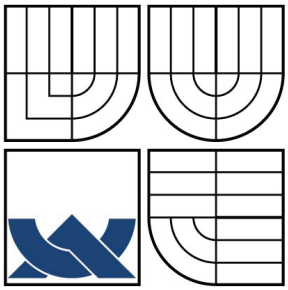
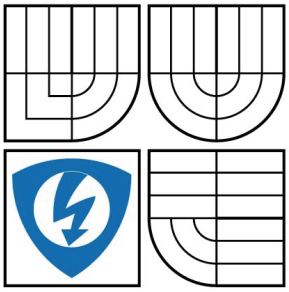


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IP MULTIMEDIA SUBSYSTEM

IP MULTIMEDIA SUBSYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Martin BENDÍČEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. Petr KOVÁŘ
SUPERVISOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bendíček Martin, Bc. *ID: 89239*
Ročník: 2 *Akademický rok: 2007/08*

NÁZEV TÉMATU:

IP Multimedia Subsystem

POKYNY PRO VYPRACOVÁNÍ:

Analyzujte budoucí požadavky telekomunikačních sítí s ohledem na konvergenci mobilních a pevných služeb a Voice over IP. Prostudujte možností nasazení IP Multimedia Subsystem jako řešení této problematiky. Seznamte se s open source platformou Open IMS Core.
Na základě teoretických poznatků realizujte experimentální systém na bázi Open IMS Core a ověřte jeho možnosti v praktickém nasazení. Diskutujte prakticky dosažené výsledky s teoretickými předpoklady.

DOPORUČENÁ LITERATURA:

[1] POIKSELKA, M., NIEMI, A., KHARTABIL, H., MAYER, G., The IMS: IP Multimedia Concepts and Services. Wiley, 2006. ISBN 978-0470019061
[2] CAMARILLO, G., GARCIA-MARTIN, M., The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Second Edition. Wiley, 2006. ISBN 978-0470018187

Termín zadání: 11.2.2008 *Termín odevzdání: 28.5.2008*

Vedoucí projektu: Ing. Petr Kovář

prof. Ing. Kamil Vrba, CSc.
předseda oborov~ rady

Licenční smlouva

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Martin Bendíček

Bytem: Klicperova 1/286, 73601,Havířov - Šumbark

Narozen/a (datum a místo): 1.12.1983, Havířov

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.

(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - □ disertační práce
   - □ diplomová práce
   - □ bakalářská práce
   - □ jiná práce, jejíž druh je specifikován jako .....

   (dále jen VŠKP nebo dílo)

Název VŠKP: IP Multimedia Subsystem

Vedoucí/ školitel VŠKP: Ing. Petr Kovář

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v*:

- □ tištěné formě – počet exemplářů 2
- □ elektronické formě – počet exemplářů 2

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - □ ihned po uzavření této smlouvy
   - □ 1 rok po uzavření této smlouvy
   - □ 3 roky po uzavření této smlouvy
   - □ 5 let po uzavření této smlouvy
   - □ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 28.5.2008

…….. ……
Nabyvatel Autor

ANOTACE

V této diplomové práci je rozebrána problematika architektury IMS. Práce popisuje její vznik a postupný vývoj, a také zde zmiňuje strukturu této architektury.

V textu práce je podrobně vysvětlena funkce nejdůležitější komponent dané architektury, mezi které patří Proxy-CSCF, Interrogating-CSCF, Serving-CSCF a HSS. Architektura IMS využívá pro sestavení, udržení a ukončení spojení protokol SIP a pro funkce managementu sítě protokol Diameter. Z toho důvodu jsou v této práci rozebrány také vlastnosti těchto dvou protokolů.

V další části tohoto dokumentu se pojednává o open-source platformě Open IMS Core a její instalaci. Prostředí Open IMS Core bylo vytvořeno pro testování technologie IMS v rámci interního projektu fraunhoferského institutu FOKUS v Berlíně. V současné době ji pro testování svých produktů a služeb v rámci IMS používá většina výrobců a poskytovatelů služeb.

V rámci této diplomové práce byla pomocí Open IMS Core vytvořena jednoduchá síť se třemi uživateli, založená na principu IMS. Právě vytvoření této sítě bylo hlavní náplní práce, a proto je v poslední kapitole nastíněn postup jejího vytváření. Nejprve je zde popsáno přidávání uživatelských záznamů do HSS, poté jsou zmíněny možnosti klientských programů a jejich nastavení. Nakonec jsou zde ukázány možnosti zachycení komunikace ve vytvořené síti. V této části kapitoly je rovněž pomocí stavových diagramů popsána registrace IMS terminálu na registrační server S-CSCF, a také proces sestavování a ukončování spojení.

Klíčová slova: Internet Protocol, IP Multimedia Subsystem, Home Subscriber Server, Call Session Control Function, Proxy-CSCF, Interrogating-CSCF, Serving-CSCF

ABSTRACT

This Master's thesis deals with IMS architecture problems. This thesis describes its creation and progress, also structure of this architecture is mentioned.

Text of this thesis describes function of the most important parts of the architecture, such as Proxy-CSCF, Interrogating-CSCF, Serving-CSCF and HSS. IMS architecture uses SIP protocol for establishment, maintenance and closing session and Diameter protocol for network management functions. That is why this thesis also describes characteristics of these two protocols.

Next part of the thesis is focused on Open-source platform Open IMS Core and its installation. Open IMS Core platform was developed for testing IMS technologies within an internal project at the Fraunhofer Institute FOKUS in Berlin. Nowadays it is used by major part of vendors and providers for testing their products and services.

As a main theme of this thesis, there was made a simple network, which consists of three users. This network is established on fundamentals of IMS and it was made in Open IMS Core. So in one chapter is outlined a process of its creation. There is described a process of adding a user list to database server HSS. In the next part there are mentioned some information about setup of virtual IMS terminal. Eventually there are shown some possibilities of how to catch traffic in the network. In this chapter there are also state diagrams, which describe a process of registration of IMS terminal and also a process of establishment, maintenance and closing session.

Keywords: Internet Protocol, IP Multimedia Subsystem, Home Subscriber Server, Call Session Control Function, Proxy-CSCF, Interrogating-CSCF, Serving-CSCF

Bibliografická citace díla

BENDÍČEK, M. *IP Multimedia Subsystem.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 55s. Vedoucí diplomové práce Ing. Petr Kovář.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma "IP Multimedia Subsystem" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne 28.5.2008 ……..

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Petru Kovářovi z Ústavu telekomunikací za velmi užitečnou metodickou pomoc, cenné rady a poskytnutí studijních materiálů pro vypracování této bakalářské práce.

V neposlední řadě bych také rád poděkoval své rodině, přátelům a známým, kteří mi, aniž by to mnohdy tušili, svojí vstřícností a tolerancí vytvořili skvělé podmínky nejen pro psaní této diplomové práce, ale i pro celé studium na vysoké škole VUT v Brně.

V Brně dne 28.5.2008 ……..

(podpis autora)

Seznam zkratek

3G	Third Generation
3GPP	Third Generation Partnership Project
AAA	Authentication Authorization Accounting
ACK	Acknowledgement
AMF	Authentication Management Field
AuC	Authentication Centre
AVP	Attribute Value Pairs
B2B	Back to Back
BGCF	Breakout Gateway Coltrol Function
CDR	Call Data Record
CSCF	Call Session Control Function
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
FHoSS	The FOKUS Home Subscriber Server
FOKUS	Fraunhofer Institut für Offene Kommunikationssysteme
GCC	GNU Compiler Collection
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Services
HSS	Home Subscriber Server
HTTP	Hyper Text Transfer Protocol
I-CSCF	Interrogating CSCF
ID	Identifier
IETF	Internet Engineering Task Force
IMEI	International Mobile Equipment Identity
IMPI	IP Multimedia Private Identity
IMPU	IP Multimedia Public User Identity
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IM-SSF	IP Multimedia Service Switching Function
IMSU	IMS Subscription
IP	Internet Protocol
IPsec	IP Security Protocol
ISIM	IP Multimedia Services Identity Module
JDK	Java Development Kit
LIA	Location Info Answer
LIR	Location Info Request
LTE	Long-Term Evolution
MAA	Multimedia Authentication Answer
MAR	Multimedia Authentication Request
MGCF	Media Gateway Controller Function
MGW	Media Gateway
MRFC	Media Resource Function Controler
MRFP	Media Resource Function Processor
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
NGN	Next Generation Network
OMA	Open Mobile Alliance

OP	Operator ID
OSA-SCS	Open Service Access Service Capability Server
P-CSCF	Proxy CSCF
PDF	Policy Decision Function
PLMN	Public Land Mobile Network
PoC	Push-to-talk over Cellular
PSI	Public Service Identifier
PSTN	Public Switching Telephone Network
QoS	Quality of Service
R&D	Research and Development
RSVP	ReSerVation Protocol
RTP	Real-time Transport Protocol
RTR	Response Time Reporter
SAA	Server Assignment Answer
SAR	Server Assignment Request
S-CSCF	Serving CSCF
SCTP	Stream Control Transmission Protocol
Ser_ims	Servers IMS
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SIP-AS	Session Initiation Protocol Application Server
SLF	Subscriber Location Function
SQN	Sequence Number
SRVLOC	Service Location Protcol
SVN	SubVersioN
TCP	Transmission Control Protocol
THIG	Topology Hiding Inter-network Gateway
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
UAA	User Authorization Answer
UAC	User Agent Client
UAR	User Authorization Request
UAS	User Agent Server
UDP	User Datagram Protocol
UE	User Equipment/Endpoint
UMTS	Universal Mobile Telecommunication System
UPSF	User Profile Server Function
URI	Uniform Resource Identifier
VLR	Visitor Location Register
VoIP	Voice over IP

Seznam obrázků

Seznam výpisů

Obsah

1 Úvod

V následujícím textu je rozebírána problematika architektury IMS (IP Multimedia Subsystem). Je popsán její vznik a postupný vývoj, a také je zde zmíněna struktura této architektury. Následně je podrobněji vysvětlena funkce jejích nejdůležitějších komponent, mezi které patří Proxy-CSCF (Call Session Control Function), Interrogating-CSCF, Serving-CSCF a HSS (Home Subscriber Server). Architektura IMS využívá pro sestavení, udržení a ukončení spojení protokol SIP (Session Initiation Protocol) a pro řízení přístupu protokol Diameter. Z toho důvodu jsou zde rozebrány vlastnosti také těchto dvou protokolů.

V další části této práce je zmíněna open source platforma Open IMS Core. Ta byla vytvořena pro testování technologie IMS v rámci interního projektu fraunhoferského institutu FOKUS (Fraunhofer Institut für Offene Kommunikationssysteme) v Berlíně. Nyní ji, pro testování svých produktů a služeb v rámci IMS, používá většina výrobců a poskytovatelů služeb. V rámci této kapitoly je zde také popsána instalace tohoto simulačního prostředí.

Hlavním cílem této diplomové práce bylo vytvoření jednoduché sítě, právě pomocí prostředí Open IMS Core. Proto je v poslední kapitole nastíněn postup jejího vytváření. Nejprve je popsáno přidávání uživatelských záznamů do HSS, poté jsou zmíněny možnosti klientských programů a jejich nastavení a nakonec zde jsou ukázány možnosti zachycení komunikace ve vytvořené síti. V této části kapitoly je rovněž pomocí stavových diagramů popsána registrace IMS terminálu na registrační server S-CSCF, a dále proces sestavování a ukončování spojení.

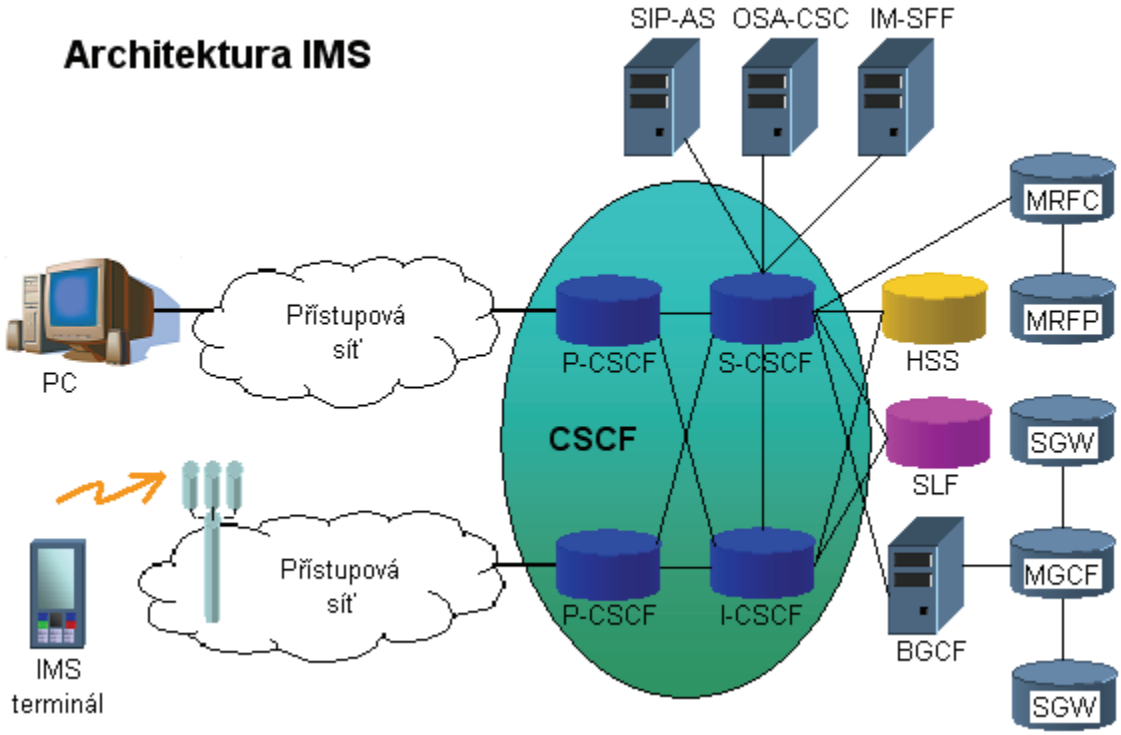
Jak vyplývá z následujících kapitol, IMS je platformě nezávislou architekturou. To je velkou výhodou, protože díky tomu může být aplikována na libovolnou síť využívající IP (Internet Protocol) protokol. IMS tak může být použito jak u bezdrátových sítí, tak i u stávajících pevných sítí, využívaných například pro VoIP (Voice over IP). Jedná se však o technologii posledních několika let, která se zatím teprve rozvíjí. Je tedy nemožné mluvit o jejím konkrétním využití v době, kdy si teprve hledá své místo na trhu i u poskytovatelů služeb. Právě zde je asi největší problém s jejím širším nasazením, neboť současní operátoři nechtějí investovat do technologie, pro kterou ještě u většiny účastnických zařízení neexistuje podpora. Jejího většího rozmachu se tedy zřejmě dočkáme až během dvou až tří následujících let. Během této doby nejspíš postupně nahradí stávající sítě s odděleným přenosem hlasu a dat. Nejpravděpodobnější způsob její první implementace však zřejmě bude do stávajících UMTS (Universal Mobile Telecommunication System) sítí.

