

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

Študijný program: Informačné systémy (9.2.6.)

Zameranie: Informačno-komunikačné siete

Číslo práce: 28360320122174

Bc. Michal Paulus

**Riešenia SIP komunikácie cez zariadenia
s mechanizmom NAT**

**Solutions supporting mechanisms of SIP
NAT traversal**

Vedúci : Ing. Pavel Segeč, PhD.

Reg. číslo : 174/2011

Máj 2012

Veľký Diel
010 26 Žilina

ZADANIE TÉMY DIPLOMOVEJ PRÁCE.

Študijný program : Informačné systémy

Zameranie: Informačno-komunikačné siete

Meno a priezvisko

Michal Paulus

Osobné číslo

552826

Názov práce v slovenskom aj anglickom jazyku

Riešenia SIP komunikácie cez zariadenia s mechanizmom NAT

Solutions supporting mechanisms of SIP NAT traversal.

Zadanie úlohy, ciele, pokyny pre vypracovanie

(Ak je málo miesta, použite opačnú stranu)

Cieľ diplomovej práce:

Cieľom práce je analyzovať súčasný stav riešenia problematiky a implementácie mechanizmov prechodu SIP cez NAT ako STUN, TURN a ICE. Na základe analýzy navrhnete a implementujete mechanizmus ICE.

Obsah:

Pri riešení práce sa zamerajte na:

- Analýza problematiky SIP komunikácie s použitím UDP, TCP alebo TLS v prostredí sietí s NAT.
- Analýza a popis činnosti jednotlivých štandardizovaných riešení ako STUN, TURN a ICE.
- Analýza vhodných riešení komunikačných knižníc podporujúcich vývoj STUN, TURN a ICE riešení.
- Návrh implementácie klienta s ICE mechanizmom.
- Realizácia, testovanie a vyhodnotenie riešenia v sieťovom prostredí.

Témy z predmetov študijného zamerania

Meno a pracovisko vedúceho DP:

Meno a pracovisko tútora DP:

5SI031: 3, 4, 5

Ing. Pavel Segeč, PhD.

21.10.2011

vedúci DP
(dátum a podpis)

tútor
(dátum a podpis)

21.10.2011

vedúci katedry
(dátum a podpis)

garant
(dátum a podpis)

Zadanie zaregistrované dňa 21. 10. 2011 pod číslom 174/2011 podpis

Abstrakt

PAULUS Michal : *Riešenia SIP komunikácie cez zariadenia s mechanizmom NAT*. [Diplomová práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky. Katedra informačných sietí. – Vedúci práce: Ing. Pavel Segeč, PhD. – Stupeň odbornej kvalifikácie: inžinier. – Žilina: FRI ŽU v Žiline 2012. – 72 strán.

Cieľom tejto diplomovej práce je poskytnúť čitateľovi komplexnú charakteristiku mechanizmov prechodu SIP cez NAT. Vybranými mechanizmami sú štandardizované riešenia STUN, TURN a ICE. Práca obsahuje teoretický popis jednotlivých riešení spolu s možnosťami ich využitia. Súčasťou práce sú aj výsledky z praktického testovania aplikácií, ktoré podporujú využitie niektorého z protokolov STUN, TURN alebo ICE. V rámci praktickej časti práce bol navrhnutý, implementovaný a testovaný aj vlastný klient s podporou ICE mechanizmu. Praktické skúsenosti s implementáciou a testovaním vlastného riešenia sú rovnako popísané v práci.

Kľúčové slová: SIP, prechod SIP cez NAT, STUN, TURN, ICE

Abstract

PAULUS Michal : *Solutions supporting mechanisms of SIP NAT traversal*.
[Diploma thesis] – University of Žilina. Faculty of Management Science and Informatics.
Department of InfoComm networks. – Adviser: Ing. Pavel Segeč, PhD. – Qualification
degree: master. – Žilina: FRI ŽU in Žilina 2010. – 72 pages.

The main aim of this Diploma thesis is to offer reader complex characteristic of SIP NAT traversal mechanisms. Selected mechanisms are standardized solutions STUN, TURN and ICE. The work includes a theoretical description of each solution along with the possibilities of its use. Results of testing applications, which support use of protocols STUN, TURN and ICE, are included in work. The work also includes the results of practical testing applications that support the use of protocols STUN, TURN and ICE. The practical part of this thesis was design, implement and test a custom client with ICE mechanism. Practical experience with implementation and testing of custom solution are described in the work.

Key words: SIP, SIP NAT traversal, STUN, TURN, ICE

Pod'akovanie

Moje pod'akovanie patrí vedúcemu diplomovej práce Ing. Pavlovi Segečovi, PhD. za odborné vedenie a pripomienky pri tvorbe tejto práce. Rovnako sa chcem za podporu pod'akovať rodine a priateľke.

OBSAH

1	ÚVOD.....	- 1 -
2	ÚVOD DO PROBLEMATIKY	- 2 -
2.1	SIP – SESSION INITIATION PROTOCOL.....	- 2 -
2.2	RTP – REAL-TIME TRANSPORT PROTOCOL.....	- 5 -
2.3	SDP – SESSION DESCRIPTION PROTOCOL	- 5 -
2.4	NAT – NETWORK ADDRESS TRANSLATOR.....	- 7 -
2.4.1	ZÁKLADNÉ NAT.....	- 7 -
2.4.2	NAPT.....	- 8 -
2.4.3	ĎALŠIE TYPY NAT	- 9 -
2.5	NAT PROBLÉMY PRI SIP	- 9 -
3	MECHANIZMY PRECHODU SIP CEZ NAT	- 11 -
3.1	STARÁ ŠPECIFIKÁCIA STUN – KLASICKÝ STUN	- 11 -
3.1.1	VARIÁCIE NAT	- 12 -
3.1.2	FORMÁT, TYP A ATRIBÚTY STUN SPRÁVY.....	- 14 -
3.1.3	URČENIE VARIÁCIE NAT.....	- 15 -
3.2	NOVÁ ŠPECIFIKÁCIA STUN – SESSION TRAVERSAL UTILITIES FOR NAT.....	- 19 -
3.2.1	FORMÁT A TYP STUN SPRÁV	- 20 -
3.2.2	ATRIBÚTY SPRÁV.....	- 21 -
3.2.3	STUN KOMUNIKÁCIA	- 23 -
3.2.4	ROZDIELY ŠPECIFIKÁCIÍ STUN - ZHRNUTIE	- 24 -
3.3	ROZŠÍRENIE TURN – TRAVERSAL USING RELAYS AROUND NAT	- 25 -
3.3.1	PRIDANÉ METÓDY A ATRIBÚTY PRE TURN	- 25 -
3.3.1.1	Nové metódy správ.....	- 26 -
3.3.1.2	Pridané atribúty správ.....	- 26 -
3.3.2	TURN KOMUNIKÁCIA.....	- 27 -
3.3.3	POUŽITIE TURN – ZHRNUTIE	- 32 -
3.4	ICE – INTERACTIVE CONNECTIVITY ESTABLISHMENT	- 33 -
3.4.1	ICE PREDPOKLADY	- 33 -

3.4.2	ZBER KANDIDÁTOV	- 34 -
3.4.3	TESTOVANIE KANDIDÁTOV	- 37 -
3.5	RIEŠENIA PRECHODU SIP CEZ NAT MIMO KLIENTA	- 42 -
4	<u>REALIZÁCIA A TESTOVANIE</u>	- 44 -
4.1	SERVEROVÁ STUN/TURN ENTITA	- 44 -
4.2	ANALÝZA PROBLEMATIKY – MOŽNÉ SCENÁRE	- 45 -
4.3	TESTOVANIE VOĽNE DOSTUPNÝCH IMPLEMENTÁCIÍ STUN, TURN A ICE	- 46 -
4.3.1	STARÁ ŠPECIFIKÁCIA STUN – KLASICKÝ STUN	- 46 -
4.3.2	NOVÁ ŠPECIFIKÁCIA STUN	- 48 -
4.3.3	TURN	- 51 -
4.3.4	ICE	- 54 -
4.4	VLASTNÁ IMPLEMENTÁCIA ICE	- 58 -
4.4.1	ANALÝZA DOSTUPNÝCH KNIŽNÍC	- 58 -
4.4.2	NÁVRH IMPLEMENTÁCIE	- 58 -
4.4.3	IMPLEMENTÁCIA RIEŠENIA	- 60 -
4.4.4	TESTOVANIE IMPLEMENTÁCIE	- 63 -
5	<u>ZÁVER.....</u>	- 71 -
	<u>POUŽITÉ ZDROJE.....</u>	- 73 -
	<u>ZOZNAM POUŽITÝCH SKRATIEK.....</u>	- 74 -
	<u>PRÍLOHY</u>	- 75 -

1 ÚVOD

Potreba riešenia SIP komunikácie cez zariadenia s mechanizmom NAT prichádza najmä z dôvodu čoraz väčšieho nasadzovania SIP architektúr v rôznych typoch sietí. SIP ako stabilná a odskúšaná technológia s množstvom otvorených produktov dokáže ponúknuť viacero typov služieb prostredníctvom IP protokolu. Medzi poskytované služby môžeme zaradiť, okrem iného, aj prenos multimediálnych dát medzi dvomi koncovými zariadeniami. Základom pre SIP architektúru je oddelenie signalizačnej časti komunikácie od samotnej výmeny multimediálnych dát. Práve úspešná výmena dát medzi dvomi koncovými bodmi spojenia predstavuje problém, ktorý je predmetom riešenia tejto diplomovej práce. Podstatou problému je nedosiahnuteľnosť zariadení, ktoré sú v rámci lokálnych sietí umiestnené za zariadením s mechanizmom NAT a majú teda pridelené privátne IP adresy. Cieľom tejto práce je preskúmať súčasné riešenia prechodu SIP signalizácie cez NAT s dôrazom na úspešnosť založenia toku médií, ktoré predstavujú štandardizované riešenia STUN, TURN a ICE. Po dohode s vedúcim práce bola problematika skúmania, z dôvodu rozsiahlosti témy, zúžená na transportný protokol UDP.

Práca je rozdelená do piatich kapitol spolu s úvodom a záverom. V kapitole číslo dva sú popísané základné protokoly a mechanizmy, ktoré sú využívané v rámci SIP architektúry. V tejto kapitole môžeme rovnako nájsť aj základnú definíciu NAT mechanizmov a popis problémov, ktoré tieto zariadenia predstavujú pre architektúru SIP.

Kapitola číslo tri obsahuje popis jednotlivých štandardizovaných riešení prechodu cez zariadenia s mechanizmom NAT. Riešenia v tomto prípade predstavujú protokoly STUN, TURN a ICE. Pri každom z protokolov je uvedená jeho základná charakteristika spolu so spôsobom riešenia problematiky prechodu cez NAT. Pri popise sa zameriavame aj na implementačné vlastnosti jednotlivých protokolov (napr. formát a typ správ), ktoré boli využité v praktickej časti práce v rámci implementácie vlastného klienta s ICE mechanizmom.

V kapitole číslo štyri sa práca zameriava na testovanie aplikácií, ktoré aktuálne podporujú využitie štandardizovaných riešení STUN, TURN a ICE v rámci SIP architektúry. Súčasťou tejto kapitoly je aj popis návrhu, implementácie a testovania vlastného klienta s ICE mechanizmom, ktorý je predmetom tejto práce.

2 ÚVOD DO PROBLEMATIKY

V tejto kapitole popíšeme služby, štandardy, protokoly a mechanizmy, ktoré sú kľúčové pre analýzu zadanej témy. Definície a popisy v jednotlivých podkapitolách sú zamerané hlavne na časti špecifikácii, ktoré súvisia s témou diplomovej práce a bližšie informácie je možné nájsť v ďalšej dostupnej literatúre, prípadne v IETF RFC štandardoch. Číslo príslušného štandardu bude uvedené pri každej podkapitole, ak ide o protokol definovaný v RFC.

2.1 SIP – Session Initiation Protocol

SIP verzia 2.0 (prvá verzia SIP 1.0 bola definovaná v RFC 2543) je v **RFC 3261** (1) definovaný ako signalizačný protokol aplikačnej vrstvy ISO OSI modelu, ktorý slúži na **vytváranie, modifikovanie a ukončovanie** spojení medzi dvomi alebo viacerými účastníkmi. Spojeniami sú myslené telefónne, multimediálne (hlas, video) a konferenčné hovory.

SIP vznikol ako jednoduchý **signalizačný** protokol, ktorý mal slúžiť viacerým aplikáciám na internete na **vytváranie a manažovanie spojení**. Nemá však slúžiť ako integrovaný komunikačný systém ale skôr ako **komponent** pre iné protokoly na vytvorenie kompletnej komunikačnej architektúry. Medzi tieto protokoly patria najmä RTP – Real Time Transport Protocol **na prenos dát v reálnom čase**, SDP – Session Description Protocol slúžiaci na **popis multimediálnych spojení** (oba protokoly budú bližšie popísané v ďalších častiach práce) ako aj ďalšie (RTSP - Real Time Streaming Protocol, MEGACO - Media Gateway Control Protocol). Ako vyplýva z predchádzajúcej časti, SIP ako taký, neposkytuje službu ale skôr základy pre implementáciu iných služieb. Napríklad, dokáže lokalizovať užívateľa a doručiť mu potrebný popis spojenia v SDP. Rovnako tak môže byť implementovaná služba na identifikáciu volajúceho, kde v SIP môže byť doručené ID, prípadne fotografia volajúceho, spolu s popisom média cez SDP (1).

V SIP je použitý princíp **architektúry klient/server**, kde klient posiela požiadavky na server a server spracováva tieto požiadavky. **Rolu servera** alebo **klienta** môžu zastupovať oba typy logických koncových bodov spojenia, ktoré sú v SIP definované (1):

- **UA (User Agent)** – logická koncová SIP entita, ktorej hlavnou funkciou je zostavenie a manažovanie spojenia s inou koncovou SIP entitou. UA môže byť implementovaná ako v **softvérovom** (Obrázok 2.1), tak v **hardvérovom** (Obrázok 2.2) SIP telefóne.



Obrázok 2.1 - softvérový SIP telefón



Obrázok 2.2 - hardvérový SIP telefón

- **Server** – SIP serverová entita, ktorá spracováva SIP žiadosti, na ktoré odpovedá, prípadne ich preposiela k iným SIP entitám. Rozlišujeme tri typy serverov: **Registrar** – spracováva žiadosti o registráciu, **Proxy** – smeruje žiadosti klientov k inej (bližšej) SIP entite a **Redirect** – vráti SIP klientovi odkaz na alternatívny server.

Obe vyššie popísané entity môžu pracovať ako **klient** (UAC), v prípade, že entita vytvára a odosiela SIP žiadosti. Naopak entita pracuje ako **server** (UAS) ak prijíma, spracúva a odpovedá na žiadosti klienta.

Transakčne orientovaný požiadavka/odpoveď (z ang. request/response) model, ktorý využíva na komunikáciu správy je charakteristický ako pre HTTP protokol, tak pre SIP. Správy v SIP sú textovo kódované a ich štýl a syntax je podobná aplikačným správam protokolu HTTP. Základné typy SIP správ sa delia na požiadavky/metódy a odpovede (1):

- **Požiadavky** – SIP metódy, ktoré sú najčastejšie používané na iniciovanie určitej akcie alebo vyžiadanie služby. Základné SIP metódy sú:
 - **Register** – registrácia účastníka na Registrar serveri (oznámenie aktuálnej IP adresy – „polohy“ UA).
 - **Invite** – zahájenie komunikácie, žiadosť o založenie spojenia.
 - **Ack** – potvrdenie spoľahlivej výmeny správ.
 - **Bye** – ukončenie spojenia medzi účastníkmi.

- **Cancel** – prerušenie aktuálne zostavovaného spojenia.
- **Options** – žiadosť o informácie od volaného bez nutnosti zostavenia spojenia.
- **Odpovede** – slúžia na potvrdenie prijatia požiadavky a informujú o stave jej spracovania. SIP odpovede podobne ako v HTTP využívajú rozdelenie do „stovkových“ tried. Význam jednotlivých tried je nasledovný:
 - **Informačné (1xx)** – požiadavka bola prijatá a je spracovávaná.
 - *100 Trying, 180 Ringing*
 - **Úspech (2xx)** – informuje o akceptovaní požiadavky a o jej úspešnom spracovaní.
 - *200OK*
 - **Presmerovanie (3xx)** – na dokončenie spracovania požiadavky je potrebná ďalšia akcia na strane klienta.
 - **Chyba klienta (4xx)** – požiadavka bola zle formulovaná alebo nemôže byť obslužená na danom SIP serveri.
 - *400 BadRequest, 401 Unauthorized, 404 NotFound, ...*
 - **Chyba servera (5xx)** – server nedokáže splniť požiadavku, aj keď je správne formulovaná.
 - *500 Server InternalError, 501 NotImplemented, ...*
 - **Globálna chyba(6xx)** – spracovanie požiadavky zlyhalo a nemôže byť vykonané na žiadnom serveri.
 - *600 BusyEverywhere, 606 NotAcceptable, ...*

Vyššie uvedené typy správ využívajú SIP koncové body na založenie multimediálneho spojenia. Časť multimediálneho spojenia môže predstavovať napríklad hlasový hovor, ktorý je posielať v RTP medzi dvomi SIP telefónmi. SIP protokol slúži na určenie základných parametrov spojenia dvoch (alebo viacerých) koncových bodov spojenia. Medzi základné parametre môžeme zaradiť **transportnú adresu a kodek**. Transportná adresa je kombinácia IP adresy a čísla portu konkrétneho transportného protokolu koncového bodu, z ktorej bude odosielať a prijímať RTP tok dát. Kodek určuje ako bude hlas kódovaný na tok dát a opačne tok dát dekódovaný na hlas. Uvedené parametre spojenia sú zapísané v SDP tele SIP správy a pomocou SIP správ doručené od jedného SIP koncového bodu k inému. Po výmene SIP správ dokáže SIP koncový bod založiť RTP tok dát na transportnú adresu iného koncového bodu s príslušným kódovaním

hlasu (v prípade, že každý klient má minimálne jednu transportnú adresu a minimálne jeden vhodný kodek).

Koncové body SIP spojenia z pohľadu výmeny signalizácie komunikujú na predvolenom porte **5060** pre **TCP**, **UDP** a **SCTP** pri nezabezpečenom type prevádzky a predvolene na porte **5061** pri šifrovanom type (TLS).

2.2 RTP – Real-Time Transport Protocol

Hlasové alebo iné dáta odosielané od jedného koncového bodu k inému pri VoIP prevádzke sú bežne posielané pomocou RTP protokolu. RTP protokol bol špeciálne vytvorený pre VoIP ako pomocný protokol na vyššej úrovni (aplikačná vrstva), ktorý sa vkladá do segmentov už existujúcich transportných protokolov (predvolene UDP). RTP poskytuje na rozdiel od UDP navyše tieto funkcie: **identifikácia typu obsahu** a **časovú synchronizáciu**. Výhodou, ktorú predstavuje identifikácia typu obsahu je, že na základe RTP hlavičky je možné určiť napríklad akým typom kodeku sú kódované hlasové dáta. Časová synchronizácia a k nej príslušné očíslovanie RTP segmentov slúži napríklad na určenie duplicitných, stratených alebo preusporiadaných segmentov (aplikácia sa tak môže napríklad pokúsiť zamaskovať stratu určitých hlasových segmentov).

RTP je štandardizovaný protokol IETF, ktorý je dokumentovaný v **RFC 3550** (2) a bol špeciálne vytvorený, ako z názvu vyplýva, pre prenos dát v reálnom čase vhodný rovnako pre **unicast** aj **multicast** a zároveň nezávislý od signalizačného protokolu (SIP, H.323). Ako podporný protokol pre RTP bol definovaný **RTCP – RTP Control Protocol**, ktorý slúži na monitorovanie prevádzky, zber štatistických údajov, zabezpečenie QoS, rýchlejšie zotavenie spojenia z kolízie a ďalšie funkcie. RTCP je rovnako ako RTP vkladá do UDP segmentov. Bežne je komunikácia vytváraná a prijímaná na **párnych číslach portov** pre RTP a najbližšom vyššom **nepárnom čísle portu** pri RTCP.

2.3 SDP – Session Description Protocol

Pri zakladaní multimediálnych spojení je potrebné špecifikovať aký **typ média** sa bude prenášať, definovať **transportné adresy** a iné dáta popisujúce spojenie zúčastnených strán. Práve protokol SDP, štandardizovaný IETF a popísaný v **RFC 4566** (3), poskytuje priestor pre reprezentáciu týchto informácií bez ohľadu na to akým spôsobom budú informácie zapísané v SDP prenášané. Na prenos SDP je teda potrebný iný protokol,

v ktorom budú zapuzdrené SDP dáta. Pre celú SIP architektúru je to práve SDP protokol, v ktorom sú posielané informácie potrebné pre založenie multimediálneho spojenia medzi dvomi SIP koncovými bodmi.

Model **ponuka/odpoved'** (z ang. offer/answer model), dokumentovaný v **RFC 3264** (4), ponúka základnú štruktúru na dohodu kompatibilných typov médií, kodekov medzi zúčastnenými stranami komunikácie tzv. **agentmi** (napr. SIP telefón). V princípe ide o výmenu dostupných spôsobov komunikácie zo strany jedného agenta k inému, či už ide o IP adresu, číslo portu, typ média, zoznam dostupných kodekov, ICE kandidátov (bližšie v kapitole ICE) a iných parametrov spojenia obsiahnutých v SDP. Týmto spôsobom dokážu volaný a volajúci určiť (ak existuje) najvhodnejší spôsob prenosu multimediálnych dát.

Popis spojenia v SDP obsahuje tieto základné informácie (3):

- **typ média** – video, audio a iné.
- **Transportný protokol** – RTP/UDP/IP, H.320 a iné.
- **Formát média/kodek** – MPEG, G.711, H.261 a iné.
- **Adresa** – IP adresa koncového bodu pre komunikáciu.
- **Port** – port koncového bodu pre komunikáciu.

Uvedené informácie sú zapísané v textovom formáte kódovania UTF-8 vo viacerých riadkoch SDP správy. Jednotlivé riadky majú tvar:

<typ> = <hodnota> ,

kde *<typ>* určuje jediným znakom typ parametra a *<hodnota>* je závislá od typu parametra. V nasledujúcej časti sú popísané niektoré typy dôležité pre potreby tejto práce:

- „v=“ – určuje verziu protokolu SDP.
- „o=“ – vlastník a identifikátor spojenia.
- „s=“ – názov spojenia.
- „c=“ – informácie potrebné pre komunikáciu (IPv4/IPv6, IP adresa).
- „m=“ – popis média, meno a adresa (audio/video/text, číslo portu).
- „a=“ – atribút bližšie určuje spojenie alebo médium.

Príklad SDP správy so základnými poľami:

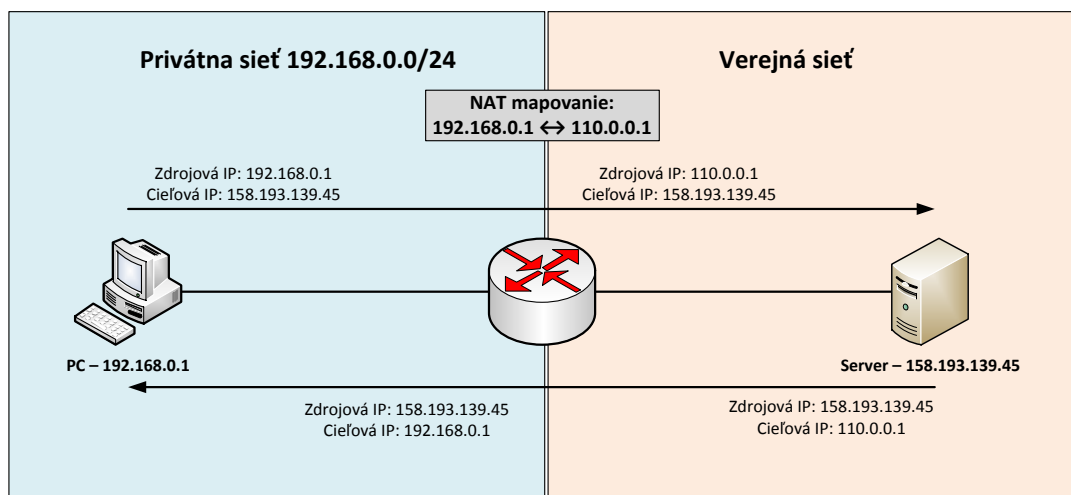
```
v=0
o=priklad 2890842807 IN IP4 10.47.16.5 //meno =
s=SDP Príklad
c=IN IP4 10.47.16.5
a=recvonly
m=audio 61965 RTP/AVP 0 108
a=sendrecv
a=rtpmap:0 BV32/8000
a=rtpmap:108 telephone-event/8000
a=fmtp:108 0-15
```

2.4 NAT – Network Address Translator

Základný typ NAT (z ang. Basic NAT) ako je špecifikovaný v **RFC 3022** (5) je mechanizmus mapovania IP adresy z jednej skupiny adries do inej. Potreba zavedenia NAT mechanizmov vznikla s rozrastajúcim sa počtom staníc pripojených k internetu a s tým spojeným klesajúcim počtom voľných IP adries. Boli zavedené pojmy a rozsahy **privátnych** (rozsahy: 10.0.0.0/8, 172.16.0.0/12 a 192.168.0.0/16 (5)) a **verejných** IP adries. Privátne adresy sú používané v rámci **privátnych** LAN sietí bez striktných pravidiel, pričom nesmú byť smerované vo **verejnej sieti (internet)**, ktorú tvoria zariadenia s **verejnými** IP adresami. Z uvedeného vyplýva, že adresy v privátnych LAN sieťach nie sú bežne dostupné z internetu, čo predstavuje hlavný problém, ktorého analýza a možné spôsoby riešenia budú hlavným obsahom nasledujúcich kapitol tejto práce. V špecifikácii (5) však nájdeme aj čiastočne rozšírený typ NAPT (Network Address **Port** Translation), ďalší mechanizmus, pomocou ktorého sú viaceré sieťové adresy a TCP/UDP porty prekladané na jedinú sieťovú adresu (zoznam adries) a jej TCP/UDP porty. V ďalších častiach tejto práce bude pojem NAT reprezentovať práve rozšírený NAPT mechanizmus.

2.4.1 Základné NAT

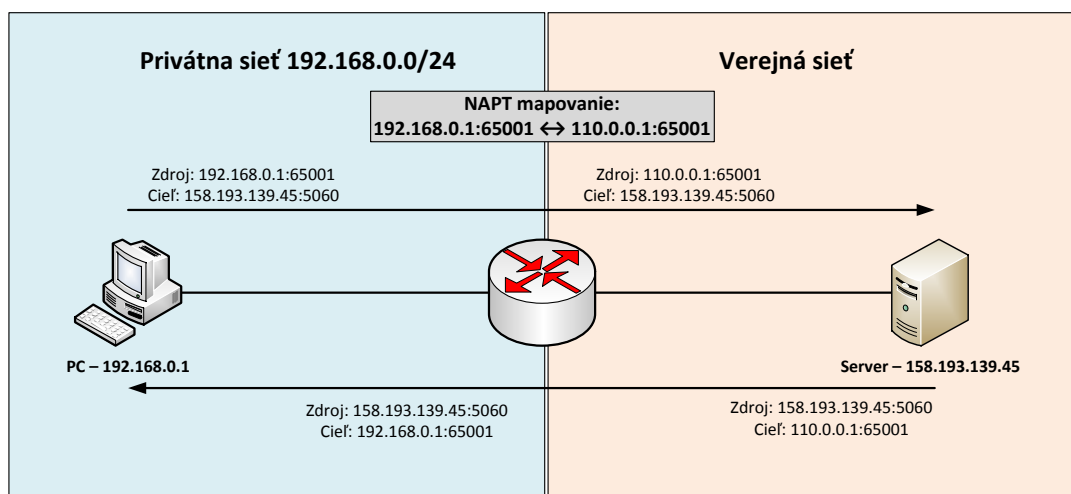
Pre tento typ NAT je charakteristický preklad IP adries staníc z privátnej oblasti, na definovaný blok externých (verejných) IP adries pri vytváraní spojení do externých sietí. V paketoch smerujúcich von z privátnej siete sa prekladá **zdrojová IP adresa**. Po preklade musí byť prerátaná kontrolná suma IP hlavičky. V opačnom smere sa naopak prekladá **cieľová IP adresa** a rovnako musí byť prerátaná kontrolná suma IP hlavičky. Príklad komunikácie a prekladu je ilustrovaný na Obrázku 2.3.



