

Přenos dat prostřednictvím mobilního komunikačního kanálu

Martin Kodovský

ABSTRAKT

Za posledních 20 let proběhl velký vývoj veřejně dostupné, bezdrátové telekomunikace. Není ale pravdou, že by dnes nějak stagnoval. Obecně opomíjená však zůstává skutečnost, že dostupné služby, které lze lehce zakomponovat do stávajících sítí jsou operátory zanedbávány. Tato práce si neklade za úkol polemizovat nad rentabilitou takových služeb, ale jednu konkrétní transparentně vytvořit. Výsledek bude ověřen v paketovém spektru GPRS sítě GSM ve veřejně dostupných podmínkách a kritérium jeho úspěšnosti bude závislé pouze na ověřitelnosti. Práce se nesnaží zrychlit samotný přenos dat, avšak zkrátit a zjednodušit koncept spojení. To však s sebou přináší nutnost analýzy v rozmanitějších podmínkách, pokud možno, na proudových datech, a to už spadá mimo rozsah této práce.

Klíčová slova: GSM, UMTS, GPRS, tunelování dat, digitální data, adresace v síti GSM/UMTS

ABSTRACT

In the past 20 years, there has been a significant progress on research of the public wireless telecommunication. On the other hand, it is not true, that it is stucked at the moment. But there is also a fact, that carrier providers are not very keen in applying easily implementable services, to be considered. This thesis resumes not in argueing about how efficient such services are from economy perspective, but to develop one. The result will be verified in the spectrum of GPRS which is a part of generally known GSM network. The verification will be the only criteria for success over here. The thesis tries not to speed the bitrate of the data up, but to siplify the concept of the establishing of the channel. A general success would be to analyze the result in a variety of spread conditions, using stream data, which is out of the scope of this thesis.

Key words: GSM, UMTS, GPRS, data tunneling, digital information, GSM/UMTS addressing

Děkuji panu profesoru Karlu Vlčkovi za teoretické vedení, cenné rady a trpělivost při sdílení poznatků. Dále bych rád poděkoval panu docentu Miroslavu Vozňákovi z Vysoké školy Báňské, který mě k této práci inspiroval.

Prohlašuji, že:

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce; byl jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo diplomovou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, 21. 5. 2012

Martin Kodovský

Obsah

| | |
|---|----|
| Úvod | 9 |
| I. TEORETICKÁ ČÁST | 10 |
| 1. Možnosti přenosu videa v GSM | 11 |
| 1.1. H.324-M | 11 |
| 1.2. VoIP | 11 |
| 1.2.1. SIP | 11 |
| 1.3. Zhodnocení možností | 12 |
| 2. Struktura sítě GSM | 13 |
| 2.1. GPRS | 14 |
| 2.1.1. Brána | 14 |
| 2.1.2. PDP kontext | 14 |
| 2.1.3. Protokoly GPRS sítě | 14 |
| 2.2. UMTS | 15 |
| 2.2.1. Protokoly UMTS sítě | 16 |
| 3. IPv4 | 17 |
| 3.1. TCP | 17 |
| 3.2. UDP | 17 |
| 4. Tunelování | 19 |
| 4.1. NAT | 19 |
| 4.2. UDP over UDP technika | 20 |
| 4.3. STUN | 20 |
| 4.4. UDP Hole Punching | 20 |
| 5. Podpora protokolů v OS | 22 |
| 5.1. Symbian | 22 |
| 5.2. Linux | 22 |
| II. PRAKTICKÁ ČÁST | 23 |
| 6. Struktura projektu | 24 |
| 7. Symbian agent | 25 |
| 7.1. Uživatelské rozhraní | 25 |
| 7.1.1. Application | 26 |
| 7.1.2. Document | 26 |
| 7.1.3. AppUI | 26 |
| 7.1.4. Model | 27 |

| | |
|--|-----------|
| 7.1.5. View | 27 |
| 7.2. Socket engine | 31 |
| 7.3. SIP engine | 31 |
| 8. Linuxová část | 32 |
| 8.1. Sofia-SIP | 32 |
| 9. Analýza výstupu projektu | 34 |
| 9.1. Propustnost | 34 |
| 9.2. Odezva | 35 |
| 10. Použité nástroje | 37 |
| Závěr | 38 |
| Conclusions | 39 |
| Seznam použité literatury | 40 |

Úvod

Motivací pro tuto práci se stal přenos videosekvence, respektive videohovoru, po GSM nebo UMTS síti. Dnes je tato služba již běžnou záležitostí u širokopásmových kabelových i bezdrátových počítačových sítí. V sítích panují však jistá omezení v přenosu (proudových) dat. Nejnovější technologií v době vzniku této práce v oblasti GSM je 4. generace této sítě. Jde o technologie známé jako WiMax, či LTE Advanced, které však v České republice nejsou běžně dostupné. Proto se tato práce ubírá spíše směrem 3G sítě. Její koncept však může být uplatněn i v nástupní generaci, neboť jedna z požadavků na 4G je, že je založena na celkové IP paketově přepínané síti. Což znamená, že i přenos hlasu je uskutečňován tímto způsobem, protože technologie je natolik rychlá a promptní.

I. TEORETICKÁ ČÁST

1. Možnosti přenosu videa v GSM

Pokud se tedy zaměříme na 3G síť, existují zde v zásadě 2 principy pro uskutečňování videohovorů. Je to použití 3G-324M rodiny protokolů a připojení přes VoIP ústřednu v ISDN subsystému.

1.1. H.324-M

Jedná se o rodinu protokolů H.245, H.223, H.223, H.324. Každý z těchto protokolů zařizuje část nutnou pro navázání spojení a vytvoření kanálu pro videohovor, či videokonferenci v rámci 3G sítě (UMTS). Je to právě mechanismus, který obešel nutnost využívat IP paketově přepínané sítě. Zajistil totiž plynulý přenos multimediálních dat, aniž by trpěl relativně vysokou chybovostí IP paketů v 3G a hlavně špatnou adresovatelností, způsobenou malým rozsahem protokolu IPv4.

1.2. VoIP

Naopak technologie VoIP byla navržena právě pro paketově přepínané sítě. Jde opět o rodinu protokolů na různých vrstvách OSI modelu. Buď je to UDP, nebo RTP. Pro signalizaci je to například ICMP, a pro samotné spojení SIP, IAX2. Podobně jako u předešlé technologie, každý ze zmíněných protokolů zastává část samotného spojení.

Tuto práci nejvíce zajíma SIP, coby zprostředkovatel spojení, nebo-li adresace.

1.2.1. SIP

Je to aplikační, řídicí protokol textového charakteru. Byl vyvinut k propojení dvou a více síťových uzlů. Většinou je používán právě v kombinaci s jiným aplikačním protokolem v závislosti na potřebách komunikačního kanálu.

Podporuje těchto 5 aspektů:

- uživatelské zaměření: ověření koncového systému použitého v komunikaci
- dostupnost: ověření souhlasu obou (více) stran ke spojení
- uživatelské možnosti: ověření možností přenosu a jeho parametrů
- kanálové nastavení: vyzvánění, nastavení parametrů spojení na obou (více) koncích
- údržba spojení: začátek a ukončení spojení, změna nastavení parametrů spojení, volání služeb

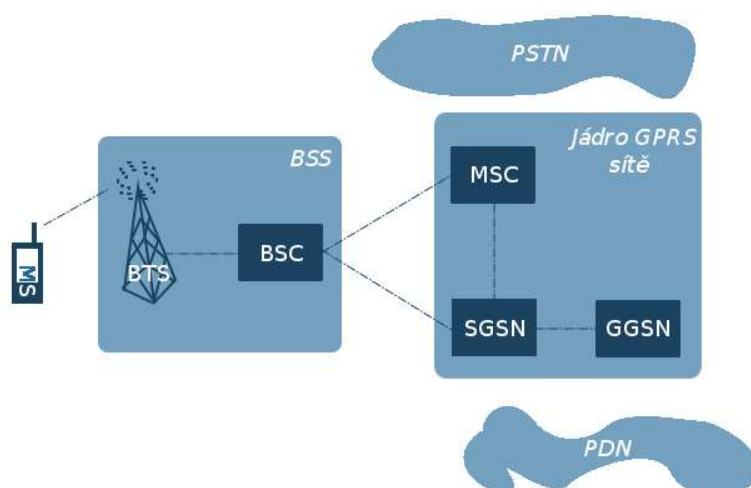
Jeho typické využití je právě ve VoIP komunikaci. Vytváří zde abstrakci nad síťovým rozhraním tak, že UDP pakety, nesoucí videohovor prostupují SIP vytvořeným kanálem.

