VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

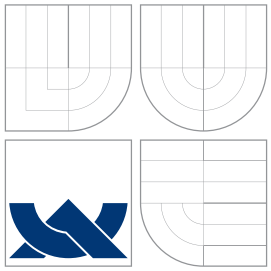
DETEKCE IDENTITY NA RŮZNÝCH VRSTVÁCH ARCHITEKTURY TCP/IP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

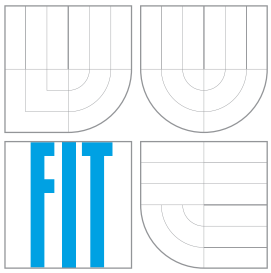
AUTOR PRÁCE
AUTHOR

MARTIN HOLKOVIČ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE IDENTITY NA RŮZNÝCH VRSTVÁCH ARCHITEKTURY TCP/IP

IDENTITY DETECTION IN TCP/IP ARCHITECTURE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN HOLKOVIČ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LIBOR POLČÁK

BRNO 2013

Abstrakt

Tato práce se zabývá detekcí identity uživatelů v rámci počítačových sítí na různých vrstvách architektury TCP/IP. Tyto identity jsou zjišťovány z protokolů běžících na příslušných vrstvách dané architektury. Z hlediska přidělování přístupu do sítí byly vybrány protokoly PPPoE a SLAAC. Druhým typem protokolu je aplikační protokol SMTP. U těchto vybraných protokolů byla analyzována jejich činnost spolu s možnostmi vytváření metainformací o příslušné komunikaci. Výsledkem analýzy jsou stavové automaty. Na základě těchto stavových automatů byl navržen a implementován software, který je určen pro účely zákonných odposlechů. Implementováný software byl následně otestován na vzorových datech, v specializované laboratoři a na produkční síti.

Abstract

This work deals with detection of users within computer networks on different layers of the TCP/IP architecture. These identities are identified by protocols running on the appropriate layers of the given architecture. PPPoE and SLAAC protocols were chosen as protocols that are used for network layer address assignments. The second type of protocol is the application protocol SMTP. We analysed communication using the chosen protocols in order to create metadata about the corresponding communication. The results of the analysis are finite state machines. Based on these finite state machines, software for legal interception was designed and implemented. Implemented software was tested on samples of data, in a specialized laboratory, and in a production network.

Klíčová slova

Zákonné odposlechy, dynamická identifikace, IRI-IIF, PPPoE, SLAAC, SMTP.

Keywords

Lawful interception, dynamic identification, IRI-IIF, PPPoE, SLAAC, SMTP.

Citace

Martin Holkovič: Detekce identity na různých vrstvách
architektury TCP/IP, bakalářská práce, Brno, FIT VUT v Brně, 2013

Detekce identity na různých vrstvách architektury TCP/IP

Prohlášení

Čestně prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Libora Polčáka a za pomoci uvedené literatury.

.....
Martin Holkovič
21. mája 2013

Poděkování

Děkuji Ing. Liborovi Polčákovi, vedoucímu bakalářské práce, za odborné vedení a pomoc při vypracovávání této práce.

Obsah

Kapitola 1

Úvod

V súčasnej dobe sú počítačové siete neoddeliteľnou súčasťou nášho života. Počítačové siete umožňujú široké možnosti uplatnenia od zábavy až po armádne účely. Všetky tieto možnosti sú založené na skutočnosti, že počítače dokážu navzájom komunikovať a tým si vymieňať informácie. Ľudia preto, aby mohli využívať nové možnosti využitia počítačov, komunikujú prostredníctvom počítačových sietí navzájom (medzi užívateľmi) alebo prostredníctvom k tomu určených poskytovateľov služieb (napr. webové stránky). Pre adresnosť účastníkov komunikácie, využívajú účastníci na svojich koncových zariadeniach tzv. identifikátory. Identifikátory slúžia pre jednoznačnú identifikáciu užívateľov, služieb, a pod.

Pomocou detekcie používania identifikátorov sme schopní detegovať identitu koncových užívateľov, ktoré tieto identifikátory využívajú. Problémom však je, že každý sieťový protokol, prípadne aplikácia si priraďovanie týchto identifikátorov udržuje vo vlastnej réžii nezávisle na ostatných. Mať tak celkový prehľad o priradení týchto identifikátorov a o ich používaní jednotlivými identitami v sieti nie je triviálna záležitosť. Avšak existujú niektoré činnosti pre ktoré je znalosť týchto informácií výhodná až nevyhnutná. Príkladom činnosti kedy je vhodné vedieť, aké identifikátory využívajú jednotlivé identity (užívatelia) je správa a bezpečnosť siete. Správcovia počítačových sietí sú zodpovední za to, aby sieť fungovala tak ako sa od nej očakáva a aby zdroje v nej umiestnené neboli zneužité.

Ďalším príkladom, kedy je nutné mať priradenie a používanie identifikátorov pod dohľadom sú systémy pre zákonné odpočúvanie, na ktoré sa v tejto práci zameriam. Počítačové siete, ktoré sú v dnešnej dobe rozšírené a používané, sú taktiež zneužívané na rôzne protiprávne účely. Typicky sa jedná o počítačovú sieť Internet. Ak predpokladáme, že orgány činné v trestnom konaní budú chcieť proti kriminálnikom bojovať, je nevyhnutné mať k dispozícii systémy umožňujúce získať prehľad o dianí v sieti. Tieto systémy sú stavané na základe noriem vydávaných európskym telekomunikačným inštitútom pre vydávanie štandardov *European Telecommunications Standards Institute* (ETSI). Jeden systém pre zákonné odpočúvanie je vyvíjaný v rámci projektu *Moderné prostriedky pre boj s kybernetickou kriminalitou na Internete novej generácie* (Sec6Net).

Úlohou tejto práce je tvorba softwaru, pomocou ktorého bude možné detegovať priraďovanie a používanie identifikátorov u vybraných protokolov. Software bude implementovaný vo forme modulov, ktorý bude zakomponovaný do projektu Sec6Net. Vzhľadom na rozšírenosť používania boli pre analýzu a implementáciu vybrané protokoly PPPoE, SLAAC a SMTP.

Vybrané protokoly boli analyzované na možnosť detekcie identity užívateľa s výstupom vo forme stavových automatov. Na základe vytvorených automatov boli moduly implementované. Implementované moduly boli následne otestované či zo sieťovej prevádzky správne

detegujú identitu užívateľov. Testovanie prebiehalo na predpripraven=ych ulo+zen=ych paketoch, v špecializovanom laboratóriu na fakulte FIT, na produkčnej sieti fakulty FIT a na internáte VUT. Po otestovaní implementácie sa prešlo k ich vyhodnoteniu a následne sa moduly zakomponovali do projektu Sec6Net.

Súčasťou tejto práce, ktorá sa zaoberá naprogramovaním softwaru do projektu Sec6Net sú nasledovné kapitoly:

- **2. kapitola** - Obsahuje popis architektúry systémov pre zákonné odpočúvanie spolu s bližším popisom bloku zaoberajúceho sa zberom informácií o priraďovaní identít a prebiehajúcich komunikácií.
- **3. kapitola** - Na začiatku kapitoly sa nachádza popis architektúry TCP/IP z hľadiska identít za ktorým nasleduje popis protokolov, ktorým sa táto práca venuje. Jedná sa o protokoly PPPoE, SLAAC a SMTP.
- **4. kapitola** - V tejto kapitole sa zameriam na možnosti vytvárania správ IRI (metadáta popisujúce sieťovú komunikáciu) pre každý popisovaný protokol z kapitoly 3.
- **5. kapitola** - Popisuje implementáciu modulov do systému pre zákonné odpočúvanie.
- **6. kapitola** - Obsahuje popis a priebeh testovania vytvorených modulov.
- **7. kapitola** - V kapitole sa vyhodnotia implementované nástroje, popíše sa problém spojený s neúplnosťou vstupných dát a navrhnú sa možné rozšírenia modulov.
- **8. kapitola** - V poslednej kapitole vyhodnotím obsah celej práce.

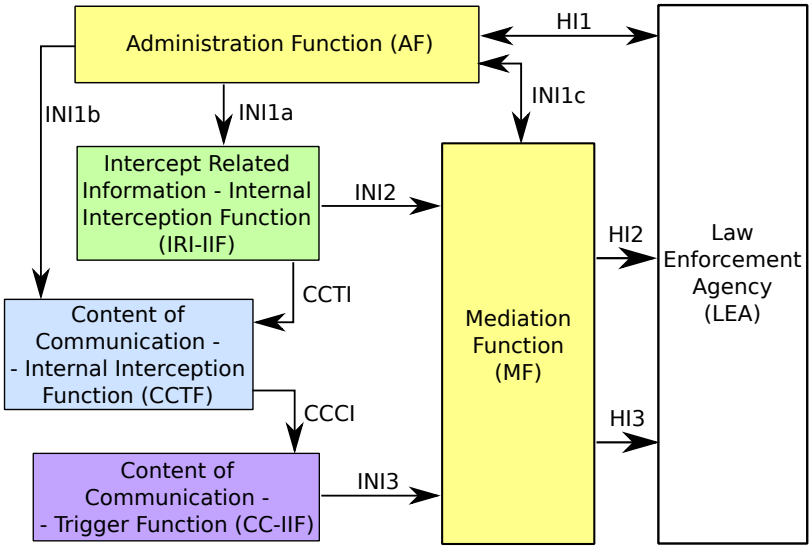
Kapitola 2

Systémy pre zákonné odpočúvanie

Systémy pre zákonné odpočúvanie sú využívané orgánmi činnými v trestnom konaní pri objasňovaní rôznych zločinov. Základnou myšlienkou je zaistiť, aby v prípade podozrenia páchania trestného činu bolo možné podozrivého odpočúvať v prostredí počítačovej siete na ktorej beží protokol *Internet Protocol* (IP) [24]. Pri podaní žiadosti o odpočúvanie získajú orgány činné v trestnom konaní buď signalizačné dáta o prebiehaných komunikáciách (metadáta) alebo identickú kópiu dát, ktoré podozrivý zo siete prijíma alebo do siete vysiela. Metadáta obsahujú základné informácie o komunikácii ako napr. kto s kým komunikuje, kedy komunikácia prebehla a pod.

2.1 Schéma systému

Súčasťou noriem popisujúcich systémy pre zákonné odpočúvanie je aj schéma systému pre zákonné odpočúvanie. Schému je možno vidieť na obrázku 2.1, jedná sa o zjednodušenú schému zo štandardu ETSI TR 102 528 [7].



Obrázok 2.1: Referenčný model systému pre zákonné odpočúvanie

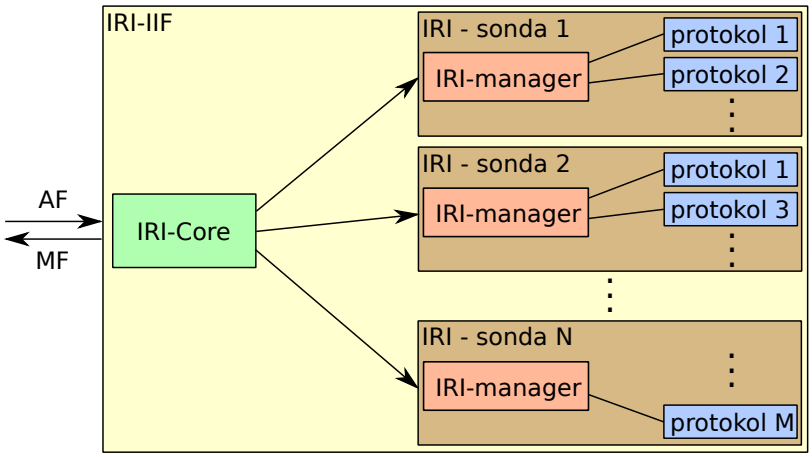
Systém pre zákonné odpočúvanie (obrázok 2.1) sa podľa štandardu ETSI TR 102 528 skladá z nasledujúcich blokov:

- **Law Enforcement Agency** (LEA, v slovensky hovoriacich krajinách sa taktiež používa označenie Orgány činné v trestnom konaní) - Jedná sa o zákonom poverenú organizáciu, ktorá má právomoci žiadať o odpočúvanie a spracovávať výsledky z týchto odpočúvaní pre ďalšie použitie (napr. dôkazový materiál pre súdne pojednávanie).
- **Administration Function** (AF, v slovensky hovoriacich krajinách sa taktiež používa označenie Administratívna funkcia) - Blok slúži na správu požiadaviek na odpočúvanie, ktoré prijíma od orgánov LEA pomocou rozhrania HI1. Rozhranie HI1 býva často vo forme papierovej podoby a je preto nutné ručne zadať parametre odpočúvania do bloku AF. V niektorých prípadoch, napr. keď nie je možné prenášať obsah komunikácie orgánom LEA, blok AF môže o tomto stave informovať orgány LEA pomocou rozhrania HI1.
- **Mediation Function** (MF, v slovensky hovoriacich krajinách sa taktiež používa označenie Mediačná funkcia) - Blok zaisťuje zber informácií a dát o prebiehajúcich komunikáciách a formátuje ich do formy vhodnej pre LEA. Informácie o prebiehajúcich komunikáciách sú prenášané rozhraním HI2, zatiaľ čo samotný obsah dát je posielaný separátnym rozhraním HI3. V prípade ak je súčasťou odpočúvania aj žiadosť o obsah komunikácie, je nutné aby boli informácie o komunikácii synchronizované so samotným obsahom komunikácie. Informácie o aktívnych odpočúvaniach blok získava z bloku AF rozhraním INI1c, ktoré taktiež využíva ako spätnú väzbu na informovanie o stave týchto odpočúvaní.
- **Intercept Related Information Internal Interception Function** (IRI-IIF) - Blok sa môže skladať z viacero zariadení, pričom každé sa zaoberá zberom informácií o prebiehajúcich komunikáciách a priradených identifikátoroch v počítačovej sieti. Podľa priradených identifikátorov následne deteguje identitu jednotlivých užívateľov. Na základe konfigurácií získaných od AF pomocou rozhrania INI1a sa tieto informácie pomocou rozhrania INI2 ďalej posielajú do bloku MF.
- **Contents of Communication Internal Interception Function** (CC-IIF) - Blok sa rovnako ako blok IRI-IIF môže skladať z viacerých zariadení. Každé zariadenie slúži na duplikovanie obsahu komunikácie odpočúvaného zariadenia a prostredníctvom rozhrania INI3 ho kopíruje do bloku MF.
- **Contents of Communication Trigger Function** (CCTF) - Na základe informácií z bloku AF, ktoré získa rozhraním INI1b a CCTI, blok identifikuje vhodné zariadenie bloku CC-IIF pre zachytávanie obsahu komunikácie. Vhodnému zariadeniu bloku CC-IIF prostredníctvom rozhrania CCCI pošle parametre na základe ktorých bude vytvárať kópie len tých dát na ktoré sa vzťahuje nejaké odpočúvanie.

2.2 Vnútorná štruktúra IRI-IIF

Úlohou bloku IRI-IIF je detekcia identít na sieti. K tomu aby dokázal blok identifikovať identity na sieti je nutné, aby mal blok prehľad o pridelených identifikátoroch. Prehľad o pridelených identifikátoroch blok získava analýzou protokolov zo siete. Pod detekciou identifikátorov sa myslí zisťovanie kedy a komu bol identifikátor priradený, a aké ďalšie

informácie sa k danému identifikátoru viažu. Tieto činnosti sa dajú chápať ako metadáta o spojeniach na sieti. Tieto metadáta sa spracovávajú neustále, aj keď nie je aktívne žiadne odpočúvanie. V prípade žiadosti o okamžité odpočúvanie totiž musí byť jasné, aké identifikátory má odpočúvaný subjekt aktuálne priradené. Normy sa detailnou špecifikáciou IRI-IIF nezaoberajú, preto vo zvyšku kapitoly sa zameriam na štruktúru systému pre zákonné odpočúvanie z hľadiska projektu Sec6Net.



Obrázok 2.2: Schéma vnútornej štruktúry IRI-IIF

Blok IRI-IIF sa vnútorne skladá z ďalších častí [15], ktoré sú zobrazené na obrázku 2.2. Hlavnou časťou je *IRI-Core*. *IRI-Core* prijíma žiadosti o odpočúvania a ich správu z bloku AF. V prípade aktívneho odpočúvania sú všetky metadáta spojené s komunikáciou odpočúvaného subjektu posielané bloku MF. Samotné metadáta nie sú vytvárané v časti *IRI-Core*, ale sú prijímané z *IRI-sond* a prípadne posielané ďalej.

Neoddeliteľnou súčasťou bloku IRI-IIF sú *IRI-sondy* (ďalej len sondy). Sonda je na vhodných miestach v sieti zapojené samostatné fyzické zariadenie, ktoré analyzuje dáta na sieti a posiela metadáta do *IRI-Core*. Bod v sieti, kde je umiestnená sonda, sa nazýva *Intercept Access Point* (IAP). Umiestnenie bodu IAP v sieti záleží na type protokolov, ktoré chceme sondou v sieti analyzovať. V každej časti sieti sa nachádzajú iné druhy protokolov a preto je pri každom analyzovanom protokole nutné sa zamyslieť nad vhodným umiestnením bodu IAP. Varianty umiestnenia bodu IAP v sieti sú zobrazené na obrázku 2.3 a ich popis je nasledujúci:

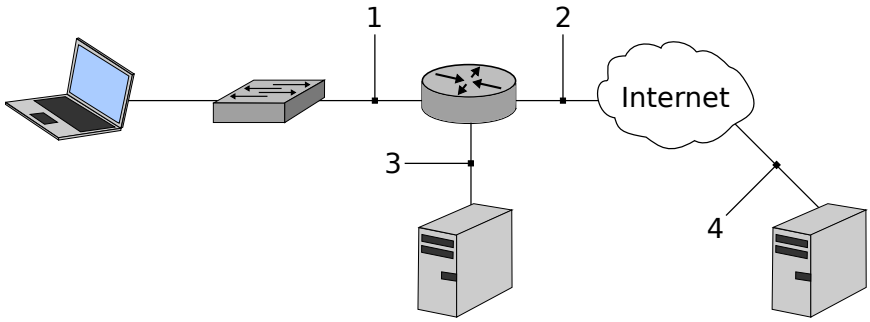
1. Jedná sa o umiestnenie bodu u *Internet Service Provider* (ISP) na rovnakej podsieti ako sa nachádzajú koncoví užívatelia. Toto umiestnenie je nevyhnutné pre protokoly ako sú SLAAC alebo PPPoE, kedy komunikácia užívateľov nemusí smerovať na žiadne servery (či už na strane ISP alebo internetu), ale smeruje do sieťovej infraštruktúry ISP. Príkladom môže byť smerovač, s ktorým užívateľ vytvára PPPoE spojenie. Najlepšie umiestnenie sond, z hľadiska ich počtu, je na linke spájajúcu podsieť užívateľov s prvým smerovačom na strane ISP.

2. Ak chceme zachytávať komunikáciu užívateľov smerujúcu do internetu, čo sa počtu sond týka, je ich najlepšie umiestnenie na linke medzi ISP a internetom. Z internetovej komunikácie sú pre zákonné odpočúvania najzaujímavejšie aplikačné protokoly.

Príkladom je SIP, SMTP a XMPP. Nevýhodou tohto umiestnenia je veľké množstvo dát, ktoré musia byť sondy schopné spracovávať v prípade veľkých ISP.

3. V prípade, že ISP vlastní servery, ktoré slúžia napr. pre centrálne overenie jeho zákazníkov, je lokalita týchto serverov vhodná pre analýzu protokolov smerujúcich na dané servery. Pre minimálne množstvo sond analyzujúcich túto komunikáciu sú možné dve varianty zapojenia sond. V závislosti od počtu a zapojenia týchto serverov, môže byť vhodné sondy umiestniť na linky spájajúce servery so sieťou alebo linky spájajúce podsieť na ktorej sa nachádzajú servery so zvyškom infraštruktúry ISP. Príkladom takto analyzovaných protokolov môže byť DHCP, RADIUS, apod.

4. Umiestnenie bodu je typické pre serverovne, kde sa nachádzajú poskytovatelia rôznych služieb na internete. Na rozdiel od bodu č. 2 na tomto mieste nevidíme všetku komunikáciu vybraných užívateľov, ale všetku komunikáciu vybraných služieb resp. protokolov. Podobne ako v bode č. 2 aj týmto bodom môže prechádzať veľké množstvo dát, ktoré musia nasadené sondy dokázať spracovávať. Toto umiestnenie je vhodné pre aplikačné protokoly rovnako ako v bode č. 2.



Obrázok 2.3: Varianty umiestnenia bodu IAP v sieti

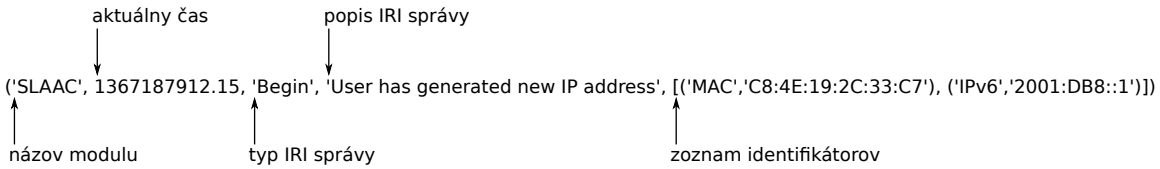
Sonda má modulárnu štruktúru (obrázok 2.2) podobne ako samotná IRI-IIF. Každá sonda obsahuje centrálny prvok s názvom *IRI-Manager* a premenlivý počet modulov. Modul je aplikácia, ktorá sa stará o analýzu len jedného konkrétneho protokolu (napr. SLAAC). Modul je z hľadiska IRI-IIF jediným prvkom, ktorý prichádza do kontaktu s paketmi na odpočúvanej sieti. Každý modul analyzuje svoj protokol, na ktorý bol vytvorený, a vytvára metadáta z analýzy paketov na sieti. Modul je teda zdroj týchto metadát, ktoré posiela ďalej na *IRI-Manager*. Úlohou *IRI-Managera* je posielanie týchto metadát do centrálneho prvku IRI-IIF (*IRI-Core*).