Obrázok 2.3 – Základné NAT

2.4.2 NATP

Ako bolo spomenuté v úvode podkapitoly, NATP rozširuje preklad IP adries o preklad transportnej informácie (**TCP, UDP port**). Toto rozšírenie umožňuje preložiť viacero privátnych IP adries na jedinú externú IP adresu. Prípadne, ak je použitý blok externých IP adries, je tento preklad realizovaný na celý blok príslušných adries. Pre odchádzajúce pakety z internej siete je teda prekladaná zdrojová IP adresa, k nej príslušné polia a navyše **zdrojové číslo portu** transportného protokolu. V opačnom smere ide o preklad **cieľovej IP adresy** a **cieľového čísla portu**. Príklad komunikácie a prekladu transportných adries je ilustrovaný na Obrázku 2.4.



Obrázok 2.4 - NATP

2.4.3 *Ďalšie typy NAT*

RFC 2663 definuje aj ďalšie typy NAT, ktoré nebudú predmetom analýzy v tejto práci z dôvodu ich menej častej implementácie a nasadenia:

- **obojsmerné NAT** (z ang. Bi-directional / Two-Way NAT) – privátne a verejné IP adresy sú dynamicky alebo staticky zviazané.
- **Dvojité NAT** (z ang. Twice NAT) – NAT prekladá zdrojovú aj cieľovú IP adresu.
- **Viacnásobné NAT** (z ang. Multihomed NAT) – riešenie, ktoré sa používa pri zálohovaných linkách s viacerými NAT pripojeniami.

V ďalšej práci budú pri jednotlivých mechanizmoch prechodu SIP cez NAT definované a popísané určité špeciálne formy (variácie) NAT, na ktoré sa odvolávajú príslušné špecifikácie.

2.5 NAT problémy pri SIP

Základným problém s NAT pri protokole SIP tvorí RTP tok aplikačných dát (hlas, video, správy, ...). Tieto dáta sú totiž posielaní medzi dvomi SIP telefónmi priamou komunikačnou cestou v sieti P2P (peer-to-peer) na rozdiel od architektúry SIP, ktorá je klient/server. Zariadenia s mechanizmom NAT, cez ktoré komunikujú klienti z privátnych sietí so servermi na verejnom segmente s pevne priradenými IP adresami (DNS záznamami) nepredstavujú veľký problém. Jediný problém predstavuje komunikácia prostredníctvom UDP protokolu, v prípade, že chce server poslať správu klientovi. Ide o prípad, kedy na NAT zariadení neexistuje príslušné mapovanie (klient ešte neposlal na sever správu alebo mapovanie už zaniklo). V tomto prípade je potrebné zabezpečiť aby bolo mapovanie pravidelne obnovované pre stálu dostupnosť klienta (mechanizmy *keepalive*). Na tento účel slúži viacero techník, ktoré sú popísané v **RFC 5626**.

Samostatný problém predstavuje RTP tok dát, ktorý ako už bolo spomenuté je P2P. V prípade, že SIP koncové body sú umiestnené za rôznymi NAT, nie je na založenie spojenia možné použiť ich privátne transportné adresy. Je teda potrebné zabezpečiť aby koncový bod dokázal určiť inú **transportnú adresu**, ktorú zapíše do SDP tela SIP paketu a navrhne ju inému koncovému bodu ako adresu na založenie RTP toku dát. Túto transportnú adresu môže predstavovať napríklad mapovaná (verejná) transportná adresa, ktorú SIP koncový bod dokáže určiť a zapísať ju do návrhu v SDP tele SIP paketu. Ako sa ukáže v ďalších častiach tejto práce ani verejná transportná adresa nemusí zaručiť, že P2P

komunikácia dvoch klientov bude možná. Ďalšie možnosti určenia najvhodnejšej transportnej adresy, problémy s tým spojené a ich riešenia budú popísané v nasledujúcich kapitolách tejto práce.

3 MECHANIZMY PRECHODU SIP CEZ NAT

V tejto kapitole budú popísané rôzne mechanizmy, ktoré sú používané protokolom SIP pri prechode cez zariadenia s mechanizmom NAT. Vo väčšine prípadov ide o podporné protokoly pomocou ktorých dokáže SIP klient určiť svoju verejne dosiahnuteľnú **transportnú adresu (IP adresa a číslo portu pre príslušný transportný protokol)**, na ktorú bude posielaný RTP tok dát od jedného alebo viacerých klientov. Uvedené mechanizmy nie sú určené len pre protokol SIP, ale je ich možné využiť pri akomkoľvek prenose dát cez smerovače (brány) s NAT mechanizmom. Príklady uvádzané pri jednotlivých protokoloch sú všeobecné, konkrétne príklady využitia určitého riešenia pri SIP architektúre budú uvedené v kapitole číslo štyri.

3.1 Stará špecifikácia STUN – klasický STUN

STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators, je protokol, ktorý vznikol v roku 2003 a je dokumentovaný v IETF štandarde **RFC 3489** (6)(v súčasnosti je STUN definovaný novým štandardom v **RFC 5389** (7), ktorého charakteristika bude popísaná v ďalších častiach tejto práce). Pomocou klasického protokolu STUN môže aplikácia určiť:

- či je medzi ňou a verejným internetom **zariadenie s mechanizmom NAT**.
- **Typ NAT**.
- **Verejnú IP adresu** mapovanú na zariadení s mechanizmom NAT.

Na základe týchto informácií môže aplikácia v privátnej sieti korektne pracovať a komunikovať aj s inými aplikáciami na verejnom internete. Samotný STUN protokol **nepredstavuje komplexné riešenie** pre všetky problémy súvisiace s NAT ale dokáže riešiť aspoň časť z nich. Rozdelenie v nasledujúcej časti rozdeľuje NAT problémy do dvoch skupín:

- problémy, ktoré **rieši** STUN protokol:
 - **detekcia** zariadenia s **NAT** mechanizmom medzi klientom a verejným internetom.
 - **Určenie mapovanej** transportnej **adresy** klienta.

- **Založenie UDP (RTP) komunikácie** na klienta za NAT (vybrané variácie NAT).
- Problémy, ktoré **nerieši** STUN protokol:
 - komunikáciu dvoch klientov, ktorí sa nachádzajú za **rovnakým NAT**.
 - Založenie UDP (RTP) komunikácie na klienta za symetrickým NAT.

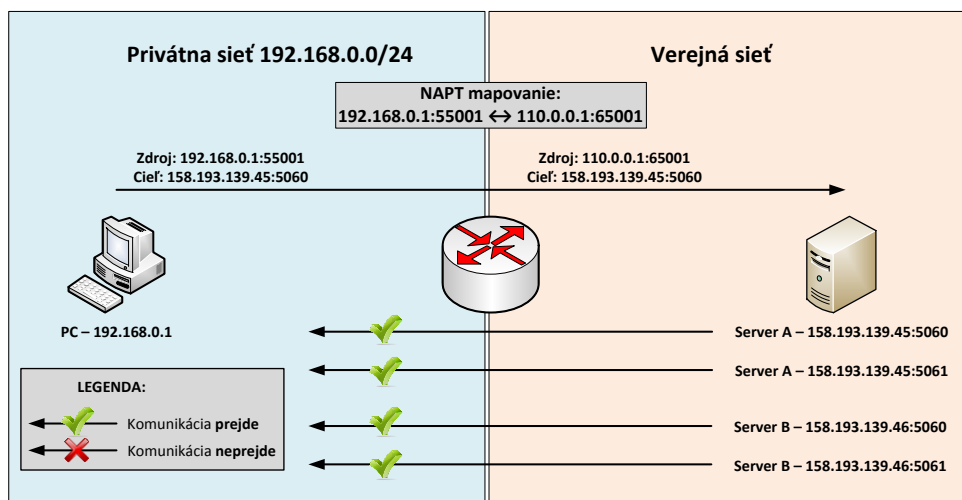
Špecifikácia (6) definuje dve STUN koncové entity, ktorými sú:

- **STUN klient** – entita, ktorá tvorí STUN požiadavky a je to spravidla PC (aplikácia bežiaci na PC – softvérový telefón alebo hardvérový SIP telefón), ktorý sa nachádza za jedným alebo viacerými zariadeniami s mechanizmom NAT.
- **STUN server** – entita, ktorá odpovedá na STUN požiadavky a je spravidla umiestnená na internete (zariadenie s verejnou IP adresou).

3.1.1 Variácie NAT

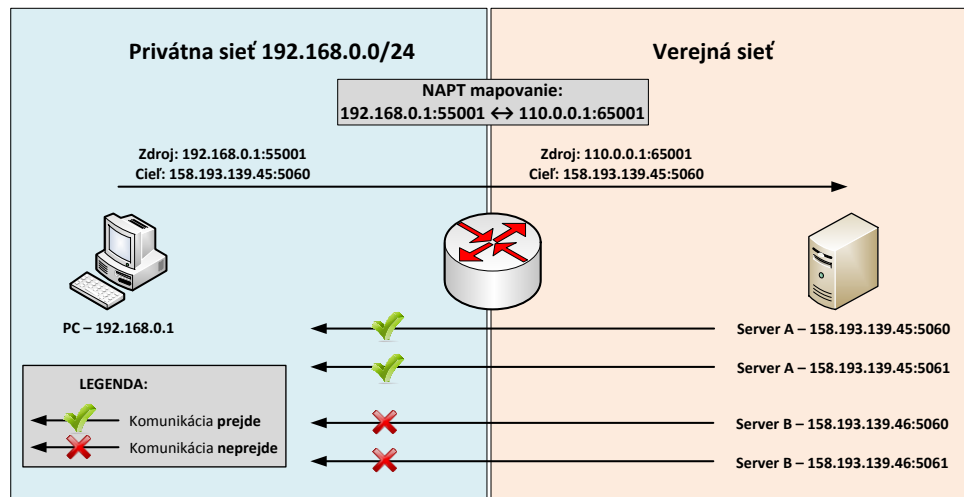
STUN protokol definuje špeciálne typy správania - **variácie NAT**, ktoré dokáže klient na základe komunikácie so serverom odhaliť a následne určiť či je možná komunikácia s iným klientom. Sú to nasledujúce štyri variácie NAT (6) (8):

- 1) **Full Cone** – variácia NAT, pri ktorej sú všetky požiadavky z internej IP adresy X a portu x mapované na **tú istú** externú IP adresu X^* a port x^* . Navyše môže každý externý koncový bod použiť vytvorené externé mapovanie $X^*:x^*$ (**IP adresa : číslo portu**) na komunikáciu s interným klientom. Táto variácia je považovaná za najmenej reštriktívnu. Komunikácia je ilustrovaná na nasledujúcom obrázku (Obrázok 3.1):



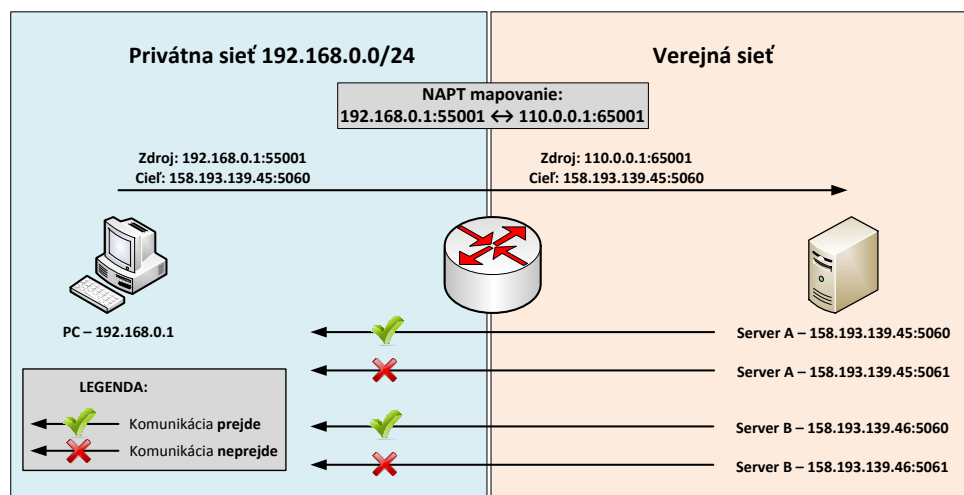
Obrázok 3.1 - Full Cone NAT

- 2) **Restricted Cone** – variácia NAT, pri ktorej sú všetky požiadavky z internej IP adresy X a portu x mapované na tú istú externú IP adresu X^* a port x^* . Na rozdiel od Full Cone NAT môže toto mapovanie použiť len externý klient s **tou istou externou IP adresou Y a ľubovoľným portom y** , ktorá už bola použitá pri mapovaní $X^*:x^*$. Komunikácia je ilustrovaná na nasledujúcom obrázku (Obrázok 3.2):



Obrázok 3.2 - Restricted Cone NAT

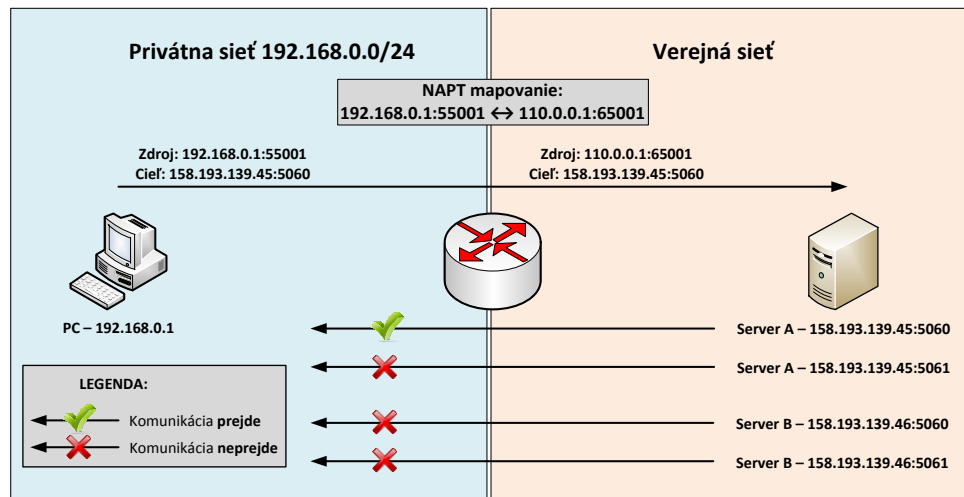
- 3) **Port Restricted Cone** – variácia NAT, ktorá je veľmi podobná Restricted Cone NAT pričom obmedzenie zahŕňa číslo portu y . Mapovanie $X^*:x^*$ môže použiť externý klient s **ľubovoľnou IP adresou Y a konkrétnym číslom portu y** , ak predtým interný klient poslal paket **práve na** externý port y . Komunikácia je ilustrovaná na nasledujúcom obrázku (Obrázok 3.3):



Obrázok 3.3 - Port Restricted Cone NAT

- 4) **Symetric** – variácia NAT, pri ktorej je komunikácia z internej IP adresy X a čísla portu x na externý koncový bod s IP adresou YI a portom yI je mapovaná na tú istú

externú kombináciu $X1*:x1*$. Ak ten istý interný klient pošle paket na iný externý koncový bod $Y2:y2$ je vytvorené nové mapovanie $X2*:x2*$. Na spätnú komunikáciu môže konkrétne mapovanie ($X1*:x1*$ alebo $X2*:x2*$) použiť len externý koncový bod, pri komunikácii s ktorým bolo toto mapovanie vytvorené. Komunikácia je ilustrovaná na nasledujúcom obrázku (Obrázok 3.4):



Obrázok 3.4 - Symetric NAT

3.1.2 Formát, typ a atribúty STUN správy

STUN je definovaný ako požiadavka/odpoveď protokol, ktorý využíva binárne orientovaný typ správ. Základný formát STUN správy, nové typy a atribúty budú definované, aj v ďalších častiach tejto práce. Stará STUN špecifikácia (6) definuje dva základné typy požiadaviek klientov:

- **Binding Request** – požiadavka slúži na určenie mapovanej transportnej adresy na zariadení s mechanizmom NAT.
- **Shared Secret Request** – slúži na rovnaký účel ako predchádzajúca správa ale na rozdiel od **Binding Request** je v tejto požiadavke zahrnuté overovanie.

K požiadavkám sú definované aj príslušné typy odpovedí servera v závislosti od úspešnej alebo neúspešnej komunikácie STUN klienta so serverom:

- **Binding Response** – požiadavka **Binding Request** bola **úspešná**.
- **Binding Error Response** – požiadavka **Binding Request** bola **neúspešná**.
- **Shared Secret Response** – požiadavka **Shared Secret Request** bola **úspešná**.
- **Shared Secret Error Response** – požiadavka **Shared Secret Request** bola **neúspešná**.

Rozdiel medzi **Binding** a **Shared Secret** požiadavke alebo odpovedi je teda len v použitej, respektíve nepoužitej autentifikácii. Všetky typy správ sú využívané pri komunikácii STUN klienta so serverom, pričom sú v správach (pri špecifikácii **RFC 3478**) zapuzdrené tieto základné typy atribútov:

- **MAPPED-ADDRESS** – atribút, ktorý sa nachádza v odpovediach STUN servera a je v ňom obsiahnutá **mapovaná IP adresa a číslo portu**, pod ktorou vidí server klienta (mapovaná transportná adresa zariadením s NAT).
- **CHANGE-REQUEST** – atribút, ktorý STUN klient použije v prípade, že chce aby odpoveď servera bola odoslaná z inej IP adresy alebo čísla portu (prípadne oboch). Atribút obsahuje dva nastaviteľné príznaky:
 - **zmena IP** – ak je nastavený na **1**, klient vyžaduje aby bola odpoveď servera odoslaná **z inej IP adresy**, než na akej bola prijatá požiadavka.
 - **Zmena portu** - ak je nastavený na **1**, klient vyžaduje aby bola odpoveď servera odoslaná **z iného čísla portu**, než na akom bola prijatá požiadavka.
- **CHANGED-ADDRESS** – obsahuje IP adresu a číslo portu v **každej odpovedi servera**, z ktorej by bola odoslaná odpoveď, ak by v atribúte CHANGE-REQUEST požiadavky klienta, boli oba príznaky nastavené na **1**.
- **RESPONSE-ADDRESS** – pomocou tohto atribútu dokáže klient v požiadavke definovať IP adresu a číslo portu, na ktorú má server odoslať odpoveď.

3.1.3 Určenie variácie NAT

Detekcia variácie NAT je založená na základnej STUN klient/server komunikácii. Je potrebné zabezpečiť aby mal STUN server možnosť komunikovať z minimálne dvoch IP adries a dvoch čísiel portov v ľubovoľnej kombinácii. Ak je splnená táto podmienka je možné využiť nasledujúce tri testy (6):

- **TEST č.1** – STUN klient pošle **Binding Request** požiadavku na STUN server **bez nastavených príznakov** v atribúte **Change-Request** a bez atribútu **Response-Address**, čo znamená že STUN server odošle odpoveď na adresu, z ktorej prišla požiadavka a z rovnakej adresy, na ktorej bola požiadavka prijatá.
- **TEST č.2** – STUN klient pošle **Binding Request** požiadavku na STUN server **s oboma nastavenými príznakmi** v atribúte **Change-Request**.

- **TEST č.3** – STUN klient pošle *Binding Request* požiadavku na STUN server len s príznakom na **zmenu portu** v atribúte *Change-Request*.

Pomocou vyššie uvedených testov môže používateľ (STUN klient) určiť správanie zariadení, ktoré sa nachádzajú medzi ním a verejným internetom a rovnako variáciu NAT, pokiaľ sa v sieti nachádza zariadenie s mechanizmom NAT. Môžu nastať nasledovné situácie:

- **dostupný verejný internet** – užívateľ má verejnú IP adresu a žiadne zariadenie nekontroluje prevádzku.
- **Firewall blokuje UDP** – užívateľ nemôže využiť na komunikáciu protokol UDP.
- **Symetrický Firewall** – medzi užívateľom a verejným internetom sa nenachádza zariadenie s mechanizmom NAT ale firewall, ktorý povolí len UDP požiadavky smerom von a odpovede, ktoré prichádzajú na zdrojovú adresu požiadavky (podobné ako symetrické NAT ale bez prekladu adresy).
- **Full Cone NAT.**
- **Restricted NAT.**
- **Port Restricted NAT.**
- **Symetric NAT.**

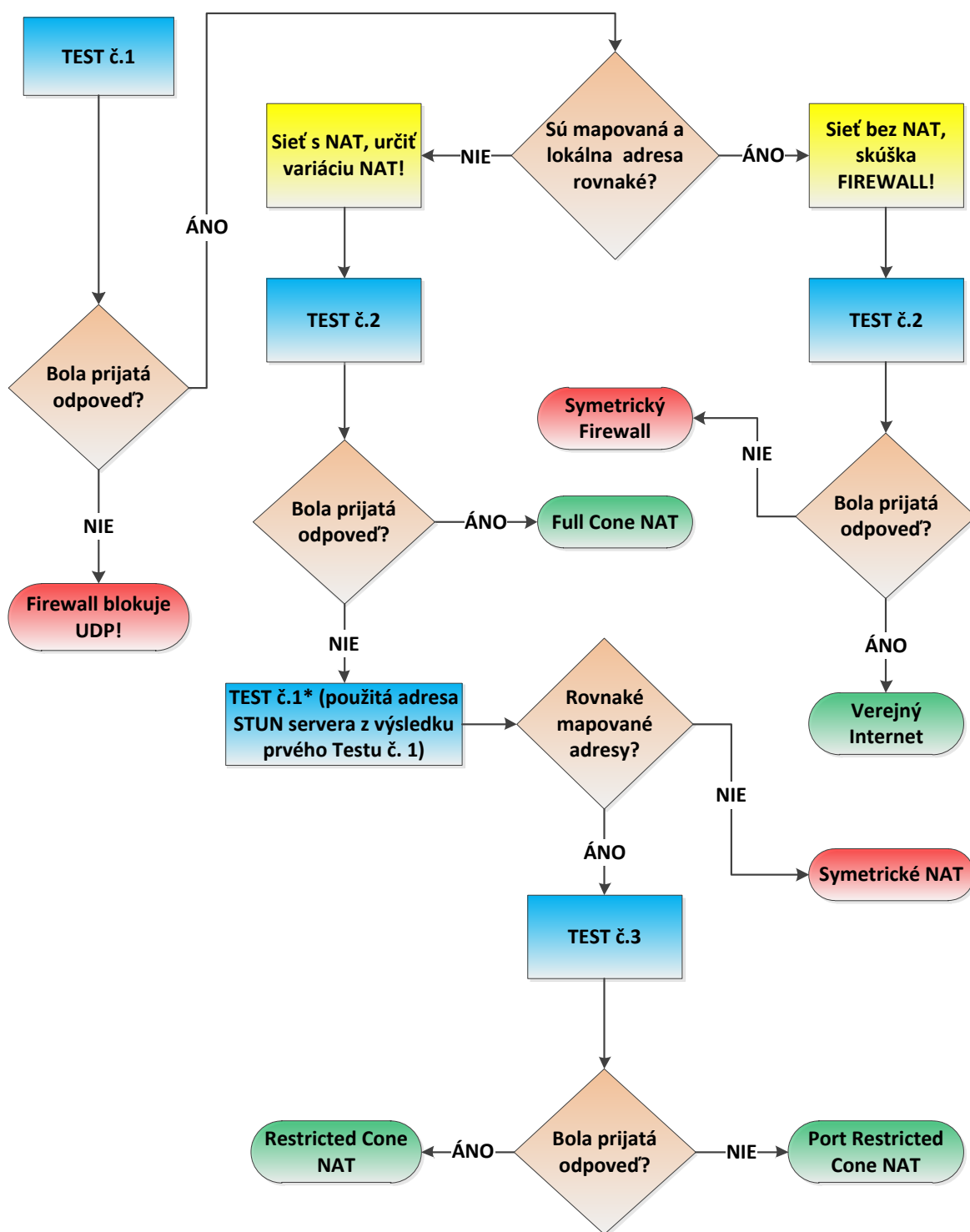
Posledné štyri situácie vyjadrujú variáciu NAT, s ktorou pracuje zariadenie s mechanizmom NAT medzi STUN klientom a externou sieťou (internetom).

STUN klient začína testovanie Testom č.1 a čaká na odpoveď od STUN servera. V prípade, že odpoveď nebola doručená, klient nie je schopný komunikovať prostredníctvom UDP (**firewall blokuje UDP**). Ak bola doručená odpoveď, klient porovná transportnú adresu v atribúte *MAPPED-ADDRESS* STUN odpovede s lokálnou transportnou adresou.

1. Ak sú adresy zhodné, klient **sa nenachádza za** žiadnym zariadením s mechanizmom **NAT**. Klient vykoná následne Test č.2. Ak nebola doručená odpoveď od servera, klient sa nachádza za firewall, ktorý sa správa podobne ako symetrické NAT (**symetrický firewall**). V prípade doručenia odpovede od STUN servera sa jedná o situáciu kedy má klient verejnú IP adresu a môže bez obmedzení komunikovať prostredníctvom UDP (**dostupný verejný internet**).

2. Ak adresy nie sú zhodné, klient **sa nachádza za** zariadením s mechanizmom **NAT**. Klient rovnako vykoná Test č.2. Ak bola doručená odpoveď, klient sa nachádza za NAT s najmenej reštriktívnou variáciou (**Full Cone NAT**). Opäť však môže nastať situácia, že odpoveď od servera nebude doručená a pre bližšie určenie správania NAT zariadenia je potrebné vykonať opäť Test č.1 (v tomto prípade je ako cieľová adresa STUN servera použitá adresa zapísaná v atribúte *CHANGED-ADDRESS* z odpovede servera pri prvej realizácii Testu č.1). Následne je transportná adresa zapísaná v atribúte *MAPPED-ADDRESS* z odpovede servera porovnaná s transportnou adresou, ktorá bola zapísaná v rovnakom atribúte pri prvej realizácii Testu č.1. V prípade, že tieto **adresy nie sú zhodné** sa klient nachádza za mechanizmom NAT, ktoré je závislé od transportnej adresy cieľového koncového bodu (**Symetric NAT**). Ak sú naopak vybrané **adresy zhodné** je potrebné vykonať Test č.3. Rovnako môže nastať prípad, kedy na STUN požiadavku nebude doručená žiadna odpoveď, v tomto prípade sa klient nachádza za NAT s variáciou **Port Restricted NAT**. Ak však odpoveď bola doručená, klient sa nachádza za NAT s variáciou **Restricted NAT**.

Využitie vyššie spomínaných testov na určenie variácie NAT a priebeh testovania je ilustrovaný na nasledujúcom diagrame (Obrázok 3.5) (6):



Obrázok 3.5 - Určenie variácie NAT

Zelenou farbou sú na diagrame naznačené výsledky testov, v prípade ktorých **je možné s klientom komunikovať** pomocou protokolu UDP. Naopak, **červenou farbou** sú vyznačené situácie, kedy takáto **komunikácia možná nebude**.