1.3. Zhodnocení možností

Oba přístupy lze porovnat z více hledisek. H.324M je ideálním řešením, které v sobě integruje spolehlivost a rychlost. Jeho hlavním úskalím však zůstává fakt, že je navržen pouze pro UMTS síť a je zcela v kompetenci operátora. Pokud operátor tuto službu aktivně nepodporuje, protokol nelze použít. Dnes je navíc služba minimálně využívána také proto, že její cena je u poskytovatelů vysoká. VoIP lze oproti tomu využít v paketovém pásmu tarifikace, která je dnes u operátorů běžně dostupná v paušálních sazbách. Její nevýhoda však tkví hlavně v tom, že narozdíl od přepojování okruhů je málo časově spolehlivá. Dnes, pokud se rozhodneme tuto službu využívat, musíme počítat s velkým datovým přenosem a častým dobíjením přístroje. Je to z toho důvodu, že pro VoIP v GSM sítích potřebujeme 3. stranu, nebo-li VoIP ústřednu s naší identifikací. Spojení je pak aktivně udržováno přes pakety putující z našeho telefonního přístroje, bránu mobilního operátora až k VoIP ústředně, kde se náš agent identifikuje a teprve může navázat spojení s druhou stranou.

2. Struktura sítě GSM

GSM vznikla jako 2. generace globální telefonní sítě, v podobě náhrady za tehdejší analogové řešení. Přechodem k digitalizaci, navyšování frekvence a obecně zlepšování technologií vedlo postupně k možnosti posílat i jiný charakter informace, než je hlas. Původně okruhové přepojování začalo být doplňováno paketovým. Co se týče paketového přepojování, po technologiích známých jako CDPD nebo japonském i-mode, přišla síť GPRS, která se na evropském i celosvětovém trhu stala dominantní. Je to technologie, která si svůj odkaz nese až do 3. generace globální telefonní sítě. Zde, v síti UMTS, dochází hlavně k agregaci paketových cest, což má pozitivní vliv na propustnost a rychlost sítě.



Obrázek 1. Zjednodušené schéma GPRS sítě.

| | |
|------|-----------------------------------|
| BSC | Base Station |
| BSS | Base Station Subsystem |
| BST | Base Transceiver Station |
| GGSN | Gateway GPRS Support Node |
| MS | Mobile Station |
| MSC | Mobile Switching Center |
| PDN | Public Data Network |
| PSTN | Public Switched Telephone Network |
| SGSN | Serving GPRS Support Node |

2.1. GPRS

Ke komerčnímu užívání docházelo postupně od roku 2000, a to ve fázích. Zprvu služba podporovala pouze P2P služby, až ve druhé fázi došlo k podpoře P2M[8], tedy k teoretické možnosti zakládat hovorové konference. V roce 2003 bylo GPRS vylepšeno o 8-fázovou PSK modulaci a dalších 5 kódových a modulačních schémat.[11] To způsobilo zvýšení propustnosti na teoretických 59,2 kbit/s/slot.

2.1.1. Brána

GGSN je označení pro bránu do PDN, respektive pro X.25 síť přes GPRS. Ta spolupracuje s SGSN, což je článek GSM infrastruktury, který zajišťuje spojení s GSM zařízeními. SGSN rovněž zajišťuje vyhledávání GSM zařízení v dosahu, uživatelské ověřování a sběr účtovacích dat pro operátora. Spolupráce mezi SGSN a GGSN probíhá na bázi GTP protokolu, který je dále dělitelný na GTP-C, GTP-U a GTP'. Tento protokol operuje nad TCP/IP a je také jedním z klíčů k vyřešení této práce.

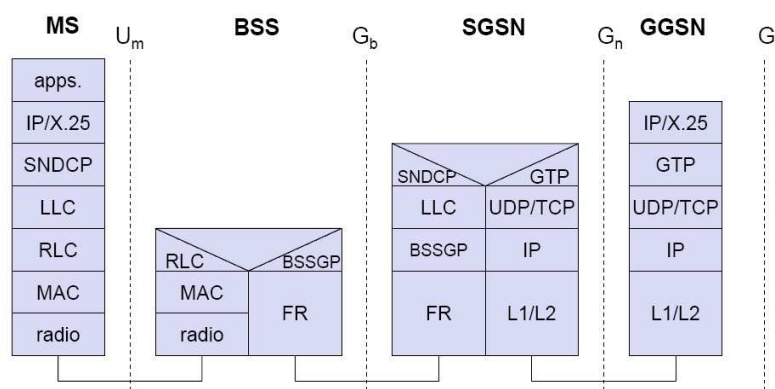
Důležitým pojmem je také PDP kontext, což je datová struktura nesoucí informace o uživatelském spojení. Je přítomna na SGSN i GGSN. Aby mohlo GSM zařízení používat GPRS, musí mít aktivní PDP kontext na bráně.

2.1.2. PDP kontext

PDP kontext poskytuje přístup do externí paketově přepínané sítě skrze PLMN síť[11]. Obsahuje směrovací informace pro paketové spojení mezi MS a GGSN tak, aby byl zajištěn přístup k externí paketově přepínané síti. Unikátně je identifikován MS PDP adresou, což je (privátní) IP adresa v rámci připojení MS a operátora. Aby bylo možné používat více služeb, které užívají rozdílné transportní mechanismy, byl představen sekundární PDP kontext. Sekundárních PDP kontextů může mít MS několik, přičemž každý sdílí jediný unikátní PDP kontext pro přístup do externí sítě libovolného transportního protokolu.

2.1.3. Protokoly GPRS sítě

Vzhledem k diverzitě fyzické vrstvy, potřeby multiplexingu a monitorování dat celého GPRS systému, a to s minimální latencí, se používá celá škála protokolů pro transport dat po páteřní síti GPRS.

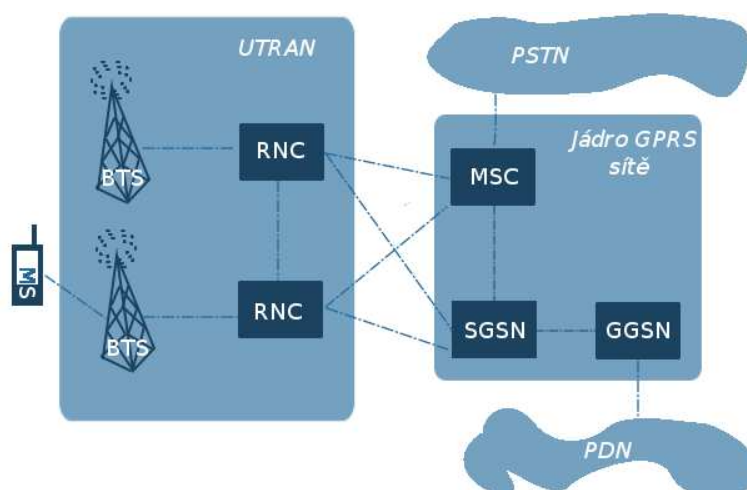


Obrázek 2. Protokolové schéma GPRS sítě.[1]

2.2. UMTS

Je 3. generací globální telefonní sítě. Jako standard byl v roce 1999 uznán organizací ITU, která rovněž vypsala žádost o vytvoření. Vyvinulo ji konsorcium 3GPP, jejímž členem je také evropská organizace ETSI. Je to síť běžící ve 230MHz spektru mezi 1885-2025MHz a 2110-2200MHz (download a upload po řadě).[2]

Její princip spočívá v agregaci paketů z různých BTS stanic. Jde o komplexnější řešení, kde do jádra telefonní sítě vstupuje více BTS stanic, pod RNC uzly, které simultánně přistupují ke SGSN. To umožňuje lepší přístup k většímu počtu časových rámců, protože mobilní stanice není omezena přístupem pouze k jedné BTS. Celý tento článek se nazývá UTRAN a je složen právě z BTS a RNC stanic.