2.3 Správy IRI

Ako bolo spomínané v predchádzajúcich odsekoch, moduly generujú metadáta, ktoré sú ďalej posielané do *IRI-Core* a prípadne do MF. Presné označenie týchto metadát sú správy *Intercept Related Information* (IRI) [7]. Formát správ IRI má formu usporiadanej n-tice, ktorá sa skladá z nasledujúcich položiek:

- **Názov modulu** - Jednoznačná identifikácia modulu.
- **Aktuálny čas** - Presný čas, v ktorom nastala detegovaná udalosť (nie čas kedy je IRI správa odoslaná)
- **Typ IRI správy** - Správy IRI sú štyroch rôznych nasledujúcich kategórií:
  - **IRI Report** - V prípade, že koncová stanica alebo klient komunikuje na sieti pred vytvorením spojenia (pred odoslaním správy*IRI Begin*), sú tieto informácie odosielané správou *IRI Report*. Typickou komunikáciou pred vytvorením spojenia je autentifikácia užívateľa.
  - **IRI Begin** - Správa *IRI Begin* reprezentuje detekciu začiatku spojenia. Príkladom môže byť priradenie IP adresy na koncovom zariadení. Priradením adresy na rozhranie je umožnená komunikácia daným rozhraním, čo sa dá interpretovať ako začiatok spojenia.
  - **IRI Continue** - Aj po začatí spojenia a odoslaní správy *IRI Begin* je možné detegovať nové informácie. Tieto nové informácie sú posielané správou *IRI Continue*.
  - **IRI End** - Správa *IRI End* je posielaná v prípade aktívneho spojenia (bola odoslaná správa IRI Begin) a detekciou jeho ukončenia. Tým pádom ďalšia komunikácia týmto spojením už nie je možná. V prípade znovu vytvorenia spojenia sa musí znovu odoslať IRI Begin.
- **Popis IRI správy** - Stručný popis udalosti, ktorá bola na sieti detegovaná.
- **Zoznam identifikátorov** - Hodnota položky je zoznam, ktorý obsahuje názov a hodnotu jednotlivých identifikátorov.

Príklad správy IRI je možné vidieť na obrázku 2.4:



Obrázok 2.4: Formát správ IRI

Kapitola 3

Popis protokolov architektúry TCP/IP

Kvôli zložitosti sieťovej komunikácie sú jednotlivé protokoly používané v sieťach na základe svojej činnosti klasifikované do vrstiev. Tieto vrstvy majú vymedzený účel a sú hierarchicky usporiadané, pričom s výnimkou okrajových vrstiev poskytuje každá vrstva služby vyššej vrstve a využíva služby nižšej vrstvy. V rámci počítačovej siete Internet je využívaná klasifikácia do vrstiev na základe architektúry TCP/IP, ktorá je zobrazená na obrázku 3.1. Protokolov používaných v sieti Internet je veľa a preto je vhodné sa zamerať len na najvhodnejšie pre účely systému pre zákonné odpočúvanie. V rámci tejto práce sa zameriam na protokoly PPPoE, SLAAC a SMTP, ktoré spolu so stručným popisom architektúry TCP/IP popíšem v tejto kapitole.

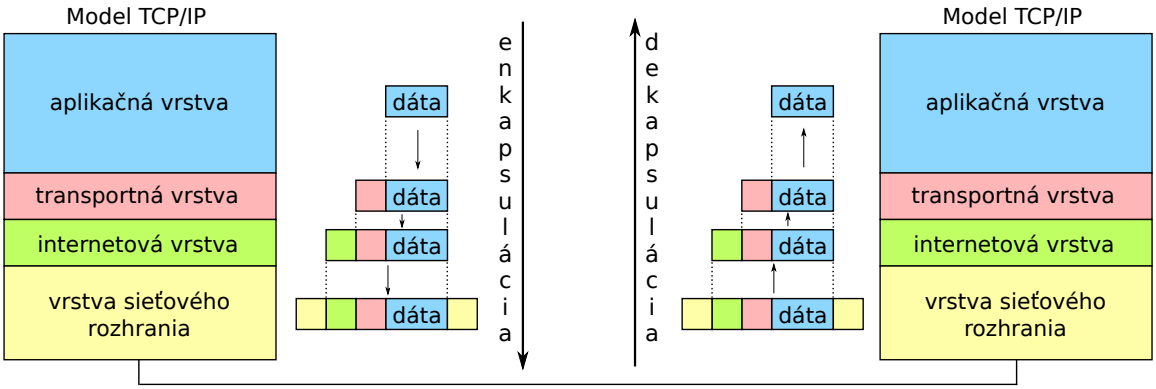
3.1 Popis architektúry TCP/IP z hľadiska identít

Každý protokol používaný v rámci počítačovej siete Internet je možné klasifikovať do niektorej vrstvy architektúry TCP/IP. Celkovo sa architektúra TCP/IP skladá z nasledujúcich štyroch vrstiev: vrstva sieťového rozhrania, internetová vrstva, transportná vrstva a aplikačná vrstva, ktoré sú znázornené na obrázku 3.1. Taktiež aj identifikátory, ktoré sú používané jednotlivými protokolmi je možné klasifikovať do týchto štyroch vrstiev.

Pri výmene dát medzi rôznymi sieťovými prvkami (vrátane koncových staníc) sa využíva enkapsulácia dát (obrázok 3.1). Enkapsulácia je proces, kedy pri odosielaní sú k samotným dátam postupne pridávané hlavičky jednotlivých vrstiev. Na vstupe sú dáta aplikácie, ktorá má záujem o prenos dát po sieti. Tieto dáta sú odoslané nižšej (transportnej) vrstve. Transportná vrstva pridá pred aplikačné dáta vlastnú hlavičku a pošle ďalšej vrstve. Tento proces sa opakuje až po vrstvu sieťového rozhrania, kde sa okrem hlavičky na koniec dát pridá aj pätička a celá, teraz už správa, sa odošle prostredníctvom média na iné zariadenie.

Zariadenie ktoré príjme správu prostredníctvom média od iného zariadenia spustí proces s názvom dekapsulácia. Jedná sa o opačný proces k procesu enkapsulácia. Zariadenia postupne odstráni všetky vrstvy až mu ostanú len samotné dáta, ktoré odovzdá cieľovej aplikácii. Súčasťou hlavičiek, ktoré sú počas enkapsulácie k dátam pridávané, sú aj príslušné identifikátory týchto vrstiev. Na ich základe sa každé zariadenie dokáže rozhodnúť, ako s príslušnými dátami vynaloží.

Z hľadiska dynamickej identifikácie užívateľa a priraďovania identifikátorov, by sa všetky štyri vrstvy TCP/IP architektúry dali priradiť do troch kategórii: identifikátory pri-



Obrázok 3.1: Architektúra TCP/IP

radené od ISP (vrstva sieťového rozhrania a internetová vrstva), identifikátory priradené operačným systémom (transportná vrstva) a identifikátory priradené aplikáciou (aplikačná vrstva). Popis jednotlivých kategórii je nasledujúci:

- **identifikátory priradené od ISP** - Do tejto kategórie patrí vrstva sieťového rozhrania a internetová vrstva.
  - Pod pojmom identifikátor vrstvy sieťového rozhrania môže byť myslené prihlasovacie meno použité pri autentifikácii užívateľa na vrstve sieťového rozhrania. V rámci tejto práce sa jedná o protokol PPPoE.
  - Identifikátory sieťového rozhrania sa dajú považovať za najdôležitejšie identifikátory, pretože sú z dôvodu enkapsulácie používané pri každej komunikácii v sieti (nezávisle na vyššej vrstve). Najznámejším príkladom je IP adresa, ktorá môže byť vo verzii IPv4 alebo IPv6. IP adresa je priradená na koncové zariadenie a jej hodnota je obsiahnutá v každej prenášanej správe. Z hľadiska odpočúvania sa jedná o veľkú výhodu, pretože pre každú správu vieme jednoznačne určiť jej odosielateľa a prijímateľa. Protokoly obsiahnuté v tejto práci, ktoré sa zaoberajú prideľovaním IP adries sú PPPoE a SLAAC.
- **identifikátory priradené operačným systémom** - Identifikátormi priraďovanými *operačným systémom* (OS) sú porty, ktoré sa nachádzajú na transportnej vrstve. Port je číselný údaj, na základe ktorého OS rozlišuje ktorej aplikácii má prijatú správu zo siete odovzdať. Tento číselný údaj môže byť náhodne pridelený OS alebo môže aplikácia o niektorý konkrétny port sama požiadať. Porty sú jediné identifikátory transportnej vrstvy architektúry TCP/IP. Vzhľadom k tomu, že sa porty neprideľujú žiadnym sieťovým protokolom, je ich priradenie určitej aplikácii možné len na aplikačnej úrovni požiadaním o konkrétne číslo portu.
- **identifikátory priradené aplikáciou** - Použitie identifikátorov priraďovaných aplikáciou je vyhradené len pre daný aplikačný protokol. Týchto identifikátorov ako aj samotných aplikačných protokolov je veľa. Je preto nutné vhodne vybrať zopár aplikačných protokolov, ktoré budú v projekte Sec6Net podporované. Zvolil som si protokol SMTP pre posielanie pošty. Príkladom identifikátoru aplikačnej vrstvy je e-mailová adresa.

3.2 Point-to-Point Protocol

Point-to-Point Protocol (PPP) [27] je protokol vrstvy sieťového rozhrania architektúry TCP/IP, ktorý slúži na vytvorenie priameho spojenia medzi dvomi zariadeniami pripojených v sieti prostredníctvom synchrónnych alebo asynchrónnych liniek. Priame spojenie umožňuje komunikáciu len medzi dvomi zariadeniami, preto v prípade ak chce zariadenie komunikovať s viacero zariadeniami prostredníctvom protokolu PPP musí zariadenie nadviazať spojenie s každým zvlášť. Toto priame spojenie umožňuje obojstrannú komunikáciu medzi danými zariadeniami, ktorá môže byť podmienená úspešnou autentifikáciou. Prostredníctvom protokolu PPP je umožnená komunikácia protokolov vyšších vrstiev (napríklad IP), ktoré vyžadujú protokoly nižších vrstiev pre správnu komunikáciu.

Spojenie PPP je vytvárané v troch fázach:

1. **Nadviazanie spojenia** - V prvej fáze sa pomocou protokolu *Link Control Protocol* (LCP) [27] naviaže komunikácia a zariadenia sa dohodnú na parametroch vytváraného spojenia. Medzi tieto parametre patrí napríklad maximálna veľkosť prenášaných správ a protokol pre autentifikáciu. Protokol LCP taktiež slúži k udržiavaniu spojenia pomocou pravidelnej výmeny správ typu „keepalive“.

2. **Autentifikácia** - Na základe dohodnutého protokolu autentifikácie v prvej fáze prebehne autentifikácia. Protokolov pre autentifikáciu existuje viacero. Spomeniem len základné PAP [13], CHAP [13], EAP [4] a MS-CHAPv2 [30]. Medzi najbežnejšie používané protokoly pre prenos týchto dát patrí PAP alebo CHAP, ktoré využívajú autentifikáciu pomocou mena a hesla. V závislosti na konfigurácii daných zariadení sa môžu autentifikovať obe zariadenia navzájom alebo len jedno zariadenie u druhého zariadenia.

3. **Spojenie ku sieťovej vrstve** - Prostredníctvom skupiny protokolov *Network Control Protocol* (NCP) [27] sa nadväzuje spojenie ku konkrétnym protokolom sieťovej vrstvy architektúry TCP/IP. Do tejto skupiny protokolov patrí napr. IPCP pre IPv4, IPv6CP pre IPv6, IPXCP pre IPX a ATCP pre AppleTalk. V niektorých konfiguráciách je zároveň možné priradiť identifikátor ku protokolu sieťovej vrstvy architektúry TCP/IP. Príkladom je priradenie IPv4 adresy protokolom IPCP [16]. Protokoly patriace do skupiny NCP vytvárajú tieto spojenia nezávisle, čo umožňuje na rozdiel od protokolu LCP vytvorenie viacerých spojení ku protokolom sieťovej vrstvy.

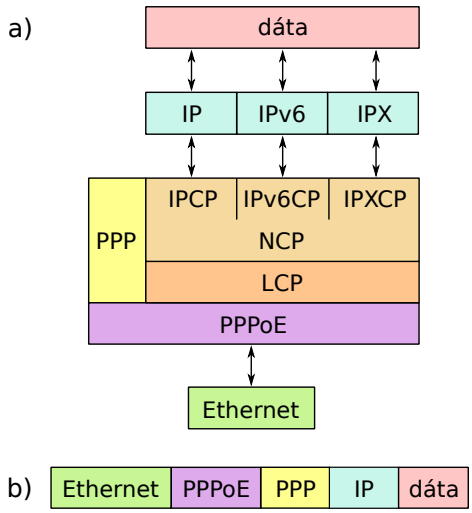
Pomocou protokolu PPP ako celku (vrátane všetkých využívaných protokolov) sa prenášajú dva rôzne typy identifikátorov. Prvým typom je prihlasovacie meno (PPP login) používané pri autentifikácii pomocou protokolov PAP alebo CHAP. Jedná sa o identifikátor vrstvy sieťového rozhrania architektúry TCP/IP(viď kapitola 3.1). V prípade, že je prihlasovacie meno v rámci siete unikátne, môže byť použité ako jedinečný identifikátor pre dohľadanie daného zariadenia v sieti, alebo užívateľa ktorému daný PPP login patrí. Druhým typom identifikátoru je identifikátor protokolu sieťovej vrstvy architektúry TCP/IP, ktorý pri niektorých konfiguráciách môže byť priradený niektorým z protokolov NCP. Z tohto dôvodu je vhodné sa týmto protokolom v rámci dynamickej identifikácie zaoberať.

3.2.1 Point-to-Point Protocol over Ethernet

Protokol *Point-to-Point Protocol over Ethernet* (PPPoE) [14] je rozšírením protokolu PPP. Protokol PPPoE je určený pre vytváranie spojení protokolu PPP nad protokolom Ethernet.

To je umožnené pridaním hlavičky PPPoE k správam protokolu PPP (obrázok 3.2). Jedným z informácií v tejto pridanej hlavičke je číslo spojenia. Pomocou tohto identifikátoru je možné rozlišovať viacero nadviazaných PPP spojení u jedného sieťového zariadenia.

Jednotlivé vrstvy protokolu PPP s rozšírením PPPoE a zoznam hlavičiek, ktoré paket pri prenose bude mať je možné vidieť na obrázku 3.2.



Obrázok 3.2: a)Jednotlivé vrstvy architektúry TCP/IP, b)hlavičky protokolu PPPoE

Pri nasadzovaní protokolu PPPoE do siete, sa do sieťovej topológie pridajú dva typy zariadení (obrázok 3.3). Zariadenie *Broadband Remote Access Server* (BRAS) je sieťové zariadenie pod správou ISP, ktoré slúži na smerovanie dátového toku typicky zo alebo na zariadenie *Digital Subscriber Line Access Multiplexers* (DSLAM) a zároveň ukončuje veľké množstvo point-to-point spojení. Zariadenie DSLAM je taktiež sieťové zariadenie pod správou ISP, na ktoré sú pripojení zákazníci pomocou technológie *Digital Subscriber Line* (DSL) a slúži na koncentráciu týchto spojení. Technológia DSL umožňuje koncovým zákazníkom pomocou k tomu určeného modemu využívať svoje telefónne linky na pripojenie k Internetu. Jednotliví zákazníci tak pomocou svojich modemov vytvárajú PPPoE spojenie so zariadením BRAS. Zapojenie zariadenia BRAS a DSLAM spolu s ukážkovou topológiou je možné vidieť na obrázku 3.3.
Postup pri nadviazaní PPP spojenia je nasledujúci:

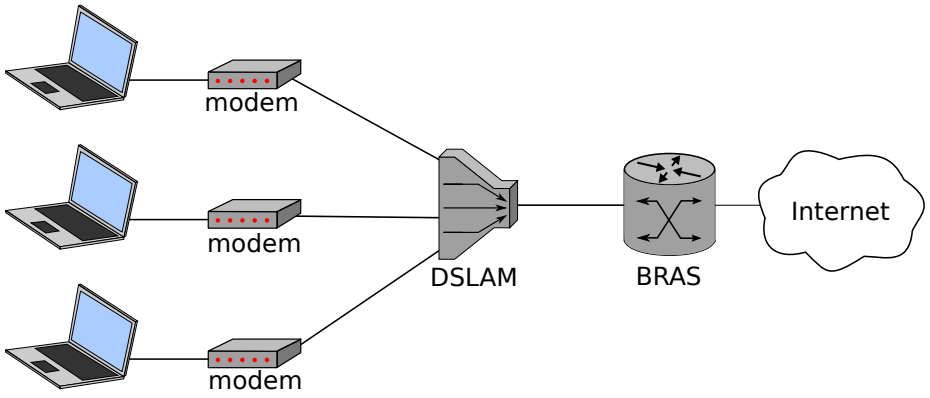
1. Najprv sa pomocou protokolu PPP (LCP) klient dohodne s BRASom na spôsobe prenosu dát a prípadnej autentifikácii.

2. Keď je autentifikácia BRASom vyžadovaná, klient sa autentifikuje. V prípade neúspešnej autentifikácie sa spojenie PPPoE okamžite ukončí.

3. Klient môže požiadať BRAS o pridelenie IP adresy. Tento krok sa odlišuje pre IPv4 a IPv6:

   (a) **IPv4** - Pomocou protokolu PPP *IP Control Protocol* (IPCP) BRAS povie klientovi svoju IP adresu a prípadne aj IP adresu pridelenú klientovi.

   (b) **IPv6** - Pomocou protokolu PPP *IPv6 Control Protocol* (IPv6CP) si BRAS aj klient navzájom odovzdajú svoje lokálne IPv6 adresy.



Obrázok 3.3: Ukážková topológia protokolu PPPoE spolu so zapojením zariadení BRAS a DSLAM

Po týchto krokoch je PPPoE spojenie medzi BRASom a klientom úspešne nadviazané. V prípade, že klient nezískal IPv4 adresu a vyžaduje ju resp. IPv6 adresu, tak práve teraz môže využiť protokoly vyšších vrstiev. Adresy získané z vyšších vrstiev sú potom nezávislé na protokole PPPoE a preto aj pri ukončení PPPoE spojenia zostávajú aj naďalej platné.

Posledným krokom v rámci PPPoE relácie je ukončenie spojenia. Vlastnosťou protokolu PPPoE je, že obidve strany periodicky odosielajú správy pre udržanie spojenia (*keepalive*), na ktoré si vzájomne odpovedajú. V prípade, že klientovi alebo BRASu nepríde odpoveď na *keepalive* správu, odošle správu typu *PPPoE Active Discovery Termination* (PADT) a ukončí spojenie. Rovnaká správa sa odosiela aj pri bežnom (vyžadovanom) ukončení spojenia alebo pri ukončení spojenia z dôvodu neúspešnej autentifikácie a pod. Príklad správ vymieňaných pri nadviazaní spojenia protokolom PPPoE sa nachádza v dodatku B.

Správy rozšírené hlavičkou PPPoE sú ďalej odosielané protokolom Ethernet. Do Ethernetovej hlavičky je zapísaný typ zapuzdrených dát *PPPoE Discovery Stage* alebo *PPPoE Session Stage* v závislosti na fáze spojenia PPPoE. Ak sa spojenie ešte len vytvára alebo je vytvorené a ukončuje sa, jedná sa o typ dát *PPPoE Discovery Stage*. V opačnom prípade, ak je spojenie nadviazané, tak sa jedná o typ *PPPoE Session Stage*.

Aby bolo možné prenášať správy PPP nad protokolom PPPoE je nutné pred samotným prenosom vytvoriť PPPoE spojenie a po ukončení PPP spojenia toto spojenie taktiež ukončiť. K tomu protokol PPPoE definuje správy *PPPoE Active Directory* (Initiation, Offer, Request, Session-confirmation a Terminate). Pomocou týchto správ sieťové zariadenie vyhľadá na zdieľanom médiu iné sieťové zariadenie, s ktorým následne naviaže spojenie. Typickým príkladom zariadenia, s ktorým sa nadväzuje spojenie, je zariadenie BRAS.

3.3 Stateless Address Autoconfiguration

Bezstavová autokonfigurácia adries - Stateless Address Autoconfiguration (SLAAC) [28] slúži pre automatické prideľovanie IPv6 adries koncovej stanici. Komunikácia prebieha medzi klientom a smerovačom pomocou protokolu *Internet Control Message Protocol for the Internet Protocol Version 6* (ICMPv6) [5]. Na rozdiel od iných protokolov pre prideľovanie adries však nie je klientovi pridelená adresa priamo SLAAC serverom, prípadne iným centrálnym prvkom, ale sú mu poskytnuté konfiguračné údaje, pomocou ktorých si stanica sama IPv6 adresu vytvorí a pridelí. Medzi tieto konfiguračné údaje patrí hlavne prefix

danej siete (horná časť IPv6 adresy) a doba platnosti tohto prefixu. Pridelené IPv6 adresy sú zároveň identifikátorom sieťovej vrstvy. Celkový postup pre pridelenie IPv6 adresy prebieha v nasledujúcich krokoch:

1. Klient si najprv vygeneruje *lokálnu IPv6 adresu* tak, že si sám zvolí identifikátor rozhrania a hornú časť nastaví na prefix fe80::/64.

   Identifikátor rozhrania je prostriedok na rozlíšenie koncových staníc v sieti s protokolom IPv6. Jedná sa o spodnú časť (dolných 64 bitov) IPv6 adresy, pričom táto hodnota je nezávislá na pridelenom prefixe (horných 64 bitov). Hodnota tohto identifikátoru môže byť odvodená z MAC adresy, nastavená manuálne alebo náhodne vygenerovaná.