Protokol STUN so zisťovaním variácií NAT sa ukázal ako **nevhodné riešenie** prechodu cez NAT. Problémom bolo, že nie všetci výrobcovia zariadení s NAT mechanizmom sa striktne pridržiavali len variácií definovaných v **RFC 3489** a navyše algoritmus na ich detekciu sa ukázal ako chybný. Ďalším problémom bol fakt, že nebolo možné určiť, či je zistená verejná adresa klienta použiteľná na komunikáciu s iným klientom (napríklad pri RTP toku dát), pretože neexistoval žiaden spôsob testovania. Vznikol tak nový prístup k riešeniu problémov s prechodom cez NAT, čo malo za následok predefinovanie STUN protokolu z **RFC 3489**, ktorý je označovaný ako **klasický STUN** (7). Nová špecifikácia STUN protokolu je popísaná v nasledujúcej časti.

3.2 Nová špecifikácia STUN – Session Traversal Utilities for NAT

Ako bolo spomenuté v predchádzajúcej časti, došlo k zmene špecifikácie protokolu STUN a rovnako tak k zmene významu skratky STUN. Zmeny v STUN sú popísané v **RFC 5389** (7), ktoré nahrádza predchádzajúce **RFC 3489**. Nová špecifikácia vníma STUN protokol ako nástroj, ktorý slúži iným protokolom na prechod cez zariadenia s mechanizmom NAT. V princípe ide o protokol, ktorý môže slúžiť na (7):

- **určenie externej IP adresy** klienta za NAT.
- **Udržiavanie otvorených NAT mapovaní** klientov za NAT.
- **Výmenu paketov** medzi dvomi koncovými bodmi spojenia.
- **Overenie konektivity** dvoch koncových bodov spojenia.

Tretí a štvrtý bod sú rozšírenia protokolu STUN – TURN a ICE, ktoré budú popísané v ďalších častiach tejto práce a sú založené na výmene STUN správ medzi dvomi koncovými bodmi spojenia.

STUN ostáva definovaný ako klient/server protokol, ktorý je transakčne orientovaný. Rozlišujeme dva typy transakcií (7):

- **požiadavka/odpoveď** - STUN klient pošle **požiadavku** na STUN server, na ktorú očakáva príslušnú **odpoveď** od STUN servera.
- **Indikácia** – STUN klient alebo server môže poslať **indikáciu** (náznak ďalšej akcie), na ktorú nie je generovaná ani očakávaná **žiadna odpoveď**.

Každá transakcia je jednoznačne identifikovaná pomocou náhodne generovaného **ID transakcie**.

3.2.1 Formát a typ STUN správy

STUN správy sú binárne kódované a skladajú sa z **hlavičky** a žiadneho alebo viacerých **atribútov** v tele správy. Hlavička má pevne definovanú veľkosť dvadsať bajtov a skladá sa z polí, ktoré sú naznačené na nasledujúcom obrázku (Obrázok 3.6).



Obrázok 3.6 - Hlavička STUN správy

Prvé dva najvýznamnejšie bity v STUN hlavičke musia byť **nastavené na hodnotu nula** (zvýraznené červené pole na Obrázku 3.6), význam ostatných polí hlavičky je nasledovný (7):

- **STUN Typ Správy** – pole dĺžky štrnásť bitov, ktoré určuje **triedu** a **metódu** STUN správy. V špecifikácii *RFC 5389* je definované len **jedna metóda spojovanie** – **0b00000000000001** (z ang. **Binding**) a **štyri triedy**:
 - **požiadavka** – **0b00**.
 - **Indikácia** – **0b01**.
 - **Úspešná odpoveď** – **0b10**.
 - **Neúspešná odpoveď** – **0b11**.

Za každou z tried a metódou sú uvedené ich binárne reprezentácie, pomocou ktorých sú zapísané v štrnástich bitoch typu správy spôsobom, naznačenom na obrázku (Obrázok 3.7).



Obrázok 3.7 - Pole STUN Typ Správy

Na obrázku modré polia M reprezentujú dvanásť bitové číslo metódy a zelené polia T triedu STUN správy. Pre každú existujúcu alebo novú metódu musí byť definované aké typy tried sú pre ňu prípustné. Pre metódu spojovanie sú prípustné

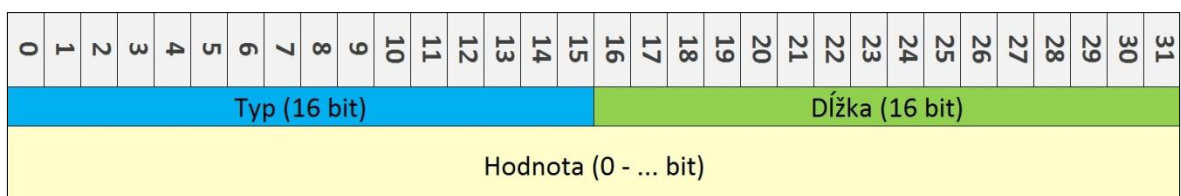
všetky triedy a ako príklad uvedieme reprezentáciu **úspešnej odpovede spajovania** a jej príslušnú binárnu reprezentáciu.

spojovanie úspešná odpoveď – 0b0000001000000001 = 0x0101

- **Dĺžka Správy** – šesťnásť bitové pole, ktoré určuje dĺžku STUN správy v bajtoch, bez dvadsať bajtovej hlavičky STUN správy.
- **Magic Cookie** – pole dĺžky tridsaťdva bitov, ktoré musí obsahovať definovanú hodnotu 0x2112A442 a určuje, že konkrétna správa je vytvorená podľa novej špecifikácie STUN.
- **ID Transakcie** – deväťdesiatšesť bitové pole, ktoré jednoznačne identifikuje STUN transakciu. Reprezentuje ho náhodné číslo, ktoré je generované STUN klientom v požiadavkách a kopírované STUN serverom do odpovede na požiadavku. Na základe tohto poľa je teda možné **spárovať požiadavky a odpovede**. Je teda dôležité aby pri tvorbe požiadaviek boli dodržané základné pravidlá, ktoré určujú že pre všetky nové transakcie musí byť **vygenerované nové ID**. To isté ID transakcie **môže byť použité** pri **opakovaní požiadavky** a rovnako v prípade, ak je požiadavka bitovo totožná s predchádzajúcou požiadavkou a je posielaná z rovnakej transportnej adresy klienta na rovnakú transportnú adresu servera.

3.2.2 Atribúty správ

Po STUN hlavičke nasledujú atribúty správy. Počet atribútov je ľubovoľný a rovnako je možné poslať STUN správu bez atribútov. Každý z atribútov je TLV (ang. Type-Length-Value – Typ-Dĺžka-Hodnota) kódovaný. Základná štruktúra atribútu je naznačená na nasledujúcom obrázku (Obrázok 3.8) (7).

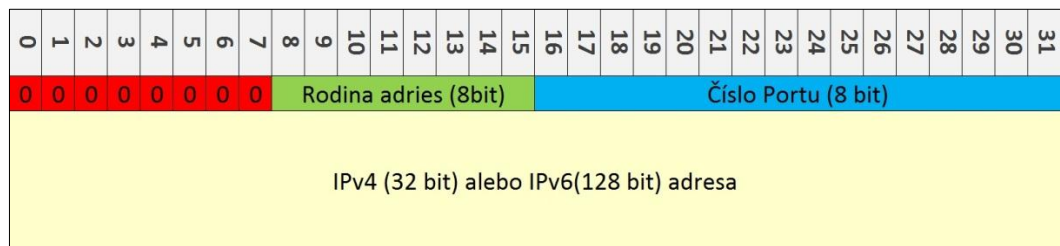


Obrázok 3.8 - Základná štruktúra atribútu

Pole **Typ** jednoznačne určuje typ atribútu. Priradenie jednotlivých číselných reprezentácií k atribútom je pod správou IANA - Internet Assigned Numbers Authority. **Dĺžka** špecifikuje veľkosť poľa **Hodnota** v bajtoch, ktoré je špecifické pre jednotlivé typy

atribútov. V nasledujúcej časti sú popísané základné STUN atribúty, ktoré boli implementované aj v praktickej časti tejto práce (7):

- **MAPPED-ADDRESS** – atribút, ktorý vyjadruje **reflexívnu transportnú adresu** (transportná adresa, pod ktorou vidí server klienta, spravidla je to verejná adresa mapovaná na NAT zariadení) klienta a je zapuzdrený **v odpovedi servera**. Atribút môže obsahovať ako rodinu IPv4, tak IPv6 adres. Štruktúra atribútu je znázornená na nasledujúcom obrázku (Obrázok 3.9).



Obrázok 3.9 - Štruktúra MAPPED-ADDRESS atribútu

Ako vyplýva z uvedeného obrázku prvých **osem bitov** je **nulových**, nasleduje kódovaný príznak **rodiny adres** IPv4 alebo IPv6, **čísla portu** a v poslednom poli sa nachádza kódovaná príslušná **IPv4** alebo IPv6 **adresa**.

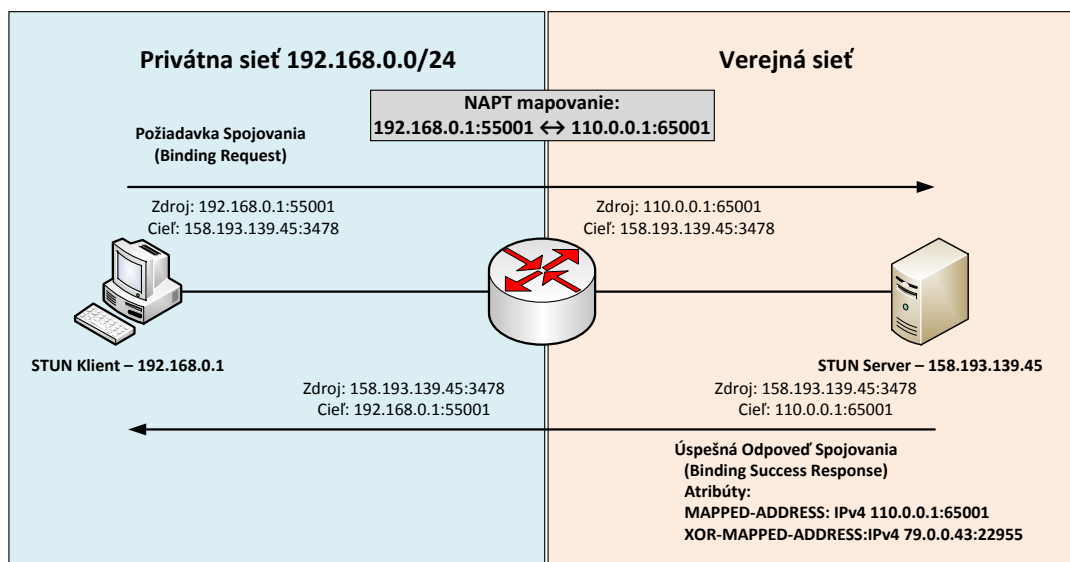
- **XOR-MAPPED-ADDRESS** – atribút rovnako reprezentuje **reflexívnu transportnú adresu** klienta a jeho štruktúra je totožná s predchádzajúcim atribútom (Obrázok 3.9). Rozdiel je vo **formáte zápisu** IPv4 alebo IPv6 adresy a čísla portu v príslušných poliach správy, kde obe hodnoty sú výsledkom funkcie XOR príslušného poľa a **poľa Magic Cookie (0x2112A442)** z hlavičky STUN správy.
- **ERROR-CODE** – atribút, ktorý slúži na bližšie určenie chyby pre typ STUN správ: **neúspešná odpoveď**, ktorá je odosielaná STUN serverom. Obsahom je **kód chyby** a **textový popis** príslušnej **chyby** s textovým kódovaním aké je použité aj v protokole SIP (podkapitola 2.1). Vybrané chybové hlásenia:
 - **400 Zlá požiadavka** – doručená požiadavka nebola syntakticky správna.
 - **420 Neznámy atribút** – požiadavka obsahovala atribút, ktorý server nedokáže spracovať.
 - **500 Chyba servera** – vyskytla sa dočasná chyba servera a požiadavku nie je možné spracovať.

- **SOFTWARE** – obsahom je textový popis softvéru (názov, verzia, ...), ktorý príslušnú STUN správu vygeneroval.

Ďalšie typy správ a atribútov je možné nájsť v **RFC 5389** (7), kde ide najmä o atribúty spojené s autentifikáciou STUN klienta voči serveru a rovnako v STUN rozširujúcich RFC (TURN, ICE, ...), z ktorých vybrané správy a atribúty budú popísané v ďalších častiach tejto práce.

3.2.3 STUN komunikácia

Ako už bolo spomenuté v predchádzajúcich častiach práce, protokol STUN bol navrhnutý na **určenie mapovanej (externej) IP adresy – reflexívna transportná adresa** klienta za NAT a na udržanie tohto mapovania otvoreného pre možnosť komunikácie s iným koncovým bodom. Na určenie reflexívnej transportnej adresy je nutná komunikácia medzi STUN klientom a STUN serverom, ktorý predvolene počúva na **UDP (TCP) porte 3478**. Klient prostredníctvom správy **Požiadavka spojovania** (z ang. Binding Request) kontaktuje server, ktorý môže na požiadavku odpovedať **Úspešnou** prípadne **Neúspešnou odpoveďou spojovania** (z ang. Binding Success/Error Response). V prípade **úspechu** server vytvorí správu **Úspešná odpoveď spojovania** a do atribútu **MAPPED-ADDRESS** alebo **XOR-MAPPED-ADDRESS** (prípadne oboch) vloží zdrojovú IP adresu požiadavky, pre ktorú je príslušná odpoveď vytváraná a odoslaná k STUN klientovi. Klient následne spracuje odpoveď a ako svoju reflexívnu transportnú adresu určí IP adresu a číslo portu z vyššie spomenutých atribútov. V prípade neúspechu sa v odpovedi servera nachádza atribút **ERROR-CODE**, na základe ktorého dokáže klient určiť príčinu neúspešného spracovania požiadavky. Úspešná komunikácia medzi STUN klientom a serverom je ilustrovaná na nasledujúcom obrázku (Obrázok 3.10).



Obrázok 3.10 - STUN komunikácia

Získanú reflexívnu transportnú adresu môže využiť na komunikáciu s klientom za NAT aj iný protokol, ako je napríklad RTP, pomocou ktorého sú v SIP architektúre prenášané multimediálne dáta. Na výmenu transportných adries dvoch, prípadne viacerých SIP klientov, je využívaný protokol SDP prenášaný v tele SIP paketu.

3.2.4 Rozdiely špecifikácií STUN - zhrnutie

Rôzne rozdiely špecifikácii protokolu STUN z **RFC 3489** a **RFC 5389** boli spomenuté v predchádzajúcich častiach práce. Viaceré funkcionality boli zo staršej špecifikácie odstránené a následne boli doplnené nové pre lepšiu rozšíriteľnosť STUN protokolu a jeho využitie v mechanizme ICE. Krátke zhrnutie rozdielov je v Tabuľke 3.1.

Funkcionalita	RFC 3478	RFC 5389
klient/server protokol	X	X
transakčne orientovaný protokol	X	X
určenie variácie NAT	X	
určenie reflexívnej transportnej adresy	X	X
prvé dva nulové bity STUN správy		X
pole Magic Cookie v hlavičke STUN správy		X
atribút MAPPED-ADDRESS	X	X
atribút CHANGE-REQUEST	X	
atribút CHANGED-ADDRESS	X	
atribút RESPONSE-ADDRESS	X	
atribút XOR-MAPPED-ADDRESS		X
podpora IPv6		X

Tabuľka 3.1 - Rozdiely STUN špecifikácií

3.3 Rozšírenie TURN – Traversal Using Relays around NAT

Riešenie viackrát spomínaných problémov komunikácie cez určité typy (variácie) NAT (napr. symetrické NAT) prináša rozšírenie novej špecifikácie protokolu STUN – **TURN protokol**. Definícia a možnosti použitia tohto protokolu sú zapísané v **RFC 5766** (9). Využitie TURN protokolu je najmä v prípade, ak P2P komunikácia (napr. prostredníctvom reflexívnych transportných adries) dvoch koncových bodov spojenia nie je možná. V tejto situácii je nutné použiť zariadenie na verejnom internete na **preposielanie (z ang. relay) komunikácie** medzi týmito koncovými bodmi. Za týmto účelom bol navrhnutý protokol TURN, ktorý slúži na určenie **základných parametrov** potrebných pre preposielanie dát. Prostredníctvom TURN protokolu je rovnako možné aj **preposielanie aplikačných dát** medzi dvomi koncovými bodmi spojenia. Základná TURN komunikácia prebieha medzi dvomi bodmi spojenia (9):

- **TURN klient** – entita, ktorá sa spravidla nachádza za zariadením s mechanizmom NAT a posiela požiadavky alebo dáta v reálnom čase na TURN server.
- **TURN server** – entita nachádzajúca sa spravidla na verejnom internete, ktorá odpovedá na požiadavky alebo preposiela aplikačné dáta TURN klientov.

Výsledkom základnej komunikácie medzi TURN klientom a serverom je získanie **prenosovej transportnej adresy** (z ang. relayed transport address), ktorá následne slúži na komunikáciu TURN klienta s inými zariadeniami v sieti.

Protokol TURN bol navrhnutý ako súčasť mechanizmu ICE, ktorý bude popísaný v ďalších častiach tejto práce a nie ako samostatné riešenie prechodu aplikačných dát cez NAT. Hlavným dôvodom prečo sa prednostne využívajú iné komunikačné cesty dvoch koncových bodov spojenia (lokálne IP adresy, STUN reflexívne transportné adresy) je vysoký nárok na **šírku pásma pripojenia** TURN servera do internetu. Pre poskytovateľov napr. komunikačných služieb, pri ktorých je nutné zabezpečiť dostupnosť používateľov na akomkoľvek mieste, je teda lacnejšie nájsť priamu dátovú cestu pre komunikujúcich a **TURN riešenie použiť iba ak priama komunikácia (P2P) nie je možná**.

3.3.1 Pridané metódy a atribúty pre TURN

Základ pre štruktúru správ a atribútov pre protokol TURN poskytuje nová špecifikácia protokolu STUN. Pre komunikáciu TURN server predvolene využíva číslo portu **3478** a často býva implementovaný ako STUN/TURN server, ktorý poskytuje služby

oboch protokolov. TURN teda nedefinuje vlastný formát správ ale rozširuje STUN protokol o nové metódy a atribúty správ.

3.3.1.1 Nové metódy správ

STUN protokol definoval len jedinú metódu: **spojovanie** a k nej príslušné triedy: požiadavka, úspešná/neúspešná odpoveď a indikácia. Pre potreby protokolu TURN boli dodefinované nasledujúce metódy, pri ktorých sú naznačené aj triedy použitia správ (9):

- **priradenie** (z ang. Allocation) – slúži na určenie prenosovej transportnej adresy TURN klienta na serveri v požiadavka/odpoveď transakcii.
- **Obnovenie** (z ang. Refresh) – pomocou tejto metódy je možné obnoviť alebo zrušiť predchádzajúce priradenie prenosovej transportnej adresy v požiadavka/odpoveď transakcii.
- **Odoslanie** (z ang. Send) – metóda, pomocou ktorej TURN klient posíla aplikačné dáta pre iného účastníka komunikácie na TURN server v transakcii triedy indikácia.
- **Dáta** (z ang. Data) – metóda, pomocou ktorej TURN server posíla aplikačné dáta od iných účastníkov komunikácie TURN klientovi v transakcii triedy indikácia.
- **Vytvorenie povolenia** (z ang. Create Permission) – metóda pomocou, ktorej TURN klient vytvorí povolenie na TURN serveri pre svoju prenosovú transportnú adresu, na spracovanie (preposielanie) aplikačných dát z konkrétnej transportnej adresy iného koncového bodu v požiadavka/odpoveď transakcii.
- **Rezervovanie kanála** (z ang. Channel Bind) – metóda slúžiaca na rezervovanie kanála na TURN serveri pre prenos aplikačných dát medzi TURN klientom a ďalším koncovým bodom spojenia v požiadavka/odpoveď transakcii.

3.3.1.2 Pridané atribúty správ

Atribúty definované v **RFC 5389** nepostačovali potrebám TURN protokolu a tak boli definované nové typy atribútov, ktoré dodržiavajú základné pravidlá pre vytváranie atribútov v novej STUN špecifikácii. TURN špecifikácia definuje nasledujúce základné typy atribútov (vybrané atribúty, ďalšie možno nájsť v **RFC 5766** a príslušných rozšíreniach) (9):

- **XOR-RELAYED-ADDRESS** – atribút, ktorý sa nachádza v odpovediach TURN servera a určuje prenosovú transportnú adresu (IP adresa a číslo portu), ktorú server

priradil klientovi na komunikáciu s iným koncovým bodom. Kódovanie a štruktúra príslušných polí je rovnaké ako pri atribúte XOR-MAPPED-ADDRESS definovanom v časti 3.2.2.

- **XOR-PEER-ADDRESS** – obsahuje transportnú adresu iného člena komunikácie z pohľadu TURN servera a posiela sa v požiadavke klienta pre určenie druhého koncového bodu komunikácie. Kódovanie a štruktúra príslušných polí je rovnaké ako pri atribúte XOR-MAPPED-ADDRESS definovanom v časti 3.2.2.
- **CHANNEL-NUMBER** – atribút obsahuje číslo kanála, ktoré je priradené konkrétnej dvojici koncových bodov spojenia (dvojicu môžu tvoriť dvaja TURN klienti alebo TURN klient a iný koncový bod spojenia, ktorý je definovaný svojou transportnou adresou).
- **LIFETIME** – slúži na časové ohraničenie dĺžky priradenia prenosovej transportnej adresy. Môže byť použité v požiadavke TURN klienta, ak žiada o inú dĺžku priradenia ako je predvolené, alebo v odpovedi servera, kde oznamuje klientovi skutočnú dĺžku priradenia.
- **EVEN-PORT** – atribút, v ktorom TURN klient môže požiadať server o priradenia párneho čísla portu pre prenosovú transportnú adresu a rezervovanie prvého vyššieho čísla portu (kompatibilita pre RTP).
- **REQUESTED-TRANSPORT** – v tomto atribúte TURN klient žiada o priradenia prenosovej transportnej adresy pre špecifický protokol. TURN RFC zatiaľ definuje len jediný protokol – UDP.
- **DATA** – obsahuje aplikačné dáta.

3.3.2 TURN komunikácia

Základná komunikácia protokolu TURN prebieha medzi TURN klientom a serverom, do tejto komunikácie môže byť následne zaradený externý koncový bod (jeden alebo viac) spojenia - **partner**, s ktorým si TURN klient posiela aplikačné dáta. V nasledujúcej časti budú popísané jednotlivé kroky zostavenia komunikácie medzi TURN klientom, TURN serverom a jedným partnerom.

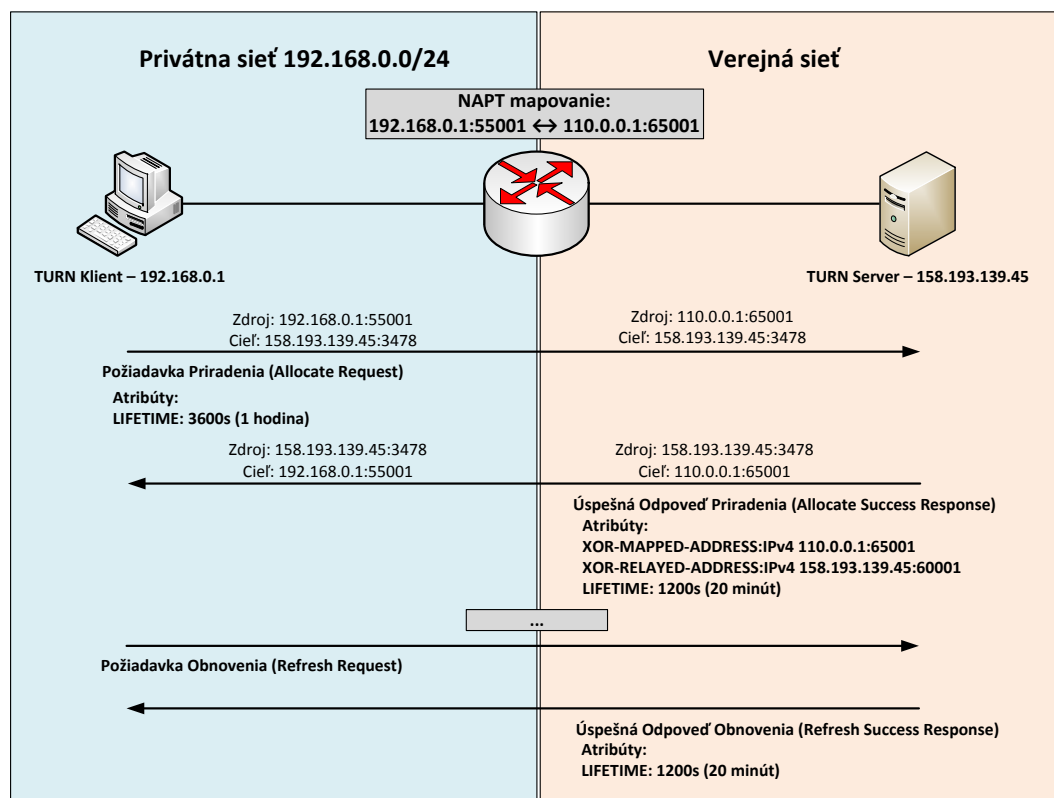
Ako prvú TURN klient posiela ***Požiadavku priradenia*** (z ang. Allocate Request), v ktorej žiada server o pridelenie prenosovej transportnej adresy. Protokol STUN definuje dve formy autentifikácie správ, ktoré sú pri tomto rozšírení používané na overenie TURN klienta pri všetkých správach, ktoré nie sú triedy indikácia. Je to z dôvodu zamedzenia

neoprávneného využívania servera, keďže ide o zariadenie, ktorého prevádzka je veľmi nákladná (overovanie správ je mimo rozsahu tejto práce a viac informácií je možné nájsť v príslušných RFC (6) (9), komunikácia medzi klientom a serverom v nasledujúcich častiach bude naznačená bez overovania). Klient v požiadavke priradenia môže uviesť dĺžku trvania priradenia v atribúte *LIFETIME*, ak má byť iná ako predvolená (10 minút), prípadne ďalšie atribúty (*EVEN-PORT*, *REQUESTED-TRANSPORT*). Server na požiadavku pošle **Neúspešnú odpoveď priradenia** (z ang. Allocate Request Error), v prípade ak neboli splnené všetky náležitosti požiadavky alebo **Úspešnú odpoveď priradenia**, ktorá obsahuje tieto základné atribúty: *XOR-RELAYED-ADDRESS*, *LIFETIME* (býva bežnou praxou, že TURN server do odpovede zahrnie aj STUN atribút *XOR-MAPPED-ADDRESS* pre rýchlejší zber kandidátov v mechanizme ICE). TURN klient a server si následne uložia takzvanú **päťicu**. Obsah päťice pre TURN klienta a server je uvedený v nasledujúcej tabuľke (Tabuľka 3.2) (9):

TURN server	TURN klient
IP adresa	IP adresa
Číslo portu	Číslo portu
Reflexívna IP adresa TURN klienta	IP adresa TURN servera
Reflexívne číslo portu TURN klienta	Číslo portu TURN servera
Transportný protokol (UDP)	Transportný protokol (UDP)

Tabuľka 3.2 - Obsah päťice TURN servera a klienta

Klient môže počas doby platnosti požiadavku priradenia kedykoľvek obnoviť pomocou správy **Požiadavka obnovenia** (z ang. Refresh Request) na ktorú server odpovedá príslušnou správou **Úspešná požiadavka obnovenia**, kde v atribúte *LIFETIME* uvedie novú platnosť prenosovej transportnej adresy (požiadavka klienta je na serveri identifikovaná práve na základe päťice, pomocou ktorej server dokáže určiť o obnovenie akej požiadavky a ktorý TURN klient žiada). Vybraná časť komunikácie je naznačená na nasledujúcom obrázku (Obrázok 3.11 – uvedené IP adresy v XOR atribútoch sú pre lepšiu názornosť uvedené pred použitím funkcie XOR).

