupinově vyměňují směrovací informace.

4.4.1 Struktura UDP datagramu

UDP datagram se skládá z UDP záhlaví (8 B) a aplikačních dat. Jedná se o minimální režii potřebnou k přenosu. Po přidání záhlaví k datům se datagram přesouvá ke zpracování na síťovou vrstvu. Zde se do položky pole protokolu přidá číselný kód (pro UDP dekadické číslo 17), aby bylo patrné, jak se má příchozí paket zpracovat na straně příjemce. UDP záhlaví obsahuje následující položky (viz Obr. 4.17):



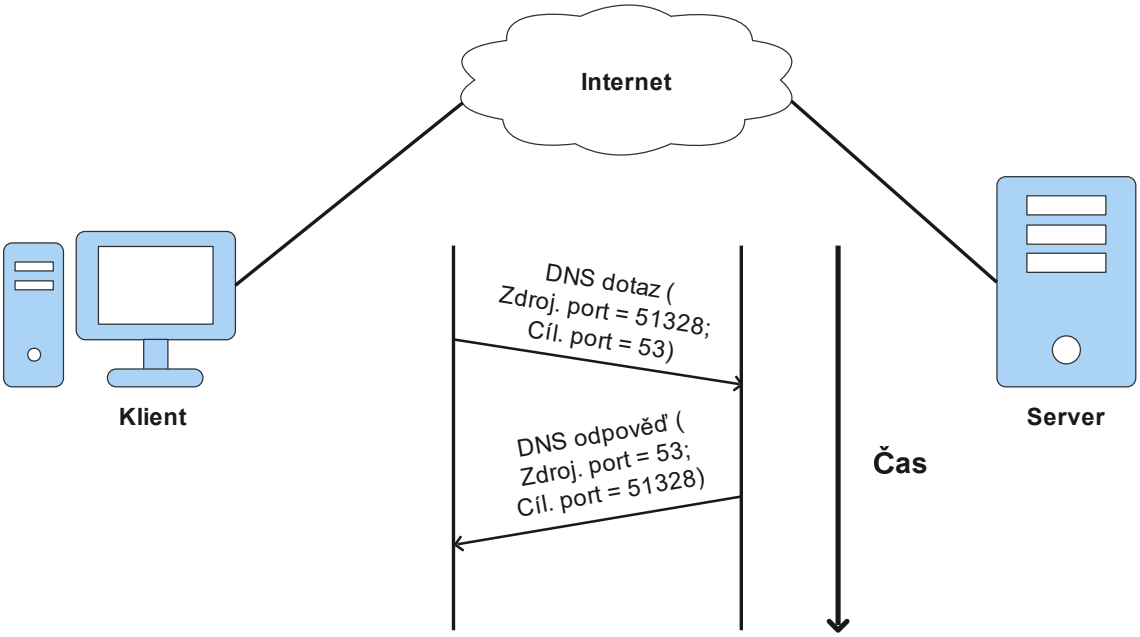
Obr. 4.17: Struktura UDP datagramu.

- **Zdrojový port** – Jedná-li se o klienta, který převážně zahajuje komunikaci se servery, zdrojový port se volí z dynamického nebo registrovaného rozsahu, viz transportní vrstva (2.2). Určuje se tak proces, který si přeje komunikovat po síti. V případě serveru se porty volí nejčastěji z rozsahu známých portů.
- **Cílový port** – Představuje přesný opak zdrojového portu. Cílový port značí proces, s nímž si zdroj přeje komunikovat. Zahajuje-li komunikaci klient, port se často nastavuje na hodnotu z rozsahu známých portů, pokud server odpovídá klientovi, nastavuje se cíl na klientký port.
- **Délka** – Celková délka datagramu, do níž se započítává velikost UDP záhlaví a aplikačních dat.
- **Kontrolní součet** – Volitelná položka v UDP záhlaví. Pokud ji zdroj nevyužije, vyplní se log. 0. Výpočet se provádí nad celým UDP datagramem (záhlaví i data) a tzv. **pseudozáhlavím**. To se vytvoří „předgenerováním“ IP záhlaví a obsahuje:
  - zdrojovou IP adresu,
  - cílovou IP adresu,
  - pole protokolu (IP záhlaví),
  - délku UDP datagramu (v bytech),
  - 1 byte nul (doplnění délky pseudozáhlaví na násobek čtyř).

  Pro případ, kdy zdroj vypočítá kontrolní součet a výsledek se rovná 0, provede se bitová inverze (pro odlišení případu, kdy zdroj nevyužívá možnosti kontrolního součtu) a všechny bity v součtu se nastaví na hodnotu log. 1 [49], [50], [52].

4.4.2 Komunikace aplikací prostřednictvím UDP

Pro demonstraci fungování UDP protokolu se využije aplikační protokol DNS (Domain Name System). Ten však nefunguje výhradně ve spojení s UDP, za jistých okolností se používá i TCP. Nyní však bude popsána jednoduchá komunikace výzva-odpověď založená na systému klient-server. Přenos ve dvou krocích probíhá následovně (viz Obr. 4.18):



Obr. 4.18: Komunikace prostřednictvím UDP.

1. Klient se dotazuje na IP adresu webového serveru, komunikuje tedy prostřednictvím DNS protokolu. Aplikační data s dotazem se přesunou na transportní vrstvu. Zde se k datům přidá UDP záhlaví a vytvoří se tak UDP datagram. Ten se skládá z následujících položek:
   - **Zdrojový port** – Pro klienta se zvolí dostupný port z dynamického rozsahu, např. 51328.
   - **Cílový port** – Komunikuje se prostřednictvím standardizovaného protokolu, cílový port se tedy zvolí ze známého rozsahu – 53.
   - **Délka** – Určité číslo (podle obsahu dotazu).
   - **Kontrolní součet** – Určité číslo (podle obsahu dotazu).

   Takto vytvořený segment se dále zapouzdřuje přes nižší vrstvy a nakonec se odešle.
2. Server přijme paket a rozbalí jeho obsah až na úroveň transportní vrstvy. V případě, že odpovídá kontrolní součet, zjistí cílový port a odesílá data příslušnému procesu.

3. Proces na serveru odpovídá klientovi. Aplikační data přichází na transportní vrstvu, kde se opět vytvoří UDP datagram s následujícími položkami v záhlaví:
   - **Zdrojový port** – Použijí se totožné porty, s nimiž byl dotaz doručen, ovšem vzájemně se vymění. Zdrojový port je tedy nastaven na hodnotu 53.
   - **Cílový port** – Nastaví se na 51328.
   - **Délka** – Určité číslo (podle obsahu dotazu).
   - **Kontrolní součet** – Určité číslo (podle obsahu dotazu).

   Datagram obdobně prochází zapouzdřením přes nižší vrstvy a odesílá se klientovi.
4. Klient přijme paket a následně probíhá stejný proces rozbalování dat až na transportní vrstvu, kde se ověří kontrolní součet. Pokud odpovídá, zjistí se proces, kterému je segment určen a odešle se až na aplikační vrstvu.

4.4.3 Využití UDP

Jak již bylo zmíněno v úvodu, protokol UDP se využívá pro multimediální přenosy. Prostřednictvím něj lze realizovat všesměrové a skupinové vysílání, což je kritické pro řadu služeb. Dále se používá pro šíření směrovacích informací (ze stejného důvodu podpory vysílání). Běžné uživatelské aplikace se vydávají tímto směrem i z důvodu, že protokol představuje malou zátěž pro síť (nízká režie – malá velikost záhlaví) a rychlou variantu komunikace. Bez nutnosti řízení toku, navazování, potvrzování a ukončování spojení lze jednoduše vyslat výzvu a očekávat odpověď. Pokud nedorazí odpověď do určitého okamžiku, vyšle aplikace opětovnou výzvu. Typickým příkladem může být výše zmíněný protokol DNS. Pro jednoduché klientské výzvy a odpovědi (každou lze přenést v 1 zprávě), které se odesílají nepravidelně, se výhodně využije protokol UDP. Ovšem pro náročnější přenosy, např. přenosy DNS záznamů mezi zónami, se využije protokol TCP.

4.5 Protokol TCP

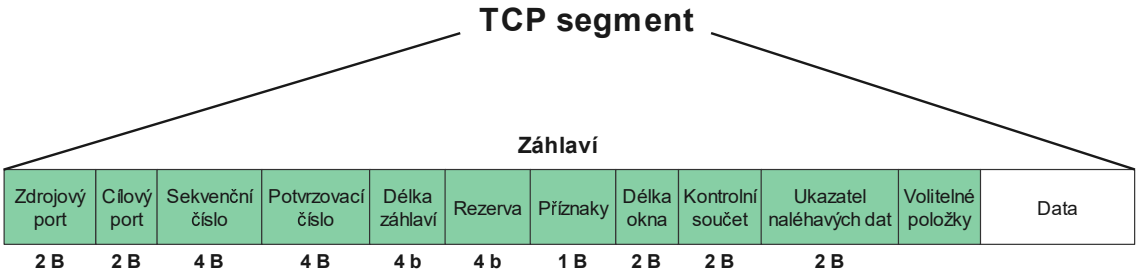
TCP (Transmission Control Protocol) představuje druhý nejdůležitější protokol na transportní vrstvě. V základu jej definuje standard RFC 793 [51], existuje však řada dalších dokumentů popisujících konkrétní mechanismy. Stejně jako UDP využívá TCP čísla portů k jednoznačné identifikaci procesů komunikujících stran. Důležitým faktem je, že porty TCP i UDP nabývají stejného číselného rozsahu, ale vzájemně jsou nezávislé. Běžně se pro standardizované aplikace (z rozsahu známých portů) využívá totožná hodnota UDP i TCP, i když se používá pouze jeden z protokolů (avšak konkrétní implementace aplikačního protokolu může využívat oba v závislosti na dané situaci).

Rozdíl mezi TCP a UDP se projevuje již při zpracování aplikačních dat. UDP přijímá bloky dat, z nichž formuje datagramy. TCP dokáže pracovat s konstantním proudem dat (bytů) a ty dále dělit na **segmenty**. Tato vlastnost se označuje pojmem segmentace. Na rozdíl od protokolu UDP představuje TCP **spolehlivý** a **spojovaný** protokol. Spolehlivost spočívá v potvrzování veškerých správně přijatých dat. Pokud dojde k situaci, kdy data byla po cestě ztracena či poškozena, vysílač je opětovně odešle. Tímto způsobem se zaručuje doručení veškerých dat koncovému příjemci. Spojovaný charakter představuje vlastnost, kdy před odesíláním uživatelských dat musí být nejdříve sestaveno spojení (virtuální okruh) mezi komunikujícími stranami. Tím se zaručuje dostupnost přijímače a schopnost přijímat data. TCP vytváří jednosměrná spojení, pro obousměrnou komunikaci tedy musí být vytvořena dvě jednosměrná spojení (např. klient $\rightarrow$ server, server $\rightarrow$ klient). Každé spojení mezi procesy se jednoznačně určuje **dvojicí socketů**, které identifikují komunikující strany. Po vyslání všech potřebných dat se spojení ukončí. TCP dále poskytuje mechanismus k řízení toku dat, který se realizuje prostřednictvím oken (viz kapitola 4.5.3).

Lze konstatovat, že většina dnešních aplikací využívá protokol TCP právě z důvodu jeho spolehlivosti, kde musí být zaručeno správné doručení veškerých dat. Jedná se převážně o datové aplikace, např. přenos informací mezi zónami (DNS), komunikace s webovými servery (HTTP), odesílání elektronické pošty (SMTP) atp. Ve všech těchto případech není primárním kritériem rychlost přenosu, ale správnost a úplnost veškerých přenesených dat.

4.5.1 Struktura TCP segmentu

Stejně jako v případě UDP datagramu se TCP segment skládá ze záhlaví a aplikačních dat. Ale již z podstaty protokolu vyplývá, že záhlaví musí přenášet více údajů a představuje tedy vyšší režii. Celkově se jedná o 20 bytů zahrnujících povinnou část. Strukturu lze vidět na Obr. 4.19.



Obr. 4.19: Struktura TCP segmentu [51], [52].

Záhlaví obsahuje následující položky:

- **Zdrojový port** – Číslo portu představující stranu odesílající data (totožný princip jako u UDP).
- **Cílový port** – Číslo portu identifikující stranu, pro niž jsou data určena.
- **Sekvenční číslo** – Protože TCP nezpracovává aplikační data na úrovni zpráv ale bytů, identifikují se číslem jednotlivé byty. Sekvenční číslo určuje pořadové číslo 1. bytu, který se v segmentu přenáší. Existuje však výjimka. Když je nastaven příznak SYN (viz dále), který určuje počáteční číslování ISN (Initial Sequence Number), data (byty) se přenáší od hodnoty SYN + 1. Vzhledem k omezené velikosti (4 B) se při dosažení maximální hodnoty číslování resetuje.
- **Potvrzovací číslo** – Označuje hodnotu 1. bytu, který příjemce očekává a zároveň tímto potvrzuje všechny byty s nižší hodnotou sekvenčního čísla než tato. Během přenosu dat (po ustanovení spojení) se potvrzovací čísla vždy vysílají. To však neplatí pro první segment během navazování spojení.
- **Délka záhlaví** – Vyjadřuje počet 32-bitových položek v TCP záhlaví.
- **Rezerva** – Rezerva pro budoucí využití[11].
- **Příznaky** – Příznaky představují důležitou položku v záhlaví. Pomocí nich lze určit význam segmentu z pohledu stavu komunikace (navazování, průběh, ukončování spojení). Současně se jedná o 8 bitů, kde každý představuje 1 příznak:
  - **ECE** – Pokud dojde k zahlcení sítě a hrozí zahazování paketů, příjemce

[11]Dokument RFC 793 definuje rezervu 6 bitů, avšak nově v dokumentu RFC 3168 [53] byly přidány dva nové příznaky (ECN, CWR), rezerva se tedy zmenšila na 4 bity.

informuje odesílatele o této skutečnosti nastavením příznaku ECE (ECN-Echo) na hodnotu log. 1[12].

  - **CWR** – Odesílatel potvrzuje TCP segment s nastaveným příznakem ECE segmentem s příznakem CWR a snižuje velikost okna. Tímto se u příjemce ukončuje označování segmentů příznaky ECE [52], [54].
  - **URG** (*Urgent*) – Nastaví-li se na hodnotu log. 1, indikuje přenos naléhavých dat a validní položku ukazatele naléhavých dat.
  - **ACK** (*Acknowledgment*) – Hodnota log. 1 vyjadřuje potvrzení přijetí dat v položce potvrzovací číslo. Kromě zcela první zprávy při navazování spojení (s příznakem SYN) a stavu jednosměrné komunikace se příznak vždy nastavuje na 1.
  - **PSH** (*Push*) – Nastavení log. 1 značí u odesílatele okamžité vyslání segmentu bez čekání na další aplikační data a u příjemce, že má data ze segmentu ihned předat patřičné aplikaci.
  - **RST** (*Reset*) – Tento příznak, nastavený na hodnotu log. 1, vyjadřuje zamítnutí TCP spojení nebo rychlé ukončení již ustanoveného spojení.
  - **SYN** (*Synchronize*) – Nastavuje se na log. 1 při požadavku na ustanovení spojení. Společně s příznakem SYN se nastavuje i počáteční sekvenční číslo (ISN).
  - **FIN** (*Final*) – Tímto příznakem odesílatel sděluje protější straně, že vyslal všechna aplikační data a dále již žádná nebude posílat. Odesílatel tedy již jen potvrzuje přijatá data od protějšího uzlu a čeká na ukončení vysílání z jeho strany.
- **Délka okna** – Představuje počet bytů, který příjemce dokáže zpracovat (odpovídá velikosti vyrovnávací paměti pro dané spojení). Počáteční byte dat se určuje na základě potvrzovacího čísla.
- **Kontrolní součet** – Počítá se z TCP záhlaví, aplikačních dat a pseudozáhlaví (stejně jako u protokolu UDP). Výpočet se odehrává nad sudým počtem bytů, v případě lichého počtu se tedy musí přidat 1 byte nul na konec dat.
- **Ukazatel naléhavých dat** – Nastaví-li se příznak URG, představuje tato položka rozsah od sekvenčního čísla (1. přenášený byte v segmentu) až po hodnotu ukazatele. Uvnitř tohoto rozsahu se nachází data interpretovaná jako naléhavá.
- **Volitelné položky** – Představují volitelnou část TCP záhlaví, která může a nemusí být využita. Jedna položka se skládá ze tří částí: typ, délka a hodnota. Každá část zabírá prostor v násobcích 1 bytu. Jako příklad lze uvést MSS (Maximum Segment Size – maximální délka segmentu) s hodnotou typu 2 a délkou

[12] Podmínkou využití příznaků ECE a CWR je podpora mechanismu ECN oběma komunikujícími stranami a všemi mezilehlými zařízeními [54].

4 byty. Na počátku spojení (zpráva odeslaná s příznakem SYN) se touto položkou stanovuje maximální délka segmentu (datová část bez záhlaví), kterou přijímač může zpracovat. Každá komunikující strana v počáteční zprávě může odeslat vlastní MSS. Pokud strany MSS nevyužijí, indikuje se protějškům, že mají vysílat segmenty maximálně výchozí délky (536 bytů) [51], [52].

4.5.2 Komunikace aplikací prostřednictvím protokolu TCP

Vzhledem ke spojovanému charakteru protokolu TCP dochází ke třem událostem během komunikace – navázání spojení, přenos dat, ukončení spojení. Všechny tyto události provází stavy, v nichž se může TCP nacházet:

- **Stav uzavření** (*CLOSED*) – Jedná se o stav, kdy se žádné spojení neudržuje ani nenavazuje[13].
- **Stav naslouchání** (*LISTEN*) – Zařízení očekává příchozí požadavek ke spojení na určitém TCP portu.
- **Stav odeslání požadavku** (*SYN-SENT*) – Zařízení odesílá požadavek ke spojení s nastaveným příznakem SYN jinému zařízení. Dále očekává přijetí požadavku od protějšku s potvrzením přijetí vlastního požadavku (SYN + ACK).
- **Stav přijetí požadavku** (*SYN-RECEIVED*) – Zařízení přijalo i odeslalo požadavek ke spojení. Od druhé komunikující strany se očekává potvrzení přijetí vlastního požadavku (příznak ACK).
- **Stav komunikace** (*ESTABLISHED*) – Spojení mezi zařízeními se úspěšně navázalo a nyní se mohou obousměrně přenášet aplikační data v segmentech.
- **Stav aktivního ukončení** (*FIN-WAIT-1*) – Zařízení nedisponuje žádnými dalšími daty k odeslání, proto odesílá zprávu (požadavek) k ukončení spojení s nastaveným příznakem FIN. Dále očekává potvrzení přijetí této zprávy (*FIN-WAIT-2*) nebo samostatný požadavek k ukončení spojení od protější strany (*CLOSING*).
- **Stav očekávání ukončení** (*FIN-WAIT-2*) – Zařízením bylo přijato potvrzení požadavku k ukončení spojení a následně se očekává zpráva s příznakem FIN od druhé strany, která ještě odesílá data (po přijetí této zprávy se přechází do stavu *TIME-WAIT*).
- **Stav očekávání lokálního ukončení** (*CLOSE-WAIT*) – Zařízení obdrželo zprávu s příznakem FIN, ale lokální proces stále disponuje aplikačními daty

[13] Z tohoto stavu se přechází dále v závislosti na roli zařízení. Typicky se pro klientskou část aktivně přechází do navazování spojení ve stavu *SYN-SENT*. U serveru se naopak pasivně očekává spojení způsobem, že se port připraví pro přijetí požadavků ke spojení. Dostává se tedy do stavu *LISTEN*. Aktivní i pasivní přechod doprovází vytvoření struktury TCB (Transmission Control Block), která uchovává důležité informace týkající se spojení.

k odeslání, proto dochází pouze k jednostrannému ukončení (po dokončení přenosu se přechází do stavu *LAST-ACK*).

- **Stav očekávání potvrzení ukončení** (*CLOSING*) – Zařízení přijalo zprávu s příznakem FIN od protější komunikující strany, odeslalo potvrzení, avšak stále neobdrželo potvrzení na vlastní ukončující zprávu (ze stavu *FIN-WAIT-1*). Po přijetí přechází do stavu *TIME-WAIT*.
- **Stav oboustranného ukončení** (*LAST-ACK*) – Zařízení dokončilo přenos dat a odesílá zprávu s příznakem FIN protějšku, který jednostranně uzavřel spojení. Nyní očekává potvrzení této zprávy. Jakmile ji obdrží, přechází do stavu *CLOSED*.
- **Stav pozdržení** (*TIME-WAIT*) – Zařízení odesílá potvrzení o přijetí zprávy s příznakem FIN od druhé strany, která ve stavu *LAST-ACK* (nebo *CLOSING*) očekává toto potvrzení. Po určitý čas se vyčkává v tomto stavu[14]. Pokud nepřijde opakovaně zpráva značící ukončení (např. kvůli chybnému přenosu či ztrátě segmentu), přechází se do stavu *CLOSED*.

Následující příklad vychází ze standardního modelu klient-server, kdy klient komunikuje se serverem za účelem přenosu webového obsahu (protokol HTTP). Klient komunikuje pod náhodně přiděleným dynamickým portem, např. 53824 a server je dostupný pod známým portem 80. Sekvenční čísla si generuje každá strana samostatně. Jedná se o náhodné 32-bitové hodnoty. Server se v popisu nachází v počátečním stavu *LISTEN*, otevřel tedy pasivně port a očekává klientské připojení. Celý průběh komunikace je zobrazen na Obr. 4.20.

Navázání TCP spojení

1. Klient chce komunikovat se serverem pro získání webového obsahu, pokusí se tedy vytvořit virtuální spojení. Vytvoří TCP segment se zdrojovým portem 53824, cílovým portem 80 a dále vygeneruje náhodné sekvenční číslo X, které reprezentuje číslování bytů vysílaných od něj (ISN). Protože se jedná o první segment v komunikaci, nastaví se příznak SYN, a pole potvrzovací číslo se nastaví na hodnotu 0 (žádná data se nepotvrzují). Nastaví se velikost okna a standardně i volitelná položka MSS (pouze u segmentů s příznakem SYN). Přechází se ze stavu *CLOSED* do *SYN-SENT*.
2. Server přijme zprávu s příznakem SYN a dostává se ze stavu *LISTEN* do *SYN-RECEIVED*. Protože žádost o spojení akceptuje, vytvoří vlastní segment se zaměněným zdrojovým a cílovým portem. Vygeneruje náhodné sekvenční

[14] Jedná se o dvojnásobek doby trvání MSL (Maximum Segment Lifetime), jež se v [51] definuje jako dvě minuty.

číslo Y, vlastní velikost okna a MSS. Z příznaků se nastavuje SYN (první segment od serveru) a ACK, čímž se potvrzuje přijetí segmentu od klienta. Pole potvrzovací číslo se nastaví na hodnotu X + 1 (další očekávané sekvenční číslo). Po doručení potvrzení klientovi se vytvoří první jednosměrné spojení.

3. Klient přijme segment od serveru, v němž se dozvídá o číslování bytů serveru (položka *sekvenční číslo*) a zároveň obdrží potvrzení své předešle odeslané zprávy s příznakem SYN (položka *potvrzovací číslo*). Přechází tedy do stavu *ESTABLISHED*. Nyní vyšle potvrzení o přijetí, v němž nastaví pouze příznak ACK. Sekvenční číslo nastaví na hodnotu X + 1 a potvrzovací číslo na Y + 1.
4. Poté, co server přijme potvrzení od klienta, přechází také do stavu *ESTABLISHED*. Vytvoří se tedy druhé jednosměrné spojení.
5. Celkové navazování spojení, nazývané též *Three-way handshake*, se tedy skládá z výměny tří zpráv, obsahujících následující příznaky:
   - **Klient** – [SYN],
   - **Server** – [SYN, ACK],
   - **Klient** – [ACK].

Komunikace procesů

1. Po přechodu do stavu *ESTABLISHED* se ustanovila obousměrná komunikace pomocí virtuálního okruhu a obě strany tedy mohou vysílat aplikační data.
2. Číslování datových bytů začíná na straně klienta číslem X + 1, na straně serveru Y + 1.
3. Aplikační data (byty) se v segmentech přenáší s ohledem na hodnotu MSS vyjednanou při ustanovení spojení. Při potvrzování se zohledňuje velikost oken (viz sekce 4.5.3), která se v průběhu komunikace může měnit na obou stranách.

Ukončení TCP spojení

1. Ve chvíli, kdy jeden z komunikujících procesů vyhodnotí, že již odeslal všechna potřebná data, dává TCP pokyn k ukončení spojení. To se projeví nastavením příznaku FIN v následujícím segmentu. Na Obr. 4.20 ukončuje spojení server. Vysílá tedy segment s potvrzením předešle přijatých dat od klienta (příznak ACK) a vysílá požadavek k ukončení spojení (příznak FIN). Po odeslání segmentu se server dostává do stavu *FIN-WAIT-1*.
2. Klient obdrží tuto zprávu a obratem ji potvrdí. Pokud jeho lokální proces stále udržuje aplikační data k odeslání, nenastavuje příznak FIN[15]. Klient přechází

[15] V takovém případě klient stále posílá data, ovšem server tato data již jen potvrzuje, žádná vlastní aplikační data klientovi nezasílá.

do stavu *CLOSE-WAIT*. Server přechází po přijetí potvrzení do stavu *FIN-WAIT-2*.

3. Poté, co i proces u klienta dokončí vysílání veškerých potřebných dat, ukončuje spojení. Vytváří se tedy segment s příznakem FIN a klient přechází do stavu *LAST-ACK*.
4. Server přijímá zprávu k ukončení spojení ze strany klienta, vyšle tedy potvrzení o jejím přijetí. V tuto chvíli přechází do stavu *TIME-WAIT* pro případ, že by klient potvrzení neobdržel a opětovně vyslal zprávu. Po vypršení časovače přechází server do stavu *CLOSED*.
5. Klient přijme potvrzení od serveru, čímž považuje komunikaci za kompletně ukončenou a přechází do stavu *CLOSED*.
6. Takto ukončené spojení se označuje pojmem *Four-way handshake* [51], [52], [55].

4.5.3 Metoda klouzavých oken

TCP zpracovává aplikační data po bytech, kterým přiděluje unikátní sekvenční čísla. Vysílat a následně potvrzovat jednotlivé byty by z přenosového hlediska bylo značně neefektivní, proto se přistupuje k mechanismu odesílání kontinuálního rozsahu bytů. Tento princip se nazývá metoda klouzavého okna (*Sliding Window*). Okno se po potvrzení posunuje, s ohledem na velikost uvedenou v záhlaví příjemce, na další navazující rozsah bytů.

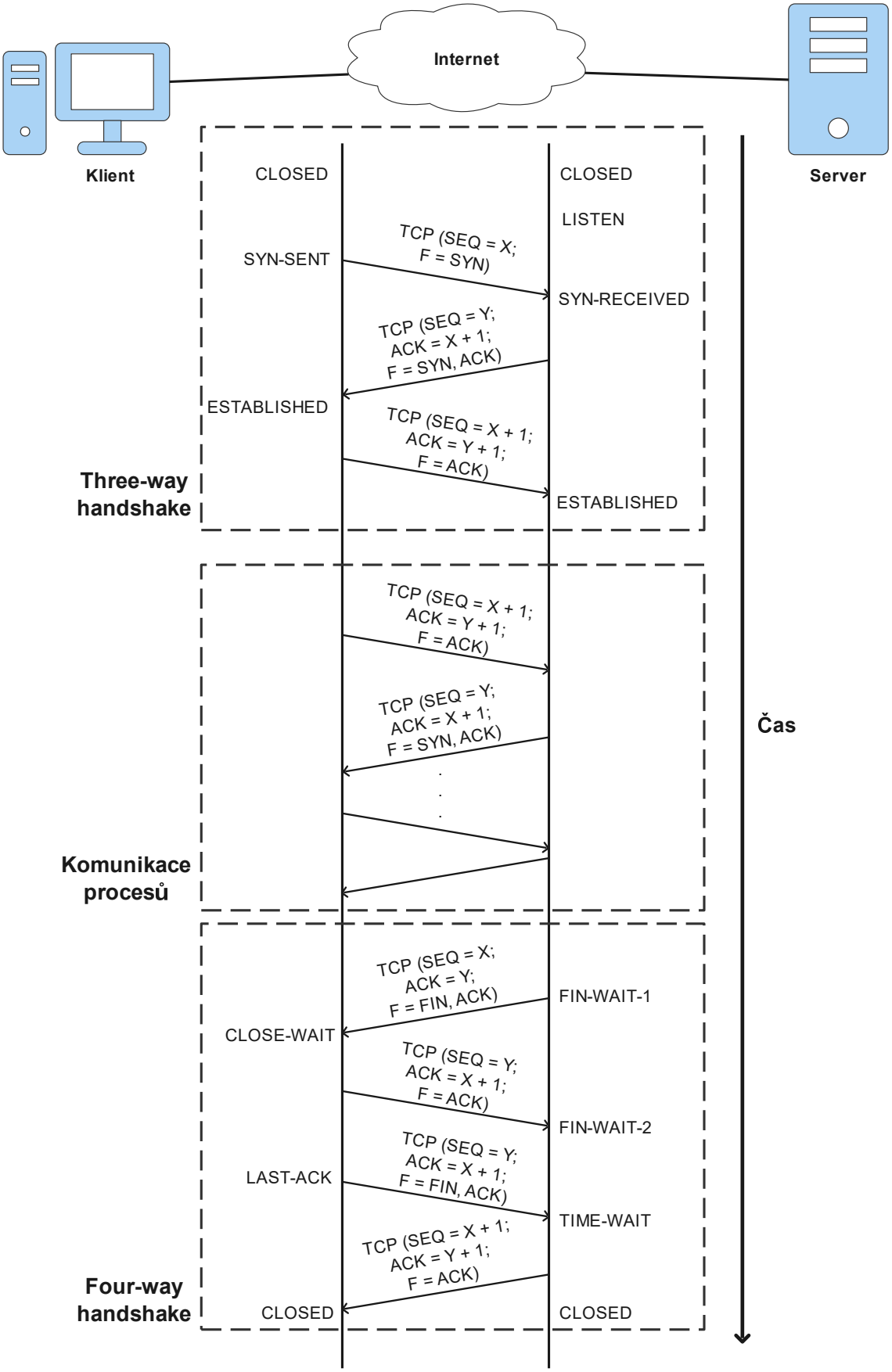
Každá z komunikujících stran specifikuje vlastní velikost okna, jež se může za běhu měnit. Velikost okna na straně klienta tedy znamená maximální počet bytů, které může server odeslat bez potvrzení[16]. Komunikující strany dále udržují informace o odeslaných a očekávaných příchozích datech. S ohledem na další položky v záhlaví tedy sekvenční číslo označuje první byte odeslaných dat, délka okna označuje celkový počet bytů, které lze odeslat a potvrzovací číslo označuje další očekávaný byte společně s potvrzením veškerých předchozích bytů (sekvenčních čísel).

Komunikace tedy probíhá s přísným sledováním mnoha parametrů. Server odesílá segmenty, jejichž velikost nepřesahuje hodnotu MSS, a zároveň nepřekračuje velikost okna, to vše stanovené klientem (stejný princip se dodržuje i v opačném směru). V případě, kdy se velikost okna vyčerpala, musí server před dalším vysíláním aplikačních dat vyčkat na potvrzení od klienta, čímž se klouzavé okno posouvá na další rozsah. Podobná situace nastává v případě, kdy klient propaguje k serveru velikost okna 0. Tento stav značí, že vyrovnávací paměť klienta je zahlcena a další přijatá data by byla zahozena. V tuto chvíli se opět pozastaví vysílání, dokud klient

[16]Ve spojení s hodnotou MSS, ustanovené na začátku komunikace, může značit počet segmentů, pokud podíl velikosti okna a MSS představuje celočíselnou hodnotu.

neoznámí novou velikost okna. Ta může být nižší oproti předchozí hodnotě (před velikostí okna 0), aby se předešlo opakovanému zahlcení. Po ustálení situace se velikost okna může dále zvětšovat.

V reálných systémech se implementuje další typ okna, které se nazývá okno zahlcení (*Congestion Window*). Používá se z důvodu, aby nedošlo k zahlcení samotné přenosové sítě, což by vedlo k zahazování paketů. Odesílatel vysílá data způsobem, aby jejich množství nepřesáhlo okno zahlcení ani okno příjemce. Běžně se začíná na menších velikostech okna, které se postupem komunikace dále zvětšuje. V případě zahlcení se okno zmenší a opakují se pokusy o zvětšování [52], [55].



Obr. 4.20: TCP komunikace.

4.6 Protokol DHCP

DHCP (Dynamic Host Configuration Protocol) protokol představuje aplikační protokol pro dynamické přidělování adres. Nejnovějším standardem, který jej definuje, je RFC 2131 [56]. V počátcích internetu, kdy se počet zařízení pohyboval v řádu jednotek, nejvýše desítek, se vystačilo se statickou konfigurací prováděnou administrátory. Značný nárust počtu uživatelských stanic, spojený s vývojem nových technologií, vyžadoval odlišný přístup s jistou dávkou automatizace, protože manuální konfigurace stovek stanic představuje jak značnou časovou náročnost, tak zvýšení pravděpodobnosti výskytu chyb způsobených lidským faktorem.

Zcela prvním přístupem automatického přidělování adres se stal **RARP** (Reverse Address Resolution Protocol). Tento protokol vyžaduje přítomnost serveru, který udržuje záznamy o párech IP-MAC. Účastník se dotazuje na svou logickou adresu na základě své fyzické adresy. Jedná se tedy o opačný proces protokolu ARP. Limitací se stala skutečnost, že protokol vyžaduje zvláštní RARP server v každé podsíti a nepřiřazuje žádné další údaje. RARP byl postupným vývojem nahrazen novějším protokolem **BOOTP** (Bootstrap Protocol). Tento protokol již umožňuje udržovat jeden centrální server, avšak stále se jedná o pevně zadané kombinace párů IP-MAC. BOOTP již představoval zásadní význam v oblasti automatizace přidělování adres (přidělování i jiných údajů než IP adresy stanice – např. výchozí brána), proto došlo k vývoji nového protokolu, který v určitém směru BOOTP pouze zdokonaluje. Nejnovějším, dnes téměř základním protokolem v každé síti, je **DHCP** [57], [58].

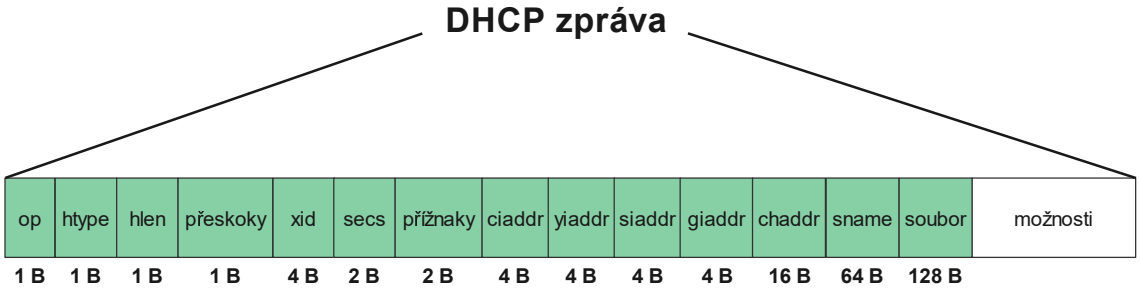
DHCP server přiděluje adresy standardně dynamicky[17] na určitou dobu, tzv. **dobu zápůjčky** (*lease time*). Tento parametr nastavuje administrátor na serveru a určuje tím, jak dlouho může klient danou konfiguraci využívat. Po uplynutí této doby musí klient adresu přestat používat a opětovně kontaktovat server pro nové přidělení konfigurace. Běžně však klient kontaktuje server před vypršením doby zápůjčky a obnovuje tento časový interval (nebo žádá o jiný, rozhodnutí připadá serveru), čímž může nadále využívat totožnou konfiguraci nepřerušovaně. Server přiděluje primárně neobsazené adresy z tzv. **bazénu** (*pool*), který definuje rozsah adres. Po vyčerpání celého rozsahu server začne prohledávat seznam dříve přidělených adres, kterým vypršela doba zápůjčky a klienti nezažádali o obnovení. Pokud server nedisponuje žádnými volnými adresami z bazénu a seznam dříve zapůjčených adres, jimž vypršel časovač, je prázdný, klienti nezískají konfiguraci a musí vyčkat na odpojení některé z aktivních stanic. Doba zápůjčky se standardně nastavuje řádově v hodinách či dnech (závisí na typu sítě).

[17] Dalšími možnostmi jsou například přidělení adres na základě informací o klientovi (fyzická adresa) nebo tzv. trvalé přidělení, kdy se klient po přidělení konfigurace pevně sváže s danou adresou.

Pokud se DHCP server nachází ve stejném síťovém segmentu jako klient, který vyžaduje konfiguraci, probíhá komunikace přímo. V případě rozvětvené sítě, která obsahuje větší počet samostatných podsítí a klient se nachází v jiné podsíti než DHCP server, musí být komunikace zprostředkována tzv. **přenosovým agentem** (*relay agent*). Jedná se o směrovač, který se nakonfiguruje do role prostředníka, jenž přeposílá DHCP zprávy mezi příslušnými podsítěmi. Směrovač přidá ke zprávám informace, které jednoznačně identifikují umístění klienta a server na základě těchto informací přiřazuje síťovou konfiguraci dle svého nastavení patřičné podsítě. Při komunikaci se využívá model klient-server [58].

4.6.1 Struktura DHCP zprávy

Protokol DHCP využívá ke svému šíření transportní protokol UDP. Z rozsahu známých portů jsou protokolu alokovány dva porty – **67** a **68**. UDP port 67 nastavuje klient jako cílový port při komunikaci s DHCP serverem (port tedy označuje proces na serveru). UDP port 68 využívá klient jako zdrojový port. V odpovědích serveru se porty vzájemně prohodí. Strukturu DHCP zprávy lze vidět na Obr. 4.21. Zpráva obsahuje pevně dané a proměnlivé položky (možnosti), které se liší s typem zpráv. Každá možnost vychází ze společného formátu, kde se definuje kód, délka a samotná data. Protokol obsahuje řadu položek, které slouží k předání informací bezdiskovým stanicím, aby mohly být správně spuštěny. Při běžné klientské konfiguraci se tyto položky nevyužívají.