   Po zvolení identifikátora si na jeho základe odvodí aj tzv. *solicited-node multicast* adresu a prihlási sa do tejto skupiny. Uvedená skupinová adresa je vytvorená spojením prefixu ff02::1:ff00:0/104 so spodnými 24 bitmi IPv6 adresy. Dôvod pre vytvorenie adresy a prihlásenie klienta do tejto skupiny je vo využití mechanizmu *Neighbor Discovery* [22], kedy sa klient nepýta všetkých užívateľov na lokálnej sieti, ale len užívateľov prihlásených do danej skupiny, či niektorý z nich už nepoužíva takto vygenerovanú adresu. Podrobnejší popis protokolu *Neighbor Discovery* je uvedený nižšie.

2. Aby si klient mohol prideliť aj globálnu IPv6 adresu, musí najprv poznať prefix podsiete, v ktorej sa nachádza. O túto informáciu môže požiadať najbližší smerovač, prípadne viacero smerovačov. Táto informácia sa prenáša opäť pomocou protokolu *Neighbor Discovery*. Získanie informácie o prefixe prebieha v nasledujúcich krokoch:

   (a) Klient odošle žiadosť o konfiguračné údaje pomocou správy *Router Solicitation*. Táto správa je odoslaná na skupinovú (multicast) adresu ff02::2, kde sú zaradené všetky smerovače v danej podsieti.

   (b) Smerovač(e) odpovedajú správou *Router Advertisement* obsahujúcu informácie o prefixu podsiete a dobe jeho platnosti.

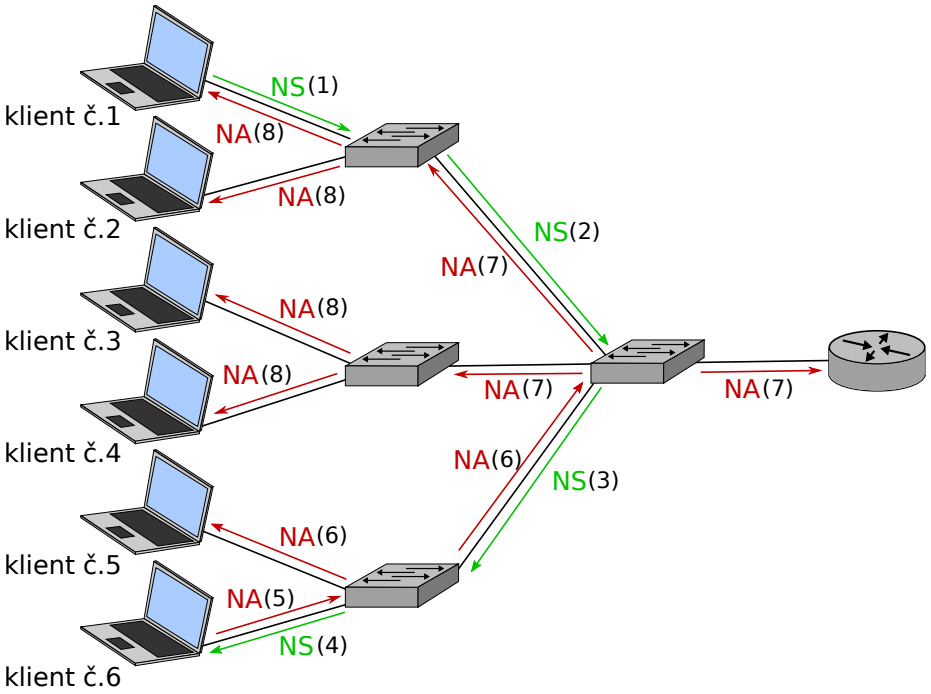
3. Akonáhle má klient k dispozícii prefix podsiete, vygeneruje si globálnu adresu buď na základe lokálnej adresy - zámenou prefixu lokálnej adresy za prefix danej podsiete alebo úplne nezávisle na lokálnej adrese (napr. náhodne). V prvom prípade sa identifikátor rozhrania (dolná časť) globálnej adresy zhoduje s identifikátorom rozhrania lokálnej adresy. V druhom prípade sa identifikátory rozhrania lokálnej a globálnej adresy líšia. Pre takto vytvorenú globálnu adresu je nutné sa opäť prihlásiť do skupiny odvodenej z *solicited-node multicast* adresy a overiť, či už nie je používaná iným klientom siete.

Overenie lokálnej a globálnej IPv6 adresy (*Duplicate Address Detection* - DAD) prebieha pomocou protokolu *Neighbor Discovery* v nasledujúcich krokoch:

1. Klient vloží vygenerovanú adresu do správy *Neighbor Solicitation (NS)* a túto správu odošle na skupinovú (multicast) adresu odvodenú od vygenerovanej IPv6 adresy.

2. Pokiaľ klientovi do určitého časového intervalu nepríde žiadna odpoveď v podobe správy *Neighbor Advertisement (NA)* oznamujúca, že vygenerovaná adresa je už používaná, považuje klient adresu za unikátnu a začne ju používať. Správa *Neighbor Advertisement* je odosielaná všetkým staniciam na lokálnej sieti.

Na obrázku 3.4 je zobrazený princíp činnosti mechanizmu DAD. Klient č. 1 kontroluje duplicitu svojej vygenerovanej adresy, pričom zbytočne neobťažuje každého klienta v lokálnej sieti. Klient č. 6 má zhodou okolností vygenerovanú tú istú IPv6 adresu a tak odpovedá správou NA.



Obrázok 3.4: Princíp činnosti protokolu DAD (čísla reprezentujú poradie správ)

Uvedeným spôsobom si môže klient vygenerovať aj viacero globálnych IPv6 adries. Doba platnosti pridelenej globálnej adresy je obmedzená na hodnotu odoslanej v rámci správy *Router Advertisement (RA)* v položke *Valid lifetime*. Klient si však platnosť pridelenej adresy predlžuje po dobu, kedy smerovač pravidelne posiela správu *RA*.

Mechanizmus, ktorým by klient oznámil ostatným sieťovým uzlom, že už svoju adresu nebude používať, nie je bohužiaľ v protokole definovaný. Pre účely monitorovania siete je však potrebné vedieť túto informáciu, kedy stanica už danú adresu nepoužíva. Jedným zo spôsobov riešenia tohto problému spočíva vo využití princípu multicastu, kedy je stanica prihlásená do skupiny iba tak dlho, ak to má pre ňu význam. Pre protokol SLAAC to teda znamená, že klient je prihlásený v *solicited node multicast* skupine len tak dlho, pokiaľ využíva aspoň jednu adresu patriacu do danej skupiny. Inými slovami, za koniec platnosti IPv6 adresy sa dá považovať okamih, kedy už klient nie je prihlásený do multicastovej adresy z nej odvodenej.

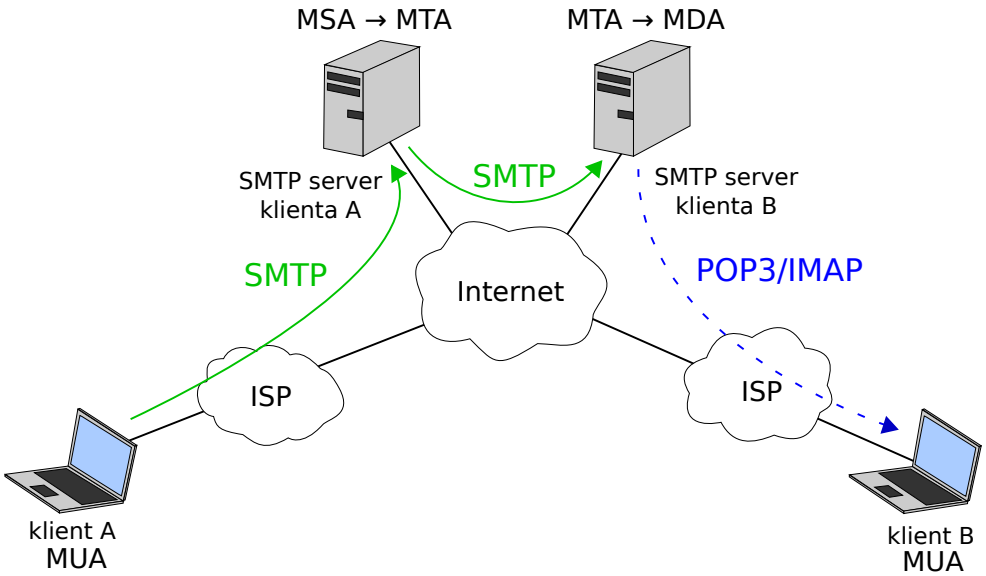
Pre detekciu či je klient stále prihlásený do mutlicastovej adresy odvodenej z jeho IPv6 adresy je možné využiť správanie smerovačov. Tie sa sami automaticky pýtajú na všetky multicastové skupiny a zisťujú, či je v nich prihlásená aspoň jedna stanica. Tieto správy sú posielané na skupinovú adresu ff02::16 reprezentujúcu všetky smerovače podporujúce multicast. Trvalým prihlásením sa do tejto skupiny a následným odpočúvaním zasielaných správ je možné takto nepriamo odvodiť platnosť jednotlivých IPv6 adries.

3.4 Simple Mail Transfer Protocol

Protokol *Simple Mail Transfer Protocol* (SMTP) [10] je aplikačný protokol a zároveň internetový štandard na posielanie elektronickej pošty (e-mail). Slúži pre prenos pošty medzi odosielateľom a adresátom, odosielateľom a jeho poštovým serverom alebo medzi dvomi rôznymi poštovými servermi. Aj keď je možné posielať poštu priamo adresátovi, v drvivej väčšine sa pošta posiela na poštový server adresáta.

Architektúra protokolu SMTP je znázornená na obrázku 3.5, ktorej popis je nasledujúci:

- **Mail User Agent** (MUA) - Program na správu e-mailov (minimálne príjem a odosielanie), ktorý je umiestnený na koncových staniciach.
- **Mail Submission Agent** (MSA) - Proces na serveri, ktorý sa stará o príjem e-mailov od MUA. Medzinárodná organizácia IANA, ktorá sa okrem iného stará o priraďovanie čísla portov na transportnej vrstve aplikačným protokolom, priradila procesu MSA port 587 [2]. Z historických dôvodov ale proces MSA môže počúvať aj na porte číslo 25.
- **Mail Transfer Agent** (MTA) - Proces na serveri, ktorý sa stará o prenos doručeného e-mailu na iný (cieľový) SMTP server. Procesu MTA bol organizáciou IANA priradený port 25.
- **Mail Delivery Agent** (MDA) - Proces na serveri, ktorý sa stará o doručenie e-mailu aplikácii MUA na strane prijímateľa. Prenos e-mailu medzi procesom MDA a MUA však už nie je zabezpečovaný protokolom SMTP, ale k tomu vyhradenými protokolmi (POP3, IMAP).



Obrázok 3.5: Zobrazenie architektúry SMTP protokolu

Činnosť jednotlivých blokov architektúry pri posielaní e-mailu je nasledujúca:

- Odosielateľ správy pomocou programu MUA napíše správu a odošle pomocou protokolu SMTP na svoj poštový server.

- Na poštovom serveri odosielateľa beží proces MSA, ktorý sa stará o prijatie správy od MUA.

- Na tom istom serveri ako sa nachádza proces MSA sa môže nachádzať aj proces MTA starajúci sa o prenos e-mailu na ďalší poštový server protokolom SMTP. Typicky je ďalším poštovým serverom server prijímateľa, na ktorom sa e-mail príjme taktiež MTA procesom. Nepoužije sa MSA proces, pretože ten sa stará len o príjem e-mailu od MUA. V prípade, že je poštový server odosielateľa a prijímateľa rovnaký, nie je nutné využiť proces MTA.

- Po prijatí e-mailu na poštový server prijímateľa je e-mail lokálne uložený. Po pripojení prijímateľa e-mailu (proces MUA) na server sa spustí proces *Mail Delivery Agent* - MDA, ktorý sa ďalej postará o doručenie adresátovi.

SMTP ku svojej činnosti využíva výhradne transportný protokol TCP. Protokol TCP sám zabezpečuje spoľahlivý prenos dát medzi oboma účastníkmi tak, aby nedošlo k strate paketov alebo k zmene ich poradia. Tým pádom sa protokol SMTP nemusí starať o zabezpečenie spoľahlivého prenosu.

Komunikácia protokolom SMTP funguje na princípe príkaz - odpoveď. Každý príkaz alebo odpoveď je posielaný v textovom formáte a ukončený koncom riadku (znakmi CRLF). Klient odošle príkazom parameter komunikácie (napr. adresu príjemcu) a server pomocou číselných kódov reaguje na daný príkaz (napr. kódom 250 - potvrdenie parametru). Súčasťou ktorejkoľvek odpovede zo strany servera môže okrem čísla kódu byť ľubovoľný text, oddelený od kódu medzerou a ukončený koncom riadku. Tento text je voliteľný a slúži na spresnenie významu danej správy, napr. doména serveru pri pripojení na server.

Pôvodne normou definované príkazy [10], sa rozšírili o tzv. *Extended SMTP* (ESMTP) [11] príkazy. Tieto nové príkazy zaviedli napr. príkaz *AUTH*, ktorý reprezentuje autentifikáciu iniciátora prenosu. V dnešnej dobe sa dá povedať, že minimálne z dôvodu autentifikácie, všetci klienti a servery podporujú a zároveň preferujú tieto ESMTP príkazy, preto vo zvyšku tejto správy budem vždy predpokladať využitie týchto príkazov.

Protokol SMTP je tak navrhnutý, že formát správ posielaných medzi užívateľom a serverom alebo medzi dvomi servermi je úplne rovnaký. Iná je len forma overovania odosielateľa e-mailu.

1. **SMTP server odosielateľa** overuje, či užívateľ posiela e-mail zo správnej zdrojovej adresy v rámci danej(svojej) domény. V prípade domény „domain.com“ tým je zaistené, aby užívateľ s adresou „user@domain.com“ nemohol odoslať e-mail z adresy „administrator@domain.com“. Server môže odosielateľa overiť buď na základe autentifikácie pomocou prihlasovacieho mena a hesla alebo podľa zdrojovej IP adresy.

2. **Zvyšné SMTP servery** už neoverujú či užívateľ použil správnu zdrojovú e-mailovú adresu v rámci domény. Užívatelia sa totiž overujú len na svojom vlastnom serveri. Zvyšné SMTP servery overujú len to, či SMTP server odosielateľa má právo posielať e-maily z príslušnej domény. K overeniu SMTP serveru slúži protokol *Domain Name System* (DNS) záznam typu SPF [29], ktorý by mala obsahovať každá doména. V zázname SPF je uvedené, z ktorých IP adries je možné odosielať e-maily z danej domény. Jedná sa o IP adresy SMTP serverov, nie užívateľov. V prípade adresy „user@domain.com“ tak každý server skontroluje či zdrojová IP adresa je obsiahnutá v SRT zázname domény „domain.com“.

Zjednodušený postup pri prenášaní správy je nasledujúci (celý postup je možné nájsť v prílohe D):

1. Klient so serverom naviažu komunikáciu a dohodnú sa na metóde autentifikácie.
2. Klient sa v prípade nutnosti podľa zvolenej metódy autentifikuje.
3. Nasleduje odoslanie identifikátorov (e-mailových adries) odosielateľa a príjemcu (resp. príjemcov).
4. Klient odošle na server odosielanú správu.
5. Posledným krokom je ukončenie spojenia, ktoré inicializuje klient.

Samotný obsah odosielanej správy je kódovaný vo formáte *Internet Message Format* (IMF) [25], ktorý okrem čistého textu a predmetu správy obsahuje aj hlavičku e-mailu. Obsahom hlavičky je napr. položka odosielateľ, adresát, ďalej čas odoslania, unikátny identifikátor danej správy atď. Unikátny identifikátor správy je zároveň jediný identifikátor, ktorá sa zisťuje z obsahu správy. Ďalšie identifikátory protokolu SMTP sú e-mail odosielateľa a email všetkých prijímateľov. Avšak tieto identifikátory nie sú získavané z obsahu správy, ale z príkazov odosielateľa.

Obrázok 3.6: Príklad komunikácie medzi klientom a serverom

V prípade, že je správa odosielaná prostredníctvom poštových serverov, je nutné aby servery vedeli kam majú správu poslať ďalej. K tomu využívajú protokol DNS [17], konkrétne záznam typu MX [18]. Server si z e-mailu adresáta zistí doménu (časť e-mailu za znakom „@“) v ktorej vyhľadá MX záznam. Ten obsahuje názov servera, na ktorý sa daná správa odošle. To ako si adresát stiahne poštu zo svojho poštového serveru už nerieši protokol SMTP, ale iné protokoly (napr. POP3 [20] a IMAP [6]).

Okrem nešifrovanej komunikácie umožňuje protokol SMTP aj šifrovanú komunikáciu [8]. Prvou možnosťou šifrovania je na aplikačnej úrovni. Jej podporu dáva server najavo v odpovedi na príkaz EHLO, ktorá obsahuje zoznam podporovaných rozširujúcich funkcií. Ak sa v tomto zozname nachádza parameter STARTTLS, je šifrovaná komunikácia podporovaná. Ak klient šifrovanú komunikáciu podporuje a preferuje, namiesto príkazu AUTH odošle príkaz STARTTLS. Server na daný príkaz odpovie kódom 220 (*Go ahead*). Všetka nasledujúca komunikácia bude prebiehať šifrovane a ukončenie spojenia je možné detegovať len na úrovni ukončenia TCP spojenia.

Ďalšou možnosťou použitia šifrovanej komunikácie je pomocou medzivrstvy umiestnenej medzi transportným protokolom TCP a aplikačným SMTP. Jedná sa o medzivrstvu SSL/TLS, ktorej úlohou je zabezpečiť šifrovaný prenos dát bez nutnosti zmeny správ aplikačného protokolu. Hlavnou výhodou tohto riešenia je, že celá komunikácia od vytvorenia spojenia až po ukončenie je šifrovaná. Tým pádom nie je vôbec možné zistiť či sa jedná o komunikáciu protokolom SMTP alebo ktorýmkoľvek iným. Na základe faktu, že nad TCP vrstvou sa nenachádza protokol SMTP ale SSL/TLS je použité aj iné číslo portu. Organizáciou IANA bolo komunikácii cez SSL/TLS priradené číslo 465.

Kapitola 4

Tvorba správ IRI

Obsahom tejto kapitoly je analýza protokolov, ktoré boli popísané v kapitole 3, za účelom detegovať priradenie, začatie a ukončenie používania identifikátorov príslušných protokolov. Súčasťou analýzy je návrh modulov do modulárnej IRI-IIF (kapitola 2.2), ktorých náplňou je generovanie správ IRI-IIF na základe udalostí spojených s identifikátormi príslušných protokolov.

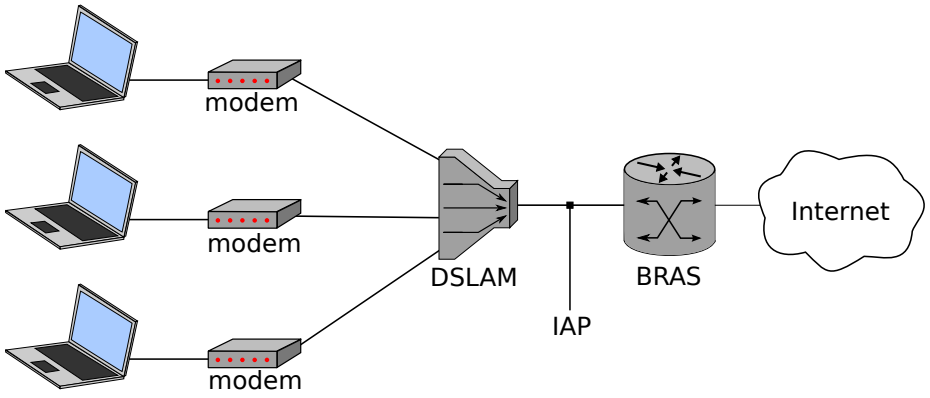
4.1 Point-to-Point Protocol over Ethernet

Modul IRI-IIF analyzuje správy protokolu PPPoE a na ich základe si udržuje tabuľku aktívnych spojení spolu s pridelenými IP adresami a prihlasovacím menom. Stavový diagram IRI-IIF je uvedený na obrázku 4.2. Každá koncová stanica prechádza pri nadväzovaní PPPoE spojenia nasledujúcimi stavmi:

- **Inicializácia** - stanica nemá nadviazané žiadne PPPoE spojenie ani pridelenú IPv4 alebo IPv6 adresu
- **Spojenie nadviazané** - stanica nadviazala spojenie s BRASom
- **Overenie autentifikácie** - stanica odoslala prihlasovacie údaje BRASu a prebieha ich overovanie
- **Spojenie bez adresy** - stanica nadviazala spojenie s BRASom, ale ešte nezískala IP adresu
- **Vyžadovanie adresy** - stanica žiada BRAS o pridelenie IPv4 alebo IPv6 adresy
- **Adresa pridelená** - stanici bola pridelená IPv4 alebo IPv6 adresa (popr. oboje)

Topológia spolu s predpokladaným umiestnením *IRI-sondy* je zobrazená na nasledujúcom obrázku 4.1.
Činnosť bloku IRI-IIF sa riadi podľa nasledujúcich pravidiel:

1. Počiatočným stavom je *Inicializácia*, v ktorom koncová stanica ešte nenadviazala spojenie a preto ani nemá pridelenú žiadnu adresu.
2. Po prijatí správy *PPPoE Active Discovery Session confirmation (PADS)* sa deteguje úspešné nadviazanie spojenia medzi klientom a BRASom, odošle sa správa *IRI Report* a vykoná sa prechod do stavu *Spojenie nadviazané*.