Obrázek 3. Zjednodušené schéma UMTS sítě.

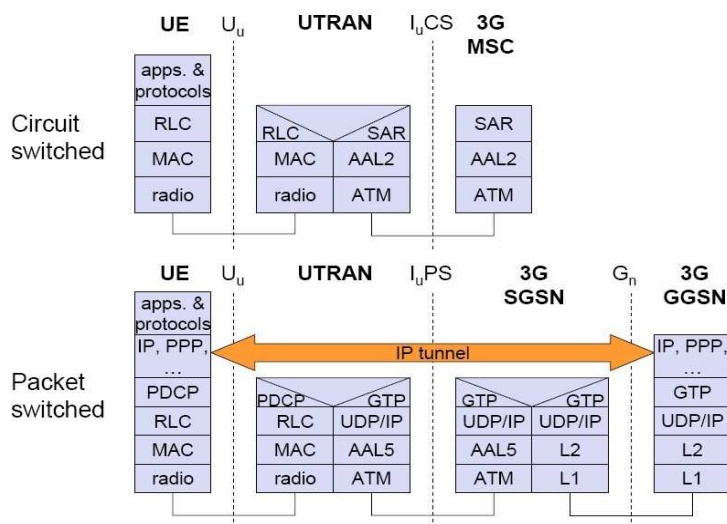
RNC Radio Network Controllers
 UTRAN Universal Terrestrial Radio Access Network

2.2.1. Protokoly UMTS sítě

Na rozdíl od služby GPRS, UMTS síť umožňuje agregaci dat z více BTS stanic. Tím je zajištěna větší přenosová rychlost v místech větší hustoty těchto stanic (za předpokladu, že jejich vytížení není maximální). Některé rutinní aktivity celého procesu spojení byly proto přeneseny z BSS (v UMTS síti UTRAN) na mobilní stanici (UE).

Změny konceptu tedy podléhají zejména MS (UE) a SGSN na straně operátora. Jde především o změnu tzv. MM – *mobility management*. Jde o systém, zajišťující kontinuitu toku dat, napříč pohybem MS od jedné BTS, ke druhé. Vlivem přidání abstraktní vrstvy agregací (URA) v UMTS vznikla potřeba nahradit LLC protokol, operující nad samotným RLC protokolem. V UMTS je tedy LLC nahrazen RRC protokolem, který zajišťuje spojení mezi MS a UTRAN a mezi každou RANAP instancí a SGSN[15, 2].

Poměrně komplexním mechanismem, který tato práce nerozebírá, je pak zajištěna nejen agregace, ale zároveň mobilita přenosových kanálů v rámci sítě UMTS. Tyto mechanismy jsou však transparentní vůči aplikačním vrstvám, ke kterým patří i PDP kontext.

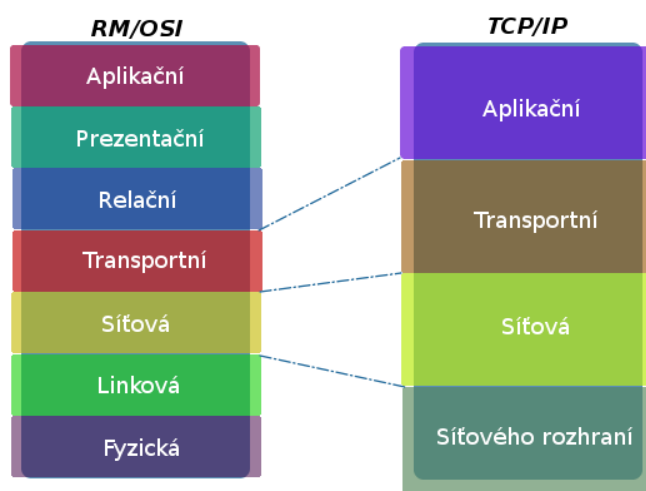


Obrázek 4. Protokolové schéma UMTS sítě.[1]

3. IPv4

Přes to, že tato 4. revize IP protokolu vyšla v raně 80tých letech minulého století, obsluhuje dnes naprostou majoritu veškeré internetové komunikace. V době, kdy vznikla tato práce, byl 32bitový adresní prostor tohoto protokolu téměř vyčerpán. Náhradou IPv4 je 128bitová IPv6. Její podpora v GSM, respektive v UMTS sítích operátorů na našem území běží duálně s podporou IPv4, ale i přes pomalý přechodu na IPv6 v PDNs, má největší zastoupení i zde.

Tato práce nedostatek IPv4 řeší v obecné rovině; totiž abstrakcí nad adresním prostorem kombinací vhodně zvolených protokolů.



Obrázek 5. Mapování vrstvy RM/OSI na vrstvy TCP/IP.

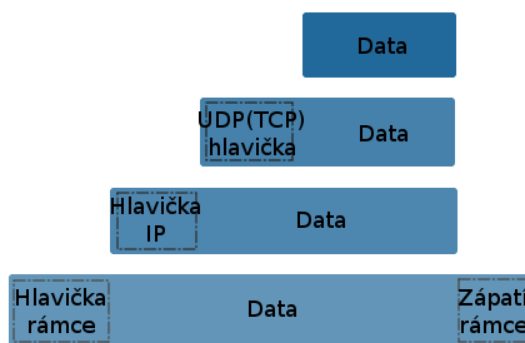
3.1. TCP

Jde o protokol na transportní vrstvě OSI modelu. Zajišťuje spolehlivé doručení paketů, ovšem za cenu větší režie. Je kompletně zdokumentován v IETF RFC 793. Jeho použití v jádru GSM sítě je právě coby podklad pro protokol komunikující mezi GGSN a SGSN – GTP. TCP je rovněž transportní protokol většiny internetových služeb, které majoritní část populace využívá ze svých mobilních telefonů.

3.2. UDP

Stejně jako TCP, i UDP je protokol na transportní vrstvě OSI modelu. Na rozdíl od TCP však nezajišťuje žádný datový kanál, a je proto nespolehlivý z hlediska doručování. Absence nutnosti velké režie z něj však dělá ideální řešení pro potřeby proudových dat jakým je například vysílaný přenos videa. Jeho kompletní dokumentace je k nalezení v IETF RFC 768.

Protokolů v rámci sítě GSM, respektive UMTS, nalezneme ještě celou řadu, a to pouze v IP paketově přepínaných okruzích. Pro potřeby této práce však tento výčet stačí.



Obrázek 6. Schéma datagramu napříč TCP/IP vrstvami.

4. Tunelování

Technika, která dala název této metodě spočívá v překonávání hraničních uzlů sítí. Tyto hraniční uzly mohou oddělovat buď fyzické sítě, nebo pouze vyšší protokolovou vrstvu. Ať už z rozdílného důvodu, vždy se zde střetávají dva různé protokoly nesoucí směrované uživatelské informace. Jeden je protokolem nosným a druhý doručovacím[12]. Nosný protokol nese uživatelská data, která jsou nutná doručit přes rozhraní doručovacího protokolu druhé straně. Doručovací protokol je buď nekompatibilní s nosným rozhraním, nebo jen adresací doručovacího protokolu.

Tunelování tvoří další abstrakci na modely OSI nebo TCP/IP. Doručovací protokol totiž většinou pracuje na vyšší, nebo stejné vrstvě, jako je protokol nosný.

Příklady tunelování jsou: VPN, SSH. Starají se o zabezpečení v LAN, respektive WAN sítích. Dalším příkladem může být GRE, který běží na síťové vrstvě OSI modelu a využívá se v routerech. Jde právě o techniku, kdy navenek neznámé privátní adresy jsou doručovány na síti, kde operují adresy veřejné.[12]

4.1. NAT

Jedná se o překladač síťových adres, který operuje na síťové vrstvě OSI modelu. Je tedy součástí routerů v GGSN. Obecně lze říci, že NAT překleneje problém IPv4 s adresním prostorem, ovšem ve skutečnosti má mnohem více funkcí.