2 IP Multimedia Subsystem

IMS (Internet Protocol Multimedia Subsystem) byl původně standardizován v roce 2003 organizací 3GPP (Third Generation Partnership Project) jako součást specifikace Release 5, a to jako nová vrstva služeb na vrcholu 3G (Third Generation) sítí, založených na IP. V Release 6, vydané v roce 2005, je definována spolupráce IMS se stávajícími okruhově spojovanými sítěmi, a také ostatními IP sítěmi. Také se zde objevuje snaha o přizpůsobení se nově vznikajícím PoC (Push-to-talk over Cellular) a službám, jejichž standardy jsou definovány OMA (Open Mobile Alliance). Následující specifikace Release 7 obsahuje snížení zpoždění, což přináší zlepšení pro real-timeové aplikace, jako je VoIP. Poslední specifikací je momentálně Release 8, která obsahuje 3GPP LTE (Long-Term Evolution) a SAE (Systém Architecture Evolution). Zahrnuje také tísňové služby (hasiči, policie, záchranná služba) založené na IMS.

IMS je globální, přístupově nezávislá architektura, jejíž standard je založen na protokolu IP. Hlavní myšlenkou IMS je sjednotit přenos hlasu a dat.

Proto, aby mohl být tento systém nasazen, musí být splněno několik požadavků. Zaprvé je zapotřebí, aby daná síť podporovala sestavování IP multimediálních spojení. Dále je třeba, aby zajišťovala QoS (Quality of Service) a spojení mezi internetem a sítěmi využívajícími přepínání okruhů. V neposlední řadě by zde měla existovat podpora pro roaming, rychlý vývoj nových služeb bez nutnosti jejich zahrnutí do standardu, a také přístup i ze sítí, které nevyužívají GPRS (General Packet Radio Services).



Obr. 1: Architektura IMS

Architektura IMS, kterou můžeme vidět na obrázku 1, je tvořena velkým množstvím komponent. Mezi její hlavní komponenty patří:

- **CSCF** (Call Session Control Function) – Jedná se o základní uzel IMS architektury, který se stará hlavně o signalizaci. Rozlišujeme tři typy. Proxy-CSCF, Interrogating-CSCF a Serving-CSCF. [3]

- **HSS** (Home Subscriber Server) – Tento prvek je obdobou domovského lokačního registru HLR (Home Location Register) v síti GSM (Global System for Mobile communications). Obsahuje všechny potřebné informace o uživatelích. [3]

Mezi další komponenty této architektury patří:

- **SLF** (Subscriber Location Function) – Tento prvek je využit pouze v případě výskytu většího množství HSS a používá se pro přiřazení správného HSS danému uživateli. [3]

- **aplikační servery SIP-AS** (Session Initiation Protocol Application Server) **OSA-SCS** (Open Service Access Service Capability Server) a **IM-SSF** (IP Multimedia Service Switching Function) – Aplikační servery umožňují přidávání multimediálních služeb do IMS. [3]

- **MRFC/P** (Media Resource Function Controler/Processor) – Tento prvek je vždy umístěn v domovské síti. MRFC vystupuje jako SIP User Agent, obsahuje SIP rozhraní pro komunikaci s S-CSCF a kontroluje zdroje v MRFP. Hlavní funkcí právě tohoto prvku je zajišťování podpory pro multimediální konference. [3]

- **MGCF** (Media Gateway Controller Function) – Tato komponenta je nutná pro spojení sítě IMS a veřejné telefonní sítě PSTN (Public Switching Telephone Network) a zahrnuje funkce s tím spojené. [3]

- **MGW** (Media Gateway) – Je koncovým prvkem pro nosný kanál z okruhově spínaných sítí a pro mediální toky z IP sítí. Jeho hlavními funkcemi jsou proto konverze, transkódování a signalizace. [3]

- **BGCF** (Breakout Gateway Coltrol Function) – Mezi hlavní funkce tohoto prvku patří vyjednávání o požadavcích s S-CSCF pro volání do PSTN. Pro SIP spojení do PSTN sítí vybírá příslušný MGCF, který pro tato spojení funguje jako výstupní bod. [3]

Komunikace v IMS využívá převážně dvou protokolů, a to protokolu SIP, který se používá pro sestavení, udržení a ukončení spojení a protokolu Diameter, jenž je využíván pro autentizaci, autorizaci a účtování. Pro vlastní přenos dat je pak využíván protokol IP.

Funkce hlavních komponent i vlastnosti protokolů SIP a Diameter budou podrobněji zmíněny v následujících, níže uvedených podkapitolách.

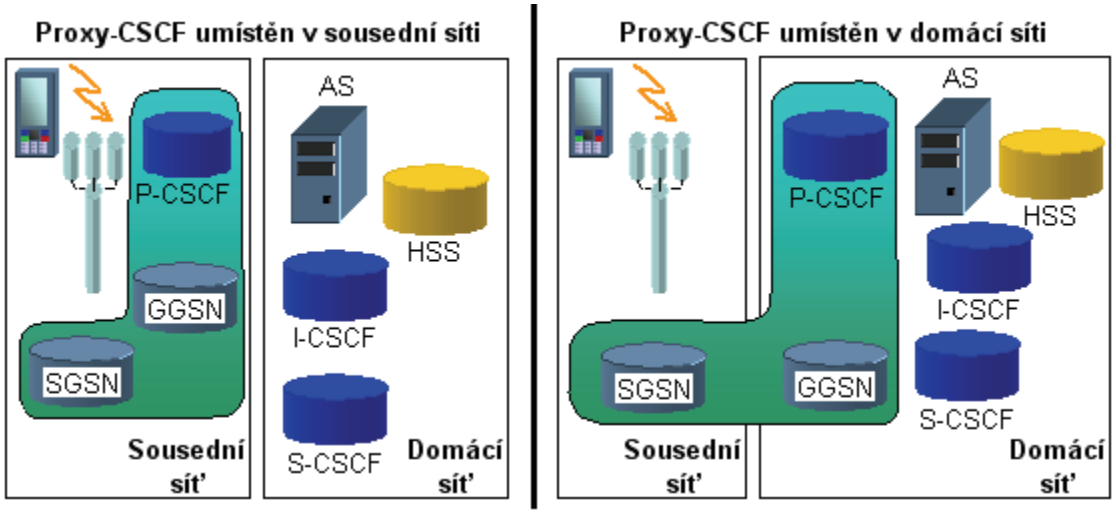
2.1 Call Session Control Function

První z komponent architektury IMS, která bude v této kapitole podrobněji zmíněna je CSCF. Zastává zde funkci SIP serveru, a dá se proto považovat za hlavní uzel této architektury. Hlavní funkcí tohoto prvku v IMS je zajišťovat signalizaci. Podle funkce můžeme rozlišit tři typy:

- P-CSCF (Proxy CSCF)
- I-CSCF (Interrogating CSCF)
- S-CSCF (Serving CSCF)

2.1.1 Proxy-CSCF

Proxy CSCF zastává funkci vstupního i výstupního SIP proxy serveru. Je tak vlastně prvním kontaktním bodem mezi IMS terminálem a sítí. Každému terminálu je přidělen během registrační fáze a po celou dobu registrace se nemění. P-CSCF může být umístěn buďto v domovské nebo v sousední síti. Jak je vidět na obrázku 2, v případě GPRS je proxy server umístěn v té síti, kde se nachází GGSN (Gateway GPRS Support Node) uzel. Ten slouží jako brána do vnější sítě.



Obr. 2: Umístění P-CSCF v domovské nebo sousední síti, podle polohy uzlu GGSN

P-CSCF sleduje tok signalizačních zpráv a může prověřovat každou zprávu. Provádí autentizaci uživatele a zároveň se stará o bezpečnost spojení. Tuto funkci zajišťuje pomocí IPsec (IP Security Protocol). Může tak předcházet IP spoofing útokům a útokům odchycením hesla. Zároveň chrání soukromí uživatelů. Ostatní uzly už poté spoléhají na autentizaci provedenou P-CSCF a vlastní kontrolu již neprovádějí. K dalším funkcím, které tento proxy server zajišťuje patří komprese a dekomprese SIP zpráv. Je tak snížen nárok na parametry používaných linek. P-CSCF může také zahrnovat funkci PDF (Policy Decision Function), která schvaluje mediální zdroje a zajišťuje tak QoS. Využívá se tak například pro správu šířky pásma. PDF však může existovat také jako samostatná funkce. K posledním funkcím, které jsou poskytovány P-CSCF patří schopnost identifikovat tísňová volání, a také generování CDR (Call Data Record). [3]

2.1.2 Interrogating-CSCF

Interrogating CSCF vystupuje jako kontaktní bod uvnitř sítě operátora. Obsluhuje tak všechna koncová spojení určená pro všechny SIP uživatele, kteří jsou zrovna registrováni v PLMN (Public Land Mobile Network). Při přijetí požadavku o registraci od účastnického koncového zařízení UE (User Equipment) získá I-CSCF z HSS jméno next hop S-CSCF. Z tohoto si můžeme lehce vyvodit, že hlavní funkcí této komponenty je právě přidělování S-CSCF jednotlivým spojením. Požadavky na HSS jsou posílány přes rozhraní Cx protokolu Diameter. Na takto přidělená S-CSCF jsou pak přímo směrovány příchozí požadavky. Stejně jako dříve zmíněný proxy server, také tento prvek generuje CDR. Až do specifikace Release 6 mohl být použit také pro skrytí vnitřní sítě před okolním světem. V tomto případě však byl I-CSCF nazýván THIG (Topology Hiding Inter-network Gateway). [3]

2.1.3 Serving-CSCF

Posledním ze tří typů CSCF je Serving CSCF, který zastává funkci centrálního bodu v oblasti signalizace IMS. Reprezentuje totiž registrátor běžných SIP sítí umístěných uvnitř prostředí 3GPP sítí. Přestože se jedná o SIP server, může vykonávat také kontrolní funkce. Vždy je umístěn v domovské síti. Nemá žádnou paměť pro ukládání informací o uživatelích, proto si musí potřebnou informaci vždy stáhnout z HSS. Ke komunikaci s HSS používá Cx a Dx rozhraní protokolu Diameter.

SIP server S-CSCF je zodpovědný za různé funkce, z nichž zřejmě tou nejdůležitější je registrace. Po registraci uživatele je totiž možné svázat IP adresu uživatelského koncového zařízení s přiřazenou SIP adresou. Mezi další funkce patří také uchovávání profilů nabízených služeb nebo rozhodovací funkce při směrování hovorů. V případě využití aplikačních serverů rozhoduje právě S-CSCF o tom, na který z nich bude SIP zpráva odeslána. Tento SIP server také zodpovídá za přenášení SIP volání do veřejné telefonní sítě PSTN. Tato volání jsou směrována na BGCF, která poté vybere vhodnou MGCF. Odsud už je pak vytvořeno spojení přímo do veřejné telefonní sítě. S-CSCF však nemusí být pouze jedno. Pro účely rozdělení zatížení, a tím pádem lepšího přístupu je někdy používáno více těchto zařízení. O tom, který SIP server pak bude použit rozhoduje I-CSCF. Poté dá o svém rozhodnutí vědět serveru HSS, a ten následně na základě došlého požadavku přidělí příslušný S-CSCF server. Stejně jako ostatní CSCF, také Serving-CSCF generuje CDR. [3]

2.2 Home Subscriber Server

HSS, někdy také označovaný jako UPSF (User Profile Server Function) je obdobou domovského lokačního registru HLR v GSM. Sdružuje však také funkce a informace, které bývají implementovány v ostatních blocích architektury GSM. Například autentizačního centra AuC (Authentication Centre) nebo návštěvnického lokačního registru VLR (Visitor Location Register). Hlavní funkcí HSS je uchovávat informace o uživatelích dané sítě. Jedná se tedy o hlavní databázi architektury IMS, vytvářející podporu pro všechny její komponenty, které se zrovna podílejí na jednotlivých spojeních.

V databázi HSS jsou uloženy všechny uživatelské profily. Ty jsou poté na požádání poskytovány SIP serveru S-CSCF, který je uchovávat nemůže. Jak již bylo popsáno výše, na základě požadavků od I-CSCF, přiděluje HSS různým uživatelům jednotlivé S-CSCF. Tato databáze dále zahrnuje také bezpečnostní informace, využívané při autentizaci a autorizaci uživatelů. Může však také poskytnout informace o jejich aktuální poloze.

Klasické 3GPP sítě používají identifikátory IMSI (International Mobile Subscriber Identity), TMSI (Temporary Mobile Subscriber Identity), IMEI (International Mobile Equipment Identity) a MSISDN (Mobile Subscriber Integrated Services Digital Network Number). IMSI je unikátní účastnický identifikátor, který je uložen na SIM (Subscriber Identity Module) kartě telefonu. Aby nemusel toto číslo posílat přes rádiové rozhraní při každé žádosti o nějakou službu, přiřadí mu systém dočasný identifikátor TMSI. Číslo TMSI je uloženo na SIM kartě a v registru VLR mobilní ústředny. Pokud se účastník s mobilní stanicí přesune na území pod kontrolou jiné ústředny, je mu novou ústřednou zasláno nové číslo TMSI a předchozí číslo je zrušeno, jak v SIM kartě, tak i ve VLR předchozí ústředny. Pouze v případech, kdy se účastník hlásí do systému po zapnutí mobilní stanice, zasílá MS do ústředny identifikátor IMSI. Ihned poté je však do mobilní stanice zaslána prozatímní identifikátor TMSI, pomocí které již může účastník se systémem dále komunikovat. Zatímco identifikátory IMSI a TMSI jsou použity pro identifikaci uživatele, identifikátor IMEI slouží pro identifikaci telefonního přístroje. Tento identifikátor je uložen v paměti mobilního přístroje. Poslední identifikátor MSISDN je telefonní číslo uživatele. [8]

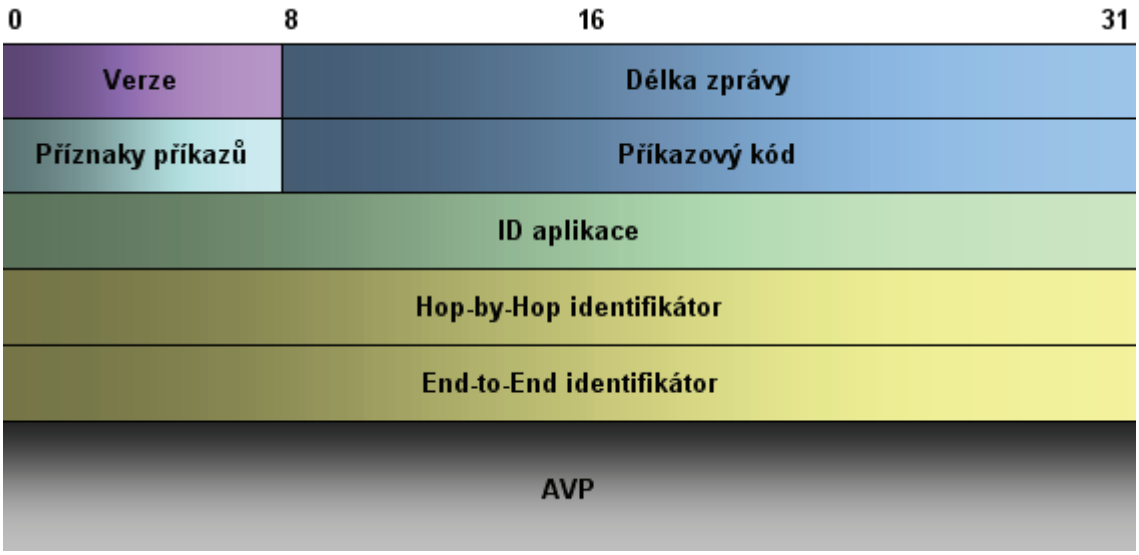
Architektura IMS však vyžaduje ještě další identifikátory, a to IMPI (IP Multimedia Private Identity) a IMPU (IP Multimedia Public User Identity). Jedná se o takzvané URI (Uniform Resource Identifier) identifikátory. Ty mohou mít formu buďto posloupnosti čísel, pak se jedná o tel-uri (*tel:+1-555-123-4567*), případně posloupnosti alfanumerických znaků. V tom případě mluvíme o sip-uri (*sip:john.doe@example.com*). V IMS jsou všechny výše uvedené identifikátory obsaženy právě v databázi HSS.