Obrázok 4.1: Ukážková topológia protokolu PPPoE s predpokladaným umiestnením IRI-sondy

3. Následne sa BRAS a klient dohadujú na spôsobe autentifikácie a niektorých ďalších parametroch daného PPPoE spojenia. Keď sa obe strany navzájom nedohodnú, odošle niektorá zo strán správu *PPPoE Active Discovery Session termination (PADT)*. Následkom bude okamžité ukončenie spojenia, v rámci modulu sa odošle správa *IRI Report* a vykoná sa prechod späť do stavu *Inicializácia*.

4. Pokiaľ BRAS vyžaduje autentifikáciu (viď predchádzajúci bod), musí mu klient odoslať požadované informácie. Zachytením správy *Authentication request* (*PPP PAP Authenticate-Request* pre PAP, *PPP CHAP Response* pre CHAP) získa modul prihlasovacie meno klienta. Následne sa prihlasovacie meno odošle spoločne so správou *IRI Report* a prechádza do stavu *Overenie autentifikácie*.

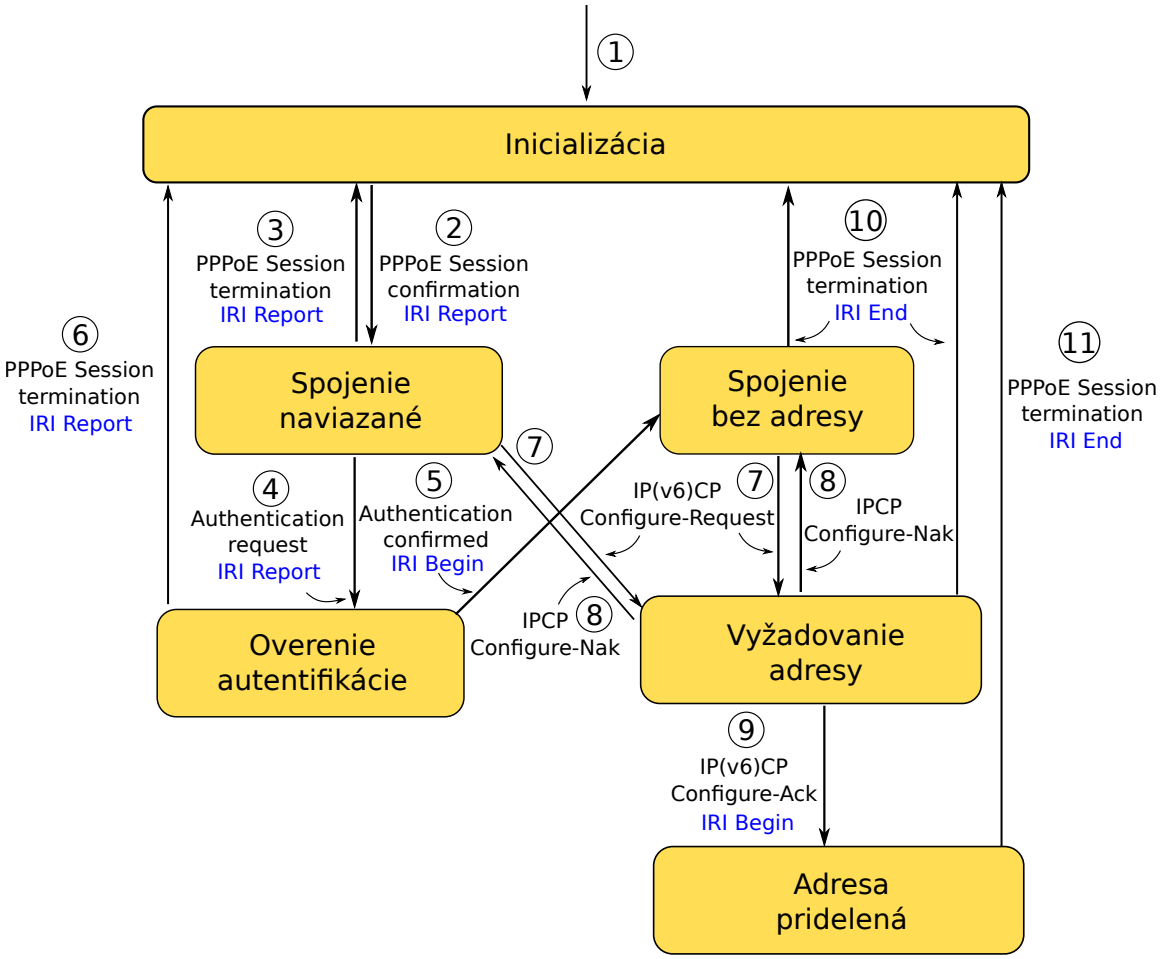
5. Úspešnú autentifikáciu modul deteguje prijatím správy *Authentication confirmed* (*PPP PAP Authenticate-ACK* pre PAP, *PPP CHAP Success* pre CHAP). Týmto spôsobom sa tiež potvrdí, že odoslané prihlasovacie údaje sú platné a modul pošle prvú správu typu *IRI Begin* informujúcu o úspešnej autentifikácii (zatiaľ bez pridelenej IP adresy). Modul nasledovne vykoná prechod do stavu *Spojenie bez adresy*.

6. Pokiaľ nebude autentifikácia úspešná (napr. klient zadá nesprávne prihlasovacie údaje), potom BRAS vynúti ukončenie spojenia pomocou správy *PADT* a modul vykoná prechod späť do stavu *Inicializácia*. Popri prechode medzi stavmi sa odošle správa *IRI Report*.

7. V prípade, že autentifikácia nebola vyžadovaná, alebo bola a dopadla úspešne, klient sa pokúsi získať IPv4 alebo IPv6 adresu. Proces prideľovania obidvoch týchto adries prebieha úplne nezávisle pomocou protokolov IPCP a IPv6CP.

    - Pokus o získanie IPv4 adresy modul deteguje prijatím správy *IPCP Configure-Request*. Súčasťou správy je aj IPv4 adresa, ktorú si klient žiada prideliť. Túto adresu si klient môže pamätať z posledného pripojenia v sieti alebo si ju môže vygenerovať náhodne. Výnimkou je situácia, kedy obsahom správy *IPCP Configure-Request* je IPv4 adresa nastavená na hodnotu 0.0.0.0. Tou dáva klient najavo, že adresu si nežiada prideliť pomocou protokolu PPPoE.

- Pokus o získanie IPv6 adresy modul deteguje prijatím správy *IPv6CP Configure-Request*. Na rozdiel od IPCP sa klient s BRASom nedohadujú na konkrétnej IP adrese, ale povedia si len identifikátory ich rozhrania. Pomocou identifikátorov rozhrania sa potom odvodí linková IPv6 adresa druhého uzla.



Obrázok 4.2: Stavový diagram IRI-IIF protokolu PPPoE

8. BRAS nemusí súhlasiť s adresou nastavenou v správe *IPCP Configure-Request* a tak odpovedá správou *IPCP Configure-Nak*, ktorá informuje o neúspešnom pridelení adresy a navrhuje klientovi inú IPv4 adresu. Detekciou tejto správy sa modul vracia do predchádzajúceho stavu a posiela správu *IRI Report*. O túto adresu môže klient pri ďalšej správe *IPCP Configure-Request* požiadať. V prípade, že BRAS adresy nepridelíuje, navrhne klientovi IPv4 adresu 0.0.0.0. Klient sa tak bude musieť pokúsiť získať IPv4 adresu iným protokolom (napr. DHCP).

9. Úspešné pridelenie adresy sa odlišuje pre protokoly IPCP (IPv4) a IPv6CP (IPv6). Spoločnou časťou je prechod modulu do stavu *Adresa pridelená*.

   - Úspešné pridelenie IPv4 adresy sa deteguje prijatím správy *IPCP Configure-Ack*. V prípade, že obsahom správy nie je adresa 0.0.0.0, modul generuje správu *IRI*

Begin obsahujúcu danú IPv4 adresu, resp. správu *IRI Continue* v závislosti či v rámci komunikácie bola odoslaná správa *IRI Begin*.

- Pri použití protokolu IPv6CP obidve strany vždy súhlasia s navrhovaným identifikátorom rozhrania a potvrdia si ich pomocou správy *IPv6CP Configure-Ack*. Modul rovnako ako BRAS si z klientovho identifikátoru odvodí adresu a tú odošle so správou *IRI Begin*, resp. *IRI Continue* v závislosti či v rámci komunikácie bola odoslaná správa *IRI Begin*.

10. V prípade ak sa klient nedohodne na IPv4 alebo IPv6 adrese nastane ukončenie spojenia. To modul deteguje zachytením správy PADT v stave *Spojenie bez adresy*, *Vyžadovanie adresy* alebo *Spojenie nadviazané*. V prípade ak sa klient úspešne autentifikoval a modul sa nachádzal v stave *Spojenie bez adresy* alebo *Vyžadovanie adresy* odošle modul správu *IRI End*, ktorá je spojená s odoslaním *IRI Begin* pri úspešnej autentifikácii.

11. Posledným možným krokom stavového konečného automatu zo stavu *Adresa pridelená* je ukončenie spojenia, ktoré modul deteguje rovnako ako v predchádzajúcich prípadoch zachytením správy *PADT*. Pri ukončení spojenia odošle modul ku každej predtým zaslanej správe *IRI Begin* príslušnú správu *IRI End*. Najviac teda môže poslať až tri *IRI End* správy odpovedajúce úspešnej autentifikácii, prideleniu IPv4 adresy a prideleniu IPv6 adresy. Pre každé PPPoE spojenie si teda musí modul uchovávať navyše informácie o tom, ktoré z IP adries boli pridelené.

4.2 Stateless Address Autoconfiguration

Blok IRI-IIF analyzuje ICMPv6 pakety a na ich základe udržuje aktuálnu tabuľku pridelených IPv6 adries spolu s ich stavom. Oproti iným protokolom pre prideľovanie IP adries ako sú napr. DHCP, RADIUS alebo PPPoE je SLAAC špecifický tým, že každá koncová stanica môže mať priradených niekoľko IPv6 adries súčasne. V rámci každej takto pridelenej adresy prebieha nezávisle proces overovania pomocou protokolu *Neighbor Discovery* (ND). Úlohou modulu IRI-IIF je sledovať tento protokol a uchovávať si tak stav pre každú jednotlivú IPv6 adresu (namiesto stavu celej klientskej stanice).

4.2.1 Analýza implementácie na rôznych OS

Ako bolo v kapitole 3.3 uvedené, vytvorenie a priradenie adresy protokolom SLAAC má na starosti koncová stanica na ktorej sa nachádza užívateľ a nie centrálny prvok ako napr. server. Je preto pred začatím tvorby stavového automatu nutné overiť správanie koncových staníc (resp. OS), či neexistujú medzi nimi rozdielne správania.

Kontrola správania rôznych OS bola zameraná na štúdium presného poradia posielaných správ pri využívaní protokolu ND. Táto štúdia prebiehala na rôznych OS, ktoré boli zvolené na základe ich kategórie (Windows, Linux, Unix) a používania. Cieľom bolo vytvoriť vzorku OS tak, aby z každej kategórie bol vybraný aspoň jeden OS a zároveň aby sa jednalo o najčastejšie používané systémy. Zoznam vybraných OS je zobrazený v tabuľke 4.1.

Vybrané OS boli následne v testovacom prostredí zapojené podľa niekoľkých rôznych schém zapojenia. Jednalo sa o zapojenia v „ideálnom“ stave (t.j. bez konfliktov v sieti) ako aj s konfliktami v sieti. Taktiež sa menilo nastavenie vybraných OS tak, aby raz boli

Skupina	Názov
Windows NT 5.1	Windows XP SP3
Windows NT 6.0	Windows Vista, Windows Vista SP SP2
Windows NT 6.1	Windows 7, Windows 7 SP1, Windows Server 2008 R2
Windows NT 6.2	Windows 8
Mac OS X	Mac OS X 10.6.2
Unix	FreeBSD 9.0, OpenBSD 5.0, Solaris 5.11
Linux, kernel 2.4.X	Debian 3.1
Linux, kernel 2.6.X	CentOS 6.2, Debian 6.0.4, Mandriva One 2011, Red Hat 5, Ubuntu 10.04 LTS
Linux, kernel 3.X	Fedora 16, Linux Mint 12, Ubuntu 11.10

Tabuľka 4.1: Zoznam analyzovaných operačných systémov

použité *privacy extension* adresy [21] a raz nie. Súčasťou tohto prostredia okrem samotných OS bol prepínač a smerovač, oba s podporou spracovania multicastu.

Po otestovaní vzoriek OS bolo detegovaných niekoľko rozdielov v ich správaní: poradie správ pri využití DAD nebolo rovnaké na všetkých operačných systémoch, niektoré systémy sa prihlasujú neskôr do multicastových skupín a existovali systémy, ktoré odosielali niektoré správy navyše. Okrem odlišností v implementácii protokolu SLAAC bolo zistené aj nedodržiavanie časového limitu pre odpoveď na výzvu smerovača ohľadom pýtania sa na aktívne multicast adresy. Aj keď smerovač vyžadoval odpoveď do 0.1s, tak niektoré odpovedi prišli aj po 0.8s.

Celkovo bolo analyzovaných 20 rôznych OS a vykonaných približne 300 testov, na základe ktorých vznikol vhodný podklad pre tvorbu stavového automatu.

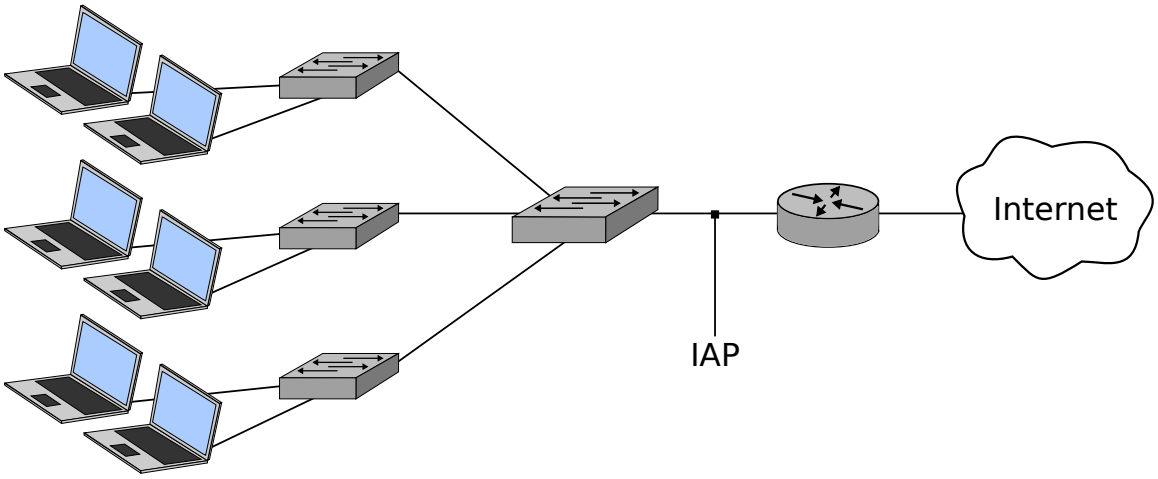
4.2.2 Tvorba automatu na základe analýzy OS

Podľa výsledkov analýzy bolo možné vytvoriť univerzálny stavový automat pokrývajúci analyzované OS. Stavový automat je zobrazený na obrázku 4.4 a má nasledujúce stavy:

- **Inicializácia** - adresa nie je pridelená žiadnemu klientovi (predvolený stav)
- **Kontrola unikátnosti** - stanica si vytvorila IPv6 adresu a zisťuje jej unikátnosť
- **Unikátna/pridelená adresa** - tento stav reprezentuje dve situácie: 1) na kontrolu duplicity adresy nikto neodpovedal a tak ju môže klient začať používať a 2) na kontrolu duplicity sa ozvala iná koncová stanica, ktorá danú adresu aktuálne používa
- **Kontrola multicast. skupiny** - smerovač zisťuje, či sa ešte v danej skupine nachádza nejaký klient

Topológia spolu s predpokladaným umiestnením *IRI-sondy* je zobrazená na nasledujúcom obrázku 4.3.

Všetky správy protokolu SLAAC sú posielané na skupinové adresy. Aby bol modul schopný tieto správy prijímať a analyzovať je nutné, aby bol v daných (multicastových) skupinách taktiež prihlásený. Najprv musí byť modul prihlásený v skupine ff02::16, do ktorej jednotliví klienti posielajú správy týkajúce sa prihlasovaní do skupín. Nasledovne,



Obrázok 4.3: Ukážková topológia protokolu SLAAC s predpokladaným umiestnením IRI-sondy

keď modul deteguje, že sa určitý klient prihlasuje do niektorej *solicited node multicastovej* skupiny, okamžite sa tam prihlási tiež. Len týmto spôsobom je modul schopný prijímať a analyzovať správy, ktoré sú uvedené v diagrame.

Dôležitá poznámka: Pokiaľ je striktne vyžadované, aby sa modul choval len pasívne t.j. nevkladal žiadne pakety do sledovanej siete (vrátane správ pre prihlásenie do multicastových skupín), potom je nutné zaistiť, aby modul získal prístup k uvedenej komunikácii iným spôsobom (napr. vhodným zapojením v rámci infraštruktúry ISP alebo konfiguráciou aktívnych prvkov.
Činnosť bloku IRI-IIF sa riadi podľa nasledujúcich pravidiel:

1. Počiatočným stavom je *Inicializácia*. V tomto stave nie je IPv6 adresa pridelená žiadnemu klientovi, presne povedané, modul IRI-IIF o takom pridelení zatiaľ nevie.

2. Prijatím správy *Neighbor Solicitation* modul deteguje situáciu, kedy si klient vygeneroval vlastnú IPv6 adresu a snaží sa o jej overenie. V súvislosti s touto udalosťou sa vykoná prechod do stavu *Kontrola unikátnosti*.

3. V rámci kontroly unikátnosti adresy môžu nastať dve situácie:

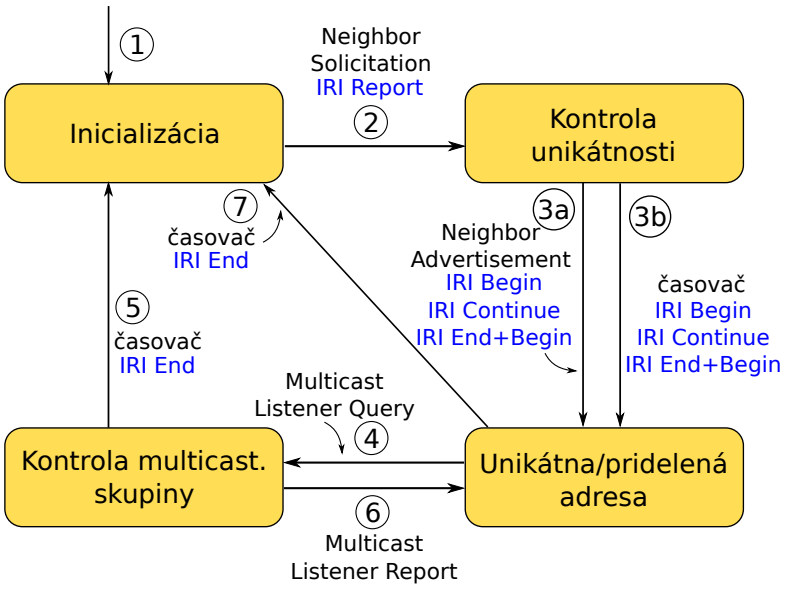
   (a) Ako odpoveď na *Neighbor Solicitation* príde správa *Neighbor Advertisement*, čo znamená, že danú adresu už používa iná stanica. Vzhľadom ku klientovi, ktorý túto duplicitnú adresu overoval sa jedná o neúspešný pokus o pridelenie adresy. Naopak z pohľadu stanice, ktorá už adresu používa sa môže jednať o 3 rôzne varianty:

      i. Ak modul o danej adrese nemá žiadny záznam, vytvorí si ho a vygeneruje správu *IRI Begin*.
      ii. Ak modul o danej adrese má záznam a obsahuje rovnakú MAC adresu ako zdrojová adresa klienta, ktorý adresu overuje, vygeneruje správu *IRI Continue*.

iii. Ak modul o danej adrese má záznam a obsahuje inú MAC adresu ako zdrojová adresa klienta, ktorý adresu overuje, znamená to, že adresa bola medzičasom priradená inému klientovi. Na pôvodný záznam sa odošle správa *IRI End* a na nový vytvorený záznam *IRI Begin*.

(b) Ak do 2 sekúnd nepríde žiadna odpoveď na kontrolu duplicity, pridelenie adresy sa týmto potvrdí. Na základe informácie o klientovi sa vykoná jedna z možností i-iii. uvedených v bode (a).

V obidvoch prípadoch vykoná modul prechod do stavu *Unikátna/pridelená adresa*. Je nutné podotknúť, že diagram znázorňuje stavy pre jednotlivé adresy a nie pre koncové stanice. Zatiaľ čo pre jednu stanicu sa môže jednať o neúspešný pokus, pre inú stanicu reprezentuje rovnaká udalosť úspešné pridelenie alebo predĺženie adresy.



Obrázok 4.4: Stavový diagram IRI-IIF protokolu SLAAC

4. Pred použitím vygenerovanej adresy sa klient musel prihlásiť do multicastovej skupiny odvodenej z IPv6 adresy (*solicited-node* adresa). Smerovač sa periodicky pýta pomocou správy *ICMPv6 Multicast Listener Query*, či je v danej skupine prihlásená nejaká stanica. Prijatím uvedenej správy vykoná modul prechod do stavu *Kontrola multicast skupiny*.

5. Pokiaľ do časového intervalu uvedeného v predchádzajúcej ICMPv6 správe nepríde aspoň jedna odpoveď od ľubovoľného klienta patriacej do danej skupiny, znamená to, že všetky IPv6 adresy patriace do danej skupiny už nie sú používané. Je to z dôvodu, že koncová stanica odpovedá na kontrolu platnosti multicastovej skupiny len v tom prípade, keď nikto iný zatiaľ neodpovedal a ubehol vopred definovaný časový interval, ktorý je pre každého klienta náhodný. Modul vygeneruje pre tieto neplatné adresy správy *IRI End* a prechádza do stavu *Inicializácia*.