Obr. 4.21: Struktura DHCP zprávy [58], [59].

Konkrétně se jedná o následující položky:

- **op** – Jedná se o operační kód zprávy, který určuje její typ. Může nabývat dvou hodnot:
  - **1** – Kód označuje DHCP žádost.
  - **2** – Kód označuje DHCP odpověď.
- **htype** – Stejně jako u protokolu ARP označuje typ linkového protokolu (např. pro Ethernet se nastavuje hodnota 1).

- **hlen** – Označuje délku adresy (v bytech) linkového protokolu (v případě Ethernetu se jedná o MAC adresu, tedy hodnota 6).
- **přeskoky** – Vyjadřuje počet přenosových agentů, přes něž byla DHCP zpráva přenesena. Klient implicitně nastavuje hodnotu 0, směrovače v rolích přenosových agentů tuto hodnotu inkrementují.
- **xid** – Identifikátor označující náhodně vygenerované číslo na straně klienta, které jednoznačně určuje transakci mezi klientem a serverem.
- **secs** – Relativní čas ve vteřinách, který určuje dobu od odeslání požadavku na získání adresy.
- **příznaky** – Aktuálně se využívá jediný bit (nejvíce vlevo), který určuje, zda má server vysílat odpovědi všesměrově (hodnota 1) nebo cíleně jedné stanici (*unicast* – hodnota 0).
- **ciaddr** – Hodnota IP adresy klienta, která se nastavuje klientem. Při počáteční komunikaci se serverem, kdy klient nemá přiřazenou žádnou adresu, se položka nastavuje na hodnotu *0.0.0.0*. V případě další komunikace, kdy již klient disponuje přiřazenou adresou, se nastavuje na danou adresu.
- **yiaddr** – Hodnota IP adresy klienta, která se nastavuje serverem. Server, který obdrží požadavek na přiřazení adresy, informuje klienta o potenciální adrese v tomto poli.
- **siaddr** – IP adresa serveru, který má být kontaktován pro další konfiguraci ve smyslu zavedení operačního systému. Tato možnost se využívá v případě bezdiskových stanic.
- **giaddr** – IP adresa směrovače, přes nějž prochází DHCP požadavek od klienta. Směrovač tuto položku nastaví na adresu lokálního rozhraní, na němž došlo k přijetí požadavku. Server následně vyhodnotí, zda se klient nachází ve stejné síti (hodnota *0.0.0.0*), nebo byl při komunikaci použit přenosový agent (IP adresa rozhraní), a podle toho přiděluje patřičnou konfiguraci.
- **chaddr** – Nastavuje se přímo linková adresa klienta (MAC adresa v případě technologie Ethernet).
- **sname** – Obsahuje název serveru, na němž je k dispozici konfigurační soubor (ekvivalentní jméno k položce *siaddr*).
- **soubor** – Název souboru se zavaděčem, který slouží pro spuštění bezdiskové stanice (název souboru uloženého např. na TFTP serveru).
- **možnosti** – Volitelné položky, které se v DHCP zprávách mohou přenášet. V těchto polích se mimo jiné přenáší konfigurační informace mezi klientem a serverem. Níže se uvádí některé z možností:
  - **Typ DHCP zprávy** (*DHCP message type*) – Přidává se ke každé DHCP zprávě a jednoznačně identifikuje, o jakou zprávu se jedná. Definuje se číselným kódem 53 a délkou jednoho bytu. Základní typy zpráv při zís-

kávání jsou nalezení serveru (*DISCOVER* – typ 1), nabídka (*OFFER* – typ 2), požadavek (*REQUEST* – typ 3) a potvrzení (*ACK* – typ 5).

- **Síťová maska** (*Subnet mask*) – Definována kódem 1 a délkou 4 byty. Vyjadřuje síťovou masku spojenou se sítí, v níž se klient nachází.
- **Výchozí brána** (*Router*) – Definována kódem 3 a délkou 4 byty. Jedná se o adresu směrovače (výchozí brány), který slouží pro komunikaci s jinými sítěmi.
- **Doménový server** (*Domain name server*) – Definován kódem 6 a délkou 4 byty. Adresa serveru, který se využije např. k překladu doménových jmen na IP adresy.
- **Požadovaná IP adresa** (*Requested IP address*) – Definována kódem 50 a délkou 4 byty. Zde klient definuje nabízenou IP adresu, již se rozhodl využít[18].
- **Doba zápůjčky IP adresy** (*IP address lease time*) – Definována kódem 51 a délkou 4 byty. Zde server definuje dobu zápůjčky.
- **Identifikátor DHCP serveru** (*DHCP server identifier*) – Definován kódem 54 a délkou 4 byty. Server zde uvádí svou IP adresu.
- **Konec možností** (*End*) – Na konec možností se vkládá jeden byte s dekadickou hodnotou 255 [56], [58], [59].

4.6.2 Komunikace pro přidělení DHCP adres

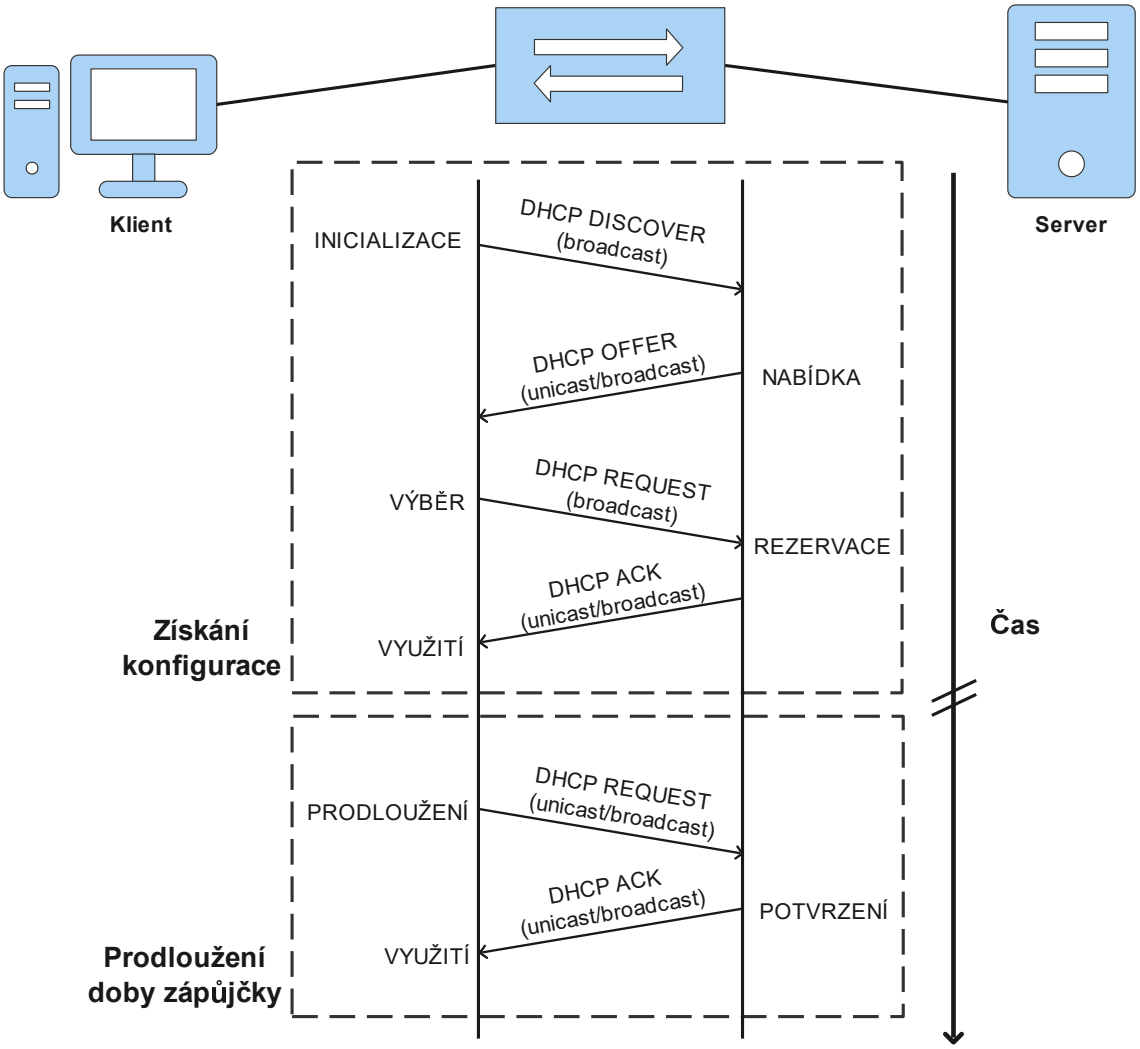
Princip komunikace je zobrazen na Obr. 4.22. Vychází se z případu, kdy se klient i server nachází ve stejné všesměrové doméně. Klient se nově připojil do sítě, neudržuje tedy žádnou předešlou konfiguraci ve své paměti. Pro správné přiřazení konfigurace musí proběhnout následující kroky:

1. Klient vygeneruje zprávu **DHCP Discover**. Touto zprávou informuje všechny DHCP servery ve své síti[19], že chce získat IP konfiguraci. Zde nastaví příslušné položky týkající se vlastní fyzické adresy (*htype*, *hlen*, *chaddr*). Vzhledem k žádné dosavadní konfiguraci uvede položky týkající se IP adres (*ciaddr*, *yiaddr*, *siaddr*, *giaddr*) do výchozí hodnoty, tedy *0.0.0.0*. Totožná hodnota platí i pro požadovanou IP adresu, protože klient neví, v jaké podsíti se nachází. Klient dále vygeneruje náhodné číslo pro identifikaci transakce. Operační kód i typ DHCP zprávy se rovnají hodnotě 1.

   Na úrovni transportní vrstvy nastaví klient zdrojový port na hodnotu 68 a cílový port na hodnotu 67. Na úrovni síťové vrstvy vysílá zprávu na všesměrovou

[18] Případně se zde definuje adresa, kterou klient dříve používal a chce ji obnovit nebo prodloužit dobu zápůjčky.

[19] Pokud se v síti nachází přenosoví agenti, přeposílají zprávu dále DHCP serverům mimo lokální síť.



Obr. 4.22: Proces DHCP komunikace.

adresu *255.255.255.255* ze zdrojové adresy *0.0.0.0*. Z výše uvedeného vyplývá, že rámec vytvořený na linkové vrstvě se odesílá taktéž na všesměrovou adresu (*ff-ff-ff-ff-ff-ff*) se zdrojovou fyzickou adresou klienta.

2. DHCP server přijme žádost od klienta a vygeneruje odpověď ve formě nabídky **DHCP Offer**. Nastavení položek týkajících se fyzické adresy probíhá stejným způsobem jako u klienta[20]. Rozdíl nastává u položek s logickými adresami. Hodnotu *ciaddr* ponechává server výchozí (*0.0.0.0*) a do položky *yiaddr* umístí IP adresu, kterou klientovi zvolí ze svého bazénu. Číslo transakce zachová z přijaté zprávy *DHCP Discover*. Do volitelných položek přidá další potřebné údaje, tedy adresu výchozí brány, masku sítě, DNS server, dobu zápůjčky, identifikátor DHCP serveru atp.

   Na transportní vrstvě zamění hodnoty zdrojového a cílového portu. Paket vy-

[20]Položka *chaddr* však zůstává beze změny na hodnotě fyzické adresy klienta.

tvořený na síťové vrstvě odesílá jako unicast[21] na adresu, kterou vybral klientovi (klient tuto adresu zatím nevyužívá, jedná se o mechanismus pro předejití plýtvání síťovými zdroji, kde by každé zařízení v síti muselo paket zpracovat). Rámec na linkové vrstvě vysílá přímo klientovi, nastaví tedy vlastní zdrojovou MAC adresu a cílovou MAC adresu klienta. Operační kód i typ DHCP zprávy se vysílají s hodnotou 2.

3. Klient obdrží nabídku od serveru, kterou akceptuje, a vygeneruje žádost **DHCP Request**. Touto zprávou sděluje klient serveru (obecně všem DHCP serverům) zájem o danou konfiguraci. Všechny pevně dané položky zůstávají shodné s hodnotami uvnitř zprávy *DHCP Discover*. Uvnitř proměnlivé sekce dochází ke změně, kdy klient žádá již konkrétní IP adresu zaslanou serverem. Nastaví zde zároveň identifikátor DHCP serveru (slouží k odlišení jednotlivých serverů v případě, kdy se v síti nachází více serverů nabízejících své informace). Číslo transakce zůstává stejné.

   Hodnoty portů na transportní vrstvě se opět zamění. Dokud klient neobdrží potvrzení žádosti, nemůže konfiguraci využívat, proto paket odesílá opět se zdrojovou adresou *0.0.0.0* a všesměrovou cílovou adresou *255.255.255.255*. Na linkové vrstvě odesílá rámec stejným způsobem, tedy všesměrově s vlastní zdrojovou fyzickou adresou. Operační kód se nastaví na hodnotu 1 a typ DHCP zprávy na 3.
4. Server akceptuje žádost a odešle definitivní potvrzení **DHCP ACK** (může se však stát, že daná adresa na serveru je již obsazená, v takovém případě server posílá záporné potvrzení **DHCP NAK**), po jehož obdržení může klient začít používat konfiguraci ke komunikaci. Struktura zprávy se prakticky neliší od nabídky (umístí se zde veškeré hodnoty potřebné pro správnou konfiguraci klienta). Server vyhradí nabízenou adresu pro klienta a vytvoří záznam složený z propůjčené IP adresy a MAC adresy klienta. Dále se nastaví stejné číslo transakce.

   Na transportní vrstvě dojde opět k záměně hodnot portů a síťový paket se odesílá danému klientovi jako unicast (se zapůjčenou IP adresou v cílové adrese). Linkový rámec se vysílá stejným způsobem, tedy s konkrétní MAC adresou klienta. Operační kód se nastaví na hodnotu 2 a typ DHCP zprávy na 5.
5. IP adresa se zapůjčuje pouze na definovanou dobu (*lease time*). Po uplynutí této doby musí klient adresu přestat používat a server ji uvolní pro další přidělení. Před uvolněním má však klient možnost zažádat server o prodloužení

[21] Existují případy, kdy server vysílá nabídku všesměrově stejným způsobem jako klient svou žádost. V takovém případě klient v počáteční zprávě nastavuje příslušný příznak, ovšem není to podmínkou.

doby zápůjčky[22].

Provede se tak odesláním zprávy **DHCP Request**. Její obsah se podobá žádosti při počátečním získávání adresy. Klient nastaví položku *ciaddr* na svou aktivní IP adresu, operační kód 1, typ DHCP zprávy 3 a vygeneruje identifikátor transakce. Položka *požadovaná adresa* se neuvádí oproti původní žádosti. Zapouzdřená data uvnitř paketu se odesílají přímo serveru, od nějž byla konfigurace přijata (podle identifikátoru DHCP serveru), tedy jako unicast se zdrojovou IP adresou klienta a cílovou IP adresou serveru. Z toho plyne i adresace na linkové vrstvě, kde se využije zdrojová MAC adresa klienta a cílová MAC adresa serveru.

6. Klient může zažádat o libovolný čas, server však určí na základě svých parametrů (minimální, výchozí, maximální doba zápůjčky) finální dobu a tu zašle klientovi.

   Server přijme žádost, kterou následně potvrdí (pokud je to možné) zprávou **DHCP ACK**. Oproti původní potvrzovací zprávě se definuje adresa klienta *ciaddr*, proměnlivé položky zůstávají totožné. Server využije stejné číslo transakce, operační kód 2 a typ DHCP zprávy 5.

   Výsledné potvrzení adresuje přímo klientské stanici, tedy na síťové i linkové vrstvě definuje konkrétní cílové adresy klienta [58], [61].

4.6.3 Další typy DHCP zpráv

Kromě předchozích zpráv definuje DHCP řadu dalších. Příkladem může být zpráva **DHCP NAK**, kterou zasílá server klientovi v případě, že jeho požadavek nemůže potvrdit (např. při snaze o prodloužení původní adresy po přesunu klientské stanice do prostoru jiné podsítě nebo při expiraci doby zápůjčky). Dále pak zpráva **DHCP RELEASE**, kterou klient informuje server o uvolnění adresy z vlastního rozhodnutí, zpráva **DHCP INFORM**, kterou zasílá klient serveru, aby získal dodatečné konfigurační parametry (disponuje již vlastní manuálně nakonfigurovanou adresou, žádá např. adresu DNS serveru) a další [56], [58].

[22]Doba, po níž by měl klient kontaktovat server s žádostí o prodloužení doby zápůjčky, se stanovuje při počáteční konfiguraci adresy. Žádá se po uplynutí poloviny doby zápůjčky. Pokud server neodpovídá, posílá se žádost všesměrově po uplynutí další doby (87,5 % doby zápůjčky) [58], [60].

5 Laboratorní úlohy

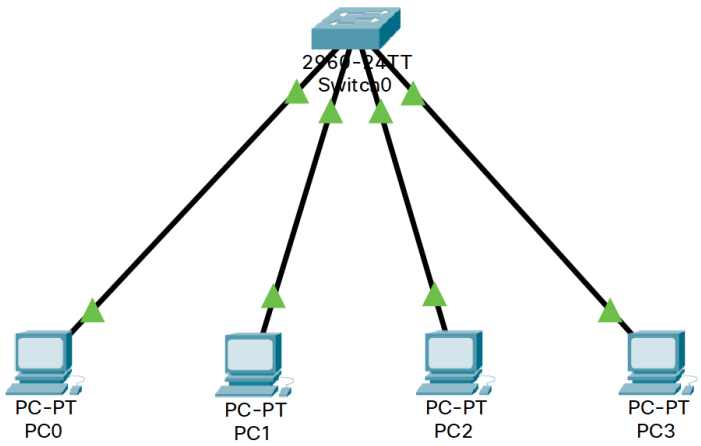
V rámci bakalářské práce budou realizovány úlohy viz níže, tato kapitola obsahuje jejich stručný popis. Kompletní návody v anglickém jazyce jsou uvedeny v příloze.

- Laboratorní úloha č. 1 – ARP protokol.
- Laboratorní úloha č. 2 – Srovnání statického a dynamického směrování.
- Laboratorní úloha č. 3 – Skupiny směrovacích protokolů Distance Vector a Link State.
- Laboratorní úloha č. 4 – TCP a UDP.
- Laboratorní úloha č. 5 – DHCP.

Verze používaného softwaru:

- Wireshark – verze 3.6.3.
- Packet Tracer – verze 8.0.1.

5.1 Laboratorní úloha č. 1 – ARP protokol



Obr. 5.1: Topologie laboratorní úlohy č. 1.

Studenti v laboratorní úloze prozkoumají protokol ARP. Práce probíhá v programu Wireshark ve spojení s lokálním PC a v Packet Traceru.

V první části se studenti seznámí s analyzátorem Wireshark a prozkoumají ARP tabulku na svém lokálním PC. Zde si prohlédnou, jaké položky tabulka obsahuje. Dále utvoří dvojice, v nichž spolupracují při plnění následujících úkolů. Pomocí příkazové řádky zjistí své adresy a následně prostřednictvím utility ping vygenerují ARP provoz, který zachytí ve Wiresharku. Studenti detailně prozkoumají obsah

přenášených rámců. Po dokončení analýzy studenti prověří, jak komunikace změnila obsah ARP tabulky. Zjistí, že přibyl jeden dynamický záznam, který následně odstraní a nahradí jej statickým záznamem. Znovu ověří komunikaci utilitou ping a zjistí, zda Wireshark zachytí nové ARP pakety. V posledním úkolu první části studenti ze zachycených paketů vykreslí grafy ve Wiresharku a porovnají výsledky s teoretickým úvodem.

V druhé části práce se studenti přesunou do prostředí Packet Traceru. Nejdříve realizují zapojení podle referenční topologie (viz Obr. 5.1) a následně provedou základní konfiguraci koncových zařízení. Práce se odehrává v simulačním módu, proto je možné zkoumat obsah jednotlivých rámců a sledovat změny MAC adres v rámcích na počátku komunikace, kdy nejsou známy cílové MAC adresy a na konci komunikace. Studenti postupně krokují simulaci a zkoumají obsah přenášených rámců. Dále prohlíží ARP tabulky počítačů a MAC tabulku přepínače.

Použité nástroje:

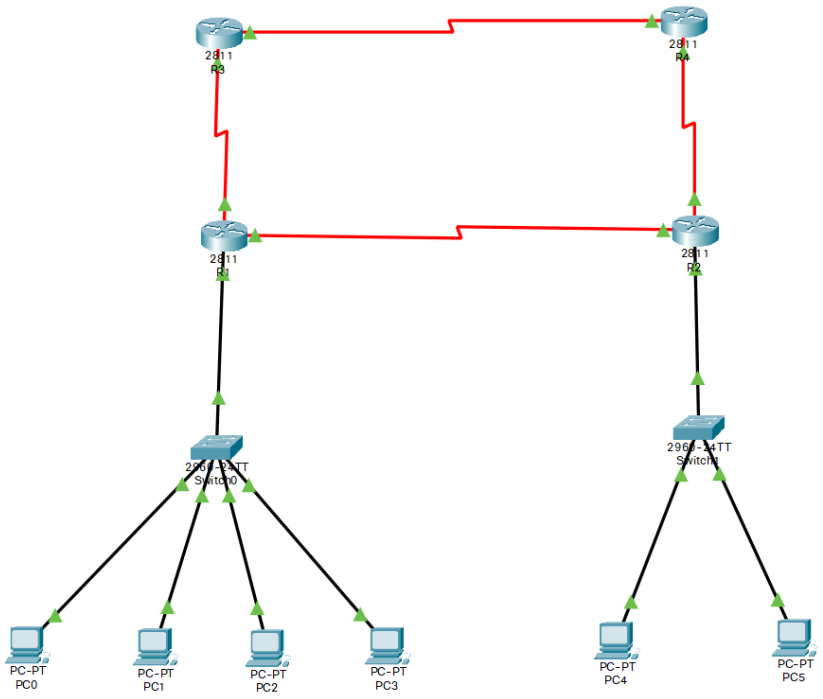
- Wireshark
  - 1 lokální PC
- Packet Tracer
  - 4 přímé kabely
  - 4 počítače
  - 1 přepínač typu Cisco 2960

5.2 Laboratorní úloha č. 2 – Srovnání statického a dynamického směrování

Druhá laboratorní úloha se zabývá statickým a dynamickým směrováním. Topologie je zobrazena na Obr. 5.2. Vysvětlí se zde základy směrování a směrovací tabulky. Úloha se dělí na dvě části.

V první části studenti nastaví IP adresy na všech zařízeních a prozkoumají počáteční stav směrovací tabulky na směrovačích. Následně nakonfigurují statické cesty pro zpřístupnění spojení mezi oběma lokálními sítěmi obsahujícími koncové uzly. Po ověření úspešné komunikace studenti přeruší spoj mezi směrovači, přes který prochází pakety (definováno během konfigurace statického směrování), a sledují, zda se síť přizpůsobí výpadku na lince vzhledem k faktu, že k cíli existuje jiná cesta.

V druhé části studenti pomocí protokolu RIPv1 nastaví dynamické směrování. Jakmile sítě vzájemně odpovídají na ICMP zprávy, studenti prozkoumají směrovací tabulky a určí, která cesta ze dvou možných byla protokolem zvolena jako nejlepší. Zvolenou cestu následně odstraní přerušením spoje a sledují, jak dynamický protokol na takovou změnu zareaguje. Studenti se dále seznámí s technikou rozložení zátěže



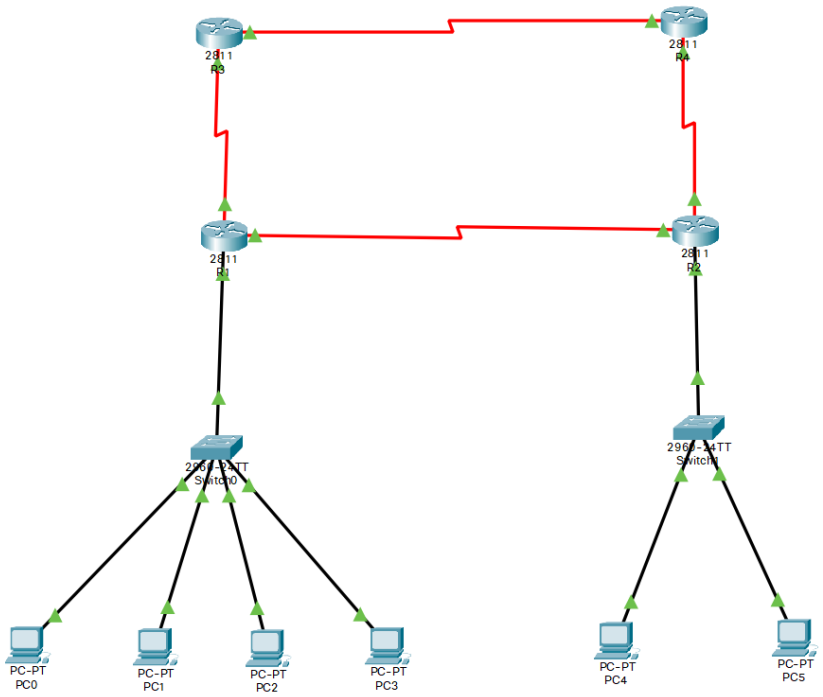
Obr. 5.2: Topologie laboratorní úlohy č. 2.

(*load balancing*).

Použité nástroje:

- Packet Tracer
  - 8 přímých kabelů
  - 6 počítačů
  - 4 sériové kabely
  - 4 směrovače typu Cisco 2811
  - 2 přepínače typu Cisco 2960

5.3 Laboratorní úloha č. 3 – Skupiny směrovacích protokolů Distance Vector a Link State



Obr. 5.3: Topologie laboratorní úlohy č. 3.

Studenti vychází ze stejné topologie jako v úloze č. 2 (viz Obr. 5.3). Účelem laboratoře je srovnání skupin dynamických protokolů *Distance Vector* (RIPv2) a *Link State* (OSPF). Dále jsou vysvětleny pojmy administrativní vzdálenost, metrika a jejich význam ve volbě směrovacích cest. Laboratorní úloha se dělí na dvě části.

V první části úlohy studenti přenastaví adresování v rámci celé sítě pro lepší pochopení odlišností protokolů RIPv1 a RIPv2. Dále upraví propustnost sériové linky mezi směrovači R1 a R2. Účelem je, aby rychlejší cesta vedla přes vyšší počet směrovačů. Po této přípravě studenti zprovozní protokol RIPv1 stejným způsobem jako v laboratorní úloze č. 2 a prozkoumají, jak se tento protokol dokáže vypořádat s podsítěmi variabilní délky. Následně nahradí protokol RIPv1 za verzi 2 a ověří, jak se tato změna projeví ve směrovacích tabulkách. Poté se sleduje pomocí utility traceroute, zda úprava propustnosti má vliv na chování protokolu.

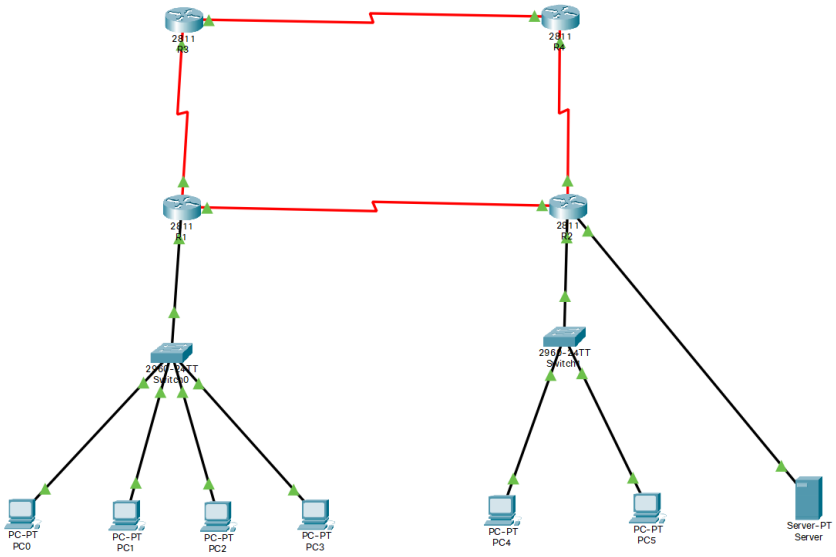
Účelem druhé části je zprovoznit protokol OSPF paralelně s RIPv2. RIPv2 se zachová z důvodu srovnání priorit směrovacích protokolů. Studenti před samotnou konfigurací spočítají kumulativní OSPF ceny pro dosažení destinací. Následně přidávají jednotlivé sítě do procesu OSPF směrování a postupně sledují, jakým způsobem

se uplatňuje priorita směrovacích protokolů při vkládání záznamů do směrovací tabulky.

Použité nástroje:

- Packet Tracer
  - 8 přímých kabelů
  - 6 počítačů
  - 4 sériové kabely
  - 4 směrovače typu Cisco 2811
  - 2 přepínače typu Cisco 2960

5.4 Laboratorní úloha č. 4 – TCP a UDP



Obr. 5.4: Topologie laboratorní úlohy č. 4.

Laboratorní úloha č. 4 ukazuje rozdíly mezi transportními protokoly TCP a UDP. Zároveň vysvětluje základy aplikačních protokolů HTTP a DNS. Kromě Packet Traceru se pracuje také v síťovém analyzátoru Wireshark. Úloha je rozdělena na část ve Wiresharku a na část v Packet Traceru.

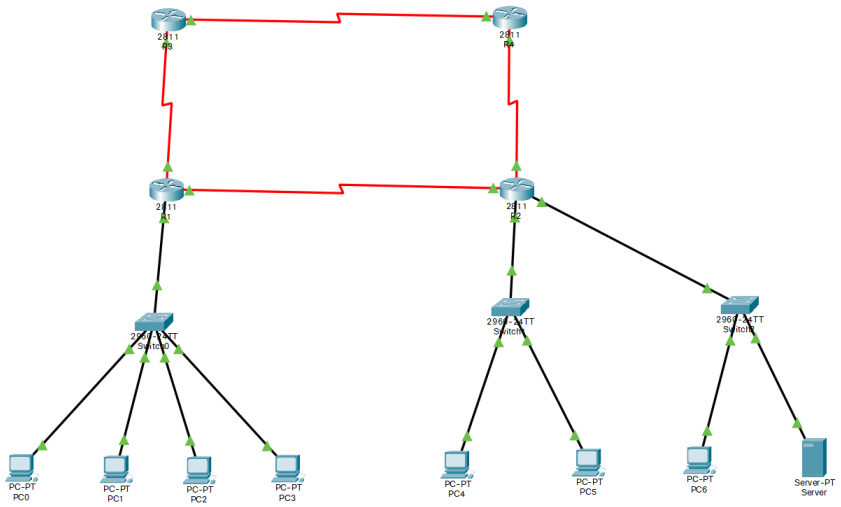
V první části studenti analyzují provoz ve Wiresharku. Zachytává se transportní protokol UDP a následně TCP při využití aplikačního protokolu DNS. K tomuto účelu se pracuje s utilitou nslookup a přístupem na stránky VUT. Princip komunikace pomocí transportního protokolu TCP se dále vysvětlí na protokolu HTTP, kde studenti zachytí připojení k nezabezpečenému webu (bez šifrování komunikace). Na závěr studenti vygenerují grafy ze zachycené komunikace a srovnají množství přenesených dat během komunikace.

Druhá část se přesouvá do prostředí Packet Traceru. Do topologie (viz Obr. 5.4) se začlení jeden server, na němž se spustí služby HTTP a DNS. Studenti nastaví základní IP konfiguraci na serveru, vytvoří jednoduchou webovou stránku (prostřednictvím HTML) a vloží do paměti statický DNS záznam, který překládá zvolené doménové jméno na IP adresu serveru. Do konfigurace koncových zařízení přidají DNS server s adresou nově nastaveného serveru. Po úspěšné konfiguraci se studenti připojí na vytvořenou webovou stránku. Přes simulační mód se sleduje celý průběh komunikace, tj. překlad doménového jména na IP adresu (UDP – DNS), následně spojení klienta se serverem, přenos HTTP paketů (TCP – HTTP) a nakonec ukončení spojení.

Použité nástroje:

- Wireshark
  - 1 lokální PC
- Packet Tracer
  - 9 přímých kabelů
  - 6 počítačů
  - 4 sériové kabely
  - 4 směrovače typu Cisco 2811
  - 2 přepínače typu Cisco 2960
  - 1 server

5.5 Laboratorní úloha č. 5 – DHCP



Obr. 5.5: Topologie laboratorní úlohy č. 5.

V laboratorní úloze číslo 5 studenti prozkoumají funkce protokolu DHCP v různých scénářích. Laboratorní úloha se dělí na dvě části.

V první části studenti spustí Wireshark a zachytí komunikaci od uvolnění adresy po kompletní transakci, během níž dochází k přidělení nové adresy z DHCP serveru. Následně podrobně prozkoumají obsah přenášených zpráv pro pochopení základního principu DHCP.

V druhé části se práce přesune do programu Packet Tracer. V topologii přibude oproti úloze č. 4 jeden přepínač a jeden PC do podsítě serveru (viz Obr. 5.5). Směrovač, ke kterému jsou klienti a server připojeni, se nastaví do role DHCP serveru. Studenti nakonfigurují dva adresní prostory, pro každou připojenou lokální síť jeden, a následně prozkoumají, z kterého prostoru se klientům přiřazují adresy. Celý proces komunikace se podrobně analyzuje. V druhé části se na směrovači vypne služba DHCP. Na zapojeném serveru se spustí služba DHCP a provedou se příslušná nastavení, aby koncová zařízení obou lokálních sítí (připojených k jednomu směrovači) obdržela adresy protokolem DHCP. Směrovač se nakonfiguruje do role DHCP přenosového agenta, který DHCP zprávy předává mezi stanicemi a DHCP serverem umístěnými v odlišných sítích. Studenti mají možnost prozkoumat, jak se změní obsah DHCP zpráv oproti stavu, kdy směrovač sloužil jako DHCP server a komunikace tak probíhala v rámci jedné sítě.

Použité nástroje:

- Wireshark
  - 1 lokální PC
- Packet Tracer
  - 11 přímých kabelů
  - 7 počítačů
  - 4 sériové kabely
  - 4 směrovače typu Cisco 2811
  - 3 přepínače typu Cisco 2960
  - 1 server

Závěr

Cílem bakalářské práce byl návrh a následná implementace 5 scénářů, které demonstrují základní principy komunikačních technologií.

Pro simulování byl zvolen nástroj Packet Tracer. Program byl vyvinut společností Cisco, která v současné době patří mezi elitu v síťovém průmyslu. Software podporuje práci omezenou pouze na Cisco zařízení, jejich rozmanitost je však značná. Lze zde najít různá zařízení od základních koncových zařízení (PC, server, IP telefon...) přes přepínače, směrovače, až po moderní IoT zařízení (senzory, detektory atd.). Silnou stránkou programu je práce ve dvou režimech. Realtime režim simuluje komunikaci v reálném čase, režim Simulation umožňuje sledovat průchod paketů jednotlivými zařízeními a analyzovat změny hodnot v položkách paketů. V laboratorních úlohách č. 1, 4 a 5 se práce odehrává také na lokálních počítačích, proto se k analýze síťového provozu využívá software Wireshark.

V práci se popisuje návrh 5 laboratorních scénářů. První laboratorní úloha se věnuje protokolu ARP. Studenti zde mají možnost prozkoumat komunikaci tohoto protokolu včetně postupného doplňování záznámů do ARP tabulek počítačů a MAC tabulky přepínače. V druhé laboratorní úloze se srovnává statické a dynamické směrování. K účelu dynamického směrování byl zvolen protokol RIPv1. Vysvětlují se zde základy nastavení směrování na zařízeních Cisco. Třetí laboratorní úloha vychází ze stejné topologie jako laboratorní úloha č. 2. Porovnávají se zde skupiny směrovacích protokolů Link State a Distance Vector. Jako zástupce skupiny Link State byl zvolen protokol OSPF, jako zástupce Distance Vector protokol RIPv2. Protokol RIPv2 byl zvolen také z důvodu možnosti srovnání obou verzí tohoto protokolu. Cílem laboratoře je také přiblížit význam administrativní vzdálenosti a metriky při výpočtech nejvhodnější cesty k cíli. Čtvrtá laboratorní úloha srovnává transportní protokoly TCP a UDP. Pro názornou ukázku funce obou protokolů se využívají aplikační protokoly HTTP (TCP) a DNS (UDP + TCP). Analyzuje se překlad doménového jména na IP adresu a následný přenos obsahu webových stránek. V páté laboratorní úloze studenti prozkoumají funkci protokolu DHCP. Na směrovači se konfiguruje služba DHCP, dále se směrovač nastaví do role přenosového agenta a služba DHCP se zprovozní na serveru. V úloze se sleduje obsah přenášených paketů během získávání IP konfigurace.

Součástí práce jsou návody jednotlivých laboratorních úloh. Dále jsou součástí elektronické přílohy také .pkt soubory, jež představují výstupy po realizaci úloh podle návodů.

Literatura

[1] COLE, Zak. Network simulation or emulation?. In: *Network World* [online]. 2017 [cit. 6. 10. 2021]. Dostupné z: <https://www.networkworld.com/article/3227076/network-simulation-or-emulation.html>.

[2] ns-3-dev. ns-3 Tutorial. In: *ns-3* [online]. 2021 [cit. 20. 10. 2021]. Dostupné z: <https://www.nsnam.org/docs/release/3.35/tutorial/ns-3-tutorial.pdf>.

[3] ns-3-dev. ns-3 Model Library. In: *ns-3* [online]. 2019 [cit. 30. 10. 2021]. Dostupné z: <https://www.nsnam.org/docs/models/singlehtml/>.

[4] Boson. NetSim 13 User Manual. In: *Boson* [online]. 2020 [cit. 30. 10. 2021]. Dostupné z: <https://www.boson.com/Files/Support/NetSim-13-User-Manual.pdf>.

[5] palo73. List of network simulators/emulators and network virtualization tools. In: *NIL - Network Information Library* [online]. 2021 [cit. 30. 10. 2021]. Dostupné z: <https://nil.uniza.sk/network-simulation-virtualization-software-list/>.