NAT dokáže nejen staticky překládat adresy 1:1, ale také routovat dynamické neregistrované adresy ve smyslu N:1, nebo se vypořádávat s kolizí adres rozhraní. Původní popis lze nalézt v dnes již nahrazeném IETF RFC 1631.

Protože samotný NAT, který běží na síťové vrstvě, nemusí dobře členit protokoly vyšších vrstev, musí i zde existovat mechanismy pro zpětnou kontrolu a obnovení původních paketů. Například při použití TCP, který obsahuje kontrolní součet (angl. *checksum*) na 128.-143. bitu hlavičky může být teoreticky NATem fragmentován. Potom dochází k přesunu více TCP paketů, z nichž první v pořadí nese kontrolní součet a hlavička již obsahuje přeloženou adresu. Ke znovusložení pak dochází na NATu destinace. I zde je tedy vidět komplexita řešení, které začalo vznikat na začátku 90tých let minulého století.

Tato práce operuje pouze na transportní vrstvě a výše (v rámci OSI, respektive TCP/IP modelu), proto i teorii NATu zkrátím na výčet toho nejdůležitějšího, co je o této technologii nutné vědět, aby mohlo proběhnout zapracování.

4.2. UDP over UDP technika

Jde o jednu z tunelovacích technik, kdy dochází k přenosu jednoho UDP paketu v těle druhého UDP paketu. Vzhledem k tomu, že na bráně je standardní UDP port povolen (operátoři v České republice podporují VoIP), tento tunel může fungovat.

Datagramy UDP jsou omezeny 1500 byty. Proto je dobré minimalizovat množství dat v těle tak, aby nedocházelo k přílišné fragmentaci.

UDP tunel tedy zajišťuje rychlý transportní kanál pro aplikace běžící na UDP nebo TCP za operátorovým NATem. Jeho nevýhodou je nespolehlivost, ke které však dochází až ve velké fragmentaci[4], což není případ této práce. Lze říci, že UDP je nativním transportním protokolem pro SIP, takže pokud UDP paket projde k mobilnímu zařízení, je to také důkaz možnosti navázání spojení SIPem.

4.3. STUN

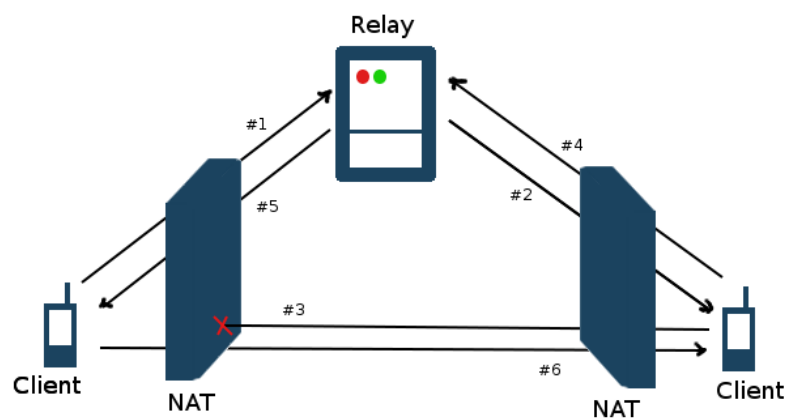
Jméno protokolu vychází z angl. *Simple Traversal of UDP Through NAT*[10]. Je to mechanismus, který byl standardizován v roce 2003 a lze ho nalézt pod označením IETF RFC3489. Narozdíl od svých předchůdců zajišťuje plnou transparentnost programům komunikujících přes uzly NAT, dokáže NAT identifikovat a částečně mapovat síť za ním.

Popsaný protokol STUN umožňuje entitám za NAT první rozpoznat přítomnost a typ NAT a posléze se naučit mapování adres alokovaných NAT uzlem. STUN nepotřebuje žádné změny NAT uzlu, pracuje s libovolným množstvím NAT uzlů za sebou mezi aplikačními entitami a veřejným Internetem.[10]

Výhodou STUN protokolu je fakt, že vhodnou implementací je do komunikace zapojen STUN server s veřejně známou IP adresou, který ve své podstatě vytvoří mapování a následně předá kontrolu nad transferem dat samotným aplikačním entitám. Pokud vezmeme v potaz například VoIP, který STUN často využívá, největší zpoždění dochází pouze v okamžiku spojování. Samotný transfer videa a zvuků obstarávají protokoly pracující s konvenční TCP/IP sítí.

4.4. UDP Hole Punching

Je technika, kterou používá mimo jiné právě STUN, aby docílil mapování na komunikujících NAT uzlech. Zároveň překlene problém datagramů, a to sice jejich statičnost.[4] U TCP spojení dochází ke kontrole příjmu paketů, a proto je nutné držet jakési stálé spojení. Aby toto spojení fungovalo i u UDP komunikace, je nutné jednou za čas odeslat tzv. keep-alive UDP paket, aby spojení nezaniklo.



Obrázek 7. Schéma techniky UDP Hole Punching za pomoci relay serveru.

Hole punching je poměrně známá technika v UDP komunikaci, ale lze ji spolehlivě použít i pro vytvoření TCP spojení.[4]

5. Podpora protokolů v OS

Pokud se zaměříme na síťovou vrstvu OSI modelu, podpora je zřejmá a vychází z ovladačů jednotlivých síťových rozhraní. Samotné operační systémy pak implementují nakládání s protokoly transportní vrstvy a vyšší. Operační systémy dnes již přímo ve svých jádrech nejčastěji obsahují tzv. TCP/IP zásobník, ke kterému mohou přímo přistupovat *sokety*.

Ty jsou identifikovány IP adresou a portem a tvoří unikátní deskriptory v paměti pro přístup k síťovému rozhraní a oddělení síťové od transportní vrstvy příchozích dat.

Většina operačních systémů implementuje API pro práci se sokety. Dnes jsou známy především BSD (Barkley sockets) používané v UNIXových systémech a WinSock ve Windows. V Symbianu, který je neméně důležitý pro tuto práci je implementováno speciální API, které je však velmi blízké zpracování právě BSD[6]. Dále nabízí POSIXové soketové API.

5.1. Symbian

Symbian soket server zprostředkovává komunikaci mezi jednotlivými *end-pointy*. Podpora protokolů je zajištěna DLL zásuvnými moduly. Interface mezi soket serverem a klientskou aplikací zajišťuje klientský modul, který má linkovatelnou hlavičku. Ta obsahuje hlavičku tříd `RSockServ` coby deklarace soket serveru a `RSocket` – jeho jednoho soket spojení[9].

Symbian sám o sobě obsahuje také knihovny pro SIP. Zároveň k nim poskytuje API, které je tvořeno z tříd jako například `CSIP`, `MSIPObserver` a další.

5.2. Linux

Linux implementuje UNIXové BSD API. V kernelu GNU/Linux je to takzvané *Protocol agnostic interface*, které se stará o soketovou vrstvu síťové komunikace[7]. Struktura, která deklaruje jednotlivé sokety je k nalezení v knihovně `sock.h`. Ta zahrnuje strukturu zvanou *proto*, kterou musí každý z protokolů implementovat. Spojení a nastavení *Protocol agnostic interface* iniciuje klientská aplikace. Díky této abstrakci není nutné přenášet pouze transportní protokoly jako je UDP a TCP, ale i IP, rámce Ethernet nebo SCTP. Na této vrstvě poté staví celá škála aplikačních protokolů, které už ovšem nejsou přímo součástí GNU/Linux, ale spíše jednotlivých distribucí nebo 3. stran.

II. PRAKTICKÁ ČÁST

6. Struktura projektu

Projekt je rozdělen do následujících částí:

- Symbian agent
- Linuxový paketový emitör

Obě části spolupracují za účelem pokrytí problému vnější adresace mobilního zařízení. Pokud je možné adresovat mobilní zařízení externě, je možné adresovat jej i z jiného mobilního zařízení. Tedy je možné uskutečnit paketový kanál, potažmo videohovor bez využití služeb 3. stran, mezi dvěma mobilními stanicemi.