6. Naopak, pokiaľ aspoň v rámci kontroly multicastovej skupiny odpovie aspoň jeden klient, potvrdí tým aj platnosť pre všetky IPv6 adresy patriacej do daného rozsahu a modul vykoná prechod späť do stavu *Unikátna/pridelená adresa*.

7. V prípade, že adresa nie je dlhodobo videná na sieti, je po približne 24 hodinách odstránená. Modul pri odstránení odošle *IRI End*.

Pri analýze chovania protokolu SLAAC na rôznych operačných systémoch bolo bohužiaľ zistené, že niektoré z nich nedodržujú predpísanú normu ako napr. neposielajú všetky pakety, ktoré by mali byť odoslané alebo poradie týchto správ je prehádzané. Pre modul sa jedná o nepríjemný problém, pretože algoritmus sa nemôže spoliehať na presné znenie normy.

4.3 Simple Mail Transfer Protocol

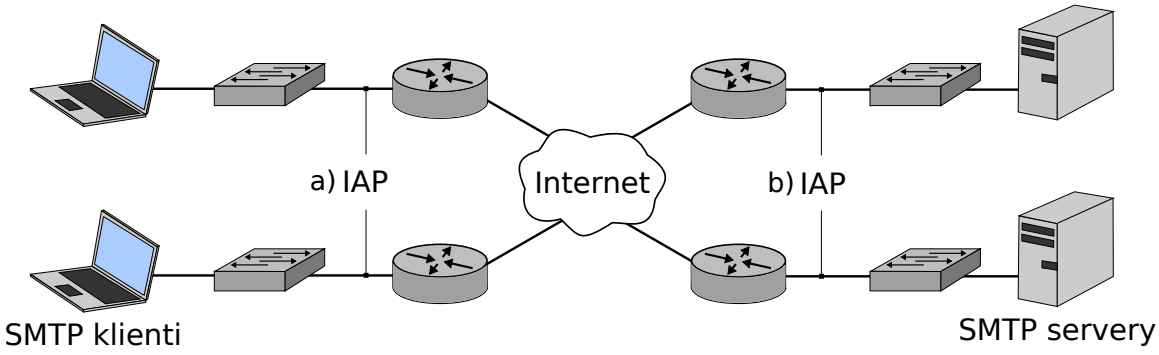
Protokol SMTP je aplikačným protokolom, čo prináša do analýzy protokolu jeden veľmi významný problém. Aplikačné protokoly je možné nastaviť tak, aby využívali ľubovoľné porty (nie len tie priradené organizáciou IANA) a preto nie je veľmi jednoduché povedať, či je nejaká komunikácia SMTP komunikáciou. Aj napriek tomu, že organizácia IANA pridelila protokolu SMTP TCP porty číslo 25 a 587, nie sú tieto údaje brané nijako do úvahy [12]. Modul IRI-IIF preto musí analyzovať každé TCP spojenie v ktorom vyhľadáva správy vo formáte SMTP protokolu.

Každé spojenie je identifikované IP adresami a portami oboch účastníkov komunikácie spolu s typom použitého transportného protokolu (v tomto prípade vždy TCP). Analýza každého spojenia začína výhradne pri nadväzovaní TCP spojenia. Ak by modul náhodou zachytil správy z už prebiehajúcej komunikácie, tak o danú komunikáciu nebude mať záujem.

V priebehu analýzy komunikácie sa modul snaží detegovať e-mail odosielateľa, e-mail prijímateľa (resp. prijímateľov), počet prijímateľov, unikátne číslo správy a veľkosť posielanej správy. Vzhľadom na stav projektu Sec6Net, nie je momentálne detegované unikátne číslo správy a veľkosť prenášaných dát. K tomu je totiž potrebné vyriešiť problém skladania TCP paketov, na ktorom sa zatiaľ v rámci projektu pracuje.

Výsledný stavový automat protokolu sa nachádza v prílohe E. Na obrázku 4.6 je zobrazená zjednodušená verzia tohto automatu, ktorého stavy sú nasledujúce:

- **Inicializácia** - v sieti nie je žiadne začínajúce TCP spojenie
- **Spojenie nadviazané** - odosielateľ nadviazal s SMTP serverom TCP spojenie
- **Užívateľ privítaný** - odosielateľ prijal od serveru zoznam podporovaných rozširujúcich funkcií
- **Pokus o autentifikáciu** - odosielateľ sa pokúša autentifikovať
- **Autentifikácia úspešná** - odosielateľ sa úspešne autentifikoval
- **Špecifikovanie prijímateľov** - odosielateľ odosiela na server zoznam príjemcov e-mailu
- **Odoslanie správy** - odosielateľ začal s prenosom e-mailovej správy
- **Správa úspešne odoslaná** - odosielateľ úspešne preniesol e-mail na SMTP server



Obrázok 4.5: Ukážková topológia protokolu SMTP s predpokladaným umiestnením IRI-sondy: a) na strane ISP, b) na strane poskytovateľa služby

Topológia spolu s predpokladaným umiestnením *IRI-sondy* je zobrazená na nasledujúcom obrázku 4.5.

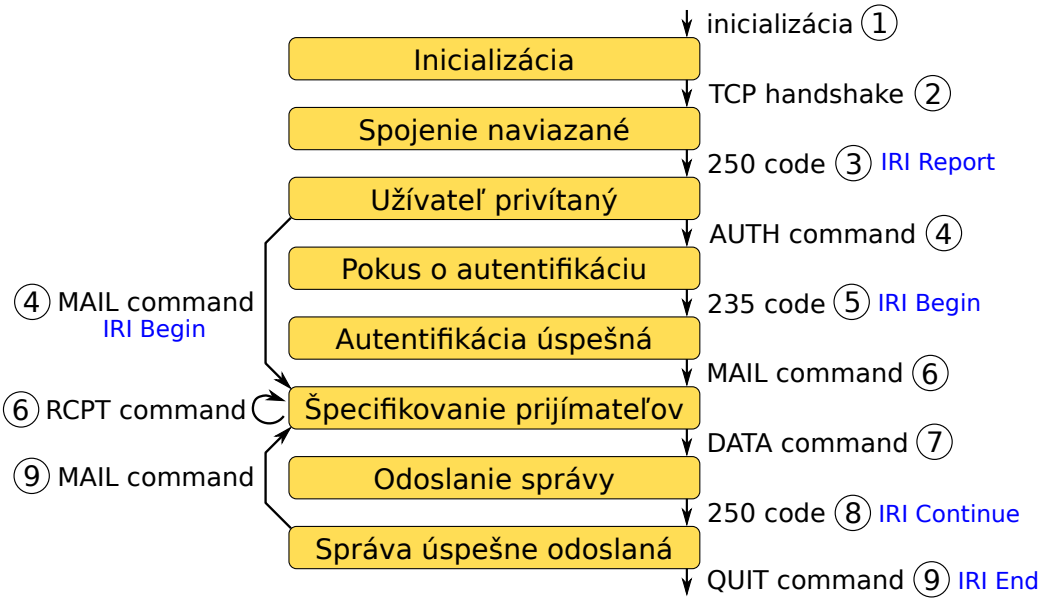
Vzhľadom k tomu že v rámci TCP komunikácie môže byť odosielateľom správy odosielateľ e-mailu aj SMTP server, použil som pri popise modulu IRI-IIF nasledujúcu konvenciu:

- **klient** - odosielateľ e-mailu v rámci TCP spojenia
- **server** - prijímateľ e-mailu v rámci TCP spojenia (vždy sa jedná o SMTP server)
- **odosielateľ** - užívateľ, ktorý posiela e-mail

Činnosť bloku IRI-IIF sa riadi podľa nasledujúcich pravidiel:

1. Počiatočným stavom je *Inicializácia*. Modul IRI-IIF sa snaží detegovať novú, ešte nepriebehajúcu, TCP komunikáciu.
2. Nové TCP spojenie sa deteguje zachytením *TCP handshake*, ktoré je nutné vykonať na začiatku každého spojenia. Klient sa so serverom dohaduje na sekvenčných číslach, ktoré budú ďalej používať. Následne sa modul presunie do stavu *Spojenie nadviazané*.
3. Po úspešnom nadviazaní spojenia odošle server uvítaciu správu s kódom 220 (*Service ready*). Klient po prijatí správy odošle príkaz EHLO, ktorým žiada server o identifikáciu a inicializáciu SMTP spojenia. Server na príkaz EHLO odpovie zoznamom parametrov označených kódom 250 (*Requested mail action okay, completed*) a oddelených koncami riadkov. V tomto zozname parametrov sa nachádza identifikácia SMTP servera a podporované rozširujúce funkcie. Po zachytení správy obsahujúcej kód 250 sa vykoná prechod do stavu *Užívateľ privítaný* a odošle sa správa *IRI Report*.
4. V závislosti na konfigurácii klienta sa môže začať autentifikácia klienta alebo špecifikovanie e-mailovej adresy odosielateľa.
    - V prípade komunikácie medzi odosielateľom e-mailu a jeho poštovým serverom sa klient väčšinou pokúsi autentifikovať. Ako prvú správu odošle klient príkaz AUTH s parametrom obsahujúcim typ autentifikácie. Modul sa presunie do stavu *Pokus o autentifikáciu*.

- Druhá možnosť je v prípade komunikácie medzi dvomi SMTP servermi alebo keď je odosielateľ e-mailu na svojom poštovom serveri autentifikovaný iným spôsobom (napr. zdrojová IP adresa) prípadne vôbec. Za týchto okolností klient odošle príkaz MAIL s parametrom obsahujúcim zdrojovú e-mailovú adresu odosielateľa. E-mail odosielateľa je zo správy dekódovaný a odoslaný správou *IRI Report* do *IRI-Core*. Súčasne s odoslaním správy IRI sa vykoná presun do stavu *Špecifikovanie prijímateľov*.



Obrázok 4.6: Stavový diagram IRI-IIF protokolu SMTP

5. Samotný priebeh autentifikácie nie je v stave *Pokus o autentifikáciu* analyzovaný. Na záver tohto procesu server odpovie klientovi buď s kódom 235 (*Authentication successful*) reprezentujúcim úspešnú autentifikáciu na základe ktorého sa modul presunie do stavu *Autentifikácia úspešná* a odošle správa *IRI Begin*, alebo s kódom 535 (*Authentication failed*) reprezentujúci neúspešnú autentifikáciu.

6. Po úspešnej autentifikácii odošle odosielateľ správy príkaz MAIL, v ktorom uvedie e-mail odosielateľa. Po špecifikovaní odosielateľa e-mailu klient začne odosielať zoznam príjemcov e-mailu. Počet prijímateľov je minimálne jeden, maximum nie je protokolom SMTP špecifikované. Klient pošle príkaz RCPT s prvým príjemcom správy a následne čaká na prijatie potvrdenia príjemcu serverom. K potvrdeniu príjemcu server opäť využije odpoveď s kódom 250 (*Requested mail action okay, completed*). Až po potvrdení príjemcu môže klient odoslať ďalšieho príjemcu pomocou rovnakého príkazu RCPT.

   Formát parametru správy RCPT môže mať viacero podôb. Najzákladnejšia podoba je zadanie plnej e-mailovej adresy prijímateľa spolu s doménou. Modul túto adresu neskôr priamo odošle správou *IRI Continue*. Ďalšou možnosťou je špecifikovanie e-mailu prijímateľa bez domény. Doména prijímateľa je tak rovnaká ako doména odosielateľa správy. Modul preto k emailu prijímateľa túto doménu taktiež pridá a takto

vzniknutú e-mailovú adresu neskôr odošle správou *IRI Continue*. Poslednou možnosťou špecifikácie prijímateľa je zadanie hodnoty „postmaster“. Hodnota „postmaster“ reprezentuje správcu poštového servera odosielateľa, pričom pri tejto hodnote nie sú nerozlišované veľké a malé písmená. Hodnota „postmaster“ sa bez pridania domény neskôr odošle správou *IRI Continue*.

7. Po zadaní posledného príjemcu správy príkazom RCPT nasleduje samotný prenos obsahu správy. Klient tento stav dáva najavo odoslaním správy DATA. Po potvrdení správy serverom začne klient odosielať správu a modul IRI-IIF prejde do stavu *Odoslanie správy*.

8. Koniec odosielania správy je detegovaný odoslaním znaku „.“ zo strany klienta. Server následne odpovie buď s kódom 250 (*Requested mail action okay, completed*) reprezentujúcim, že všetko prebehlo v poriadku alebo s kódom reprezentujúcim číslo chyby. Pri odoslaní kódu 250 modul odošle správu *IRI Continue* so všetkými parametrami, ktoré dokázal zistiť (e-mail odosielateľa, e-maily prijímateľov, atď.). Dôvod prečo nie sú e-mailové adresy po zadaní príkazu MAIL alebo RCPT odosielané ihneď je popísané v poslednom bode - príkaz RSET. Zachytením kódu 250 sa modul presunie do stavu *Správa úspešne odoslaná*.

9. Po úspešnom odoslaní správy môže klient príkazom QUIT ukončiť SMTP spojenie, po ktorom bude nasledovať ukončenie TCP spojenia. Modul tak odošle správu *IRI End* a vráti sa do stavu *Inicializácia*. Avšak ak má klient ďalšie správy na odoslanie, nemusí ukončovať spojenie a vytvárať nové. Namiesto toho môže znovu odoslať správu MAIL s platnou hodnotou e-mailu odosielateľa čím inicializuje prenos ďalšej správy. Modul na túto správu zareaguje odoslaním správy *IRI Continue* a prechodom do stavu *Špecifikovanie prijímateľov*.

10. Počas celej doby komunikácie môže klient odoslať príkaz RSET alebo QUIT. Pri príkaze QUIT modul deteguje ukončenie spojenia a v prípade, že počas spojenia bola odoslaná správa *IRI Begin*, tak sa odošle správa *IRI End*. Príkaz RSET dáva zmysel len po autentifikácii užívateľa a slúži k vynúteniu zmazania všetkých parametrov správy (e-mail odosielateľa a prijímateľov) na strane servera. Tieto parametre je tak nutné zadať znovu. Z tohto dôvodu sa identifikátory neodosielajú ihneď po detekcii. Pri detekovaní príkazu MAIL po príkaze RSET sa modul presunie do stavu *Špecifikovanie prijímateľov*.

Kapitola 5

Návrh a implementácia modulov IRI-IIF

Obsahom tejto kapitoly je popis implementácie softwaru jednotlivých modulov. Výber prostredia a programovacieho jazyka pre implementáciu modulov nemohol byť ľubovoľný, pretože výsledný software je nutné zakomponovať do projektu Sec6Net. Bolo zvolené prostredie OS Linux na ktorom beží celá IRI-IIF a programovací jazyk Python.

K tomu, aby moduly mohli spracovávať pakety zo siete bola využitá knižnica *pcapy* [3]. Vďaka knižnici *pcapy* dostávajú moduly pakety zo siete v podobe poľa bajtov správy. Obsahom poľa bajtov je zachytený rámec (dáta na vrstve sieťového rozhrania). Knižnica *pcapy* okrem spracovávania paketov zo živej siete umožňuje aj spracovávanie paketov z uloženého súboru. Táto funkcionalita je pre protokoly PPPoE a SMTP podporovaná, avšak protokol SLAAC spracovávanie paketov týmto spôsobom neumožňuje. Dôvod prečo tomu tak je bude vysvetlený pri popise implementácie protokolu SLAAC 5.2. Okrem použitej knižnice *pcapy* na spracovávanie paketov je implementácia každého protokolu úplne unikátna, preto popíšem každý protokol zvlášť.

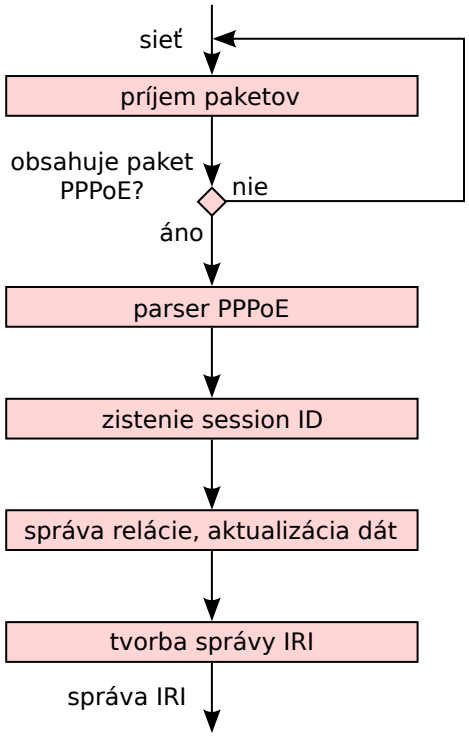
5.1 Point-to-Point Protocol over Ethernet

Vývojový diagram implementovaného modulu je zobrazený na obrázku 5.1. Popis diagramu a implementácie bude vysvetlený v nasledujúcich odstavcoch.

Prijatý rámec nie je nutné dekapsulovať a obrezávať, pretože to či správa obsahuje PPPoE dáta alebo nie, je uvedená na vrstve sieťového rozhrania. Ak je hodnota zapuzdreného protokolu 0x8863 (*PPPoE Discovery*) alebo 0x8864 (*PPPoE Session*) modul deteguje, že prijaté dáta sú určené na ďalšie spracovanie a odstráni Ethernetovú hlavičku a pätičku.

Informácie o aktívnych spojeniach sú ukladané podľa hodnoty SESSION_ID, ktorá je vždy obsiahnutá v hlavičke protokolu PPPoE a jednoznačne identifikuje spojenie. Následne prebehne spracovanie dát a na základe stavového automatu z obrázku 4.2 sa rozhodne o ďalšom postupe.

Úložisko jednotlivých detegovaných spojení je implementované vo formáte hash tabuľky, ktorej index je hodnota SESSION_ID. V hash tabuľke sú uložené všetky známe informácie o spojení: stav spojenia (automatu), číslo PPPoE spojenia, MAC adresa klienta, prihlasovacie meno klienta a priradené IP adresy.



Obrázok 5.1: Vývojový diagram modulu PPPoE

5.2 Stateless Address Autoconfiguration

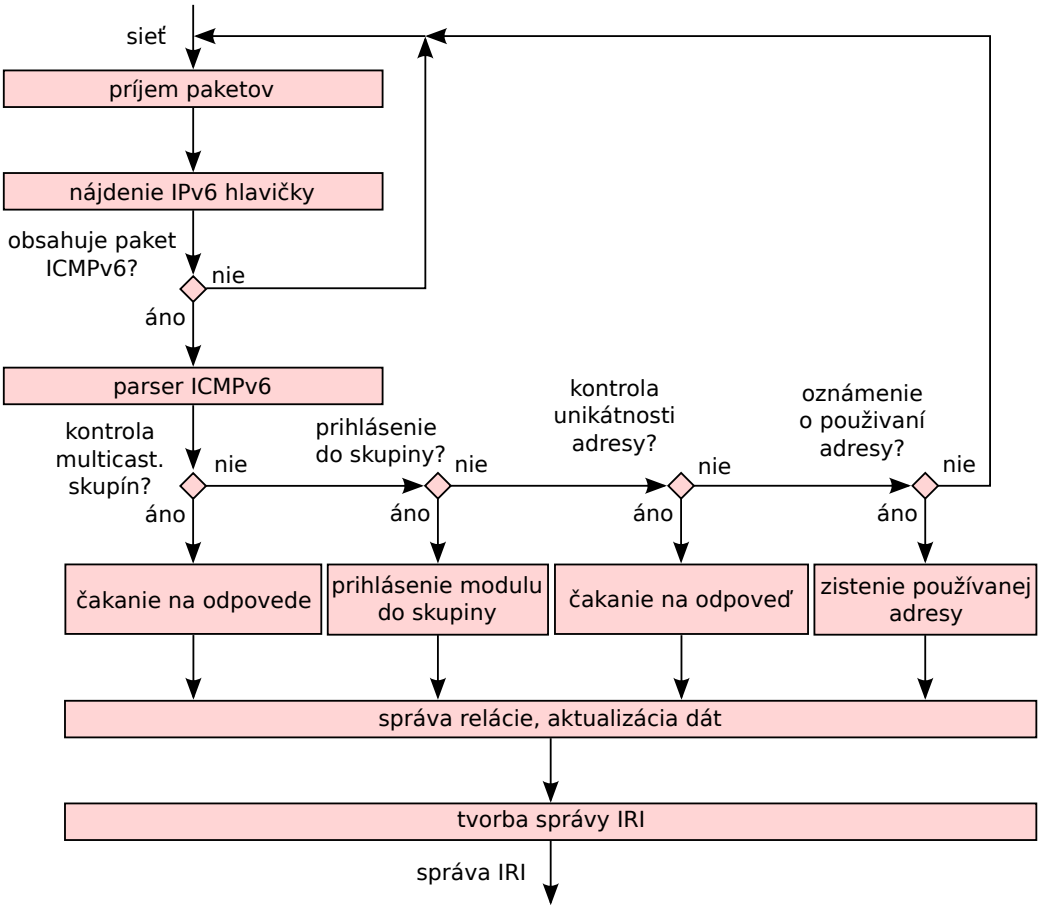
Vývojový diagram implementovaného modulu je zobrazený na obrázku 5.2. Popis diagramu a implementácie bude vysvetlený v nasledujúcich odstavcoch.

Príjem rámcov je limitovaný na príjem zo siete. Pri príjme dát zo siete sú rámce knižnicou *pcapy* okamžite, bez dodržania časového odstupu medzi paketmi, odovzdávané na spracovanie. Modul SLAAC však z dôvodu využitia viacerých časovačov vyžaduje dodržiavanie časového odstupu medzi paketmi. Toto obmedzenie je možné odstrániť využitím programov pre prehrávanie uložených paketov na sieťové rozhranie *loopback*, ktoré by následne modul spracovával.