[6] YUEN, Steven. Getting Started with GNS3. In: *GNS3* [online]. 2020 [cit. 13. 10. 2021]. Dostupné z: <https://docs.gns3.com/docs/>.

[7] DZERKALS, Uldis. EVE-NG Community Cookbook. In: *EVE-NG* [online]. 2021 [cit. 27. 10. 2021]. Dostupné z: <https://www.eve-ng.net/index.php/documentation/community-cookbook/>

[8] CHANNA, Jag. Deploying EVE-NG On Google Cloud Platform: Part 3. In: *Jag Channa* [online]. 2021 [cit. 2022-04-17]. Dostupné z: <https://www.jagchanna.ca/deploying-eve-ng-on-gcp-part3/>.

[9] Cisco Systems, Inc. Cisco IOS Command Hierarchy. In: *Cisco* [online]. 2002 [cit. 29. 10. 2021]. Dostupné z: <https://www.cisco.com/E-Learning/bulk/public/tac/cim/cib/using_cisco_ios_software/02_cisco_ios_hierarchy.htm>.

[10] SHARPE, Richard, Ed WARNICKE a Ulf LAMPING. Wireshark User-s Guide. In: *Wireshark* [online]. 2020 [cit. 28. 11. 2021]. Dostupné z: <https://www.wireshark.org/download/docs/Wireshark%20User's%20Guide.pdf>.

[11] DYE, Mark A., Rick MCDONALD a Antoon W. RUFI. *Network Fundamentals, CCNA Exploration Companion Guide*. Indianopolis: Cisco Press, 2008. ISBN 978-1-58713-208-7.

[12] LearnCisco. Data Encapsulation. In: *LearnCisco* [online]. 2015 [cit. 14. 11. 2021]. Dostupné z: <https://www.learncisco.net/courses/ccna/part-1-internetworking/data-encapsulation.html>.

[13] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 4. vydání. Brno: CP Books, 2005. ISBN 80-722-6675-6.

[14] TOUCH, Joe, Eliot LEAR a další. Service Name and Transport Protocol Port Number Registry. In: *Internet Assigned Numbers Authority* [online]. 2021 [cit. 17. 11. 2021]. Dostupné z: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.

[15] BOUŠKA, Petr. TCP/IP - Routing - směrování. In: *SAMURAJ-cz* [online]. 2007 [cit. 11. 02. 2022]. Dostupné z: <https://www.samuraj-cz.com/clanek/tcpip-routing-smerovani/>.

[16] GLENN, Walter. How to Add a Static TCP/IP Route to the Windows Routing Table. In: *How-To Geek* [online]. 2017 [cit. 09. 02. 2022]. Dostupné z: <https://www.howtogeek.com/howto/windows/adding-a-tcpip-route-to-the-windows-routing-table/>.

[17] GRAZIANI, Rick a Allan JOHNSON. *Routing Protocols and Concepts, CCNA Exploration Companion Guide*. Indianapolis: Cisco Press, 2007. ISBN 978-1-58713-206-3.

[18] CHALLEN, Geoffrey. Autonomous System (AS). In: *Network Encyclopedia* [online]. 2016 [cit. 16. 02. 2022]. Dostupné z: <https://networkencyclopedia.com/autonomous-system-as/>

[19] Cisco Systems, Inc. What are Autonomous System Numbers (ASN)?. In: *ThousandEyes* [online]. 2022 [cit. 16. 02. 2022]. Dostupné z: <https://www.thousandeyes.com/learning/glossary/as-autonomous-system>.

[20] SHELDON, Robert. Autonomous system (AS). In: *TechTarget* [online]. 2019 [cit. 16. 02. 2022]. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/autonomous-system>.

[21] HUNT, Craig. *TCP/IP Network Administration*. 3rd edition. Sebastopol: O'Reilly & Associates, 2002. ISBN 0-596-00297-1.

[22] ARIN. Autonomous System Numbers. In: *American Registry for Internet Numbers* [online]. 2022 [cit. 16. 02. 2022]. Dostupné z: <https://www.arin.net/resources/guide/asn/>.

[23] WHITE, Russ, Danny MCPHERSON a Srihari SANGLI. *Practical BGP*. Carmel (Indiana): Pearson Education (US), 2004. ISBN 0321127005.

[24] PLUMMER, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826. In: *RFC Editor* [online]. 1982 [cit. 28. 11. 2021]. Dostupné z: <https://www.rfc-editor.org/info/rfc826>.

[25] IPCisco. Address Resolution Protocol (ARP). In: *IPCisco* [online]. 2021 [cit. 17. 11. 2021]. Dostupné z: <https://ipcisco.com/lesson/address-resolution-protocol-arp/>.

[26] OmniSecu. Address Resolution Protocol Tutorial, How ARP work, ARP Message Format. In: *OmniSecu* [online]. 2021 [cit. 19. 11. 2021]. Dostupné z: <https://www.omnisecu.com/tcpip/address-resolution-protocol-arp.php>.

[27] Practical Networking. Traditional ARP. In: *Practical Networking* [online]. 2021 [cit. 19. 11. 2021]. Dostupné z: <https://www.practicalnetworking.net/series/arp/traditional-arp/>.

[28] BALCHUNAS, Aaron. Routing Information Protocol. In: *Router Alley* [online]. 2012 [cit. 23. 02. 2022]. Dostupné z: <http://www.routeralley.com/guides/rip.pdf>

[29] SHARMA, Saurabh. Routing Information Protocol (RIP). In: *GeeksForGeeks* [online]. 2021 [cit. 23. 02. 2022]. Dostupné z: <https://www.geeksforgeeks.org/routing-information-protocol-rip/>.

[30] HEDRICK, C., "Routing Information Protocol", RFC 1058, DOI 10.17487/RFC1058. In: *RFC Editor* [online]. 1988 [cit. 25. 02. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc1058>.

[31] DONAHUE, Gary A. *Kompletní průvodce síťového experta*. Brno: Computer Press, 2009. ISBN 978-80-251-2247-1.

[32] MALKIN, G., "RIP Version 2 - Carrying Additional Information", RFC 1723, DOI 10.17487/RFC1723. In: *RFC Editor* [online]. 1994 [cit. 27. 02. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc1723>.

[33] Juniper Networks, Inc. RIP Authentication. In: *Juniper Networks* [online]. 2021 [cit. 02. 03. 2022]. Dostupné z: <https://www.juniper.net/documentation/us/en/software/junos/rip/topics/topic-map/rip-authentication.html>.

[34] SEGEČ, Pavel. Configuration of the RIPv2 authentication. In: *NIL - Network Information Library* [online]. 2009 [cit. 02. 03. 2022]]. Dostupné z: <https://nil.uniza.sk/configuration-ripv2-authentication/>.

[35] MALKIN, G. a R. MINNEAR, "RIPng for IPv6", RFC 2080, DOI 10.17487/RFC2080. In: *RFC Editor* [online]. 1997 [cit. 02. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc2080>.

[36] Juniper Networks, Inc. RIP and RIPng Overview. In: *Juniper Networks* [online]. 2021 [cit. 02. 03. 2022]. Dostupné z: <https://www.juniper.net/documentation/us/en/software/junos/rip/topics/topic-map/rip-and-ripng-overview.html>.

[37] MOY, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328. In: *RFC Editor* [online]. 1998 [cit. 09. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc2328>.

[38] MOY, J., "OSPF Version 2", RFC 1247, DOI 10.17487/RFC1247. In: *RFC Editor* [online]. 1991 [cit. 09. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc1247>.

[39] COLTUN, R., FERGUSON, D., a J. MOY, "OSPF for IPv6", RFC 2740, DOI 10.17487/RFC2740. In: *RFC Editor* [online]. 1999 [cit. 09. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc2740>.

[40] BALCHUNAS, Aaron. Open Shortest Path First. In: *Router Alley* [online]. 2007 [cit. 09. 03. 2022]. Dostupné z: <https://www.routeralley.com/guides/ospf.pdf>

[41] Packet Coders. OSPF Area-s Explained. In: *Packet Coders* [online]. 2019 [cit. 09. 03. 2022]. Dostupné z: <https://www.packetcoders.io/ospf-areas-explained/>.

[42] MOY, J., "OSPF Version 2", RFC 1583, DOI 10.17487/RFC1583. In: *RFC Editor* [online]. 1994 [cit. 09. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc1583>.

[43] MOGUL, J. a S. DEERING, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191. In: *RFC Editor* [online]. 1990 [cit. 11. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc1191>.

[44] NetworkLessons. OSPF LSA Types Explained. In: *NetworkLessons* [online]. 2022 [cit. 11. 03. 2022]. Dostupné z: <https://networklessons.com/ospf/ospf-lsa-types-explained>.

[45] GENG, Irene. 6 Types of OSPF LSA. In: *Router-Switch* [online]. 2018 [cit. 11. 03. 2022]. Dostupné z: <https://www.router-switch.com/faq/6-types-of-ospf-lsa.html>.

[46] Cisco Systems, Inc. OSPF Neighbor States. In: *Cisco* [online]. 2014 [cit. 12. 03. 2022]. Dostupné z: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13685-13.html>

[47] O, Samuel. OSPF Neighbor Adjacency. In: *Expert Network Consultant* [online]. 2018 [cit. 12. 03. 2022]. Dostupné z: <https://www.expertnetworkconsultant.com/configuring/ospf-neighbor-adjacency/>.

[48] ABIY, Thaddeus, Hannah PANG, Christopher WILLIAMS a další. Dijkstra's Shortest Path Algorithm. In: *Brilliant* [online]. 2022 [cit. 16. 03. 2022]. Dostupné z: <https://brilliant.org/wiki/dijkstras-short-path-finder/>.

[49] POSTEL, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768. In: *RFC Editor* [online]. 1980 [cit. 30. 03. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc768>.

[50] KOZIEROK, Charles M. TCP/IP User Datagram Protocol (UDP). In: *The TCP/IP Guide* [online]. 2005 [cit. 30. 03. 2022]. Dostupné z: <http://www.tcpipguide.com/free/t_TCPIPUserDatagramProtocolUDP.htm>.

[51] POSTEL, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793. In: *RFC Editor* [online]. 1981 [cit. 01. 04. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc793>.

[52] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

[53] RAMAKRISHNAN, K., FLOYD, S., a D. BLACK, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168. In: *RFC Editor* [online]. 2001 [cit. 02. 04. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc3168>.

[54] Juniper Networks, Inc. Understanding CoS Explicit Congestion Notification. In: *Juniper Networks* [online]. 2021 [cit. 02. 04. 2022]. Dostupné z: <https://www.juniper.net/documentation/us/en/software/junos/cos/topics/concept/cos-qfx-series-explicit-congestion-notification-understanding.html>.

[55] KOZIEROK, Charles M. TCP/IP Transmission Control Protocol (TCP). In: *The TCP/IP Guide* [online]. 2005 [cit. 03. 04. 2022]. Dostupné z: <http://www.tcpipguide.com/free/t_TCPIPTransmissionControlProtocolTCP.htm>.

[56] DROMS, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131. In: *RFC Editor* [online]. 1997 [cit. 28. 04. 2022]. Dostupné z: <https://www.rfc-editor.org/info/rfc2131>.

[57] NetworkLessons. BOOTP (Bootstrap Protocol). In: *NetworkLessons* [online]. 2022 [cit. 28. 04. 2022]. Dostupné z: <https://networklessons.com/cisco/ccie-routing-switching-written/bootp-bootstrap-protocol>.

[58] DROMS, Ralph a Ted LEMON. *DHCP Příručka administrátora: [kompletní návod pro konfiguraci a správu DHCP služeb, klientů a serveru]*. Brno: Computer Press, 2004. ISBN 80-251-0130-4.

[59] Huawei. DHCP Messages. In: *Huawei* [online]. 2019 [cit. 29. 04. 2022]. Dostupné z: <https://support.huawei.com/enterprise/en/doc/EDOC1100058931/25cd2dfc/dhcp-messages#section_dc_vrp_dhcp_feature_000702>.

[60] MENS, Rudy. DHCP Lease Time — What is it and How does it work?. In: *LazyAdmin* [online]. 2019 [cit. 05. 05. 2022]. Dostupné z: <https://lazyadmin.nl/home-network/dhcp-lease-time/>.

[61] DROMS, Ralph a Ted LEMON. *The DHCP Handbook*. 2nd Edition. Indianaopolis, Indiana: Sams Publishing, 2002. ISBN 978-0672323270.

Seznam symbolů a zkratek

ABR	Area Border Router
ARP	Address Resolution Protocol
AS	Autonomous System
ASBR	Autonomous System Border Router
ASN	Autonomous System Number
ATM	Asynchronous Transfer Mode
BDR	Backup Designated Router
BGP	Border Gateway Protocol
BOOTP	Bootstrap Protocol
CCNA	Cisco Certified Network Associate
CCNP	Cisco Certified Network Professional
CIDR	Classless Inter-Domain Routing
CLI	Command Line Interface – příkazový řádek
DBD	Database Description
DCE/DTE	Data Communications Equipment / Data Terminal Equipment
DES	Discrete-event simulation – diskrétní simulace
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DR	Designated Router
EGP	Exterior Gateway Protocol
ENARSI	Implementing Cisco Enterprise Advanced Routing and Services
ENCOR	Implementing and Operating Cisco Enterprise Network Core Technologies
EVE-NG	Emulated Virtual Environment – Next Generation

GNS3	Graphical Network Simulator-3
GUI	Graphic User Interface – grafické uživatelské rozhraní
IANA	Internet Assigned Numbers Authority
IoT	Internet of Things – internet věcí
IP	Internet Protocol
ISN	Initial Sequence Number
ISO	International Organization for Standardization – Mezinárodní organizace pro normalizaci
ISP	Internet Service Provider
LAN	Local Area Network – lokální síť
LSA	Link State Advertisement
LSAck	Link State Acknowledgment
LSDB	Link State Database
LSP	Link State Packet
LSR	Link State Request
LSU	Link State Update
MAC	Media Access Control
MD5	Message-Digest algorithm
MSL	Maximum Segment Lifetime
MSS	Maximum Segment Size – maximální délka segmentu
MTU	Maximum Transmission Unit
NDP	Neighbor Discovery Protocol
ns-3	Network Simulator 3
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First

OUI	Organizationally Unique Identifier – jedinečný identifikátor výrobce
PDU	Protocol Data Unit
RARP	Reverse Address Resolution Protocol
RIP	Routing Information Protocol
RM	Reference Model – referenční model
SPF	Shortest Path First
SPT	Shortest Path Tree
TCB	Transmission Control Block
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VIRL	Virtual Internet Routing Lab
VLSM	Variable-Length Subnet Mask
VM	Virtual Machine – virtuální stroj
VPC	Virtual PC – virtuální počítač
WAN	Wide Area Network

Seznam příloh

A Lab 1 – ARP protocol

In this laboratory you should become familiar with the ARP protocol and in the end fully understand it.

Objectives

1. Examine the ARP table on your local computer.
2. Make a pair with your colleague and try to capture and analyze the ARP communication between your computers in Wireshark.
3. Delete the record obtained in objective 2. Add the addresses (IP and MAC) of your colleague to the ARP table using the static method and then examine the communication in Wireshark.
4. Display the graph of captured packets in Wireshark.
5. Create the reference topology in Packet Tracer (see Fig. A.1).
6. Generate and examine the ARP communication in Packet Tracer Simulation mode. Explore the switch's MAC table.

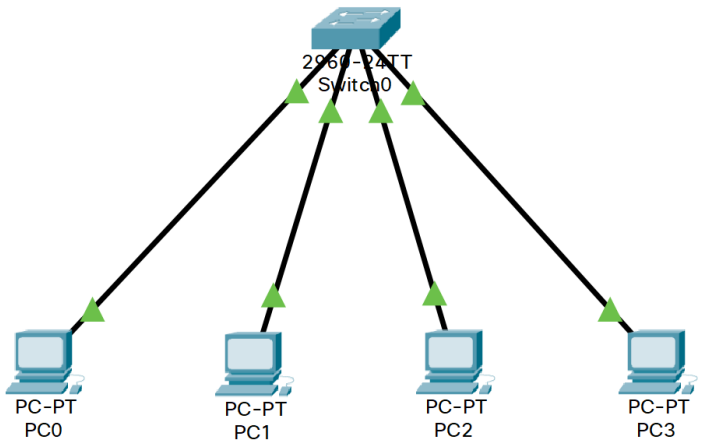


Fig. A.1: Lab 1 reference topology.

A.1 Introduction

ARP (Address Resolution Protocol), defined in RFC 826 [1], is used in computer networks to map the logical IPv4 address (32-bit address used for addressing on the ISO/OSI network layer) to the MAC address (48-bit physical address used for addressing on the ISO/OSI link layer). It operates between the OSI network and link layers. When the communication starts, an IP packet is created by adding the IP header, which consists of source and destination IP address (and many other items), to the data. But when the device wants to communicate, it must know the MAC address (on the Ethernet networks) of the destination device as well. There are 2 scenarios:

1. The device wants to communicate with another device on the local network.
2. The device wants to communicate with a device outside the local network.

In both cases, the device initiating the communication needs to know the MAC address of the destination. In the first scenario, the desired MAC address belongs to the **destination device** on the local network. In the second scenario, the destination MAC address must be set to the physical address of the **default gateway** (commonly router).

ARP packets are divided into 2 groups: *ARP request* and *ARP response*. ARP request is used by the device initiating the communication to resolve the IPv4 address of the destination device to its physical (MAC) address. You can imagine the request as: "*Hey, I am host A and I want to communicate with Host B with this IP address. What is your MAC address?*"ARP response is sent as the response to the ARP request, where the device (which the request was sent to) sends its physical address. You can imagine the response as: "*Hey Host A, I am Host B, I recognized my IP address and here is my MAC address.*"ARP packets are encapsulated in the Ethernet frames (see Fig. A.2).

The records (IP-MAC bindings) are stored in the **ARP table** (cache memory). The table basically consist of:

- **Internet Address** – The logical (IP) address of the destination device.
- **Physical Address** – The MAC address of the destination device.
- **Type** – The way the record was inserted into the table. There are 2 ways: *dynamic* and *static*. Dynamic record is a record learnt by the ARP process (ARP request and reply). Static record is a record manually inserted by the user.

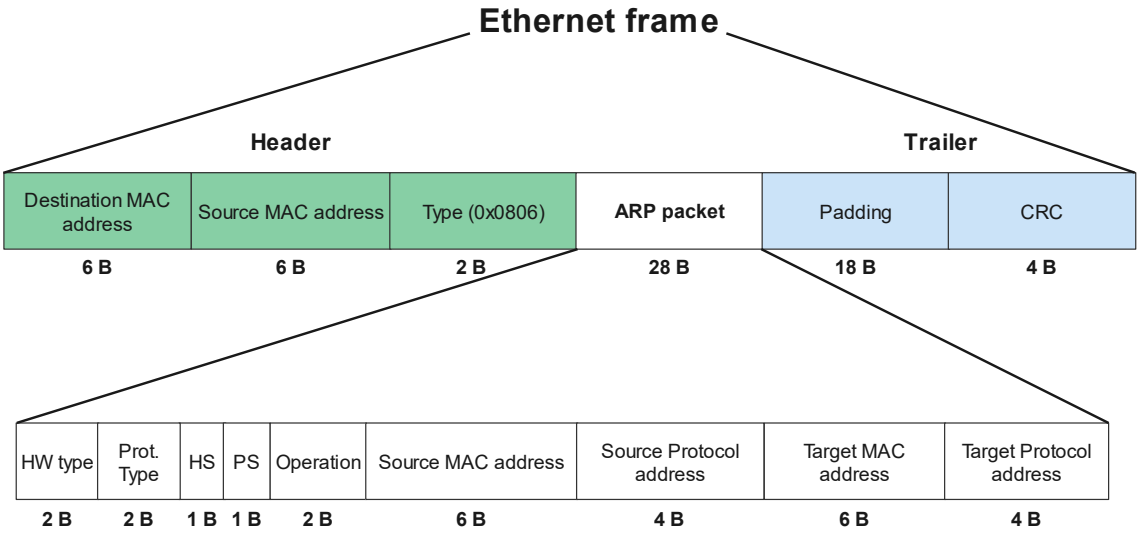


Fig. A.2: ARP packet encapsulated inside the Ethernet frame.

Principle of communication

1. The source device (Host A) wants to communicate with the destination device (Host B). Host A searches its ARP table to find the IP address of the Host B that is bound with its MAC (Media Access Control) address. If the record is found, Host A sends a message directly to the Host B.
2. If the record is not present, the ARP request is generated. Host A sets its own MAC address as the Source MAC address in both the Ethernet frame and the ARP packet (which is encapsulated inside the Ethernet frame) and sets its own IP address as the Source Protocol address. Ethernet Destination MAC address is set to Broadcast, Target MAC address is left blank (default value). Target Protocol Address is set to the Host B's IP address.
3. As the Destination MAC address is set to broadcast, all the devices on the local network receive and process the frame (see Fig. A.3). Each device de-encapsulates the ARP packet and compares the Target Protocol Address with its own IP (Internet Protocol) address. Only Host B finds a match. Every other device will drop the frame.
4. Host B adds a record to its own ARP table with the addresses of the Host A (if it is not already present) and generates ARP response. It sets original source addresses as the destination addresses and sets its own addresses (both logical and physical) as the source addresses.
5. ARP response is now sent as unicast directly to the Host A (see Fig. A.4).
6. Host A receives the ARP response, fills its own ARP table with the received addresses (Source MAC and Source Protocol) and sends the original message directly to the Host B.

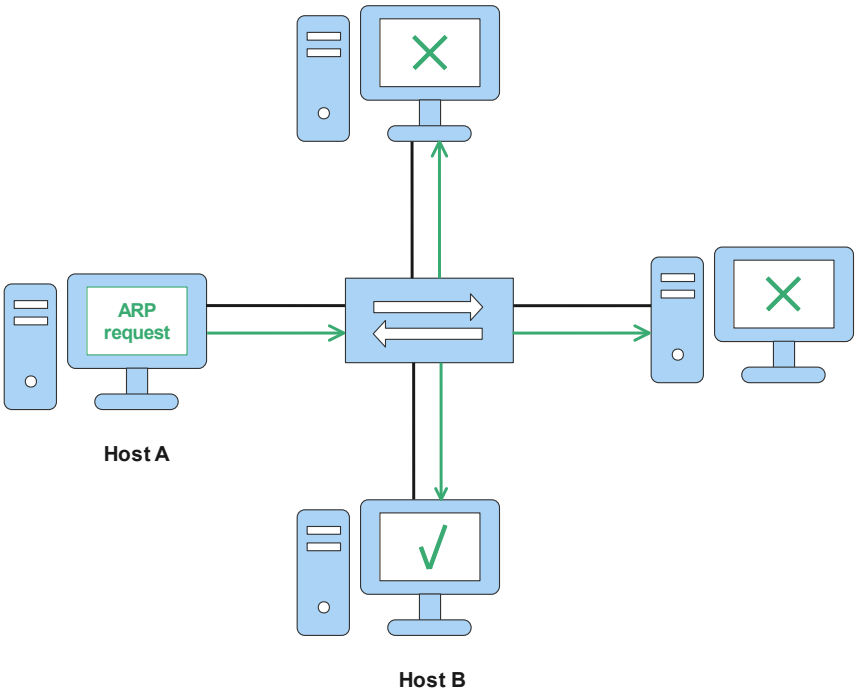


Fig. A.3: ARP request.

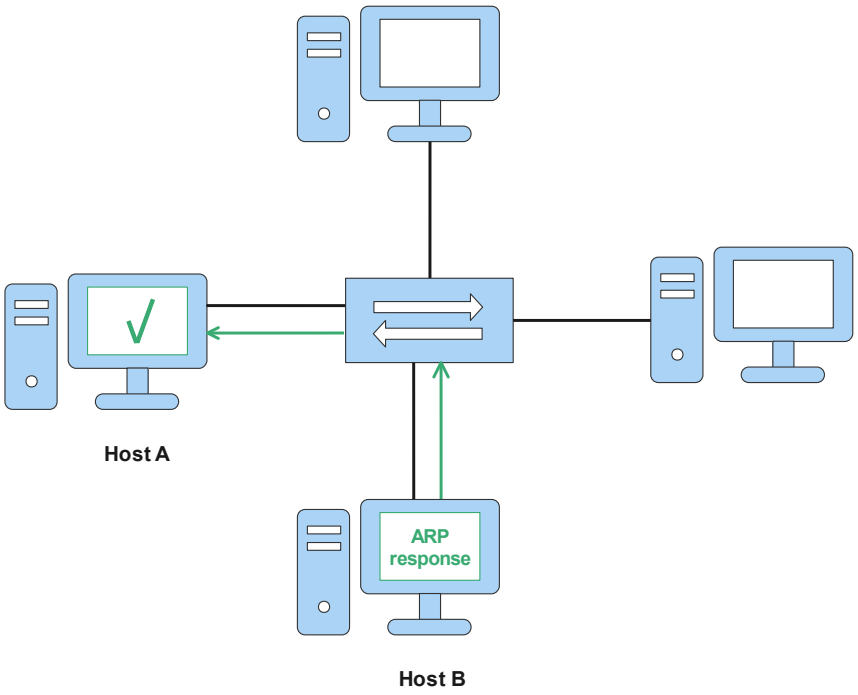


Fig. A.4: ARP response.

A.2 Wireshark

Wireshark is an open source network packet analyzer where you can capture and analyze live traffic. At the beginning, you choose the interface where you want to capture the traffic and then all the captured packets are visible. The main window is divided to the following sections (see Fig. A.5):

1. The **Menu** and **Main toolbar** are used to do some specific actions based on the chosen tool. Commonly used items are the *Start capturing packets* (symbol of blue fin under the File) and *Stop capturing packets* (the symbol of square beside the fin).
2. The **Filter Toolbar** allows users to set filter on the specific protocol, source/destination address, port etc. and display only the desired packets.
3. The **Packet List Pane** displays all the captured packets. It is further divided to the following sections by default:
   - *No.* – The number of a packet in order it was captured.
   - *Time* – The timestamp of the packet.
   - *Source* – The source IP address.
   - *Destination* – The destination IP address.
   - *Protocol* – The protocol name abbreviation.
   - *Length* – The length of a packet.
   - *Info* – Information about the packet content.
4. The **Packet Details Pane** displays the details about selected packet (chosen in the **Packet list pane**) in rows, ie. link layer protocol, IP protocol etc. You can also click on each row to display further details about each protocol (like information inside the header).
5. The **Packet Bytes Pane** displays the data inside the selected packet from the **Packet List pane** and highlights the bytes corresponding to the items selected in the **Packet Details Pane** [2].

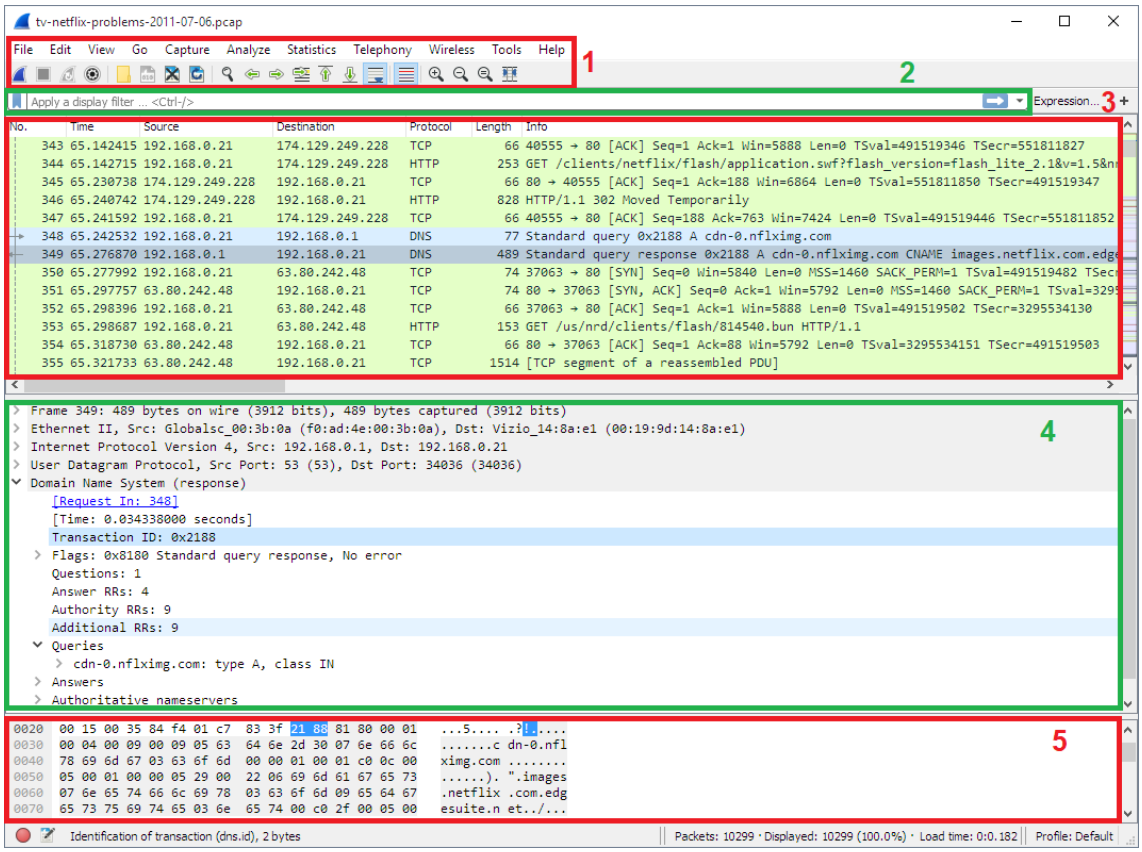


Fig. A.5: Wireshark Main window.

A.2.1 Objective 1

1. Open the **CMD** (Command Prompt) on your local computer.
2. Execute the `arp -a` command, which is used to display the contents of the ARP table.
3. You will see the records divided to the sections by interfaces, which have the IP address assigned. Explore what these interfaces are by issuing the `ipconfig` command. Here you get a list of interfaces with their name and IP address assigned.
4. As mentioned in the introduction, you can see that in the ARP table there are 3 columns: *Internet Address*, *Physical Address* and *Type*. Now focus on the Ethernet interface (the name should be something like *Ethernet adapter Ethernet* in the output of the `ipconfig` command) which is identified by its IP address. Among some static records, there should be at least one dynamic record – **the default gateway** (check the `ipconfig` output to verify the address). It usually uses the first host address of the subnet range. Its presence means that your computer has already communicated with the router (e.g., to obtain the dynamic IP address).

5. You can notice that there is one special static record with the *ff-ff-ff-ff-ff-ff* MAC address.
   What is this record used for?

A.2.2 Objective 2

1. For this objective, you should team up with your colleague next to you. Once you're in a team, you can continue to the next step.
2. If you have closed CMD from the previous objective, open it again and execute command `ipconfig /all`. This command gives you detailed information about the network configuration of your local interfaces.
3. Find the Ethernet interface (same as in the objective 1) and check the obtained data. You should be especially interested in the *Physical Address* and the *IPv4 Address*. Write the addresses somewhere for the future use.
4. Now get the addresses of your colleague and check your ARP table, if the record of your neighbor is not present in the table[1].
5. Let's open Wireshark. At first, the list of available interfaces, where you can capture the traffic, is displayed. Choose your Ethernet interface.
6. Once you have the interface selected, you can probably see dozens of packets being captured. Now you will apply the filter for the ARP protocol. In the **Filter Toolbar** (see Fig. A.5), type `arp` and press enter. If the background color of the Filter Toolbar changes to green, the filter is correctly applied.
7. Keep the Wireshark running and return to the CMD. Now **only one member** of the team will generate the ARP request to obtain the MAC address of the second member, but the traffic will be caught on both computers. This is for the ability to communicate over Ethernet. For this purpose, you will use the **ping**[2] utility.
8. In the CMD, execute `ping <IP>` where **<IP>** is the IP address of your colleague. Now return to the Wireshark. You should both see two ARP messages similar to the Fig. A.6. As mentioned in the introduction, ARP requests are sent to all the hosts on the local network, so you will see also requests from other colleagues. For this purpose, you can edit the filter to display only the ARP communication of your pair. Use the following command:
   `arp.src.proto_ipv4 == <IP> or arp.dst.proto_ipv4 == <IP>` where

[1]NOTE: If the record is present, you should delete it. For this purpose, close the CMD and open it with admin privileges. Once opened, use the command `arp -d <IP>` where **<IP>** is the IPv4 address of your colleague.

[2]Ping is a software utility used to test if the host is reachable over the IP network. It sends out *ICMP Echo request* message and awaits *ICMP Echo reply* message. For more information visit the [3].

<IP> represents your IPv4 address.

Fig. A.6: ARP request and reply captured in Wireshark.

9. In the Fig. A.6, you can see the ARP request is sent as a broadcast and info contains: "*Who has 192.168.0.4? Tell 192.168.0.1*". A host needs to be identified by its IP address first. If a match is found, the response is sent. The ARP response is sent as a unicast (directly to the source of ARP request) and info contains: "*192.168.0.4 is at 00:50:79:66:68:03*". As you can see, some host found a match with its own IP address and sent a response where he mentions his MAC address. The length of both packets is 64 bytes (which corresponds to the Fig. A.2). Now let's examine both packets more in depth.
10. **ARP request**
    Click on the ARP request packet (see Fig. A.7). There will be 3 lines displayed in the **Packet Details Pane**. We will be interested in the last 2 lines, i.e. Ethernet II and Address Resolution Protocol (see Fig. A.2). Now expand the Ethernet II line. There are following items we are interested in:
    - **Destination Address** – The destination MAC address identifying the destination of a frame.
    - **Source Address** – The source MAC address identifying the source host.
    - **Type** – The value expressing what protocol is encapsulated inside the frame.

    Determine your values of Destination, Source and Type. Next, expand the Address Resolution Protocol line. The following items are important:
    - **Opcode** – The value identifying the type of ARP message: *request* or *response*.
    - **Sender MAC address** – The MAC address of a host sending a request.
    - **Sender IP address** – The IP address of a host sending a request.
    - **Target MAC address** – The MAC address of a host the request is destined for.
      What is the value of this item?[3]
    - **Target IP address** – The IP address of a host the request is destined for.

    Compare the values with your own addresses and addresses of your colleague. *What is the destination MAC address and who will receive the frame?*

[3] You can probably see a different value than the value displayed in the Fig. A.7. Your field should contain only the zeroes. This is the default value when the destination address is not known. For more information you can visit the [4].

Why are the Destination MAC address and Target MAC address different?

```
Frame 41: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Destination: Broadcast (ff:ff:ff:ff:ff:ff)
› Source: Private_66:68:00 (00:50:79:66:68:00)
  Type: ARP (0x0806)
  Padding: 000000000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Sender IP address: 192.168.0.1
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
  Target IP address: 192.168.0.4
```

Fig. A.7: ARP request detailed in Wireshark.

11. **ARP response**

    Now move from the ARP request to the ARP response packet (see Fig. A.8). The lines remain the same. Expand both Ethernet II and Address Resolution Protocol lines and compare your own values with the values from a request.

```
Frame 42: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: Private_66:68:03 (00:50:79:66:68:03), Dst: Private_66:68:00 (00:50:79:66:68:00)
› Destination: Private_66:68:00 (00:50:79:66:68:00)
› Source: Private_66:68:03 (00:50:79:66:68:03)
  Type: ARP (0x0806)
  Padding: 000000000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Private_66:68:03 (00:50:79:66:68:03)
  Sender IP address: 192.168.0.4
  Target MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Target IP address: 192.168.0.1
```

Fig. A.8: ARP response detailed in Wireshark.

A.2.3 Objective 3

1. Open CMD and display the contents of the ARP table.
   Is the expected record present in the table?
   What is the type of a record?
2. Now you will both remove the obtained record. For this purpose you must run CMD with administrator privileges[4]. **Write somewhere both logical and physical addresses before deletion for the future use!** Execute command `arp -d <IP>` where **<IP>** represents the IPv4 address of your colleague[5]. In the end, both of you should have removed the records obtained from objective 2.
3. Now you are going to add the record manually. Use the command `arp -s <IP> <MAC>`. Both IP (**<IP>**) and MAC (**<MAC>**) addresses are the adresses of your colleague. For the MAC address use dash as the delimeter. Once issued, display the contents of the ARP table.
   What is the type of the newly added record?
4. Go back to the Wireshark. Check if you are still capturing packets (fin is grey, square shines red) and the filter from objective 2 is still active. If not, apply it again.
5. Use `ping` command in the CMD to test the availability of your colleague.
   Were any new ARP packets captured? Why?
   Static records are pretty rarely used but can be useful when communicating with a device, whose address does not change throughout long period communication. Dynamic records have their timeout, mostly 2 minutes. If the record is not used for communication during this period, it is automatically discarded. When used, timeout is increased by another 2 minutes. Maximum time the record can be valid is 10 minutes, then it is discarded. This is different for static records. There is no timeout for these, they remain valid until manually deleted or system reset (ARP table is stored in cache memory which is cleared with reset).

[4]NOTE: Write `cmd` to the Windows search bar, then right click on it and press "Run as administrator".

[5]NOTE: To clear the whole ARP table you would execute `arp -d *`.

A.2.4 Objective 4

1. In a Wireshark, stop capturing packets (by clicking on the red square).
2. Wireshark gives you the possibility to display captured packets in graph. You can achieve this in **Statistics > I/O Graphs**.
3. New window appears. You can probably see more than 1 graph. If not, there's nothing wrong. One of the displayed graphs should display *All Packets*, which represents all the captured packets on your interface without regard to protocol types. Below the displayed graphs, there is a section with settings for each graph. You will be interested in the following values:
   - **Enabled** – If checked, the graph is displayed.
   - **Graph Name** – The name of a graph.
   - **Display Filter** – You can limit the graph only to the certain packets based on protocol, IP address etc.
   - **Y Axis** – As X axis represents the time, you can choose what the Y axis displays. You can select *Packets*, *Bytes*, *Bits* matching the filter per time interval and others.