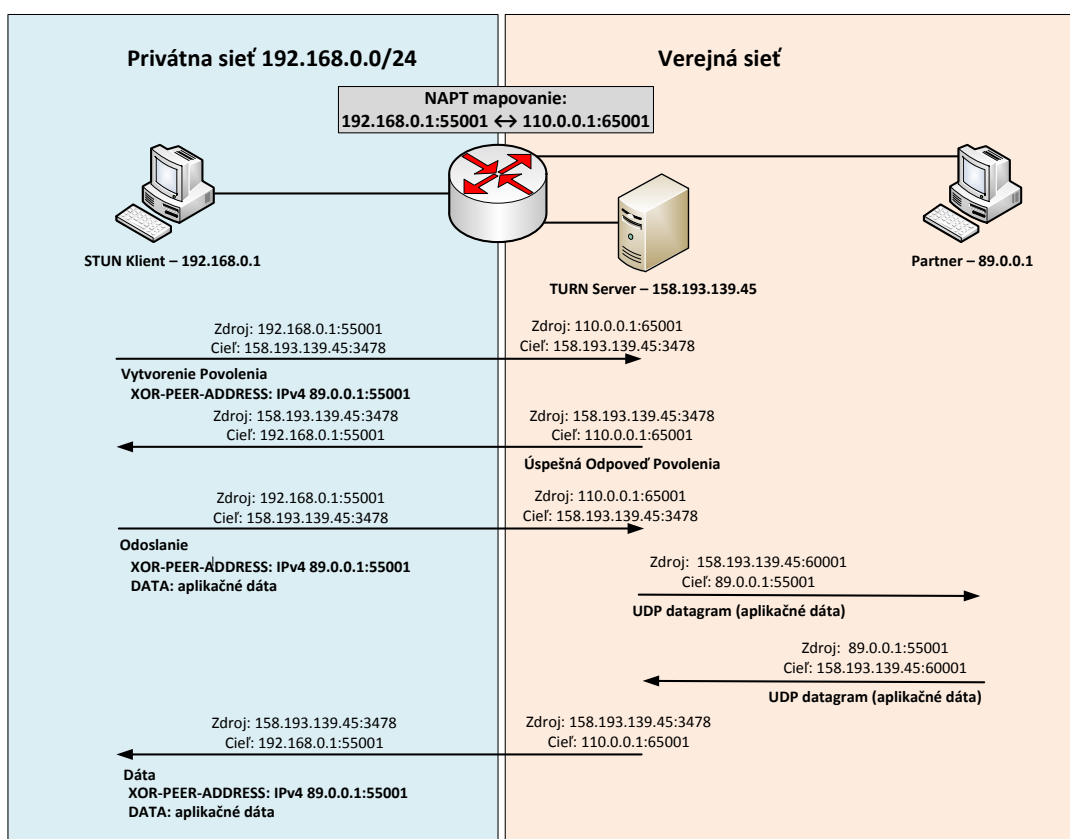


Obrázok 3.11 - Základná TURN komunikácia

Po úspechu požiadavky priradenia je možné prenosovú transportnú adresu použiť na komunikáciu TURN klientov navzájom alebo TURN klienta s iným klientom, ktorý dokáže prijímať a posilať UDP datagramy – **partner**, ktorý má svoju transportnú adresu (táto adresa je ľubovoľná a môže to byť napr. verejná IP adresa alebo STUN reflexívna transportná adresa). Existujú dva spôsoby ktorými môže byť táto komunikácia realizovaná a pri oboch spôsoboch je dôležité zistiť transportnú adresu druhej strany komunikácie (pri protokole SIP je táto informácia vymieňaná prostredníctvom SDP tela SIP *Invite* alebo 200 OK správ):

1. **Metóda Odoslanie a Dáta** – pri tejto metóde sú využívané metódy **Odosielanie** a **Dáta** pre prenos aplikačných dát a metóda **Vytvorenie povolenia** pre definovanie povolených transportných adries partnerov alebo TURN klientov. Prvým krokom TURN klienta je vytvorenie povolenia pre komunikačnú adresu partnera, kedy klient posla požiadavku **Vytvorenie povolenia** (z ang. Create Permission Request) s atribútom **XOR-PEER-ADDRESS**, v ktorom je uvedená transportná adresa partnera (v jednej správe môže byť uvedených súčasne viacero adries). TURN server následne k prenosovej transportnej adrese TURN klienta priradí povolenie pre konkrétnu transportnú adresu partnera a odosiela správu **Úspešné**

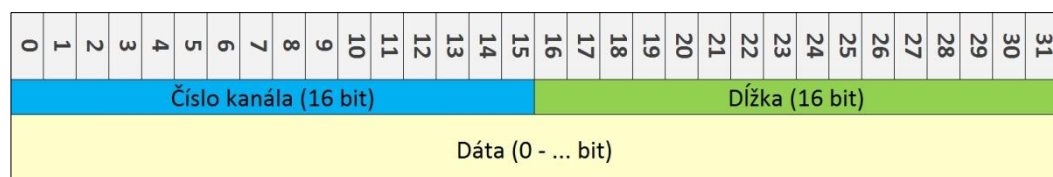
vytvorenie povolenia (z ang. Create Permission Success Response), partner teraz môže využívať prenosovú transportnú adresu na posielanie aplikačných dát v UDP datagramoch TURN klientovi (platnosť jednotlivých povolení má podľa štandardu platnosť päť minút a je potrebné ich pravidelne aktualizovať). Aplikačné dáta, ktoré prídu na server od partnera sú zapuzdrené do TURN správy **Dáta** v atribúte **DATA** a následne preposlané TURN klientovi. V opačnom smere klient posiela aplikačné dáta na predvolenú IP adresu a port (3478) TURN servera v správe **Odoslanie**. V atribúte **DATA** správy **Odoslanie** sú aplikačné dáta a v atribúte **XOR-PEER-ADDRESS** je transportná adresa partnera, pre ktorého sú aplikačné dáta určené. TURN server z atribútu **DATA** vytvorí UDP datagram a pošle ho partnerovi na príslušnú transportnú adresu uvedenú v atribúte **XOR-PEER-ADDRESS**, kde ako zdrojovú adresu UDP datagramu uvedenie prenosovú transportnú adresu TURN klienta. Komunikácia medzi TURN klientom a jedným jeho partnerom je ilustrovaná na Obrázku 3.12.



Obrázok 3.12 - Metódy Odoslanie a Dáta

2. **Metóda rezervovania kanálov** – slúži na rovnaký účel ako predchádzajúca metóda ale s určitými rozdielmi. Vytváranie povolenia nahrádza rezervovanie čísla kanála, o ktoré TURN klient požiada v správe **Rezervovanie kanála**

a príslušnom atribúte *CHANNEL-NUMBER*. Tento obsahuje dvojbajtové číslo (0x4000 – 0x7FFF), predstavujúce číslo kanála priradeného TURN klientom príslušnému partnerovi (partner je definovaný v atribúte *XOR-PEER-ADDRESS*). TURN server overí či transportná adresa partnera nemá už pridelené iné číslo kanála a či je v zadanom rozmedzí (0x4000 – 0x7FFF). Ak sú tieto podmienky splnené server priradí transportnej adrese partnera zvolené číslo kanála a zároveň vytvorí povolenie pre spracovanie UDP datagramov pre túto adresu. Následne odošle odpoveď **Úspešné rezervovanie kanála**. Najväčším rozdielom medzi dvomi metódami je v spôsobe posielania aplikačných dát medzi klientom a serverom. TURN špecifikácia definuje pre túto metódu nový typ správ: **Dáta kanála** (z ang. Channel Data), ktoré nezapadajú do koncepcie STUN správ ako všetky ostatné. Na nasledujúcom obrázku (Obrázok 3.13) je naznačený formát novej TURN správy:



Obrázok 3.13 - Formát správy Dáta kanála

Pole **Číslo kanála** predstavuje rezervované číslo kanála TURN klientom na serveri a pole **Dĺžka** určuje veľkosť aplikačných dát zapísaných v poli **Dáta**. Keďže číslo kanála je úzko spojené s transportnou adresou partnera, posielanie aplikačných dát medzi TURN klientom a TURN serverom prebieha výhradne týmto typom správ. Oba koncové body spojenia dokážu na základe prvých dvoch najvýznamnejších bajtov správy určiť pre koho sú aplikačné dáta určené. Spracovanie rozbalených aplikačných dát zo správy *Dáta kanála* na TURN serveri smerujúcich k partnerovi je totožné s predchádzajúcim prípadom. Komunikácia medzi TURN klientom a jedným jeho partnerom je ilustrovaná na Obrázku 3.14.

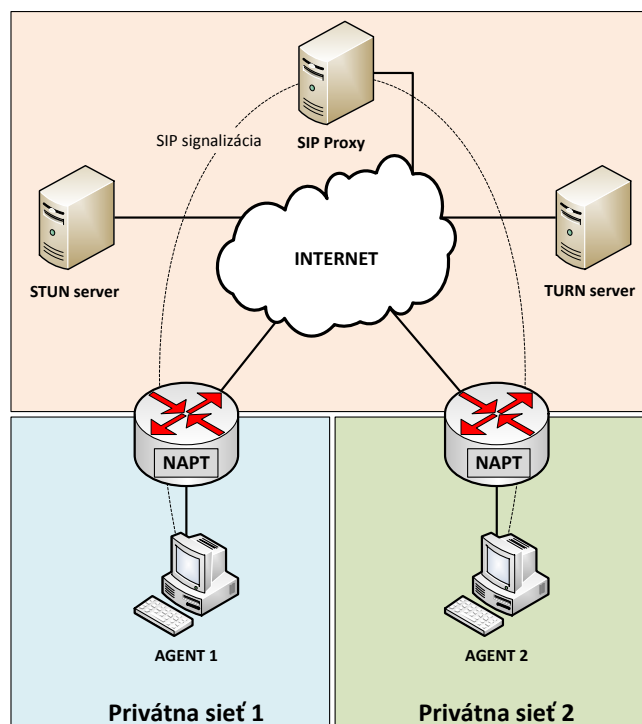
Zabezpečenie využívania TURN metód len v prípade, že iný spôsob komunikácie dvoch koncových bodov nie je možné, rieši protokol ICE, ktorý pracuje s viacerými transportnými adresami koncových bodov. Základnou ideou ICE, ktorý bude popísaný v nasledujúcej podkapitole, je nájsť **najvhodnejší pár transportných adries** pre komunikáciu.

3.4 ICE – Interactive Connectivity Establishment

Protokol ICE je podľa špecifikácie v **RFC 5245** (10) technika prechodu cez NAT pre UDP multimediálne spojenia založené pomocou modelu ponuka/odpoveď (SDP v SIP). Jedná sa o rozšírenie tohto modelu, ktoré do SDP správ namiesto jednej transportnej adresy pre založenie RTP toku dát vloží viacero možných transportných adries – **ICE kandidátov**. Medzi kandidátov sú, okrem iných, zaradené aj transportné adresy získané pomocou techník popísaných v podkapitolách protokolov STUN (nová špecifikácia) a TURN. ICE na rozdiel od iných riešení ponúka možnosť testovania kandidátov a až na základe **úspešnosti testov** je určený **najvhodnejší pár transportných adries** pre komunikáciu (RTP tok dát).

3.4.1 ICE predpoklady

Základom pre ICE sú dva koncové body komunikácie – agenti (SIP klienti), ktorí chcú navzájom komunikovať. Agenti musia byť schopní medzi sebou komunikovať prostredníctvom signalizačného protokolu (bežne je to protokol SIP), ktorý zabezpečí výmenu SDP správ modelu ponuka/odpoveď. ICE úplne **upúšťa od určovania topológie siete** medzi dvomi agentmi (určovanie počtu a typov NAT), namiesto toho dokáže poskytnúť dostatočné množstvo informácií na nájdenie **jednej alebo viacerých komunikačných ciest** medzi agentmi. Základné postavenie jednotlivých ICE komponentov je ilustrované na nasledujúcom obrázku (Obrázok 3.15).



Obrázok 3.15 - ICE topológia

Z uvedeného obrázku vyplýva, že pre implementáciu ICE mechanizmu je potrebné zabezpečiť funkčnosť dvoch typov serverov STUN a TURN. Ako bolo spomenuté v predchádzajúcej podkapitole tieto dva serveri môžu byť zastúpené jedným STUN/TURN serverom, ktorý podporuje obe funkcionality. Pre zabezpečenie minimálne jednej komunikačnej cesty v každom prípade je nutná implementácia protokolu TURN v ICE agentoch, v praxi je však bežné, že sa táto možnosť vynecháva. Dôvodom je využitie iba v prípade, ak je NAT mapovanie klienta závislé na cieľovej IP adrese a porte (symetrické NAT) na úkor vysokých nárokov na prevádzku TURN servera.

3.4.2 Zber kandidátov

Určenie komunikačných ICE kandidátov je základ pre implementáciu ICE mechanizmu. Pod zberom kandidátov rozumieme získanie (určenie) všetkých možných transportných adries agenta. ICE definuje štyri základné typy kandidátov (10):

- **Host'ovský** – transportná adresa na priamo pripojenom rozhraní. Môže to byť fyzické, bezdrôtové, VPN alebo iné rozhranie. Každé rozhranie sa stáva Host'ovským ICE kandidátom.
- **Server-Reflexívny** – reflexívna transportná adresa agenta z pohľadu STUN servera.

- **Prenosový** – prenosová transportná adresa agenta alokovaná na TURN serveri.
- **Partner-Reflexívny** – transportná adresa agenta (A) z pohľadu iného agenta (B). Tento typ kandidáta je určený počas fáze testovania konektivity kandidátov v prípade, že v komunikácii s agentom (B), je zdrojová adresa agenta (A) iná ako Server-Reflexívny kandidát agenta (A).

Druhý a tretí typ kandidáta ICE agent získa prostredníctvom techník, ktoré boli popísané pre protokoly STUN (nová špecifikácia) a TURN. Server-Reflexívny a Prenosový kandidát môžu a nemusia byť určený pre každého Host'ovského kandidáta. Kandidát však musí byť určený pre každý komponent komunikácie (napr. pre komunikáciu v SIP prostredníctvom RTP sú potrebné dva komponenty: RTP a RTCP). V prípade, že ICE agent má jedno fyzické rozhranie a využíva STUN aj TURN server a pre komunikáciu chce využiť RTP, výsledkom zberu je šesť kandidátov (traja pre RTP a traja pre RTCP).

Po zbere kandidátov sú každému z nich priradené dva príznaky **základ** a **priorita**. Každý z týchto príznakov je využívaný v procese testovania kandidátov a ich hodnoty sú určované samostatne pre každého kandidáta. Príznak **základ** sa využíva pre zjednodušenie testovania kandidátov, kedy je vyhodnocovaná „podobnosť“ dvoch takmer rovnakých kandidátov pre dva rôzne komponenty komunikácie (odlišnosť kandidátov je len v čísle portu). Dvaja kandidáti majú **rovnaký základ**, ak sú splnené všetky nasledujúce podmienky (10):

- **typ** ICE kandidáta je u oboch **rovnaký**.
- **Základňa** (lokálna IP adresa agenta, z ktorej bola odoslaná požiadavka pre získanie kandidáta; pre Host'ovského kandidáta je to kandidát samotný) oboch kandidátov je **rovnaká** pričom **číslo portu môže byť odlišné**.
- **IP adresa STUN** alebo **TURN** servera, ktorý bol použitý pre určenie Server-Reflexívneho a Prenosového kandidáta je **rovnaká**.
- Oba kandidáti boli získaný prostredníctvom **rovnakého transportného protokolu** (UDP, TCP, ...).

Príznak **priorita** je použitý pri prioritizovaní kandidátov a vyjadruje prednosť v použití jedného kandidáta pred iným (vyššia priorita je lepšia). Každý kandidát musí mať jedinečnú prioritu, ktorú predstavuje prirodzené číslo od 1 do $2^{31} - 1$ a je určené pomocou nasledujúceho vzťahu (10):

$$priorita = 2^{24} * preferencia\ typu + 2^8 * lokálna\ preferencia + (256 - ID\ komponentu),$$

kde:

- **preferencia typu** je hodnota od 0 do 126 (126 je najväčšia preferencia) a je pridelená ku konkrétnemu typu ICE kandidáta. Pri pridelovaní hodnôt sú podľa štandardu odporúčané nasledujúce hodnoty:
 - **126** – pre Host'ovského kandidáta.
 - **110** – pre Partner-Reflexívneho kandidáta.
 - **100** – pre Server-Reflexívneho kandidáta.
 - **0** - pre Prenosového kandidáta.

Z uvedených hodnôt môžeme vyčítať, že ICE uprednostňuje použitie lokálnej siete pred verejnou. A rovnako, že využitie centrálného bodu na preposielanie dát (TURN server) je najmenej preferovanou možnosťou komunikačnej cesty.

- **Lokálna preferencia** je hodnota od 0 do 65535 (65535 je najväčšia preferencia) a vyjadruje prednosť jednej IP adresy pred inou, z ktorej bol získaný ten istý typ kandidáta (využitie pre agentov, ktorý disponujú viacerými sieťovými kartami alebo komunikujú pomocou oboch IP protokolov – IPv4 aj IPv6).
- **ID komponentu** je hodnota od 1 do 256 a slúži na rozlíšenie dvoch komponentov komunikácie (napr. RTP a RTCP).

ICE agent následne vylúči kandidátov, ktorí nebudú overovaní z dôvodu redundancie. Kandidát je považovaný za **redundantného**, ak je jeho základ a transportná adresa zhodná s iným kandidátom. Vylúčený je ICE kandidát s nižšou prioritou.

Nie všetky SIP telefóny rozumejú ICE protokolu a tak je definovaný aj **predvolený kandidát** (v SDP je predvolený kandidát zapísaný v „c:“ a „m:“ riadkoch), ktorý slúži práve na komunikáciu s agentmi bez ICE podpory. Tento kandidát má byť podľa špecifikácie určený podľa nasledovného poradia (ďalší kandidát je vyberaný, ak predchádzajúci nie je známy) (10):

1. Prenosový kandidát
2. Server-Reflexívny
3. Host'ovský kandidát

Po zbere kandidátov prichádza na rad ďalšia fáza ICE mechanizmu a tou je **testovanie**. Pred testovaním je však potrebné aby jeden agent poznal kandidátov druhého agenta a naopak. Na výmenu kandidátov je bežne využitý SDP model ponuka/odpoveď v signalizačnom protokole SIP. ICE na tento účel využíva nový SDP atribút pre popis média, v ktorom sú zapísané údaje o ICE kandidátovi – **základ, ID komponentu, transportný protokol, prioritu, IP adresu, číslo portu, typ kandidáta** (ICE špecifikácia definuje aj ďalšie atribúty média, ktoré sú využívané napr. pri overovaní agenta, tieto atribúty sú popísané v **RFC 5245**). Príklad zápisu údajov ICE kandidáta v SDP (základ = 1, ID komponentu = 1, transportný protokol = UDP, prioritu = 659136, IP adresa = 192.168.0.4, číslo portu = 56574, typ kandidáta = host):

```
a=candidate:1 1 UDP 659136 192.168.0.4 56574 host
```

3.4.3 Testovanie kandidátov

Fáza testovania začína, keď jeden agent pošle zoznam svojich kandidátov (v SDP) inému agentovi a stáva sa tak **kontrolným agentom**. Agent, ktorý prijme SDP ponuku sa stáva **kontrolovaným agentom** a začína so zberom svojich kandidátov. Zber kandidátov je rovnaký ako bol popísaný v predchádzajúcej časti, pričom kontrolovaný agent po ukončení zberu odošle zoznam svojich kandidátov v SDP kontrolnému agentovi. Hlavnou úlohou kontrolného agenta je určiť konečný pár kandidátov, ktorý bude použitý pre založenie komunikácie.

Párovanie kandidátov je totožné pre oboch agentov a ide o proces, pri ktorom vlastní kandidáti agenta – **lokálni kandidáti**, vytvoria páry s príslušnými **vzdialenými kandidátmi**. Lokálny kandidát so vzdialeným môže vytvoriť pár len ak majú **zhodné číslo komponentu a verziu IP** protokolu. Pre každý pár je následne vypočítaná **priorita páru**, ktorej základ tvoria priority jednotlivých kandidátov. Konkrétna hodnota je určená pomocou vzťahu (10):

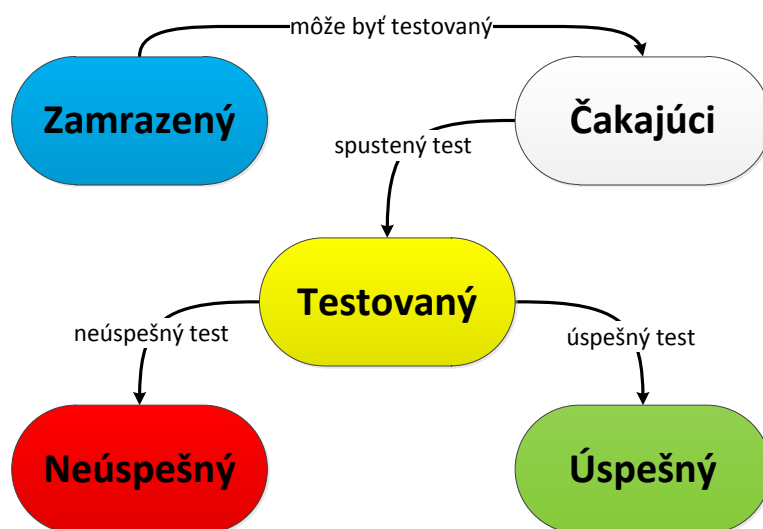
$$\text{priorita páru} = 2^{32} * \text{MIN}(P_L, P_V) + 2 * \text{MAX}(P_L, P_V) + (P_L > P_V ? 1 : 0),$$

kde P_L je priorita lokálneho a P_V vzdialeného kandidáta. MIN a MAX predstavujú funkciu, ktorá vyberie minimum alebo maximum z príslušných hodnôt. Výsledkom vzťahu $(P_L > P_V ? 1 : 0)$ je jedna ak $P_L > P_V$, inak je to nula. Pomocou priorít sú páry **zostupne usporiadané** (ak majú dva páry rovnakú prioritu ich poradie je ľubovoľné).

Pre určenie, či je konkrétny pár funkčný je ho potrebné **otestovať**. Prvým krokom je vybrať zo zoznamu párov tie, ktorých transportnú adresu lokálneho kandidáta nie je možné použiť na odoslanie (je to každý Server-Reflexívny kandidát, pretože z implementačného hľadiska nie je možné vytvoriť soket, ktorého zdrojová adresa je mapovaná transportná adresa klienta). Transportné adresy Server-Reflexívnych kandidátov sú nahradené príslušnými **základňami** (lokálnymi adresami) jednotlivých kandidátov. Dôsledkom tohto kroku je, že v zozname ICE párov sa nachádzajú redundantné páry, z ktorých sú **vyradené** všetky okrem jedného s **najvyššou prioritou**. Ďalším krokom je definovanie stavu jednotlivých párov, pričom pár môže byť v jednom z nasledujúcich stavov (10):

- **zamrazený** – pár ešte nebol testovaný a test nie je možné vykonať pokiaľ test iného páru nebude úspešný.
- **Čakajúci** – test pre tento pár bude vykonaný, keď príslušný pár bude mať najvyššiu prioritu v zozname testovaných párov.
- **Testovaný** – pár je aktuálne v procese testovania.
- **Úspešný** – pár bol otestovaný a test bol úspešný.
- **Neúspešný** – pár bol otestovaný ale test bol neúspešný.

Možný pohyb páru medzi jednotlivými stavmi je ilustrovaný na nasledujúcom diagrame (Obrázok 3.16).



Obrázok 3.16 - Diagram stavov ICE párov

Počiatočný stav pre všetky testované páry je *Zamrazený*. Do stavu *Čakajúci* sa dostane na začiatok pár s **najnižším číslom komponentu a najvyššou prioritou** (pre jeden typ média

sa samostatne vytvoria viaceré zoznamy párov, každý pre jeden komponent média), pár(y) s vyšším číslom komponentu a pár(y) s rovnakým príznakom základ (pri páre je to spojenie príznakov oboch kandidátov) zostávajú v stave *Zamrazený*. Zo stavu *Zamrazený* do stavu *Čakajúci* prechádza pár za predpokladu, že v zozname párov s rovnakým číslom komponentu neexistuje žiaden pár v stave *Čakajúci*. Rovnako za predpokladu, že testovaný pár v zozname párov pre komponent média (A) prešiel do stavu *Úspešný* je pár v zozname pre iný komponent média (B), s rovnakým príznakom základ, presunutý do stavu *Čakajúci*. Tento mechanizmus je využitý najmä z dôvodu aby páry, u ktorých je pravdepodobné zlyhanie (ak zlyhá test kandidát pre jeden komponent média je veľmi pravdepodobné, že dôjde aj k zlyhaniu testu kandidáta pre iný komponent pretože sa kandidáti líšia len v čísle portu), boli testované neskôr napriek tomu, že majú vyššiu prioritu. Pomocou odloženia týchto testov môže ICE agent skôr nájsť úspešný pár, čo je napríklad využívané pri agresívnom type nominácie (bude popísaná v ďalšej časti práce).

Pri samotnom testovaní sú u oboch agentov použité dva typy testov (10):

- **plánovaný** – zo zoznamu párov je vybraný vždy ten s najvyššou prioritou a je následne otestovaný (presunutie páru zo stavu *Čakajúci* do *Testovaný*).
- **Podmienенý** – pri tomto teste agent reaguje na testovacie správy iného agenta prednostným testovaním rovnakého ICE páru.

Oba typy testov sú zabezpečované pomocou správ STUN protokolu modelu požiadavka/odpoveď. STUN požiadavka je posielená z transportnej adresy lokálneho kandidáta (táto adresa musí byť použitá aj pre neskorší tok aplikačných dát) na transportnú adresu vzdialeného kandidáta jednotlivých párov. Požiadavka môže obsahovať (okrem iných) tieto základné atribúty, ktoré rozširujú zoznam už skôr definovaných STUN atribútov (10):

- **PRIORITY** – hodnota tohto atribútu predstavuje prioritu Partner-Reflexívneho kandidáta, v prípade ak si prijatá požiadavka vyžiada zaradenie tohto typu kandidáta do ICE testovania (hodnota je vypočítaná na základe vzťahu pre určenie priority kandidáta).
- **ICE-CONTROLLED** – atribút je do požiadavky zaradený ak požiadavku odosiela kontrolný agent.
- **ICE-CONTROLLING** – na rozdiel od predchádzajúceho je atribút do požiadavky zaradený ak požiadavku odosiela kontrolovaný agent.

- **USE-CANDIDATE** – slúži ako značka pre definovanie páru, ktorý bude použitý pre komunikáciu.