Ak modul príjme rámec so zapuzdreným protokolom 0x86dd (IPv6), prijatý rámec je nutné dekapsulovať a obrezať tak, aby sme mohli pracovať s hlavičkou na internetovej vrstve. Na internetovej vrstve sa taktiež zisťuje informácia o zapuzdrenom protokole. Pri nájdení hodnoty 0x3a (ICMPv6) je hlavička internetovej vrstvy odstránená a zapuzdrené dáta analyzované.

Protokol ICMPv6 obsahuje veľké množstvo správ pričom modul SLAAC zaujímajú len: *Multicast Listener Query* (smerovač kontroluje aktívne multicastové skupiny), *Multicast Listener Report* (prihlásenie stanice do multicastovej skupiny), *Multicast Listener Report Message v2* (prihlásenie stanice do multicastovej skupiny), *Neighbor Solicitation* (stanica kontroluje unikátnosť novej IPv6 adresy) a *Neighbor Advertisement* (stanica informuje o používaní IPv6 adresy).

V závislosti na správe ICMPv6 sa modul môže prihlasiť do multicastovej skupiny, čakať na reakciu staníc na sieti alebo detegovať používanie IPv6 adresy. Tak či onak, sa na základe stavového automatu z obrázku 4.2 rozhodne o ďalšom postupe.



Obrázok 5.2: Vývojový diagram modulu SLAAC

Pre uloženie všetkých potrebných informácií sú použité tri rôzne úložiská. Prvé je úložisko aktívnych adries, implementované ako hash tabuľka, ktorá obsahuje aktuálne priradené IPv6 adresy k MAC adresám. Druhé, úložisko overujúcich adries, je zoznam IPv6 adries spolu s MAC adresami, ktoré ešte neboli overené pre unikátnosť v sieti. Tretie úložisko je úložisko multicastových adries, obsahujúci zoznam multicastových adries v ktorých je modul aktuálne prihlásený. Ako bolo vysvetlené v kapitole 4.2, modul sa zároveň s klientmi prihlasuje do multicastových adries.

Zapúzdrený protokol SLAAC je následne analyzovaný a na základe stavového automatu z obrázku 4.4 sa rozhodne o ďalšom postupe.

Modul SLAAC taktiež využíva viacero časovačov, ktoré slúžia k udržiavaniu konzistencie medzi sieťou a úložiskami. Všetky časovače sú imlementované pomocou signálovej komunikácie a ich hodnoty sú nasledujúce:

- **2x za sekundu** - Spustí sa kontrola s cieľom všetky adresy, ktoré sú minimálne sekundu v stave nepotvrdené, potvrdiť ako unikátne v sieti. Ak si OS kontroluje unikátnosť adresy, ktorá je na sieti duplicitná, odpoveď na kontrolu prichádza okamžite. Sekunda je preto rezerva pri prípadnom oneskorení príjmu paketu na module. Všetky potvrdené adresy sú presunuté z úložiska overujúcich adries do úložiska aktívnych adries. Taktiež sa v rámci tohto časovača kontroluje, či nie sú v prvom úložisku adre-

sy, ktoré v nastavenom časovom limite neodpovedali na kontrolu smerovača ohľadom aktívnych multicastových adries. Vzhľadom na výsledky analýzy OS 4.2.2 je ku časovému limitu odpovedi pripočítaná 1 sekunda, aby sa stihli zachytiť všetky odpovede. Adresy, ktoré v limite neodpovedali, sa odstránia.

- **2x za minútu** - Aby modul zbytočne nezaťažoval sieťové zariadenia na sieti, je nežiadúce aby bol prihlásený v multicastových skupinách dlhšiu dobu ako je čas potrebný pre priradenie adresy. Na základe analýzy OS 4.2.2 bolo zistené, že všetky OS stihli v priebehu 15 sekúnd overiť všetky svoje priraďované adresy. Vzhľadom na dôležitosť správ vymieňaných v priebehu overovania adries, je ku intervalu 15 sekúnd pripočítaná väčšia rezerva 30 sekúnd. Každých 30 sekúnd sa tak zo zoznamu multicastových skupín odstraňujú tie položky, ktoré sú staršie ako 45 sekúnd.

- **1x za hodinu** - Po tom ako ľubovoľná adresa nie je na sieti videná 24 hodín je zmazaná z hash tabuľky. Každá aktívne komunikujúca adresa je totiž časom pomocou správ ND prenášaná po sieti. V prípade, že adresa je v sieti pripojená, len nie je aktívna, je predčasne označená za nepoužívanú. Avšak v prípade opätovnej aktívnosti adresy, sa znovu odošlú po sieti ND správy vďaka ktorým sa adresa znovu deteguje.

5.3 Simple Mail Transfer Protocol

Pri module SMTP bola použitá experimentálna metóda spracovávania paketov zo siete. Postupom času ako sa projekt Sec6Net vyvíjal, vznikla myšlienka centralizovať spracovávanie paketov. Na základe tejto myšlienky je implementácia protokolu SMTP výrazne odlišná od predošlých protokolov. Pri centrálnom spracovávaní totiž modul nezískava dáta zo siete priamo pomocou knižnice *pcapy*, ale centrálne pomocou softwaru k tomu určenému. Centrálne spracovanie paketov však pre moduly zatiaľ nebolo implementované a tak som si vytvoril vlastný prototyp tohto riešenia (ďalej len *parser*), ktorý sa skladá z viacerých súborov:

- **parser_main.py** - Spúšťač a hlavný súbor parseru. Parser sa spustí na náhodnom porte, pričom tento port je zobrazený, ako argument programu, pri výpise procesov aby sa na neho mohli moduly jednoducho - bez nutnosti použitia statického portu pripojiť. Parser spustí odpočúvanie na každom sieťovom rozhraní, pričom podporuje ich ľubovoľné transparentné zlučovanie pre moduly. Po spustení odpočúvaní na rozhraniach sa vytvorí socket na ktorom bude parser očakávať nové pripojenia.

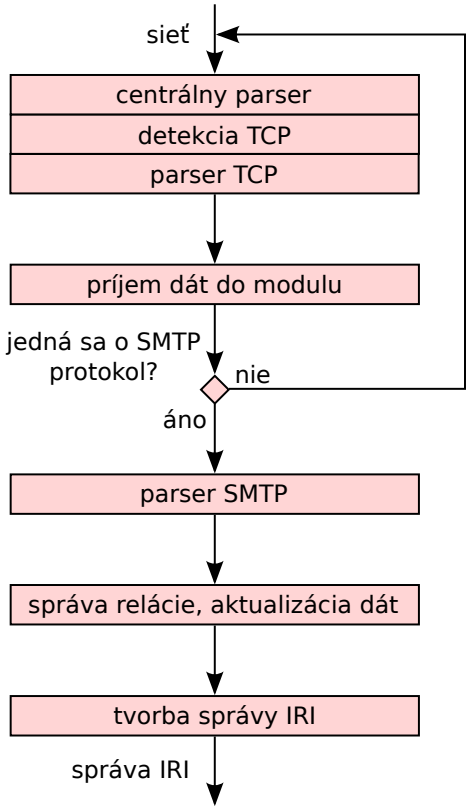
  Po pripojení nového modulu, odošle modul prostredníctvom socketovej komunikácie parseru svoju konfiguráciu. Súčasťou konfigurácie je zoznam rozhraní a protokolov, o ktoré má modul záujem. Taktiež je súčasťou konfigurácie textový reťazec, obsahujúci príkaz na znovu spustenie modulu pri jeho neočakávanom páde. Modul totiž dokáže okamžite identifikovať pád modulu a má tak možnosť na túto akciu rýchlo zareagovať.

  Následne modul analyzuje pakety na rozhraniach a v prípade, že sa jedná o protokol a rozhranie o ktoré má niektorý modul záujem, je tento protokol poslaný príslušnému modulu. Cieľovému modulu tak nie je odoslaný celý paket ale len dáta obrezaného protokolu (napr. *TCP payload*) o ktorý mal záujem spolu s identifikátormi na nižších sieťových vrstvách.

- **parser_functions.py, parser_protocols.py** - Súbor *parser_functions.py* obsahuje pomocné funkcie pre fungovanie parseru a modulu. Súbor *parser_protocols.py* obsahuje funkcie na spracovanie jednotlivých protokolov. Okrem cieľových protokolov o ktoré majú moduly záujem obsahuje aj rôzne protokoly nižších vrstiev vrátane tunelovacích. Pridaním funkcie na spracovanie nového protokolu do *parser_protocols.py* tak vznikne podpora tohto protokolu na všetkých moduloch.

- **client.py** - Jedná sa o súbor, ktorý je zakomponovaný do modulu. Modul si prostredníctvom funkcií z tohto súboru autonómne získa informácie potrebné k pripojeniu k parseru a pripojí sa. V prípade, že parser nie je spustený alebo sa počas behu modulu nečakane ukončí, modul si sám parser dokáže spustiť. Súbor *client.py* sa po pripojení na parser stará o správny príjem žiadaných dát od parseru a ich prekonvertovanie do ľahko použiteľnej podoby (pole hodnôt).

Vývojový diagram implementovaného modulu je zobrazený na obrázku 5.3. Popis diagramu a implementácie bude vysvetlený v nasledujúcich odstavcoch.



Obrázok 5.3: Vývojový diagram modulu SMTP

Modul sa prostredníctvom socketovej komunikácie spojí s parserom, ktorému odošle zoznam rozhraní o ktoré má záujem spolu s informáciou, že má záujem len o príjem dát protokolu TCP. Centrálny parser dekapsuluje a obrezáva vrstvy z prijatého rámca až po nájdení TCP hlavičky. Oproti spracovávaniu rámca na moduloch, umožňuje parser okrem spracovania IP hlavičky spracovávať aj ďalšie hlavičky protokolov, ktoré sa môžu medzi Ethernetovou a TCP hlavičkou vyskytovať. Môže sa jednať o hlavičky protokolov MPLS,

PPPoE, L2TP, GRE, Teredo a taktiež hlavičky protokolov TCP a UDP pri použití ako tunelovacie protokoly. Detegovaný protokol TCP je následne spracovaný a odoslaný modulu.

Po prijatí spracovanej TCP hlavičky modul na základe obsahu TCP dát a prípadného stavu spojenia zistí, či prijaté TCP dáta sú dáta protokolu SMTP alebo nie. V prípade ak sa jedná o SMTP dáta, prebehne spracovanie týchto dát a podľa automatu z obrázku 4.6 resp. E.1 sa rozhodne o ďalšom postupe.

Pre uloženie údajov o komunikáciách sa používajú tri úložiská. V prvom sú uložené identifikátory nižších vrstiev, aktuálny stav spojenia a identifikátory spojené s príslušnou e-mailovou správou (odosielateľ a prijímatelia). Obsahom druhého úložiska je zoznam identifikátorov, ktoré však zatiaľ neboli potvrdené serverom príslušným kódom. Tretie úložisko obsahuje zoznam potvrdzujúcich správ vzťahujúcich sa k identifikátorom e-mailovej správy, ktoré zatiaľ neboli prijaté. Príkladom situácie, kedy je prijaté potvrdenie identifikátoru ešte pred prijatím správy je pri prehodení paketov na sieti. Hodnoty zo všetkých tabuliek sú odstraňované naraz pri ukončení spojenia.

Kapitola 6

Testovanie modulov IRI-IIF

Po dokončení implementácie modulov sa prešlo k ich otestovaniu. Každý modul bol otestovaný na vstupných testovacích súboroch vo formáte *pcap*, ktoré boli v rámci projektu Sec6Net predpripravené. Pre tieto súbore bola vopred vytvorená postupnosť udalostí, ktorú sa očakávajú, že modul deteguje a odošle vo forme správ IRI. Testovanie prebiehalo na počítači s nainštalovaným systémom pre zákonné odpočúvania, na ktorom boli uložené pakety odoslané modulom pre spracovanie. Účelom testovania bolo porovnávanie výstupných správ modulu (správy IRI) s postupnosťou udalostí, ktorá sa od modulu očakávava.

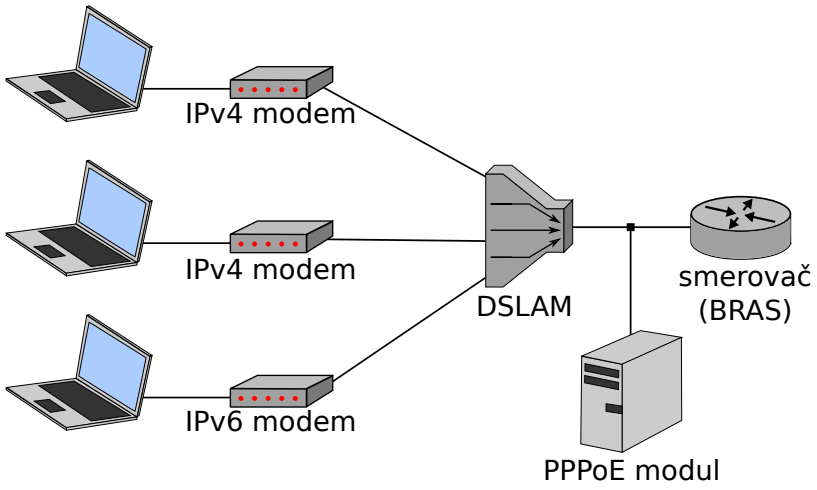
6.1 Point-to-Point Protocol over Ethernet

Testovacie súbory vo formáte *pcap* pre protokol PPPoE obsahovali vytvorenie spojenia, autentifikáciu pripájaného klienta a priradenie IP adries. Celkovo bolo otestovaných 8 rôznych *pcap* súborov, z ktorých bol pre ukážku odosielaných IRI správ vybraný jeden (viď tabuľka 6.1). Vo vybranom *pcap* súbore sa pripojil užívateľ s prihlasovacím menom „username“, ktorému bola priradená IPv4 adresa „192.168.16.1“. Pri všetkých *pcap* súboroch bol výstup modulu zhodný so zoznamom udalostí, ktoré sa od modulu očakávalo, že deteguje.

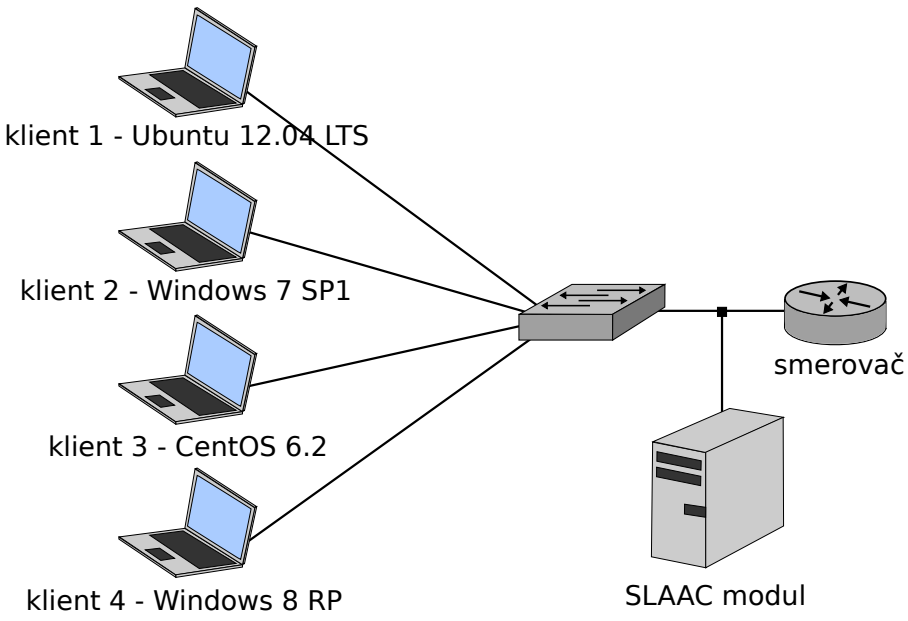
Poradie udalosti	**Typ IRI správy**	**Popis události prenesený v správe IRI**	**Odoslané identifikátory**
1	Report	Klient naviazal spojenie s BRASOM	PPP SESSION: 0001 MAC: 00:11:3b:06:02:5e
2	Report	Užívateľ sa pokúša autentifikovať	PPP SESSION: 0001 MAC: 00:11:3b:06:02:5e PPP LOGIN: username
3	Report	Bolo detegované prihlasovacie meno užívateľa	PPP SESSION: 0001 MAC: 00:11:3b:06:02:5e PPP LOGIN: username
4	Begin	Užívateľovi bola priradená IP adresa	PPP SESSION: 0001 MAC: 00:11:3b:06:02:5e PPP LOGIN: username IP: 192.168.16.1

Tabuľka 6.1: Zoznam správ IRI odosielaných modulom PPPoE

Modul bol ďalej otestovaný v špecializovanom laboratóriu na fakulte FIT. V laboratóriu bolo podľa schémy zapojenia, ktorá je zobrazená na obrázku 6.1, zapojené zariadenie DSLAM, smerovač od firmy Cisco, ktoré bolo nakonfigurované ako BRAS a niekoľko modemov pre ADSL pripojenie. Pri testovaní sa pomocou zapínania a vypínania modemov simulovalo náhodné správanie reálnych užívateľov. Výstup z modulu tak bol kontrolovaný priamo so simulovaným prostredím. Výstupy modulu správne korespondovali so simulovanými udalosťami.



Obrázok 6.1: Schéma zapojenia testovacieho prostredia pre PPPoE modul



Obrázok 6.2: Schéma zapojenia testovacieho prostredia pre SLAAC modul

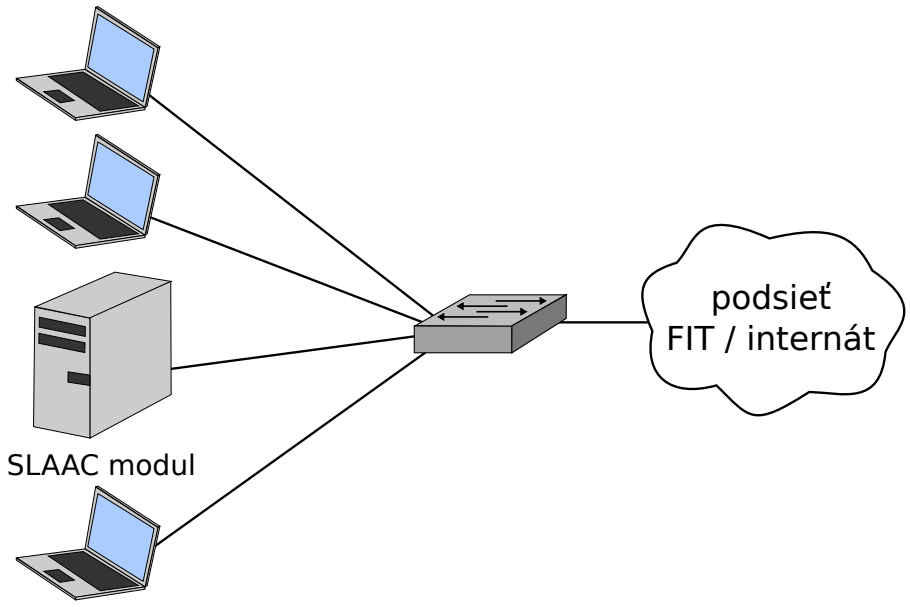
6.2 Stateless Address Autoconfiguration

Testovacie súbory vo formáte *pcap*, s uloženými paketmi, boli vytvorené pri analyzovaní OS (kapitola 4.2.2). Celkovo bolo vytvorených 101 súborov, ktoré obsahovali správy odosielané OS pri použití protokolu SLAAC. Zoznam odosielaných IRI správ modulom SLAAC, pre jeden z týchto súborov, je zobrazený v tabuľke 6.2. V uvedenej tabuľke je detegovaný OS CentOS 6.2 pri zapojení do siete a so zapnutými *privacy extension* adresami. Pri všetkých *pcap* súboroch bol výstup modulu zhodný so zoznamom udalostí, ktoré sa od modulu očakávalo, že deteguje.

Ďalej bol modul (rovnako ako protokol PPPoE) otestovaný v špecializovanom laboratóriu na fakulte FIT. Schéma zapojenia testovacieho prostredia je znázornená na obrázku 6.2. Výstupy modulu správne korespondovali s udalosťami vytvorenými v testovaciom prostredí.