   Below the settings section, there are some additional items. Some of them are:
   - [+] – Add a new graph.
   - [−] – Remove a graph.
   - **Interval** – The interval period for the graph.
4. Now remove all the graphs by selecting them and clicking on the [–].
5. Add a new graph by clicking on the [+]. Tick **Enabled**, change the **Graph Name** to "arp"and set the **Display Filter** to `arp`. You are free to change the **Color** to the color you want. Now select the **Bytes** value in the **Y Axis** column. Set **Interval** to **1 sec**[6].
6. In the Fig. A.10, you can see the output with the previous settings[7]. The input data are displayed in the Fig. A.9. Bytes are exported to the Tab. A.1 and packets to the Tab. A.2. The first deflection represents PCs checking for the duplicate address when statically assigned (this is called *gratuitous ARP*, you don't have to see these necessarily as you have probably dynamic address assigned from the DHCP server). There should be only one message for the correct assignment and that is ARP request. No response should be received. The last 3 peaks represent ARP request and reply communication each. You should see similar peak. If you zoom in (using the mouse wheel), you can see that peak has 128 bytes.

[6]NOTE: If your graph is too much elongated, use the **Reset** button to make recalculation.

[7]For better readability, the following graphs are generated using Matlab. Most of the aspects of the Wireshark's graphs are preserved, including axis names, scale and the function values. Functions are shifted in time so they start at zero time.

Why is it 128 bytes?

7. Now change the **Y Axis** to **Packets** (see Fig. A.11).
   How many packets are transferred during 1 peak?

No.	Time	Source	Destination	Protocol	Length	Info
5	35.819496	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
6	36.832330	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
7	37.358683	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
8	37.843343	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
9	38.359146	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
10	38.858842	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
11	39.358935	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
12	39.873878	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
13	40.405112	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
14	40.888502	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
15	41.416994	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
16	42.426050	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
17	49.137627	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.2? Tell 192.168.0.1
18	49.137627	Private_66:68:01	Private_66:68:00	ARP	64	192.168.0.2 is at 00:50:79:66:68:01
29	55.993312	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.3? Tell 192.168.0.1
30	55.993312	Private_66:68:02	Private_66:68:00	ARP	64	192.168.0.3 is at 00:50:79:66:68:02
41	63.153478	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.4? Tell 192.168.0.1
42	63.153478	Private_66:68:03	Private_66:68:00	ARP	64	192.168.0.4 is at 00:50:79:66:68:03

Fig. A.9: Input data for the I/O graph.

Tab. A.1: Table of bytes sent during the ARP communication.

Seconds	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bytes	64	128	128	128	128	128	64	0	0	0	0	0	0	128
Seconds	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Bytes	0	0	0	0	0	0	128	0	0	0	0	0	0	128

Tab. A.2: Table of packets sent during the ARP communication.

Seconds	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Packets	1	2	2	2	2	2	1	0	0	0	0	0	0	2
Seconds	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Packets	0	0	0	0	0	0	2	0	0	0	0	0	0	2

A.3 Packet Tracer

Packet Tracer is a proprietary software developed by the Cisco company used for simulating the networks. Cisco is one of the most well known companies in the network engineering. The software is primarily intended for the CCNA (Cisco Certified Network Associate) academy, but upon registration everyone can get it for free with

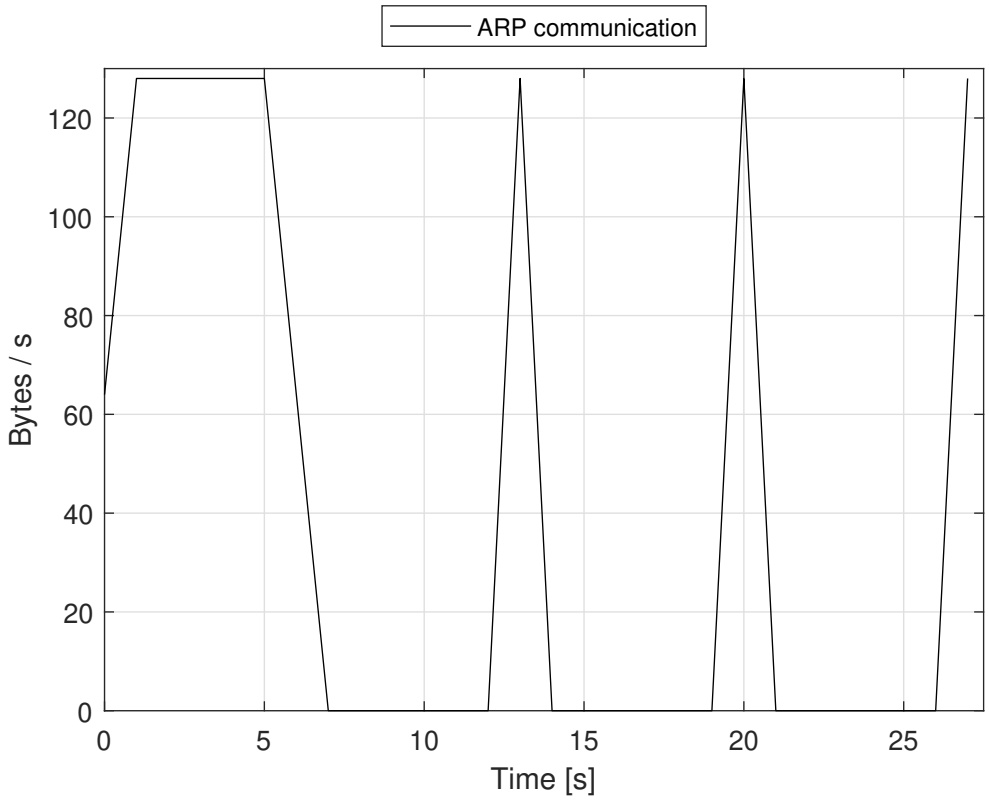


Fig. A.10: Bytes sent during ARP communication.

the basic course named *Getting Started with Cisco Packet Tracer*. You are able to work here with the **Cisco** devices only.

Device configuration

The devices in Packet Tracer can be configured via the **CLI**, **Config** and **Desktop**.

- **CLI** - CLI (Command Line Interface) is the most common way of configuring intermediary devices including switches, routers etc. You must have some knowledge of commands, but when you get used to it, this can be the fastest way of configuring devices. You can use **question mark** (**?**) in each mode to display a comprehensive list of commands. You can also use question mark while typing the keywords to get list of available keywords matching the starting string or after typing them to get the list of arguments and optional parameters.
- **Config** - This is the fast way of **basic** configuration. This allows you to set device name, IP addresses on interfaces etc. The extended configuration still needs to be performed using the CLI. Wireless (home) routers are the exception, because they don't have CLI and all the configuration is performed using

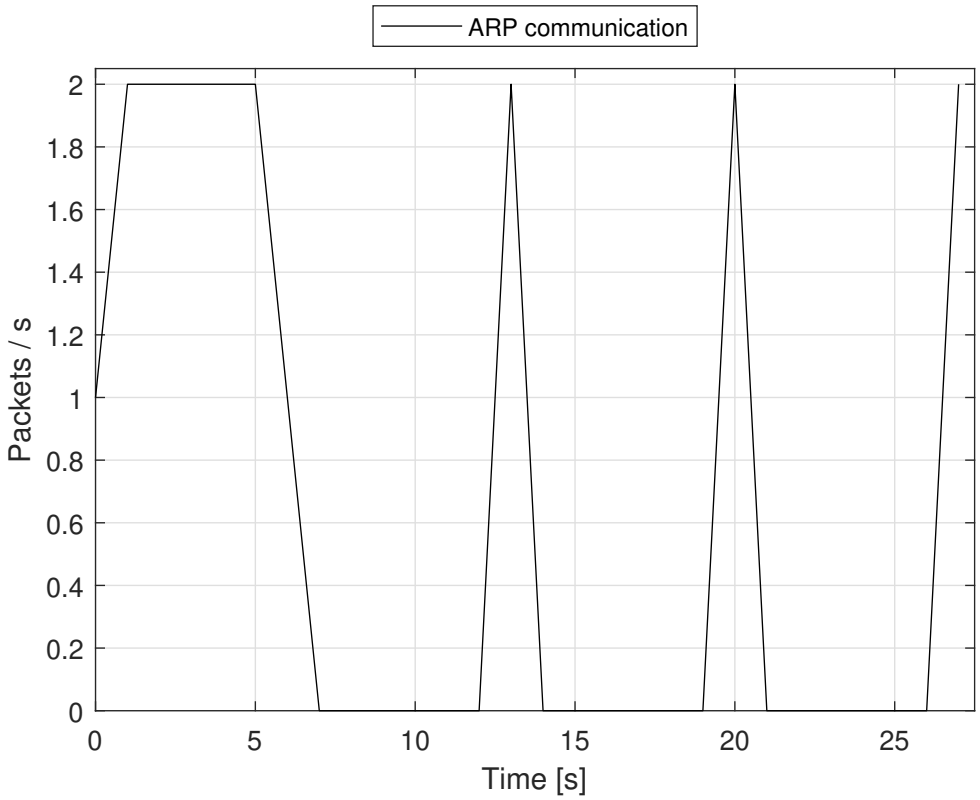


Fig. A.11: Packets sent during ARP communication.

the GUI (Graphic User Interface).

- **Desktop** - This is the way of configuring end devices like PCs and servers. You find here many icons representing user network routine. The examples are IP Configuration, Terminal, Web Browser and Email.

Cisco CLI hierarchy

There are 3 basic modes in which you operate while working with the Cisco CLI [5]:

- **User EXEC mode** – Only basic network monitoring commands are available. Mode is identified by the `>` prompt:
  `Switch>`
- **Privileged EXEC mode** – This mode should be always password protected as you can get to all the commands (including displaying running configuration, routing table etc.) and to all the modes. Mode is identified by the `#` prompt:
  `Switch#`
- **Configuration mode** - You get to the configuration mode from the **Privileged EXEC mode**. Configuration mode is not password protected as you

access it from the most authorized mode. You write all the configuration commands here concerning the global router settings or enter the interface configuration mode (to set individual interfaces) and other modes. Configuration mode is identified by the `(config)` keyword.

```
Switch(config)#
```

Packet Tracer modes

There are 2 modes in which you can operate in Packet Tracer: **Realtime** and **Simulation.**

1. **Realtime** – In the Realtime mode, the communication occurs as it would be in the real word. This means that after using network utilities (like ping, traceroute etc.) or visiting the web sites you see the output immediately.
2. **Simulation** – In the Simulation mode, individual packets are being tracked. You can examine the content of packets while they move from 1 device to another.

Environment description

Packet Tracer is divided to the following sections (see Fig. A.12):

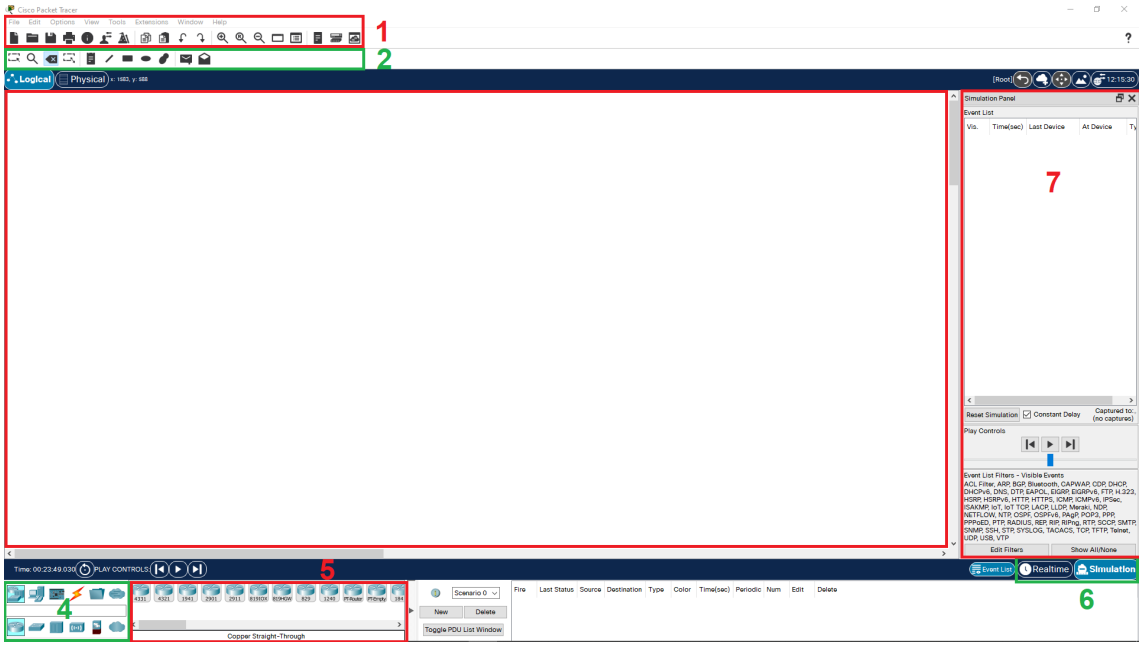


Fig. A.12: Packet Tracer environment.

1. The **Menu** and **Main toolbar** are used to do some basic actions like saving and opening the file, setting the global software preferences etc.

2. This **toolbar** is used for the Workspace in which you can:
   - **Select** more devices.
   - **Inspect** devices tables (routing table, ARP table etc.).
   - **Delete** individual elements.

   Then you can draw shapes to differentiate network areas.
3. In the **Workspace** the topologies are created.
4. **Network components** contain all the available elements (devices, cables...) in Packet Tracer. There are 2 rows [6]:
   - Upper row contains device groups based on their type.
   - Bottom row contains subgroups of the selected device group. For example the **Network Devices** group contains **Routers**, **Switches**, **Hubs** subgroups etc.
5. **Device-specific selection box** contains specific individual devices available in the selected group/subgroup.
6. **Realtime/Simulation tabs** allow you to switch the mode in which you want to simulate.
7. If the **Simulation** mode is selected, new window **Simulation Panel** appears. It contains the following sections:
   - **Event List** – The packet flow appears here.
   - **Reset Simulation** – Clears all the current packet flow (displayed in the Event List)
   - **Play Controls** – Here you can manually click on the **Go Back to Previous Event** and **Capture then forward** buttons to move simulation one step back or forward (step is meant moving packet from one device to another). You can also use **Play** button to run the simulation automatically without the need for manual clicking. The speed of simulation is controlled by the slider below.
   - **Event List Filters – Visible Events** – Displays protocols which are being traced. In the **Edit Filters** you can manually select which protocols (packets) you want to trace. By clicking on the **Show All/None** button you select all the protocols to be traced or none.

A.3.1 Objective 5

1. Open the Packet Tracer program.
2. In the **Network components** section, select the **Network Devices** group. Then select the **Switches** subgroup and in the **Device-specific selection box** select the switch named **2960**. You just click on it and then click to the **Workplace** where you want it to be placed or just follow the principle **Drag and drop**, where you drag the switch to the Workspace and then drop it.
3. Now let's add the PCs. Select the **End Devices** group and add **4 PCs** to the Workspace.
4. Now you will connect the devices. Select the **Connections** group and choose **Copper Straight-Through** cable (the solid black line icon). Notice your cursor changes. Click on the first PC and choose the **FastEthernet0** interface. Then click on the switch. Notice it has 24 Fast Ethernet ports and 2 Gigabit Ethernet ports. Select FastEthernet0/1[8]. Connect the rest 3 PCs the same way as the first computer. Use consecutive port numbers. Wait for all the lights to shine green and continue to the next step.
5. Save your current progress by clicking on the **File > Save As ...** and choose the appropriate name for the lab.

A.3.2 Objective 6

1. Once you have the topology created, you will generate the communication. But first you must configure device addresses. Click on the first PC, select **Desktop** tab and open **IP Configuration** window. Here you can set IPv4 and IPv6 configuration. We will use IPv4 address space **192.168.1.0/24** which means you have 254 available addresses for hosts (1 is reserved as the network address and 1 as the broadcast address). Set the host **IPv4 Address** and **Subnet Mask** to the first address within the host range (subnet mask should be filled in automatically based on the classful addressing)[9]. Configure the rest of the computers the same way. Use the consecutive addresses (see Tab. A.3).
2. Save your current progress by clicking on the **File > Save**.

[8]NOTE: After you select the interface, the devices are interconnected. You can notice that there is a green triangle by the PC and orange triangle by the switch. Green color means everything is working. Orange color means that something is happening in the background before putting it to the working or other state. In this case, orange color represents STP process running. STP is beyond the scope of this lab.

[9]NOTE: As you will be communicating only within the local network, you don't have to specify the **default gateway**. But it is necessary to specify it when communicating with other than local networks.

3. Open the first PC, select **Desktop** tab and open the **Command Prompt** window. Issue the `ipconfig /all` command (press Spacebar key until the prompt appears again). Notice the MAC address is now in the different format than on your local PC. Create a table similar to the Tab. A.3 and write the IPv4 and MAC addresses of all PCs to it.
   Be careful to write the Ethernet MAC address and not the Bluetooth MAC address!

Tab. A.3: Table of PC addresses.

PC	IPv4 Address	MAC address
PC0	192.168.1.1	0001.C9BC.63CC
PC1	192.168.1.2	000A.F323.6BDC
PC2	192.168.1.3	0001.9781.2486
PC3	192.168.1.4	0030.F2D1.A4E1

4. Switch to the **Simulation** mode. Use **Show All/None** to clear the filter list. Click on the **Edit Filters** button and tick **ARP** int the **IPv4** tab. Close the filter window.
5. You are going to generate ARP communication between PC0 and PC2. But before it, check if the ARP tables are empty on both computers by issuing the `arp -a` command in the Command Prompt.
6. On the PC0 issue `ping 192.168.1.3`. Notice that the ARP message was generated in the **Event List**. The following items are available:
   - **Time (sec)** – Timestamp of a packet.
   - **Last Device** – Name of the device the packet came from.
   - **At Device** – Name of the device where the packet currently is.
   - **Type** - Packet protocol.

   You can display the contents of the packet by clicking on the row in the Event List. The **OSI Model** is displayed. Notice that only 2 lowest layers contain data. You can see source and destination addresses of the Ethernet frame and encapsulated ARP packet inside the L2. Under the OSI Model, there is a description what currently happens on the device. You can also click on the **Outbound PDU Details** to display the contents of the packet.
   What is the source and target IP address?
   What is the source and target MAC address?
   What is the destination MAC address?
   What is the Opcode value?

7. Before continuing further, you are going to examine the **MAC address table** of the switch. Click on the switch end select **CLI** tab. If no prompt appears, click on the screen and press enter. You are now in the **User mode** identified by the

```
Switch>
```

prompt. Enter the `enable` command to enter the privileged EXEC mode. Now issue the `show mac-address-table` command to display the contents of the MAC table. You can notice that all the PCs are already present as seen in the Tab. A.4. Can you guess why? Table contains the following values:

- **Vlan** – Vlan the port is assigned to.
- **Mac Address** – MAC address of the device connected to the port.
- **Type** – Type of the record (static or dynamic).
- **Ports** – Port the frame was received on.

Tab. A.4: MAC table.

Vlan	Mac Address	Type	Ports
1	0001.9781.2486	DYNAMIC	Fa0/3
1	0001.c9bc.63cc	DYNAMIC	Fa0/1
1	000a.f323.6bdc	DYNAMIC	Fa0/2
1	0030.f2d1.a4e1	DYNAMIC	Fa0/4

Switch builds a MAC table to build the topology (map connected devices) so it can forward traffic directly to the host based on the destination MAC address. When the packet arrives at the switch, it checks its MAC table if the inbound port is already mapped to the source MAC address. If not, the record is created. Next the switch checks the destination MAC address. If the record of the destination MAC address and outbound port is present in the table, frame is forwarded **only** to this port. If not, the frame is flooded out the **all** ports (including those present in the MAC table) **except the inbound port**. The same process happens if the destination MAC address is set to **broadcast**. While ARP table stores records of the logical addresses mapped to the physical addresses, MAC table stores records of the ports mapped to the physical addresses. To see the process of filling the MAC table, clear the current content by issuing the `clear mac-address-table`.

8. Click **Capture then forward**. Now you can see that the frame arrived at the switch. Examine the MAC table.
   What is the current content?

Now explore the ARP packet (in the Event List). Notice that the **Inbound PDU Details** tab appeared. This is because some devices (eg. router) change the MAC address of the frame. If you compare the Inbound and Outbound PDU Details, they are the same in this case.

9. Click **Capture then forward**. The switch floods the frame out all the ports except the inbound port. As you can see, all the PCs receive the frame, but only one accepts it (see Fig. A.13) since it recognizes its IP address in the "**Target IP Address**"field.

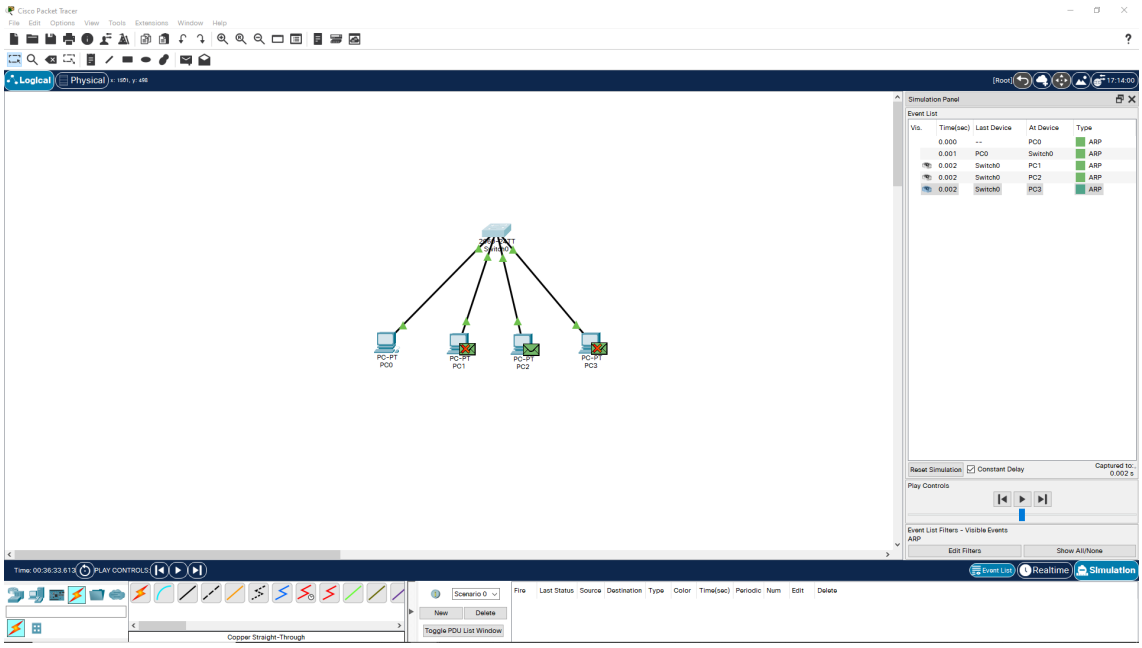


Fig. A.13: Switch sending the broadcast frame.

10. Click **Capture then forward**. You can see that PC2 sent packet to the switch. Examine PC2's ARP table.
    Does it contain some record? What device does the address combination correspond to?
    Now examine the switch MAC table and check if the content changed. Then display the ARP packet. Check the **Inbound PDU Details** as it has the content generated by the PC2.
    What is the source and target IP address?
    What is the source and target MAC address?
    What is the destination MAC address?
    What is the Opcode value?
11. Click **Capture then forward**.
    Where did the switch send packet? Did the switch flood packet the same way as the ARP request?

12. Click **Capture then forward** again. Return to the Command Prompt of PC0. You should see that the ICMP replies were successfully received[10]. Display the contents of the ARP table.
    Does it contain some record? What device does the address combination correspond to?
13. Save this topology for future use. You are going to continue with the files from previous lab in each lab.

A.4 Final questions

1. What is the destination MAC address for the ARP request?
2. What is the Opcode (Operation) value for the ARP request and response?
3. What values does the ARP table contain?
4. What is the size of ARP packet?
5. What is the difference between static and dynamic records in ARP table?
6. At which OSI layers does the ARP operate?
7. What is the difference between ARP and MAC table?
8. What does the switch do with a packet whose destination address is not contained in the MAC table?

[10]You did not see any ICMP packets as the filter is set to ARP only.

Literature

[1] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826. In: *RFC Editor* [online]. 1982 [cit. 28. 11. 2021]. Available at: <https://www.rfc-editor.org/info/rfc826>.

[2] SHARPE, Richard, Ed WARNICKE and Ulf LAMPING. Wireshark User-s Guide. In: *Wireshark* [online]. 2020 [cit. 28. 11. 2021]. Available at: <https://www.wireshark.org/download/docs/Wireshark%20User's%20Guide.pdf>.

[3] EDWARDS, Jeff. IT Basics: The Ping Utility Explained. In: *WhatsUp Gold* [online]. 2020 [cit. 29. 11. 2021]. Available at: <https://www.whatsupgold.com/blog/it-basics-the-ping-utility-explained>.

[4] Wireshark. Gratuitous_ARP. In: *Wireshark* [online]. 2020 [cit. 22. 05. 2022]. Available at: <https://wiki.wireshark.org/Gratuitous_ARP>.

[5] Cisco Systems, Inc. Cisco IOS Command Hierarchy. In: *Cisco* [online]. 2002 [cit. 01. 12. 2021]. Available at: <https://www.cisco.com/E-Learning/bulk/public/tac/cim/cib/using_cisco_ios_software/02_cisco_ios_hierarchy.htm>.

[6] A, Jesin. *Packet Tracer Network Simulator*. Birmingham: Packt Publishing, 2014. ISBN 978-1-78217-042-6.

B Lab 2 – Comparison of static and dynamic routing

In this laboratory you are going to compare static and dynamic routing and the reactions on a link failure.

Objectives

1. Create the reference topology in Packet Tracer (see Fig. B.1).
2. Set up addressing on all the devices.
3. Explore the contents of routing tables on the routers.
4. Configure static routing so all the devices can reach each other.
5. Remove the link connecting R1 and R2 and explore if it affects the reachability between LANs.
6. Reconnect the link and replace static routing with dynamic routing using the RIP protocol. Verify the state of full convergence.
7. Remove the link connecting R1 and R2 again and explore if it affects the reachability between LANs.

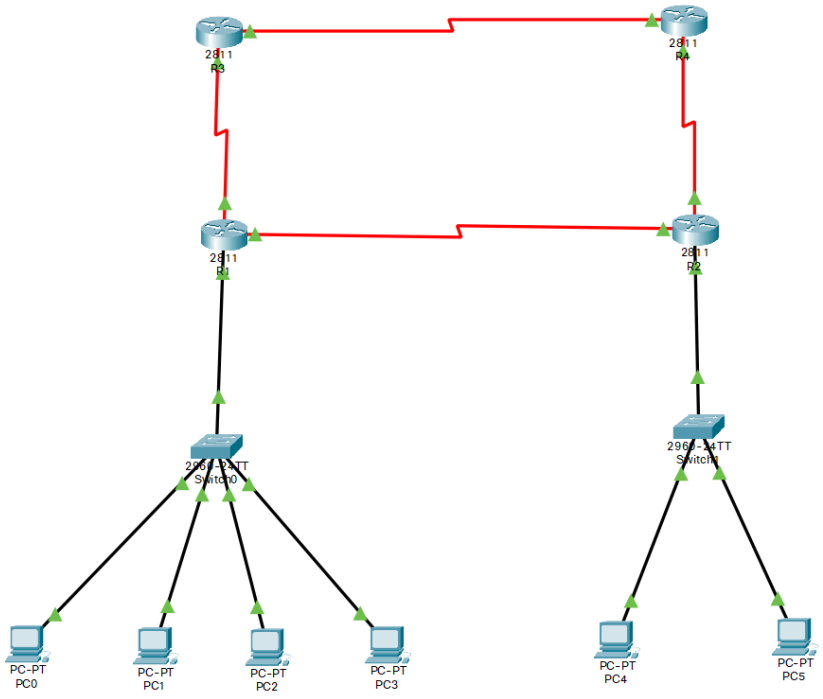


Fig. B.1: Lab 2 reference topology.

B.1 Introduction

Routing is the process of searching for the best way to reach the destination and sending the data this way. The process takes place at layer 3 of the reference ISO/OSI model, meaning it uses basically logical addresses for the computing. Devices performing routing are called routers. Many parameters are considered during the decision of choosing the best route, including administrative distance, prefix length and metrics. There are two types of routing – static and dynamic.

Static routing

Static routing is efficient in small networks. Administrator manually configures routes on all the routers so that they can reach each other. The advantages of static routing are ease of configuration inside small networks (a few routers) and full administrator's control over the routes packets are taking to reach the destination. Then there are performance advantages, because static routing requires low CPU usage and doesn't use network bandwidth to send routing updates. Concerning security, static routing is highly secure for administrator's absolute control over the routing configuration (as no fraudulent routes can be spoofed). There are many disadvantages as well. Static routing cannot adapt to the topology changes as everything is manually set up. Administrator's intervention is required every time there is a change (e.g. connection/disconnection of a network, addition of a new device and so on). With a growing network there is also increasing chance of human factor failure. Default route is an example of static routing, which is used when there is no matching address in the routing table instead of dropping a packet.

Dynamic routing

Dynamic routing is preferably used in medium to large networks. It implements routing protocols that are using certain routing algorithms. The main advantage is in the simplicity of configuration. Administrator just implements the routing protocol on routers and leaves the protocol to do the decision process of choosing the best paths. Routers exchange routing information from which they learn about all remote networks. If there is a change in the topology, routing protocols can flexibly react to it and adapt to the change by disseminating the information about the newly added (or disconnected) networks or by calculating alternative paths to the destination. The disadvantages are the complexity of routing protocols (the algorithms they are using) and less security as route spoofing can take place without additional security mechanisms. There are also higher hardware requirements (CPU processing and

memory space) and consumption of certain network bandwidth for the routing information exchange. The examples of routing protocols are RIP (Routing Information Protocol) and OSPF (Open Shortest Path First) [1], [2].

RIP protocol

Routing Information Protocol is one of the oldest ever used routing protocols. It belongs to the group of distance-vector routing protocols, which basically means its metrics (set of information about the path deciding about its priority) consists of the number of hops (routers along the path) packets must travel through to reach the destination. The maximum number of hops is 15, higher metrics means an unreachable destination (commonly defined as 16). There are 3 versions of the protocol: RIPv1, RIPv2 and RIPng. In this lab the RIPv1 (RIP version 1), defined in the RFC 1058 [3], is going to be used. RIPv1 supports classful routing, addresses are therefore assigned based on the network classes (see Tab. B.1). Classes A, B and C are used for addressing of the end and intermediary devices. RIPv2 supports classless routing and RIPng is destined for IPv6 networks.

Tab. B.1: Addresses divided into the network classes.

Class	**Prefix**	**Net mask**	**Start address**	**End address**
Class A	8	255.0.0.0	0.0.0.0	127.255.255.255
Class B	16	255.255.0.0	128.0.0.0	191.255.255.255
Class C	24	255.255.255.0	192.0.0.0	223.255.255.255
Class D	X	X	224.0.0.0	239.255.255.255
Class E	X	X	240.0.0.0	255.255.255.255

B.2 Workflow

B.2.1 Objective 1

1. From the previous lab, you have already 4 PCs connected to a switch. Open the file by clicking on the **File > Open ...** and save it as a new lab file.
2. Add another 2 PCs to the topology as well as another **2960** switch. Connect the 2 PCs to the newly added switch by using the **Copper Straight-Through** cable. They will make the second LAN (Local Area Network) in our network.
3. From the **Routers** subgroup (next to the **Switches** subgroup in the **Network Devices** group), select the **2811** type and add 4 of these.
4. Arange all the devices so the topology looks like in the Fig. B.1. Connect the switches with routers above them. For this purpose, choose **Copper Straight-Through** cable again. Use port **FastEthernet0/24** on the switch and **FastEthernet0/0** on the router on both LANs.
5. As you can see in the topology, routers are connected via a "red flash"which represents the serial connection. You find it in the Connections group. Select the **Serial DTE** (Data Terminal Equipment) and click on an arbitrary router. You can notice that there is not suitable interface for the serial connection. What can be done about it? Press ESC twice to disrupt the connection and then click on the router. The Physical tab is automatically selected. On the left side of a window, there are available modules[1] that can be added to the routers. We will use the **WIC-2T** module which adds 2 serial interfaces. Before this can happen, the router needs to be shutted down. On the right side of a window, there is a router portrayed. By clicking on a switch in the bottom right corner (next to the power cable) you turn it off. Now select the module and drag it to one of the available slots represented by the 4 smaller black boxes (the big one is not compatible with our module). Once you have that, don't forget to turn on the router again. Repeat the same process on the 3 remaining routers.
6. Select the **Serial DTE** cable again and click on a router. You can see that 2 new interfaces appeared. Connect the routers according to the Fig. B.2.
7. Rename the routers according to the same figure to avoid future misunderstandings. You can do that by clicking on the router's name (e.g. "Router0") and typing new name.
8. Save your current progress by clicking on the **File > Save**. Don't forget to save your progress after completing individual objectives!

[1]These modules expand devices in a way of adding new interfaces to it.

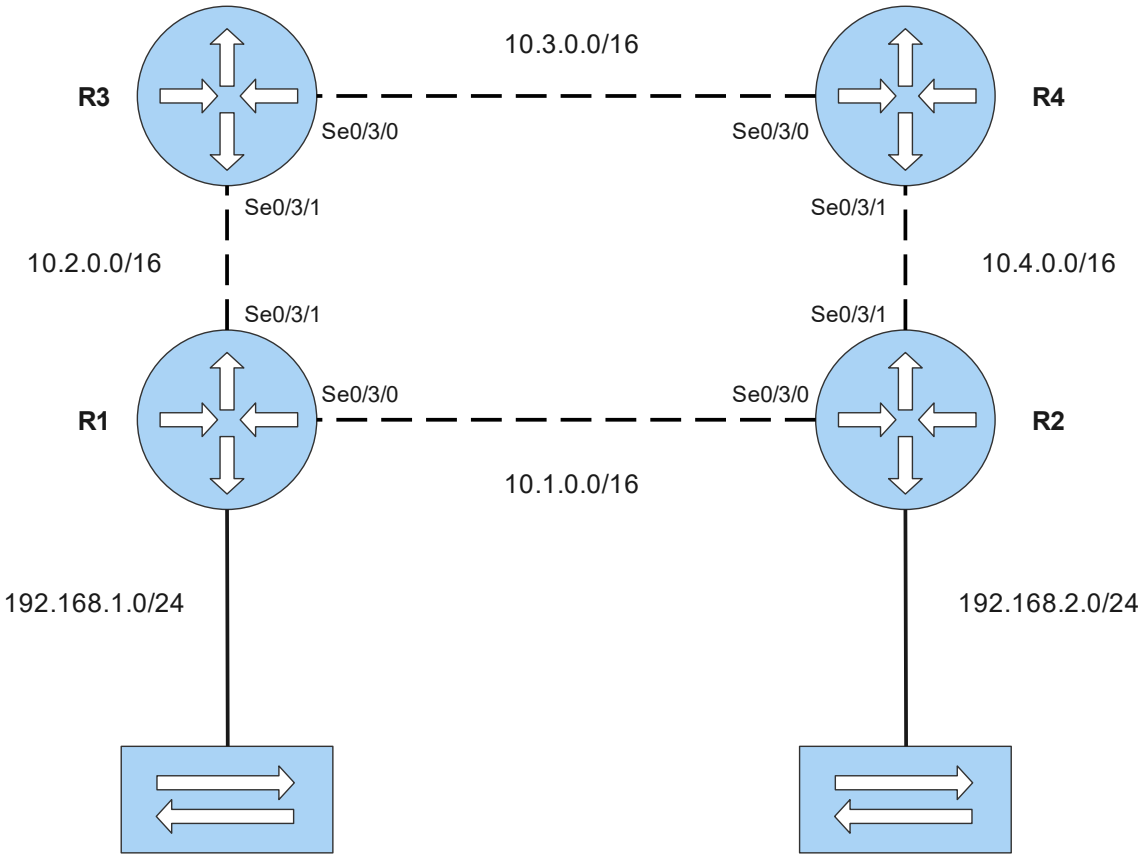


Fig. B.2: Lab 2 logical topology.

B.2.2 Objective 2

1. One note before you begin to configure. If you mess up some configuration command, you can always delete it by repeating the command but using `no` at the start of it. So for example, if you set the wrong IP address (*10.0.0.1/30* instead of *10.0.1.1/30*), the removal will look as following[2]:

```
Router(config)#interface Serial0/0/1
Router(config-if)#ip address 10.0.0.1 255.255.255.252
Router(config-if)#no ip address 10.0.0.1
255.255.255.252
Router(config-if)#ip address 10.0.1.1 255.255.255.252
```

2. You are going to follow the addressing scheme displayed in the Fig. B.2[3]. Set

[2]NOTE: It is possible that you will see command sections, like the one displayed, where the command is divided into two lines. This is due to space issues, so be aware there is only one space character between the lines (IP address and the subnet mask in this example).

[3]You can also place addresses to the topology in a form of note for better orientation. Select the **Place Note** from the **toolbar**, click to the space and write the address space.

the IP addresses on the newly added PCs the same way as in the first lab. Assign first 2 available host addresses. Now the thing you have to do on all the PCs is configuring the **Default Gateway** address which is under the subnet mask in the same tab. Choose the last host address available in the subnet PCs belong to.