V této kapitole je ještě třeba zmínit další komponentu architektury IMS, která má vztah k HSS. Jedná se o prvek SLF. Tato jednoduchá databáze je použita v případech, kdy existuje více než jeden domovský server HSS. V těchto situacích pak mapuje uživatelské adresy do správného HSS. Pomocí dotazů uzlu SLF, jejichž vstupem je uživatelská adresa, jsou získány adresy konkrétních HSS. Ty už pak obsahují všechny potřebné informace, vztahující se k uživatelům. HSS a SLF spolu komunikují pomocí komunikačního protokolu Diameter. [3]

2.3 Protokol Diameter

Protokol Diameter je takzvaný AAA (Authentication Authorization Accounting) protokol. Z této zkratky vyplývá, že je využíván převážně pro funkce managementu sítě, jako jsou autentizace a autorizace uživatelů, a také pro zpoplatnění služeb. Autentizace znamená proces ověření identity uživatele, neboli subjektu. Ověří se tedy jestli je subjekt opravdu tím, za koho se vydává. Poté následuje proces autorizace, kdy dochází k rozhodování, zda bude subjektu žádajícímu o přístup ke zdroji, neboli objektu, tento skutečně povolen. Jedná se tedy o ověření zda má daný subjekt, již s ověřenou identitou, udělena práva na úkony, o které žádá. Rozhodovací komponentou je v případě IMS blok I-CSCF, který rozhoduje na základě informací uložených v centrální databázi HSS. U funkce zpoplatnění (Accounting) se jedná o sbírání informací o využívání zdroje subjektem pro účely plánování, stanovení ceny, kontroly účtů a fakturace. Diameter používá takzvanou hop-by-hop bezpečnost, což znamená, že komunikace pomocí tohoto protokolu je zabezpečována mezi každými dvěma uzly. Tento protokol může pracovat jak lokálně tak i v roamingu.

Protokol Diameter byl vyvinut sdružením IETF (Internet Engineering Task Force) jako následovník protokolu Radius. Jeho poslední verze pochází ze září roku 2003. Název tohoto protokolu byl zvolen podle anglického významu názvu svého předchůdce. Slovo radius totiž znamená poloměr. Pro jeho následovníka byl tedy zvolen název Diameter, což v angličtině znamená průměr, dvojnásobek poloměru. Protokol Diameter sice rozšiřuje protokol Radius, není s ním však zpětně kompatibilní. Podporuje sice přenos Radiusu, je však zapotřebí, aby byly zprávy protokolu Radius rozšířeny o atributy potřebné pro protokol Diameter.



Obr. 3: Formát paketu protokolu Diameter

V následujícím textu budou uvedeny hlavní rozdíly mezi protokoly Radius a Diameter. Prvním takovým rozdílem je, že zatímco Radius používá nespolehlivý protokol UDP (User Datagram Protocol), Diameter používá buďto protokol TCP (Transmission Control Protocol) nebo SCTP (Stream Control Transmission Protocol). Díky tomu může použít zabezpečení na transportní vrstvě, a to IPsec nebo TLS (Transport Layer Security). Diameter také nabízí větší adresní prostor pro AVPs (Attribute Value Pairs) a 8bitové identifikátory Radiusu nahrazuje 32bitovými. Jedná se o protokol typu klient – server, který používá komunikaci typu Request/Answer. Pro vyhledání okolních uzlů může být použita buď manuální konfigurace pomocí SRVLOC (Service Location Protocol) nebo dynamické vyhledávání pomocí DNS (Domain Name Server). Dalším vylepšením u tohoto protokolu je, že oznamuje chyby. Také má lepší podporu roamingu a je snadněji rozšiřitelný o nové atributy a příkazy. [3] Formát paketu protokolu Diameter je vidět na obrázku 3.

2.4 Session Initiation Protocol

Protokol SIP byl vytvořen v roce 1999 jako RFC 2543. Od té doby se o jeho další rozvoj stará IETF a SIP Forum.

Protokol SIP je jednoduchý obecný protokol pro navazování interaktivních komunikačních relací mezi dvěma či více koncovými zařízeními. Tato zařízení si je sám schopen vyhledat. SIP se však používá nejen pro inicializaci, ale také pro modifikaci a ukončování interaktivních relací. Obecně řečeno je tedy SIP využit pro komunikaci v reálném čase. SIP patří mezi protokoly aplikační vrstvy a většinou pracuje nad transportním protokolem UDP. Není to však pravidlem a může využít i TCP protokol.

SIP poskytuje služby pro lokalizaci uživatele, což je určení koncového systému pro danou komunikaci. Dále se stará o navázání spojení, tedy stanovení parametrů pro volající a volanou stranu. Lze pomocí něj také zjišťovat dostupnost volané strany a sledování její přítomnosti. Pomocí tohoto protokolu je zároveň umožněno určování média a jeho parametrů. Naproti tomu však SIP nemůže provádět management interaktivních relací po jejich navázání. Také neumí zajistit požadovanou kvalitu služeb QoS, protože neumí upřednostňovat jeden provoz před druhým, ani rezervovat potřebné síťové prostředky. Může však spolupracovat s protokoly, které mohou QoS zajistit. Mezi takové protokoly patří například RSVP (ReSerVation Protocol). SIP není určen ani pro přenosy velkých objemů dat, jako například protokol HTTP (Hyper Text Transfer Protocol). Namísto toho přenáší jen malé objemy dat potřebné pro navázání interaktivních relací nebo krátké textové zprávy.

SIP pracuje na principu klient-server. Můžeme tedy rozlišit dva typy koncových zařízení, klienta UAC – User Agent Client a server UAS – User Agent Server. Mezi klienty patří uživatelská zařízení jako SIP telefony nebo PC s klientským softwarem. Mezi servery řadíme brány do jiných sítí.

Servery mohou fungovat jako proxy servery, kdy zastupují klienty při předávání požadavků SIP na další server. Nebo jako redirect servery, kdy klienta informují o dalším skoku v síti, kam se má zpráva poslat. Klient nebo proxy server pak kontaktuje doporučené zařízení sám. Registraci momentálního umístění klientů zpracovávají registrační servery, které informace o uživatelských agentech aktualizují v serveru umístění nebo databázi. Z výše uvedeného popisu bloků architektury IMS můžeme vidět, že funkci SIP proxy serveru zastává P-CSCF, funkci redirect serveru I-CSCF a jako registrační server zde vystupuje S-CSCF. Potřebnou databázi obsahuje HSS.

Pro směrování pomocí protokolu SIP se používají takzvané SIP URL (Unique Resource Locator). Ty mají obecně tvar *sip: user@domain.* Na místě uživatele však může být použito i telefonní číslo. Například adresa počítače uživatele xbendi00 v doméně vutbr.cz by měla následující tvar *sip:xbendi@vutbr.cz*. Při použití telefonního čísla místo jména uživatele by se její tvar změnil na *sip:+420-123456789@vutbr.cz.* Toto telefonní číslo by pak bylo dosažitelné prostřednictvím brány. Díky tomu, že SIP podporuje jak webovou adresaci, tak telefonní čísla, může komunikace bez problémů přecházet mezi telefonními a datovými sítěmi.

Základní výhodou protokolu SIP je jeho jednoduchost, díky které potřebuje jen malý počet zpráv. To umožňuje aplikacím nezávislost na síťové vrstvě, která jim umožní pracovat nad různými typy sítí. Uživatele lze tedy v síti lokalizovat a komunikovat s nimi bez ohledu na použitá koncová zařízení. [7]

3 Open IMS Core

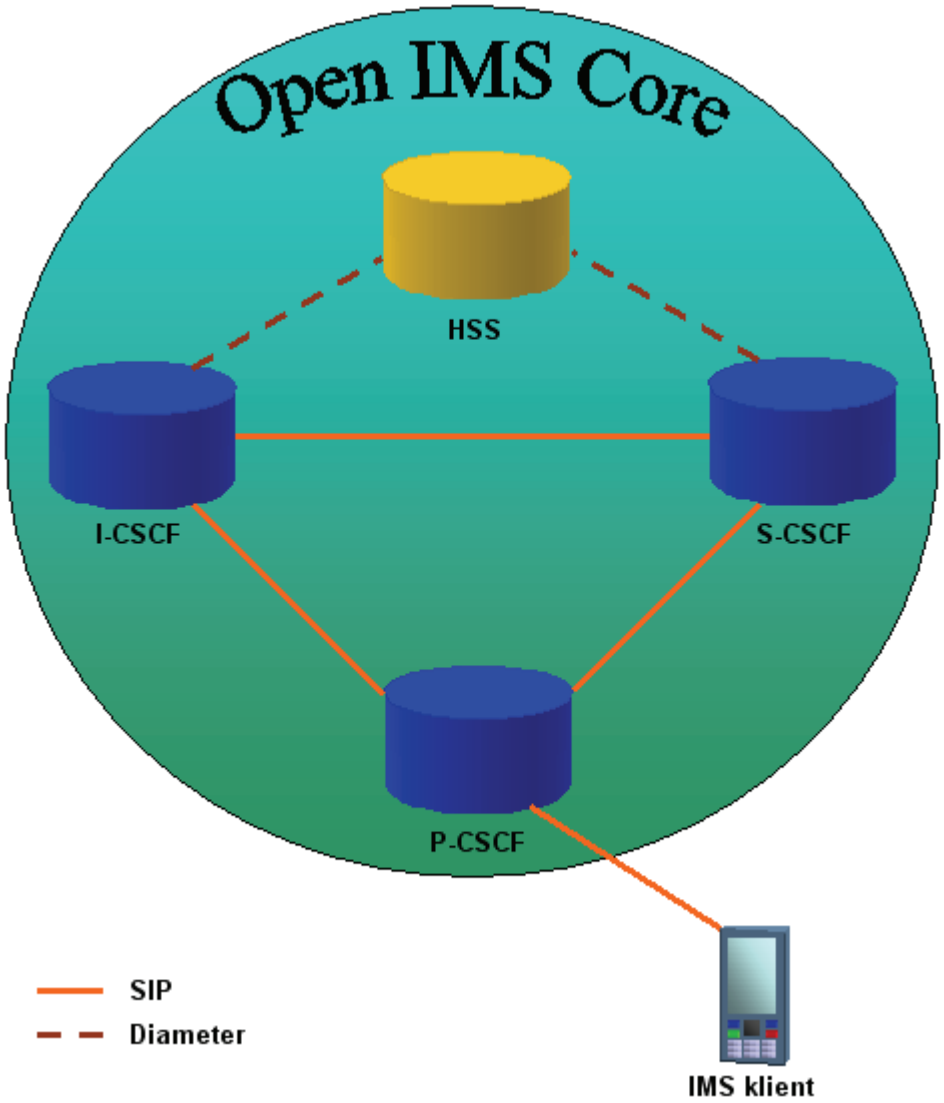
Open IMS Core byl vytvořen v rámci interního projektu fraunhoferského institutu FOKUS v Berlíně, jako open source implementace základních prvků architektury IMS. V posledních dvou letech se tento projekt úspěšně rozběhl a z Open IMS Core se stalo základní testovací prostředí v oblasti IMS. V současné době ho tak využívá většina výrobců i operátorů, jak pevných, tak mobilních služeb, pro zkoušení, ladění a dokazování vlastností jimi nabízených služeb. Pro svou flexibilitu a výkon se Open IMS Core vyhledávaným nástrojem v mnoha národních i mezinárodních R&D (Research and Development) projektech.

Kvůli decentralizované povaze IMS spotřebovává směrování velkou část celkového výkonu systému. Koncová zařízení si totiž během sestavování a ukončování svých spojení vyměňují a zpracovávají velké množství SIP zpráv. SIP proxy servery musí ovládat veškerý provoz s co nejmenším zpožděním, aby zajistily co nejmenší celkový čas pro nastavení spojení. Funkce, které to zajišťují, jsou obsaženy v CSCF (P-CSCF, I-CSCF, S-CSCF). Z toho důvodu jsou hlavními částmi Open IMS Coru. Jednou z částí CSCF jsou registrační služby, které udržují cesty k uživatelům, jejich předvolby a data. Používají se pro zjištění polohy účastníka, nastavení jeho vlastních služeb a pro ochranu IMS jádra před možnými útoky. Tyto typy registračních služeb jsou částí jádra sítě a umožňují získávat správné cílové adresy pro směrování zpráv. V neposlední řadě musí být IMS jádro schopno v určitých případech, jako jsou vynucené ukončení hovoru nebo B2B (Back to Back) hovory, vystupovat v signalizačním procesu jako koncové zařízení. Protože správná funkčnost CSCF spoléhá na informace o specifických službách uživatelských profilů a vyhledávací funkce uživatelsky definovaných CSCF, byl do Open IMS Core přidán také HSS.

Současné nebo Group-list Management služby, stejně jako ostatní služby, jsou implementovány do aplikačních serverů. Ty se mohou aktivně účastnit signalizačních i datových přenosů, nejsou však považovány za součást jádra IMS. Stejně tak do jeho jádra nelze zahrnout koncová zařízení UE (User Endpoints). Ta generují signálový provoz.

Open IMS Core projekt byl v listopadu roku 2006 připojen k Open Source projektu, který je prvotně zaměřen na všechny oblasti týkající se výzkumu a vývoje NGN (Next Generation Network) služeb a IMS testů pro zavedení hlasu do oblasti Open Source. [1, 2]

Open IMS Core lze instalovat pouze na operační systémy postavené na linuxovém jádru. Postup jeho instalace bude popsán v následující kapitole.



Obr. 4: Základní prvky Open IMS Core

3.1 Postup instalace

Open IMS Core se skládá ze dvou hlavních částí. První z nich, FHoSS (The FOKUS Home Subscriber Server), simuluje funkce domovského uživatelského serveru HSS. Zatímco druhá část, pojmenovaná Ser_ims (Servers IMS), obsahuje kód pro simulace všech tří serverů CSCF. Tedy P-CSCF, I-CSCF a S-CSCF.

Při instalaci byl použit návod, který je umístěn na stránkách http://www.openimscore.org. V následujícím textu bude tento instalační postup popsán. Avšak dříve než se přistoupí k vlastní instalaci, je zapotřebí zkontrolovat, zda počítač, na kterém se chystáme instalaci provést, splňuje všechny hardwarové i softwarové požadavky. Jelikož tato instalace není na hardwarové požadavky nijak náročná a splňuje je prakticky každý momentálně dostupný počítač, budou zde podrobněji zmíněny pouze požadavky na software.