Poradie udalosti	**Typ IRI správy**	**Popis události prenesený v správe IRI**	**Odoslané identifikátory**
1	Begin	Užívateľ si vygeneroval novú IP adresu	MAC: 00:0c:29:a1:fa:76 IPv6: fe80::20c:29ff:fea1:fa76
2	Begin	Užívateľ si vygeneroval novú IP adresu	MAC: 00:0c:29:a1:fa:76 IPv6: 2001:abcd::dc5a:5d0:1066:d346
3	Begin	Užívateľ si vygeneroval novú IP adresu	MAC: 00:0c:29:a1:fa:76 IPv6: 2001:abcd::20c:29ff:fea1:fa76

Tabuľka 6.2: Zoznam správ IRI odosielaných modulom PPPoE

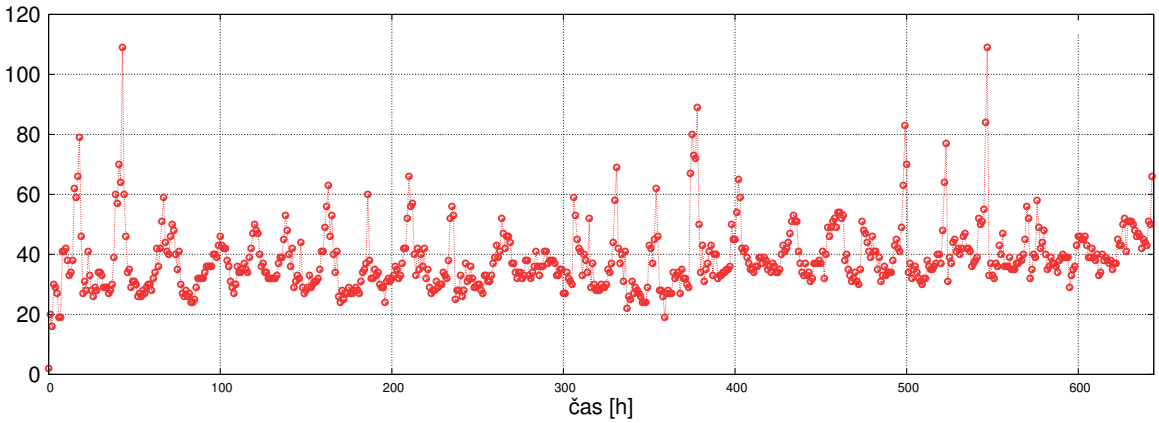


Obrázok 6.3: Schéma zapojenia modulu SLAAC v produkčných sieťach

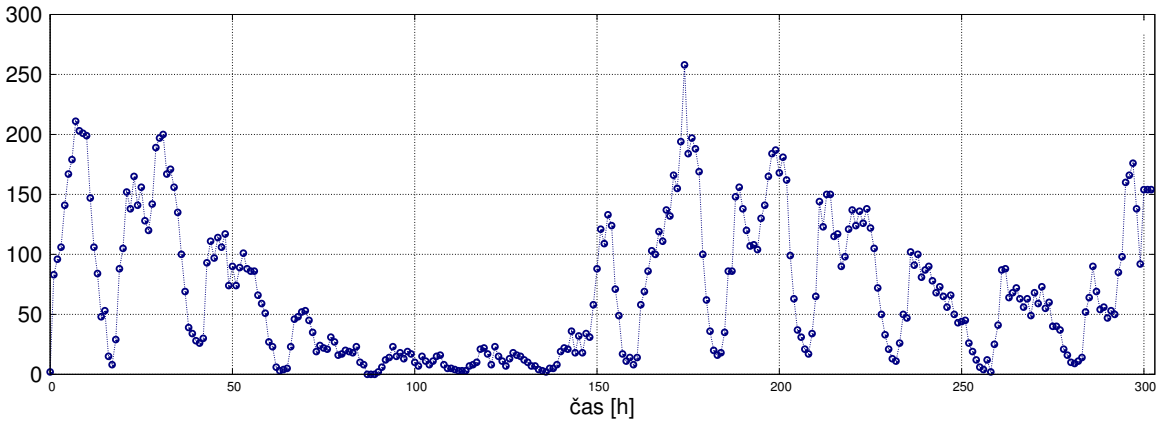
Modul bol navyše otestovaný na dvoch produkčných sieťach. Na rozdiel od predchádzajúcich testov, nebola celá sieť pod kontrolou a nebolo možné kontrolovať výstup modulu s každým zapojeným zariadením. Bolo však možné otestovať, ako sa modul vysporiada

s dlhodobým zapojením v takejto sieti a taktiež bolo možné otestovať, či medzi všetkými nekontrolovanými stanicami dokáže modul detegovať stanicu, ktorá bola pod kontrolou. Schéma zapojenia modulu bola na oboch produkčných schémach rovnaká, tak ako je zobrazená na obrázku 6.3.

Prvou produkčnou sieťou bola jedna z podsietí na fakulte FIT. Graf znázorňujúci počet aktuálne zapojených adries v sieti počas 26 dňového testovania je zobrazený na obrázku 6.5. Druhou produkčnou sieťou bola sieť nachádzajúca sa na internáte VUT. Graf znázorňujúci počet aktuálne zapojených adries v sieti počas 14 dňového testovania je zobrazený na obrázku 6.5. Na obidvoch grafoch je možné vidieť, že modul úspešne detegoval nové adresy (graf rastie) a taktiež úspešne detegoval nepoužívané adresy (graf klesá.)



Obrázok 6.4: Počet aktuálne zapojených adries na podsieti fakulty FIT počas 26 dňového testovania



Obrázok 6.5: Počet aktuálne zapojených adries na na podsieti internátu VUT počas 14 dňového testovania

Najväčším rozdielom medzi oboma grafmi, znázorňujúcimi počet zapojených adries, je maximálny počet zapojených adries v jednom okamžiku. Zatiaľ čo na podsieti fakulty FIT sa jednalo o hodnotu 109 na internáte to bolo až 258. Ďalším veľmi výrazným rozdielom je samotný priebeh grafu. Graf z podsieti na FITe sa po úvodnej inicializácii už nikdy nedostane na hodnotu 0, zatiaľ čo na podsieti internátu áno. Dôvod je ten, že na fakultnej

podsieti sa nachádzajú niektoré počítače, ktoré sú spustené bez prestávky.

6.3 Simple Mail Transfer Protocol

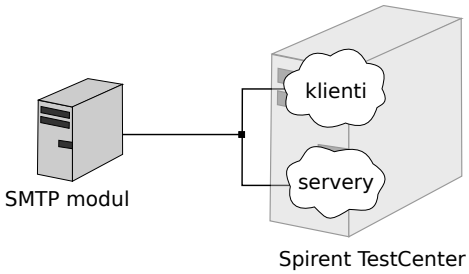
Uložené pakety pre protokol SMTP obsahovali vytvorenie spojenia, autentifikáciu pripájaného klienta a odoslanie e-mailu jednému príjemcovi. Celkovo bolo v rámci projektu Sec6Net predpripravené 2 testovacie súbory. Ďalší súbor bol stiahnutý z internetu [1]. V tabuľke 6.3 je uvedený zoznam správ odosielaných modulom pri analyzovaní *pcap* súboru stiahnutého z internetu. Súbor zachytáva pripojenia užívateľa z IPv4 adresy 10.10.1.4 a z e-mailu „gurpartap@patriots.in“, ktorý odošle správu na e-mail „raj_deol2002in@yahoo.co.in“. Pri všetkých *pcap* súboroch bol výstup modulu zhodný so zoznamom udalostí, ktoré sa od modulu očakávalo, že deteguje.

Poradie udalosti	**Typ IRI správy**	**Popis události prenesený v správe IRI**	**Odoslané identifikátory**
1	Report	Užívateľ sa pripojil na SMTP server	TCP*: (10.10.1.4, 1470, 74.53.140.153, 25)
2	Begin	Užívateľ bol úspešne autentifikovaný	TCP*: (10.10.1.4, 1470, 74.53.140.153, 25)
3	Continue	Užívateľ úspešne odoslal e-mail	TCP*: (10.10.1.4, 1470, 74.53.140.153, 25) E-mail sender: gurpartap@patriots.in Receivers count: 1 E-mail receivers: raj_deol2002in@yahoo.co.in
4	End	Užívateľ sa odpojil od SMTP serveru	TCP*: (10.10.1.4, 1470, 74.53.140.153, 25)

* (zdrojová IP, zdrojový port, cieľová IP, cieľový port)

Tabuľka 6.3: Zoznam správ IRI odosielaných modulom SMTP

Aj modul SMTP bol testovaný v špecializovanom laboratóriu na FITe, tentoraz ale iným spôsobom. V laboratóriu bolo zapojené testovacie zariadenie *Spirent TestCenter*, ktoré simulovalo SMTP servery a SMTP klientov. Schéma zapojenia je znázornená na obrázku 6.6. Zariadenie *Spirent TestCenter* vytváralo na sieti podľa vopred vytvorených pravidiel správy simulujúce reálne SMTP komunikácie. Celkovo bolo vytvorených 9 rôznych súborov s celkovým počtom 4387 SMTP spojení. Tieto súbory boli následne odoslané modulu SMTP pre analýzu. Výsledok analýzy dopadol rovnako úspešne, ako pre predpripravené súbory v rámci projektu Sec6Net.



Obrázok 6.6: Schéma zapojenia testovacieho prostredia pre SMTP modul

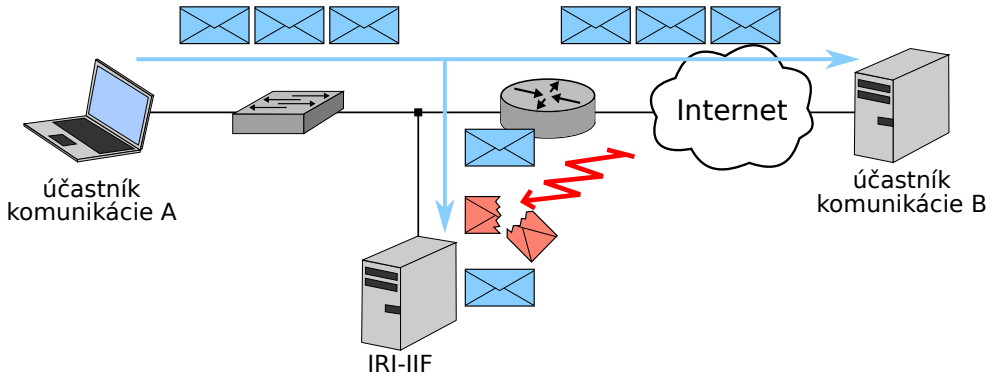
Kapitola 7

Vyhodnotenie nástrojov

Všetky moduly boli otestované tak ako to popisuje kapitola 6. Testovanie bolo vždy úspešné, pretože moduly identifikovali len tie spojenia a ich atribúty, ktoré mali. Nedetegovali žiadne spojenia ani ich atribúty navyše a ani žiadne nechýbali. Pri všetkých testoch však modul dostával všetky pakety zo siete, čo je síce očakávané, avšak pri reálnom nasadení sa môže stať, že niektoré dáta moduly neprijmú.

7.1 Neúplnosť vstupných dát

Pri reálnom nasadení môže vzniknúť situácia, kedy moduly neuvidia celú komunikáciu medzi dvoma účastníkmi. Niektoré pakety prenášané medzi dvomi účastníkmi sa nemusia úspešne preniesť do systému pre zákonné odpočúvanie, resp. bloku IRI-IIF. Môže sa tak stať vplyvom preťaženia sieťovej štruktúry, kedy sa nestihnú kopírovať všetky pakety a niektoré sú zahodené, alebo sa paket môže poškodiť (a následne zahodiť) počas prenosu do IRI-IIF ako zobrazuje obrázok 7.1.



Obrázok 7.1: Poškodenie paketu smerujúceho k IRI-IIF

Všetky moduly boli analyzované, ako problém s neúplnosťou vstupných dát môže ovplyvniť ich činnosť.

7.1.1 Point-to-Point Protocol over Ethernet

V prípade strate dát pri vytváraní spojenia, nie je problém tieto informácie (číslo spojenia a MAC adresa) získať neskoršie. Počas autentifikácie sa prenáša užívateľské meno. Pri strate tohto údaju už užívateľské meno síce nezískame, ale aj tak sme schopní pokračovať v analýze protokolu. Posledným typom správ je IPCP alebo IPv6CP pre priradenie IP adresy. Tieto správy sa prenášajú síce niekoľko krát, ale iba jedna z nich nesie informáciu o priradenej IP adrese. Pri jej strate už nikdy nezískame hodnotu priradenej adresy.

7.1.2 Stateless Address Autoconfiguration

Strata paketu pri protokole SLAAC môže vzniknúť pri viacerých situáciách. Ak si stanica vygeneruje unikátnu adresu (stanica ešte nevie, že je unikátna) a odošle správu NS pre overenie unikátnosti adresy, žiadna ďalšia správa už nebude odoslaná. V tomto prípade modul nezistí priradenie adresy, avšak postupom času môže niektorá iná stanica začať komunikovať s danou IPv6 adresou. Počas tejto komunikácie sa môžu na sieti vyskytnúť správy NS a NA spojené s vyhľadaním vlastníka adresy, takže by sa priradenie adresy detegovalo s oneskorením.

Pri vygenerovaní duplicitnej adresy klientom, sú sieťou prenášané dve správy. Prvá (NS) skúma unikátnosť adresy a druhá (NA) oznamuje o jej používaní. V prípade strate prvého paketu, sa úspešne predĺži platnosť adresy klienta, ktorý aktuálne danú adresu má priradenú. Avšak pri strate druhého paketu, si bude modul myslieť, že klient vlastniaci danú adresu už adresu nepoužíva a zároveň si bude myslieť, že adresa bola priradená novému klientovi. Časom by sa síce modul mohol dozvedieť platné informácie, rovnako ako v predchádzajúcom odstavci, avšak dovtedy by bol označený vlastník IPv6 adresy niekto úplne iný.

Situácia pri chýbajúcej odpovedi na výzvu je rovnaká aj pri kontrole aktívnosti multicastových skupín od smerovača. Smerovač zisťuje všetky aktívne multicastové skupiny, avšak modul niektorú odpoveď neuvidí. Modul tak všetky IPv6 adresy spadajúce do daného multicastového rozsahu označí ako ďalej nepoužívané.

7.1.3 Simple Mail Transfer Protocol

Na každý príkaz odosielaný klientom musí server odpovedať príslušným kódom. Čiže aby sme si boli istí, že správa bola úspešné spracovaná musíme prijať obidve správy (príkaz aj odpoveď). Pri vytváraní spojenia nie je problém detegovať, že niektorá správa bola zrejme zahodená, pretože stav spojenia je posunutý o jeden krok dopredu.

Stavový automat na obrázku E.1 už počíta s tým, že počas komunikácie nemusí modul niektorý paket prijať. V prípade straty dvoch a viac paketov však modul nemusí byť schopný správne spracovať údaje a je šanca, že sa automat „zasekne“ na aktuálnom stave. Jediná činnosť, ktorú by bol schopný detegovať je ukončenie spojenia.

Ak sa automat dostane do stavu, kedy odosielateľ špecifikuje e-mailové adresy prijímateľa a odosielateľa, je situácia odlišná. Modul si ukladá indexy (*TCP Sequence number*) správ obsahujúce e-mailové adresy aj indexy správ (*TCP Acknowledgement number*), ktoré sú odpoveďami zo serveru potvrdené. Pri prechode automatu do stavu odoslania e-mailu sa vytvorí prienik týchto dvoch množín a vznikne zoznam e-mailových adries potvrdených serverom.

7.2 Ďalšie možné rozšírenia modulov

Existuje viacero pohľadov, podľa ktorých by sa dali moduly rozšíriť.

7.2.1 Zvýšenie počtu protokolov

Najzákladnejšou metódou rozšírenia je zvýšenie počtu podporovaných protokolov. V rámci TCP/IP architektúry, hlavne na aplikačnej vrstve, existuje veľké množstvo protokolov, ktoré má zmysel analyzovať. Na aplikačnej vrstve sa môže jednať o aplikačné protokoly spracovávajúce e-mailovú poštu (POP3, IMAP), protokoly spracovávajúce prenos hlasu (SIP, SCCP) a veľa ďalších.

7.2.2 Zlepšenie odolnosti proti výpadku

Aj keď sú jednotlivé moduly aj zvyšná časť IRI-IIF dôkladne testované, vždy existuje možnosť neočakávaného správania niektorej časti.

V prípade, že spadne modul, spustí sa síce automaticky znovu v priebehu krátkej chvíle, avšak všetky vnútorné informácie modulu budú stratené. V prípade pádu modulu SMTP pri odosielaní e-mailu tak všetky aktívne SMTP komunikácie budú stratené. Existuje preto možnosť, aby sa vnútorné informácie každého modulu paralelne ukladali na úložisko, ktoré aj pri páde modulu nestratí svoj obsah. Najideálnejšie je riešenie prostredníctvom databázy. Pri každom spustení modulu, by sa automaticky pozrel do databázy a v prípade neprázdneho obsahu by skopíroval obsah úložiska do svojej internej pamäti.

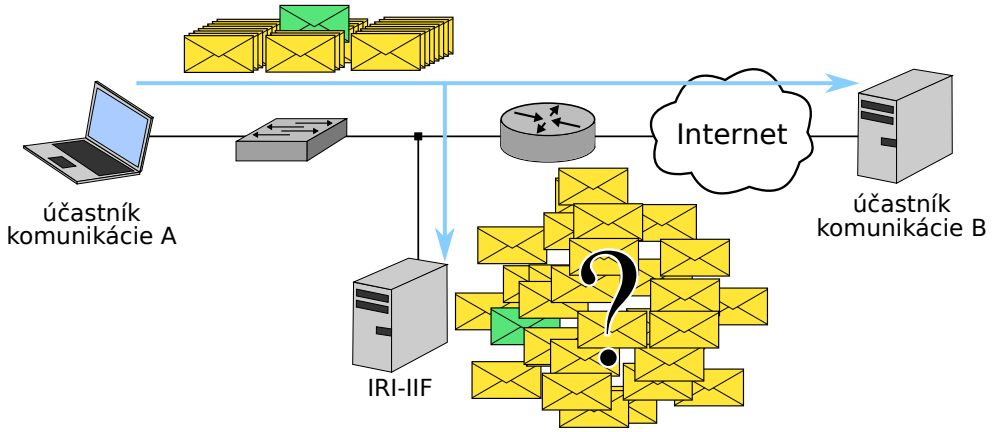
Ďalším problémom by mohol byť pád *IRI-Core*. Modul by sa snažil odosielať správy IRI, avšak pri odosielaní by sa vyskytla chyba. V prípade ak by sa táto chyba vyskytla pri súčasnej implementácii modulov, správa IRI by bola zahodená a aj všetky ďalšie až po obnovenie spojenia s *IRI-Core*. Príkladom riešenia je vytvorenie pomocného zoznamu, do ktorého by sa neúspešne odoslané IRI správy vkladali. Pri detekcii obnovenia spojenia by sa celý zoznam správ postupne odoslal do *IRI-Core* a všetky ďalšie správy by išli priamo do *IRI-Core*.

7.2.3 Ochrana proti útokom

Aj napriek tomu, že systémy pre zákonné odpočúvania sú určené na boj s kybernetickou kriminalitou, nie je vylúčené, že systémy sa nestanú cieľom útokov kybernetických útočníkov. Cieľom útoku môže byť ľubovoľná časť celého systému, aj moduly pre analýzu protokolov. Existuje veľké množstvo útokov, no popíšem ich len niekoľko.

Zahltenie modulu veľkým množstvom dát je útok, ktorý je veľmi jednoduchý na realizáciu. Útočník sa pokúsi zneužiť, že operačná pamäť alebo kapacita disku nie je na systéme, na ktorom bežia moduly neobmedzená. Modul si pre každé spojenie musí uchovávať viacero informácií, ktoré sú z modulu odstránené až pri ukončení daného spojenia. Na rozdiel od typických *Distributed Denial of Service* (DDoS) útokov tak stačí len pomocou malej šírky pásma vytvoriť veľké množstvo spojení a udržiavať ich aktívne. Riešením môže byť návrh mechanizmu, ktorý by tento typ útoku dokázal detegovať a najlepšie mu aj odolal. Príkladom riešenia môže byť detekcia veľkého množstva aktívnych spojení, ktoré nemenia svoj stav.

Ďalší typ útoku spočíva v zakrytí reálnej komunikácie v množstve falošných [19, 23, 26] ako zobrazuje obrázok 7.2. Príkladom môže byť, keď útočník chce odoslať e-mail druhému útočníkovi bez toho, aby bol odosielateľ dohľadateľný. Odosielateľ tak začne vytvárať



Obrázok 7.2: Zakrytie skutočnej komunikácie v množstve falošných

veľké množstvo falošných SMTP spojení v ktorých odošle rovnaký e-mail na tú istú cieľovú e-mailovú adresu avšak z rôznych zdrojových e-mailových adries. Medzi týmito SMTP spojeniami vytvorí aj jedno skutočné spojenie, pomocou ktorého reálne odošle e-mail na poštový server prijímateľa. Pri neskoršom zisťovaní, kto všetko komunikoval s prijímateľom danej správy by sa zobrazil výpis veľkého množstva zdrojových e-mailových adries pričom by nebolo možné zistiť skutočného odosielateľa. Riešenie by teda muselo byť založené na rozlíšení falošnej komunikácie od skutočnej.

Kapitola 8

Záver

Cieľom tejto práce bolo naprogramovanie softwaru pre detekciu identít užívateľov na počítačovej sieti, ktorý je možný zakomponovať do systému pre zákonné odpočúvanie vyvíjaný projektom Sec6Net. K tomu bolo potrebné vytvoriť software pre analýzu prietoku dát v počítačových sieťach založených na architektúre TCP/IP. Bakalárska práca je zameraná pre vybrané protokoly PPPoE, SLAAC a SMTP.

V rámci bakalárskej práve bola naštudovaná štruktúra systémov pre zákonné odpočúvanie (viď kapitola 3). Súčasťou tohto popisu bola podrobnejšia analýza bloku IRI-IIF, ktorý sa zaoberá dynamickou identifikáciou identít, a tvorbou správ IRI popisujúcich udalostí v sieti.

Každý vybraný protokol bol podrobne popísaný a analyzovaný so zameraním sa na detekciu identity používateľov. Cieľom tejto analýzy bolo zistiť, z ktorých správ a ktoré všetky identifikátory je možné zo správ detegovať. Na základe tejto analýzy bol navrhnutý stavový automat a podmienky vytvárania správ IRI (viď kapitola 4). Súčasťou analýzy bol tiež návrh vhodného umiestnenia modulu do topológie siete.

Na základe analýzy boli implementované jednotlivé modulov (viď kapitola 5). Implementácia modulov bola realizovaná v jazyku Python pod systémom Linux. Moduly boli implementované tak, aby dokázali bežať v rámci systému Sec6Net. Pretože však moduly môžu byť užitočné aj mimo systému Sec6Net, je možné ich používať aj samostatne (napr. pri správe sietí).

Testovanie výsledného softwaru prebehlo v laboratórnych podmienkach na fakulte FIT, na produkčnej sieti fakulty FIT a na internáte VUT (viď kapitola 6). Na základe výsledkov testovania boli moduly vyhodnotené a zakomponované do projektu Sec6Net. Priebežné výsledky boli spracované aj vo forme technickej správy publikovanej v rámci projektu Sec6Net [15].

Na túto bakalársku prácu by sa dalo nadviazať buď vytvorením ďalších modulov pre detekciu priraďovania a používania identifikátorov, zlepšením aktuálnych modulov proti výpadku alebo ochrane proti útokom.