Po prijatí požiadavky vzdialený agent skontroluje, či je zdrojová transportná adresa požiadavky zhodná s transportnou adresou niektorého zo vzdialených kandidátov. V prípade, že tomu tak nie je, agent zaradí medzi svojich ICE kandidátov nového Partner-Reflexívneho kandidáta. Lokálny kandidát, na ktorom bola prijatá požiadavka vytvorí pár so vzdialeným kandidátom. Výsledkom je ICE pár, pre ktorého je potrebné naplánovať podmienený typ testu. Pár je vyhľadaný v zozname párov plánovaného testu a presunutý do zoznamu párov podmieneného testu. Na každú požiadavku vzdialený agent reaguje STUN odpoveďou, v ktorej atribúte *XOR-MAPPED-ADDRESS* uvedie zdrojovú adresu požiadavky. Zdrojová adresa odpovede je transportná adresa, na ktorej bola prijatá požiadavka. Cieľová adresa odpovede je transportná adresa, z ktorej bola požiadavka prijatá. Lokálny agent vyhodnotí testovanie ako úspešné (stav páru sa mení z *Testovaný* na *Úspešný*) ak sú splnené nasledujúce podmienky (10):

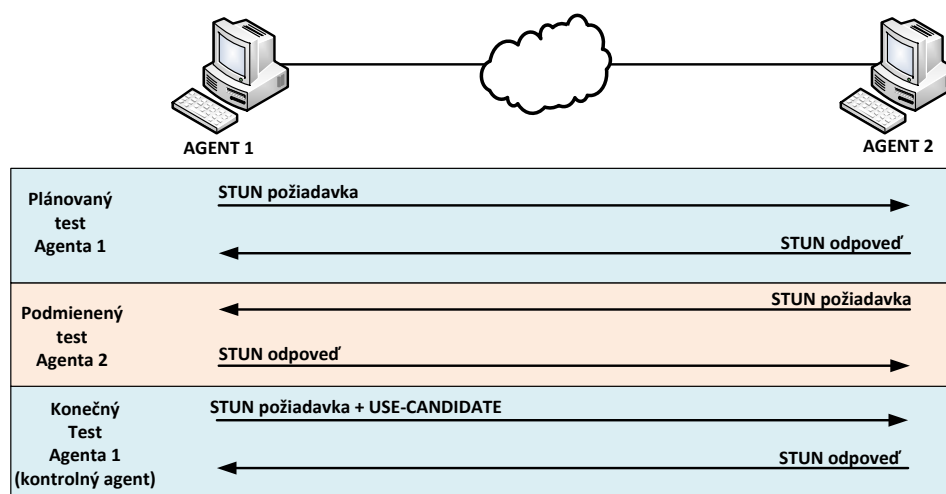
- bola **prijatá STUN odpoveď** od vzdialeného agenta (úspešná STUN transakcia).
- **Zdrojová transportná adresa odpovede je zhodná s cieľovou transportnou adresou požiadavky** (v prípade že adresy nie sú zhodné, lokálny agent definuje nového vzdialeného Partner-Reflexívneho kandidáta a pár je presunutý do stavu *Neúspešný*).
- **Cieľová transportná adresa odpovede je zhodná so zdrojovou transportnou adresou požiadavky.**

Lokálny agent po doručení odpovede porovná transportnú adresu lokálneho kandidáta s adresou zapísanou v atribúte *XOR-MAPPED-ADDRESS* odpovede. V prípade že adresy nie sú zhodné, agent definuje nového lokálneho Partner-Reflexívneho kandidáta a nový ICE pár zhodný s aktuálne testovaným párom. V novom páre je však ako lokálny kandidát uvedený práve nový Partner-Reflexívny kandidát. Pôvodne testovaný alebo nový ICE pár je zaradený do zoznamu **platných párov**. Následne je vybraný ďalší pár na testovanie. Je ním prvý pár zo zoznamu párov podmieneného testu, ak sa v tomto zozname nejaký pár nachádza. V prípade, že je tento zoznam prázdny je vybraný pár s najvyššou prioritou v stave *Čakajúci* zo zoznamu párov plánovaného testu. Ak sa v zozname párov nenachádza žiaden pár v stave *Čakajúci*, je do toho stavu presunutý pár s najvyššou prioritou v stave *Zamrazený*. Testovanie párov končí, ak sú všetky páry v stave *Úspešný* alebo *Neúspešný*.

Pri konečnom vyhodnocovaní testovania a určovaní páru, ktorý bude použitý na komunikáciu dvoch agentov môžu byť použité dve metódy nominácie (10):

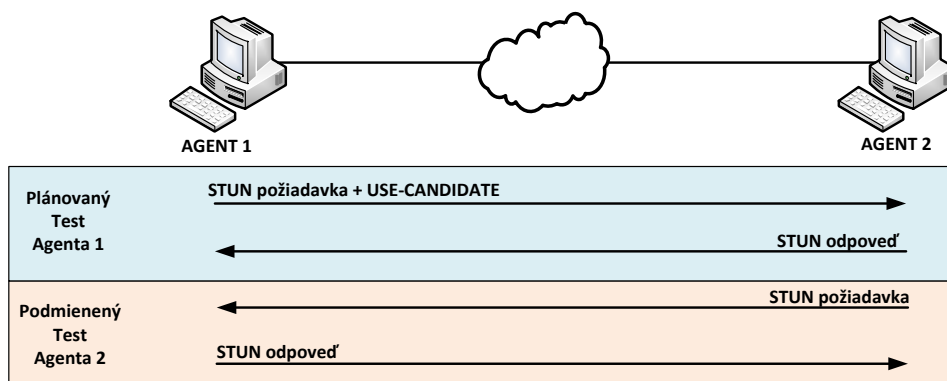
- **obyčajná nominácia.**
- **Agresívna nominácia.**

Obe metódy vychádzajú z rovnakého základu, ktorým je postupné testovanie párov od najvyššej po najnižšiu prioritu. Rozdiel je v ukončení ICE testovania. Pri metóde **obyčajnej nominácie** agent postupne testuje páry pokiaľ nie je v zozname platných párov aspoň jeden pár, pre každý typ média. Následne agent vyberie jeden pár zo zoznamu platných párov (špecifikácia nedefinuje, ktorý pár je najvhodnejší, toto rozhodnutie je na konkrétnej implementácii agenta) a pre tento pár vykoná ešte jeden test. V tomto prípade však medzi atribúty STUN požiadavky kontrolný agent zaradi atribút *USE-CANDIDATE*, čím naznačí, že zvolený pár **má byť použitý pre komunikáciu**. Ak kontrolovaný agent prijme požiadavku s týmto atribútom, na komunikáciu použije zvolený pár a odošle STUN odpoveď. Komunikácia dvoch agentov pri obyčajnej nominácii je ilustrovaná na Obrázku 3.17 (10).



Obrázok 3.17 - Obyčajná nominácia

Rozdiel v metóde **agresívnej nominácie** je, že atribút *USE-CANDIDATE* sa nachádza v každej STUN požiadavke a po prvom úspešnom teste je testovanie ukončené. Prvý úspešne testovaný pár je následne použitý na komunikáciu. Výhodou druhej metódy je v rýchlosti zahájenia komunikácie. Komunikácia dvoch agentov pri agresívnej nominácii je ilustrovaná na Obrázku 3.18.



Obrázok 3.18 - Agresívna nominácia

Výsledkom jednej alebo druhej metódy je **ICE komunikačný pár**, ktorý je možné použiť pri výmene aplikačných dát. V prípade SIP je to výmena RTP multimediálnych dát medzi dvomi alebo viacerými SIP telefónmi. V predchádzajúcich častiach bol popísaný princíp mechanizmu ICE, konkrétnejšie kroky pri jednotlivých bodoch (zmeny stavov ICE páru, stavy ICE zoznamov jednotlivých párov, zaradenie Partner-Reflexívneho kandidáta a ďalšie detaily ICE mechanizmu) je možné nájsť v príslušnom **RFC 5245**. Konkrétny, komplexnejší príklad ICE komunikácie bude uvedený v ďalšej časti tejto práce.

3.5 Riešenia prechodu SIP cez NAT mimo klienta

Všetky metódy a mechanizmy prechodu cez NAT, ktoré boli doteraz popísané v práci sú orientované na vedomosť SIP klienta o tom, že medzi ním a iným klientom je zariadenie s mechanizmom NAT. Je teda na klientovi aby dokázal zistiť svoju verejnú transportnú adresu, adresu STUN/TURN servera, overil komunikačné cesty a podnikol ďalšie kroky, ktoré boli spomenuté. Existujú však riešenia, pri ktorých sa klient nezaujíma o topológiu siete, cez ktorú chce komunikovať. Pri týchto riešeniach sú použité techniky, pri ktorých sú za schopnosť klienta komunikovať, aj keď sa nachádza za NAT, zodpovedné iné zariadenia.

Jedným zo zariadení, ktoré dokážu zabezpečiť komunikáciu klientov za NAT je **ALG – Application Layer (Level) Gateway** (v preklade brána aplikačnej vrstvy). V prípade protokolu SIP ALG prepisuje SDP telo SIP správy, kde sú podľa potreby lokálne transportné adresy nahrádzané externými transportnými adresami a naopak. Na založenie RTP toku dát sú teda použité „prepísané“ adresy. Ďalším navrhovaným riešením je protokol **MIDCOM** (Middlebox Communication protocol), pomocou ktorého je možné

riadiť NAT mapovania na príslušných zariadeniach (viac o tomto protokole je možné nájsť v **RFC 3303**).

Ďalšiu skupinu možností tvoria mechanizmy, ktoré sú podobné riešeniu TURN. V princípe ide o techniku, v ktorej sú aplikačné dáta (akýkoľvek typ média) klientov preposielané prostredníctvom jedného bodu na verejnom internete. Jedným z týchto riešení je RTP Proxy, pri ktorom sa vyžaduje špeciálne nastavenie SIP Proxy servera. Podobnosť s protokolom TURN je v zamieňaní transportných adries za adresy rezervované na RTP Proxy serveri. Táto zámena však nie je riadená klientom ale SIP Proxy serverom. Riešenie, v ktorom sú spolu zastúpené SIP Proxy a RTP Proxy možno nájsť aj pod názvom B2BUA – Back-2-Back User Agent (toto riešenie rozdeľuje SIP aj RTP komunikáciu na dve časti, pričom B2BUA zastupuje oboch SIP UA a je umiestnený na verejnom internete).

4 REALIZÁCIA A TESTOVANIE

V tejto kapitole bude popísané testovanie jednotlivých riešení prechodu cez NAT, ktoré boli teoreticky popísané v predchádzajúcich častiach práce. V rámci SIP architektúry bola ako SIP Registrar/Proxy server použitá voľne dostupná implementácia Kamilio v3.2.0. Inštalovaný SIP server počúva na IP adrese 158.193.139.45 a predvolenom SIP porte 5060.

4.1 Serverová STUN/TURN entita

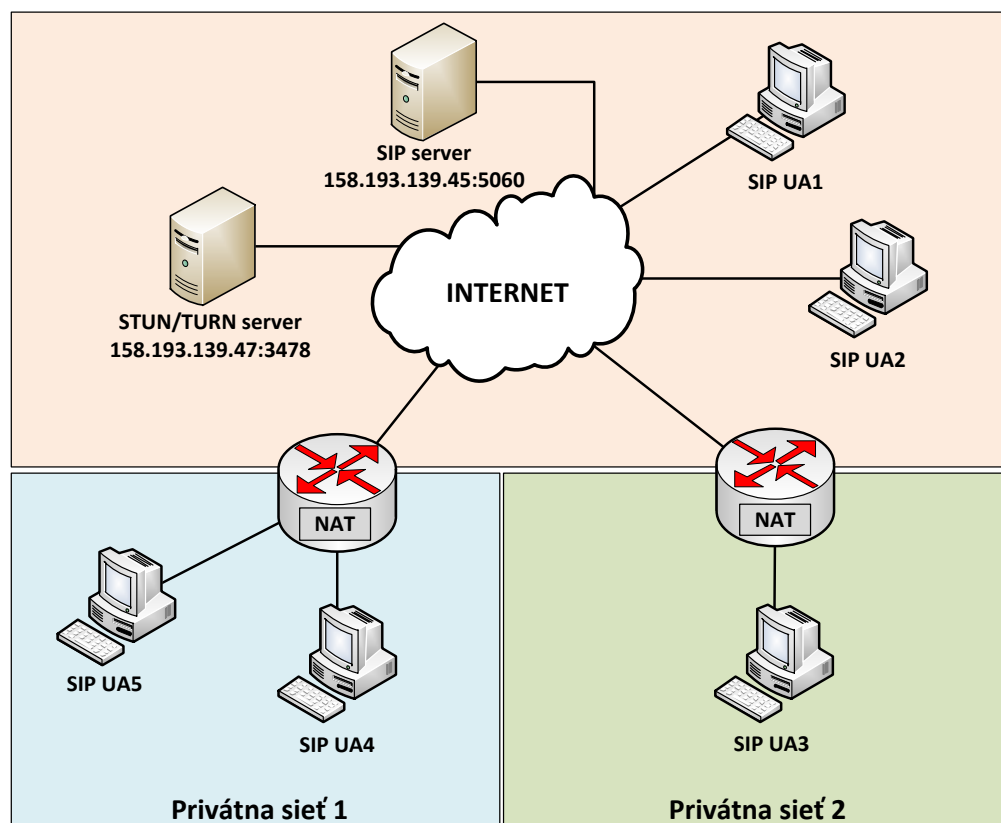
V čase riešenia tejto práce bolo dostupných viacero voľne dostupných implementácií STUN servera avšak len málo implementácií podporujúcich funkcionality TURN servera. Pre testovanie starej špecifikácie STUN protokolu bola použitá implementácia STUN servera, ktorá je voľne dostupná z repozitárov pre operačný systém Debian. Návod na inštaláciu a nastavenie tejto konkrétnej implementácie je možné nájsť na (11). Ako STUN server je možné využiť aj niektorý z voľne dostupných STUN serverov, z ktorých časť je možné nájsť v zozname verejných STUN serverov na (12). Väčšina z verejných STUN serverov a rovnako aj implementácia dostupná z repozitárov pre OS Debian, podporujú komunikáciu so STUN klientom prostredníctvom STUN protokolu starej aj novej špecifikácie.

Pri hľadaní implementácie TURN servera sme narazili na problém s malým počtom voľne dostupných riešení. Väčšinou sa jednalo o riešenia s otvoreným zdrojovým kódom, ktoré boli nedostatočne dokumentované a nepodarilo sa ich úspešne nasadiť v sieťovom prostredí. Nakoniec sa nám však podarilo nájsť funkčnú implementáciu TURN servera – **Restund v0.4.0**, ktorá podporuje funkcionality oboch protokolov STUN a TURN. Návod na inštaláciu a nastavenie Restund servera je možné nájsť na (13). Uvedený návod umožňuje nastavenie Restund STUN/TURN servera bez možnosti overenia užívateľa, keďže dokumentácia k Restund dostatočne neuvádza spôsob nastavenia tejto funkcionality. Napriek tomu, že štandard uvádza túto funkcionality ako nutnú k implementácii TURN klientov a serverov, Restund dokáže odpovedať aj na požiadavky, ktoré neobsahujú atribúty potrebné k overeniu klienta. Takisto voľne dostupný SIP klient X-lite 4 (14) poskytuje možnosť vytvorenia STUN/TURN požiadavky bez nutnosti zadať overovacie údaje klienta. Riešenie TURN servera, ktoré podporuje možnosť overenia

klienta poskytuje projekt **numb** (15). Numb umožňuje po registrácii užívateľa využívať projektový STUN/TURN server. V čase testovania však tento server nedokázal poskytnúť funkcionality preposielania RTP toku dát medzi dvomi SIP telefónmi.

4.2 Analýza problematiky – možné scenáre

Pri riešení problematiky prechodu SIP cez NAT je jedným z najdôležitejších aspektov poloha SIP UA v sieti. Pokiaľ by obaja SIP UA, ktorý chcú komunikovať, mali zabezpečené voľne použiteľné (v sieti sa nenachádza žiaden firewall) verejné transportné adresy, nie je potrebný žiaden podporný mechanizmus pre založenie RTP toku dát medzi týmito UA. Príkladom je postavenie ilustrované na Obrázku 4.1, v prípade, že by medzi sebou chceli komunikovať SIP UA1 a SIP UA2.



Obrázok 4.1 - Testovacia topológia

Takéto postavenie dvoch SIP UA je však v praxi veľmi málo použiteľné a obmedzuje sa skôr na testovanie v laboratórnych podmienkach. Takisto málo pravdepodobné, ale možné je postavenie, pri ktorom je nutné riešiť prechod cez NAT v prípade, že jeden SIP UA má verejnú transportnú adresu a druhý sa nachádza v privátnej sieti, ktorá je oddelená od verejnej siete zariadením s mechanizmom NAT. Príklad takéhoto postavenia SIP UA je

ilustrované na Obrázku 4.1, v prípade, že by spolu chceli komunikovať SIP UA1 a SIP UA3. Oveľa bežnejšími situáciami sú postavenia klientov, pri ktorých sa obaja klienti nachádzajú v privátnych sieťach. Môžu nastať dve možnosti takéhoto postavenia. Prvou možnosťou je, že sa dvaja SIP UA nachádzajú v dvoch rôznych privátnych sieťach s odlišnými verejnými transportnými adresami, ktoré sú pridelené NAT zariadeniam. Na Obrázku 4.1 ide o postavenie, pri ktorom by spolu chceli komunikovať napríklad SIP UA3 a SIP UA4. Druhou možnosťou je postavenie SIP UA4 a SIP UA5 z Obrázku 4.1, pri ktorom sú obaja SIP UA umiestnený v rovnakej privátnej sieti. V nasledujúcej tabuľke (Tabuľka 4.1) sú uvedené možné postavenia dvoch SIP UA, na ktoré sa budú odvolávať popisy jednotlivých štandardizovaných riešení v ďalších častiach tejto práce (stĺpec „Symetrické NAT“ vyjadruje, či minimálne jedno z NAT zariadení v sieti používa NAT mechanizmus, ktorý je závislý na transportnej adrese cieľového koncového bodu – symetrické NAT).

Označenie postavenia	Komunikujúci č.1	Komunikujúci č.2	Symetrické NAT
Postavenie č.1	SIP UA1	SIP UA3	nie
Postavenie č.2	SIP UA1	SIP UA3	áno
Postavenie č.3	SIP UA3	SIP UA4	nie
Postavenie č.4	SIP UA3	SIP UA4	áno
Postavenie č.5	SIP UA4	SIP UA5	nie
Postavenie č.6	SIP UA4	SIP UA5	áno

Tabuľka 4.1 - Postavenia dvoch SIP UA

4.3 Testovanie voľne dostupných implementácií STUN, TURN a ICE

Testovanie jednotlivých špecifikácií protokolov STUN, TURN a ICE bolo realizované v laboratórnych a domácich podmienkach, pričom boli modelované postavenia SIP UA z Tabuľky 4.1. Na testovanie boli (pokiaľ nebude uvedené inak) využívané implementácie Restund ako STUN/TURN server a X-lite 4 ako STUN/TURN/ICE klient. Dôvodom využitia týchto implementácií bola najlepšia podpora testovaných protokolov. Na sledovanie a analýzu komunikácie bol na strane klienta použitý softvér **Wireshark** a na strane servera softvér **tcpdump**.

4.3.1 Stará špecifikácia STUN – klasický STUN

Ako bolo spomenuté v predchádzajúcich častiach práce, klasický STUN, sa v praxi ukázal ako nevhodné riešenie prechodu cez NAT. Napriek tomu je riešenie stále možné nájsť v niektorých implementáciách SIP UA. Starú špecifikáciu STUN protokolu

využívajú aj rôzne jednoduché aplikácie, v ktorých je implementovaný STUN klient a na základe komunikácie so STUN serverom dokážu odhaliť mapovanú transportnú adresu klienta a určiť, či pomocou tejto adresy bude možné s klientom komunikovať (príklad takejto aplikácie - WinStun je možné nájsť na CD nosiči priloženom k práci: CD/WinStun/WinStun.msi). Testované implementácie sa však nesnažili stanoviť konkrétnu variáciu NAT ale len určiť, či komunikácia cez príslušné NAT zariadenie, **bude alebo nebude možná**. Toto správanie aplikácií bolo určené na základe sledovania komunikácie medzi STUN klientom a serverom, v ktorej boli vynechané niektoré kroky pre určenie konkrétnej variácie NAT (v rámci testovania nebol nájdený STUN klient, ktorý by bol implementovaný podľa štandardu a dodržiaval postupnosť krokov z Obrázku 3.5). Pri príklade, ktorý bude uvedený bola ako STUN server použitá vyššie spomínaná implementácia STUN servera z repozitárov OS Debian a ako STUN klient aplikácia X-lite 4. Pri starej STUN špecifikácii musí STUN server disponovať dvomi IP adresami a dvomi číslami portov, ktoré musia byť použiteľné v ľubovoľnej kombinácii. Použité boli IP adresy: 158.193.139.45 a 158.193.139.47 a čísla portov: 3478 a 3479.

Uvedieme príklad, v ktorom na základe sledovania správ dokážeme prehlásiť, že mapovanie na NAT zariadení je závislé od transportnej adresy cieľového koncového bodu (symetrické NAT). Sledovaná komunikácia medzi STUN klientom a serverom v programe Wireshark je ilustrovaná na Obrázoku 4.2:

No.	Source	Destination	Protocol	Info
106	192.168.0.101	158.193.139.45	CLASSIC-STUN	Message: Binding Request
107	158.193.139.45	192.168.0.101	CLASSIC-STUN	Message: Binding Response
108	192.168.0.101	158.193.139.47	CLASSIC-STUN	Message: Binding Request
109	158.193.139.47	192.168.0.101	CLASSIC-STUN	Message: Binding Response

Obrázok 4.2 – STUN komunikácia

STUN klient v správe číslo 106 posiela *Požiadavku spojovania (Binding Request)* na predvolenú transportnú adresu STUN servera 158.193.139.45:3478. Server na požiadavku odpovedá *Odpoveďou spojovania (Binding Response)* (správa číslo 107), v ktorej atribúte *MAPPED-ADDRESS* je uvedená mapovaná transportná adresa klienta 85.248.45.84:59559 (Obrázok 4.3):

```
Attribute: MAPPED-ADDRESS
Attribute Type: MAPPED-ADDRESS (0x0001)
Attribute Length: 8
Protocol Family: IPv4 (0x0001)
Port: 59559
IP: 85.248.45.84 (85.248.45.84)
```

Obrázok 4.3 - Mapovaná transportná adresa

V tej istej odpovedi servera je uvedený aj atribút *CHANGED-ADDRESS*, v ktorom je uvedená iná transportná adresa STUN servera, konkrétne 158.193.139.47:3478. STUN klient následne posielajú novú *Požiadavku spojovania* práve na adresu 158.193.139.47:3478 (správa číslo 108). Server na požiadavku odpovedá *Odpoveďou spojovania* (správa číslo 109), v ktorej atribúte *MAPPED-ADDRESS* je uvedená **nová** mapovaná transportná adresa klienta 85.248.45.84:63383 (Obrázok 4.4):

```
Attribute: MAPPED-ADDRESS
Attribute Type: MAPPED-ADDRESS (0x0001)
Attribute Length: 8
Protocol Family: IPv4 (0x0001)
Port: 63383
IP: 85.248.45.84 (85.248.45.84)
```

Obrázok 4.4 – Nová mapovaná transportná adresa

Keďže STUN požiadavky boli posielané na rôzne transportné adresy STUN servera a mapované transportné adresy boli odlišné, je možné určiť, že mapovanie na NAT zariadení je závislé od transportnej adresy cieľového koncového bodu a jedná sa teda o **symetrické NAT**. Napriek tomu, že STUN klient dokázal určiť, že v tomto prípade nebude RTP komunikácia možná, v SDP modeli ponuka/odpoveď by uviedol práve určenú mapovanú transportnú adresu. Tento krok by mal za následok neúspešné zostavenie spojenia (zostavenie spojenia je pre obe špecifikácie STUN rovnaké a konkrétny príklad bude uvedený v časti 4.3.2). Sledovanú komunikáciu vyššie uvedeného príkladu je možné nájsť na CD nosiči priloženom k práci (CD/Klasicky_Stun/priklad.pcap) s príslušným filtrom pre Wireshark (CD/Klasicky_Stun/priklad.txt).

4.3.2 Nová špecifikácia STUN

Ako bolo spomenuté v predchádzajúcich častiach práce, nová špecifikácia STUN protokolu upúšťa od určovania variácie NAT. V rámci implementácií SIP UA sa využitie STUN protokolu zúžilo na **určenie mapovanej transportnej adresy**. Túto adresu volajúci SIP UA získa (prostredníctvom integrovaného STUN klienta) od STUN servera pred

No.	Source	Destination	Protocol	Info
259	158.193.139.45	10.0.0.3	SIP/SDP	Request: INVITE sip:user2@158.193.139.214:9254;transport=udp;rinst
260	10.0.0.3	158.193.139.47	STUN	Binding Request
261	10.0.0.3	158.193.139.47	STUN	Binding Request
262	158.193.139.47	10.0.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 158.193.139.214:62110
263	158.193.139.47	10.0.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 158.193.139.214:62111
264	10.0.0.3	158.193.139.45	SIP	Status: 100 Trying
265	10.0.0.3	158.193.139.45	SIP	Status: 180 Ringing
274	10.0.0.3	158.193.139.45	SIP/SDP	Status: 200 OK, with session description

Frame 274: 935 bytes on wire (7480 bits), 935 bytes captured (7480 bits)

Ethernet II, Src: D-Linkin_70:ed:6e (1c:af:f7:70:ed:6e), Dst: Cisco_e2:e3:05 (00:23:5e:e2:e3:05)

Internet Protocol Version 4, Src: 10.0.0.3 (10.0.0.3), Dst: 158.193.139.45 (158.193.139.45)

User Datagram Protocol, Src Port: 9254 (9254), Dst Port: sip (5060)

Session Initiation Protocol

Status-Line: SIP/2.0 200 OK

Message Header

Message Body

Session Description Protocol

Session Description Protocol Version (v): 0

Owner/Creator, Session Id (o): - 12977314817175108 1 IN IP4 10.0.0.3

Session Name (s): CounterPath X-Lite 4.1

Connection Information (c): IN IP4 158.193.139.214

Time Description, active time (t): 0 0

Media Description, name and address (m): audio 62110 RTP/AVP 0 8 101

Obrázok 4.6 - Komunikácia volaného SIP UA

Volajúci SIP UA si po doručení SIP 200 OK uloží transportnú adresu volaného SIP UA z SDP. Teraz obaja SIP UA poznajú navzájom svoje verejné transportné adresy. RTP tok multimediálnych dát obaja SIP UA založia na príslušné transportné adresy opačného koncového bodu získané z SDP modelu ponuka/odpoveď. Ilustráciu sledovanej RTP komunikácie oboch SIP UA je možné nájsť v prílohe č.1. Kompletnú, vyššie popísanú, komunikáciu je možné nájsť aj na priloženom CD k práci (CD/Novy_Stun/).

Úspešné založenie RTP spojenia pomocou techník novej špecifikácie STUN protokolu je možné len v obmedzenej miere. Konkrétne ide o **Postavenie č.1** a **Postavenie č.3** dvoch SIP UA z **Tabuľky 4.1**. V týchto konkrétnych prípadoch je možné využiť mapované transportné adresy SIP UA za NAT. V prípade Postavení č.2 a č.4 spôsobí problémy pri založení spojenia správanie NAT zariadenia. Ako je známe z definície symetrického NAT, mapovanú transportnú adresu môže využiť len koncový bod, pre ktorý bolo mapovanie vytvorené. Mapovanie by bolo však vytvorené pre transportnú adresu STUN servera, ktorá určite nebude zhodná s mapovanou (zdrojovou) transportnou adresou RTP toku dát. Pri Postaveniach č.5 a č.6 by obaja SIP UA získali rovnakú mapovanú transportnú adresu a RTP tok dát by smerovač nedokázal správne smerovať (zdrojová IP adresa RTP toku dát by bola zhodná s cieľovou IP adresou). Pri Postavení č.6 by sa k uvedenému problému navyše pridalo obmedzenie spojené so správaním symetrického NAT.