Aby mohl být Open IMS Core nainstalován, je zapotřebí mít na disku alespoň 100 MB volného místa. Dále je zapotřebí mít nainstalováno GCC (GNU Compiler Collection) 3 nebo 4, a JDK (Java Development Kit) minimálně verze 1.5. Další nezbytnou podmínkou je nainstalovaný a spuštěný databázový systém MySQL. Dále je nutné nainstalovat balíčky libxml2 a libmysql, oba ve verzi development. Jádro použitého linuxového systému by mělo být alespoň verze 2.6. Pro možnost využití IPsec a TLS bezpečnosti je třeba mít ipsec nástroje a openssl. V neposlední řadě je třeba mít nainstalován a spuštěn DNS server Bind.

Návod uvedený na výše zmíněných stránkách obsahuje také zkrácenou verzi návodu, která obsahuje pouze příkazy pro vytvoření potřebných adresářů na disku, stažení zdrojového kódu aplikace a jeho zkompilování. Tento návod je vhodný v případě, že už byla instalace někdy provedena. Může nám pomoct připomenout si potřebné příkazy, případně z něj můžeme vytvořit instalační skript. Pro první instalaci, která zde byla prováděna, se však nehodí, protože předpokládá, že všechny potřebné podpůrné aplikace jsou již nainstalovány a nakonfigurovány. Proto neobsahuje tento zkrácený postup jejich instalaci ani nastavení. Tento postup rychlé instalace můžeme vidět ve výpisu 1.

```
mkdir /opt/OpenIMSCore    // Vytvoření složky
cd /opt/OpenIMSCore       // Přepnutí se do vytvořené složky

mkdir FHoSS

// Stáhnutí zdrojového kódu ze zadané adresy pomocí Subversion
svn checkout
http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk FHoSS

mkdir ser_ims
svn checkout
http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk ser_ims

cd FHoSS
ant compile deploy        // Kompilace zdrojového kódu FHoSS
cd ..                     // Změna adresáře o úroveň výš

cd ser_ims
make install-libs all     // Instalace knihoven ser_ims
cd ..
                          // Nastavení databází
mysql -u root -p < FHoSS/scripts/hss_db.sql
mysql -u root -p < FHoSS/scripts/userdata.sql

mysql -u root -p < ser_ims/cfg/icscf.sql

cp ser_ims/cfg/*.cfg .
cp ser_ims/cfg/*.xml .
cp ser_ims/cfg/*.sh .
```

Výpis 1: Příklad rychlého instalačního postupu

Jak již bylo zmíněno, v našem případě byl zvolen kompletní postup se všemi nastaveními. Prvním krokem je stažení zdrojového kódu Open IMS Core na disk počítače. Stažení zdrojového kódu bylo provedeno pomocí SVN (SubVersioN). To však bylo nejprve zapotřebí nainstalovat. Po jeho instalaci byl na disku počítače vytvořen adresář */opt/OpenIMSCore*, který je hlavním adresářem celého programu. Poté je třeba změnit pomocí příkazu chown vlastníka souboru na sebe. Následně by měly být v tomto hlavním adresáři vytvořeny podadresáře pro jednotlivé části Open IMS Core. Názvy podadresářů se shodují s názvy těchto části. Nejprve je tedy vytvořen adresář ser_ims a poté FHoSS. Následně už může být stažena poslední verze zdrojového kódu ze subversion serveru skupiny BerliOS. Popsaný postup prvního kroku lze provést pomocí následujících příkazů.

```
// Přidání balíčku subversion
sudo apt-get install subversion

// Vytvoření hlavního adresáře
sudo mkdir /opt/OpenIMSCore/

// Změna vlastníka souboru (username se nahradí jménem uživatele)
sudo chown -R username /opt/OpenIMSCore/

// Přepnutí do hlavního adresáře aplikace Open IMS Core
cd /opt/OpenIMSCore

// Vytvoření podadresářů ser_ims a FHoSS
mkdir ser_ims
mkdir FHoSS

// Stažení zdrojového kódu pro ser_ims a FHoSS ze subversion serveru
svn checkout
http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk ser_ims
svn checkout
http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk FHoSS
```

Výpis 2: První krok instalačního postupu

V druhém kroku je zapotřebí nainstalovat potřebné balíčky podpůrných aplikací, které Open IMS Core vyžaduje. Konkrétně se jedná o balíčky sun-java-jdk, mysql-server, libmysqlclient15-dev, libxml2-dev, bind, ant, flex a bison. Tyto balíčky mohou být, u zvoleného operačního systému Ubuntu 7.10 Gutsy Gibbon, nainstalovány pomocí nástroje Synaptic package manager nebo z příkazové řádky. V této práci byla vybrána instalace z příkazové řádky pomocí následujícího příkazu.

```
// Instalace potřebných balíčků
sudo apt-get install sun-java6-jdk mysql-server libmysqlclient15-dev
libxml2 libxml2-dev bind9 ant flex bison
```

Výpis 3: Druhý krok instalačního postupu

Během třetího kroku je třeba nastavit DHCP (Dynamic Host Configuration Protocol) a DNS server. Protože v našem případě byl DNS server provozován přímo na počítači bylo zapotřebí odkomentovat v konfiguračním souboru */etc/dhcp3/dhclient.conf* řádek *prepend domain_name_servers 127.0.0.1;.* Tím dojde k nastavení localhosta. Úprava konfiguračního souboru byla provedena v textovém editoru vi. Pak se musí zkopírovat open-ims DNS soubor do adresáře DNS serveru Bind a nastavit mu příslušná práva. Následně je ještě třeba upravit konfigurační soubor Bindu */etc/bind/named.conf.* Do tohoto souboru je nutno vložit následující řádky.

```
zone "open-ims.test" IN {
        type master;
        file "pri/open-ims.dnszone";
        notify no;
};
```

Pak už pouze stačí DNS server Bind restartovat a ověřit správné nastavení příkazem *ping*. Popsaný postup byl proveden pomocí příkazů, které můžeme vidět ve výpisu 4.

```
// Zkopírování open-ims.dnszone souboru do adresáře DNS serveru Bind
cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone /var/bind/pri/

// Nastavení práv zkopírovanému souboru
chown -R named:named /var/bind/pri/open-ims.dnszone

// Restartování DNS serveru Bind
/etc/init.d/named restart

// Kontrola nastavení příkazem ping
ping pcscf.open-ims.test
```

Výpis 4: Třetí krok instalačního postupu

Ve čtvrtém kroku byly nastaveny databáze potřebné pro správné fungování Open IMS Core. V ser_ims byla nastavena databáze pro I-CSCF. V adresáři FHoSS bylo nutno nastavit databázi uživatelských dat a databázi pro HSS. Následují příkazy opět prezentují výše popsaný postup.

```
// Nastavení databáze pro I-CSCF
mysql -uroot -p < ser_ims/cfg/icscf.sql

// Nastavení databáze pro HSS
mysql -uroot -p < FHoSS/scripts/hssdb.sql

// Nastavení databáze uživatelských dat
mysql -uroot -p < FHoSS/scripts/userdata.sql
```

Výpis 5: Čtvrtý krok instalačního postupu

V pátém kroku se provádí kompilace. Nejprve je zkompilován zdrojový kód pro CSCF servery. Proto se kompilace provádí v adresáři */opt/OpenIMSCore/ser_ims.* Následně je zapotřebí nastavit proměnnou JAVA_HOME tak, aby byla správně nastavena cesta k nástrojům java. Pak už můžeme provést i kompilaci zdrojového kódu pro HSS. Tu však provádíme v adresáři */opt/OpenIMSCore/FHoSS.* V případě, že počítač není připojen k internetu, případně je za firewallem může nastat problém s kompilací zdrojového kódu pro HSS. Jelikož byl použitý počítač připojen až za firewallem nastal právě výše zmíněný problém. Řešením je stažení souboru z adresy http://www.w3.org/2001/xml.xsd a jeho nakopírování do adresáře */opt/OpenIMSCore/FHoSS/xsd/*. Poté již proběhla kompilace bez problémů. Zmíněný postup je prezentován ve výpisu 6.

```
// Přepnutí do adresáře ser_ims
cd ser_ims

// Kompilace zdrojového kódu pro CSCF
sudo make install-libs all

// Změna adresáře o úroveň výš
cd ..

// Nastavení proměnné JAVA_HOME
export JAVA_HOME=”/usr/lib/jvm/java-1.6.0-sun”

// Stažení a zkopírování souboru z http://www.w3.org/2001/xml.xsd
schemaLocation="http://www.w3.org/2001/xml.xsd"/> to
schemaLocation="file:///opt/OpenIMSCore/FHoSS/xsd/xml.xsd"/>

// Přepnutí do adresáře FHoSS
cd FHoSS

// Kompilace zdrojového kódu pro HSS
ant compile deploy

// Změna adresáře o úroveň výš
cd ..
```

Výpis 6: Pátý krok instalačního postupu

V šestém kroku se konečně dostáváme ke spuštění Open IMS Core. Nejdříve je třeba zkopírovat konfigurační soubory do jejich Open IMS Core adresářů. Poté již můžeme přejít k vlastnímu spuštění. Nejprve spustíme jednotlivé servery CSCF. Každý v novém terminálovém okně. Nové terminálové okno můžeme otevřít pomocí klávesové zkratky Ctrl+Shift+T.

Poté se přepneme do adresáře *FHoSS/deploy* a spustíme HSS. Zde však může opět nastat problém s proměnnou JAVA_HOME. Aplikace hlásí chybně zadanou cestu i když je nastavena správně a potřebné soubory jsou umístěny přesně tam kde mají být. Nastíněný postup spouštění Open IMS Core byl proveden pomocí následujících příkazů. [4, 5, 6]

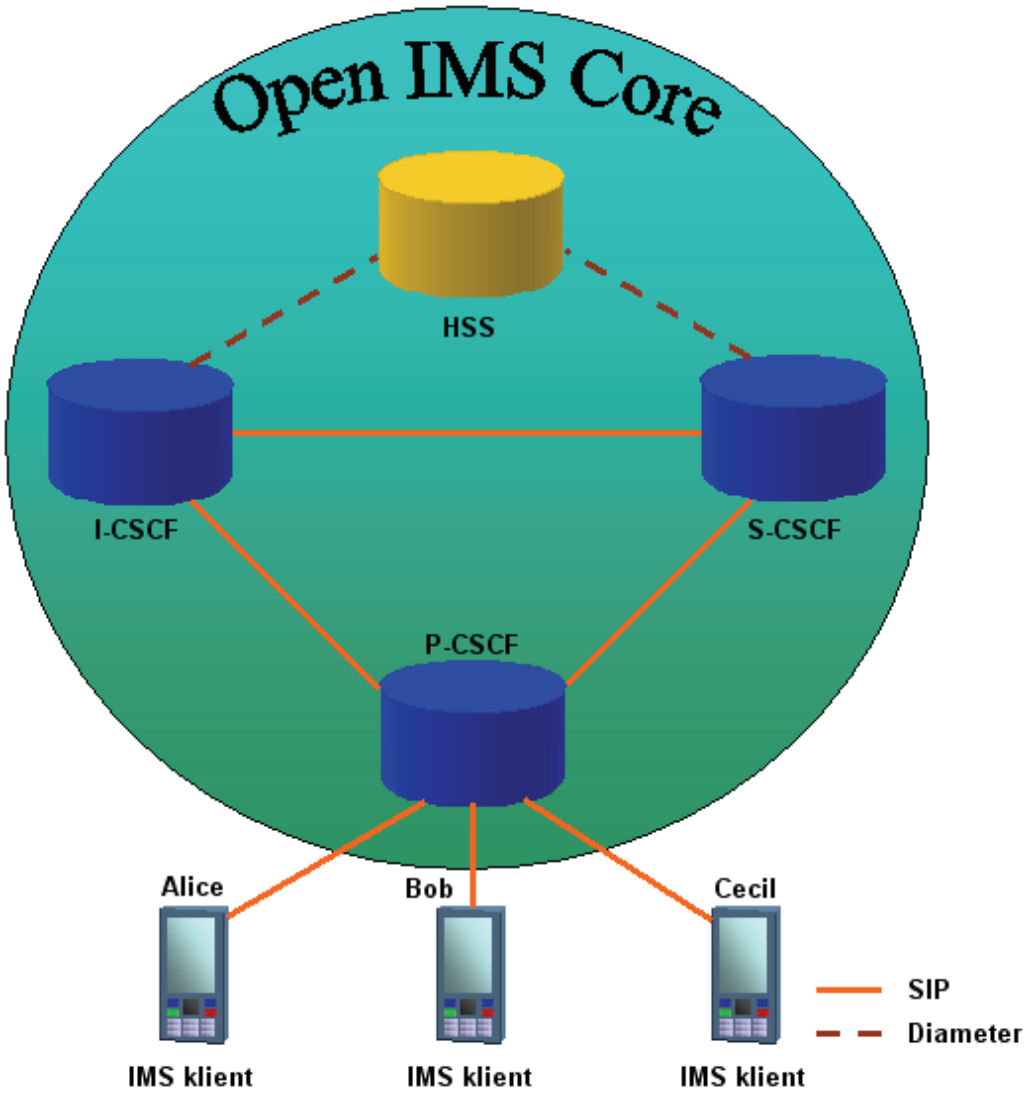
```
// Zkopírování konfiguračních souborů do jejich Open IMS Core adresářů
cp /opt/OpenIMSCore/ser_ims/cfg/* /opt/OpenIMSCore/

// Spuštění jednotlivých CSCF serverů (každý příkaz je zadán v novém
terminálovém okně)
./pcscf.sh
Ctrl+Shift+T
./icscf.sh
Ctrl+Shift+T
./scscf.sh

// Otevření nového terminálového okna
Ctrl+Shift+T

// Přepnutí do adresáře FHoSS/deploy
cd FHoSS/deploy/

// Spuštění HSS
./startup
```

Výpis 7: Spuštění Open IMS Core

4 Realizace

Cílem této diplomové práce bylo vytvořit jednoduchou síť na principu IMS za pomoci simulačního prostředí Open IMS Core. Nejdříve však bylo zapotřebí odstranit problémy, které vznikly ještě při vlastní instalaci simulačního prostředí na linuxový operační systém Ubuntu 7.10 Gutsy Gibbon. Tyto chyby se však ani po konzultacích s fraunhoferským institutem FOKUS ani jinými uživateli Open IMS Core nepodařilo odstranit. Proto bylo pro tuto práci využito ISO určené pro VMware Player, kde je Open IMS Core již naistalován, tentokrát na linuxovém operačním systému Gentoo.

Pomocí tohoto image byla vytvořena síť na principu technologie IMS, která obsahuje tři uživatele Alice, Bob a Cecil, kteří spolu mohou komunikovat a posílat si zprávy. Open IMS Core zde simuluje skutečnou IMS síť a její prvky. Vlastní komunikace pak probíhá pomocí nainstalovaných klientů. Provoz v této síti je sledován a zachytáván pomocí programu Wireshark.