Literatúra

[1] Sample Captures. [[online]], citované 12.5.2013.
URL http://wiki.wireshark.org/SampleCaptures

[2] Service Name and Transport Protocol Port Number Registry. [[online]], citované 2.5.2013.
URL http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml

[3] What is Pcapy? [[online]], citované 2.5.2013.
URL http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=tool&name=Pcapy

[4] Aboba, B.; Blunk, L.; Vollbrecht, J.; aj.: Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard), Červen 2004, updated by RFC 5247.
URL http://www.ietf.org/rfc/rfc3748.txt

[5] Conta, A.; Deering, S.; Gupta, M.: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443 (Draft Standard), Březen 2006.
URL http://www.ietf.org/rfc/rfc4443.txt

[6] Crispin, M.: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 3501 (Proposed Standard), Březen 2003.
URL http://www.ietf.org/rfc/rfc3501.txt

[7] European Telecommunications Standards Institute: *ETSI TR 102 528: Lawful Interception (LI); Interception domain Architecture for IP networks*. 10 2006, version 1.1.1.

[8] Hoffman, P.: SMTP Service Extension for Secure SMTP over Transport Layer Security. RFC 3207 (Proposed Standard), Únor 2002.
URL http://www.ietf.org/rfc/rfc3207.txt

[9] Josefsson, S.: The Base16, Base32, and Base64 Data Encodings. RFC 4648 (Proposed Standard), Říjen 2006.
URL http://www.ietf.org/rfc/rfc4648.txt

[10] Klensin, J.: Simple Mail Transfer Protocol. RFC 5321 (Draft Standard), Říjen 2008.
URL http://www.ietf.org/rfc/rfc5321.txt

[11] Klensin, J.; Freed, N.; Rose, M.; aj.: SMTP Service Extensions. RFC 1869 (Standard), Listopad 1995.
URL http://www.ietf.org/rfc/rfc1869.txt

[12] Kobierský, P.: *Hardwarová akcelerace identifikace protokolů*. Diplomová práce, Vysoké učení technické v Brně, 2008.

[13] Lloyd, B.; Simpson, W.: PPP Authentication Protocols. RFC 1334 (Proposed Standard), Říjen 1992.
URL http://www.ietf.org/rfc/rfc1334.txt

[14] Mamakos, L.; Lidl, K.; Evarts, J.; aj.: A Method for Transmitting PPP Over Ethernet (PPPoE). RFC 2516 (Informational), Únor 1999.
URL http://www.ietf.org/rfc/rfc2516.txt

[15] Martínek, T.; Kramoliš, P.; Holkovič, M.; aj.: Dynamická identifikace uživatelů v prostředí sítí IPv4 a IPv6. Technická správa FIT-TR-2012-006, Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic, 2012.

[16] McGregor, G.: The PPP Internet Protocol Control Protocol (IPCP). RFC 1332 (Proposed Standard), Květen 1992.
URL http://www.ietf.org/rfc/rfc1332.txt

[17] Mockapetris, P.: Domain names - concepts and facilities. RFC 1034 (Standard), Listopad 1987.
URL http://www.ietf.org/rfc/rfc1034.txt

[18] Mockapetris, P.: Domain names - implementation and specification. RFC 1035 (Standard), Listopad 1987.
URL http://www.ietf.org/rfc/rfc1035.txt

[19] Moore, A.; Papagiannaki, K.; Blaze, M.: Toward the Accurate Identification of Network Applications. In. *Proceedings of the Passive and Active Measurement Workshop (PAM2005)*, March/Apri 2005.

[20] Myers, J.; Rose, M.: Post Office Protocol - Version 3. RFC 1939 (Standard), Květen 1996.
URL http://www.ietf.org/rfc/rfc1939.txt

[21] Narten, T.; Draves, R.; Krishnan, S.: Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941 (Draft Standard), Září 2007.
URL http://www.ietf.org/rfc/rfc4941.txt

[22] Narten, T.; Nordmark, E.; Simpson, W.; aj.: Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), Září 2007.
URL http://www.ietf.org/rfc/rfc4861.txt

[23] Polčák, L.; Hranický, R.: Útoky na systémy pro zákonné odposlechy. Technická správa FIT-TR-2012-008, Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic, 2012.

[24] Postel, J.: Internet Protocol. RFC 791 (Standard), Září 1981.
URL http://www.ietf.org/rfc/rfc791.txt

[25] Resnick, P.: Internet Message Format. RFC 5322 (Draft Standard), Říjen 2008.
URL http://www.ietf.org/rfc/rfc5322.txt

[26] Sen, S.; Spatscheck, O.; Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA: ACM, 2004, s. 512–521.

[27] Simpson, W.: The Point-to-Point Protocol (PPP). RFC 1661 (Standard), Červenec 1994.
URL http://www.ietf.org/rfc/rfc1661.txt

[28] Thomson, S.; Narten, T.; Jinmei, T.: IPv6 Stateless Address Autoconfiguration. RFC 4862 (Draft Standard), Září 2007.
URL http://www.ietf.org/rfc/rfc4862.txt

[29] Wong, M.; Schlitt, W.: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental), Duben 2006.
URL http://www.ietf.org/rfc/rfc4408.txt

[30] Zorn, G.: Microsoft PPP CHAP Extensions, Version 2. RFC 2759 (Informational), Leden 2000.
URL http://www.ietf.org/rfc/rfc2759.txt

Dodatok A

Zoznam použitých skratiek

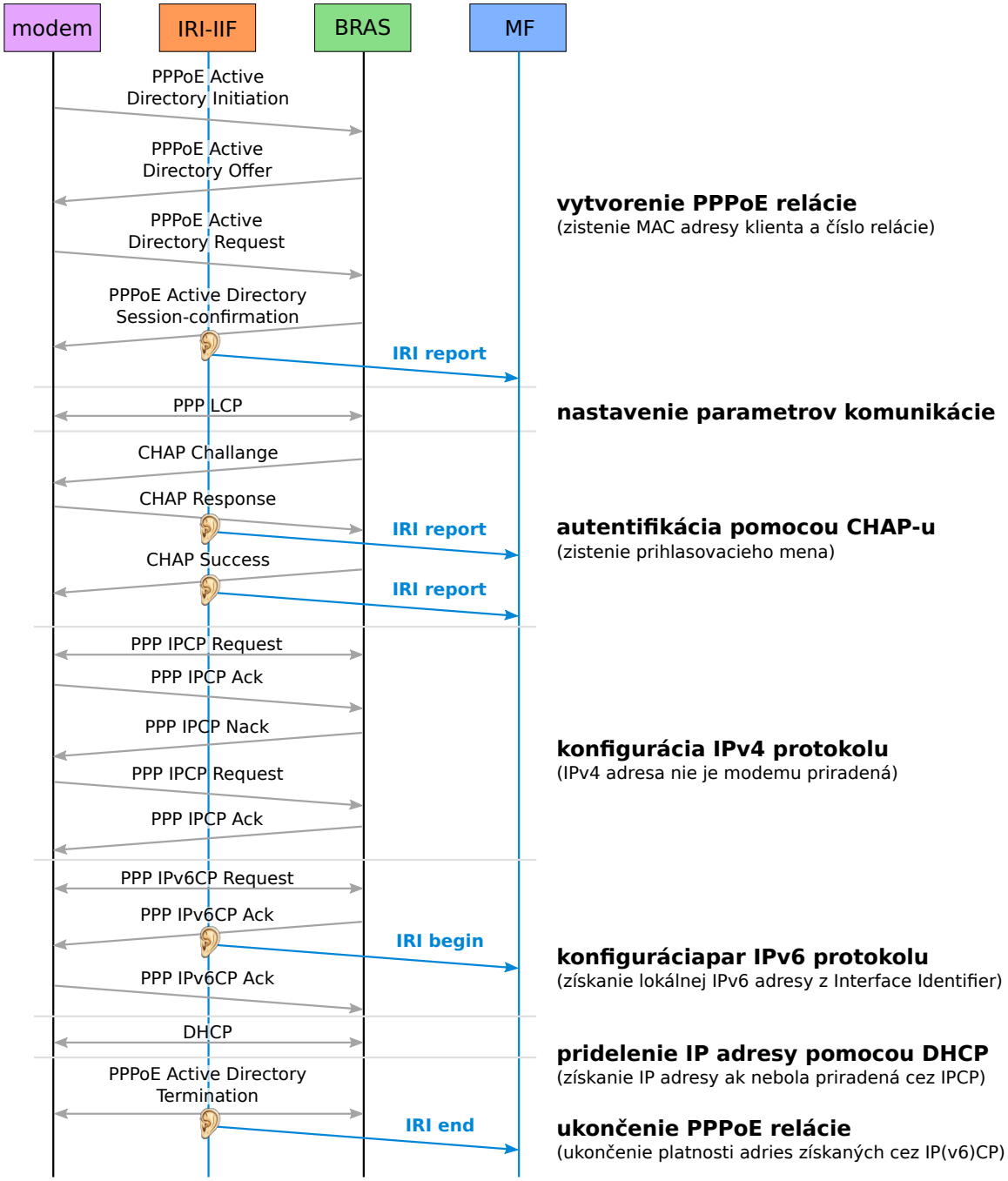
AF	Administration Function
BRAS	Broadband Remote Access Server
CC-IIF	Contents of Communication Internal Interception Function
CCTF	Contents of Communication Trigger Function
DAD	Duplicate Address Detection
DDOS	Distributed Denial of Service
DNS	Domain Name System
DSLAM	Digital Subscriber Line Access Multiplexers
DSL	Digital Subscriber Line
ESMTP	Extended Simple Mail Transfer Protocol
ETSI	European Telecommunications Standards Institute
IAP	Intercept Access Point
ICMPv6	Internet Control Message Protocol for the Internet Protocol Version 6
IMF	Internet Message Format
IPCP	IP Control Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPv6CP	IPv6 Control Protocol
IRI	Intercept Related Information
IRI-IIF	Intercept Related Information Internal Interception Function
ISP	Internet Service Provider
LCP	Link Control Protocol
LEA	Law Enforcement Agency
MDA	Mail Delivery Agent
MF	Mediation Function
MSA	Mail Submission Agent
MTA	Mail Transfer Agent
MUA	Mail User Agent
NA	Neighbor Advertisement
NCP	Network Control Protocol
ND	Neighbor Discovery

NS	Neighbor Solicitation
OS	operačný systém
PADS	PPPoE Active Discovery Session confirmation
PADT	PPPoE Active Discovery Termination
PPPoE	Point-to-Point Protocol over Ethernet
PPP	Point-to-Point Protocol
RA	Router Advertisement
Sec6Net	projekt Moderné prostriedky pre boj s kybernetickou kriminalitou na Internete novej generácie
SLAAC	Stateless Address Autoconfiguration
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol

Dodatok B

Príklad PPPoE spojenia

Na obrázku B.1 je zobrazený príklad nadviazania a ukončenia spojenia PPPoE. Pri zostavovaní spojenia je použitá autentifikácia protokolom CHAP, IPv4 adresa je namiesto protokolom IPCP priradená protokolom DHCP a oba konce spojenia podporujú protokol IPv6. Ako je možné na obrázku vidieť, modul neposiela pri pridelení adresy DHCP protokolom žiadne správy IRI. Aj napriek tomu, že je adresa priradená nad protokolom PPP, jedná sa o protokol DHCP ku ktorému je nutné vytvoriť vlastný modul pre IRI-IIF.



Obrázok B.1: Ukážkové nadviazanie a ukončenie spojenia typu PPPoE spolu so zobrazením IRI-IIF správ

Dodatok C

Zoznam vytváraných IRI správ

Pre každý protokol v kapitole 4 sú pri popise činnosti bloku IRI-IIF uvedené IRI správy, ktoré sú počas analýzy každého protokolu vytvárané. Obsahom tohto dodatku je stručné zhrnutie týchto správ spolu s popisom detegovanej udalosti, typom správy IRI a zoznamom nových identifikátorov, ktoré boli pri danej udalosti detegované.

C.1 Point-to-Point Protocol over Ethernet

Udalosť	**Typ správy**	**Identifikátory**
Klient nadviazal spojenie s BRASom	Report	číslo PPP relácie, MAC adresa
Užívateľ sa pokúša autentizovať	Report	prihlasovacie meno
Užívateľ sa úspešne autentizoval	Begin	-
Ukončenie spojenia pred priradením adresy alebo autentizáciou	Report	-
Autentizovanému užívateľovi bola priradená adresa protokolom PPPoE	Continue	IP adresa
Neautentizovanému užívateľovi bola priradená adresa protokolom PPPoE	Begin	IP adresa
Ukončenie spojenia po priradení adresy alebo autentizácii	End	-

C.2 Stateless Address Autoconfiguration

Udalosť	**Typ správy**	**Identifikátory**
Overená unikátnosť IPv6 adresy, ktorá nie je uložená	Begin	MAC adresa, IPv6 adresa
Overená unikátnosť IPv6 adresy, ktorá už je uložená	Continue	-
Overená unikátnosť IPv6 adresy, ktorá už je uložená ale pre inú MAC adresu	End	-
Oznámenie o používaní novej IPv6 adresy	Begin	MAC adresa, IPv6 adresa
Oznámenie o používaní IPv6 adresy, ktorá je uložená	Continue	-
Oznámenie o používaní IPv6 adresy, ktorá je uložená ale pre inú MAC adresu	End	-
Odhlásenie klienta z multicastovej skupiny	End	-
Žiadna odpoveď na správu Multicast Listener Query	End	-
Adresa nebola dlhodobo videná na sieti	End	-

C.3 Simple Mail Transfer Protocol

Udalosť	**Typ správy**	**Identifikátory**
Vytvorenie spojenia s SMTP serverom	Report	IP adresy, čísla portov
Odmietnutie spojenia SMTP serverom	Report	IP adresy, čísla portov
Predčasné ukončenie spojenia (pred IRI Begin)	Report	-
Neúspešná autentifikácia klienta	Report	-
Úspešná autentifikácia klienta	Begin	-
Ukončenie spojenia (po IRI Begin)	End	-
Úspešné odoslanie e-mailu	Continue	e-mailové adresy, počet príjemcov
Neúspešné odoslanie e-mailu	Continue	e-mailové adresy, počet príjemcov
Prepnutie do šifrovaného módu	Report	-

Dodatok D

Posielanie správy protokolom SMTP

Pri popise protokolu v kapitole 3.4 je zobrazený stručný postup pri odosielaní správy protokolom SMTP. V nasledujúcom zozname je celý postup popísaný podrobne:

1. Klient nadviaže spojenie pomocou protokolu TCP na port, na ktorom beží služba SMTP (štandardne č. 587). Klient zatiaľ neposiela žiadne príkazy a čaká na odpoveď.
2. Server len na základe nadviazania TCP spojenia deteguje pripojeného užívateľa a odpovie s kódom 220 reprezentujúci, že služba na serveri beží a je pripravená prijímať požiadavky.
3. Klient odošle príkaz *Extended Hello* (EHLO) pomocou ktorej žiada server o jeho identifikovanie a zároveň tým inicializuje SMTP konverzáciu.
4. Server odpovie správou, ktorej obsahom bude séria kódov 250. Jednotlivé kódy sú oddelené zakončením riadku (znakmi CRLF) pričom každý kód obsahuje iný parameter. Obsahom týchto parametrov je identifikátor servera a zoznam podporovaných rozšírení na strane servera. Príkladom rozšírenia je parameter *AUTH* uvádzajúci podporované metódy autentifikácie.
5. V prípade, že server aj klient podporuje šifrovanie (rozšírenie STARTTLS) a klient ho zároveň aj preferuje, odošle klient príkaz STARTTLS. Všetka nasledujúca komunikácia bude šifrovaná. Ďalším krokom bude bod č. 21.
6. Klient odošle príkaz *AUTH* s parametrom o ktorú metódu autentifikácie má záujem. Podporované metódy sú *PLAIN* - prihlasovacie meno a heslo odoslané nešifrovane ako jeden textový reťazec, *LOGIN* - prihlasovacie meno a heslo odoslané taktiež nešifrovane, avšak pomocou 2 správ a *CRAM-MD5* - prihlasovacie meno je spolu so zahashovaným heslom odoslané v jednej správe. V našom prípade klient odošle príkaz *AUTH LOGIN*.
7. Server odpovie kódom 334 a parametrom obsahujúci text „Username:“. Parameter je kódovaný pomocou formátu *base64* [9].
8. Na základe výberu metódy autentizácie *LOGIN*, klient odošle na server nešifrované užívateľské meno. Hodnota je zakódovaná vo formáte base64.

9. Server znovu odpovie s kódom 334, teraz ale parameter obsahuje text „Password:“. Parameter je taktiež kódovaný pomocou formátu *base64*.

10. Klient odošle na server nešifrované užívateľské heslo, zakódované vo formáte *base64*.

11. Server následne odpovie kódom 235 (autentizácia úspešná) alebo 535 (autentizácia neúspešná). Pri úspešnej autentizácii je týmto kompletne nadviazané spojenie a užívateľ môže začať špecifikovať parametre správy.

12. Klient príkazom *MAIL* špecifikuje e-mail odosielateľa správy.

13. Server potvrdí správnosť údaju kódom 250.

14. Klient odošle príkaz *RCPT TO* obsahujúci parameter e-mail prijímateľa.

15. Server potvrdí správnosť údaju kódom 250.

16. Klient môže posledný príkaz *RCPT TO* opakovať viac krát, podľa počtu prijímateľov danej správy, pričom server odpovie s príslušným kódom na každý príkaz samostatne. Ak klient uviedol všetkých prijímateľov, odošle príkaz *DATA*. Príkazom *DATA*, dáva klient najavo, že obsahom nasledujúcich správ bude samotný e-mail zakódovaný vo formáte IMF.

17. Server pomocou kódu 354, dá najavo, že očakáva text správy.

18. Klient následne odošle sériu správ, ktorých obsahom bude segmentovaná odosielaná správa. Po dokončení prenosu správy odošle ďalší segment správy s obsahom „.“. Týmto oznamuje serveru koniec posielania správy.

19. Server potvrdí príjem správy kódom 250 a správu odošle. V prípade, že má klient viac e-mailov na odoslanie, nemusí spojenie ukončovať a vytvárať znovu ďalšie. Ak chce klient odoslať ďalšiu správu, pošle znovu príkaz *MAIL* a tým sa spojenie vracia späť do bodu č. 12.

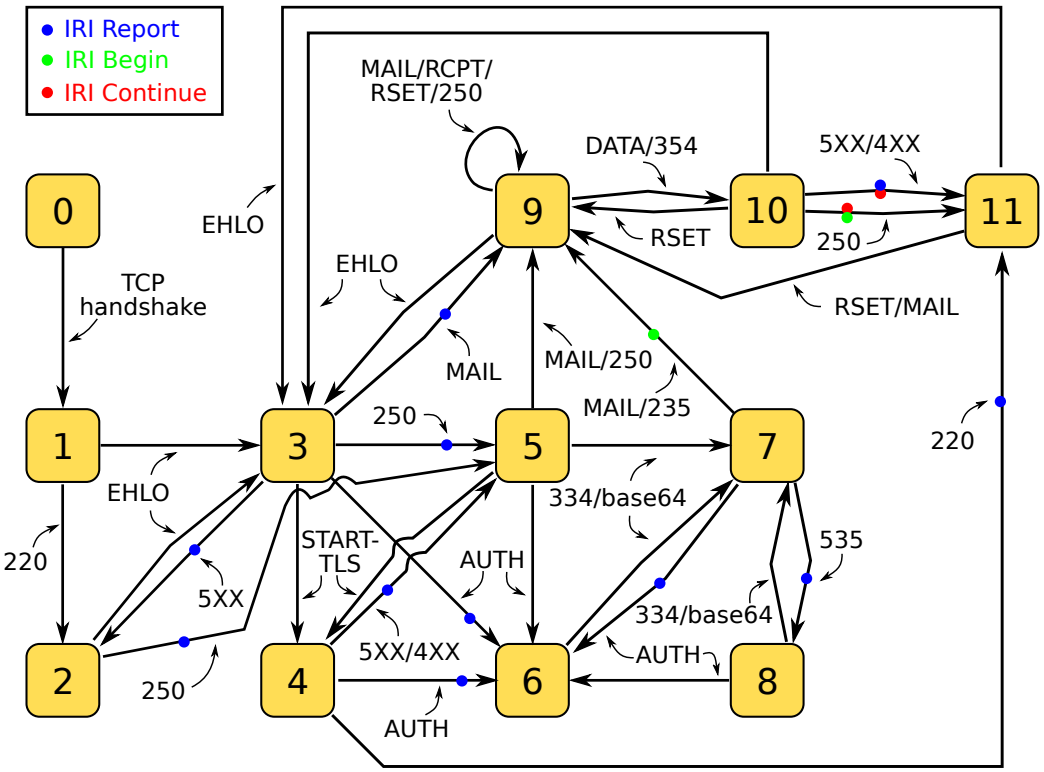
20. Ak klient už nemá žiadne ďalšie správy na odoslanie, odošle príkaz *QUIT*. Všetky akcie sú v réžii klienta a preto aj ukončenie spojenia vždy musí inicializovať klient.

21. Server potvrdí žiadosť o ukončenie spojenia kódom 221 a ukončí spojenie.

Dodatok E

Stavový automat modulu SMTP

V kapitole 4.3 je popísaný zjednodušený stavový automat protokolu SMTP spolu s jeho vyobrazením na obrázku 4.6. Úplný stavov automat, ktorý taktiež počíta s neúplnosťou vstupných dát, je zobrazený na obrázku E.1. To či je posielaná správa IRI, je znázornené pomocou krúžkov na prechodoch medzi stavmi. Farba krúžku upresňuje o aký typ IRI správy sa jedná. Pri prechode medzi stavom č. 10 a č. 11 sú znázornené dva krúžky rôznej farby. To aká správa sa odošle závisí podľa toho či už bola odoslaná správa IRI Begin alebo nie tak ako je to popísané v kapitole 2.3.



Obrázok E.1: Úplný stavový automat modulu SMTP