3. Now let's configure addressing on the routers. Select R1 and click on the CLI tab. If you are asked for the initial configuration, type no. As you can see, you are in the User mode. For configuring the addresses, you must be in the **Configuration mode**. You get there by typing `enable` command in the User mode and then `configure terminal` in the Privileged EXEC mode. So the sequence looks like this:

```
Router>enable
Router#configure terminal
Router(config)#
```

4. For clarity, change the CLI name of a router by typing `hostname <NAME>` where `<NAME>` is a name of the router.
5. Follow the subsequent sequence of commands to configure the address of the LAN interface:

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.168.1.254 255.255.255.0
R1(config-if)#no shutdown
```

   The interface should go up and change color from red through amber to green. You can verify correct configuration by issuing `ping 192.168.1.254` from local PCs. Now let's explain all the commands. The `interface` command specifies the interface you are about to configure and is followed by the type and number of it. Inside the interface configuration, there is `ip address` command followed by the IP (Internet Protocol) address of the interface and subnet mask. The `no shutdown` command brings the interface up[4].
6. Using the same procedure you can configure all the necessary interfaces (type `exit` to return 1 level up, type `end` to return immediately to the **Privileged EXEC mode** from wherever you are). The addressing you will use corresponds to the following rules:
   - R1 and R4 are using the **first available host addresses** from their particular networks on the serial interfaces.
   - R2 and R3 are using the **last available host addresses** from their particular networks on the serial interfaces.

[4]By default, all the router interfaces are down. It is a security feature.

So for example let's configure the link between R1 and R2:

```
R1(config)#interface Serial0/3/0
R1(config-if)#ip address 10.1.0.1 255.255.0.0
R1(config-if)#no shutdown
```

```
R2(config)#interface Serial0/3/0
R2(config-if)#ip address 10.1.255.254 255.255.0.0
R2(config-if)#no shutdown
```

7. Configure the remaining interfaces on R1 and R2 (Serial0/3/1) and then the rest of the routers. Don't forget to enable interfaces.

B.2.3 Objective 3

1. Now you will explore the routing table and its contents. Routing table contains all the network records router knows about. 3 types of records can be present in the table:
   - Directly connected networks.
   - Static routes (manually set up).
   - Dynamic routes (learnt via dynamic protocol).

   To display the contents of routing table on the router, issue the `show ip route` command in the **Privileged EXEC mode**. Output should be similar to the Fig. B.3.

```
R1>en
R1>enable
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.0.0/16 is directly connected, Serial0/3/0
L       10.1.0.1/32 is directly connected, Serial0/3/0
C       10.2.0.0/16 is directly connected, Serial0/3/1
L       10.2.0.1/32 is directly connected, Serial0/3/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.254/32 is directly connected, FastEthernet0/0
```

Fig. B.3: Routing table of R1.

2. Let's examine individual items inside the table. In the upper section, there are codes available for the routes explaining how the route was learnt. In the bottom section, there are individual records of paths.

```
C   10.2.0.0/16 is directly connected, Serial0/3/1
```

   - **C** – Identifies that the route is directly connected to the router.
   - **10.2.0.0** – Network address.
   - **/16** – Prefix for the network address. The subnet mask equivalent is *255.255.0.0*.
   - **is directly connected** – Also specifies that the route is directly connected.
   - **Serial0/3/1** – Local interface the network is reachable from.

   What does the L code record represent?
3. Explore the routing tables of all routers and determine what networks do they store records about.

B.2.4 Objective 4

1. Now that you have everything prepared, let's configure static routing. To be able to do that, you have to be in the **Configuration mode**. The following command is used to configure static routes:

```
Router(config)#ip route <network-address> <subnet-mask>
{<ip-address> | <exit-interface>}
```

   Individual items have the following meaning:
   - **<network-address>** – IP address of the destination network.
   - **<subnet-mask>** – Subnet mask of the destination network.
   - **{<ip-address> | <exit-interface>}** – The curly brackets indicate that one of the items inside is required. The pipe symbol separates individual parameters. The **<ip-address>** represents the next-hop IP address, i.e. the IP address of the interface on the other side of a link. The **<exit-interface>** stands for local exit interface the network is reachable from[5].

[5] In the point-to-point networks (serial link is a typical example), the link can be seen as a tunnel with one way in and one way out. As there is only one destination, an IP address doesn't need to be specified. But this doesn't apply in the multiaccess networks such as Ethernet. Ethernet can have multiple devices connected in the network, such as several routers connected via switch. In this case, an IP address must be specified as the unique destination identifier (the ARP proccess occurs to resolve the IP address to MAC address). You can use a combination of both of these to speed up the sending process by avoiding recursive lookup when only the IP address is specified [4].

2. While configuring static routes, there is a simple rule:
   - Configure **all the remote networks** on **all the devices**[6].

   This means you have to specify every network on the router you want it to be able to communicate with. The same process must happen on the other side (router that the network is connected to) to establish bidirectional communication.
3. So for example let's configure static routes on the R1 router. Based on the logical topology (see Fig. B.2), we can determine that R1 doesn't know about the *10.3.0.0/16*, *10.4.0.0/16* and *192.168.2.0/24* networks. So now let's configure routes to these destination networks.

```
R1(config)#ip route 10.3.0.0 255.255.0.0 Serial0/3/1
R1(config)#ip route 10.4.0.0 255.255.0.0 Serial0/3/0
R1(config)#ip route 192.168.2.0 255.255.255.0
Serial0/3/0
```

   As you can see, we are applying the logic of reaching the destination via as shortest path as possible. So this means that *10.3.0.0/16* network can be reached through R3 instead of travelling through R2 and then R4. The same process is used for the *10.4.0.0/16* and *192.168.2.0/24* networks. They can be reached directly through R2 instead of sending data through R3 and R4.
4. Explore the contents of the routing table (output should look like in the Fig. B.4).
   What records did appear?
   What code do they have assigned?
5. To make it clear, let's configure another router, R3. R3 doesn't know about the *10.1.0.0/16*, *10.4.0.0/16*, *192.168.1.0/24* and *192.168.2.0/24* networks. Now the process of configuration is the same as in the third step.

```
R3(config)#ip route 10.1.0.0 255.255.0.0 Serial0/3/1
R3(config)#ip route 10.4.0.0 255.255.0.0 Serial0/3/0
R3(config)#ip route 192.168.1.0 255.255.255.0
Serial0/3/1
R3(config)#ip route 192.168.2.0 255.255.255.0
Serial0/3/1
```

   There is a little difference though. As you probably noticed, the 192.168.2.0/24 network can be reached through R1 and R4 with the same length of a way. So during the configuration, it doesn't really matter what way you choose.

[6]With the exception of using default routes and summary routes, but we are not using them in this lab.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.0.0/16 is directly connected, Serial0/3/0
L       10.1.0.1/32 is directly connected, Serial0/3/0
C       10.2.0.0/16 is directly connected, Serial0/3/1
L       10.2.0.1/32 is directly connected, Serial0/3/1
S       10.3.0.0/16 is directly connected, Serial0/3/1
S       10.4.0.0/16 is directly connected, Serial0/3/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.254/32 is directly connected, FastEthernet0/0
S    192.168.2.0/24 is directly connected, Serial0/3/0
```

Fig. B.4: R1's routing table with static routes.

6. Configure the remaining 2 routers and verify the connectivity of both LANs to all other networks. Use the `Serial0/3/0` interface on R4 to reach the R1's LAN.
   What networks must be configured on the R2?
   What networks must be configured on the R4?

B.2.5 Objective 5

1. If you have verified reachability of all the networks, let's make a change in the topology. We will remove the link connecting R1 and R2. Use the Delete tool (from the **toolbar** menu) and click on the serial link.
2. Explore the routing table of R1.
   What networks were removed from the routing table? Why?
3. Test the connectivity from R1's LAN to all the networks.
   Is R2's LAN and the 10.4.0.0/16 network reachable? If not, why is that when there still exists route via R3 and R4?
   Use the ping utility to test connectivity from any PC inside R1's local LAN to active interfaces on R4. Were they successful? If not, why is that when they are both on the same device?

B.2.6 Objective 6

1. Connect the R1 and R2 again using the same link and interfaces. Explore the routing table of R1.
   Did any new records appeared? Why? Hint: You just removed the interface, you did not delete records from the memory of the router.
2. Delete all static routes from all the routers. As mentioned in the Objective 2, repeat the commands using the `no` at the beginning of the command. So for example to clear static routes on R1:

```
R1(config)#no ip route 10.3.0.0 255.255.0.0 Serial0/3/1
R1(config)#no ip route 10.4.0.0 255.255.0.0 Serial0/3/0
R1(config)#no ip route 192.168.2.0 255.255.255.0
Serial0/3/0
```

   For speeding up the process, you can display running configuration on the router by issuing the `show running-config` command in the **Privileged EXEC mode**. Then list through its content (using the Spacebar keyboard button) until you find the route commands. Then you can copy them and paste them behind the `no` keyword. After deleting all routes, display the contents of routing table on each router to ensure that all the static routes were successfully removed.
3. Now you will configure the dynamic routing using the RIP version 1 protocol. Configuration is quite simple, see below:

```
R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 10.0.0.0
```

   By the `router rip` command you are specifying that the routing via RIP protocol will take place. Next you are adding networks after the `network` command that will participate in the routing process. **You are specifying only the directly connected networks!** In other words, all the interfaces belonging to the specified network will send and receive routing updates by default. Notice that you are not using subnet mask in the command. RIP protocol uses classful addressing for the mask determination, so if you type subnet address, it is still converted to the classful network address and according to the class, the network mask is selected. Without other command, RIPv1 is in use by default.
4. Enable RIP protocol on the remaining routers.
5. RIP is a request-response protocol so when an interface where the protocol

is enabled changes state to up, RIP request along with the contents of local routing table (with some exceptions) is sent. Routing updates (containing the records from routing table) are sent every 30 seconds by default. Wait for the state of **convergence**, which is the state where all the routers in the network have consistent information about all other networks.

6. Explore the contents of R1's routing table (output should look like in the Fig. B.5).
   How many new records are present?
   What is their code and what does it represent (see upper codes section)?

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.0.0/16 is directly connected, Serial0/3/0
L       10.1.0.1/32 is directly connected, Serial0/3/0
C       10.2.0.0/16 is directly connected, Serial0/3/1
L       10.2.0.1/32 is directly connected, Serial0/3/1
R       10.3.0.0/16 [120/1] via 10.2.255.254, 00:00:08, Serial0/3/1
R       10.4.0.0/16 [120/1] via 10.1.255.254, 00:00:28, Serial0/3/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.254/32 is directly connected, FastEthernet0/0
R    192.168.2.0/24 [120/1] via 10.1.255.254, 00:00:28, Serial0/3/0
```

Fig. B.5: R1's routing table with RIPv1.

Let's analyze the contents of new records:

```
R   10.3.0.0/16 [120/1] via 10.2.255.254, 00:00:08,
Serial0/3/1
```

- **R** – Code for the route.
- **10.3.0.0/16** – Network address with its prefix.
- **[120/1]** – Will be explained in the next lab.
- **via 10.2.255.254** – The next-hop IP address.
- **00:00:08** – Time from the last update in the [hh:mm:ss] format.
- **Serial0/3/1** – The outgoing local interface.

7. Explore the contents of R3's routing table. Recall that while configuring static routing, you could choose which route to use to reach the R2's LAN. As (according to the RIP's routing algorithm) there are 2 routes to reach this LAN

with equal distance, RIP uses both these routes and divides traffic equally between them (see Fig. B.6). This process is called **load balancing**.

```
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
R       10.1.0.0/16 [120/1] via 10.2.0.1, 00:00:12, Serial0/3/1
C       10.2.0.0/16 is directly connected, Serial0/3/1
L       10.2.255.254/32 is directly connected, Serial0/3/1
C       10.3.0.0/16 is directly connected, Serial0/3/0
L       10.3.255.254/32 is directly connected, Serial0/3/0
R       10.4.0.0/16 [120/1] via 10.3.0.1, 00:00:22, Serial0/3/0
R    192.168.1.0/24 [120/1] via 10.2.0.1, 00:00:12, Serial0/3/1
R    192.168.2.0/24 [120/2] via 10.2.0.1, 00:00:12, Serial0/3/1
                    [120/2] via 10.3.0.1, 00:00:22, Serial0/3/0
```

Fig. B.6: R3's routing table showing active load balancing.

B.2.7 Objective 7

1. Remove the link connecting R1 and R2 the same way as in the Objective 5.
2. Give it a few seconds and then explore the routing table of R1.
   Are the networks 10.4.0.0/16 and 192.168.2.0/24 still available? If yes, what route is used to reach them?
   Display the R3's routing table. Is load balancing still applied?
3. Connect the link again and save this lab for the future use.

B.3 Final questions

1. What is the difference between static and dynamic routing?
2. What is routing table and what does it contain?
3. What is the next-hop IP address?
4. What must be specified while configuring the static routes on Cisco devices?
5. What is a load balancing?

Literature

[1] JACOBS, David. Static vs. dynamic routing: What is the difference?. In: *TechTarget* [online]. 2021 [cit. 02. 03. 2022]. Available at: <https://www.techtarget.com/searchnetworking/answer/Static-and-dynamic-routing>.

[2] TAWDE, Swati. Static Routing vs Dynamic Routing. In: *EduCBA* [online]. 2022 [it. 02. 03. 2022]. Available at: <https://www.educba.com/static-routing-vs-dynamic-routing/>

[3] Hedrick, C., "Routing Information Protocol", RFC 1058, DOI 10.17487/RFC1058. In: *RFC Editor* [online]. 1988 [cit. 02. 03. 2022]. Available at: <https://www.rfc-editor.org/info/rfc1058>.

[4] GRAZIANI, Rick a Allan JOHNSON. *Routing Protocols and Concepts, CCNA Exploration Companion Guide*. Indianapolis: Cisco Press, 2007. ISBN 978-1-58713-206-3.

C Lab 3 – Dynamic routing protocol groups – Distance Vector and Link State

In this laboratory you are going to examine and compare two groups of dynamic routing protocols – Distance Vector and Link State. For the explanation of Distance Vector group, you are going to use RIPv2 and OSPF for the Link State group.

Objectives

1. Open the Packet Tracer file from the previous lab. You are going to use the same topology (see Fig. C.1). Change the addressing scheme according to the Fig. C.2.
2. Modify bandwidth of the serial link connecting R1 and R2.
3. Configure RIPv1 and explore the contents of routing table.
4. Replace RIPv1 with version 2 and examine changes in the routing table.
5. Configure OSPF and examine different approach of the protocol when selecting the routing paths.

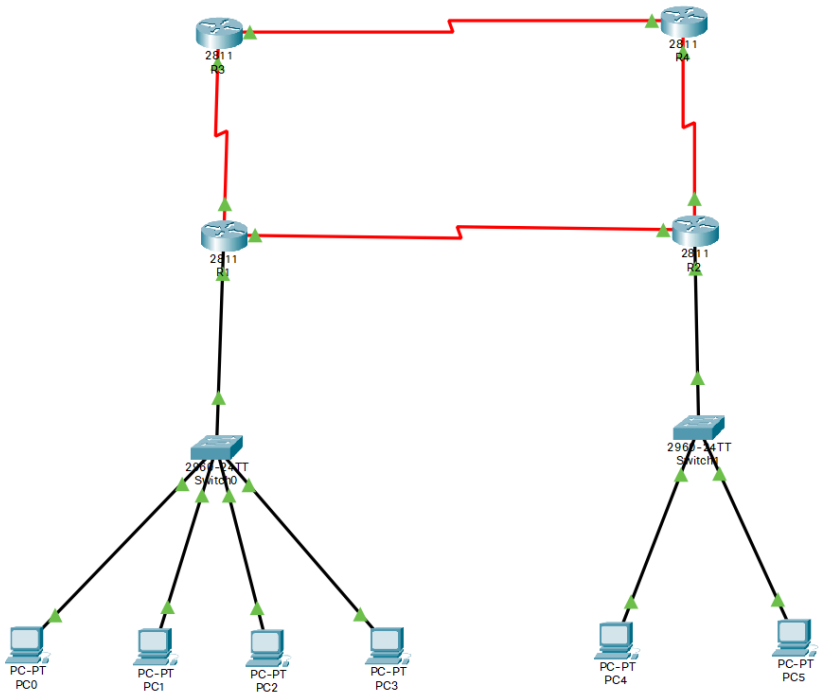


Fig. C.1: Lab 3 reference topology.

C.1 Introduction

There are 2 main groups of dynamic routing protocols, both with widely used protocols today.

Distance Vector group

Distance Vector is the original design of routing protocols. As the name implies, algorithm uses distance metric for the calculation. In this case, distance is represented by the number of hops (routers) to reach the destination. For complete vector, there must be also a direction, so the next-hop router is determined.

It is based on the **Bellman-Ford algorithm**. Routers are receiving updates from the neighbors about distant networks with the information about distance. Before flooding this information to other neighbors (except the one that the update was received from), routers analyze the information to add unknown networks to its routing table or update existing records (for example with lower metric, meaning network can be reached via shorter path). The **shortest path** is chosen and installed to the routing table. Then it increments the value for each metric in the update (typically by 1) and floods it to its neighbors. Update flooding happens at regular intervals where routers are sending its complete routing table (with some limitations that are beyond the scope of this lab). In the end, the only information that the routers maintain are its directly connected networks, distant networks and directions to reach them. Direction is represented by the next-hop router or outgoing interface. RIP (Routing Information Protocol) protocol is a typical representative of this group [1].

Link State group

With the invention of new technologies, growing of networks and number of users, Distance Vector algorithm became unsuitable, so the new method had to be implemented. Link State algorithm builds a complete map of the topology based on the received information from other routers and then it runs calculations to fill a routing table with the **fastest paths**. An interface is meant by the link and state describes its properties, such as status, IP address, subnet mask, bandwidth etc.

It uses **Dijkstra's algorithm**, also known as SPF (Shortest Path First) algorithm. After all the information is gathered in the local database, the algorithm is being run to compute SPT (Shortest Path Tree). The principle resides in placing the router, executing the algorithm, as the root and other routers as the leaves. In the end, the tree contains always one, the fastest path to reach the destination. Fastest path is represented by the lowest metric. Each router computes its SPT locally,

independently of the others. OSPF (Open Shortest Path First) protocol is probably the most used protocol from this group.

OSPF protocol

OSPF protocol is a classless Link State protocol implemented by many vendors. Currently, 2 versions are available, version 2 is defined in the RFC 2328 [2] and is used in the IPv4 networks. Version 3 exists for IPv6 networks and is defined in the RFC 2740 [3]. There are 5 types of messages that the routers are exchanging – Hello, DBD (Database Description), LSU (Link State Update), LSR (Link State Request) and LSAck (Link State Acknowledgment) [4]. Routers are not simply sending updates and waiting for the response from the neighbors, but they are forming relations with them called **neighbor adjacencies**. There are 7 states they must go through to form full adjacency – Down, Init, 2-way, ExStart, Exchange, Loading and Full [5]. There are cases where the process of forming adjacency ends in the 2-way state.

OSPF uses the concept of areas, which means that routers are grouped to the portions of network in which they share routing information. Each router must belong to some area in order to participate in the routing process. Commonly **area 0** is used as the first. If network grows larger and more areas are implemented, area 0 represents the backbone area which other areas are connected through.

Each link has associated a cost value with it that represents actual metric after calculating the cumulative cost across the network (sum of all costs). Cisco implements OSPF using the bandwidth of the outgoing interface as a cost value. The following formula is used to calculate the cost of interfaces:

$$cost = \frac{\text{Reference bandwidth}}{\text{Interface bandwidth}} = \frac{10^8}{\text{Interface bandwidth [bps]}} \tag{C.1}$$

As shown in the fraction C.1, Cisco uses 100 Mbps as its reference value of bandwidth.

C.2 Workflow

C.2.1 Objective 1

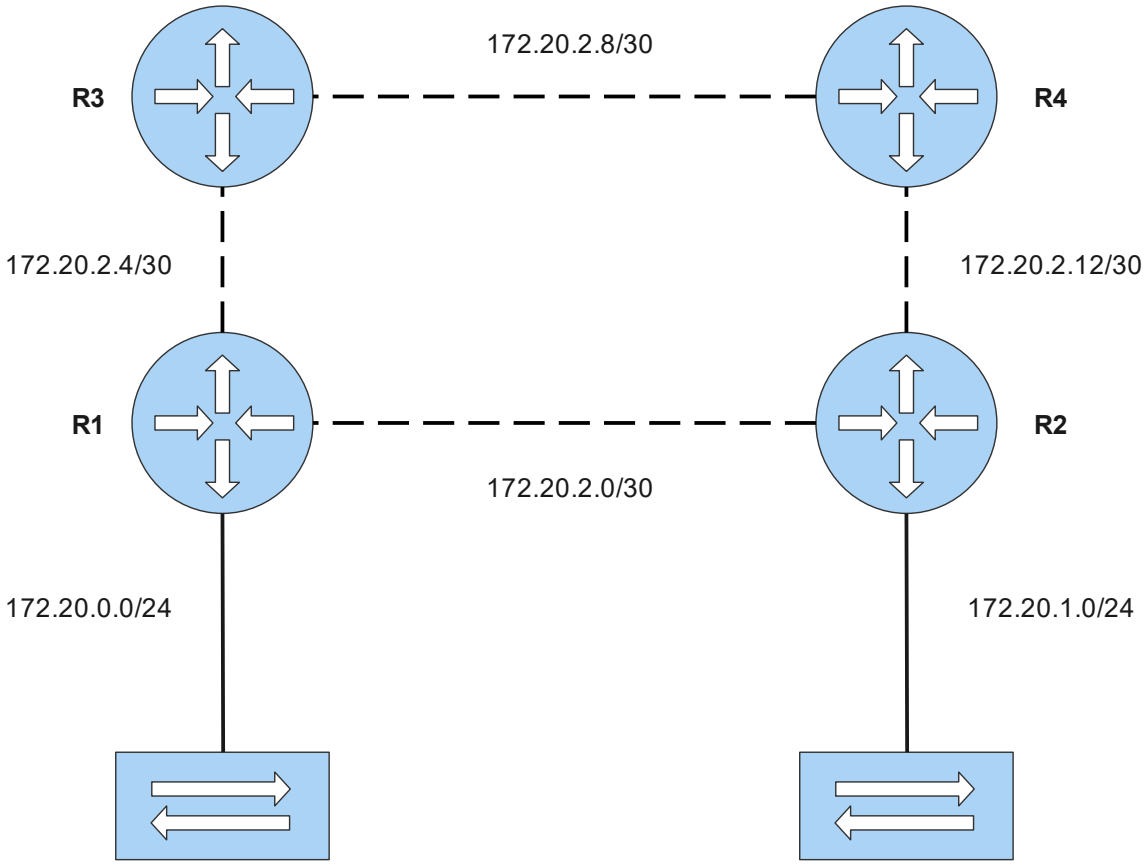


Fig. C.2: Lab 3 addressing scheme.

1. As we're using the same topology as in the Lab 2, open the file you were supposed to save and save it under a new name.
2. Before changing the addresses[1], it will be convenient to disable RIP so you can begin configuration from scratch. You can do this by typing the `no router rip` command in the **Configuration mode**.

   ```
   R1(config)#no router rip
   ```

   Issue this command on all the routers.
3. Now you can change the addressing scheme. Use addresses shown in the Fig. C.2 and follow the same rule as in the Lab 2. That means use the **first available host addresses** on the R1 and R4 routers, and use the **last available host addresses** on the R2 and R3 routers. As for LANs, use the last

[1] You are going to change the addressing scheme so you can observe the different approach of classful and classless routing protocols.

available host address for the default gateway and the first available host addresses for the PCs.

4. Before continuing to the next step, verify connectivity between PCs and their default gateway as well as connectivity between neighbor routers.
5. Save your current progress by clicking on the **File > Save**. Don't forget to save your progress after completing individual objectives!

C.2.2 Objective 2

1. Now that you have the addresses configured, let's edit the bandwidth of the serial link between R1 and R2[2]. Before doing that, let's find out what value is currently set. Do this by typing `show interface Serial0/3/0` in the **Privileged EXEC mode** on R1.

```
R1#show interface Serial0/3/0
```

   The output should look like in the Fig. C.3. You can see that the default bandwidth is 1544 kbps. Now let's change it to 128 kbps on R1. The commands to achieve this are as following:

```
R1(config)#interface Serial0/3/0
R1(config-if)#bandwidth 128
```

   Note that you must be in the interface configuration mode. You are changing the bandwidth value by the `bandwidth` command followed by the actual value in kbps. Verify the configuration by examining the interface bandwidth using the show command.
2. For the correct future calculations, this process must be performed on both ends of the link so repeat the same process on the R2 router.

C.2.3 Objective 3

1. Configure RIPv1 on all the routers. The process is the same as in the Lab 2 but with different addressing scheme. But let's add another feature. Basically, RIP enabled routers are sending updates from all the interfaces participating in the routing process, meaning that updates are also flooded to the LAN (Local Area Network). As RIP updates are sent as broadcast, each device inside LAN processes these updates and drops them in the end. But this

[2]By the following procedure, you are not changing the actual physical bandwidth of the link. You are just editing the value that is used by the OSPF to compute the link cost [1]. Take it as the simulation of lower bandwidth for this lab.

```
R1#show interface Serial0/3/0
Serial0/3/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 172.20.2.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1 packets input, 28 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1 packets output, 28 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Fig. C.3: R1's default serial link bandwidth.

consumes bandwidth and computing resources of the devices inside LAN. To prevent this from happening, we can use the technique of **passive interfaces** to prevent routers from flooding updates to the LANs but still preserve LAN's participation in the routing process between routers. You can achive this by the `passive-interface <interface>` command inside RIP configuration, where `<interface>` stands for the actual router interface. So the configuration on R1 would look like this:

```
R1(config)#router rip
R1(config-router)#passive-interface FastEthernet0/0
```

2. Verify connectivity between routers and reachability of other networks from LANs.
   Were pings successful between routers?
   Can LAN devices reach other networks and vice versa?
3. You can notice that routers can reach each other but LAN devices cannot reach (and cannot be reached) other routers. How is that possible?
4. Examine R3's routing table. The output should look like in the Fig. C.4. You

can see that none LAN is present in the routing table. RIPv1 does not send subnet masks with routing updates, only the information about metric, so the calculation of mask is left to other routers. We can say that subnets are supported only if they are the same length. As LANs are in the same network range of *172.20.0.0/16* class B address space as networks between routers, but with different subnet masks, they are not sent. The process behind it can be found in the [1].

```
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 6 subnets, 2 masks
R       172.20.2.0/30 [120/1] via 172.20.2.5, 00:00:20, Serial0/3/1
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.6/32 is directly connected, Serial0/3/1
C       172.20.2.8/30 is directly connected, Serial0/3/0
L       172.20.2.10/32 is directly connected, Serial0/3/0
R       172.20.2.12/30 [120/1] via 172.20.2.9, 00:00:21, Serial0/3/0
```

Fig. C.4: Routing table of R3 with RIPv1 enabled.

C.2.4 Objective 4

1. Let's replace RIPv1 with version 2. The process is simple, just enter `version 2` in the RIP configuration mode. Perform this on all the routers.

```
R1(config)#router rip
R1(config-router)#version 2
```

2. Click **Fast Forward Time** button (represented by two arrows above the Network components section) which shifts time by 30 seconds. Test connectivity of LAN devices with other routers.
   Were pings successful?
3. Examine R3's routing table.
   What new records did appear?
4. You can see that all the networks are now reachable. As RIPv2 is classless protocol, it sends subnet masks in its update messages so CIDR (Classless Inter-Domain Routing) and VLSM (Variable-Length Subnet Mask) mechanisms are supported. Subnet sizes may vary and they are still sent.

5. Let's take a closer look to dynamically learnt records. As seen in the Fig. C.5, apart from the items you learned in the Lab 2 about, there is some item that wasn't explained. The two numbers enclosed inside square brackets and separated by slash. For example, let's take the following record:

```
R   172.20.0.0/24 [120/1] via 172.20.2.5, 00:00:21,
Serial0/3/1
```

The format is `[Administrative Distance/Metric]`.

Administrative Distance

Administrative Distance represents priority of the way that the route was learnt. If the way to the same destination is learnt from more sources, only the one from the source with **lowest** Administrative Distance value is chosen and installed to the routing table [6]. Some examples are shown in the Tab. C.1.

Tab. C.1: Table of Administrative Distances [6].

Route Source	Administrative Distance
0	Directly connected
1	Static route
110	OSPF
120	RIP

Metric

If more routes are learnt from the same routing source, the more suitable one is chosen based on the metric value. **Lower** metric represents **better** path. As RIP protocol uses the hop count as its metric, number of routers to reach the destination is considered.

6. You can see that the Administrative Distance in the example above is equal to 120, which means that the route source is RIP protocol[3]. Metric value is 1, so the destination can be reached via 1 next router. As for the *172.20.1.0/24* network, there is no more sufficient route source and there are also two routes to the same destination with the same metric. This means that both routes are installed to the routing table and load balancing is applied.

[3]Both versions 1 and 2 have the same Administrative Distance.

7. Now let's check the chosen path from the perspective of LAN devices. Open Command Prompt on any of the PCs from R1's LAN and use `tracert`[4] command followed by the IP address of any of the PCs belonging to R2's LAN (see Fig. C.6).
8. So summing all these things together, the **shortest path** (based on the number of routers along the way) is chosen.

```
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 8 subnets, 3 masks
R       172.20.0.0/24 [120/1] via 172.20.2.5, 00:00:21, Serial0/3/1
R       172.20.1.0/24 [120/2] via 172.20.2.9, 00:00:21, Serial0/3/0
                      [120/2] via 172.20.2.5, 00:00:21, Serial0/3/1
R       172.20.2.0/30 [120/1] via 172.20.2.5, 00:00:21, Serial0/3/1
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.6/32 is directly connected, Serial0/3/1
C       172.20.2.8/30 is directly connected, Serial0/3/0
L       172.20.2.10/32 is directly connected, Serial0/3/0
R       172.20.2.12/30 [120/1] via 172.20.2.9, 00:00:21, Serial0/3/0
```

Fig. C.5: Routing table of R3 with RIPv2 enabled.

```
C:\>tracert 172.20.1.1

Tracing route to 172.20.1.1 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      172.20.0.254
  2    6 ms      0 ms      4 ms      172.20.2.2
  3    *         0 ms      1 ms      172.20.1.1

Trace complete.
```

Fig. C.6: Output of tracert command with RIPv2 configured.

[4]Tracert (or traceroute) is a network utility to trace the route from source to destination and list all the routers along the path. It is based on the ICMP messages (Windows) or UDP datagrams (Cisco, Linux) in conjunction with TTL value inside IP packet. The output consists of RTT (Round Trip Time) and IP address of the device [7].

C.2.5 Objective 5

1. Let's move to the second crucial part of this lab, and that is configuration of OSPF. To demonstrate the purpose of Administrative Distance, you are going to preserve the RIP protocol. So in the end, you will have 2 routing protocols configured on the same router.
2. OSPF configuration is quite different from RIP. Bellow is the syntax to establish routing on the router:

```
Router(config)#router ospf <process-id>
Router(config-router)#network <network-number>
<wildcard-mask> area <area-id>
```

   Now let's explain meaning of the items:

   - **<process-id>** – Number of process that is to be run (or already running in case of configuration adjustment) on the router. This ID is locally significant, each router can have different number.
   - **<network-number>** – The address of the connected network.
   - **<wildcard-mask>** – Wildcard mask is the inverse of a subnet mask in classic format. You can calculate it by subtracting classic subnet mask from the *255.255.255.255* value. So for example, subnet mask in the classic format *255.255.255.0* would be expressed as *0.0.0.255* in the wildcard format.
   - **<area-id>** – Number of the area which the network (more specifically the router interface) belongs to. Information about the network is propagated inside this area.

   For the clarification, OSPF configuration is shown on the R1:

```
R1(config)#router ospf 1
R1(config-router)#passive-interface FastEthernet0/0
R1(config-router)#network 172.20.0.0 0.0.0.255 area 0
R1(config-router)#network 172.20.2.0 0.0.0.3 area 0
R1(config-router)#network 172.20.2.4 0.0.0.3 area 0
```

3. To verify that both routing protocols are active on the router, issue the `show ip protocols` in the **Privileged EXEC mode**. There are informations about the current configuration of routing protocols. You should find two lines as following:

- Routing Protocol is "rip".
- Routing Protocol is "ospf 1".

4. Before continuing further, try to calculate the OSPF metric to reach the destinations from the perspective of R1 based on the C.1 equation in the Introduction section. Remember, metric is based on the cumulative cost, which means sum of all the costs along the way to reach the destination (destination cost included). Some of the costs are shown in the Tab. C.2[5].

Tab. C.2: Cost of certain interfaces on the Cisco devices.

Interface	**Cost**
Serial (Packet Tracer default)	64
Ethernet	10
FastEthernet	1
GigabitEthernet	1

5. Configure OSPF on the R2 router.
6. Explore R1's routing table. The output should look like in the Fig. C.7.
   What routes are reachable via OSPF? Does the metric correspond to your cost calculations?
   What networks are still reachable via RIP? Why?
   You can see that both RIP and OSPF learnt networks are present in the routing table. RIP learnt networks leading to the same destinations as those learnt via OSPF still exist, but router chooses the routing source with lower Administrative Distance value. In this case, OSPF with the value of 110 is the preferred one over RIP with the value of 120.
7. Configure OSPF on the rest of the routers.
8. Click the **Fast Forward Time** button and then perform the same tracert test as in the Objective 4 and figure out what path is chosen to reach the R2's LAN.
   What path is used? Can you determine why before continuing to the next step?
9. Explore the contents of the R1's routing table. See Fig. C.8 for the current output. You can see that there are no more RIP learnt paths. Every distance network is reachable via OSPF.

[5]You can notice that FastEthernet and GigabitEthernet interfaces have the same cost. As the reference value is 100 Mbps, interfaces faster than FastEthernet would not have integer values, which is required, and therefore are rounded. There are mechanisms to set more precise calculating or explicitly define cost.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 9 subnets, 3 masks
C       172.20.0.0/24 is directly connected, FastEthernet0/0
L       172.20.0.254/32 is directly connected, FastEthernet0/0
O       172.20.1.0/24 [110/782] via 172.20.2.2, 00:35:07, Serial0/3/0
C       172.20.2.0/30 is directly connected, Serial0/3/0
L       172.20.2.1/32 is directly connected, Serial0/3/0
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.5/32 is directly connected, Serial0/3/1
R       172.20.2.8/30 [120/1] via 172.20.2.6, 00:00:09, Serial0/3/1
O       172.20.2.12/30 [110/845] via 172.20.2.2, 00:35:07, Serial0/3/0
```

Fig. C.7: Routing table of R1 with OSPF enabled.

Does the metric of new networks correspond to your calculated cumulative cost?

You can see that although the R2's LAN can be reached directly through the neighboring router, it is not the preferred way. As OSPF uses bandwidth for the metric calculation, more appropriate way was selected. The cumulative cost of three times default serial link bandwidth is still lower than the cost of our modified link.

10. Putting all these pieces of information together, OSPF chooses the **fastest path**.
11. Save your current progress for the future lab.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 9 subnets, 3 masks
C       172.20.0.0/24 is directly connected, FastEthernet0/0
L       172.20.0.254/32 is directly connected, FastEthernet0/0
O       172.20.1.0/24 [110/193] via 172.20.2.6, 00:02:32, Serial0/3/1
C       172.20.2.0/30 is directly connected, Serial0/3/0
L       172.20.2.1/32 is directly connected, Serial0/3/0
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.5/32 is directly connected, Serial0/3/1
O       172.20.2.8/30 [110/128] via 172.20.2.6, 00:03:20, Serial0/3/1
O       172.20.2.12/30 [110/192] via 172.20.2.6, 00:02:32, Serial0/3/1
```

Fig. C.8: Routing table of R1 with complete OSPF.

C.3 Final questions

1. What is used as a metric for Distance Vector routing protocols?
2. What is used as a metric for Link State routing protocols?
3. What is the difference between RIP version 1 and version 2 in terms of subnetting?
4. What criteria are considered while choosing the best path to the routing table?
5. If the same network is learnt via RIP and OSPF, what path is added to the routing table? Why?

Literature

[1] GRAZIANI, Rick a Allan JOHNSON. *Routing Protocols and Concepts, CCNA Exploration Companion Guide*. Indianapolis: Cisco Press, 2007. ISBN 978-1-58713-206-3.

[2] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328. In: *RFC Editor* [online]. 1998 [cit. 17. 03. 2022]. Available at: <https://www.rfc-editor.org/info/rfc2328>.

[3] Coltun, R., Ferguson, D., and J. Moy, "OSPF for IPv6", RFC 2740, DOI 10.17487/RFC2740. In: *RFC Editor* [online]. 1999 [cit. 17. 03. 2022]. Available at: <https://www.rfc-editor.org/info/rfc2740>.

[4] IBM. Packet types for OSPF. In: *IBM* [online]. 2021 [cit. 17. 03. 2022]. Available at: <https://www.ibm.com/docs/en/i/7.1?topic=concepts-packet-types-ospf>

[5] O., Samuel. OSPF Neighbor Adjacency. In: *Expert Network Consultant* [online]. 2018 [cit. 17. 03. 2022]. Available at: <https://www.expertnetworkconsultant.com/configuring/ospf-neighbor-adjacency/>.

[6] Cisco. What Is Administrative Distance?. In: *Cisco* [online]. 2020 [cit. 18. 03. 2022]. Available at: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html>.

[7] N-able. What Is Traceroute and How Does It Work?. In: *N-able* [online]. 2020 [cit. 19. 03. 2022]. Available at: <https://www.n-able.com/blog/what-is-traceroute-how-does-it-work>.

D Lab 4 – TCP and UDP

You are going to learn about transport layer protocols which provide end-to-end connectivity. There are two main protocols used – TCP and UDP. There is also SCTP protocol, but it is out of scope of this lab.

Objectives

1. Analyze UDP packets using the DNS protocol in Wireshark.
2. Analyze TCP packets using the DNS protocol in Wireshark.
3. View graphs and compare the two protocols.
4. Generate HTTP traffic in Wireshark.
5. Create the reference topology in Packet Tracer (see Fig. D.1), then configure addresses and add the new network to OSPF routing process.
6. Start the DNS and HTTP services and create a web page on the server.
7. Configure the DNS server address on the PCs.
8. Access the server web page and analyze the traffic.

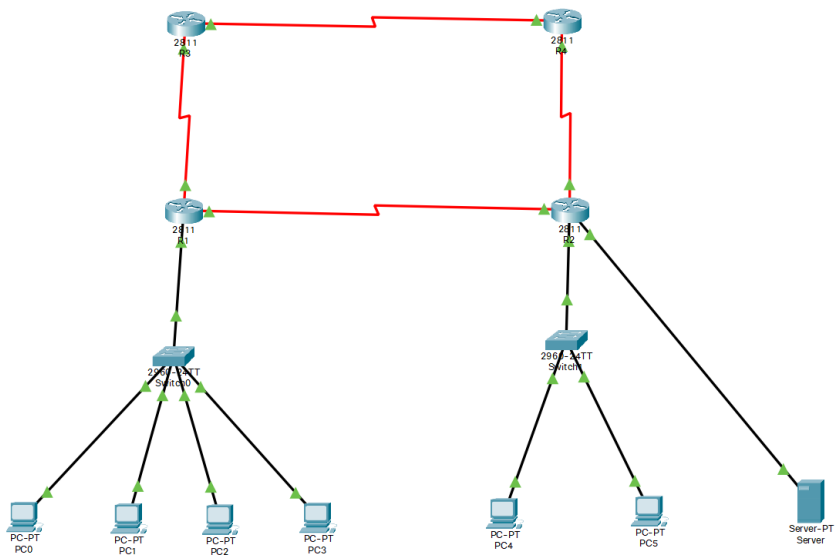


Fig. D.1: Lab 4 reference topology.