Obr. 5: Struktura vytvořené sítě

4.1 Odstraňování problémů

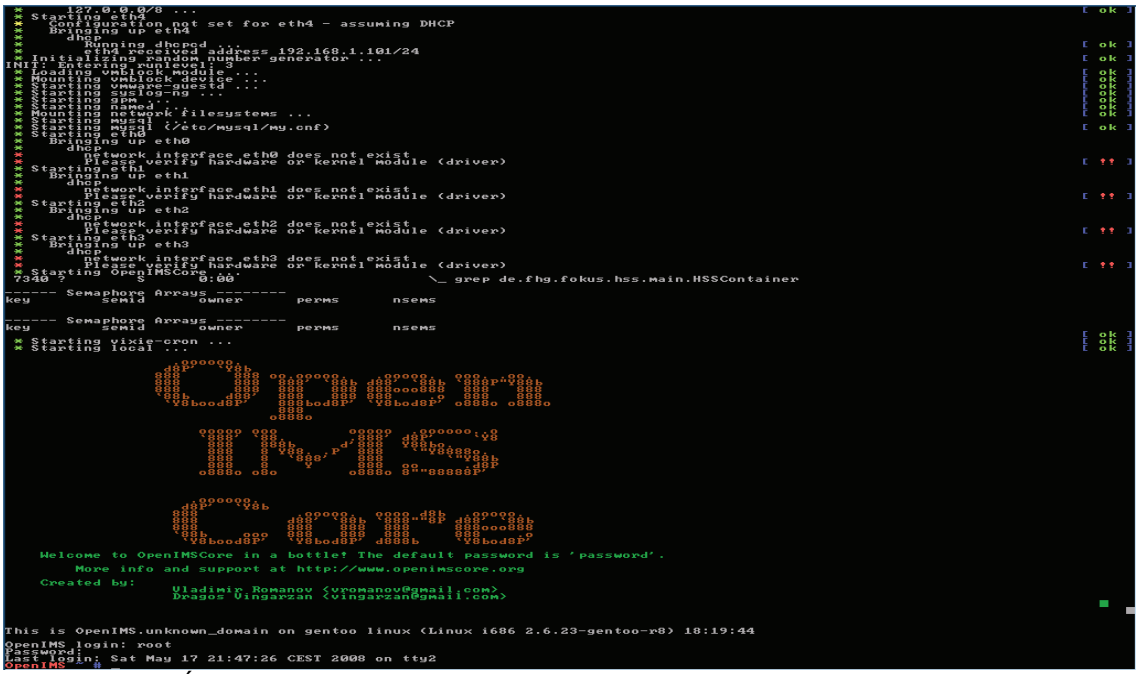
Jak bylo zmíněno na konci předchozí kapitoly, při pokusu o spuštění HSS hlásila aplikace chybně zadanou cestu k nástrojům java. Jedná se o známou chybu, pro jejíž vyřešení radí instalační manuál znovu předefinovat proměnnou JAVA_HOME. Tento krok však v našem případě nepřinesl žádné zlepšení. Proto jsme se tuto chybu pokusili odstranit nahrazením balíčku java jdk 1.5 balíčkem java6 jdk. Kontrola, zda proběhla náhrada balíčku bez problémů, byla provedena pomocí příkazu *java –version*. Podle reakce na tento příkaz, který můžeme vidět ve výpisu 8, je zřejmé, že náhrada balíčku proběhla v pořádku a v operačním systému zůstala pouze nová verze javy. Přesto, že by měla pro správný běh aplikace stačit už java verze 1.5, vedl tento krok k úspěšnému spuštění HSS.

```
java -version

java version "1.6.0_03"
Java(TM) SE Runtime Environment (build 1.6.0_03-b05)
Java HotSpot(TM) Client VM (build 1.6.0_03-b05, mixed mode, sharing)
```

Výpis 8: Kontrola verze javy po její náhradě vyšší verzí

Následně se však objevily další chyby, tentokrát u S-CSCF a I-CSCF. SIP server S-CSCF hlásí, stejně jako I-CSCF, chybu spojení s HSS. Kontaktní bod I-CSCF tak není schopen inicializovat databázi v HSS, a z toho důvodu nemůže uživateli ani přiřadit S-CSCF. Tento problém jsme se pokusili odstranit pomocí opětovného nastavení DNS serveru.

Bohužel, jak již bylo řečeno výše, ani po konzultacích s jinými uživateli Open IMS Core ani s fraunhoferským institutem FOKUS se tento problém nepodařilo odstranit. Proto bylo pro tuto diplomovou práci použito ISO vytvořené pro využití ve VMware Playeru, kde je Open IMS Core již nainstalován, tentokrát na linuxovém operačním systému Gentoo.



Obr. 6: Úvodní obrazovka po spuštění OS Gentoo s nainstalovaným Open IMS Core

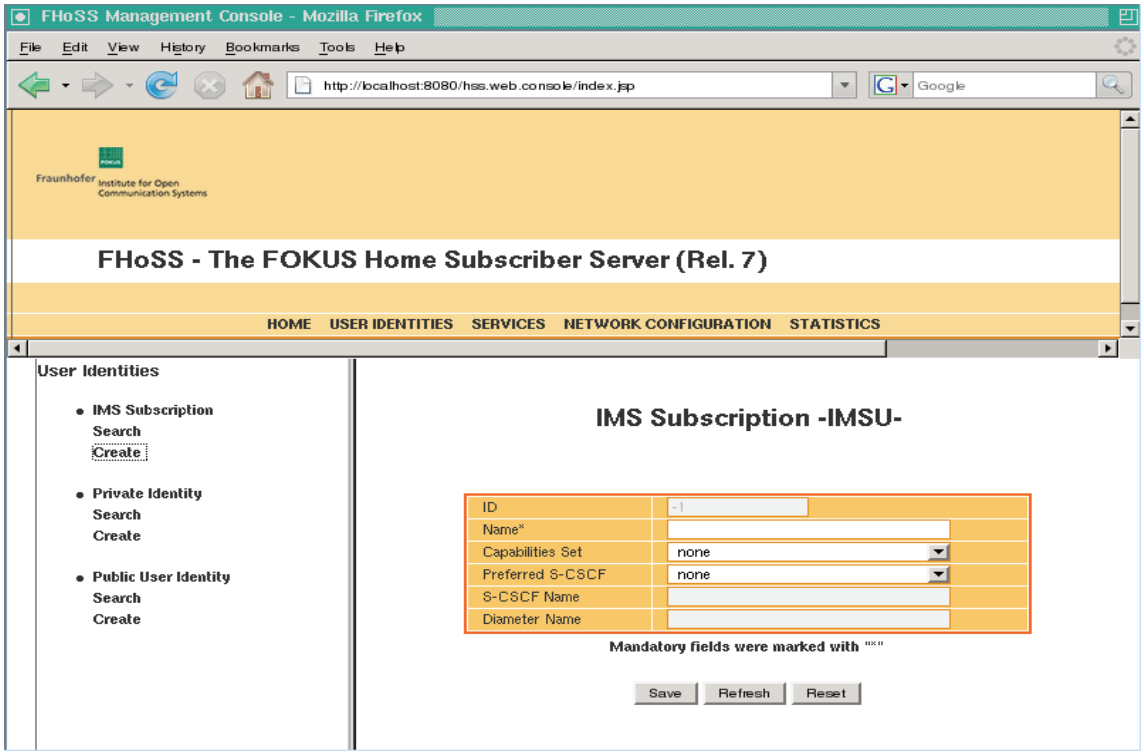
4.2 Přidávání uživatelů

Pokud chceme do sítě založené na principu IMS přidat uživatele, musíme o něm nejprve vytvořit záznam v databázi uložené v HSS. Vytváření tohoto záznamu se skládá ze tří po sobě následujících kroků. Prvním z nich je vytvoření IMSU (IMS Subscription). Dalším krokem je vytvoření privátní identity IMPI a posledním krokem při vytváření záznamu o uživateli v HSS, je nastavení jeho veřejné identity IMPU.

Open IMS Core nabízí dvě možnosti jak záznam o uživateli vytvořit. Pro konfiguraci HSS v konzolovém režimu může být použit připravený skript *add-imscore-user.sh,* umístěný v adresáři */opt/OpenIMSCore/ser_ims/cfg*. Práce s tímto skriptem je však zbytečně složitá a uživatelsky nepříliš přátelská, proto byla zvolena možnost druhá. Tou je vytvoření potřebného záznamu přes webové rozhraní databázového serveru HSS. Konfiguraci pomocí webového rozhraní lze však použít pouze v grafickém režimu, do kterého se lze, u operačního systému Gentoo, přepnout zadáním příkazu *startx*. Další podmínkou, kterou je nutno splnit, aby mohl být tento způsob konfigurace použit, je spuštění databázového serveru. Webové rozhraní je totiž jeho součástí, a pokud by nebyl spuštěn server, nefungovalo by ani toto rozhraní. Na zmíněné rozhraní se lze dostat po zadání webové adresy http://localhost:8080 do internetového prohlížeče. Pro následné přihlášení je třeba zadat login: hssAdmin a heslo: hss.

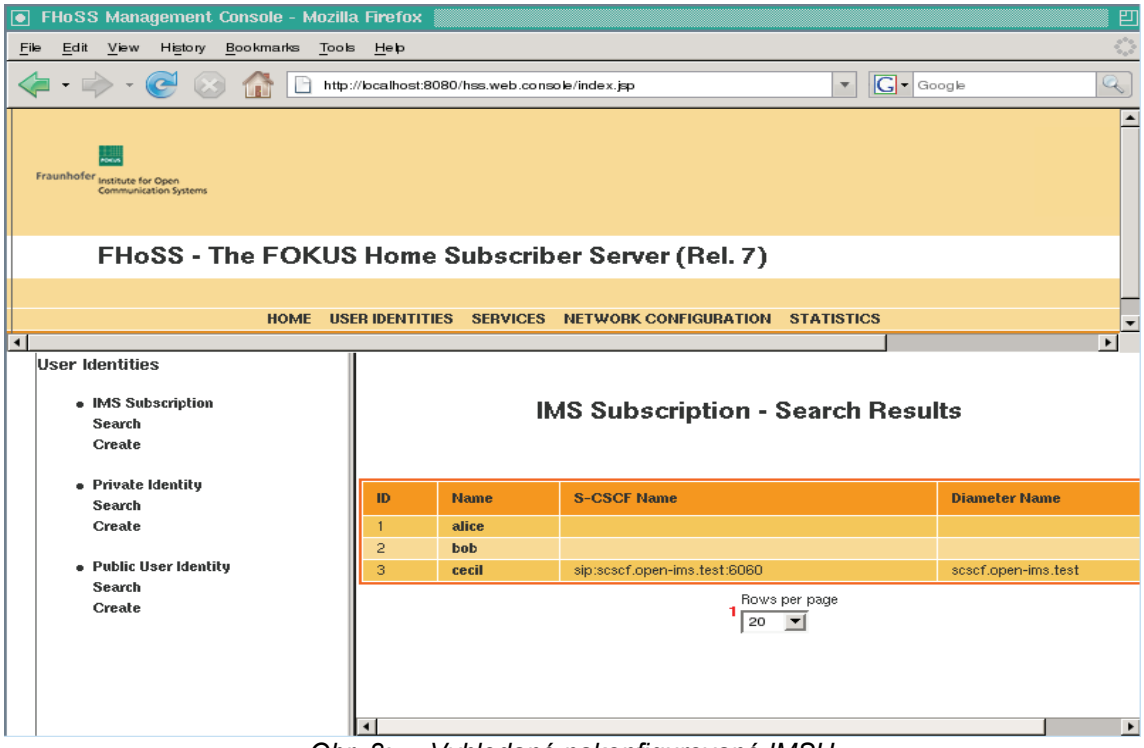
Jak můžeme vidět na následujících obrázcích, pokud chceme po přihlášení k webovému rozhraní přidat do databáze HSS novou uživatelskou identitu, vybereme z menu v horní části okna položku User Identities. Poté se nám v levé části okna objeví nabídka, ze které si můžeme pro IMSU, IMPI a IMPU vybrat ze dvou možností. Buďto můžeme vyhledat již vytvořené identity, které můžeme následně upravit, nebo můžeme vytvořit zcela novou identitu. V následujícím textu bude popsán postup přidání záznamu pro uživatele Cecil. Postup pro vytvoření uživatelů Alice a Bob byl zcela totožný.

Prvním krokem při vytváření nové uživatelské identity je vytvoření IMSU. V menu IMS Subscription tedy vybereme položku *Create*, a poté se nám objeví okno pro jeho nastavení. To můžeme vidět na obr.7. Z obrázku je zřejmé, že by v tomto kroku měla být nastavena položka *Name*, tedy jméno, kterým se bude uživatel prezentovat. Dále sada přiřazených vlastností *Capabilities Set* a nakonec preferovaný SIP server S-CSCF, tedy položka *Preferred S-CSCF*. Položka *Name*, která vlastně udává IMSU identitu uživatele, byla pro Cecil nastavena jako *cecil*. Pro *Capabilities Set* byla použita defaultní sada vlastností Open IMS Core *cap_set1* a jako preferovaný S-CSCF byl nastaven S-CSCF server *sip:scscf.open-ims.test* pracující na portu 6060. Záznam se uloží pomocí tlačítka *Save* umístěného pod konfigurační tabulkou. Po uložení záznamu je uživateli přiděleno nejbližší volné ID (Identifier). V našem případě byl uživatel Cecil přidán jako třetí, proto mu bylo přiřazeno ID 3. Při registraci uživatele Je pak uživateli ještě přidělen *S-CSCF Name* a *Diameter Name* v našem případě *sip:scscf.open-ims.test:6060* a *scscf.open-ims.test*.



Obr. 7: Okno pro nastavení IMSU

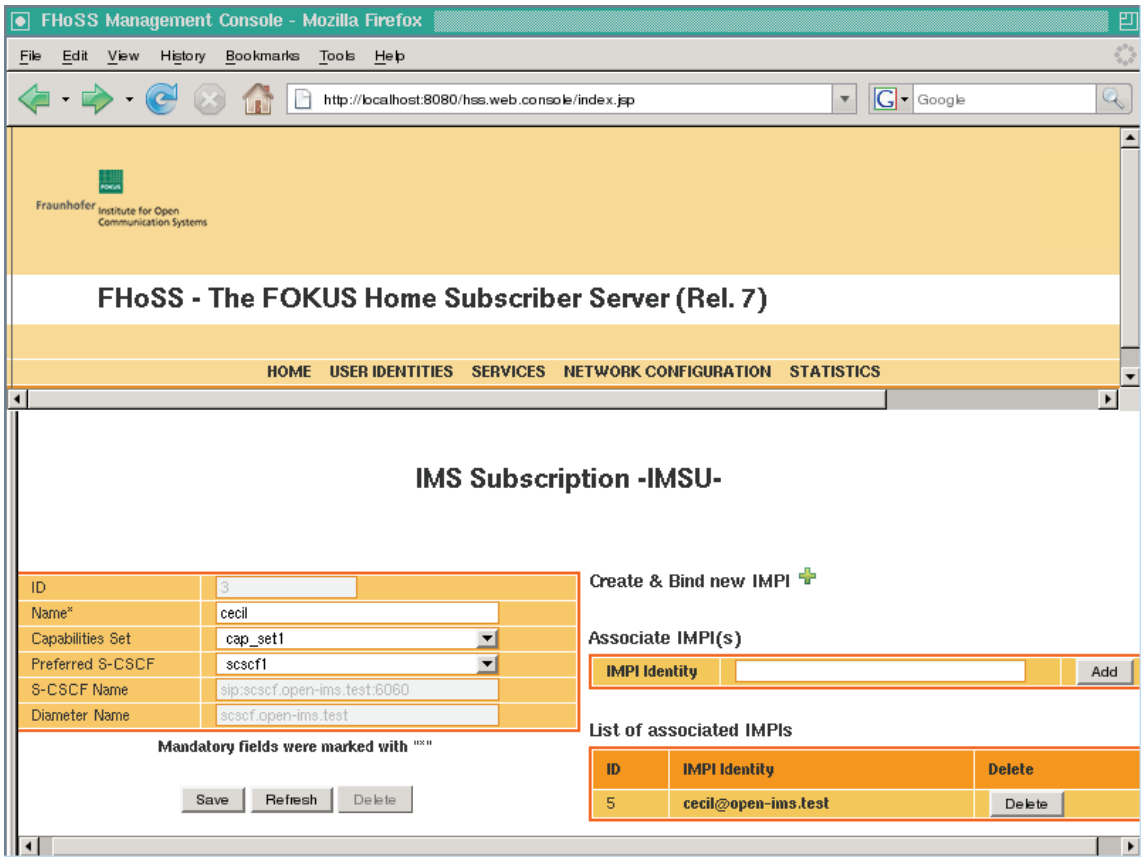
Již vytvořené uživatelské identity IMSU můžeme vyhledat pomocí položky *Search* v menu *IMS Subscription*. Vyhledávat můžeme podle ID uživatele, jeho jména nebo podle názvu serveru S-CSCF. V tom případě jsou vypsáni všichni uživatelé, kterým je daný server přiřazen. Pokud nejsou do vyhledávací tabulky zadány žádné údaje, ukáží se po vyhledání všechny uživatelské záznamy. Tuto situaci můžeme vidět na obr.8.