Agentová část v tomto případě poběží na mobilním zařízení. Jejím úkolem bude vytvořit PDP kontext na straně operátorovy brány a zobrazit dostupná adresační data, kterými jsou privátní a veřejná adresa. Jinými slovy: privátní IP adresa MS a veřejná IP adresa GGSN. Protokolem pro komunikaci je SIP, který má vhodné vlastnosti pro procházení NATem, v našem případě operátorovou GGSN bránou.

Demonstrací pro uskutečnitelnost daného přenosu, je prostý UDP paket, v jehož datové části bude jednoduchý textový řetězec. Po navázání spojení se tato zpráva objeví na displeji mobilního zařízení.

Linuxová část slouží jako paketový emitör, tedy jako vysílač. Pro svou činnost potřebuje mít zakomponovaný mechanismus, který dokáže nosnou zprávu tunelovat. Tento mechanismus bere veřejnou IP adresu destinace a port, jako svůj vstup. Jsou to právě informace, které jsou poskytnuty agentovou částí. Tato část rovněž obsahuje prostý vyhodnocovač latence spojení, nad jehož výstupem lze provést analýzu statistik, pro teoretické zhodnocení.

Vzhledem k faktu, že agentová (mobilní) naslouchací část používá nejspíše GPRS endpoint, vyskytuje se za NATem. Před samotným nasloucháním je tedy nutné “prošťouchnout” NAT adresou a portem vysílacího zařízení, tedy emitovat takový UDP paket směrem k rozhraní, kde běží linuxová část. O to se stará metoda SendTo(...) třídy RSocket.

To vymezuje i další předpoklad, a sice že linuxová část běží na rozhraní s veřejně známou adresou. Použití naslouchání broadcastem se doposud testuje, avšak s největší pravděpodobností bude v praxi nutné tyto informace sdělovat po jiném komunikačním kanálu – například synchronizačním protokolu samotné GSM.

7. Symbian agent

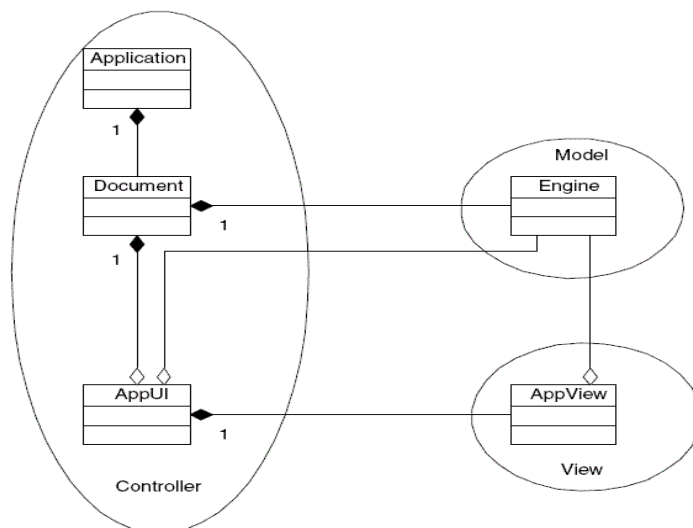
Je to aplikace vyvinutá pro operační systém Symbian (konkrétně Symbian OS 9.3, S60 rel. 3.2). Po spuštění se automaticky připojuje přes GPRS komunikační port a snaží se vytvořit PDP kontext na bráně a tím pádem se zaregistrovat k připojení k GPRS. Po navázání kontaktu s operátorovou bránou zobrazí svou privátní a veřejnou adresu a otevře socket pro naslouchání na portu 5060 (standardní UDP port, který standardně zachovává i SIP). Tím jeho inicializace končí.

V momentu, kdy přijde z brány směrovaný UDP paket na tento port, pokusí se přechít jeho strukturu, potažmo data a v nich najít SIP zprávu. Samotné data zprávy (což v našem případě bude textový řetězec), vypíše do konzole.

Vzhledem k povaze systému Symbian a jeho nativního jazyka je program vytvořen v rámci objektového paradigma. Základními prvky jsou uživatelské rozhraní, socket engine a sip engine.

7.1. Uživatelské rozhraní

Program je psán v jazyce C++. Staví na principech návrhového vzoru MVC (Model-View-Controller) a Observer. V systému Symbian do části *Model* řadíme třídy odvozené z `CAknApplication`, `CAknDocument` a `CAknAppViewUI`. V části *View*, je to `CAknView`, popřípadě pokud program disponuje pouze jedním pohledem, může tento pohled reprezentovat samotný prvek uživatelského rozhraní.



Obrázek 8. Model View Controller v Symbianu.[1]

7.1.1. Application

Tato část programu je vstupním bodem. Vytváří programový zásobník a konstruuje Document. Třída se nazývá `CPingApp` a dědí ze systémové `CAknApplication`. SDK pro Symbian rovněž vytváří zdrojové cesty, přidává lokalizace a zdrojové soubory. Například reprezentace vstupního bodu Symbian agenta je ikona, zobrazená mezi jinými v grafickém menu Symbianu.



Obrázek 9. Označená ikona Symbian agenta – Ping.

7.1.2. Document

`CPingDocument` je třídou implementující tuto část programu. Jeho jednou členskou proměnnou je pole deskriptorů, uchovávající stav konzole mezi jednotlivými pohledy. Zároveň v konstruktoru vytváří `AppUI`.

7.1.3. AppUI

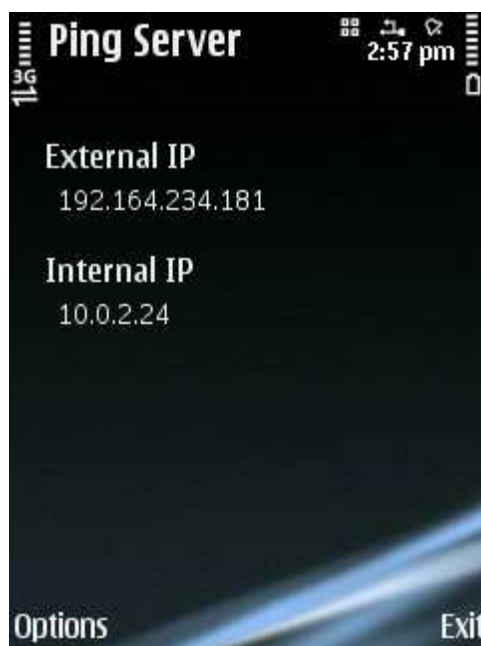
Klíčová třída `CPingAppUI` vytváří rozhraní pro uživatelský vstup a výstup, a je vlastníkem jednotlivých pohledů. Její metody obsluhují, případně distribuují události generované stiskem klávesnice. Je to zároveň *observer* sledující změny stavu programu tak, aby na ně socket engine (`CPingServerEngine`) mohl reagovat, například obnovením spojení.

7.1.4. Model

Model je obecná část programu, která obsahuje data a manipuluje s nimi. Tyto data představují lokální a externí IP adresa, popřípadě end-point, což je GPRS (resp. EDGE nebo 3G) rozhraní daného telefonu. Tato data slouží jako indikátor spojení. Stará se o ně třída `CPingServerEngine`. Manipulaci samotných zpráv přes soket zajišťuje `CPingSIPEngine`, který kromě detekce také extrahuje zprávu z příchozího SIP paketu.

7.1.5. View

Program obsahuje 2 pohledy, grafický seznam a konzoli. Grafický seznam zobrazuje lokální a externí adresu zachycenou instancí `CPingAppUI` od socket engineu. Konzole zobrazuje jakoukoliv změnu vzniklou v části Model. Oba pohledy jsou součástí C++ templatu, čímž je zjednodušená jejich konstrukce a obsluha.



Obrázek 10. Grafický pohled programu Ping.

Na hlavní obrazovce uživatel vidí dva pasivní boxy. První – nadepsaný *External IP* – zobrazuje po úspěšném připojení k síti GPRS veřejnou adresu mobilního zařízení (tedy adresu GGSN brány), a druhý – nadepsaný *Internal IP* – zobrazuje lokální adresu, udržovanou pouze v samotném zařízení a PDP kontextu přítomného jak v SGSN, tak GGSN.



Obrázek 11. Konozolový pohled programu Ping.