4.3.3 TURN

Protokol TURN ako riešenie prechodu cez NAT umožňuje komunikáciu dvoch SIP UA v ľubovoľnom postavení (Postavenia č.1 – č.6 z Tabuľky 4.1) ak obaja SIP UA majú integrovaného TURN klienta. Jedinou možnosťou kedy by napriek nasadeniu TURN servera nebola možná výmena multimediálnych dát je v prípade, ak by medzi sebou komunikovali SIP UA1 s integrovaným TURN klientom a SIP UA2 bez integrovaného TURN klienta nachádzajúci sa za symetrickým NAT. SIP UA2 by v tomto prípade vystupoval ako (TURN) partner. TURN sever SIP UA1 by ale nemohol využiť mapovanú transportnú adresu partnera kvôli vyššie spomínaným obmedzeniam symetrického NAT.

Je nutné pripomenúť, že ak aspoň jeden SIP UA využíva TURN protokol, **nie je komunikácia** medzi dvomi SIP UA **P2P**. V prípade TURN riešenia sa totiž P2P delí na dvojnásobnú klient/server komunikáciu.

Testovanie TURN protokolu bolo realizované pomocou implementácií X-lite 4 a Restund. K úspešnej výmene aplikačných (hlasových) dát medzi dvomi SIP UA však došlo len v prípade ak obaja SIP UA mali zapnuté použitie TURN protokolu a na výmenu aplikačných dát bola využitá TURN metóda rezervovania kanálov (túto metódu však nie je možné manuálne zvoliť a výber metódy odosielania aplikačných dát je teda náhodný, čo má za následok, že nie vždy je výmena multimediálnych dát úspešná). Dôvodom nefunkčnosti TURN metódy, ktorá využíva STUN správy *Odosielanie* a *Dáta*, bolo nevytvorenie povolenia (správa *Vytvorenie povolenia*) integrovaným TURN klientom v SIP UA X-lite 4 pre spracovanie aplikačných dát z/pre konkrétnu transportnú adresu iného koncového bodu spojenia (TURN klient alebo TURN partner). Na strane Restund TURN servera bolo problémom spracovanie aplikačných dát, ktoré boli na server doručené prostredníctvom protokolu RTP. Restund STUN/TURN server nedokázal takto doručené dáta od TURN partnera spracovať a preposlať k TURN klientovi.

Ako príklad TURN komunikácie bude z vyššie spomenutých dôvodov uvedená len komunikácia dvoch SIP UA s integrovaným TURN klientom prostredníctvom metódy rezervovania kanálov (SIP UA sú v Postavení č.6 z Tabuľky 4.1). Volajúci SIP UA pred odoslaním SIP *Invite* správy žiada prostredníctvom integrovaného TURN klienta o pridelenie dvoch (RTP a RTCP) prenosových transportných adries TURN server. Žiadosti sú realizované prostredníctvom STUN správy *Požiadavka priradenia* (*Allocate Request*), na ktorú TURN server odpovedá *Úspešnou odpoveďou priradenia* (*Allocate*

Success Response), v ktorej atribúte *XOR-RELAYED-ADDRESS* je uvedená príslušná prenosová transportná adresa. Do SDP tela SIP *Invite* správy je v tomto prípade zapísaná prenosová transportná adresa s párnym číslom portu (RTP). Úvodná komunikácia medzi volajúcim SIP UA a TURN serverom je ilustrovaná na Obrázku 4.7.

No.	Source	Destination	Protocol	Info
197	192.168.0.105	158.193.139.47	STUN	Allocate Request UDP
198	158.193.139.47	192.168.0.105	STUN	Allocate Success Response XOR-RELAYED-ADDRESS: 158.193.139.47:36440
199	192.168.0.105	158.193.139.47	STUN	Allocate Request UDP
200	158.193.139.47	192.168.0.105	STUN	Allocate Success Response XOR-RELAYED-ADDRESS: 158.193.139.47:36441
201	192.168.0.105	158.193.139.45	SIP/SDP	Request: INVITE sip:user1@158.193.139.45, with session description

Frame 201: 824 bytes on wire (6592 bits), 824 bytes captured (6592 bits) Ethernet II, Src: LiteonTe_49:c2:94 (74:de:2b:49:c2:94), Dst: Cisco-Li_00:40:0b (00:1e:e5:00:40:0b) Internet Protocol Version 4, Src: 192.168.0.105 (192.168.0.105), Dst: 158.193.139.45 (158.193.139.45) User Datagram Protocol, Src Port: 42640 (42640), Dst Port: sip (5060) Session Initiation Protocol - Request-Line: INVITE sip:user1@158.193.139.45 SIP/2.0 - Message Header - Message Body - Session Description Protocol - Session Description Protocol Version (v): 0 - Owner/Creator, Session Id (o): - 12974317523877764 1 IN IP4 192.168.0.105 - Session Name (s): CounterPath X-Lite 4.1 - Connection Information (c): IN IP4 158.193.139.47 - Time Description, active time (t): 0 0 - Media Description, name and address (m): audio 36440 RTP/AVP 107 0 8 101

Obrázok 4.7 – Komunikácia volajúceho SIP UA

Z uvedeného obrázku vyplýva, že SIP UA boli úspešne pridelené dve prenosové transportné adresy: 158.193.139.47:36440 a 158.193.139.47:36441. Uvedenou SIP *Invite* správou, ktorá je odoslaná na SIP Proxy server je zahájený SDP model ponuka/odpoveď.

Volaný SIP UA po doručení SIP *Invite* správy rovnako začína proces určenia svojich prenosových transportných adries, ktorý je rovnaký ako pri volajúcom SIP UA. V tomto prípade však volajúci zapíše získanú prenosovú transportnú adresu (s párnym číslom portu) do SDP tela SIP *200 OK* správy a táto je odoslaná na SIP Proxy server (ukončenie SDP modelu ponuka/odpoveď). Sledovaná komunikácia volaného SIP UA po doručení SIP *Invite* správy je ilustrovaná na Obrázku 4.8.

No.	Source	Destination	Protocol	Info
112	158.193.139.45	192.168.0.104	SIP/SDP	Request: INVITE sip:user1@85.248.45.84:60081;rinstance=9718b648adee
113	192.168.0.104	158.193.139.47	STUN	Allocate Request UDP
114	192.168.0.104	158.193.139.45	SIP	Status: 180 Ringing
115	158.193.139.47	192.168.0.104	STUN	Allocate Success Response XOR-RELAYED-ADDRESS: 158.193.139.47:34442
116	192.168.0.104	158.193.139.47	STUN	Allocate Request UDP
117	158.193.139.47	192.168.0.104	STUN	Allocate Success Response XOR-RELAYED-ADDRESS: 158.193.139.47:34443
130	192.168.0.104	158.193.139.45	SIP/SDP	Status: 200 OK, with session description
Frame 130: 951 bytes on wire (7608 bits), 951 bytes captured (7608 bits) Ethernet II, Src: AskeyCom_30:c7:c2 (00:21:63:30:c7:c2), Dst: Cisco-Li_00:40:0b (00:1e:e5:00:40:0b) Internet Protocol Version 4, Src: 192.168.0.104 (192.168.0.104), Dst: 158.193.139.45 (158.193.139.45) User Datagram Protocol, Src Port: 50516 (50516), Dst Port: sip (5060) Session Initiation Protocol Status-Line: SIP/2.0 200 OK Message Header Message Body Session Description Protocol Session Description Protocol version (v): 0 Owner/Creator, Session Id (o): - 12974317528379355 1 IN IP4 192.168.0.104 Session Name (s): CounterPath X-Lite 4.1 Connection Information (c): IN IP4 158.193.139.47 Time Description, active time (t): 0 0 Media Description, name and address (m): audio 34442 RTP/AVP 107 0 8 101				

Obrázok 4.8 - Komunikácia volaného SIP UA

Samotná výmena prenosových transportných adries SIP UA na založenie multimediálneho spojenia pri TURN metóde rezervovania kanálov nestačí. Každý SIP UA prostredníctvom integrovaného TURN klienta musí definovať číslo kanála pre príslušnú prenosovú transportnú adresu iného SIP UA, aby bolo možné využívať správy *Dáta kanála* na posielanie multimediálnych dát. Rezervovanie príslušných čísel kanálov je realizované prostredníctvom správy *Rezervovanie kanála (Channel-Bind Request)*, v ktorej atribúte *XOR-PEER-ADDRESS* je uvedená transportná adresa iného koncového bodu. Proces rezervovania kanálov prebieha u oboch SIP UA rovnakým spôsobom po úspešnom určení vlastných prenosových transportných adries a doručení prenosových transportných adries vzdialeného SIP UA. V nasledujúcej časti bude uvedený proces rezervovania kanálov z pohľadu volajúceho SIP UA (Obrázok 4.9).

No.	Source	Destination	Protocol	Info
208	158.193.139.45	192.168.0.105	SIP/SDP	Status: 200 OK, with session description
209	192.168.0.105	158.193.139.47	STUN	Channel-Bind Request ChannelNumber=0x4cd5
210	192.168.0.105	158.193.139.47	STUN	Channel-Bind Request ChannelNumber=0x4f50
212	158.193.139.47	192.168.0.105	STUN	Channel-Bind Success Response
214	158.193.139.47	192.168.0.105	STUN	Channel-Bind Success Response
Frame 209: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) Ethernet II, Src: LiteonTe_49:c2:94 (74:de:2b:49:c2:94), Dst: Cisco-Li_00:40:0b Internet Protocol Version 4, Src: 192.168.0.105 (192.168.0.105), Dst: 158.193.139.47 User Datagram Protocol, Src Port: 50958 (50958), Dst Port: stun (3478) Session Traversal Utilities for NAT [Response In: 212] Message Type: 0x0009 (Channel-Bind Request) Message Length: 68 Message Cookie: 2112a442 Message Transaction ID: ee8ac3feb603a0959e35eeb9 Attributes SOFTWARE CHANNEL-NUMBER: 0x4cd5 XOR-PEER-ADDRESS: 158.193.139.47:34442				

Obrázok 4.9 - Rezervovanie kanála

Volajúci SIP UA ako je naznačené na Obrázku 4.9 si správou číslo 209 pre komunikáciu s transportnou adresou 158.193.139.47:34442 volaného SIP UA úspešne rezervuje číslo kanála 0x4CD5 (v nasledujúcich správach prebehne aj rezervovanie čísla kanála 0x4F50 pre transportnú adresu 158.193.139.47:34443).

Po procese rezervovania kanálov sú multimediálne dáta medzi SIP UA vymieňané prostredníctvom správy *Dáta kanála (Channel-Data TURN message)* ako je ilustrované na Obrázku 4.10.

No.	Source	Destination	Protocol	Info
243	192.168.0.105	158.193.139.47	STUN	ChannelData TURN Message
244	158.193.139.47	192.168.0.105	STUN	ChannelData TURN Message
245	192.168.0.105	158.193.139.47	STUN	ChannelData TURN Message
246	192.168.0.105	158.193.139.47	STUN	ChannelData TURN Message
247	158.193.139.47	192.168.0.105	STUN	ChannelData TURN Message
Frame 243: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)				
Ethernet II, Src: LiteonTe_49:c2:94 (74:de:2b:49:c2:94), Dst: Cisco-Li_00:40:0b (00:1e:e5:00:40:0b)				
Internet Protocol Version 4, Src: 192.168.0.105 (192.168.0.105), Dst: 158.193.139.47 (158.193.139.47)				
User Datagram Protocol, Src Port: 50958 (50958), Dst Port: stun (3478)				
Session Traversal Utilities for NAT, TURN ChannelData Message				
Channel Number: 0x4cd5				
Message Length: 92				
Data (92 bytes)				

Obrázok 4.10 - komunikácia prostredníctvom správ Dáta kanála

Kompletnú, vyššie popísanú, komunikáciu je možné nájsť aj na priloženom CD k práci (CD/Turn/).

4.3.4 ICE

V čase testovania riešení SIP prechodu cez NAT nebola nájdená žiadna implementácia SIP UA, ktorá by podporovala ICE mechanizmus ako je popísaný v štandarde (10). SIP softvérový telefón X-lite 4 má však aspoň čiastočnú podporu ICE testovania kandidátov. V nižšie uvedenom príklade chcú spolu komunikovať dvaja SIP UA, ktorí sú v Postavení č.5 z Tabuľky 4.1. Obaja SIP UA majú zapnutú podporu ICE protokolu.

Volajúci SIP UA pred začiatkom zostavenia spojenia určí vlastné mapované transportné adresy prostredníctvom STUN protokolu, ako bolo popísané v časti 4.3.2. Na rozdiel od využitia len samotného STUN protokolu sú v SDP tele SIP *Invite* správy uvedení popri mapovanej transportnej adrese, zapísanej v SDP poliach „c“ a „m“, aj lokálny **ICE kandidáti**. SIP *Invite* správa (začiatok SDP modelu ponuka/odpoveď) je uvedená na Obrázku 4.11.

No.	Source	Destination	Protocol	Info
357	192.168.0.3	158.193.139.47	STUN	Binding Request
358	192.168.0.3	158.193.139.47	STUN	Binding Request
359	158.193.139.47	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 158.193.139.197:52856
360	158.193.139.47	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 158.193.139.197:52857
361	192.168.0.3	158.193.139.45	SIP/SDP	Request: INVITE sip:user2@158.193.139.45;transport=udp, with session
Session Initiation Protocol				
Request-Line: INVITE sip:user2@158.193.139.45;transport=udp SIP/2.0				
Message Header				
Message Body				
Session Description Protocol				
Session Description Protocol version (v): 0				
Owner/Creator, Session Id (o): - 12977316740732389 1 IN IP4 192.168.0.3				
Session Name (s): CounterPath X-Lite 4.1				
Connection Information (c): IN IP4 158.193.139.197				
Time Description, active time (t): 0 0				
Session Attribute (a): ice-ufrag:71035c				
Session Attribute (a): ice-pwd:0e1c183595ee520167b1144efdf6a69c				
Media Description, name and address (m): audio 52856 RTP/AVP 0 8 102 3 101				
Media Attribute (a): rtpmap:102 L16/16000				
Media Attribute (a): rtpmap:101 telephone-event/8000				
Media Attribute (a): fmp:101 0-15				
Media Attribute (a): sendrecv				
Media Attribute (a): candidate:1 1 UDP 659136 192.168.0.3 52856 typ host				
Media Attribute (a): candidate:2 1 UDP 659084 158.193.139.197 52856 typ srflx raddr 192.168.0.3 rport 52856				
Media Attribute (a): candidate:1 2 UDP 659134 192.168.0.3 52857 typ host				
Media Attribute (a): candidate:2 2 UDP 659082 158.193.139.197 52857 typ srflx raddr 192.168.0.3 rport 52857				

Obrázok 4.11 – Komunikácia volajúceho SIP UA

Na obrázku môžeme vidieť, že v SDP poliach „c“ a „m“ je uvedená transportná adresa 158.193.139.197:52856. Táto adresa reprezentuje **predvoleného ICE kandidáta**, ktorý má byť použitý v prípade ak volaný SIP UA nerozumie ICE protokolu (ďalšie testy X-lite 4 ukázali, že je táto adresa použitá aj v prípade, ak je testovanie ICE kandidátov neúspešné). V SDP poliach atribútu média „a“ sú zapísaní štyria ICE kandidáti. Prvý a tretí v poradí sú **Host'ovskí ICE kandidáti (typ host)** pre RTP a RTCP. Druhý a štvrtý kandidát sú **Server-Reflexívni ICE kandidáti (typ srflx)** pre RTP a RTCP.

Volaný SIP UA po doručení SIP *Invite* správy takisto určí vlastné mapované transportné adresy pomocou STUN protokolu. V SDP tele SIP *200 OK* správy odosiela následne zoznam svojich ICE kandidátov k volajúcemu SIP UA. Konkrétna SIP *200 OK* správa je uvedená v prílohe č.2.

Po doručení SIP *200 OK* správy k volajúcemu SIP UA majú obaja SIP UA k dispozícii zoznam lokálnych a vzdialených ICE kandidátov, z ktorých sú vytvorené ICE páry. V procese testovania párov by mali byť podľa špecifikácie otestované všetky kombinácie lokálnych a vzdialených ICE kandidátov pre rovnaký komponent média. Pri testovaní implementácie ICE mechanizmu v X-lite 4 však bolo odpozorované, že aplikácia vykonáva len testovanie pre Host'ovský typ kandidátov. Sledovaný úsek testovania ICE párov z pohľadu volajúceho SIP UA (IP adresa: 192.168.0.3) je znázornený na Obrázku 4.12.

No.	Source	Destination	Protocol	Info
505	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
508	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56574
509	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
510	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56575
512	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
513	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52856
514	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
516	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52857

+	Frame 505: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)
+	Ethernet II, Src: D-LinkIn_70:ed:6e (1c:af:f7:70:ed:6e), Dst: D-LinkIn_70:ed:31 (1c:af:f7:70:ed:31)
+	Internet Protocol Version 4, Src: 192.168.0.4 (192.168.0.4), Dst: 192.168.0.3 (192.168.0.3)
+	User Datagram Protocol, Src Port: 56574 (56574), Dst Port: 52856 (52856)
+	Session Traversal Utilities for NAT
	[Response In: 508]
+	Message Type: 0x0001 (Binding Request)
	Message Length: 120
	Message Cookie: 2112a442
	Message Transaction ID: aed42c0918f62f9fd6687c0f
+	Attributes
+	USERNAME: 71035c:b53bb8
+	SOFTWARE
+	PRIORITY
+	ICE-CONTROLLED

Obrázok 4.12 - ICE testovanie

Z obrázku môžeme vyčítať, že testovanie ICE párov je zabezpečené správami protokolu STUN. Ako prvý začal testovanie párov volaný SIP UA (prostredníctvom integrovaného ICE agenta, zobrazené správy č.505-510) a poslal STUN *Požiadavku spojovania* z transportnej adresy 192.168.0.4:56574 (lokálny ICE kandidát) na transportnú adresu 192.168.0.3:52856 (vzdialený ICE kandidát). Uvedená požiadavka obsahuje atribút *ICE-CONTROLLED*, čo znamená, že volaný SIP UA je v pozícii kontrolovaného ICE agenta. Na STUN požiadavku volajúci SIP UA odpovedá *Úspešnou odpoveďou spojovania* s atribútom *XOR-MAPPED-ADDRESS*, v ktorom je uvedená zdrojová adresa STUN požiadavky. STUN odpoveď je odoslaná z transportnej adresy 192.168.0.3:52856 (lokálny ICE kandidát) na transportnú adresu 192.168.0.4:56574 (vzdialený ICE kandidát), uvedená správa je ilustrovaná v prílohe č.3. Volaný SIP UA po doručení STUN odpovede môže testovaný ICE pár označiť ako úspešný. Rovnakým spôsobom je z pohľadu volaného SIP UA testovaný aj ICE pár pre RTCP (správy č.509 a č.510). Ako vyplýva z definície ICE mechanizmu každý pár musí byť otestovaný oboma ICE agentmi. Testovanie ICE párov volajúcim SIP UA (zobrazené správy č.512-516) prebieha podobným spôsobom ako vyššie popísané testovanie. Volajúci SIP UA rovnako posla *Požiadavku spojovania* z transportnej adresy 192.168.0.3:52856 (lokálny ICE kandidát) na transportnú adresu 192.168.0.4:56574 (vzdialený ICE kandidát). V STUN požiadavke sú však zaradené dva dôležité atribúty: *ICE-CONTROLLING* a *USE-CANDIDATE*. *ICE-CONTROLLING* atribút naznačuje, že SIP UA je v pozícii kontrolného ICE agenta, čo ho zároveň oprávňuje do požiadavky zaradiť atribút *USE-CANDIDATE*, ktorý určuje, že testovaný pár, má byť použitý, v prípade úspešného testovania, na výmenu multimediálnych dát medzi dvomi SIP

UA. STUN požiadavka s uvedenými atribútmi je ilustrovaná v prílohe č.4. Rovnakým spôsobom je z pohľadu volajúceho SIP UA testovaný aj ICE pár pre RTCP (správy č.514 a č.516). Využitie atribútu *USE-CANDIDATE* v každej *Požiadavke spojovania* naznačuje, že ICE agent využíva agresívny typ nominácie páru.

Po úspešnom ukončení testov sú multimediálne dáta medzi oboma SIP UA vymieňané prostredníctvom Host'ovských ICE kandidátov, ktorí boli nominovaní (atribút *USE-CANDIDATE*) volajúcim SIP UA počas testovania. X-lite 4 na výmenu multimediálnych dát v tomto prípade nevyužíva RTP protokol (napriek tomu, že boli úspešne testovaní kandidáti pre RTP aj RTCP). Dáta sú zapuzdrené v transportnom protokole UDP a na komunikáciu sú využití len kandidáti s párnymi číslami portov. Sledovaná výmena, v tomto prípade hlasových dát, medzi SIP UA z pohľadu volajúceho je ilustrovaná na Obrázku 4.13.

No.	Source	Destination	Protocol	Info
541	192.168.0.3	192.168.0.4	UDP	Source port: 52856 Destination port: 56574
542	192.168.0.4	192.168.0.3	UDP	Source port: 56574 Destination port: 52856
543	192.168.0.3	192.168.0.4	UDP	Source port: 52856 Destination port: 56574
544	192.168.0.4	192.168.0.3	UDP	Source port: 56574 Destination port: 52856
545	192.168.0.3	192.168.0.4	UDP	Source port: 52856 Destination port: 56574
546	192.168.0.4	192.168.0.3	UDP	Source port: 56574 Destination port: 52856
Frame 541: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) Ethernet II, Src: D-LinkIn_70:ed:31 (1c:af:f7:70:ed:31), Dst: D-LinkIn_70:ed:6e (1c:af:f7:70:ed:6e) Internet Protocol Version 4, Src: 192.168.0.3 (192.168.0.3), Dst: 192.168.0.4 (192.168.0.4) User Datagram Protocol, Src Port: 52856 (52856), Dst Port: 56574 (56574) Data (172 bytes) Data: 800008150005acbc4b9e382effff7efefe7cfdfe7efefdfef... [Length: 172]				

Obrázok 4.12 - Výmena hlasových dát medzi SIP UA

Kompletnú, vyššie popísanú, komunikáciu je možné nájsť aj na priloženom CD k práci (CD/Ice/).

ICE mechanizmus ako je implementovaný v SIP softvérovom telefóne X-lite 4 je možné na úspešné založenie multimediálneho spojenia použiť v Postaveniach č.1, 3, 5 a 6 z Tabuľky 4.1. Pri zvyšných Postaveniach č.2 a 4 by bol na založenie multimediálneho spojenia využitý predvolený ICE kandidát (zapísaný v SDP poliach „c“ a „m“). Týmto kandidátom by bola mapovaná transportná adresa získaná prostredníctvom protokolu STUN, čo by pri obmedzeniach spojených so symetrickým NAT spôsobilo neúspešné zostavenie multimediálneho spojenia medzi dvoma SIP UA.

4.4 Vlastná implementácia ICE

Ako bolo spomenuté v predchádzajúcej časti, žiadna z dostupných implementácií ICE mechanizmu nepodporovala určenie a testovanie všetkých typov ICE kandidátov. Z tohto dôvodu sme sa rozhodli implementovať vlastného ICE agenta, ktorý bude podporovať zber kandidátov od STUN a TURN servera a ich následne testovanie podľa špecifikácie (10). Nami navrhnutú aplikáciu **IceClient** je možné využiť ako testovací nástroj pre rôzne typy sietí. Na základe testov dokáže aplikácia **určiť platné ICE páry**, ktoré je možné využiť na výmenu aplikačných dát. Na základe výsledkov testovania je možné predpovedať, či je nutné nasadenie STUN alebo TURN servera v konkrétnej sieti. S nasadením týchto typov serverov je samozrejme spojená potrebná podpora v rámci SIP UA pre protokoly STUN, TURN a ICE.

4.4.1 Analýza dostupných knižníc

Z hľadiska implementácie ICE agenta je dôležitá podpora správ v STUN formáte. Využitie dostupných STUN knižníc pre vytváranie a spracovanie STUN správ bolo však veľmi náročné z viacerých dôvodov. Jednalo sa prevažne o nedostatočnú dokumentáciu, vo väčšine prípadov proprietárnych riešení, ktorých analýza by bola z časového hľadiska veľmi náročná. Pri týchto STUN knižniciach navyše nebolo možné zaručiť ich úspešné nasadenie, pretože už pri ich kompilácii sa vyskytli viaceré chyby. Od autorov knižníc neboli poskytnuté kroky k odstráneniu týchto chýb a teda k úspešnému nasadeniu knižníc. Z týchto dôvodov sme sa rozhodli implementovať aj samotné vytváranie a spracovanie STUN správ.

4.4.2 Návrh implementácie

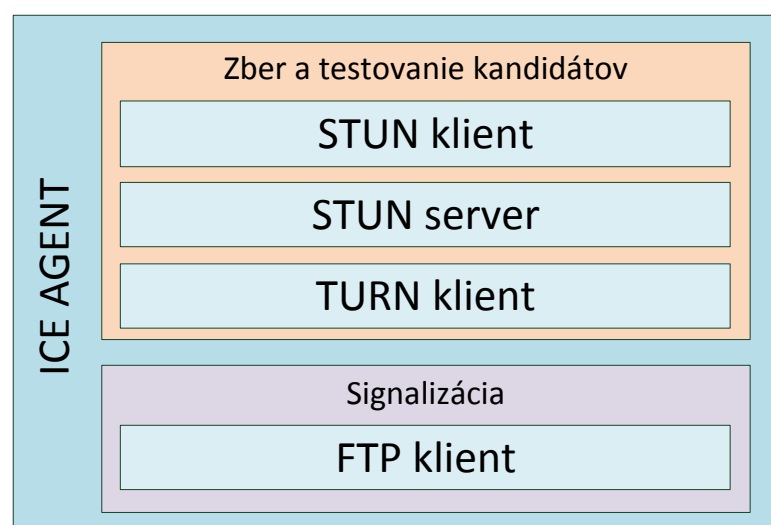
ICE agent ako je popísaný v (10) sa skladá z viacerých komponentov. Pre zber a testovanie ICE kandidátov sú to nasledujúce komponenty:

- STUN klient.
- STUN server.
- TURN klient.

STUN server komponent nie je však nutné implementovať v plnom rozsahu špecifikácie (7), pretože jeho využitie v rámci ICE agenta sa zužuje len na odosielanie správy *Úspešná odpoveď spojovania* na správu *Požiadavka spojovania* STUN klienta. STUN klient okrem

vytvárania požiadaviek musí vedieť spracovať STUN odpovede a atribúty, ktoré sú v nej zapísané. Pre základnú funkcionálnu ICE agenta je nutné aby STUN klient dokázal spracovať atribút *XOR-MAPPED-ADDRESS*, v ktorom je zapísaná transportná adresa STUN klienta z pohľadu STUN servera. Komponent STUN klient je potrebný pri zbere ICE kandidátov a tiež pri procese testovania ICE párov. V procese testovania je navyše potrebný aj komponent STUN server. Pre získanie TURN prenosových kandidátov je potrebná implementácia komponentu TURN klienta v rámci ICE agenta. Základom implementácie TURN klienta je rezervovanie prenosovej transportnej adresy a vytvorenie príslušných povolení na spracovanie dátového toku od iných koncových bodov spojenia, pre túto prenosovú transportnú adresu. Ďalšou dôležitou implementačnou súčasťou TURN klienta je posielanie aplikačných dát medzi TURN klientom a TURN serverom. Ako je známe z časti 3.3.2 na tento účel špecifikácia TURN protokolu definuje dve metódy: metódu *Odoslanie* a *Dáta* a metódu rezervovania kanálov. Pre výmenu aplikačných dát medzi TURN klientom a serverom je nutné v implementácii zabezpečiť podporu pre minimálne jednu z týchto metód.