Obr. 8: Vyhledané nakonfigurované IMSU

Jak je na obr.8 vidět, ve výpisu záznamů je uvedeno ID uživatele, jeho nastavené jméno a v případě, že je uživatel registrován, je u něj také uveden jemu přiřazený S-CSCF server a Diameter Name. V situaci zobrazené na obr.8 je registrován pouze uživatel Cecil.

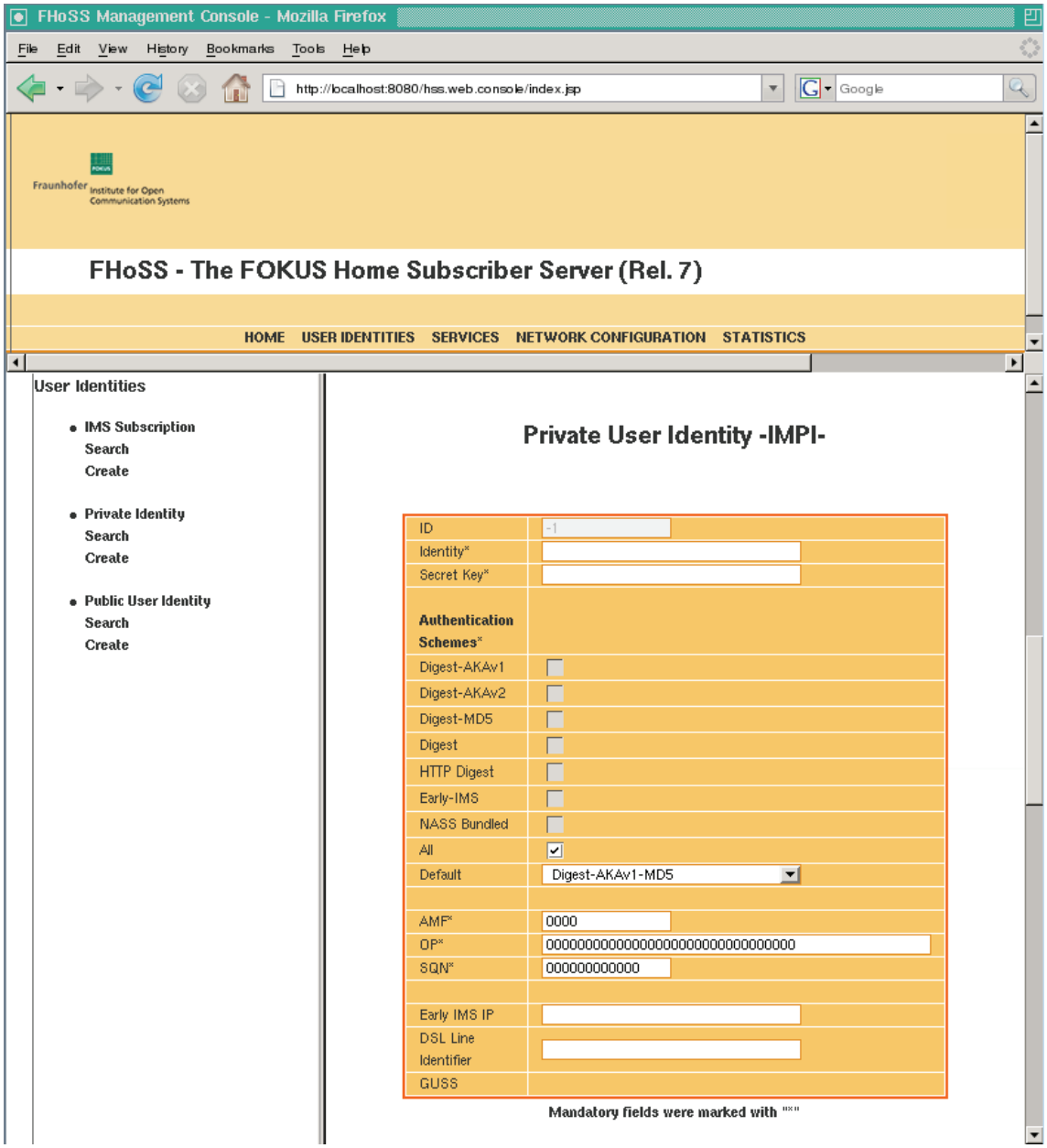
Podrobnosti o záznamu daného uživatele můžeme získat po kliknutí na jeho jméno. Poté se nám, jak můžeme vidět na obr.9, zobrazí okno s podrobnostmi o tomto záznamu. Kromě již zmíněných informací o IMSU záznamu zde můžeme uživateli přiřadit také privátní identitu IMPI.



Obr. 9: Náhled nastaveného IMSU pro uživatele Cecil

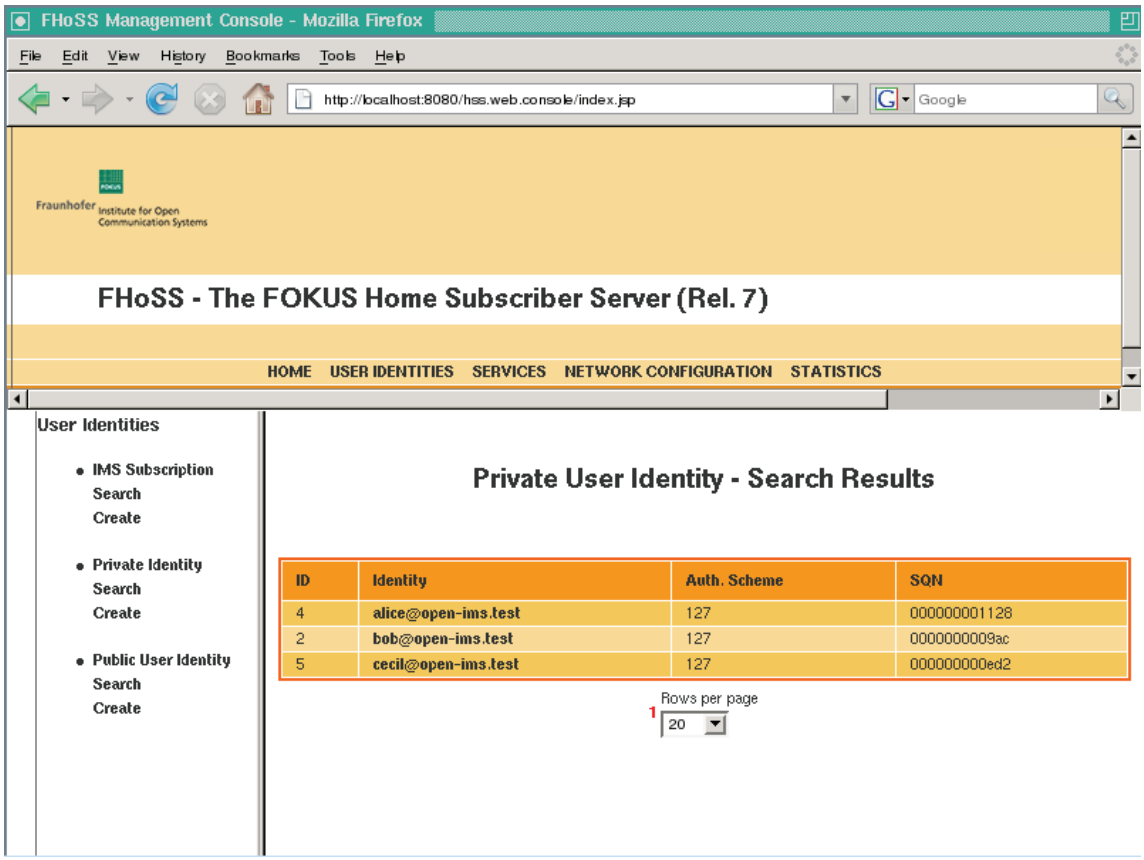
Pro nastavení privátní identity uživatele IMPI se nám otevře okno, které můžeme vidět na obr.10. K tomuto oknu se můžeme dostat buďto z již zmíněného náhledu podrobností o IMSU uživatele, nebo pomocí položky *Create* v menu *Private Identity*. Jak je z obrázku vidět, nastavujeme zde privátní identitu, tajný klíč, autentizační schéma, proměnnou AMF (Authentication Management Field) využívanou při autentizaci, OP (Operator ID), neboli identifikátor operátora vytvářejícího oprávnění, a také sekvenční číslo SQN (Sequence Number).

Položka *Identity* byla pro uživatele Cecil nastavena na *cecil@open-ims.test*. Jako *Secret Key* zde bylo pro jednoduchost použito slovo *cecil*. U autentizačních schémat byla zvolena možnost povolení všech uvedených metod, avšak jako defaultní byla zvolena autentizační metoda Digest-AKAv1-MD5. Hodnoty AMF, OP a SQN jsou udávány v hexadecimálním tvaru. Proměnná AMF má délku 4 B a její hodnota je podle návodu na stránkách institutu FOKUS nastavena na 0000. Délka proměnné OP je 32 B, a proto je, opět podle návodu, nastavena na dvaatřicet nul. Velikost sekvenčního čísla SQN je 12 B. Toto číslo může být nastaveno libovolně, protože při registraci je uživateli přiřazeno nové sekvenční číslo, kterým je to staré přepsáno. V našem případě bylo zvoleno vyplnění pole dvanácti nulami. ID je danému IMPI přiřazeno opět až po uložení záznamu.



Obr. 10: Okno pro nastavení IMPI

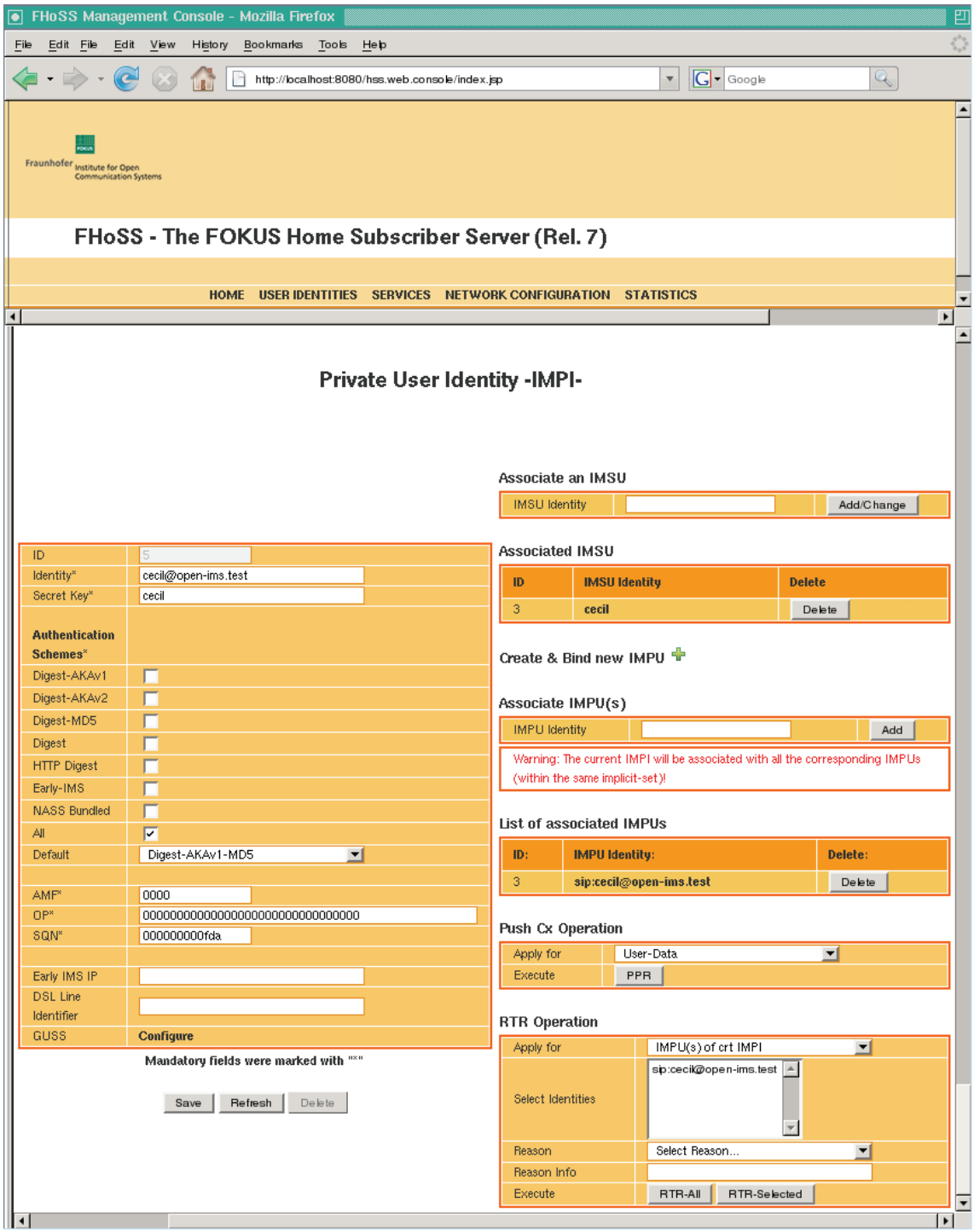
Tak jako v případě IMSU, můžeme i zde vyhledat, pomocí položky *Search* v menu *Private Identity*, již vytvořené privátní identity IMPI. V případě IMPI však můžeme vyhledávat pouze podle ID nebo podle privátní identity. Avšak stejně jako u IMSU platí, že pokud nejsou do vyhledávací tabulky zadány žádné údaje, vypíší se po vyhledání všechny dosud vytvořené privátní identity. Tato situaci je zobrazena na obr.11. Z obrázku můžeme vidět, že výpis identit v tabulce obsahuje přiřazené ID, privátní identitu uživatele, kód nastaveného autentizačního schématu a aktuální přiřazené sekvenční číslo.