Uživatelský vstup z klávesnice telefonu je redukován na pohyb ve standardním menu. Uživatel má k dispozici tyto funkce:

- Obnova PDP kontextu a připojení (Refresh)
- Vymazání konzole (Clear Log)
- Přepnutí pohledu (Modes)
- Generování předvolené zprávy (Dummy Message)
- Zavření aplikace (Exit)



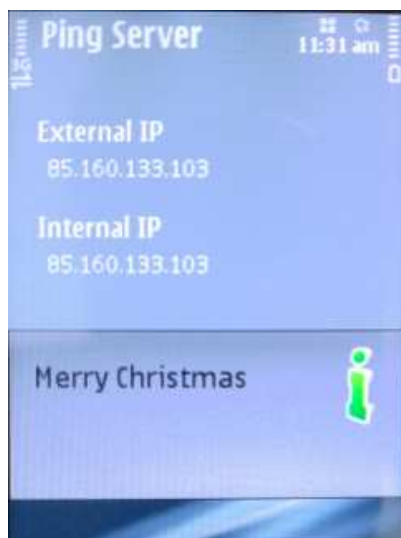
Obrázek 12. Menu programu Ping.

Po zvolení a potvrzení první položky z menu se systém uživatele dotáže na přístupový bod k síti. Uživatel by měl vybírat z těch přístupových bodů, které se váží k paketovému spektru poskytovanému operátorem. Systém však bude fungovat i v prostředí bezdrátové sítě WiFi, ale i v síti zprostředkované bluetooth rozhraním.



Obrázek 13. Menu programu Ping.

Po přijetí zprávy se uživateli zobrazí informační box s danou SIP zprávou. Zpráva se také zaznamená v konzoli.



Obrázek 14. Přijatá zpráva v informačním boxu.

7.2. Socket engine

Socket engine je DLL modul toho agenta, který zajišťuje vytváření a ukončování spojení, sledování pohybu na síti. Využívá k tomu dříve zmíněné třídy `RSocket` a `RSockServer`.

Je vlastněn `CPingDocument` třídou, která jeho referenci předává i dalším z řady kontrolérů. Ihned v konstruktoru se vytváří připojení, které zakládá PDP kontext na straně operátora a následně soket pro odposlech UDP portu pro příchozí zprávy. V každém kroku se plní záznamový buffer, který ukládá zprávy z instancí tříd starajících se o jednotlivé kroky připojení.

Tento modul je znovu použitelný v jiných, na Symbianu založených, aplikacích, protože je kompilovaný zvlášť, jako dynamická knihovna.

Socket engine zároveň tvoří model v návrhovém vzoru MVC, tedy centrální logiku celé aplikace.

7.3. SIP engine

SIP protokol do této práce vstupuje nejen jako iniciátor spojení, ale také jako nosič zprávy. Sám o sobě to tento protokol nemůže zvládnout. Jde o SDP, který popisuje vlastnosti zahajovaného kanálu. Také není určen k přenosu vlastní zprávy, ale atribut `i=*` nám toto umožní.

SIP engine využívá standardní Symbian SIP stack pro práci s tímto protokolem. Linux klient je pro změnu postaven na dále zmíněném Sofia-SIP stacku, což je soubor open-source knihoven pro práci s tímto protokolem. Skloubením těchto dvou řešení program ověří, že předpoklady a implementace dosavadního řešení jsou správná.

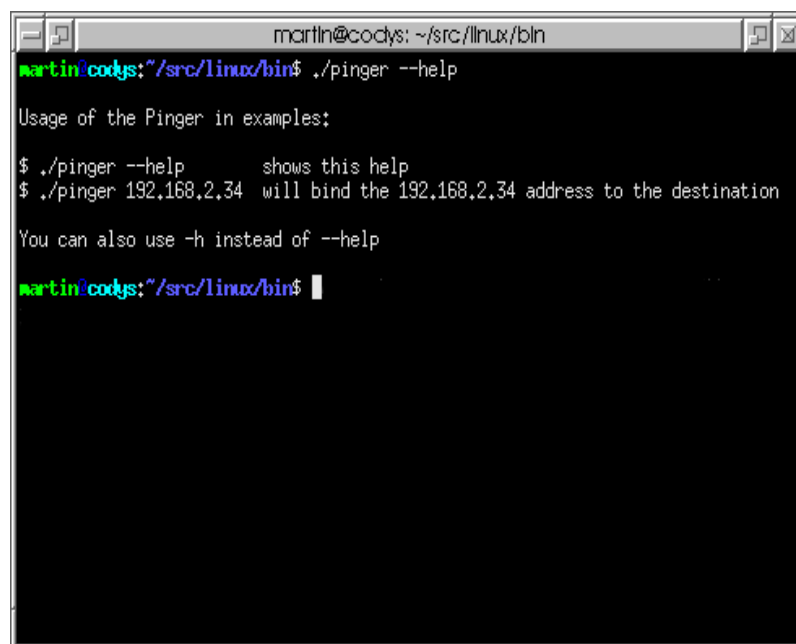
Do agenta je obsluha SIP protokolu zakomponována dynamickou knihovnou. SIP engine je připraven k zahajování spojení a dekodování síťového provozu na ním určeném portu. V této práci je to port 5062 pro odposlech na agentové (serverové) straně. Engine je navíc sledován observerem, což je `CPingAppUI` třída. Všechny události jsou distribuovány jednotlivým pohledům.

Pro příjem zprávy se používá virtuální metoda standardní třídy Symbian SIP stacku, `IMReceived`, která je volána z knihovny s deskriptorem dané zprávy a jejím odesílatelem.

8. Linuxová část

Jde o jednoduchý konzolový program, který bere jeden parametr. Je to IP adresa, na které telefon naslouchá

Uživatel je o tomto parametru notifikován v případě zadání špatného počtu, popřípadě formátu těchto parametrů. Program rovněž obsahuje nápovědu, která se vyvolá s přepínačem `--help`.



```
martin@codys: ~/src/linux/bin
martin@codys:~/src/linux/bin$ ./pinger --help

Usage of the Pinger in examples:

$ ./pinger --help      shows this help
$ ./pinger 192.168.2.34 will bind the 192.168.2.34 address to the destination

You can also use -h instead of --help

martin@codys:~/src/linux/bin$
```

Obrázek 15. Zobrazení nápovědy v programu pinger.

Program využívá síťových prostředků užívaných systémem GNU/Linux. Proto by uživatel, který ho spouští, měl mít dostatečná práva, aby tyto prostředky mohl využít, popřípadě ho spouštět pod právy superuživatele.

Program používá NUA rozhraní Sofia-SIP stacku pro inicializaci a zpracování SIP zpráv.

8.1. Sofia-SIP

Jedná se o open-source stack implementující většinu standardu IETF RFC3261. Vzhledem k faktu, že tento soubor knihoven byl vyvinut firmou Nokia, existuje i port do Symbianu.

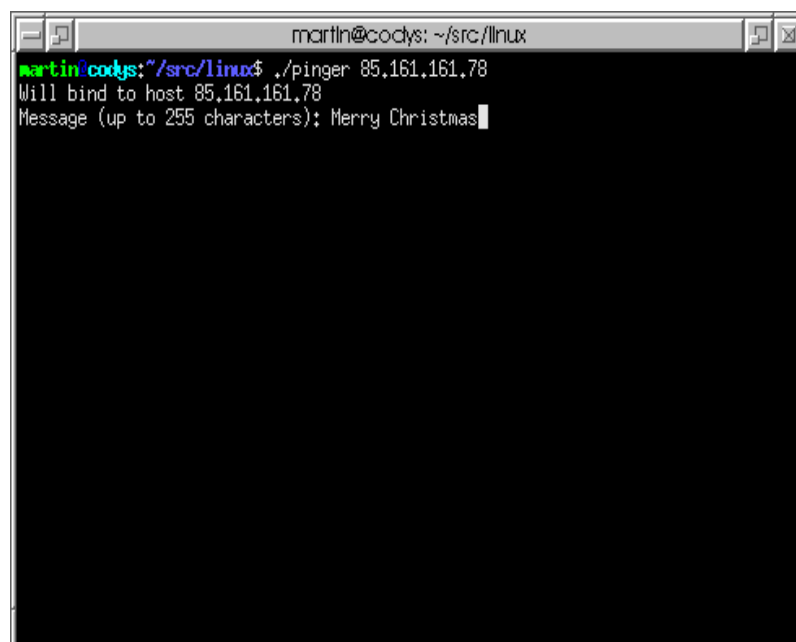
Mimo jiné, knihovna podporuje i NAT traversal, který je součástí jiného standardu – RFC3489. Ten lze použít i mimo standardních volání metod pro obsluhu SIP.