D.1 Introduction

As you can already know, IP addresses (network layer) are used to uniquely identify the devices on the internet. IP packets are used to deliver data to the proper device. In reality, it's not the computers that communicate themselves, but the processes running on them. Therefore, there is one other layer needed for addressing to differentiate local processes on the device. Transport layer protocols use port numbers as addresses. The numbers are used from the range of 0–65535:

- **Well-known ports** – This group uses numbers from 0 to 1023. These are used by the servers providing common services. Clients commonly use them as destination ports when initiating communication.
- **Registered ports** – These range from 1024 to 49151. They are used both by the clients and servers. Clients can have certain range reserved (based on the operating system), for server purposes they must be registered by the IANA.
- **Dynamic ports** – Ports ranging from 49152 to 65535 are dynamically assigned to the clients when initiating communication.

TCP and UDP ranges are the same, but independent of each other. It is common for well-known ports that even though application is using only TCP or UDP, the same port number from both ranges is assigned to it.

UDP

UDP (User Datagram Protocol) is defined in RFC 768 [1]. It is considered as unreliable and connectionless protocol [2]. Unreliable means that there is no acknowledgment system to confirm the correct reception of the data sent. UDP just sends data and doesn't care if they arrive or not. Therefore, reliability is left to the applications. Connectionless implies that there is no verification of existence of the destination and no virtual circuit for data transmission is established before the actual communication begins. UDP receives data from the application layer, adds header to them and the UDP **datagram** is created. The structure of datagram can be seen in the Fig. D.2.

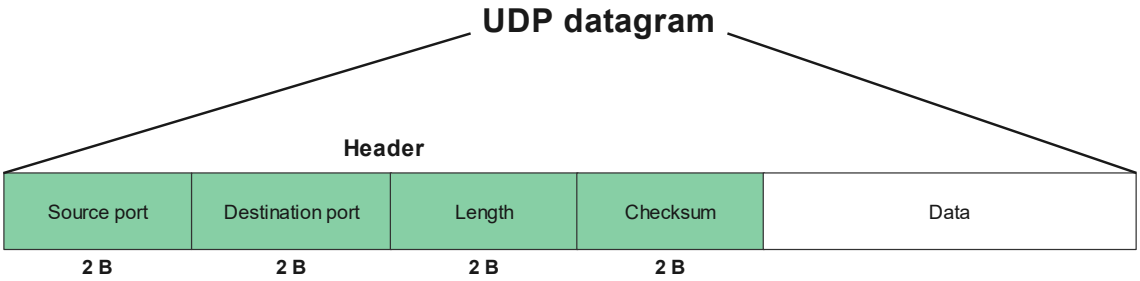


Fig. D.2: UDP datagram structure.

The header consists of:

- **Source port** – Used by the device (process) that wants to communicate (for example sending request).
- **Destination port** – This identifies the destination process on the device the data are destined for. When the destination answers (for example sends response), source and destination ports are switched.
- **Length** – The length of the whole datagram (header + data) in bytes.
- **Checksum** – This is used by the destination to check if some errors occured during the transmission.

You can see that the header (additional overhead for transmission) is just 8 bytes long. It is one of the features that is suitable for many applications (and often required). UDP is simple protocol that uses low overhead, does not establish connections between devices, does not acknowledge data and therefore is considered fast. This is utilized by applications using simple request-response communication model and by **multimedia**. Multimedia are using UDP, because it allows them to use multicast and broadcast addresses (which are also used by routing protocols to send updates). They also accept certain level of data loss, which means one hundred percent reliable delivery is not required.

DNS

DNS (Domain Name System) protocol is used by every user whether he knows it or not. This protocol makes life easier for anyone who wants to use internet communication (for accessing web pages, communicating with other devices etc.). It allows to assign names to the devices instead of accessing them via their logical addresses (for example accessing domain *www.youtube.com* instead of using its IP address *142.251.37.110*). This works also in reverse naturally. So in other words, DNS translates domain names to its respective IP addresses and vice versa. The protocol implements standard query-response model. Records are stored as a combination of IP address and domain name. DNS uses hierarchical design. First, local cache on the PC is examined for the presence of dynamic records or **hosts** file for the presence of static records. If there is no match, DNS server is contacted (local server and if there is no match, then the ISP server is contacted). If the server does not store record either, it uses root (superior) servers that lead him to the lower levels and to the desired domain name. DNS server then stores this record and informs PC about it. DNS uses TCP and UDP port number 53.

TCP

TCP (Transmission Control Protocol) is defined in the document RFC 793 [3]. As compared to the UDP protocol, TCP is realiable and connection oriented. Reliability is achieved by sending acknowledgment messages to the devices the data were received from. As regards connection orientation, before transmitting data, the devices must first establish virtual circuit. This procedure is called **three-way handshake**. Devices must agree on the connection, set the numbering and acknowledge the messages received from the other end. Then the data can be transmitted bidirectionally. After all the information are transmitted, the connection must be ended. Normally the procedure is accomplished by 4 messages (four-way handshake), but faster alternative is possible. TCP receives application data in form of byte stream and divides them to the blocks called **segments** by adding header to them. A header containing only the mandatory items takes up 20 bytes. The segment structure is shown in the Fig. D.3.

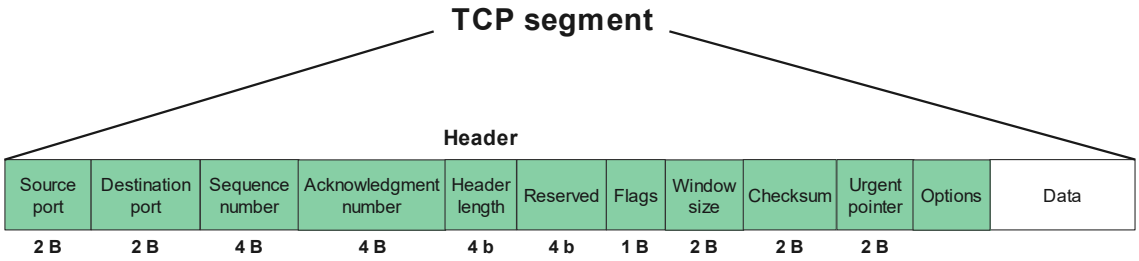


Fig. D.3: TCP segment structure.

The following items can be found inside TCP header:

- **Source port** – The same meaning as in the UDP. Identifies number of the local process.
- **Destination port** – Identifies number of the destination process.
- **Sequence number** – As TCP is reliable protocol, it needs to maintain information about the data sent and received. Each byte is assigned sequence number to reach this. The sequence number in the TCP header determines the first byte of transmitted data inside segment.
- **Acknowledgment number** – Devices acknowledge received data by this number. Every sequence number (byte) less than the value in this item is acknowledged. This also means, that this value determines the next byte (sequence number) that the sender is expecting from the other device.
- **Header length** – Number of 4-byte items in the header.
- **Reserved** – Space reserved for the future use.
- **Flags** – This is important part of the TCP. The state of communication can be determined by analyzing flags. There are 8 flags (each represented by 1 bit),

but for the purpose of this lab only a few are important:
  - **SYN** – This flag is used during the initiation of connection. Parameters of the connection are determined, like ISN (Initial Sequence Number), window size, optional parameters etc.
  - **ACK** – When this flag is set, the segment sender acknowledges the receipt of data sent by the destination. It is common in bidirectional communication that each device sends new data and also acknowledges the data sent from the other device.
  - **FIN** – When process finishes data sending, it indicates by setting the FIN flag that it has no other data to send and wants to finish the connection. Both devices have to send this flag to end the connection. Commonly four-way handshake is used for this purpose, but there is also abbreviated three-way handshake version[1].
- **Window size** – Defines maximum number of bytes that can be sent by the other device without acknowledgment reception. It is common that the window size is changing during the communication.
- **Checksum** – The same meaning as in the UDP
- **Urgent pointer** – Used when urgent data are carried beside common application data.
- **Options** – Available options that can be used (for example during the connection initiation).

TCP provides every aspect of reliability so the application does not need to worry if the data are received correctly or received at all. It also provides flow control mechanisms to ensure continuous transmission with data loss (that the TCP can influent). Sliding window is an example of such mechanism. The character of TCP suggests that it is not very suitable for the real-time applications. On the other hand, this is exactly what **data** applications demand. Fast transmission is not the primary factor, but everything must be delivered plus delivered correctly. Examples of such protocols are FTP, SSH, SMTP, HTTP and many others.

Communication

To understand how the communication is realized, look at the Fig. D.4. The connection begins with the three-way handshake. Only the most important parameters are mentioned:

1. Client generates connection "request"by sending the message with **SYN** flag set. The ISN number is generated to the Sequence number item. The first

[1]There is also an option available using the special flag, but it is not discussed in this lab.

byte of data will be sent (after virtual circuit is established) with the number ISN + 1.
2. Server receives the message and answers with **SYN** and **ACK** flags set. It sends its own ISN generated and acknowledges the ISN sent from the client.
3. Client receives the message and acknowledges it by sending the expected data (determined by the Acknowledgment number inside header) and acknowledging the ISN of the server. The **ACK** flag is set.

Now that the connection is established, data are transmitted. In the end, the connection is ended and the process looks like following[2]:

1. The server sends its application data and sets the **FIN** and **ACK** flags. This means that the server is acknowledging the data received from the client, sends its last application data and wants to end the connection.
2. The client sends its data and acknowledges the received segment. Only the **ACK** flag is set.
3. The client then sends new message with **FIN** flag set[3] indicating that it also wants to close the connection.
4. The server receives message with the FIN flag set and acknowledges it. It sets just the **ACK** flag.

As mentioned earlier, connection termination can be abbreviated with the three-way handshake. The sequence would look like this:

1. **Server** – [FIN, ACK],
2. **Client** – [FIN, ACK],
3. **Server** – [ACK].

Both UDP and TCP are using **sockets** to uniquely identify the communicating processes over the internet. Socket is a combination of IP address and port number in the format **IP_address:port_number** (for example *192.168.0.1:51257*). In case of TCP, unique connection between 2 devices is determined by the pair of sockets, each identifying one end.

[2]In the example, server is sending the FIN first but any side can initiate the connection ending.

[3]Here the point is that the server informed client about no other application data to be sent but the client still can have data to be sent. So the client can continue with sending and server just acknowledges these data. After the local process on client has no more data to be sent, the message with FIN flag is also generated.

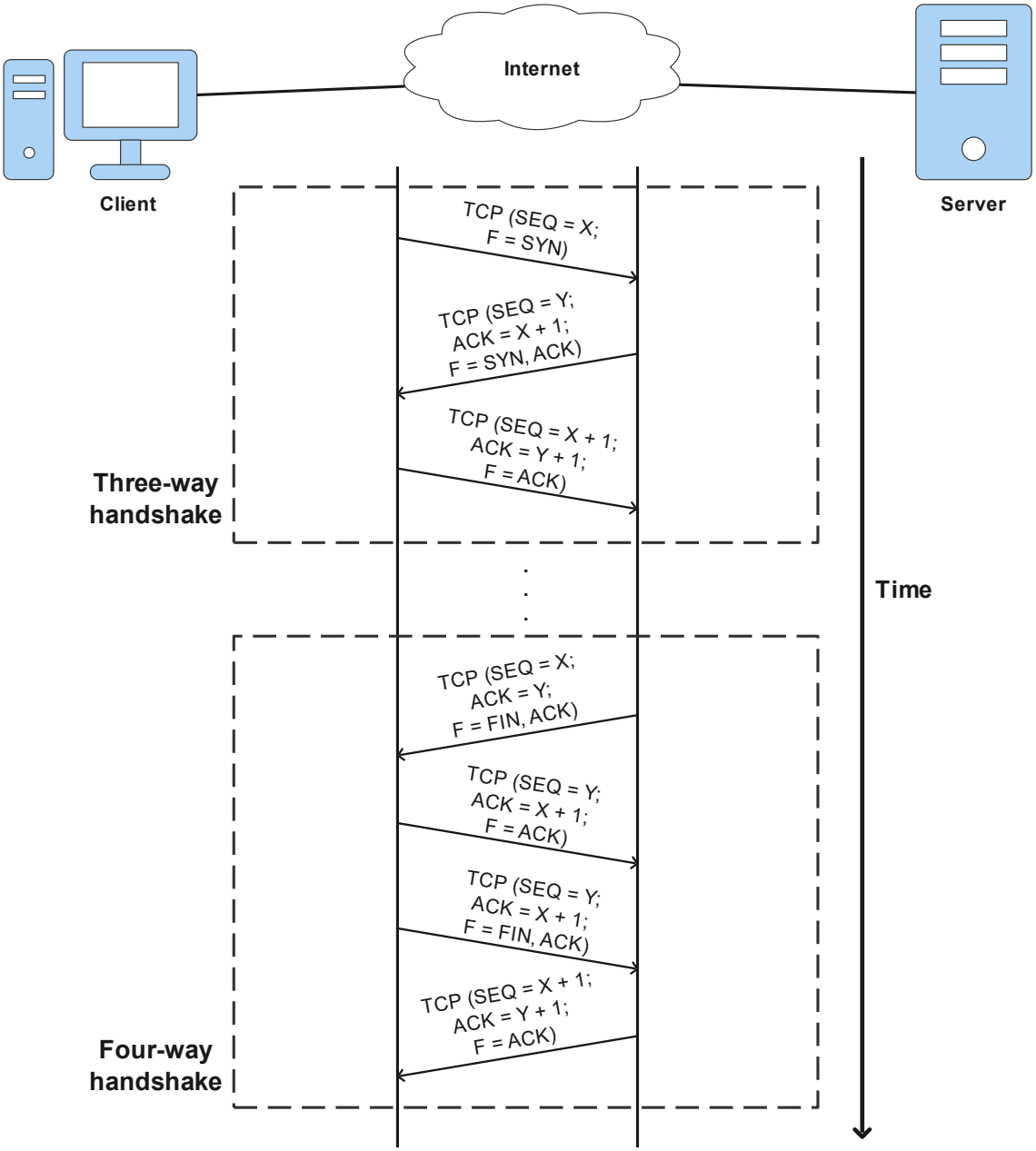


Fig. D.4: TCP communication.

HTTP

HTTP (Hypertext Transfer Protocol) is basically used for the communication with web servers. It is based on the request-response model. If the client wants to visit the page, he types domain name (or IP address) in the web browser, generates request (GET method) and sends it to the server. Server hosting the page sends its content in response (200 OK if everything is correct) to the client [4]. HTTP uses TCP port 80 (but UDP port 80 is also assigned to it). Interesting thing about HTTP is that if there is some error (either on the client or server side), and thus 200 OK

response cannot be generated, the other status code than 200 is generated. It can tell client where exactly did the problem occur. For example 4xx code indicates error on the client side and 5xx indicates error on the server side. For more information you can visit [5].

D.2 Wireshark

D.2.1 Objective 1

1. In the first part of this lab you are going to analyze captured packets in the Wireshark. Open the software and select Ethernet interface. You can stop capturing the packets right now by clicking on the red square (in the **Main toolbar**).
2. Open the **CMD** (Command Prompt). For the purpose of DNS communication, you are going to use the **nslookup**[4] utility. You are going to ask for the IP address of the *www.vut.cz* domain. This would be the standard process if you entered this domain name to your web browser.
3. Start capturing packets in Wireshark (by clicking on the blue fin icon). Now in the CMD, execute `nslookup` command which takes you to the interactive mode. Now enter `set type=A` to query only the IPv4 addresses. Finally, query the IP address for domain by typing `www.vut.cz`. The sequence of commands is displayed in the Fig. D.5.
4. Now let's examine the output. The first two lines indicate the domain name of the local DNS server (`Server: UnKnown`) and its IP address (`Address: 192.168.206.2`). The last two lines display the answer[5]. So in the example, domain is available under the IP address *147.229.2.90*.
5. Switch to the Wireshark window. Enter the following command to the **Filter Toolbar**:

```
dns.qry.name==www.vut.cz.localdomain
```

[4]This is the utility used for querying domain names and receiving its respective IP addresses or vice versa. It has also many other functions, but these will be sufficient for this lab. The communication is based on the standard DNS model query-response. There are 2 modes – interactive and non-interactive. In the interactive mode, you just type `nslookup` command and then you are inside this utility (command prompt changes to >). Here you can type just the parameters, settings etc. without the `nslookup` command at the beginning. In the non-interactive mode, you just enter `nslookup` command followed by the parameter (for example `nslookup www.google.com`).

[5]NOTE: It is possible that your result won't contain the ".localdomain"suffix. If so, don't include the suffix in the following commands.

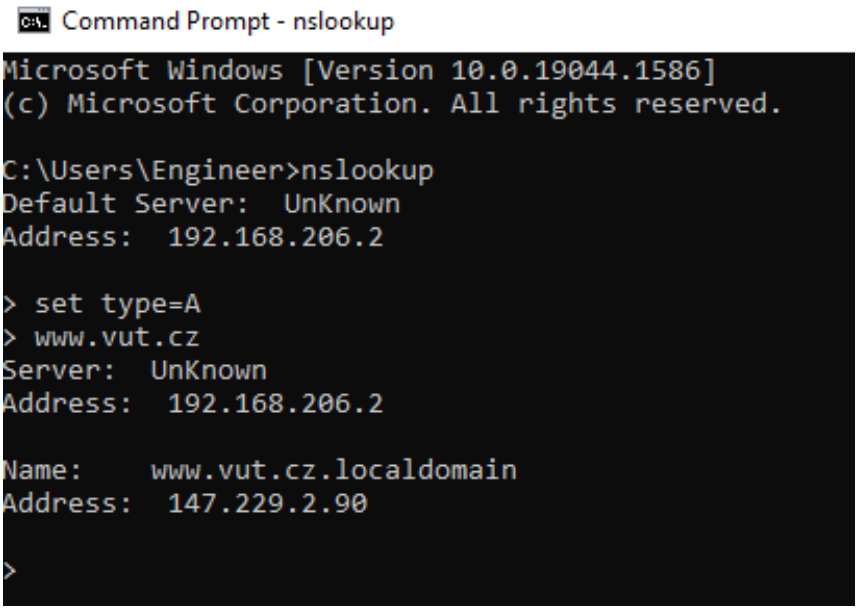


Fig. D.5: Command sequence for DNS query.

This displays only the DNS query-response we issued from all the other DNS communications that were also captured. The output should look like in the Fig. D.6.

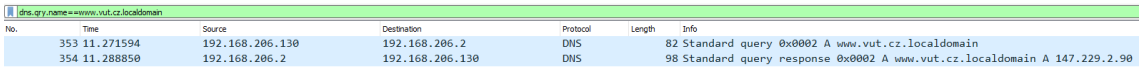


Fig. D.6: Captured DNS query-response via UDP.

6. Let's expand the query message. You can see that UDP is used as the transport layer protocol. Expand User Datagram Protocol section. The output should be similar to the Fig. D.7. The source port is assigned from the dynamic range and the destination port is well-known port assigned for the DNS service. Now look at the length and UDP payload.
   Comparing these two values, does the header size correspond to the introduction information?
7. Now display the response and expand the User Datagram Protocol section.
   How did the source port and destination port change?
   Does the UDP header size still correspond?

D.2.2 Objective 2

1. Now we are going to analyze the same communication but using the TCP as the transport layer protocol. Basically, DNS uses UDP for simple query-

```
User Datagram Protocol, Src Port: 62980, Dst Port: 53
    Source Port: 62980
    Destination Port: 53
    Length: 48
    Checksum: 0x1e18 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 6]
  > [Timestamps]
    UDP payload (40 bytes)
```

Fig. D.7: UDP contents of DNS query.

response communication (low data transmission) and TCP is used for zone transfers (larger amount of data where consistency is required). Using the nslookup tool, you can enforce usage of TCP.

2. Switch to the CMD and type `set vc`. This enforces DNS to establish virtual circuit and therefore use TCP.
3. Issue the query for *www.vut.cz* again. Now move back to the Wireshark. With the filter still applied, you can see that 2 new records appeared (see Fig. D.8)

dns.qry.name==www.vut.cz.localdomain

No.	Time	Source	Destination	Protocol	Length	Info
690	55.222422	192.168.206.130	192.168.206.2	DNS	82	Standard query 0x0002 A www.vut.cz.localdomain
691	55.231905	192.168.206.2	192.168.206.130	DNS	98	Standard query response 0x0002 A www.vut.cz.localdomain A 147.229.2.90
1159	299.409314	192.168.206.130	192.168.206.2	DNS	96	Standard query 0x0003 A www.vut.cz.localdomain
1161	299.419027	192.168.206.2	192.168.206.130	DNS	171	Standard query response 0x0003 No such name A www.vut.cz.localdomain SOA a.root-servers.net

Fig. D.8: Captured DNS query-response via TCP.

4. Right click on one of the newly added messages and then **Follow > TCP Stream**. Close the popped up window and take a look at the conversation flow. Although the same query-response communication was issued, many more packets were transmitted (see Fig. D.9).

tcp.stream eq 19

No.	Time	Source	Destination	Protocol	Length	Info
1154	299.407363	192.168.206.130	192.168.206.2	TCP	66	65230 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1157	299.408587	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1158	299.408654	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1159	299.409314	192.168.206.130	192.168.206.2	DNS	96	Standard query 0x0003 A www.vut.cz.localdomain
1160	299.409547	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [ACK] Seq=1 Ack=43 Win=64240 Len=0
1161	299.419027	192.168.206.2	192.168.206.130	DNS	171	Standard query response 0x0003 No such name A www.vut.cz.localdomain SOA a.root-servers.net
1162	299.419632	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [FIN, ACK] Seq=43 Ack=118 Win=64123 Len=0
1163	299.419738	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [ACK] Seq=118 Ack=44 Win=64239 Len=0
1165	299.420290	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [FIN, PSH, ACK] Seq=118 Ack=44 Win=64239 Len=0
1168	299.420653	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [ACK] Seq=44 Ack=119 Win=64123 Len=0

Fig. D.9: Captured TCP stream using DNS.

5. At first, three TCP messages were exchanged to establish virtual circuit. Click through all the messages and examine the contents inside Transmission Control Protocol section. Source and destination port numbers principle remains the same. For further analysis, the important items will be **Sequence number**, **Acknowledgment number** and **Flags**.

6. First message generates Sequence number 0[6] and sets the SYN flag. No acknowledgment is set as no data are being acknowledged (see Fig. D.10).

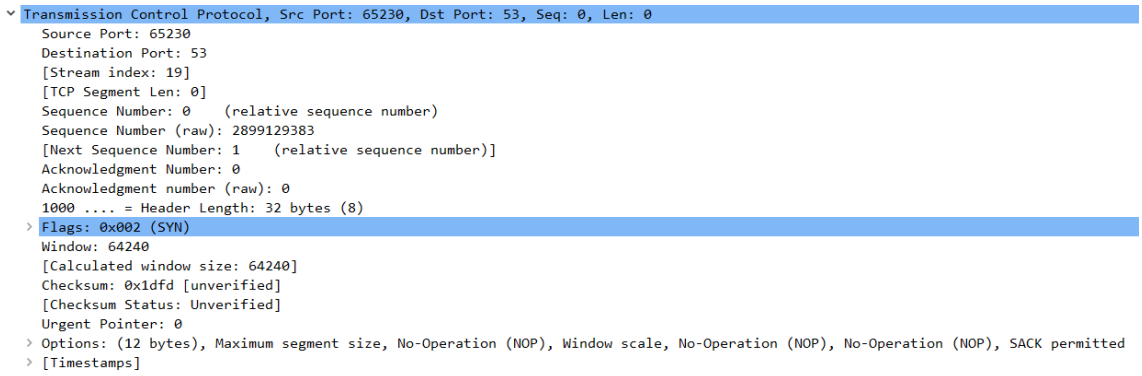


Fig. D.10: First TCP message.

Look at the Header Length and Options rows. Excluding optional parameters, how many bytes does the TCP header take? Does it correspond to the introduction?

7. Compare the remaining two messages with the three-way handshake information in the introduction.
8. The data themselves are transmitted starting with the DNS query. Examine the TCP section inside packet.
   Compare the Sequence Number and Acknowledgment Number with the values in previous neighboring packet. Do they differ? Why?
9. DNS server acknowledges the incoming query.
   What Acknowledgment Number does he set? What does it mean?
10. Server then answers with the query. The following 4 packets represent the four-way handshake to close the connection. Examine the TCP section in the first packet sent by the client.
    What flags are set?
11. Compare the four-way handshake process with the theory in introduction.
12. In the CMD, set nslookup behavior to the default state (using UDP for simple transmission) by typing `set novc`.

[6] You can see that both communicating ends generated Sequence number 0 in the Info column. That is just the relative number that Wireshark displays for clarity. Real sequence numbers are randomly generated numbers and their real value can be seen in the **Sequence number (raw)** row.

D.2.3 Objective 3

1. Now let's compare the two transmissions in graphs. Stop the capturing by clicking on the red square.
2. Use the `dns.qry.name==www.vut.cz.localdomain` filter again to display the DNS conversations. First click on one of the packets that used UDP as the transport protocol. Right click on it and then select **Follow > UDP Stream**. Click **File > Export Specified Packets...** and save it to your desktop under the "DNS_UDP"name.
3. Now export the TCP stream the same way as UDP. So use the initial filter again, follow the TCP Stream of DNS packets using TCP as the transport layer protocol and save it to your desktop under the "DNS_TCP"name.
4. Now select **File > Open** and select the DNS_UDP file from your desktop. Now click **File > Merge...** and select the DNS_TCP file. You will compare the two transmissions, but for that it would be sufficient if they started at the same time, but TCP is shifted in time. For this purpose, select the **first** packet of TCP conversation, right click on it and select **Set/Unset Time Reference**. Now the time value changes to the ***REF*** and following packets are using the relative time to the first packet (see Fig. D.11).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.206.130	192.168.206.2	DNS	82	Standard query 0x0002 A www.vut.cz.localdomain
2	0.009483	192.168.206.2	192.168.206.130	DNS	98	Standard query response 0x0002 A www.vut.cz.localdomain A 147.229.2.90
3	*REF*	192.168.206.130	192.168.206.2	TCP	66	65230 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.001224	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
5	0.001291	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	0.001951	192.168.206.130	192.168.206.2	DNS	96	Standard query 0x0003 A www.vut.cz.localdomain
7	0.002184	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [ACK] Seq=1 Ack=43 Win=64240 Len=0
8	0.011664	192.168.206.2	192.168.206.130	DNS	171	Standard query response 0x0003 No such name A www.vut.cz.localdomain SOA a.root-servers.net
9	0.012269	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [FIN, ACK] Seq=43 Ack=118 Win=64123 Len=0
10	0.012375	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [ACK] Seq=118 Ack=44 Win=64239 Len=0
11	0.012927	192.168.206.2	192.168.206.130	TCP	60	53 → 65230 [FIN, PSH, ACK] Seq=118 Ack=44 Win=64239 Len=0
12	0.013290	192.168.206.130	192.168.206.2	TCP	54	65230 → 53 [ACK] Seq=44 Ack=119 Win=64123 Len=0

Fig. D.11: TCP and UDP conversations merged.

5. Click **Statistics > I/O Graphs**. In the newly popped up window, delete all the default graphs (the [–] icon). Add 2 new graphs by clicking on the [+] icon. Name the first graph as "UDP transmission"and the second one as "TCP transmission".
6. In the **Display Filter** column, set `udp` for the UDP transmission and `tcp` for the TCP transmission. Change colors at your discretion and set **Y Axis** to **Packets**. Change **Interval** to **1 ms** and tick **Enabled**[7]. The output should look like in the Fig. D.12. Data for the graph are exported to the Tab. D.1 for UDP transmission and to the Tab. D.2 for the TCP transmission.
7. More packets were used for the complete transmission in case of TCP than with UDP. This was also apparent when you displayed the whole streams.

[7]There still applies the rule that if you have scattered graphs, click Reset to automatically set scale.

Tab. D.1: Table of packets sent during the UDP transmission.

Seconds	0	1	2	3	4	5	6	7	8	9
Packets	1	0	0	0	0	0	0	0	0	1

Tab. D.2: Table of packets sent during the TCP transmission.

Seconds	0	1	2	3	4	5	6	7	8	9
Packets	1	3	1	0	0	0	0	1	3	1

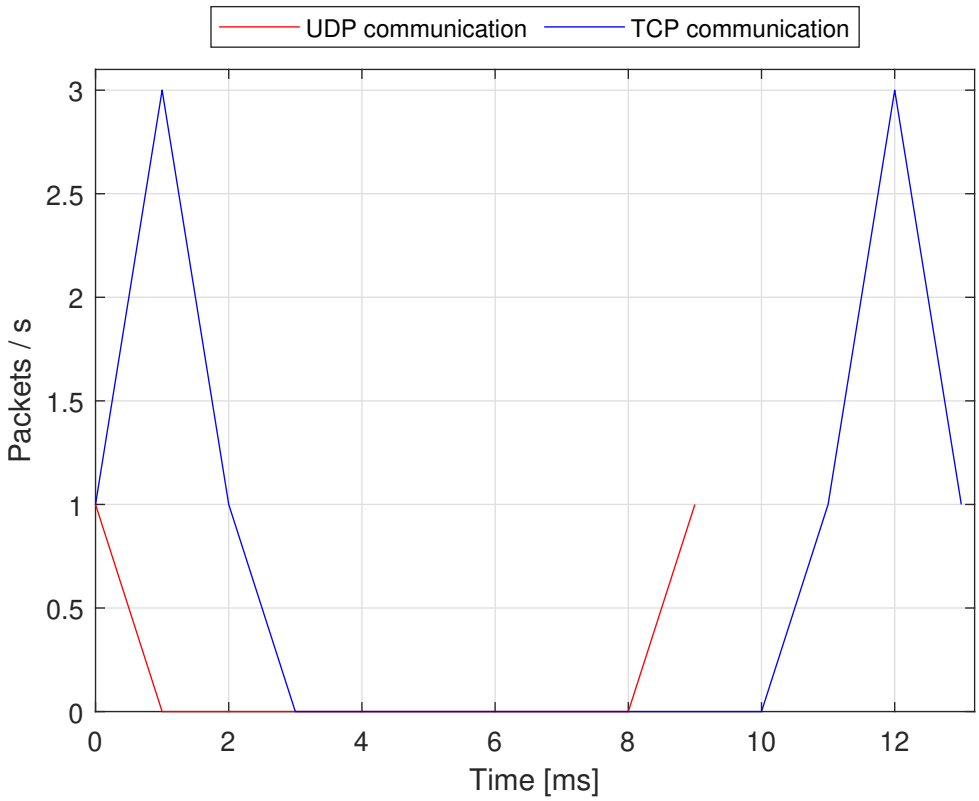


Fig. D.12: Packets sent during DNS transmissions using UDP and TCP.

Now let's compare the two protocols in terms of transmitted bytes. Change the **Y axis** to **Bytes**. The output is shown in the Fig. D.13. Data for the graph are exported to the Tab. D.3 for UDP transmission and to the Tab. D.4 for the TCP transmission.

8. You can see that in comparison with UDP, TCP transferred many more bytes to reach the same result.
   In terms of time, which protocol was faster in full query-response transfer?

Tab. D.3: Table of bytes sent during the UDP transmission.

Seconds	0	1	2	3	4	5	6	7	8	9
Bytes	82	0	0	0	0	0	0	0	0	98

Tab. D.4: Table of bytes sent during the TCP transmission.

Seconds	0	1	2	3	4	5	6	7	8	9
Bytes	66	210	60	0	0	0	0	171	174	54

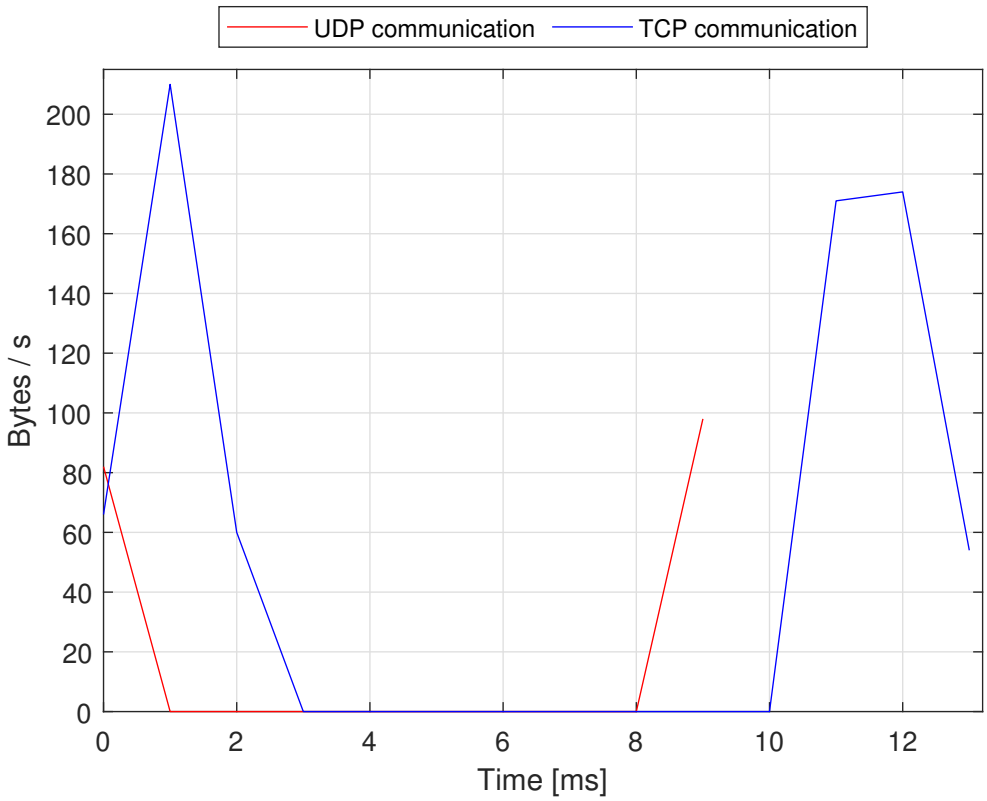


Fig. D.13: Bytes sent during DNS transmissions using UDP and TCP.

D.2.4 Objective 4

1. Now you will generate the HTTP traffic to see more complex TCP transmissions. Start capturing again. For the purpose of HTTP you will visit the *http://www.neverssl.com* website. Open your browser, visit the web page and after all the content is loaded, close the browser and stop capturing.
2. Put `http` inside the filter. GET method is displayed with its respective answer.

Right click on any of the packets and select **Follow > TCP Stream**. In the newly popped up window there is apparent standard HTML structure and everything can be read[8]. Close the window and look at the TCP packets.

3. At the beginning, you can see standard virtual circuit establishment and then HTTP data exchanged in TCP segments.
4. If there was some data loss[9] during the transmission, it could look like in the Fig. D.14. The black highlighted rows mean there were some errors during the transmission. The **TCP Previous segment not captured** message tells us that a segment with higher sequence number was received than it was expected. Look at the Fig. D.15. Compare its Sequence Number with Ack number from the packet No. 3 (in the Info column). The next sequence number expected from client is 1, but the received sequence number is 1478, which means that 1477 bytes were lost during the transmission.
   You can see that there was already one packet sent with the sequence number 1 (packet No. 5) from the server. Why is sequence number 1 still expected from the client? (Hint: Look at the last item in the Info column of packet No. 5.)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.206.130	65.9.96.54	TCP	66	49836 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.005701	65.9.96.54	192.168.206.130	TCP	60	80 → 49836 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.005739	192.168.206.130	65.9.96.54	TCP	54	49836 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.030098	192.168.206.130	65.9.96.54	HTTP	555	GET /online HTTP/1.1
5	0.030323	65.9.96.54	192.168.206.130	TCP	60	80 → 49836 [ACK] Seq=1 Ack=502 Win=64240 Len=0
6	0.038092	65.9.96.54	192.168.206.130	HTTP	1376	[TCP Previous segment not captured] Continuation
7	0.038151	192.168.206.130	65.9.96.54	TCP	54	[TCP Dup ACK 3#1] 49836 → 80 [ACK] Seq=502 Ack=1 Win=64240 Len=0
8	0.140320	65.9.96.54	192.168.206.130	TCP	1514	[TCP Retransmission] 80 → 49836 [ACK] Seq=1 Ack=502 Win=64240 Len=1460
9	0.140320	65.9.96.54	192.168.206.130	TCP	1393	[TCP Retransmission] 80 → 49836 [PSH, ACK] Seq=1461 Ack=502 Win=64240 Len=1339
10	0.140425	192.168.206.130	65.9.96.54	TCP	54	49836 → 80 [ACK] Seq=502 Ack=2800 Win=64240 Len=0
11	0.194479	192.168.206.130	65.9.96.54	HTTP	483	GET /favicon.ico HTTP/1.1
12	0.194716	65.9.96.54	192.168.206.130	TCP	60	80 → 49836 [ACK] Seq=2800 Ack=931 Win=64240 Len=0
13	0.201109	65.9.96.54	192.168.206.130	TCP	554	80 → 49836 [PSH, ACK] Seq=2800 Ack=931 Win=64240 Len=500 [TCP segment of a reassembled PDU]
14	0.201353	65.9.96.54	192.168.206.130	HTTP	178	HTTP/1.1 200 OK (PNG)
15	0.201389	192.168.206.130	65.9.96.54	TCP	54	49836 → 80 [ACK] Seq=931 Ack=3424 Win=63616 Len=0

Fig. D.14: Packet loss during the TCP transmission.

5. Client therefore sends **TCP Dup ACK**. It is called duplicate ACK because it was already sent once. In our case, it is the duplicate of packet No. 3. Client is telling that he is still expecting sequence number 1 from the server.
6. When the server receives this message, it is the indicator that there was some packet loss during the transmission. Therefore, he sends data he already sent again (messages marked as **TCP Retransmission**). The two segments are transmitted, one with sequence number 1 and the second with the sequence number:

$$1 + [\text{length of previous segment}]$$

[8]That is because HTTP does not use encryption of data and whoever captures the traffic can read it. So if you would use other methods (like POST) and send data like password to the webpage form, you would take a risk that your sensitive data could leak out. This security risk was removed with the invention of HTTPS that encrypts the data.