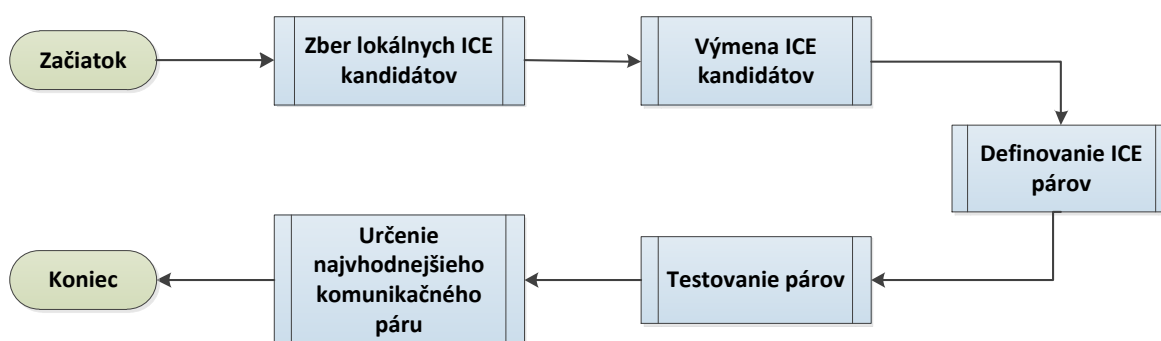
Súčasťou ICE agenta je aj komponent pre signalizáciu, ktorý slúži na výmenu ICE kandidátov medzi dvomi ICE agentmi. Signalizačným komponentom je bežne SIP klient, avšak môže nim byť ľubovoľný iný spôsob s využitím tzv. randezvous protokolu. Pri našej implementácii sme sa, z dôvodu nižšej časovej náročnosti implementácie, rozhodli na výmenu ICE kandidátov využiť FTP protokol. Signalizačným komponentom v rámci našej implementácie bude teda FTP klient. Graficky znázornené komponenty navrhovaného ICE agenta sú ilustrované na Obrázku 4.14.



Obrázok 3.14 - Komponenty ICE agenta

Tvorbu a testovanie ICE párov, ktoré boli vytvorené z lokálnych a vzdialených ICE kandidátov, sme sa rozhodli implementovať podľa špecifikácie protokolu ICE ako bola popísaná v časti 3.4.3. Čiastočným zjednodušením je implementácia iba jedného typu média, ktorý pre našu implementáciu predstavujú textové správy. V rámci testovania ICE párov je nutné zvoliť typ nominácie (agresívna alebo obyčajná nominácia), ktorý bude použitý na určenie konečného komunikačného páru.

Navrhovaný ICE proces určenia najvhodnejšieho komunikačného páru, ktorý vychádza zo špecifikácie (10), je možné popísať prostredníctvom procesného diagramu, ktorý je ilustrovaný na Obrázku 4.15.



Obrázok 4.14 - ICE proces

Na obrázku môžeme vidieť päť hlavných podprocesov:

- zber lokálnych ICE kandidátov.
- Výmena ICE kandidátov.
- Definovanie ICE párov.
- Testovanie párov.
- Určenie najvhodnejšieho komunikačného páru.

Ilustráciu jednotlivých podprocesov je možné nájsť v prílohe č. 5.

4.4.3 Implementácia riešenia

Implementácia riešenia vlastného ICE agenta bola realizovaná vo vývojovom prostredí Visual Studio 2010 v programovacom jazyku C#. Ako prvé bolo potrebné definovať triedy a metódy pre prácu so STUN správami, ktoré tvoria základ pre komponenty Zberu a testovania kandidátov z Obrázku 4.14. Naša implementácia podporuje vytváranie STUN správ ľubovoľnej triedy a metódy. Dôležitú súčasť vytvárania

STUN správ tvoria atribúty správy. Pri implementácii sme sa zamerali na atribúty, ktoré boli popísané v kapitole číslo tri. Naša implementácia podporuje pridávanie atribútov, ktoré sú dôležité pri vytváraní STUN požiadaviek (komponenty STUN a TURN klient) a STUN odpovedí (komponent STUN server) so zameraním na ICE mechanizmus. Rovnako bolo nutné zabezpečiť aj spracovanie atribútov nachádzajúcich sa v STUN odpovediach (komponenty STUN a TURN klient) a STUN požiadavkách (komponent STUN server) pri komunikácii ICE agenta s inými entitami. Entitou v našom prípade môže byť iný ICE agent alebo STUN/TURN server. Zoznam podporovaných atribútov v aplikácii IceClient je možné nájsť v prílohe č.6.

Výmena ICE kandidátov, ako bolo spomenuté v predchádzajúcej časti, je realizovaná prostredníctvom FTP klienta, ktorý bol tiež implementovaný v rámci ICE agenta. Po určení kandidátov ICE agent prostredníctvom FTP klienta odošle dátový súbor s lokálnymi ICE kandidátmi na predefinovaný a obom ICE agentom dostupný FTP server. Následne z FTP servera stiahne vzdialených ICE kandidátov iného ICE agenta. Ako FTP server bola použitá aplikácia **vsftpd**, ktorá je voľne dostupná z repozitárov pre OS Debian. FTP server bol nainštalovaný a spustený na katedrovom serveri a je dosiahnuteľný na IP adrese 158.193.139.47.

V nasledujúcej časti práce uvedieme niektoré kritické body vývoja aplikácie, v ktorých bolo potrebné riešiť problémy spojené s implementáciou, prípadne na riešenie konkrétnej funkcionality bolo možné využiť viacero metód. Prvým krokom ICE agenta je získanie ICE kandidátov. Ako je známe z časti 3.4, pri zbere kandidátov dokáže ICE agent určiť tri typy ICE kandidátov: Host'ovského, Server-Reflexívneho a Prenosového. Prvým problémom je určenie lokálnej transportnej adresy, ktorá je zároveň host'ovským ICE kandidátom. IP adresu lokálneho adaptéra je možné kódovo získať od operačného systému. Problém však nastáva pri definovaní čísla portu, pomocou ktorého bude aplikácia komunikovať. Je nutné zabezpečiť aby vybrané číslo portu nebolo v rámci operačného systému obsadené inou aplikáciou. Tento problém sme vyriešili náhodným výberom čísla portu z definovateľného rozsahu (napr. 62000 - 63000) a následným testovaním. Test určí, či je možné inicializovať soket s príslušnou IP adresou lokálneho adaptéra a vybraného čísla portu. V prípade, že sa nepodarí inicializovať soket, je možné zvoliť iné číslo portu a test zopakovať. Na určenie Server-Reflexívneho a Prenosového ICE kandidáta je využitá odpoveď TURN servera *Úspešná odpoveď priradenia* (atribúty *XOR-RELAYED-ADDRESS* a *XOR-MAPPED-ADDRESS*) na TURN *Požiadavku priradenia*. V prípade

neúspechu požiadavky (Prenosový ICE kandidát nebude definovaný) alebo ak nie je v odpovedi zaradený atribút *XOR-MAPPED-ADDRESS*, je využitá *Požiadavka spojovania* a k nej príslušná odpoveď *Úspešná požiadavka spojovania* na určenie Server-Reflexívneho kandidáta.

Funkcionalitu odosielania a prijímania aplikačných dát v rámci TURN klienta je možné riešiť dvomi metódami ako bolo popísané v časti 3.3.2. Pre našu implementáciu sme vybrali metódu *Odosielanie a Dáta*, ktorá využíva STUN správy triedy *Indikácia*. Pre túto možnosť sme sa rozhodli najmä z dôvodu otestovania, či je spracovanie správ *Odosielanie a Dáta* podporované na nami nasadenom TURN serveri (Restund). Navyše nami využité médium, ktorými sú textové správy, nie je tak náročné na spracovanie TURN entitami ako napríklad hlasové dáta, pri ktorých by bolo nasadenie metódy rezervovania kanálov výhodnejšie.

Proces testovania ICE párov bol implementovaný pomocou viacerých vlákien, ktoré zabezpečovali súčasné prijímanie a odosielanie STUN správ. Na prijaté správy bol navyše vytvorený softvérový buffer, ktorý zabezpečil postupné spracovanie prijatých STUN správ. Pre testovacie zoznamy ICE párov boli vytvorené viaceré prioritné fronty, z ktorých bol vždy vybraný konkrétny pár pre testovanie na základe pravidiel definovaných v 3.4.3. Testovacie STUN požiadavky sú na základe lokálneho ICE kandidáta odosielané prostredníctvom UDP datagramu, v prípade, že sa nejedná o Prenosového lokálneho kandidáta. Odoslanie je v tomto prípade zabezpečované komponentom STUN klient. Druhou možnosťou je odoslanie STUN požiadavky zapuzdrenej v atribúte *Dáta* STUN správy *Indikácia Odoslania*, v prípade, že sa jedná o Prenosového lokálneho kandidáta. V tomto prípade je odoslanie zabezpečované komponentom TURN klient. Pri prijatí STUN požiadaviek je rovnako potrebné rozlišovať prostredníctvom ktorého ICE kandidáta bola požiadavka prijatá a určiť či bude STUN odpoveď odoslaná prostredníctvom STUN alebo TURN klienta.

Nominácia ICE páru, ktorý bude využitý na výmenu aplikačných dát medzi dvomi ICE agentmi môže byť implementovaná dvomi spôsobmi: agresívnou alebo obyčajnou nomináciou. Pre našu implementáciu sme zvolili metódu obyčajnej nominácie. Komunikačný pár je možné určiť po ukončení procesu testovania ľubovoľne, pričom páry sú navrhnuté utriedené zostupne podľa priorít. Dôvodom pre implementovanie metódy obyčajnej nominácie bolo, aby aplikácia určila všetky platné komunikačné páry. Pri

agresívnej metóde nominovania ICE párov by navyše mohol byť vybraný komunikačný pár, ktorého využitie je na základe priority párov menej výhodné.

Výsledkom implementácie je aplikácia IceClient, ktorá dokáže určiť najvhodnejší pár transportných adries pre komunikáciu medzi dvomi klientmi a následné použitie tohto komunikačného páru na výmenu textových správ. V aplikácii je možné samostatne využiť aj jednotlivých klientov. STUN klienta na určenie mapovanej transportnej adresy a TURN klienta na pridelenie prenosovej transportnej adresy a výmenu správ s iným koncovým bodom prostredníctvom správ *Indikácia Odoslania* a *Dáta*.

4.4.4 Testovanie implementácie

V rámci testovania našej implementácie ICE agenta sme sa zamerali na postavenia dvoch, v tomto prípade ICE agentov, z Tabuľky 4.1. Praktický príklad behu a vyhodnotenia výsledkov našej aplikácie uvedieme na Postavení č.3 z Tabuľky 4.1. Čiastočná analýza zvyšných postavení z uvedenej tabuľky bude uvedené na konci tejto časti práce.

Po spustení aplikácie IceClient sú používateľovi zobrazené základné nastavenia. Úvodnú obrazovku aplikácie IceClient je možné nájsť v prílohe č.7. Nastavenia súvisia s lokálnym nastavením čísla portu (*Local Settings*), ktoré je náhodne vygenerované a nastavením pripojenia na FTP server (*FTP Settings*). Pre pripojenie na FTP server je potrebné zadať jeho IP adresu alebo doménové meno v poli *Address* a prihlasovacie údaje v poliach *Username* a *Password*. Dôležitým nastavením je nastavenie ICE agenta (*ICE Settings*), v ktorom je potrebné zvoliť jedného z dvoch užívateľov (*User 1* alebo *User 2*). Pre správnu funkciu aplikácie je potrebné zvoliť rozdielnych užívateľov pri dvoch spustených aplikáciách na rôznych PC.

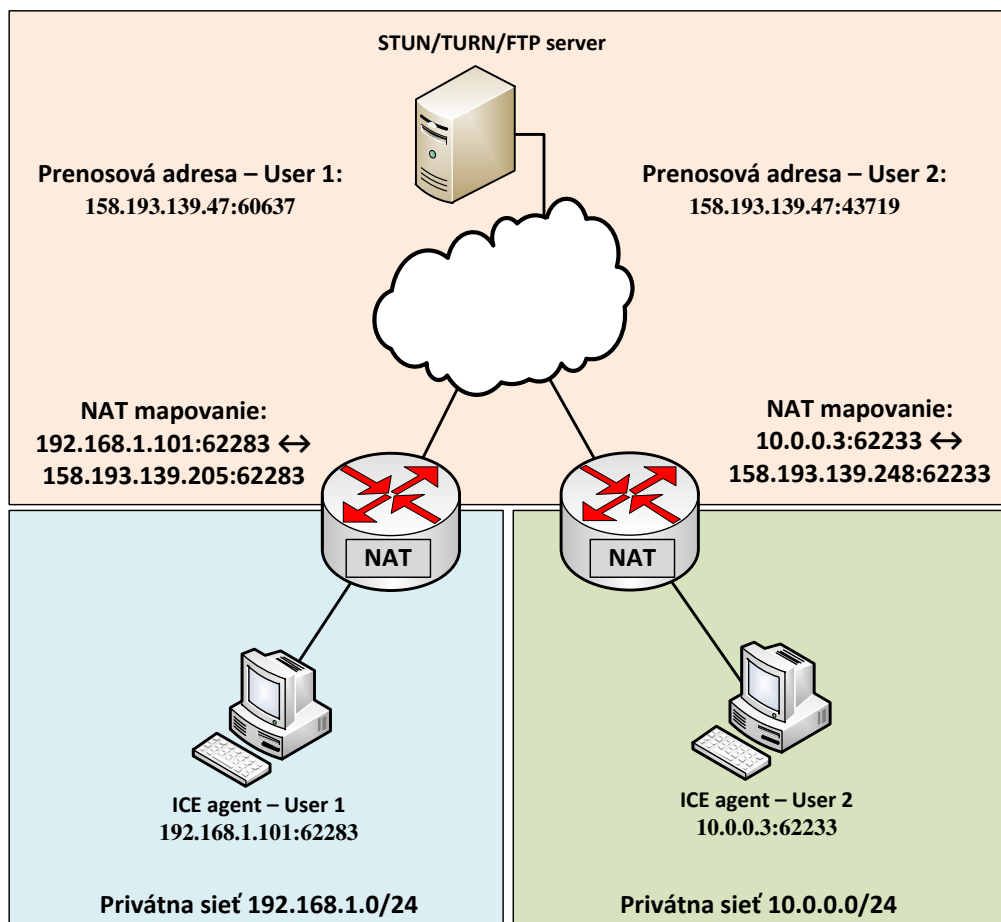
Integrovaného ICE agenta je možné spustiť v záložke *ICE*. Záložka *ICE* je ilustrovaná v prílohe č.8. Pomocou tlačidla *Gather ICE candidates* dokáže aplikácia určiť ICE kandidátov. Na určenie ICE kandidátov je potrebný STUN/TURN server, ktorého transportnú adresu je možné zadať prostredníctvom polí *IP* a *Port* v časti *STUN/TURN server*. Konkrétnym príkladom zberu kandidátov oboch ICE agentov pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: TYP PRIORITA IP:PORT):

```
Local ICE candidates [User 1]:  
Host 2130706431 192.168.1.101:62283
```

```
Server-Reflexive 1694498815 158.193.139.205:62283
Relayed 16777215 158.193.139.47:60637
```

```
Local ICE candidates [User 2]:
Host 2130706431 10.0.0.3:62233
Server-Reflexive 1694498815 158.193.139.248:62233
Relayed 16777215 158.193.139.47:43719
```

Postavenie jednotlivých zariadení v sieti je ilustrované na Obrázku 4.16.



Obrázok 4.16 – Príklad postavenia zariadení

Ďalším krokom po zbere ICE kandidátov je ich výmena medzi dvomi ICE agentmi. V rámci aplikácie je výmena realizovaná prostredníctvom tlačidla *Save local candidates*, pre uloženie lokálnych ICE kandidátov na FTP server a tlačidla *Load remote candidates*, pre načítanie ICE kandidátov vzdialeného ICE agenta z FTP servera. Konkrétnym príkladom výmeny kandidátov z pohľadu ICE agenta *User 1* pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: TYP PRIORITA IP:PORT):

```
Saved ICE candidates:
Host 2130706431 192.168.1.101:62283
Server-Reflexive 1694498815 158.193.139.205:62283
Relayed 16777215 158.193.139.47:60637
```

```
Loaded ICE candidates:
Host 2130706431 10.0.0.3:62233
Server-Reflexive 1694498815 158.193.139.248:62233
Relayed 16777215 158.193.139.47:43719
```

Po výmene ICE kandidátov musí každý ICE agent vytvoriť jednotlivé ICE páry, ktoré sú vytvárané z lokálnych a vzdialených ICE kandidátov systémom „každý s každým“. Na vytvorenie konkrétnych ICE párov slúži v aplikácii IceClient tlačidlo *Create ICE pairs* (vytvorenie ICE párov je pri testovaní potrebné vykonať na oboch spustených aplikáciách IceClient). Konkrétnym príkladom vytvorených ICE párov z pohľadu ICE agenta *User 1* pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: STAV PRIORITA lokálna IP:PORT vzdialená IP:PORT):

```
Actual ICE pairs:
FROZEN 9151314442783293438 Local:192.168.1.101:62283
                                Remote: 10.0.0.3:62233
FROZEN 7277816997797167103 Local:192.168.1.101:62283
                                Remote: 158.193.139.248:62233
FROZEN 7277816997797167102 Local:158.193.139.205:62283
                                Remote: 10.0.0.3:62233
FROZEN 7277816996924751870 Local:158.193.139.205:62283
                                Remote: 158.193.139.248:62233
FROZEN 72057594004373503 Local:192.168.1.101:62283
                                Remote: 158.193.139.47:43719
FROZEN 72057594004373502 Local:158.193.139.47:60637
                                Remote: 10.0.0.3:62233
FROZEN 72057593131958271 Local:158.193.139.205:62283
                                Remote: 158.193.139.47:43719
FROZEN 72057593131958270 Local:158.193.139.47:60637
                                Remote: 158.193.139.248:62233
FROZEN 72057589776515070 Local:158.193.139.47:60637
                                Remote: 158.193.139.47:43719
```

V uvedenom výpise je deväť ICE párov zostupne usporiadaných podľa priority. Môžeme si všimnúť, že najvyššiu prioritu má pár, ktorý je kombináciou Host'ovských ICE kandidátov. Naopak najnižšia priorita bola určená páru, ktorý je kombináciou dvoch Prenosových kandidátov, čo znamená, že využitie TURN servera je pri ICE mechanizme pokladané za najmenej výhodnú formu výmeny aplikačných dát. Pred samotným testovaním ICE párov je potrebné na základe pravidiel definovaných v 3.4.3 odstrániť niektoré ICE páry.

Základom pre zníženie počtu párov je nahradenie lokálnych Server-Reflexívnych ICE kandidátov ich základňami. Tento krok má za následok výskyt redundantných položiek v zozname ICE párov. V prípade, že sa nejaký pár vyskytuje v zozname viackrát, je v zozname ponechaný len pár s najvyššou prioritou. Páry, ktoré ostanú v zozname po odstránení redundantných párov, sú určené na testovanie. Konkrétnym príkladom vybraných ICE párov, ktoré budú testované z pohľadu ICE agenta *User 1* pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: STAV PRIORITA lokálna IP:PORT vzdialená IP:PORT):

```
3 pairs pruned!!!
```

```
Pairs selected for test:
```

```
FROZEN 9151314442783293438 Local:192.168.1.101:62283
                                Remote: 10.0.0.3:62233
FROZEN 7277816997797167103 Local:192.168.1.101:62283
                                Remote: 158.193.139.248:62233
FROZEN 72057594004373503   Local:192.168.1.101:62283
                                Remote: 158.193.139.47:43719
FROZEN 72057594004373502   Local:158.193.139.47:60637
                                Remote: 10.0.0.3:62233
FROZEN 72057593131958270   Local:158.193.139.47:60637
                                Remote: 158.193.139.248:62233
FROZEN 72057589776515070   Local:158.193.139.47:60637
                                Remote: 158.193.139.47:43719
```

Počet ICE párov, ktoré boli vybrané na testovanie je znížený z pôvodných deväť na šesť, čo naznačuje aj hlásenie aplikácie o odstránení troch párov (*3 pairs pruned*).

Samotné testovanie ICE párov prebieha po stlačení tlačidla *Test pairs*. Tlačidlo spustí testovanie ICE párov na lokálnom ICE agentovi a zároveň, pri prvom doručení testovacej STUN požiadavky, je testovanie automaticky spustené aj na vzdialenom ICE agentovi. Celkové testovanie ICE párov sa skladá z viacerých čiastkových testov. Je dôležité pripomenúť, že aj v prípade, ak jeden ICE agent ukončil testovanie všetkých párov môže byť testovanie opäť spustené. Táto situácia nastane pri doručení STUN požiadavky, ktorá spôsobí, že niektorý ICE pár je potrebné znovu otestovať. V aplikácii IceClient je aktuálne spustenie testovania naznačené výpisom *Test is running...*. Ukončenie testovania je naznačené výpisom *Test ended!*.

Ukončenie testovania je možné zaručiť po ukončení zobrazovania vyššie uvedených výpisov. Výsledok testovania, ktorý predstavuje zoznam platných párov, je

možné v aplikácii zobrazíť stlačením tlačidla *Show valid pairs*. Konkrétnym príkladom platných ICE párov oboch ICE agentov pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: lokálny kandidát IP:PORT vzdialený kandidát IP:PORT):

```
Valid pairs [4] [User 1]:  
Local:158.193.139.205:62283 Remote:158.193.139.248:62233  
Local:158.193.139.205:62283 Remote:158.193.139.47:43719  
Local:158.193.139.47:60637 Remote:158.193.139.248:62233  
Local:158.193.139.47:60637 Remote:158.193.139.47:43719
```

```
Valid pairs [4] [User 2]:  
Local:158.193.139.248:62233 Remote:158.193.139.205:62283  
Local:158.193.139.248:62233 Remote:158.193.139.47:60637  
Local:158.193.139.47:43719 Remote:158.193.139.205:62283  
Local:158.193.139.47:43719 Remote:158.193.139.47:60637
```

Z uvedených výpisov vyplýva, že obaja ICE agenti majú v zozname platných párov rovnaké ICE páry (rozdielom je len zámena lokálnych a vzdialených ICE kandidátov). Uvedené páry sú v zozname utriedené zostupne na základe priorít.

Ako bolo spomenuté v predchádzajúcej časti, pre našu implementáciu ICE agenta bol využitý obyčajný typ nominácie komunikačného páru transportných adries. Možnosť nominovať komunikačný pár má kontrolný ICE agent, ktorým je v tomto prípade *User 1*. Konkrétny pár je v aplikácii IceClient možné zvolíť výberom páru z rozbaľovacieho zoznamu *Nominate ICE Pair* a následným stlačením tlačidla *Nominate pair*. Platné páry sú v rozbaľovacom zozname usporiadané zostupne podľa priorít. Po nominovaní páru sa u oboch ICE agentov objaví hlásenie o nominovanom páre, ktorý bude využitý na výmenu textových správ. Konkrétnym príkladom nominovania komunikačného páru u oboch ICE agentov pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient (formát zápisu: lokálny kandidát IP:PORT vzdialený kandidát IP:PORT):

```
[User 1]  
Nominated pair:  
Local:158.193.139.205:62283 Remote: 158.193.139.248:62233  
Try CHAT!!!
```

```
[User 2]  
Nominated pair:  
Local:158.193.139.248:62233 Remote: 158.193.139.205:62283  
Try CHAT!!!
```

Z uvedeného výpisu môžeme vyčítať, že bol zvolený pár s najvyššou prioritou, ktorý na oboch stranách tvorili Server-Reflexívni ICE kandidáti. Na testovanie funkčnosti výmeny aplikačných dát je v aplikácii IceClient navrhnutý jednoduchý klient pre výmenu textových správ, ktorého je možné nájsť v záložke *Chat*. Konkrétnym príkladom výmeny textových správ pre Postavenie č.3 je nasledovný výpis z aplikácie IceClient:

```
[User 1]
Message From[158.193.139.248:62233]: test
Message From[local]: odpoved na test

[User 2]
Message From[local]: test
Message From[158.193.139.205:62283]: odpoved na test
```

Kompletný výpis z aplikácie IceClient spolu so sledovanou komunikáciou v programe Wireshark oboch používateľov pre Postavenie č.3 je možné nájsť na priloženom CD nosiči k práci (CD/IceClient/Postavenia/Postavenie3/).

Na CD nosiči k práci je možné nájsť aj výpisy z aplikácie IceClient a sledovanú komunikáciu v programe Wireshark pre zvyšné postavenia ICE agentov z Tabuľky 4.1 (CD/IceClient/Postavenia/). V nasledujúcej časti budú zo zvyšných postavení uvedené len niektoré zaujímavé situácie, ktoré boli pozorované pri testovaní. Na úvod musíme podotknúť, že použité NAT zariadenie pre symetrické NAT sa nesprávalo úplne podľa špecifikácie, čo malo v niektorých prípadoch za následok odhalenie väčšieho počtu platných párov. Nami nasadené symetrické NAT vytvorilo nové mapovanie len v prípade zmeny IP adresy cieľového koncového bodu (ak IP adresa ostala zachovaná a došlo len k zmene čísla portu, NAT mapovanie ostalo nezmenené). Konkrétny príklad bude uvedený v ďalšej časti práce.

Ako bolo spomenuté v predchádzajúcich častiach práce, počas testovania ICE párov môže dôjsť k odhaleniu nového typu kandidáta tzv. Partner-Reflexívneho kandidáta. Ako príklad uvidíme Postavenie č.2 z Tabuľky 4.1 a výpis aplikácie IceClient, ktorý obsahuje zoznam platných párov a zoznam lokálnych ICE kandidátov používateľa *User 2*. Používateľ *User 2* sa v tomto konkrétnom príklade nachádza za symetrickým NAT a používateľ *User 1* má verejnú IP adresu. Výpis z aplikácie IceClient je nasledovný:

Valid pairs [4]:

```
Local:85.248.45.84:59581 Remote:158.193.139.223:62284
Local:85.248.45.84:59571 Remote:158.193.139.47:32862
Local:158.193.139.47:44674 Remote:158.193.139.223:62284
Local:158.193.139.47:44674 Remote:158.193.139.47:32862
```

Local ICE candidates:

```
Host 2130706431 192.168.0.101:62893
Server-Reflexive 1694498815 85.248.45.84:59571
Relayed 16777215 158.193.139.47:44674
Peer-Reflexive 1862270975 85.248.45.84:59581
```

Z uvedeného výpisu môžeme vyčítať, že na NAT zariadení používateľa *User 2* bolo pre komunikáciu s verejnou IP adresou používateľa *User 1* vytvorené nové mapovanie, čo malo za následok odhalenie nového Partner-Reflexívneho kandidáta. Partner-Reflexívny kandidát používateľa *User 2* zároveň tvorí platný pár s verejnou transportnou adresou používateľa *User 1*.