Obr. 11: Vyhledané nakonfigurované privátní identity

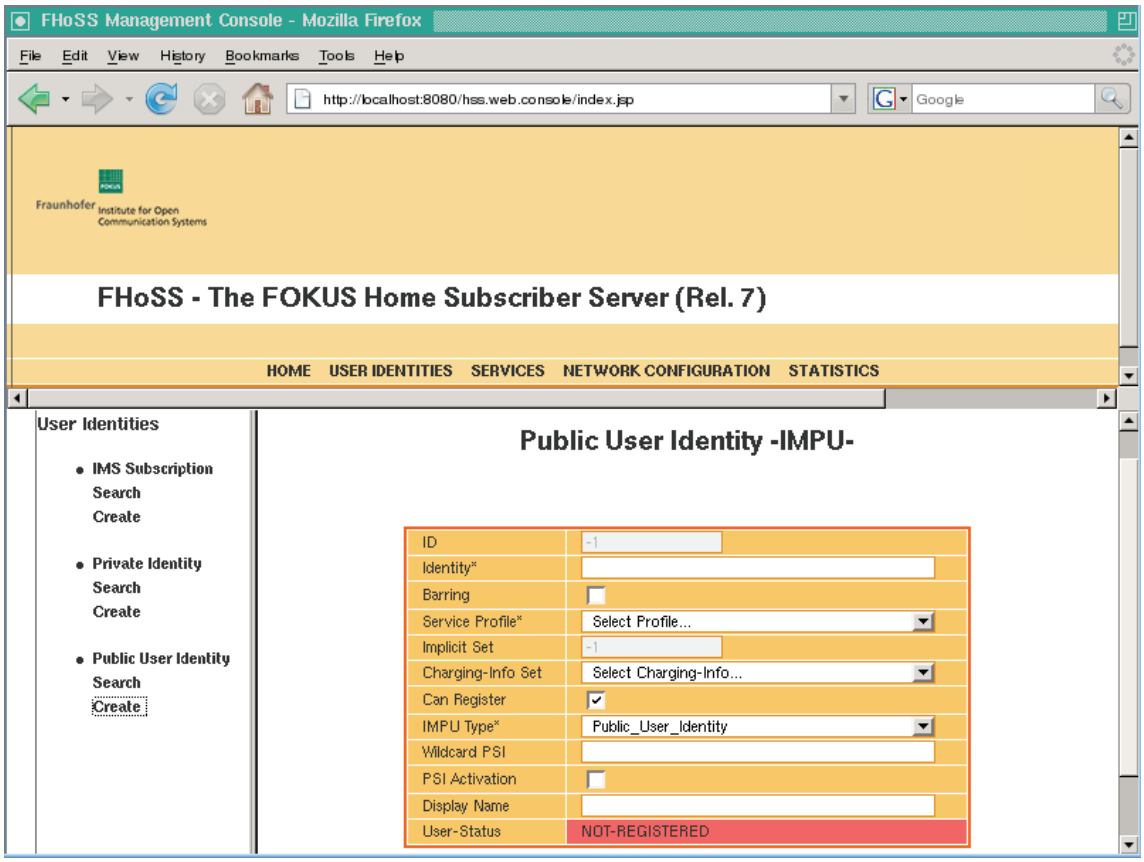
Podrobnosti o dosavadním záznamu daného uživatele můžeme získat po kliknutí na název jeho privátní identity ve vypsané tabulce. Následně jsou nám pak zobrazeny podrobné informace o vybraném IMPI, tak jak je můžeme vidět na obr.12.

Kromě informací o nastaveném IMPI zde můžeme nalézt také informace o nastaveném IMSU a jeho ID, dále také možnost nastavení, na které operace se má využít rozhraní Cx. Nakonec je zde umístěno nastavení RTR (Response Time Reporter) operací. Ty umožňují monitorování výkonu sítě a zařízení na základě měření doby odezvy (Response Time) a dostupnosti síťových zařízení. Stejně jako byla u zobrazených podrobností o IMSU možnost přiřadit uživateli IMPI, také zde můžeme danému uživateli přiřadit jeho, tentokrát veřejnou, identitu IMPU.



Obr. 12: Náhled nastaveného IMPI pro uživatele Cecil

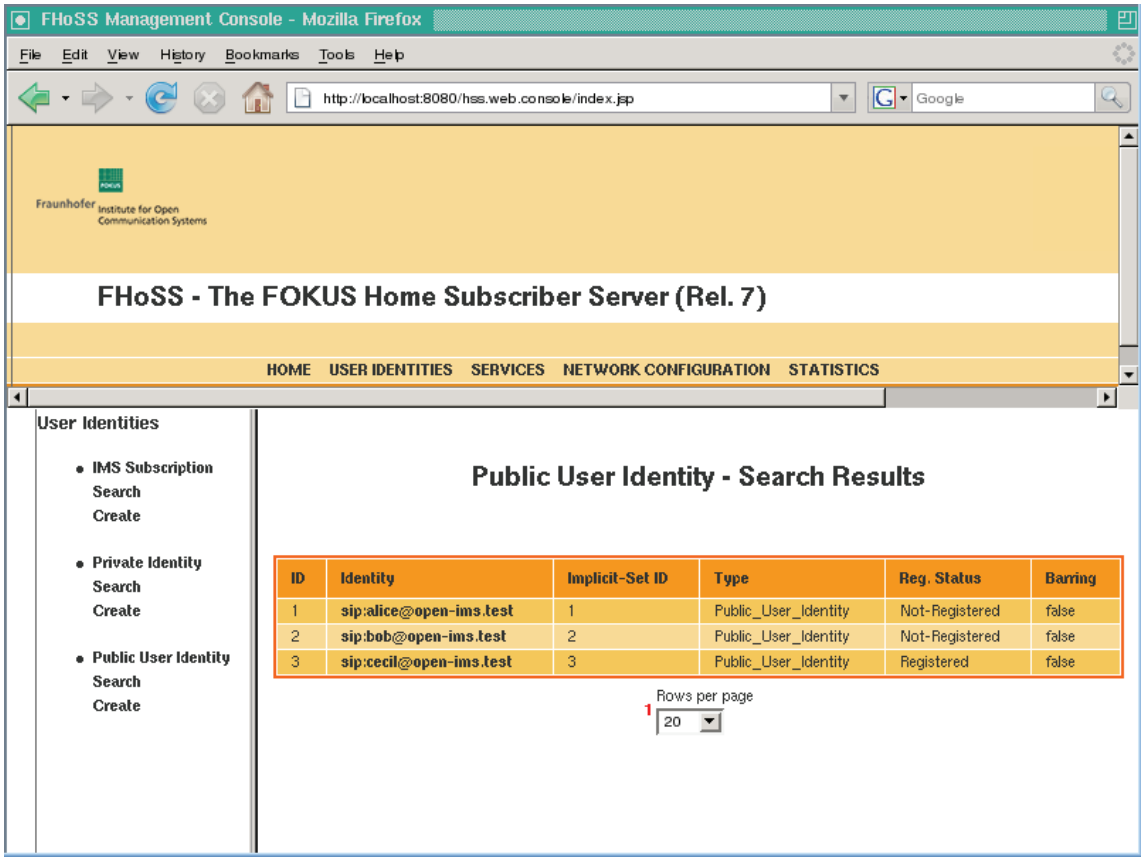
Veřejnou identitu uživatele IMPU můžeme nastavovat v konfiguračním okně, které se nám zpřístupní buďto po kliknutí na řádek *Create & Bind new IMPU* ve výpisu podrobností o IMPI, nebo pomocí položky *Create* v menu *Public User Identity*. Toto konfigurační okno můžeme vidět na obrázku 13. Z tohoto obrázku můžeme opět vidět položky nutné pro vytvoření nové veřejné identity IMPU. Musíme zde nastavit veřejnou identitu, možnost barringu, profil služeb, možnost registrovat se a také typ IMPU.



Obr. 13: Okno pro nastavení IMPU

Položka *Identity* zde v tomto případě představuje veřejnou identitu. Tato identita byla nastavena jako *sip:cecil@open-ims.test*. Další položkou, kterou je pro vytvoření IMPU nutno nastavit je zaškrtávací políčko *Barring*. Barring je doplňková služba, která zajišťuje omezení odchozích nebo příchozích hovorů. Neumožňuje však práci s konkrétními telefonními čísly, ale dovoluje omezit pouze určitou skupinu volání (např. skupinu odchozích volání do zahraničí). U tohoto zaškrtávacího políčka je podle zaškrtnutí nastavována hodnota true nebo false. Pro tohoto uživatele byl barring nastaven na *false.*

Pro následující položku *Service Profile* byl vybrán defaultní profil služeb Open IMS Core označený jako *default_sp*. Dále zde byla zaškrtnuta možnost registrace. Poslední důležitou položkou pro nastavení IMPU je *IMPU Type*, který je určen pro výběr typu IMPU. Můžeme zde vybrat ze tří možností, a to *Public_User_Identity*, *Distinct_PSI* (Public Service Identifier) a *Wildcarded_PSI*. Jelikož první jmenovaná možnost je určena pro IMS klienty, zatímco poslední dvě pro SIP klienty, byla vybrána možnost *Public_User_Identity.* Poslední řádek tabulky obsahuje položku *User-Status*, která nám zobrazuje, zda je uživatel registrován či nikoliv. Vytvořený záznam poté opět uložíme pomocí tlačítka *Save*. ID je příslušnému IMPU přiřazeno opět až po uložení záznamu.

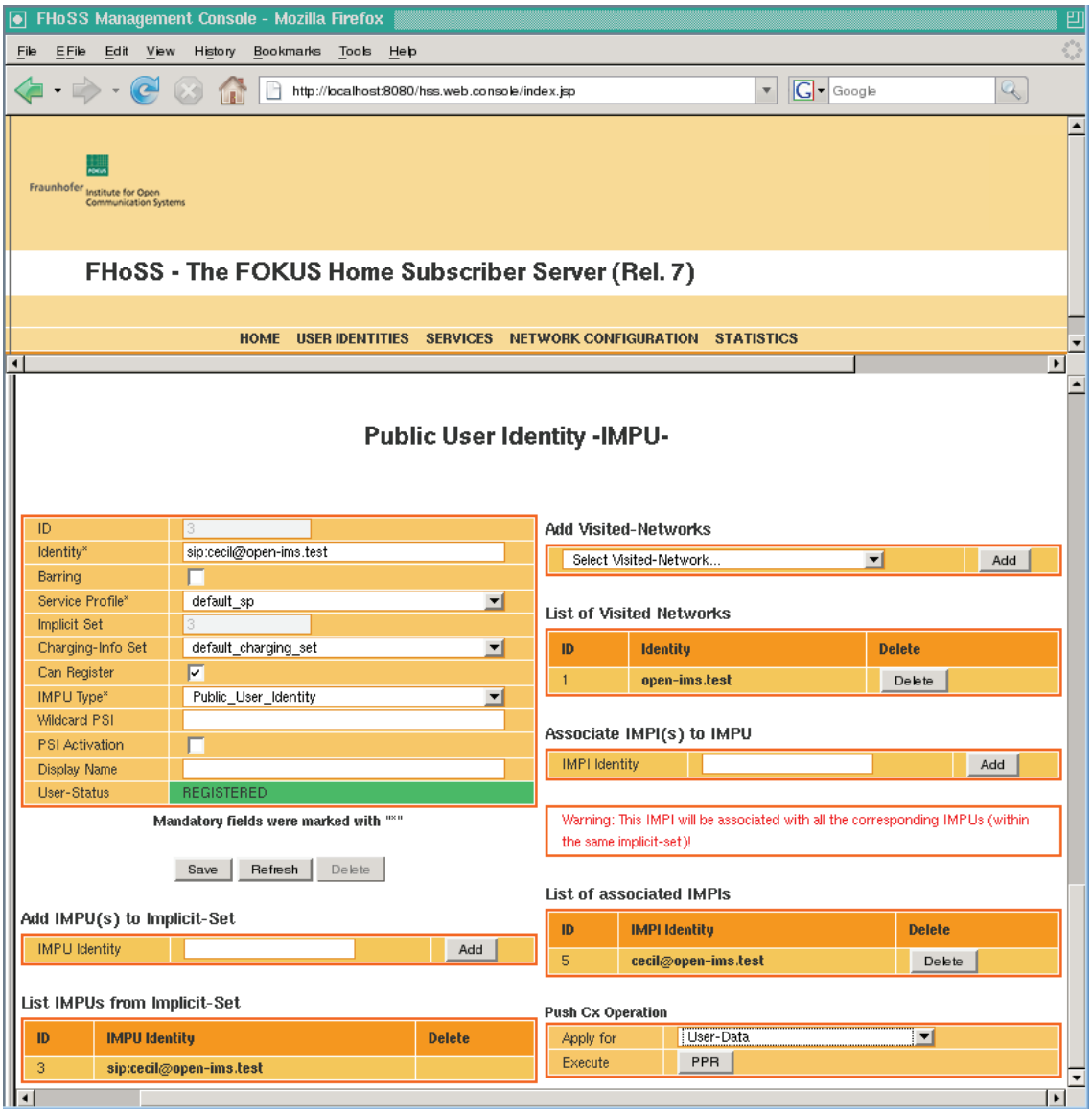


Obr. 14: Vyhledané nakonfigurované veřejné identity

Také zde můžeme pomocí položky *Search* v menu *Public User Identity* vyhledat již vytvořené veřejné identity IMPU. V tomto případě máme možnost vyhledávat podle podobných parametrů jako v případě IMPI, tedy podle ID a identity. Zde však přibylo ještě vyhledávání podle tzv. implicitně nastaveného ID, které značí ID přiřazené uživateli v rámci ukládání IMSU. Stejně jako u předchozích vyhledávání pro IMSU a IMPI, platí i zde, že pokud nejsou do vyhledávací tabulky zadány žádné údaje, vypíší se všechny uložené IMPU záznamy, jak můžeme vidět na obr.14. Z něj můžeme vidět, že výpis obsahuje přiřazené ID, veřejnou identitu uživatele, implicitně nastavené ID a sloupeček *Type*, který představuje typ IMPU. Dále můžeme ve výpisu nalézt informaci o tom, zda je uživatel s daným IMPU registrován, nebo jestli podporuje doplňkovou službu barring.

Podrobnosti o konečném záznamu daného uživatele můžeme získat po kliknutí na název jeho veřejné identity v tabulce výpisů. Následně se nám zobrazí podrobné informace o IMPU vybraného uživatele. Na obr.15 jsou tyto podrobnosti zobrazeny pro uživatele Cecil.

V tomto výpisu podrobností jsou nám k dispozici ucelené informace o uživateli. Kromě informací o nastaveném IMPU tak v něm můžeme nalézt i informaci o nastaveném jménu IMSU a jeho ID. Dále zde také vidíme nastavenou identitu IMPI s jejím ID, a rovněž ID a identitu pro IMPU. Stejně jako u podrobného výpisu o daném IMPI je také tady možnost nastavení operací, pro které se má využít rozhraní Cx. Podle položky User-Status můžeme vidět, že uživatel Cecil je momentálně registrován.