[9]NOTE: It is possible that you did not experience any data loss. If that is so, just settle with the described example.

```
Transmission Control Protocol, Src Port: 80, Dst Port: 49836, Seq: 1478, Ack: 502, Len: 1322
    Source Port: 80
    Destination Port: 49836
    [Stream index: 0]
    [TCP Segment Len: 1322]
    Sequence Number: 1478    (relative sequence number)
    Sequence Number (raw): 1386302646
    [Next Sequence Number: 2800    (relative sequence number)]
    Acknowledgment Number: 502    (relative ack number)
    Acknowledgment number (raw): 2157239626
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x8c73 [unverified]
```

Fig. D.15: "TCP Previous segment not captured"segment portion.

7. The following messages were transmitted without any packet loss. You have just seen the power of TCP. If there are some troubles during the transmission, TCP takes care of reliable delivery of all the data to its destination (more specifically it will do its best for it). If packet loss occurred during the UDP transmission, it would not resend the data as in the case of TCP, because it does not have any reliability mechanisms (except for checksum that indicates errors in the received data). If you look at the Fig. D.16, with no reliable mechanisms, it's up to the application to handle the packet loss. In this case, there was timer set and as there was no response received, the application sent query again and then the response was received.

No.	Time	Source	Destination	Protocol	Length	Info
13	2.988537	192.168.206.130	192.168.206.2	DNS	70	Standard query 0x0003 A www.vut.cz
18	4.989581	192.168.206.130	192.168.206.2	DNS	70	Standard query 0x0004 A www.vut.cz
19	4.994613	192.168.206.2	192.168.206.130	DNS	86	Standard query response 0x0004 A www.vut.cz A 147.229.2.90

Fig. D.16: DNS repeated query with UDP used.

D.3 Packet Tracer

D.3.1 Objective 5

1. Let's move to the Packet Tracer. Open the saved topology from the previous laboratory exercise and save it under a new name.
2. In the **End Devices** group, select the **Server-PT** device and add it to the topology under the R2 router (see Fig. D.1). Rename the device to Server and connect it with R2 (port **FastEthernet0/1**) using the **Copper Straight-Through** cable.
3. The server subnet has the address space *172.20.3.0/24* assigned (see Fig. D.17). Configure **the first host address** available on the server and **the last host address** on the router. The configuration of the server is the same as on the PCs, thus **click on the server > Desktop > IP Configuration** and fill in the IPv4 Address, Subnet Mask and Default Gateway. Configure the router interface[10].

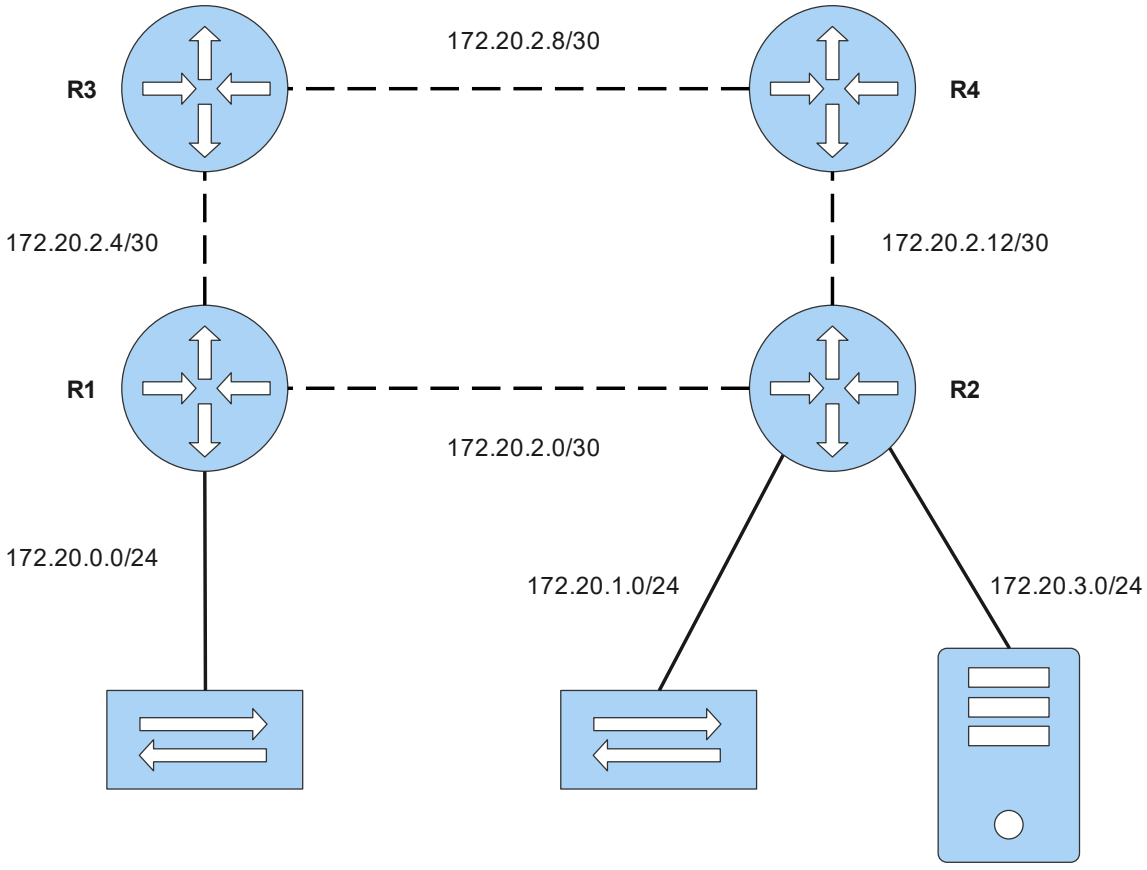


Fig. D.17: Lab 4 addressing.

[10]NOTE: Don't forget to enable the interface.

4. Explore the R1's routing table. Your output should look like in the Fig. D.18). You can see that the newly added server subnet is already present without any further configurations.
   Can you determine why? Hint: Look at the code of the path and recall the difference in configurations of the two routing protocols you were configuring in the previous lab. Also recall the network class principle of RIP.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/16 is variably subnetted, 10 subnets, 3 masks
C       172.20.0.0/24 is directly connected, FastEthernet0/0
L       172.20.0.254/32 is directly connected, FastEthernet0/0
O       172.20.1.0/24 [110/193] via 172.20.2.6, 00:03:46, Serial0/3/1
C       172.20.2.0/30 is directly connected, Serial0/3/0
L       172.20.2.1/32 is directly connected, Serial0/3/0
C       172.20.2.4/30 is directly connected, Serial0/3/1
L       172.20.2.5/32 is directly connected, Serial0/3/1
O       172.20.2.8/30 [110/128] via 172.20.2.6, 00:04:19, Serial0/3/1
O       172.20.2.12/30 [110/192] via 172.20.2.6, 00:03:56, Serial0/3/1
R       172.20.3.0/24 [120/1] via 172.20.2.2, 00:00:30, Serial0/3/0
```

Fig. D.18: R1's routing table with RIP path.

5. As we want to have routes learnt only via OSPF in the routing tables, let's add the *172.20.3.0/24* subnet to the OSPF routing process. If you don't remember the command sequence use the previous laboratory manual to repeat.
6. Display the R1's routing table again and verify that the OSPF record is present instead of the RIP record.
7. Use `ping` command to test the reachability of server to the R1's LAN.
8. Save your current progress by clicking on the **File** > **Save**. Don't forget to save your progress after completing individual objectives!

D.3.2 Objective 6

1. If the connectivity is functional, let's move to the configuration of services on the server. You will make it the DNS server as well as the web server for the R1's LAN. In order to do this, the services must be enabled.
2. Click on the server and open the **Services** tab. You will configure the web service first. As you already know from the introduction, web traffic uses HTTP

protocol (or HTTPS) for communication, so select the **HTTP** from the services column. The basic setup is visible on the Fig. D.19. Enable the HTTP protocol and disable the HTTPS protocol (by ticking the **On** option for the HTTP and **Off** for the HTTPS). You can see in the File Manager section that there are many files used for the basic web page. The only important file for us is the **index.html**, so preserve this one and delete all other files.

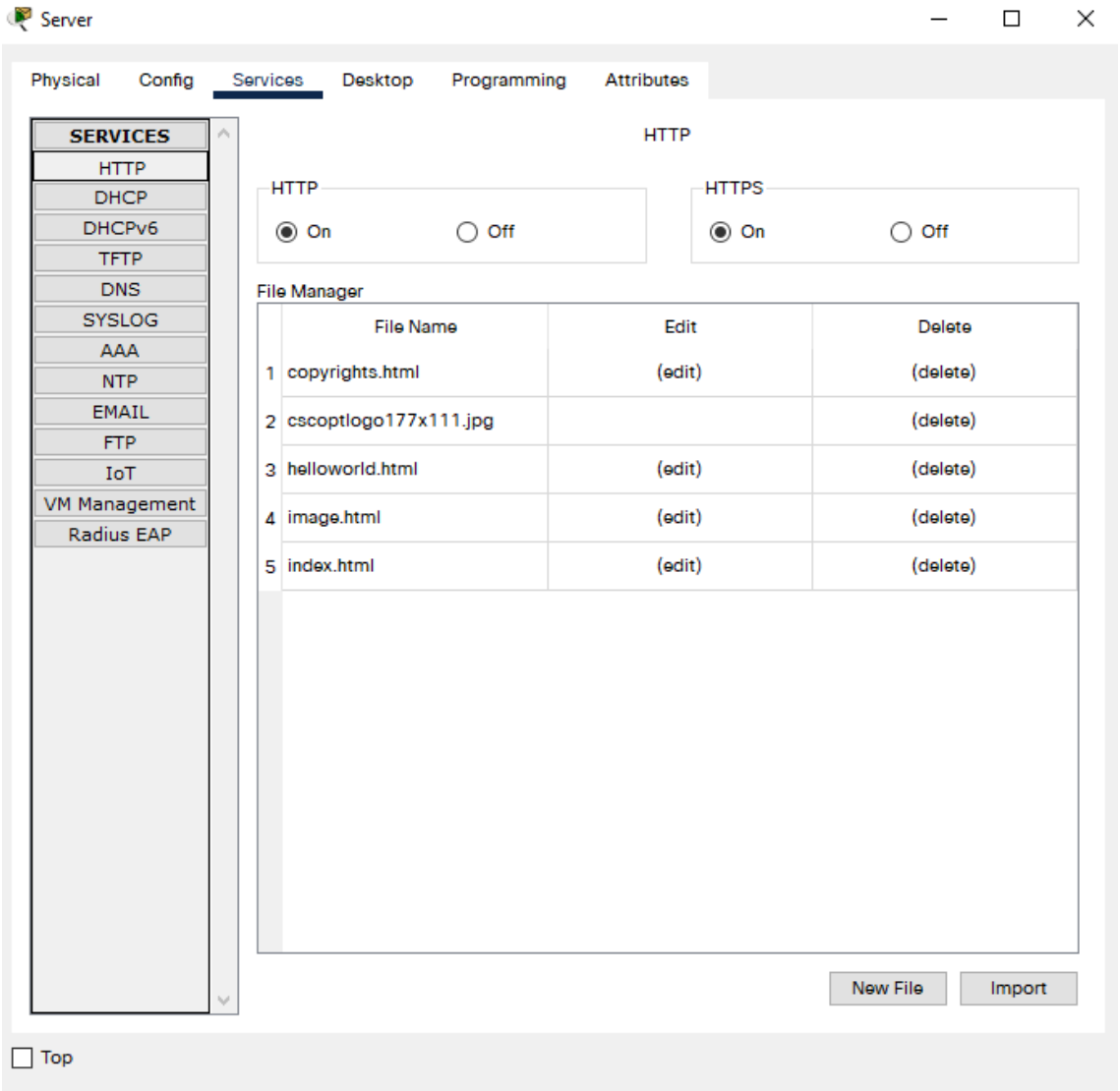


Fig. D.19: Basic server web configuration.

3. Edit the index.html file. There is some HTML code already generated, but you will use simple custome one. You can use the one below:

```
<html>
<center><h1 style="color:red">My custom web
page</h1></center>
```

```
        <hr>
        <p>Good job, I successfully accessed the web page!</p>
        </html>
```

Save it and move to the next point.

4. Next configure the DNS service. First turn it on by ticking the **On** option. Next you will add the record consisting of the domain name and the IP address. Add `www.mywebpage.com` to the **Name** form and IPv4 address of the server to the **Address** form. The **Type** should be set to **A Record**. Finally click on the **Add** button. The result is displayed in the Fig. D.20.

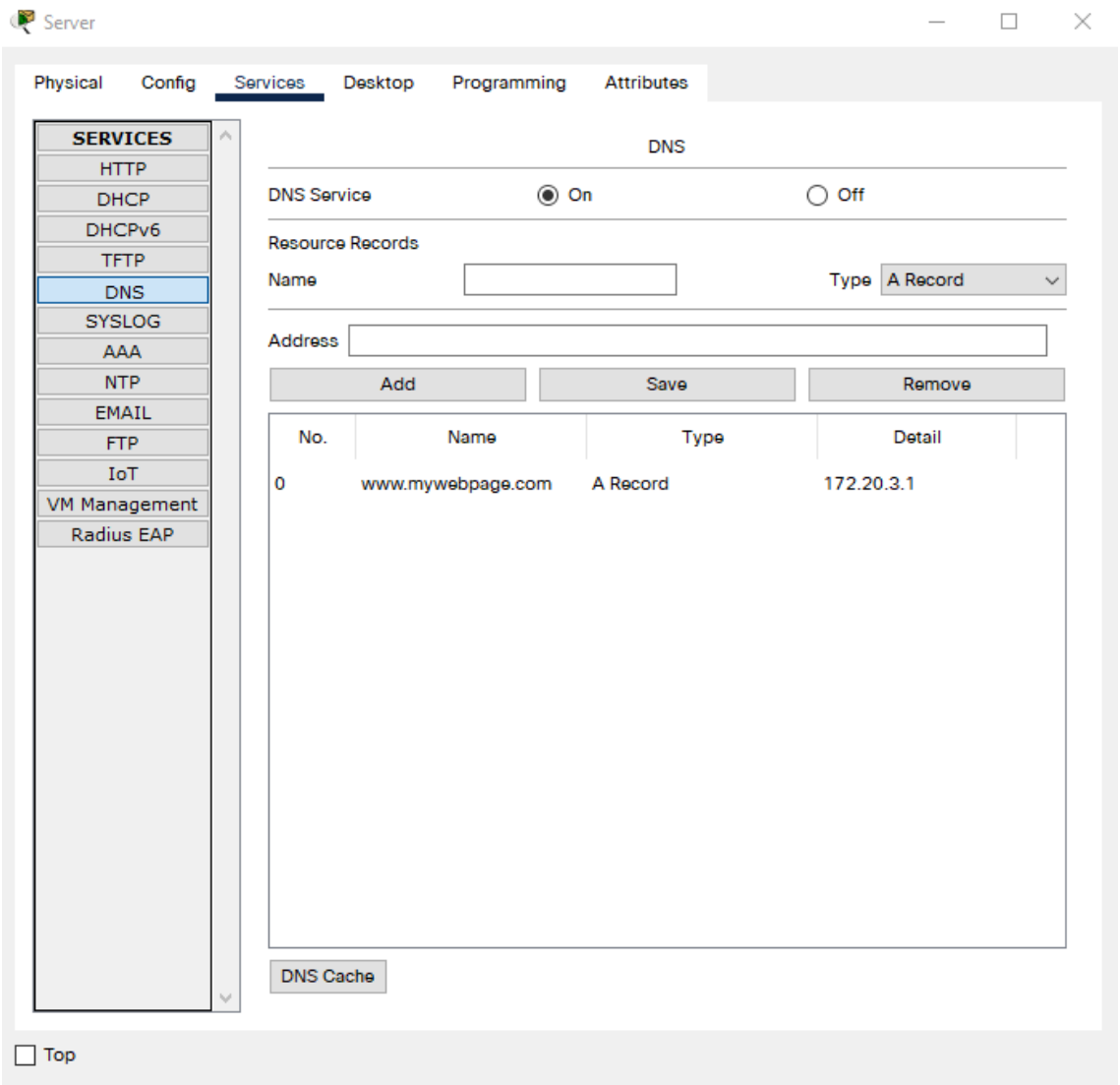


Fig. D.20: Server DNS configuration.

D.3.3 Objective 7

1. Now that you have the server fully prepared for the access, you have to set the DNS server address on the clients so that they know which server to use for the domain name translations.
2. Set up the **DNS Server** address on PCs of R1's LAN[11]. You can find the configuration in the same tab where you configured the end device addresses.

D.3.4 Objective 8

1. Let's generate the communication by accessing the web page. For the purpose of packet analysis, switch to the **Simulation** mode.
2. Click the **Show All/None** button to clear the filter. Next select the **Edit Filters**, list through the tabs and choose the following protocols for the capturing:
   - UDP,
   - TCP,
   - DNS,
   - HTTP.

   Now you can close the tab.
3. Select any PC from the R1's LAN and open the **Web Browser** from the **Desktop** tab. Enter the `www.mywebpage.com` address to the URL form and click **Go**. **Don't close the PC window during the whole communication!**
4. You can see that the DNS datagram appeared in the **Event List**. Open the datagram and analyse its content.
   What is the source port?
   What is the destination port? What is the destination IP address?
5. Click **Capture then forward** until the datagram reaches the server. During the transmission, you can observe the path the datagram is taking to reach its destination (according to the routing table records).
6. Open the packet. You can see that the **Inbound** and **Outbound PDU Details** are present. Explore the outbound section.
   How did the source and destination port change?
   Look at the DNS Answer. What is the IP address the server is responding for the queried domain?
7. Click **Capture then forward** until the datagram reaches the PC. Now there is new TCP segment generated. Open it and explore its content.

[11] You can also set the R2's LAN, but it is not necessary for the future purposes.

What is the source port?
What is the destination port?
What is the sequence number?
What is the acknowledgment number?
How many flags are set (the bits with the value of 1)? With the knowledge of TCP communication, what flags do these bits represent?

8. Click **Capture then forward** until the segment reaches the server.
   How did the source and destination port change? What is the sequence number?
   What is the acknowledgment number?
   How many flags are now set? What flags are used?
9. You can now click **Capture then forward** so that the TCP segment is received by the PC. As you probably guessed correctly, the three-way handshake communication just occurred between the devices. Two segments were already exchanged, the third one is needed. Also the HTTP message appears. Explore the contents of TCP segment and HTTP message and compare their values.
   What is the sequence number?
   What is the acknowledgment number?
   What kind of HTTP message is transferred?
10. Click **Capture then forward** until the segment reaches the server. Both the TCP segment finishing the three-way handshake and HTTP message are now transferred. Server receives the segment finishing the virtual circuit establishment before the HTTP request arrives, so the request can be processed and answer sent correctly.
11. Click **Capture then forward** once more so the HTTP message reaches the server. Analyze the contents and compare the inbound and outbound PDU details as in the previous steps.
    What kind of HTTP message is now transferred to the client?
12. Click **Capture then forward** until the HTTP message reaches the client. New TCP segment is now generated.
    Did the web page displayed correctly? Compare the sequence and acknowledgment numbers of the incoming HTTP message and outgoing TCP segment. How did they change?
    How many flags are sent in the outgoing TCP segment? What kind of flags probably is that? Hint: The web page was successfully transferred and there is no other HTTP data to be sent.
13. Click **Capture then forward** until the segment reaches the server. Compare the sequence numbers, acknowledgment numbers and the number of flags inside incoming and outgoing segments. You can probably derive that this is the

faster alternative of four-way handshake in closing the virtual circuit.

14. Click **Capture then forward** until the segment reaches the client.
    How many flags are set in the outgoing segment?

    Obviously this is the last segment to close the connection. Verify this by clicking **Capture then forward** until the segment reaches the server.
    Is there another segment generated for the client?
15. One last note in conclusion. If you look back at the communication and write out the flag bits in order as they are shown, you should come to the conclusion that this is the order of flags[12]:

    ```
    0b X X X ACK PSH X SYN FIN
    ```

16. Save the current progress for a future lab.

D.4 Final questions

1. What items does the UDP header contain?
2. What happens when the packet is lost during the UDP communication in comparison with TCP?
3. What precedes and follows the data exchange via TCP protocol?
4. How is the TCP connection (virtual circuit) establishment called?
5. What group of ports do the servers use for common services?

[12]The PSH flag was not explained, but basically it tells the transport layer to send the data to the application without expecting additional data.

Literature

[1] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768. In: *RFC Editor* [online]. 1980 [cit. 08. 04. 2022]. Available at: <https://www.rfc-editor.org/info/rfc768>.

[2] COOK, Matt. TCP vs. UDP: What-s the Difference?. In: *Lifesize* [online]. 2017 [cit. 08. 04. 2022]. Available at: <https://www.lifesize.com/en/blog/tcp-vs-udp/>.

[3] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793. In: *RFC Editor* [online]. 1981 [cit. 08. 04. 2022]. Available at: <https://www.rfc-editor.org/info/rfc793>.

[4] FOX, Pamela. Hypertext Transfer Protocol (HTTP). In: *Khan Academy* [online]. 2022 [cit. 08. 04. 2022]. Available at: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:web-protocols/a/hypertext-transfer-protocol-http>.

[5] MDN contributors. HTTP response status codes. In: *MDN Web Docs* [online]. 2022 [cit. 08. 04. 2022]. Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>.

E Lab 5 – DHCP

In this laboratory you are going to analyze DHCP protocol and configure various scenarios to analyze its functionality.

Objectives

1. Release and renew IP configuration via DHCP and capture communication with DHCP server in Wireshark.
2. Analyze DHCP packet content during communication states.
3. Generate graphs from the captured communication.
4. Create the reference topology in Packet Tracer (see Fig. E.1).
5. Configure R2 as the DHCP server for both connected LANs. Set PCs to use DHCP and analyze communication.
6. Enable DHCP service on the LAN server and configure R2 as the DHCP relay agent. Release and renew IP configuration via DHCP and explore the relay agent's role.

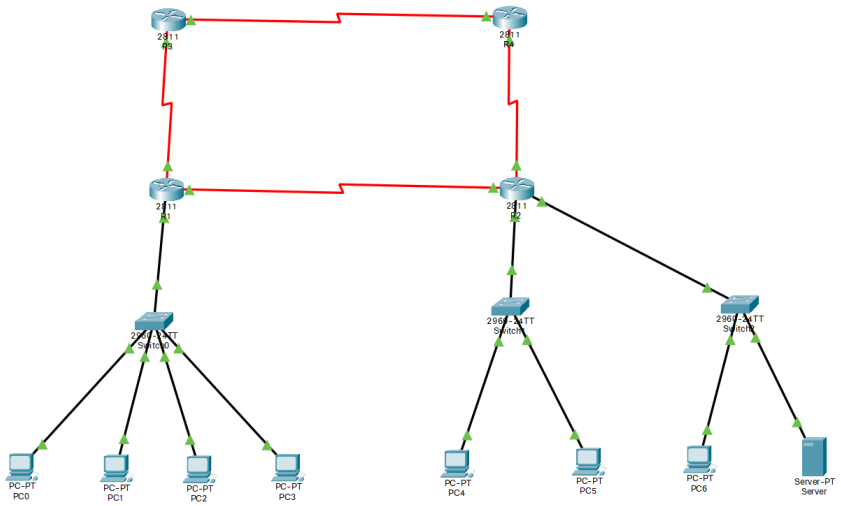


Fig. E.1: Lab 5 reference topology.

E.1 Introduction

DHCP (Dynamic Host Configuration Protocol) is protocol used for providing IP configuration to the clients without the intervention of administrator (except for initial configuration of DHCP server). In the early days of internet, every client had to be configured manually. This was not a big deal as there were only a few clients in local networks. With the internet growing, manual configuration started to be challenging as the number of clients started to grow enormously and administrators had to keep records of each pair of client device and its assigned IP address.

Protocols like RARP and BOOTP were invented, but they had certain limitations (for example statically defined IP-MAC records). New protocol, DHCP, defined in the RFC 2131 [1], is successor to the BOOTP.

DHCP servers assign IP addresses to its clients dynamically and thus no permanent records are created (although options for making static records from dynamic exist). Addresses are assigned from the **pool**, where administrators define the address range for clients[1]. Commonly, clients obtain the addresses and are allowed to use them for a certain time called **lease time**. This is the parameter configured by administrators during the DHCP server setup. After this time expires, clients must stop using the address and release it back to the pool. Such situation would force clients to immediately break connections to the internet and reconnet after new address is assigned (after the standard DHCP communication process). Other method is used. Clients in relatively defined time (before the expiration), which is half the lease time [2], contact DHCP server and request lease time extension. This leads to the continuous address usage and no break moments are present.

Two scenarios of DHCP server placement may exist in the networks. Either server is placed in the same subnet as the client or server is placed in the different network. In the same subnet scenario, standard DHCP communication occurs. In the different subnet scenario, some extra part has to be added to allow communication outside the local subnet as broadcast communication mechanisms are used. This part is called **relay agent**. Relay agents are commonly routers configured to this role. Their task is to forward DHCP messages to the appropriate subnets (where client and server are located) with some extra information added[2].

[1] Other parameters like subnet mask, default gateway address, DNS server and others are also defined.

[2] These information are used to uniquely identify client in the subnet as server may receive many requests from different clients scattered across many subnets.

DHCP message structure

DHCP uses UDP protocol for data delivery. Client communicates via port 68 and server via port 67. The DHCP structure contains many items (see Fig. E.2). The items are divided into two categories – fixed and optional [3].

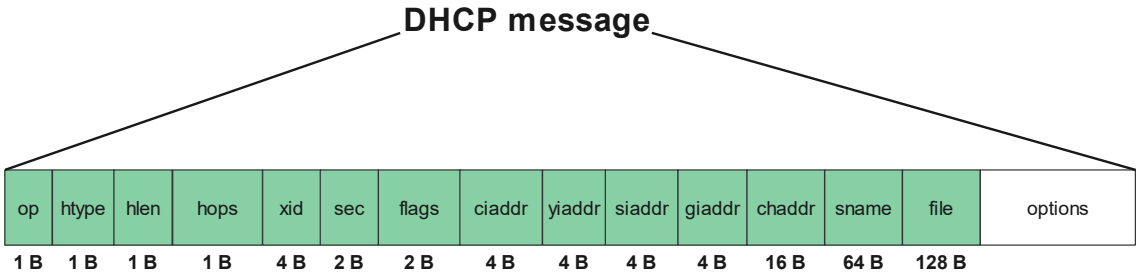


Fig. E.2: DHCP message structure.

Only some of the parameters are explained for this lab's purpose and their respective Wireshark field name is given in the parentheses:

- **op** (*Message type*) – The operational code is used to identify the message type. Possible values for this parameter are:
  - **1** – Request message (sent from client).
  - **2** – Reply message (sent from server).
- **htype** (*Hardware type*) – Type of hardware address (similar to the ARP HW type). Value 1 is used for the Ethernet.
- **hlen** (*Hardware address length*) – Length of the hardware address.
- **hops** (*Hops*) – Number of relay agents the message travelled through before reaching the server.
- **xid** (*Transaction ID*) – Random number generated by the client that uniquely identifies the transaction between client and server.
- **secs** (*Seconds elapsed*) – Time measured in seconds that has passed since the client request was sent.
- **ciaddr** (*Client IP address*) – The client's IP address set by the client. During the initial communication for IP address assignment, the address is set to *0.0.0.0*. During the lease time extension, client sets its assigned IP address.
- **yiaddr** (*Your (client) IP address*) – The client address set by a server to inform client which address from the pool it may get (or already got) assigned.
- **giaddr** (*Relay agent IP address*) – The gateway IP address set to *0.0.0.0* by default (by a client). When a router is set to the relay agent's role, it uses its local interface address (the interface connected to the subnet where client is located) to determine, which pool should DHCP server use to assign the address.

- **chaddr** (*Client MAC address*) – The hardware address of the client.
- **options** – The optional parameters used during the communication. Mainly the items specifying the configuration for a client and identification of a server are used. Each option is defined by the code, length and value. Some of them are listed below:
  - **Subnet mask** – Defines the subnet mask client should use for correct configuration. The code is 1 and the length of 4 bytes.
  - **Router** – Used for the default gateway address. The code is 3 and the length of 4 bytes.
  - **Domain name server** – Here the DNS server address is set. The code is 6 and the length of 4 bytes.
  - **Requested IP address** – Used by the client when more than 1 offer is received[3]. Client selects IP address and informs servers about his choice. The code is 50 and the length of 4 bytes.
  - **IP address lease time** – Lease time configured on the server. The code is 51 and the length of 4 bytes.
  - **DHCP message type** – This item is used for the message type identification. The code is 53 and the length of 1 byte. Common values are *Discover, Offer, Request, Reply.*
  - **DHCP server identifier** – DHCP server sets its own IP address for the following communication. The code is 54 and the length of 4 bytes.

DHCP communication process

As mentioned earlier, broadcast mechanisms are used to deliver messages during the initial address assignment process. Basically, 4 messages are transferred before the client is allowed to use IP configuration. The whole communication consists of – DHCP Discover, DHCP Offer, DHCP Request, DHCP ACK. For the purpose of lease time extension, the unicast communication occurs. Only the DHCP Request and DHCP ACK messages are exchanged between client and server. The messages are shown in the Fig. E.3. Communication process proceeds as follows:

1. Client sends out discovery message to contact all the possible DHCP servers about his interest to obtain IP configuration.
2. Servers send their offer from the appropriate address pool.
3. The client chooses one of the received offers and informs all the servers about his choice.
4. The server, that was chosen by the client, reserves IP address for the client and sends out acknowledgment to confirm the choice.

[3]The case where more than 1 DHCP server is present in the network.

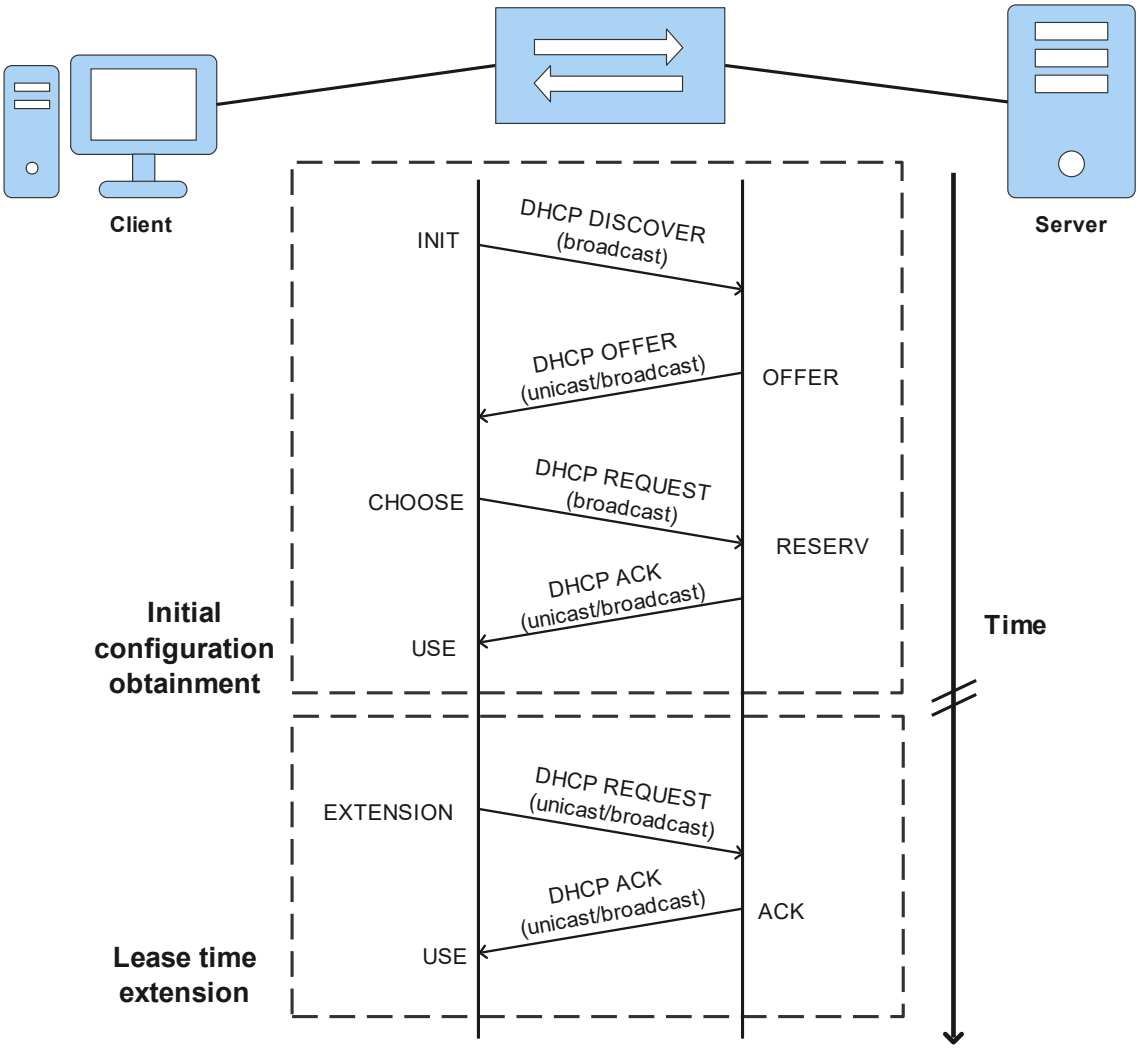


Fig. E.3: DHCP communication.

5. After certain time, client contacts the server to extend the lease time so he can continue using the address.
6. Server acknowledges the request (if it's possible) and extends the lease time for the client.

E.2 Wireshark

E.2.1 Objective 1

1. First of all, let's confirm that the way the IP address is assigned to your PC is via DHCP. From the Windows desktop, click `Start > Settings > Network & Internet > Change adapter options`. You should see available interfaces as shown in the Fig. E.4.

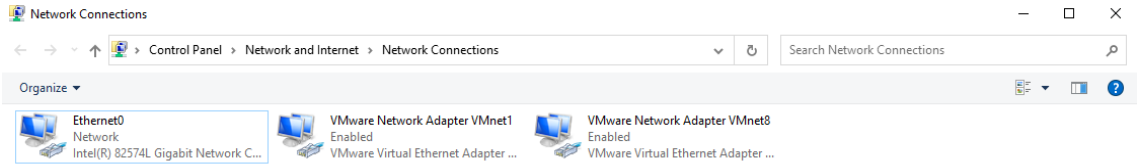


Fig. E.4: Windows interfaces.

   Right click on the appropriate Ethernet interface and select `Properties`. From the list of items shown, choose `Internet Protocol Version 4 (TCP/IPv4)` and click on the `Properties`. As you can see in the Fig. E.5, two methods of setting the IP address are available:
   - Obtain an IP address automatically.
   - Use the following IP address.

   The first option corresponds to the DHCP, the second one to the manual configuration. You should have the automatic method selected.
   The way we will use to release and renew the IP address is using the command line.
2. Open the Wireshark application and start capturing traffic on the Ethernet interface.
3. Open the CMD and display the current configuration of your Ethernet interface (write somewhere your MAC and IP addresses).
4. type `ipconfig /release`. With this command, you voluntarily release your IP configuration from **all** the interfaces where configuration was assigned via the DHCP. Now type `ipconfig /renew`. With this command, you are asking for the IP configuration from DHCP servers. Your output should look similar to the Fig. E.6.
5. Display the configuration again.
   Did you get assigned different IP address?
6. Now move to the Wireshark and stop capturing the packets.

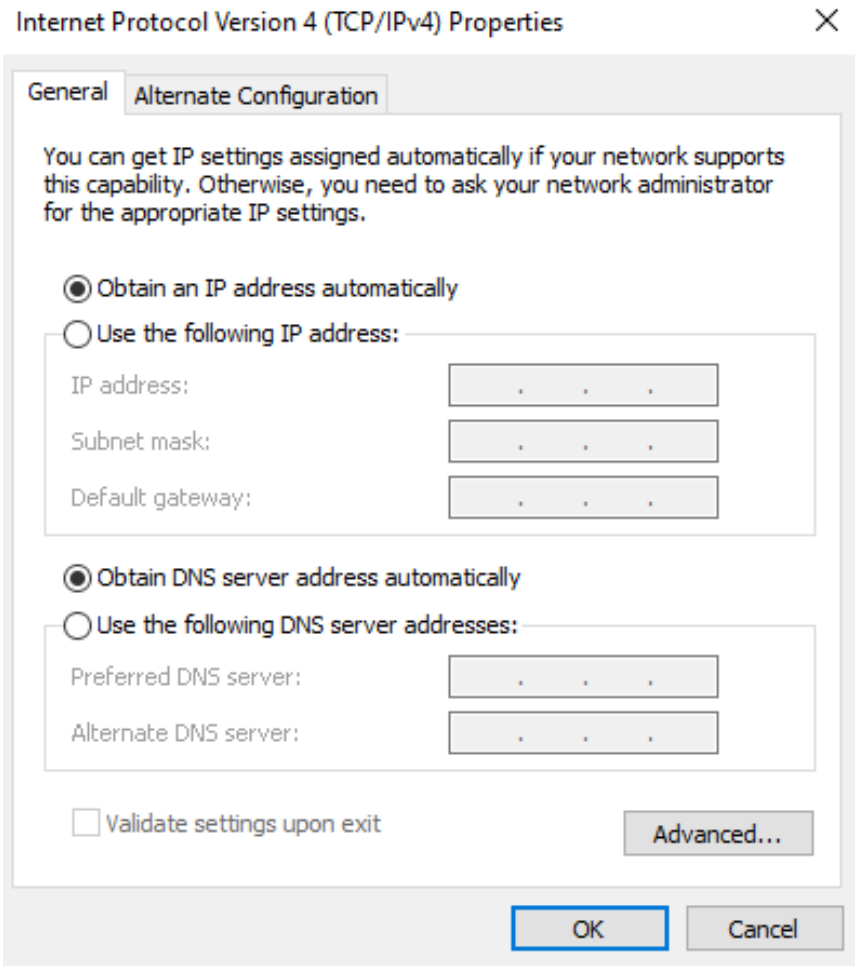


Fig. E.5: Windows IP address configuration options.

E.2.2 Objective 2

1. As you have probably captured more than just DHCP packets, apply the appropriate filter to display only the DHCP communication. Your output should look like in the Fig. E.7. Look at the information in the **Info** column and compare the message sequence with the theory.
2. Let's analyze the first packet which is the *DHCP Release* message. Select the packet and unroll the Dynamic Host Configuration Protocol section. Look at the items and get familiar with their meaning using the information given in the introduction.
   What is the message type?
   What fields are used to identify the client's physical address? What are their values?
   What logical addresses does the client specify? What is their meaning?
   What is the numeric value of DHCP Release message type?