V projektu je použitý poslední release této knihovny: 1.12.11.

Program je psán v jazyce ANSI/C C99 standardu. V těle programu je volána smyčka Sofia-SIP stacku, která je omezena jedním krokem, a sice odesláním zprávy. Ta je odeslána na adresu danou PDP kontextem Symbian agenta. Program tuto adresu dostává jako vstup pomocí argumentu z příkazové řadky.

Po spuštění požaduje uživatelský vstup – text odesílané zprávy. Ta je potom pomocí NUA rozhraní Sofia-SIP stacku zpracována a odeslána na adresu.

Pokud by smyčka byla delší, program by mohl vyčkat na odpověď ze strany agenta. Projekt tuto situaci využil k měření zpoždění odesílaného paketu.

A screenshot of a terminal window with a title bar that reads 'martin@codys: ~/src/linux'. The terminal shows the following text: 'martin@codys:~/src/linux\$./pinger 85.161.161.78', 'Will bind to host 85.161.161.78', and 'Message (up to 255 characters): Merry Christmas'. The cursor is positioned at the end of the message line.

```
martin@codys:~/src/linux$ ./pinger 85.161.161.78
Will bind to host 85.161.161.78
Message (up to 255 characters): Merry Christmas
```

Obrázek 16. Odesílání zprávy prostřednictvím pingeru.

9. Analýza výstupu projektu

Výsledky této práce přinesly ověření tunelování po paketově přepínané síti za standardních podmínek operátora. Pro reálné použití, a to například na proudových datech, je třeba uvést alespoň teoretické charakteristiky takového kanálu.

Jedním z kritérií je například odezva celého systému udávaná tzv. latencí, což je zpoždění mezi odesláním a doručením paketu. K největšímu zpoždění dochází mezi MS a BSS, a to z důvodu velké chybovosti vzruchu informace v rádiovém pásmu. Přesto jsou výsledky celého systému blízké datům uvedených v tabulce:

| | | |
|-----------------------------|------|------|
| Velikost ping paketu (byte) | 10 | 1500 |
| Ping paketů posláno | 1000 | 1000 |
| Echo přijmuto od | 1000 | 1000 |
| Ztráty | 0 | 0 |
| Minimální čas odezvy (ms) | 508 | 2098 |
| Maximální čas odezvy (ms) | 883 | 2414 |
| Průměrný čas odezvy (ms) | 590 | 2103 |

Tabulka 1. Latence ping paketů dle velikosti.
[14]

V kódovém schématu CS4 dosahuje GPRS až 59kbit/s rychlosti[14]. EDGE Evolution je zhruba 4krát rychlejší a u UMTS je teoretická propustnost až 14,4Mbit/s.

Ve skutečnosti však rychlost GPRS osciluje někde mezi 40-50kbit/s, EDGE kolem 120-160kbit/s a UMTS(HSDPA) u Telefonicy O2 kolem 2,5Mbit/s. U všech technologií je nutné brát ohled na zatížení sítě a vzdálenost od připojené stanice, protože podle toho se pak upravují kódová schémata, a tím i celková propustnost.

9.1. Propustnost

Vzhledem k faktu, že samotný STUN propustnost v konečném důsledku neovlivňuje, protože po vytvoření komunikačního kanálu se komunikace redukuje na adresaci NATů protějšku, práce uvádí pouze teoretické a naměřené hodnoty bez samotného tunelování.

Dále bylo provedeno měření zhruba na 6 místech, pokaždé s jinou charakteristikou signálu. Měření proběhlo několikrát za sebou na každém z míst, kvůli lepším statistickým výsledkům. Jejich průměr je v následující tabulce:

| | Download | Upload | Latence |
|-------------|----------|-----------|---------|
| UMTS(HSDPA) | 3.6Mbps | 1.4Mbps | 140ms |
| EDGE | 238Kbps | 118.4Kbps | ~500ms |
| GPRS | 80Kbps | 40Kbps | ~500ms |

Tabulka 2. Teoretické hodnoty technologií poskytovaných O2 Telefonickou.

| | Download | Upload | Latence |
|-------------|-----------|------------|---------|
| UMTS(HSDPA) | 233.1Kbps | 109.79Kbps | 275ms |
| EDGE | 166Kbps | 65.53Kbps | 421ms |
| GPRS | 80Kbps | 40Kbps | 506ms |

Tabulka 3. Naměřené technologií poskytovaných O2 Telefonickou.

9.2. Odezva

Doba připojování k operátorovi je také podstatným faktorem. V případě, že by se jednalo o hovor, nejdelší prodleva by nastala při samotném paketovém připojení obou stanic k operátorovi. Tabulka pod textem zachycuje měření času od spuštění vytvoření kontextu na bráně až do obdržení její adresy za předpokladu, že kontext neexistoval.

| | Doba připojování |
|-------------|------------------|
| UMTS(HSDPA) | 3.75s |
| EDGE | 4.1s |
| GPRS | 7s |

Tabulka 4. Doba připojování k bráně v ms.

Za použití signalizace přes SS7 hovorového spektra GSM signálu místo STUN (relay) serveru, by spojení proběhlo teoreticky asi za 20-25 vteřin, s odezvou pod 600ms, za předpokladu, že by se obě strany nacházely v oblasti pokryté technologií HSDPA a síť by nebyla příliš vytížená.

Výsledný datový tok by se následně odvíjel od pomalejšího uploadu, tedy řádově 900Kbit/s.

V oblasti pokryté pouze nejhorším GPRS pro obě strany, by spojení proběhlo zhruba za 50s, s latencí přes 1s, což činí spojení pro videohovor téměř nepoužitelné. Datový tok by se pohyboval také na úrovni 30Kbit/s, což stačí stěží na přenesení videa o rozměrech 320x240px v dostačující kvalitě.

Za použití kodeku H.264 se dostáváme na bitrate okolo 190Kbit/s, což při dobrém signálu, vzdálenosti od Node-B stanice a přiměřené vytíženosti sítě lze dosáhnout.

10. Použité nástroje

Veškerý vývoj proběhl na notebooku osazeným procesorem Pentium M Centrino 1,6GHz, 1GB operační paměti a 60GB diskem. Na notebooku běží GNU/Linux jádra 2.6.32-5-686, dist. Debian 6.0 Squeeze.

Na samotný vývoj byly použity tyto softwarové nástroje:

1. gnupoc-1.21-8 – instalátor a wrapper použitého SDK pro GNU/Linux
2. S60_3rd_Edition_FP2_SDK – SDK pro vývoj na Symbian verze v použitém telefonu
3. CodeSourcery GNU ARM Chaintool – kompilátor C/C++ na platformu ARM
4. gcc-4.4 + dostupné standardní knihovny – kompilátor C pro GNU/Linux
5. GNU Emacs 23.2.1 – textový editor

Jako testovací přístroj jsem použil vlastní mobilní telefon Nokia E75 RM-412, s procesorem ARM 11 369MHz, podporující GPRS třídy 32, EDGE 32, HSDPA a běžně dostupným paušálním připojením k mobilnímu internetu (až 3G) od O2 Telefonica.

Závěr

Práce byla koncipována k nalezení řešení optimálnějšího komunikačního kanálu na běžně dostupné bezdrátové telefonní síti. Nutno podotknout, že v průběhu nalézání řešení jsem narazil na mnohá úskalí, ale zároveň na jiná dílčí řešení, která mi dopomohla práci dokončit. Výhodou koncepce zapracované v této práci je, že i přes její malou budoucí životnost, ji lze nyní použít například právě pro přenos videa v obou směrech po paketově přepínané síti bez nutnosti udržovat spojení s jinou ústřednou v rámci ISDN subsystému. Toto spojení by vyžadovalo synchronizaci, která není součástí této práce. Ta by mohla být prováděna signalizačním protokolem hlasového GSM spektra. Přesněji řečeno zasíláním SMS, či flash SMS zpráv a jejich automatickým zpracováním na telefonních přístrojích. Tímto se autor bude nadále zabývat.