Na záver uvedieme príklad Postavenia č.6 z Tabuľky 4.1. V tomto prípade sú obaja ICE agenti v jednej privátnej sieti, ktorú od verejného internetu oddeľuje NAT zariadenie so symetrickým NAT. Ako už bolo spomenuté správanie nami použitého NAT zariadenia nie je úplne totožné so špecifikáciou symetrického NAT. Výsledkom testovania ICE párov bol nasledujúci zoznam platných párov z pohľadu používateľa *User 1*:

Valid pairs [4]:

```
Local:192.168.0.104:62920 Remote: 192.168.0.101:62612
Local:85.248.45.84:63745 Remote: 158.193.139.47:36540
Local:158.193.139.47:33334 Remote: 85.248.45.84:61389
Local:158.193.139.47:33334 Remote: 158.193.139.47:36540
```

Server-Reflexívny ICE kandidát, ktorý bol získaný prostredníctvom STUN servera s transportnou adresou 158.193.139.47:3478 je nasledovný:

Local ICE candidates [User 1]:

```
Server-Reflexive 1694498815 85.248.45.84:63745
```

Ako môžeme vidieť lokálna transportná adresa zvýrazneného páru v zozname platných párov je totožná s transportnou adresou Server-Reflexívneho ICE kandidáta. Je teda možné prehlásiť, že mapovanie získané prostredníctvom STUN servera je použité aj v prípade komunikácie s TURN prenosovou transportnou adresou používateľa *User 2*. V prípade implementácie symetrického NAT podľa špecifikácie by však pre komunikáciu s touto adresou bolo vytvorené nové mapovanie. Nové mapovanie by ale nebolo použiteľné,

pretože TURN server by nespracoval dáta poslané z transportnej adresy nového mapovania. Dôvodom by bolo neexistujúce povolenie na TURN serveri pre spracovanie dát z novej mapovanej transportnej adresy. Rovnaká situácia by nastala z pohľadu používateľa *User 2* v prípade páru, ktorý je v zozname platných párov na treťom mieste. Dôsledkom implementácie symetrického NAT by teda bol neúspech ICE testov pri pároch, ktoré sú tvorené kombináciou Server-Reflexívneho a Prenosového ICE kandidáta. Výsledkom testovania by bol skrátený zoznam platných párov, ktorý by vyzeral nasledovne:

```
Valid pairs [2]:  
Local:192.168.0.104:62920 Remote: 192.168.0.101:62612  
Local:158.193.139.47:33334 Remote: 158.193.139.47:36540
```

Z uvedeného zoznamu párov vyplýva, že na komunikáciu by bolo možné použiť len páry ktoré sú tvorené dvomi Host'ovskými ICE kandidátmi alebo dvomi Prenosovými ICE kandidátmi.

Testovanú implementáciu IceClient je možné nájsť spolu so zdrojovým kódom na priloženom CD nosiči k práci (CD/IceClient/Aplikacia/).

5 ZÁVER

V práci boli popísané viaceré podporné mechanizmy slúžiace na prechod SIP cez zariadenia s mechanizmom NAT. Pri všetkých popísaných mechanizmoch sa jednalo o relatívne nové štandardizované protokoly. Je dôležité pripomenúť, že protokoly STUN a TURN boli navrhnuté ako podporené mechanizmy pre protokol ICE. Ako bolo spomenuté, ICE mechanizmus predstavuje komplexné riešenie úspešnej výmeny multimediálnych dát aj cez zariadenia s mechanizmom NAT, v rámci ľubovoľnej SIP architektúry. Protokoly STUN a TURN je samozrejme možné využiť aj samostatne avšak s určitými obmedzeniami. Pomocou samotného protokolu STUN nedokážeme zaručiť úspešné zostavenie multimediálneho spojenia pre všetky typy sietí a pri využití samostatného protokolu TURN je nutné splniť vysoké nároky na prevádzku TURN servera.

V rámci analýzy súčasnej implementácie jednotlivých mechanizmov STUN, TURN a ICE v dostupných SIP aplikáciách, sme narazili na ich slabú podporu. SIP UA aplikácie najčastejšie podporovali len získanie verejnej transportnej adresy prostredníctvom protokolu STUN. Na tento účel je možné nájsť aj viacero voľne dostupných implementácií STUN servera, prípadne využiť niektorý z verejných STUN serverov. Najväčším problémom bola podpora protokolu TURN. V rámci implementácií bola nájdená len jediná aplikácia podporujúca funkcionality entity TURN klienta a rovnako jediná aplikácia podporujúca funkcionality TURN servera. Pri testovaní mechanizmu ICE v rámci SIP UA bola odhalená len čiastočná podpora zberu a testovania ICE kandidátov. Podpora ICE mechanizmu sa v rámci dostupných aplikácií zúžila len na testovanie kandidátov v rámci lokálnej siete a úplne upustila od využitia TURN protokolu pri zbere ICE kandidátov.

Jedným z cieľov práce bolo navrhnúť a implementovať klienta s mechanizmom ICE. Na splnenie tohto cieľa bola navrhnutá a implementovaná aplikácia IceClient. Nami navrhnutú aplikáciu IceClient je možné využiť ako testovací nástroj pre rôzne typy sietí. Na základe testov dokáže aplikácia určiť platné ICE páry, ktoré je možné využiť na výmenu aplikačných dát. Na základe výsledkov testovania je následne možné predpovedať, či je nutné nasadenie STUN alebo TURN servera v konkrétnej sieti. S nasadením týchto typov serverov je samozrejme spojená potrebná podpora v rámci

implementácií SIP UA pre protokoly STUN, TURN a ICE. Na záver môžeme dodať, že mechanizmus ICE, ako riešenie prechodu cez NAT, sa javí ako veľmi efektívny. Najväčšími výhodami ICE je testovanie komunikačných párov pred ich samotným použitím na založenie multimedialného toku dát a preferované použitie P2P komunikácie medzi dvomi koncovými bodmi spojenia pred využitím TURN servera na preposielanie komunikácie.

POUŽITÉ ZDROJE

1. **J., Rosenberg.** *SIP: Session Initiation Protocol*. June 2002. RFC 3261.
2. **Schulzrinne, H.** *RTP: A Transport Protocol for Real-Time Applications*. July 2003. RFC 3550.
3. **Handley, M.** *SDP: Session Description Protocol*. July 2006. RFC 4566.
4. **Rosenberg, J.** *An Offer/Answer Model with the Session Description Protocol (SDP)*. June 2002. RFC 3264.
5. **Srisuresh, P.** *Traditional IP Network Address Translator (Traditional NAT)*. January 2001. RFC 3022.
6. **Rosenberg, J.** *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*. March 2003. RFC 3489.
7. **Rosenberg, J.** *Session Traversal Utilities for NAT (STUN)*. October 2008. RFC 5389.
8. **Geoff Huston, APNIC.** *Anatomy: A Look Inside Network Address Translators*. [Online] http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html.
9. **Mahy, R.** *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)*. April 2010. RFC 5766.
10. **Rosenberg, J.** *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols*. April 2010. RFC 5245.
11. **palo73.** STUN server na debiane. [Online] <http://nil.uniza.sk/sip/nat-fw/stun-server-na-debiane>.
12. **jht2.** STUN - Public STUN servers. [Online] <http://www.voip-info.org/wiki/view/STUN>.
13. **paulus.** Installing and configuring Restund - STUN/TURN server. [Online] <http://nil.uniza.sk/sk/node/508>.
14. **CounterPath.** X-lite 4. [Online] <http://www.counterpath.com/x-lite.html>.
15. **Viagénie.** Numb STUN/TURN server. [Online] <http://numb.viagenie.ca/>.

ZOZNAM POUŽITÝCH SKRATIEK

ALG	Application Layer (Level) Gateway
B2BUA	Back-to-Back User Agent
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
LAN	Local Area Network
MIDCOM	Middlebox Communication protocol
NAT	Network address translation
OSI	Open Systems Interconnection
QoS	Quality of Service
RFC	Request for Comments Pages
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

PRÍLOHY

Príloha č.1

No.	Source	Destination	Protocol	Info
303	192.168.0.3	158.193.139.214	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827
304	158.193.139.214	192.168.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C
305	192.168.0.3	158.193.139.214	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827
306	158.193.139.214	192.168.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C
307	192.168.0.3	158.193.139.214	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827
308	158.193.139.214	192.168.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C

Príklad RTP toku dát volajúceho

No.	Source	Destination	Protocol	Info
281	10.0.0.3	158.193.139.197	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C
282	158.193.139.197	10.0.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827
283	10.0.0.3	158.193.139.197	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C
284	158.193.139.197	10.0.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827
285	10.0.0.3	158.193.139.197	RTP	PT=ITU-T G.711 PCMU, SSRC=0xD3B71C8C
286	158.193.139.197	10.0.0.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0x21771827

Príklad RTP toku dát volaného

Príloha č.2

No.	Source	Destination	Protocol	Info
249	192.168.0.4	158.193.139.45	SIP/SDP	Status: 200 OK, with session description
Session Initiation Protocol Status-Line: SIP/2.0 200 OK Message Header Message Body Session Description Protocol Session Description Protocol version (v): 0 Owner/Creator, Session Id (o): - 12977316751607072 1 IN IP4 192.168.0.4 Session Name (s): CounterPath X-Lite 4.1 Connection Information (c): IN IP4 158.193.139.197 Time Description, active time (t): 0 0 Session Attribute (a): ice-ufrag:b53bb8 Session Attribute (a): ice-pwd:2af8a872633ef0ebc71035a97a302124 Media Description, name and address (m): audio 56574 RTP/AVP 0 8 101 Media Attribute (a): rtpmap:101 telephone-event/8000 Media Attribute (a): fmtp:101 0-15 Media Attribute (a): sendrecv Media Attribute (a): candidate:1 1 UDP 659136 192.168.0.4 56574 typ host Media Attribute (a): candidate:2 1 UDP 659084 158.193.139.197 56574 typ srflx raddr 192.168.0.4 rport 56574 Media Attribute (a): candidate:1 2 UDP 659134 192.168.0.4 56575 typ host Media Attribute (a): candidate:2 2 UDP 659082 158.193.139.197 56575 typ srflx raddr 192.168.0.4 rport 56575				

Správa 200 OK odosielaná volaným SIP UA

Príloha č.3

No.	Source	Destination	Protocol	Info
505	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
508	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56574
509	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
510	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56575
512	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
513	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52856
514	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
516	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52857
Frame 508: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) Ethernet II, Src: D-LinkIn_70:ed:31 (1c:af:f7:70:ed:31), Dst: D-LinkIn_70:ed:6e (1c:af:f7:70:ed:6e) Internet Protocol Version 4, Src: 192.168.0.3 (192.168.0.3), Dst: 192.168.0.4 (192.168.0.4) User Datagram Protocol, Src Port: 52856 (52856), Dst Port: 56574 (56574) Session Traversal Utilities for NAT [Request In: 505] [Time: 0.002489000 seconds] Message Type: 0x0101 (Binding Success Response) Message Length: 92 Message Cookie: 2112a442 Message Transaction ID: aed42c0918f62f9fd6687c0f Attributes XOR-MAPPED-ADDRESS: 192.168.0.4:56574 SOFTWARE MESSAGE-INTEGRITY FINGERPRINT				

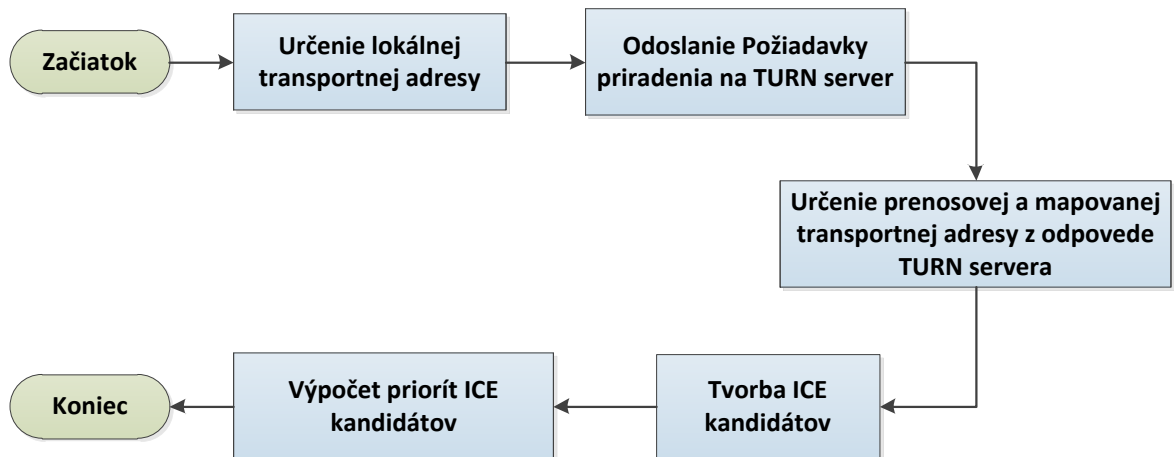
STUN odpoveď ICE testovania

Príloha č.4

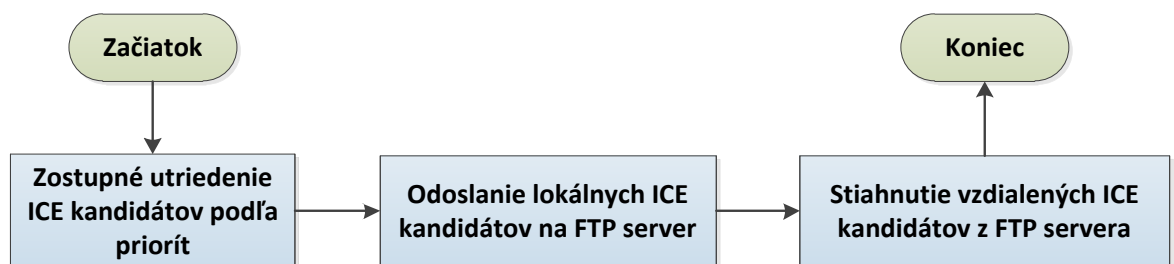
No.	Source	Destination	Protocol	Info
505	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
508	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56574
509	192.168.0.4	192.168.0.3	STUN	Binding Request user: 71035c:b53bb8
510	192.168.0.3	192.168.0.4	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.4:56575
512	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
513	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52856
514	192.168.0.3	192.168.0.4	STUN	Binding Request user: b53bb8:71035c
516	192.168.0.4	192.168.0.3	STUN	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.0.3:52857
Frame 512: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) Ethernet II, Src: D-LinkIn_70:ed:31 (1c:af:f7:70:ed:31), Dst: D-LinkIn_70:ed:6e (1c:af:f7:70:ed:6e) Internet Protocol Version 4, Src: 192.168.0.3 (192.168.0.3), Dst: 192.168.0.4 (192.168.0.4) User Datagram Protocol, Src Port: 52856 (52856), Dst Port: 56574 (56574) Session Traversal Utilities for NAT [Response In: 513] Message Type: 0x0001 (Binding Request) Message Length: 124 Message Cookie: 2112a442 Message Transaction ID: 31590c23b7ed60c72fc22de2 Attributes USERNAME: b53bb8:71035c SOFTWARE PRIORITY USE-CANDIDATE ICE-CONTROLLING MESSAGE-INTEGRITY FINGERPRINT				

Testovanie ICE páru spojené s nomináciou

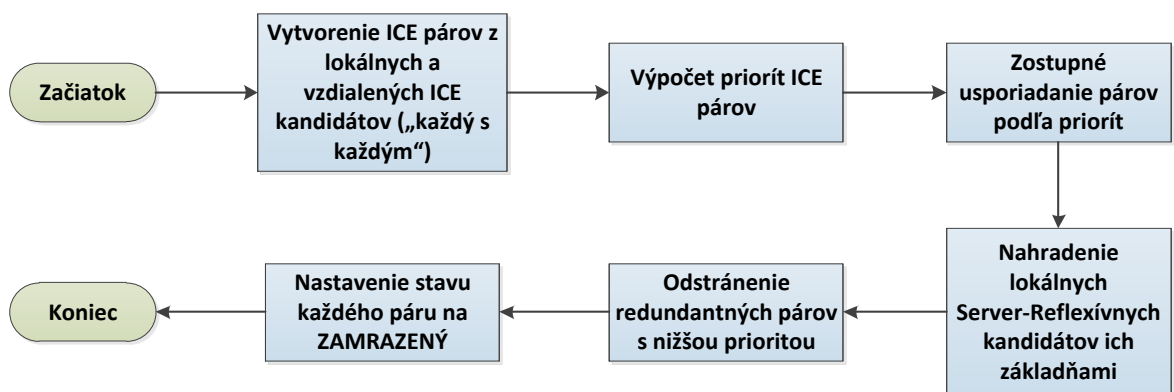
Príloha č.5



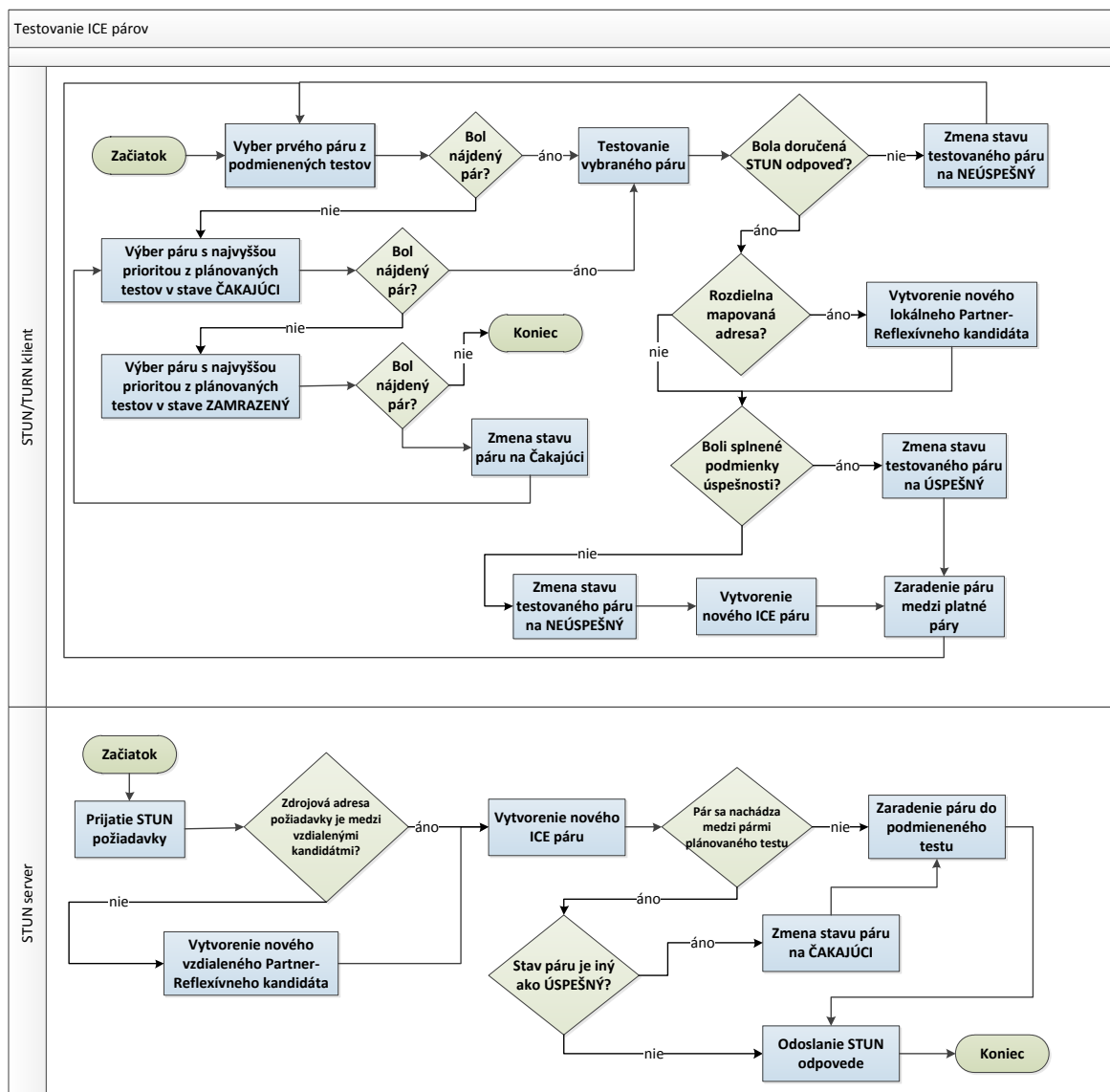
Podproces - Zber lokálnych ICE kandidátov



Podproces - Výmena ICE kandidátov

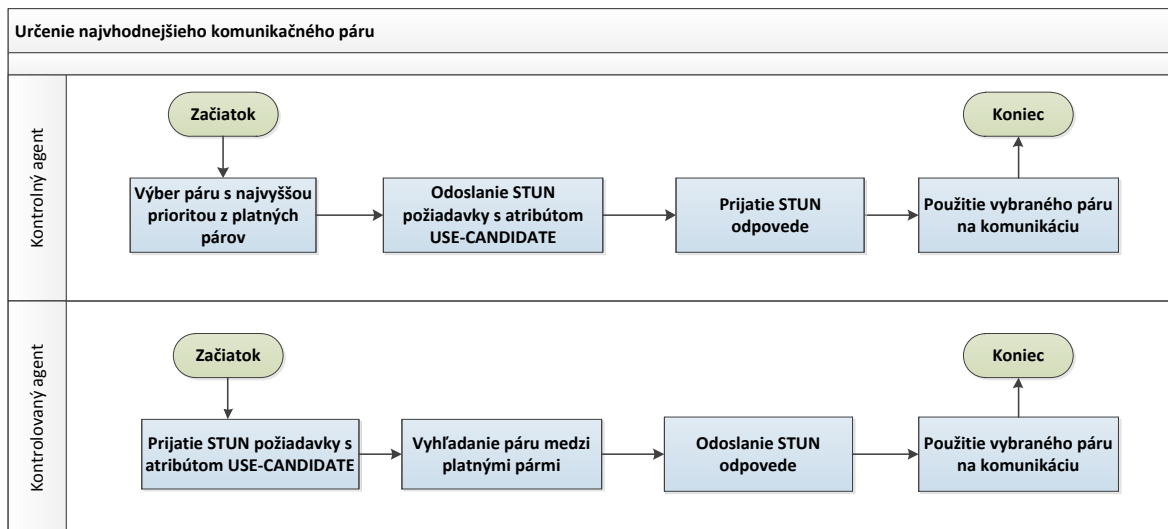


Podproces – Definovanie ICE párov



Podproces – Testovanie ICE párov

V podprocese testovania ICE párov sa nachádza proces **Vytvorenie nového páru**. Tento proces vytvorí nový pomocný ICE pár, ktorý je v prípade nájdenia totožného páru, v ľubovoľnom zozname, nahradený práve nájdeným párom.



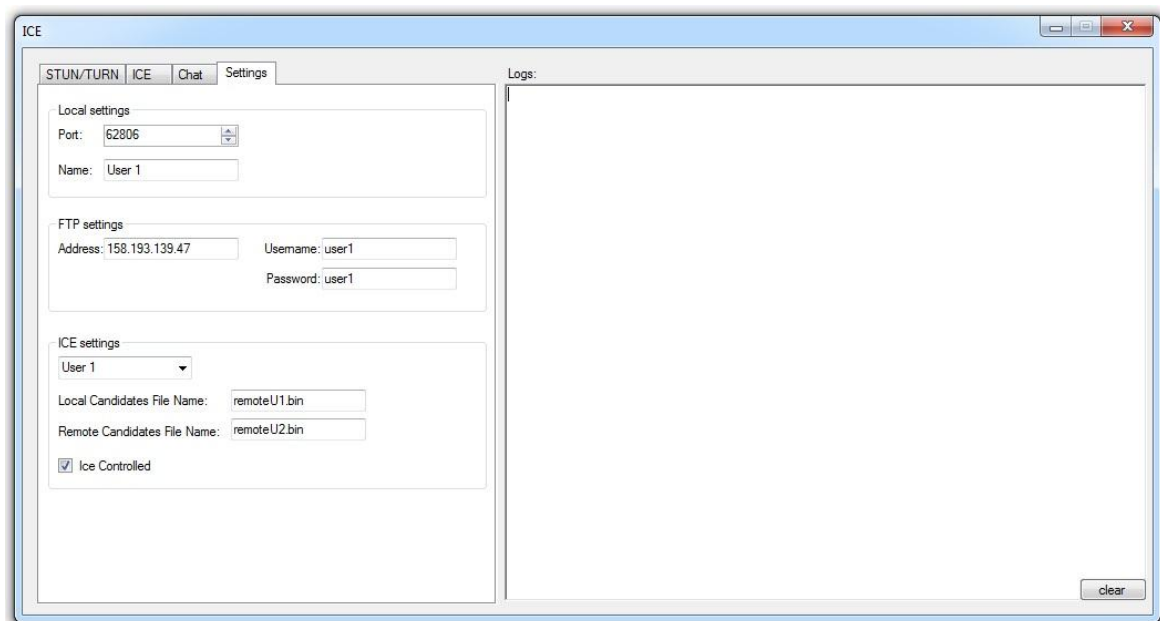
Podproces – Určenie najvhodnejšieho komunikačného páru

Príloha č.6

Podpora vytvárania atribútov	Podpora spracovania atribútov
DATA	DATA
ICE-CONTROLLED	ERROR-CODE
ICE-CONTROLLING	ICE-CONTROLLED
LIFETIME	ICE-CONTROLLING
PRIORITY	MAPPED-ADDRESS
REQUEST-TRANSPORT	PRIORITY
SOFTWARE	SOFTWARE
USE-CANDIDATE	XOR-MAPPED-ADDRESS
XOR-MAPPED-ADDRESS	XOR-PEER-ADDRESS
XOR-PEER-ADDRESS	XOR-RELAYED-ADDRESS

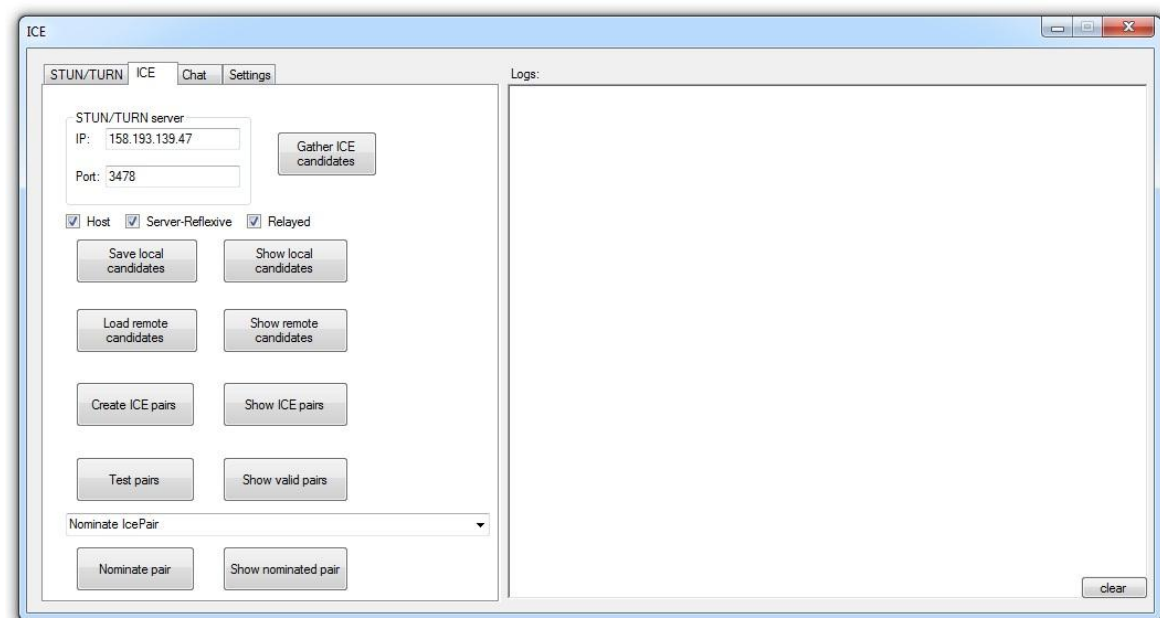
Tabuľka podporovaných STUN atribútov

Príloha č.7



Úvodná obrazovka IceClient

Príloha č.8



Záložka ICE