```
C:\Users\Engineer>ipconfig /release

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::154c:fdbe:6408:9f4f%14
   Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::68fa:7cd5:929c:88f4%3
   Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8421:c0f9:689e:27d3%6
   Default Gateway . . . . . :

C:\Users\Engineer>ipconfig /renew

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::154c:fdbe:6408:9f4f%14
   IPv4 Address. . . . . . . . . . . : 192.168.206.130
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . : 192.168.206.2

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::68fa:7cd5:929c:88f4%3
   IPv4 Address. . . . . . . . . . . : 192.168.219.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8421:c0f9:689e:27d3%6
   IPv4 Address. . . . . . . . . . . : 192.168.5.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . :

C:\Users\Engineer>
```

Fig. E.6: Windows IP release and renew.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.206.130	192.168.206.254	DHCP	342	DHCP Release - Transaction ID 0x98536a14
2	4.512738	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xdae4ce80
3	5.541319	192.168.206.254	192.168.206.130	DHCP	342	DHCP Offer - Transaction ID 0xdae4ce80
4	5.542027	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0xdae4ce80
5	5.542139	192.168.206.254	192.168.206.130	DHCP	342	DHCP ACK - Transaction ID 0xdae4ce80

Fig. E.7: Captured DHCP communication.

What is the source port?
What is the destination port?
What is the destination IP address?

What is the destination MAC address?

3. Now move to the next packet – *DHCP Discover*. Here the renewal process of IP address begins. As you can see, source IP address changed to the *0.0.0.0* value. As client released its address, no valid IP address is configured on the interface. Therefore, default address is used as the source[4]. Also as client released complete configuration, he sends the message as broadcast to inform all the possible DHCP servers about his interest to obtain address. Compare the values inside DHCP section with the previous packet.
   What is the message type?
   How did the Client IP address change?
   Is the DHCP Server Identifier present? Why?
   What is the source port?
   What is the destination port?
   What is the destination IP address?
   What is the destination MAC address?
   You can probably notice the special option *Requested IP address* is present although he has not received any offers. This is due to the fact that client remembers its previous address and wishes to continue with its usage. Server that provided given address decides whether it is possible or not and provides him available address.
4. Move to the *DHCP Offer* packet. Server reads the requested IP address and checks its availability. In most of the cases, clients get the same address offered.
   What is the message type?
   What parameters are included as the options?
   What is the IP address lease time?
   What is the source port?
   What is the destination port?
   What is the destination IP address?
   What is the destination MAC address?
   Notice the strange situation where server directs the offer to the offered IP address even though the client does not use it yet. This prevents other devices from processing the offer further beyond the network layer.
5. The next packet is *DHCP Request*. The client receives the offer, accepts it and generates request to inform every possible DHCP server about his choice.
   What is the message type?
   What IP addresses does the client specify in the options fields?
   What is the source port?

[4]The *0.0.0.0* address just indicates that the device doesn't have valid IP address, it is not used for the communication, where it would be used as the destination address.

What is the destination port?
What is the destination IP address?
What is the destination MAC address?

6. The final step is the confirmation of the server that the client can start to use the offered configuration. For this purpose the *DHCP ACK* message is used.
   What is the message type?
   Compare this message with the offer? What items did change?
   Did the transaction ID change during the whole communication?
   What is the source port?
   What is the destination port?
   What is the destination IP address?
   What is the destination MAC address?
   After the client receives the acknowledgment, he can start communicating with the given parameters.
7. For capturing the lease time extension communication you would have to wait half the time of lease time, but there is no time for that. To explore the communication see Fig. E.8. You can see that the new transaction occured between client and server consisting of *DHCP Request* and *DHCP ACK* messages. The original lease time offerred by the server is shown in the Fig. E.9.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.206.130	192.168.206.254	DHCP	342	DHCP Release - Transaction ID 0x33eb98c9
2	4.550595	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xf65f124b
3	4.833345	192.168.206.254	192.168.206.130	DHCP	342	DHCP Offer - Transaction ID 0xf65f124b
4	4.833924	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0xf65f124b
5	4.834072	192.168.206.254	192.168.206.130	DHCP	342	DHCP ACK - Transaction ID 0xf65f124b
6	904.710463	192.168.206.130	192.168.206.254	DHCP	358	DHCP Request - Transaction ID 0xee1996cd
7	904.710687	192.168.206.254	192.168.206.130	DHCP	342	DHCP ACK - Transaction ID 0xee1996cd

Fig. E.8: Captured DHCP lease time extension.

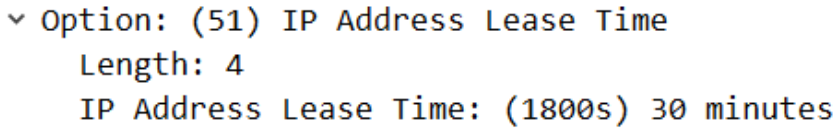


Fig. E.9: Offered DHCP lease time.

Determine the time it took to begin the lease time extension process from the original configuration assignment (use the **Time** column where the values in seconds are displayed).
How many minutes did it take to start the process? Does it correspond to the introduction?
How many minutes (or hours/days, with regard to your lease time, choose the

most appropriate time value) would you have to wait to see the lease time extension process on your computer?

E.2.3 Objective 3

1. Now let's generate the graphs of the captured communication.
2. Use the standard procedure, i.e. **Statistics > I/O Graphs**, delete all the graphs, add new one where you set the **Graph Name** "DHCP communication", set **Display Filter** to `dhcp`, the **Color** as you like. Next set the **Y Axis** to **Packets** and **Interval** to **1 sec**. Don't forget to tick **Enabled**. The output should look like in the Fig. E.10. The data for the graph are exported to the Tab. E.1.

Tab. E.1: Table of packets sent during the DHCP communication.

Seconds	0	1	2	3	4	5
Packets	1	0	0	0	1	3

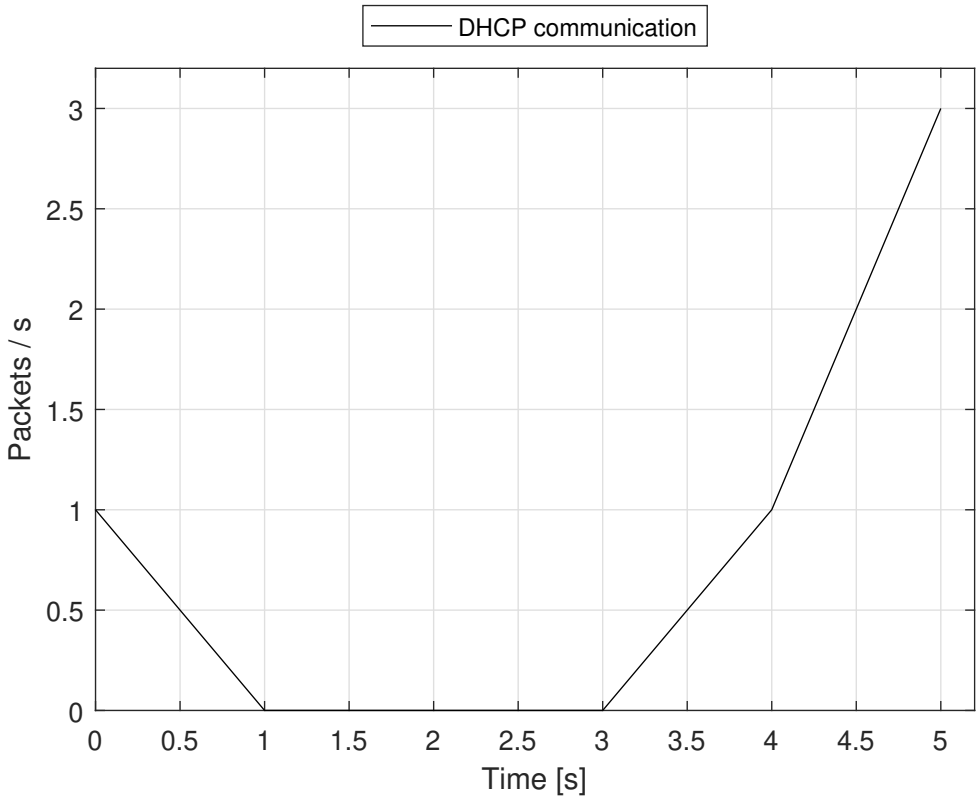


Fig. E.10: Packets sent during the DHCP communication.

3. Change the **Y Axis** to **Bytes** and redraw the graph. The output should look similar to the Fig. E.11. The data for the graph are exported to the Tab. E.2.

Tab. E.2: Table of bytes sent during the DHCP communication.

Seconds	0	1	2	3	4	5
Bytes	342	0	0	0	344	1054

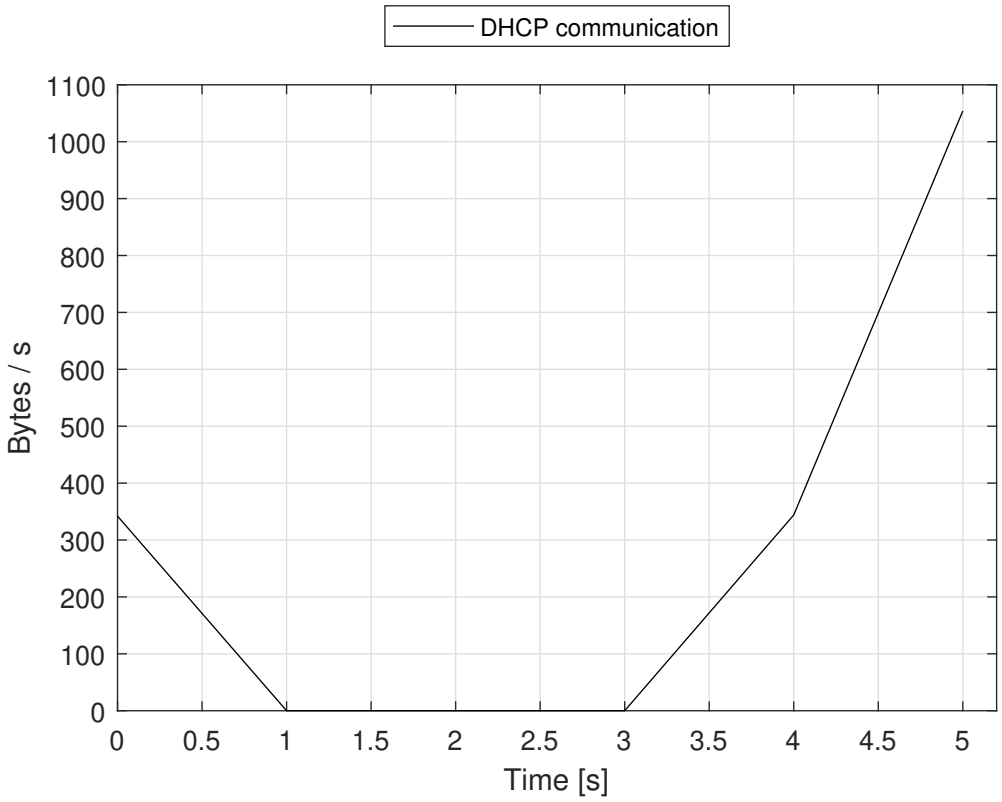


Fig. E.11: Bytes sent during the DHCP communication.

E.3 Packet Tracer

E.3.1 Objective 4

1. Open the file you were supposed to save in the last laboratory and save it under a new name. Here you will add some devices to the topology according to the Fig. E.1.
2. Remove the link connecting Server with R2.
3. Add one **2960** switch from the **Switches** subgroup inside **Network Devices** group. Then add one PC (that could serve as the administrator's PC) from the **End Devices** group.
4. Interconnect the devices as following:
   - Connect PC's **FastEthernet0** port with switch's **FastEthernet0/1** port using the **Copper Straight-Through** cable.
   - Connect Server's **FastEthernet0** port with switch's **FastEthernet0/2** port using the **Copper Straight-Through** cable.
   - Connect R2's **FastEthernet0/1** port with switch's **FastEthernet0/24** port using the **Copper Straight-Through** cable.
5. The addresses on the router and server should be preserved. Assign the **second available host address** to the PC and verify connectivity.
6. Save your current progress by clicking on the **File > Save**. Don't forget to save your progress after completing individual objectives!

E.3.2 Objective 5

1. Now that you have complete topology, let's set the R2 router to the role of a DHCP server. You will define two pools on the router. One for the LAN with *172.20.1.0/24* address space and one for *172.20.3.0/24* address space. The following steps have to be accomplished to set the server correctly:
   - Exclude the IP addresses that shouldn't be assigned by the DHCP server. Basically, these addresses represent statically assigned addresses to the servers, router interfaces and devices like printers etc.
   - Configure the pool where you specify address range, default gateway, DNS server, lease time etc.
2. First of all, let's explicitly define the addresses that shouldn't be assigned by the DHCP server.
   How many addresses will be excluded?

The sequence of commands is as following:

```
R2(config)#ip dhcp excluded-address 172.20.1.254
R2(config)#ip dhcp excluded-address 172.20.3.1
172.20.3.10
R2(config)#ip dhcp excluded-address 172.20.3.254
```

As you have probably correctly estimated, 3 addresses are defined as static. Two for the router interfaces and one for the server.
The `ip dhcp excluded-address` command followed by the IP address defines which individual addresses shouldn't be assigned by the server[5]. But you can also define a range of addresses. This is the case for the *172.20.3.0/24* network. Here we suppose that other servers will be connected to the network in the future, so we reserve them 10 addresses (including the server's that is already connected) with specifying the **low** and **high** IP addresses. Every address within this range including the two specified will be omitted during the address assignment.

3. The next step is to define individual pools. The following pool represents the *172.20.1.0/24* network:

```
R2(config)#ip dhcp pool CLIENT-NETWORK
R2(dhcp-config)#network 172.20.1.0 255.255.255.0
R2(dhcp-config)#default-router 172.20.1.254
R2(dhcp-config)#dns-server 172.20.3.1
```

The `ip dhcp pool` command defines the name of the pool where configuration parameters will be specified next. In our case, the name "CLIENT-NETWORK"is used. The `network` network command followed by the network IP address and subnet mask determines the address space for the given pool. The `default-router` command determines the default gateway of the subnet and the `dns-server` defines, as the name suggests, the DNS server.

4. Configure the DHCP pool for the server subnet using the same sequence of commands, but with correct addresses for the subnet. Use the "SERVER-NETWORK"as the pool name.
5. Everything is ready now for the dynamic configuration of clients. Let's move to the **Simulation** mode and filter only the DHCP. Choose one PC from the client network, go to the **IP Configuration** and switch from *Static* to *Dynamic*. Minimize the PC configuration window. You should now see one packet generated (*DHCP Discover message*). Explore its content.

[5]Notice that the command is typed in the **Configuration mode**, not in the specific pool configuration.

Are there any IP addresses set in the DHCP section?

What is the source port?

What is the destination port?

What is the destination IP address?

What is the destination MAC address?

6. Click **Capture then forward** until the packet reaches the router[6]. The *DHCP Offer* message is generated. The output should look like in the Fig. E.12.

   Which pool is used to assign the IP configuration?

   What IP addresses does the router provide to the client?

   What is the source port?

   What is the destination port?

   What is the destination IP address?

   What is the destination MAC address?

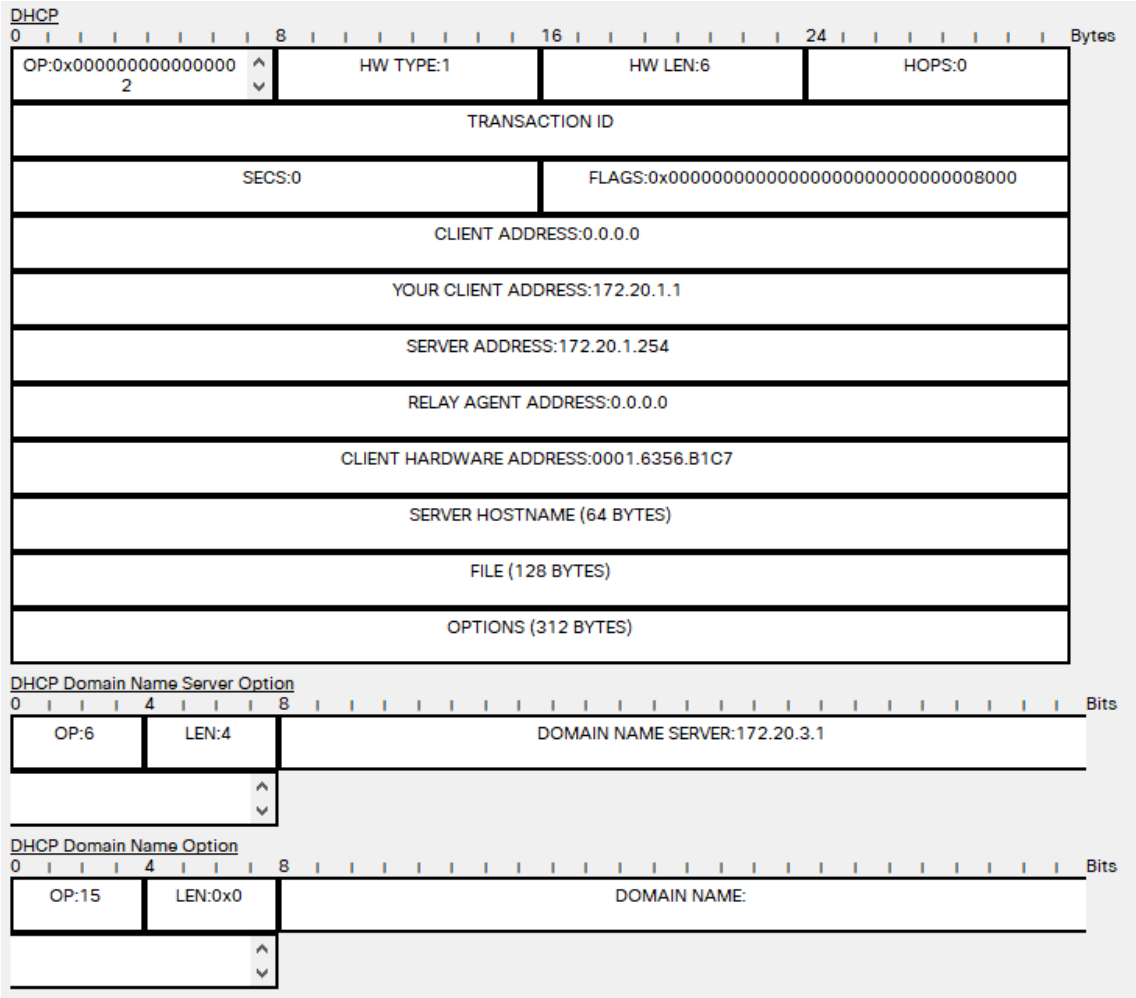


Fig. E.12: DHCP Offer message generated by the router.

[6]If the *Outbound PDU Details* tab does not appear, click **Capture then forward** once again.

7. Click **Capture then forward** until the *DHCP Request* message is generated and explore its content.
   What IP addresses does the client specify?
   What is the destination IP address?
   What is the destination MAC address?
8. Click **Capture then forward** until the *DHCP Ack* message is generated by the router and explore its content.
   How does the packet content differ from the DHCP Offer message?
   What is the destination IP address?
   What is the destination MAC address?
9. Now click **Capture then forward** until the *DHCP Ack* message reaches the client. Open the PC configuration window again. You should see the applied configuration as in the Fig. E.13.

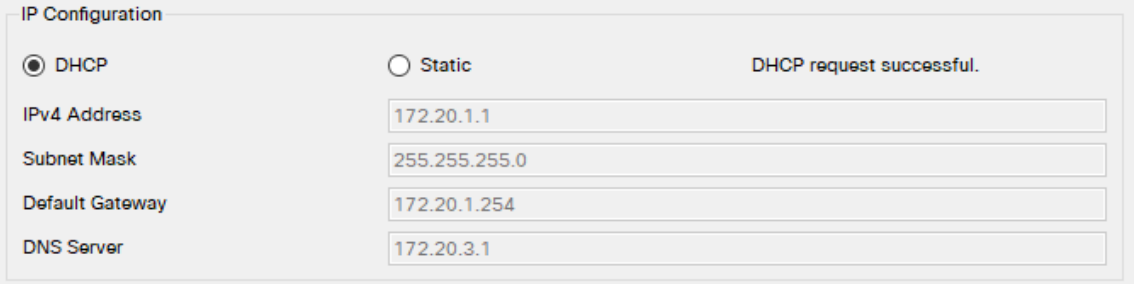


Fig. E.13: DHCP client configuration.

10. Switch to the **Realtime** mode and verify the correct configuration by accessing the server's web page (*www.mywebpage.com*).
11. Switch to the **Simulation** mode again and use the same process for address assignment on the other PC in the same subnet.
    What IP address does it get assigned?
12. Now use the DHCP to obtain IP address on the administrator's PC (in the server network).
    Which pool is used to choose the address?
    What IP address is assigned to it?
13. Switch to the **Realtime** mode and verify connectivity between the devices.

E.3.3 Objective 6

1. In this part of the laboratory, you will configure the DHCP service on the server. Before this happens, disable the DHCP service on the R2 router using the `no service dhcp` in the **Configuration mode**. By typing this command

the DHCP configuration is preserved but the service is turned off[7]. Verify this by switching the admnistrator's PC configuration to *Static* and back to *Dynamic*. The IP address from the *169.254.0.0/16*[8] address range should be assigned.

2. Now configure the service on the Server device. Select **DHCP** in the **Services** tab. You can see that the service is turned off by default and one pool is already present. Let's configure a pool for the client network. Use the same information in the *Pool Name*, *Default Gateway* and *DNS Server* fields as in the router configuration. Set the *Start IP Address* to *172.20.1.1* and *Subnet Mask* to *255.255.255.0*. Set the *Maximum Number of Users* to 100 and click **Save**. The new pool is created as displayed in the Fig. E.14.

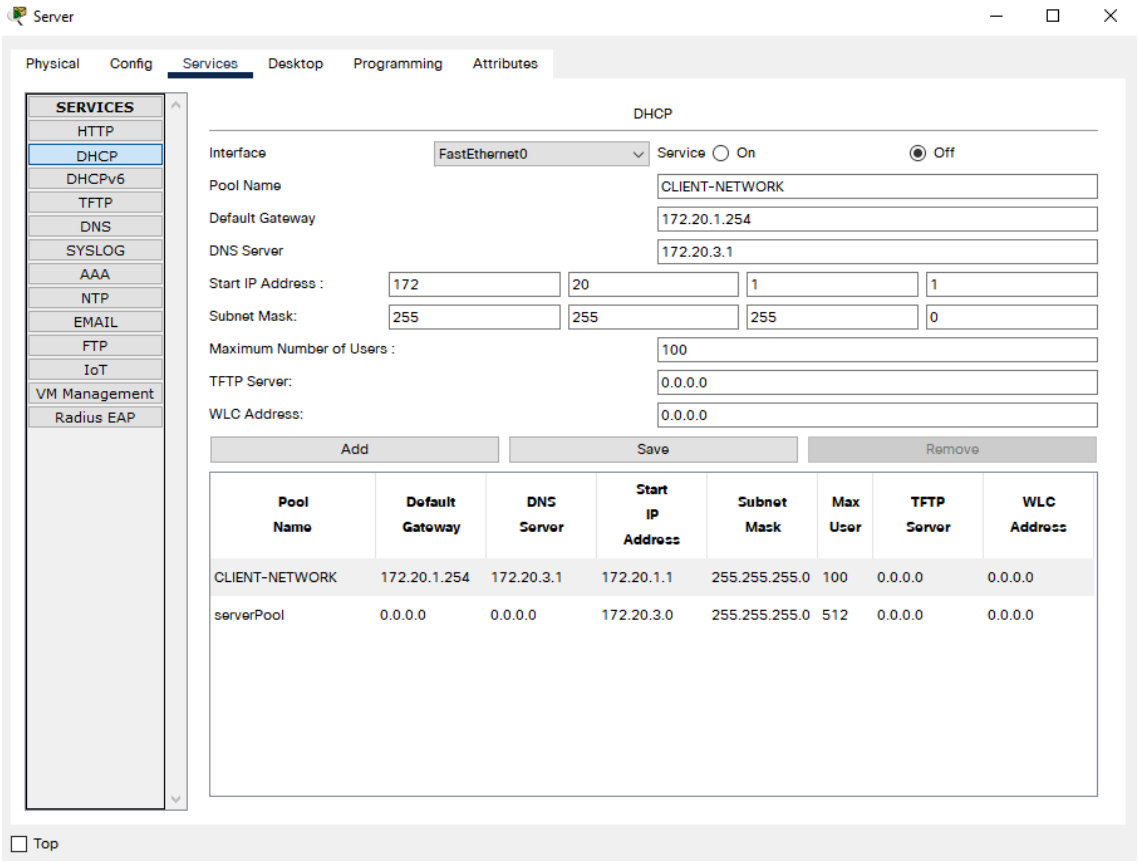


Fig. E.14: DHCP client network configuration on the server.

3. Create a pool for the server network the same way as the client network. This time click on **Add** to preserve the client network configuration (otherwise it

[7]NOTE: When you close and open the .pkt file again, the service is enabled by default. So with every start you have to manually turn the service off.

[8]This specific address range called APIPA (Automatic Private IP Addressing) is used by the operating systems to assign addresses automatically when there is no DHCP server reachable [4].

would be overwritten). Use the same parameters as in the router configuration. Set the *Start IP Address* to *172.20.3.11*, *Subnet Mask* to *255.255.255.0* and the *Maximum Number of Users* to 10. Enable the service by ticking the **On** option. The final result is displayed in the Fig. E.15.

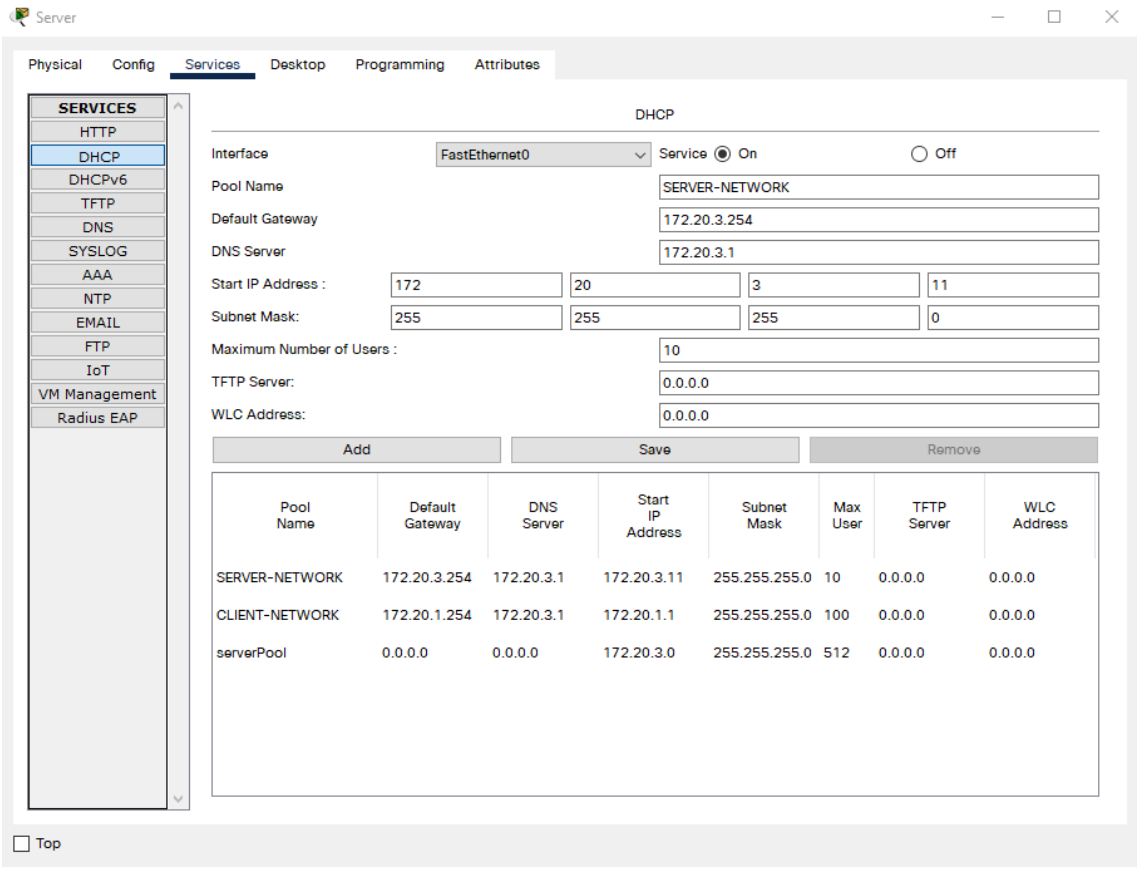


Fig. E.15: Final server DHCP configuration.

4. On the administrator's PC, set the IP configuration to *Static* and back to *Dynamic*. It should receive the first host address specified in the pool (as displayed in the Fig. E.16)[9].
5. Reload DHCP configuration on the PCs in the client network as in the case of administrator's PC.
   From which range did the PCs receive the addresses? Why is that?
   Switch back to *Static* configuration.
6. As the server is in the different network, it is necessary to configure the router to the role of DHCP relay agent. The configuration is quite simple:

[9]If it did receive the IP address from the default server pool, select this pool in the DHCP service configuration and click **Save** to move it to the top. Then reload the IP configuration on the PC.

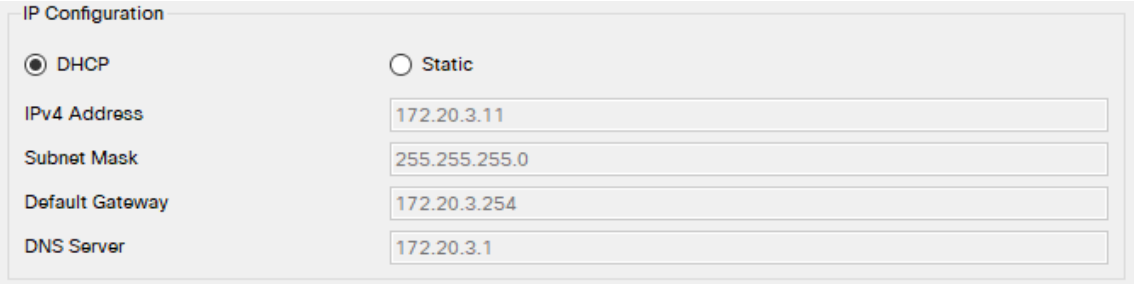


Fig. E.16: Administrator's PC DHCP configuration.

```
R2(config)#interface FastEthernet0/0
R2(config-if)#ip helper-address 172.20.3.1
```

First the interface is selected. Then the `ip helper-address` command followed by the DHCP server address is typed. This configuration achieves the router to forward DHCP messages, that are arriving at the **FastEthernet0/0** interface, to the DHCP server identified by the *172.20.3.1* address.

7. Switch to the **Simulation** mode and enable only one PC from the client network to obtain the DHCP configuration. Minimize the PC configuration window. New packet (the *DHCP Discover* message) is generated. Click **Capture then forward** until the packet reaches the router. Explore the contents of *Inbound PDU Details* and *Outbound PDU Details* and compare the two. The DHCP section inside *Outbound PDU Details* should look as in the Fig. E.17.
   What changes occur inside the DHCP section?
   How do the source and destination MAC addresses change? Hint: If you are not sure about the destination MAC address, display complete configuration of the server interface.
   How do the source and destination IP addresses change?
8. Click **Capture then forward** until the packet (the *DHCP Discover* message) reaches the server[10].
   What IP addresses are included in the DHCP section?
   What is the destination IP address?
   What is the destination MAC address?
9. Click **Capture then forward** until the packet (the *DHCP Offer* message) reaches the router and compare the contents of incoming and outgoing packet.
   How does the DHCP section change?
   How do the source and destination MAC addresses change?
   How do the source and destination IP addresses change?

[10]If the *Outbound PDU Details* tab does not appear, click **Capture then forward** once again.

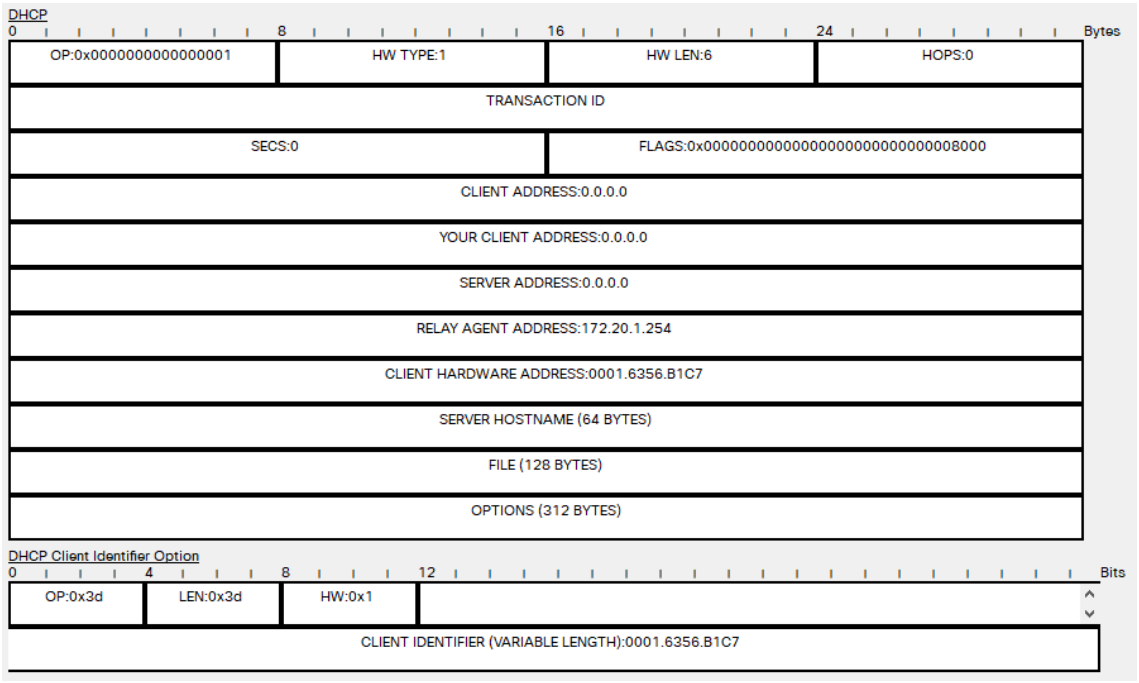


Fig. E.17: DHCP section inside Outbound PDU details.

10. Click **Capture then forward** until the packet reaches the PC and new message (the *DHCP Request* message) is generated.
    Does the content of DHCP section differ from the incoming offer?
    What is the destination IP address?
    What is the destination MAC address?
11. Click **Capture then forward** until the packet reaches the router.
    What fields are changed inside DHCP section of the outgoing packet compared to the incoming packet?
    What is the destination IP address?
    What is the destination MAC address?
12. Click **Capture then forward** until the packet reaches the server and the *DHCP Ack* message is generated.
    Are there any additional parameters in comparison with the offer?
    What is the destination IP address?
    What is the destination MAC address?
13. Click **Capture then forward** until the packet reaches the router.
    How do the source and destination MAC addresses change?
    How do the source and destination IP addresses change?
    Click **Capture then forward** until the packet reaches the PC. Check if the configuration is set in the minimized window. Switch to the **Realtime** mode and verify connectivity by accessing the *www.mywebpage.com* web page.

14. Obtain the IP address on the other PC in the same subnet.
    What address does it get assigned?
15. Verify connectivity with other devices.

E.4 Final questions

1. How many messages are exchanged during the initial address assignment? What are they called?
2. What is defined by the lease time?
3. What is the relay agent used for?
4. When a relay agent is used in the DHCP communication, how does a server identify which pool to use to assign the address?
5. How does a relay agent alter destination IP and MAC addresses while forwarding the DHCP messages?

Literature

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131. In: *RFC Editor* [online]. 1997 [cit. 13. 05. 2022]. Available at:
<https://www.rfc-editor.org/info/rfc2131>.

[2] MENS, Rudy. DHCP Lease Time — What is it and How does it work?. In: *LazyAdmin* [online]. 2019 [cit. 13. 05. 2022]. Available at:
<https://lazyadmin.nl/home-network/dhcp-lease-time/>.

[3] Huawei. DHCP Messages. In: *Huawei* [online]. 2019 [cit. 13. 05. 2022]. Available at:
<https://support.huawei.com/enterprise/en/doc/EDOC1100058931/25cd2dfc/dhcp-messages#section_dc_vrp_dhcp_feature_000702>.

[4] TechTarget Contributor. Automatic Private IP Addressing (APIPA). In: *TechTarget* [online]. 2005 [cit. 16. 05. 2022]. Available at:
<https://www.techtarget.com/whatis/definition/Automatic-Private-IP-Addressing-APIPA>.

Acronyms

APIPA Automatic Private IP Addressing

ARP Address Resolution Protocol

CCNA Cisco Certified Network Associate

CIDR Classless Inter-Domain Routing

CLI Command Line Interface

CMD Command Prompt

DBD Database Description

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DTE Data Terminal Equipment

GUI Graphic User Interface

HTTP Hypertext Transfer Protocol

IP Internet Protocol

ISN Initial Sequence Number

LAN Local Area Network

LSAck Link State Acknowledgment

LSR Link State Request

LSU Link State Update

MAC Media Access Control

OSPF Open Shortest Path First

RIP Routing Information Protocol

RTT Round Trip Time

SPF Shortest Path First

SPT Shortest Path Tree

TCP Transmission Control Protocol

VLSM Variable-Length Subnet Mask