Otázkou zůstává, jak bude vypadat tatáž síť po přechodu na IPv6. Budou se rozvíjet formy distribuovaných sítí, když výpočetní výkony dnešních zařízení by mohly převzít část režie ústředen a adresovatelnost již nebude problém? Nechť toto zůstane otázkou na čtenáře a na budoucnost, která je jistě světlá.

Conclusions

The thesis' concept was to find a solution for more optimized communication channel using the general public wireless telephony network. It is good to say, that whilst developing the solution I found many problems, but also many other partial solutions, that already had taken their places and eventually led me to finish it. The advantage of this solution is its capability for example for video transmission over the packet service, without need of having a 3rd party switching centre in the scope of ISDN subsystem. And it is advantage in spite of the solution's lifetime will not be long. Such a usage – videocalls over the packet service – would require yet a synchronisation, which is not a part of this thesis. This synchronisation could be handled by signalling protocol in the voice spectrum of GSM. To be more precise, by SMS or respectively flash SMS and their automatized processing on the GSM devices. This is the way the thesis will carry on. The question still remains: How will the GSM network look like, when IPv6 is fully deployed? When the addressability will be a problem no more, will this shed some light on research and development of distribution networks? Perhaps it should go this way, considering the fact the computing power of the devices known today is already rather high. Let this be a question for the reader and for the future, that is certainly bright enough.

Seznam použité literatury

- [1] CONNING TECH. *Mobile Network Evolution: GPRS*. In: *Blogspot* [online blog]. July 2010 [cit. 2012-06-30]
Dostupné z: <http://www.fee.vutbr.cz/peringer/SIMLIB/>
- [2] ETOH, M. *Next Generation Mobile Systems 3G and Beyond*. England: J.Wiley & Sons. 2005. ISBN-10 0-470-09151-7 (HB).
- [3] ETUTORIALS.ORG. *PDP* Elektronická publikace, 2008. Dostupné z: <http://etutorials.org/Mobile+devices/gprs+mobile+internet/Chapter+7+Signaling+Plane/PDP+context+Management/>
- [4] FORD Brian, et. al SRISURESH, Pyda a Dan Kegel *Peer-to-Peer Communication Across Network Address Translators* Proceedings of USENIX-05, pages 179-192, 2005. Dostupné z: http://static.usenix.org/event/usenix05/tech/general/full_papers/ford/ford.pdf
- [5] GLISIC, G.S. *Advanced Wireless Communication 4G Technologies*. England: J.Wiley & Sons. 2004. ISBN-10 0-470-86776-0.
- [6] HARRISON, Richard. *Programujeme aplikace pro Symbian v jazyce C++*. Computer Press. První vydání. 2006. ISBN: 80-251-1243-8
- [7] JONES, M. Tim. *Linux Network System*. [online], 2007. [cit. 2012-05-25]. Dostupné z: <http://www.ibm.com/developerworks/linux/library/l-linux-networking-stack>
- [8] KELLER, Rachel. *GPRS*. [online], 2001. Dostupné z: <http://misnt.indstate.edu/harper/Students/GPRS/GPRS.html>
- [9] NOKIA. *Fundamentals of Symbian C++/Sockets* [online], 2011. [cit. 2012-07-01]
Dostupné z: http://www.developer.nokia.com/Community/Wiki/Fundamentals_of_Symbian.C%2B%2BSockets
- [10] ROSENBERG, Jonathan et al. IETF RFC 3489: *STUN: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)* [online], March 2003. Dostupné z: <http://www.ietf.org/rfc/rfc3489.txt>
- [11] SEURRE, E. et al. SAVELLI, P. a PIETRI J.-P. *GPRS for Mobile Internet*, ©2003, ISBN 1-58053-600-X.
Dostupné z: <http://flylib.com/books/en/3.68.1.1/1/>

-
- [12] SIMPSON, W. IETF RFC 1853: *IP in IP Tunneling*, [online], October 1995. Dostupné z: <http://tools.ietf.org/html/rfc1853>
 - [13] VLČEK, Karel, Prof. *Kompresa a kódová zabezpečení v multimediálních komunikacích* Ed. Prof. Ing. Dušan Levický, CSc., Prof. Ing. Vladimír Šebesta, CSc.. 2. vyd. Praha: BEN Technická literatura. 2004. ISBN 80-7300-134-9.
 - [14] WAGENTRISTL, Kurt. *Diplomarbeit: GPRS Performance Evaluation*. Wien, 2004. Institut für Nachrichtentechnik und Hochfrequenztechnik der Technischen Universität Wien.
 - [15] YI-BING, Lin et al. YIEH-RAN, Haung, YUAN-KAY, Chen a Imrich Chlamatac. Mobility management: from GPRS to UMTS. *Wireless Communication and Mobile Computing*. 2001, Vol. 1, Issue 4, October/December, p. 339-359. ISSN: 1530-8677

Seznam použitých symbolů a zkratek

| | |
|--------------|---|
| 2G | 2. generace GSM sítě |
| 3G | 3. generace GSM sítě známá jako UMTS |
| 3GPP | 3G konsorcium za účelem vytvoření 3G sítě |
| 4G | 4. generace GSM sítě |
| API | Application programming interface |
| BSD | Berkley socket API |
| CDPD | Cellular Digital Packet Data |
| DLL | Dynamic Library |
| GGSN | Gateway GPRS Support Node |
| GPRS | General Packet Radio Service |
| GRE | Generic Routing Encapsulation |
| GSM | Groupe Spécial Mobile |
| GTP | GPRS tunneling protocol |
| IAX2 | Asteriskový aplikační protokol |
| ICMP | Internet Control Message Protocol |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| ISDN | Integrated Services Digital Network |
| LAN | Local Area Network |
| LTE Advanced | Long Term Evolution |
| NAT | Network Address Translation |
| OSI | Open Systems Interconnection |
| P2P | Point-to-Point |
| PDN | Public Data Network |
| PDP kontext | Packet Data Profile |
| PSK | Phase Shift Keying |
| RFC | Request for Comment – označení dokumentace podle IETF |
| SDP | Session Description Protocol |
| SGSN | Serving GPRS Support Node |
| SIP | Session Initiation Protocol |
| SSH | Secure Shell – aplikační protokol |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| UMTS | Universal Mobile Telephony System |
| VoIP | Video over IP |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WiMax | Worldwide Interoperability for Microwave Access |

Seznam obrázků

| | | |
|-----|--|----|
| 1. | Zjednodušené schéma GPRS sítě. | 13 |
| 2. | Protokolové schéma GPRS sítě.[1] | 15 |
| 3. | Zjednodušené schéma UMTS sítě. | 15 |
| 4. | Protokolové schéma UMTS sítě.[1] | 16 |
| 5. | Mapování vrstvy RM/OSI na vrstvy TCP/IP. | 17 |
| 6. | Schéma datagramu napříč TCP/IP vrstvami. | 18 |
| 7. | Schéma techniky UDP Hole Punching za pomoci relay serveru. . . | 21 |
| 8. | Model View Controller v Symbianu.[1] | 25 |
| 9. | Označená ikona Symbian agenta – Ping. | 26 |
| 10. | Grafický pohled programu Ping. | 27 |
| 11. | Konozolový pohled programu Ping. | 28 |
| 12. | Menu programu Ping. | 29 |
| 13. | Menu programu Ping. | 29 |
| 14. | Přijatá zpráva v informačním boxu. | 30 |
| 15. | Zobrazení nápovědy v programu pinger. | 32 |
| 16. | Odesílání zprávy prostřednictvím pingeru. | 33 |

Seznam tabulek

| | | |
|----|---|----|
| 1. | Latence ping paketů dle velikosti. | 34 |
| 2. | Teoretické hodnoty technologií poskytovaných O2 Telefonickou. . . | 35 |
| 3. | Naměřené technologií poskytovaných O2 Telefonickou. | 35 |
| 4. | Doba připojování k bráně v ms. | 35 |