Obr. 15: Náhled nastaveného IMPU pro uživatele Cecil

4.3 Klientské Programy

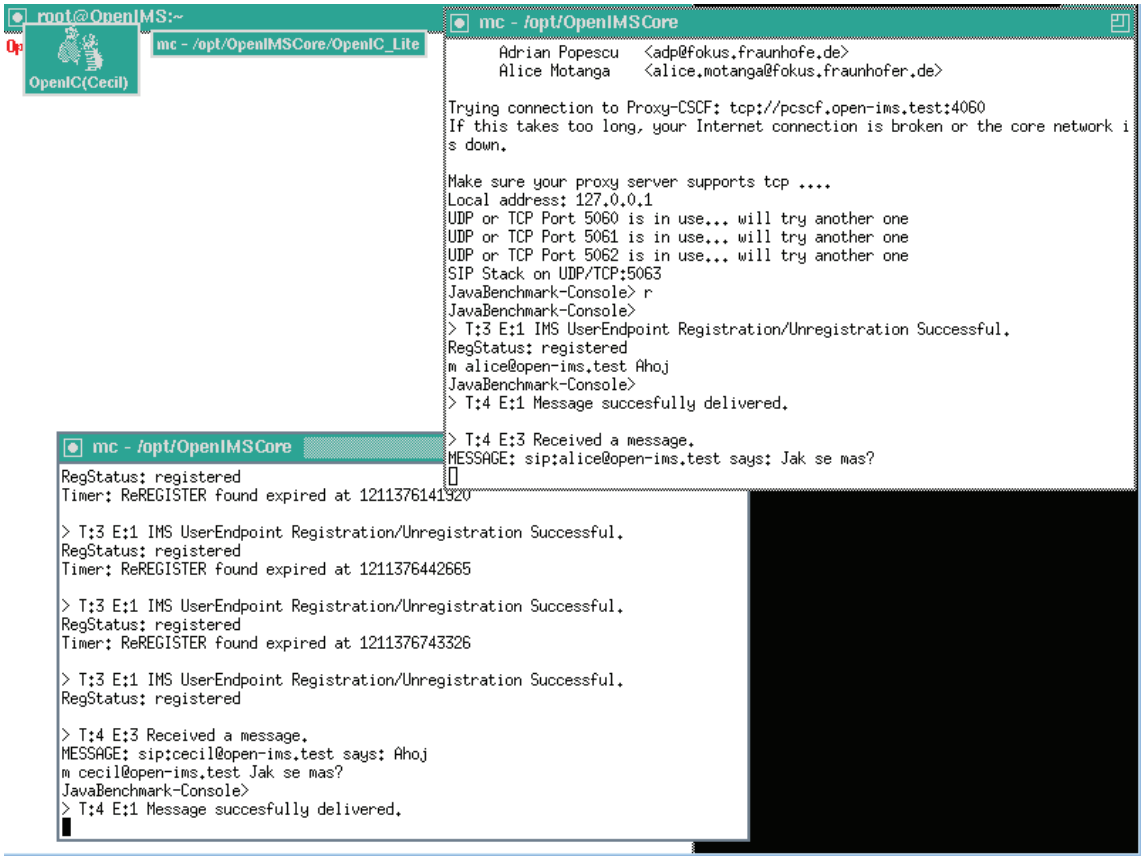
Nyní je již vytvořen záznam o uživateli v HSS. Vlastní komunikace však probíhá mezi klienty, a proto je pro úspěšné sestavení spojení zapotřebí IMS klienty nainstalovat, a posléze je nastavit. V našem případě bylo zvoleno použití klientského programu OpenIC_Lite, který je free verzí komerčního OpenIC klienta, vyvynutého fraunhoferským institutem FOKUS. Ten nabízí buďto konzolové využití anebo ovládání přes grafické rozhraní.

Při využívání konzolového programu je zapotřebí ručně vytvořit konfigurační soubor, který obsahuje všechny potřebné informace pro registraci daného uživatele. Tyto informace jsou, až na výjimky, stejné jako ty, které byly zadávány při vytváření uživatelského záznamu v HSS. Konfigurační soubor můžeme vidět na výpisu 9. Je z něj patrné, že musíme zadat jméno, kterým se bude uživatel prezentovat, veřejnou a privátní identitu, a také realm. Ten určuje, do které domény uživatel patří. Dále je zde zadáván tajný klíč a adresa proxy serveru P-CSCF. Následně je zadávána hodnota proměnných AMF a OP, jejichž význam byl vysvětlen v kapitole o nastavování IMPI, a sekvenčního čísla SQN. Uživateli je však při každé registraci přiděleno nové SQN. Dále zde nastavujeme použití autentizačního klíče, možnost simulace ISIM (IP Multimedia Services Identity Module) a aktuální index sekvenčního vektoru. Nakonec jsou nastavovány proměnné IND_LEN, delta a L, které jsou důležité pro generování SQN. Hodnoty pro nastavení těchto proměnných byly získány z rad a výpisů konfiguračních souborů na www.berlios.de. Nastavení položek konfiguračního souboru pro uživatele Cecil je vidět ve výpisu 9.

```
// Jméno, které bude zobrazováno u uživatele Cecil
displayName=Cecil
// Veřejná identita uživatele Cecil
publicIdentities=sip:cecil@open-ims.test
// Určení, do které domény uživatel patří
realm=open-ims.test
// Privátní idetita uživatele Cecil
privateIdentity=cecil@open-ims.test
// Tajný klíč
secretKey=cecil
// Adresa SIP proxy serveru
proxyCSCF=pcscf.open-ims.test:4060/UDP
// Sekvenční číslo přiřazené uživateli Cecil
SQN=000000000f59
// Authentication Management Field
AMF=0000
// Identifikátor operátora, který vytváří oprávnění
OP=00000000000000000000000000000000
// Povolení použití autentizačního klíče
useAK=true
// Simulování ISIM vypnuto
simulateISIM=false
// Aktuální index sekvenčního vektoru
sqnVectorCurrentIndex=0
// Proměnné potřebné pro generování sekvenčních čísel SQN
IND_LEN=5
delta=268435456
L=32
```

Výpis 9: Konfigurační soubor konzolového klientského programu Cecil.cfg

Toto konzolové rozhraní umožňuje sestavování spojení nebo posílání zpráv s jinými uživateli. Na obr.16 je zobrazen příklad, kdy uživatel Alice posílá uživateli Cecil zprávu “Ahoj“ a ten odpovídá “Jak se máš?“.



Obr. 16: Posílání zpráv mezi dvěma konzolovými klienty Alice a Cecil

Konzolové rozhraní můžeme ovládat pomocí následujících příkazů.

- **x** (Exit) – umožňuje opustit aplikaci
- **r** (Register) – registruje uživatelský koncový bod
- **ur** (Unregister) – odregistruje uživatelský koncový bod
- **m** (Message) – používá se v syntaxi m <URI> <msg> a umožňuje zaslat zprávu msg na uživateli s daným URI
- **c** (Call) – používá se v syntaxi c <URI> a zahájí volání uživatele s daným URI
- **y** (Yes) – používá se v syntaxi y <ID> a přijme příchozí hovor s daným ID
- **n** (No) – používá se v syntaxi n <ID> a odmítne příchozí hovor s daným ID

- **a** (Abort) – používá se v syntaxi a <ID> a umožňuje zrušit odchozí volání s daným ID, které ještě nebylo druhou stranou přijato a dosud u ní vyzvání

- **h** (Hangup) – používá se v syntaxi h <ID> a ukončí již probíhající hovor s daným ID

Jak již bylo řečeno druhou možností, kterou OpenIC_Lite nabízí je ovládání pomocí grafického rozhraní. Tento způsob ovládání klientského programu je uživatelsky příjemnější, a také nabízí více funkcí. Kromě možnosti sestavovat spojení a posílat zprávy tak umožňuje vytvářet seznam kontaktů, poskytuje uživateli výpis volaných, přijatých a zmeškaných URI, a také možnost zachytávání provozu mezi terminálem a proxy serverem P-CSCF a následné zobrazení obsahu zachycených paketů. Ukázku takto zachycené SIP zprávy *REGISTER* můžeme vidět na obr.19.

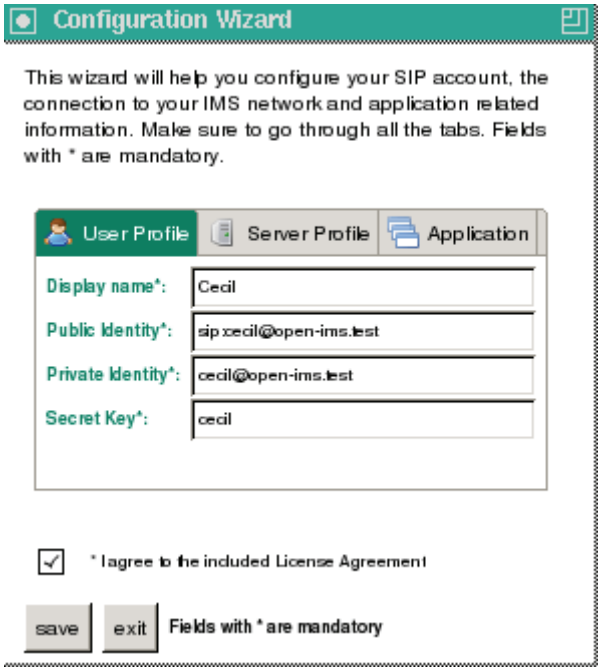
Abychom mohli tento virtuální terminál používat, musíme ho nejprve nainstalovat. Instalace na operační systémy založené na linuxovém jádře je velmi jednoduchá. Nejdříve je zapotřebí stáhnout zkomprimovaný soubor OpenIC_Lite.tar.gz. Ten je poté třeba rozbalit a v nově vzniklém adresáři už stačí jen přiřadit spouštěcímu skriptu OpenIC_Lite.sh práva pro spuštění. Od této chvíle pak můžeme OpenIC_Lite spouštět pomocí tohoto skriptu.

```
// Rozbalení zkomprimovaného souboru
tar –zxvf OpenIC_Lite.tar.gz
// Udělení práv pro spuštění souboru OpenIC_Lite.sh
chmod 744 OpenIC_Lite.sh
// Spuštění skriptu OpenIC_Lite.sh
./ OpenIC_Lite.sh
```

Výpis 10: Příkazy pro rozbalení, změnu práv a spuštění OpenIC_Lite

Jak můžeme vidět ve výše uvedeném výpisu příkazů. Aby mohl být spouštěn skript OpenIC_Lite.sh byla mu přidělena práva označená čísly 744. Tato čísla reprezentují v osmičkové soustavě binární tříbitové číslo, kde nejvýznamnější bit udává právo pro čtení souboru, druhý nejvýznamnější právo pro zápis do souboru a nejméně významný bit udává právě právo pro spuštění souboru. Pokud je právo uděleno nabývá bit hodnoty 1, v opačném případě hodnoty 0. První číslo převedené do osmičkové soustavy udává práva přidělená uživateli, druhé práva skupiny a poslední číslo reprezentuje práva pro ostatní. Práva zadaná kódem 744 tak udělují uživateli právo čtení, zápisu i spouštění. Práva skupiny i ostatních jsou omezena pouze na čtení.

Při prvním zapnutí grafického terminálu OpenIC_Lite se spustí Configuration Wizard, což je grafické prostředí pro nastavení konfiguračního souboru klientského programu. Jsou zde nastavovány přibližně stejné informace jako u konfiguračního souboru konzolového rozhraní. Příklad okna pro nastavení uživatelského profilu je zobrazen na obr.17.



Obr. 17: Configuration Wizard, ukázka nastavení uživatelského profilu

Configuration Wizard obsahuje tři části. Jedná se o část *User Profile*, kde je nastavováno zobrazované jméno uživatele, privátní a veřejná identita uživatele, a také jeho tajný klíč. Druhou částí je *Server Profile*, kde je nastavována IP adresa proxy serveru P-CSCF v IMS síti uživatele, a dále číslo portu P-CSCF serveru v této síti. Poslední nastavovanou položkou v této části je *Realm*, který určuje doménové jméno IMS sítě, do které uživatel patří. V poslední části *Application* můžeme nastavit povolení automatického přijímání hovorů, automatické registrace uživatele po spuštění terminálu a povolení zvukové signalizace, upozorňující na příchozí zprávy. Celý záznam je pak třeba uložit pomocí tlačítka *Save* ve spodní části okna.

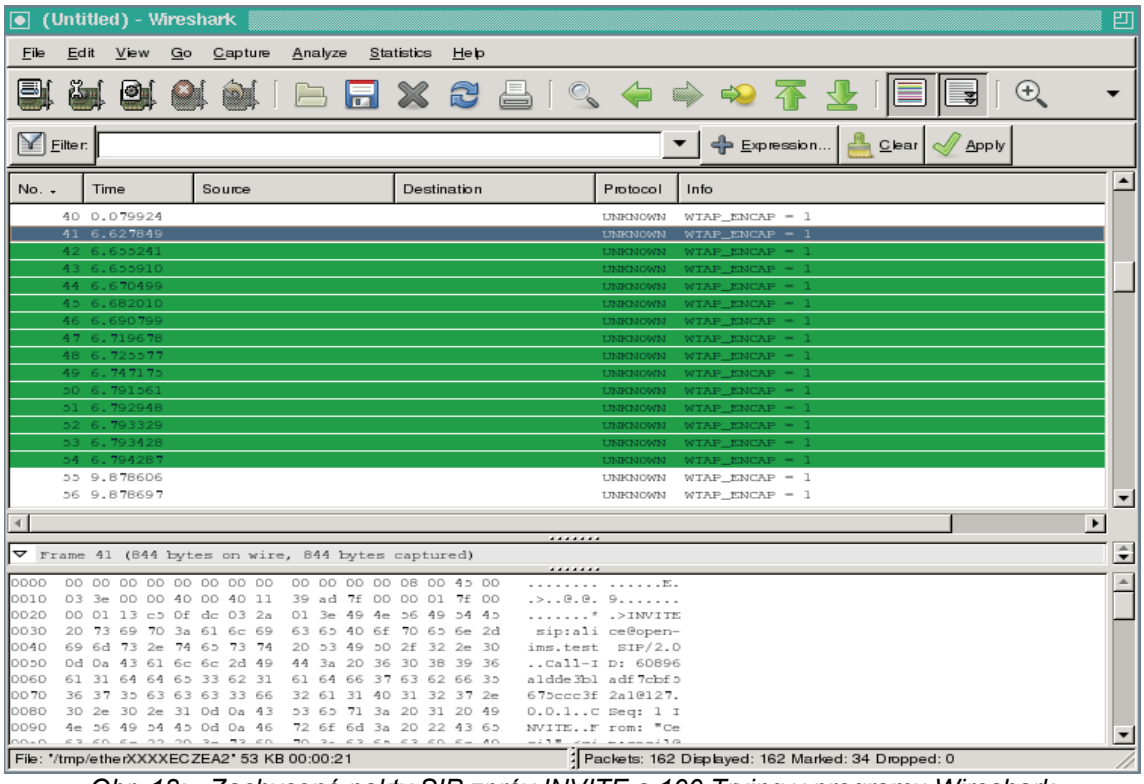
Po uložení záznamu je automaticky vygenerován konfigurační soubor profile.cfg. Jeho obsah můžeme vidět na výpisu 11. Je z něj vidět, že se zde vyskytují i položky, které nebyly zadány přes Configuration Wizard. Tyto položky byly automaticky nastaveny klientským programem při vytváření konfiguračního souboru.

```
#OpenIMS Client config file
// Identifikátor operátora, který vytváří oprávnění (vygenerováno
// automaticky)
OP=00000000000000000000000000000000
// Tajný klíč
secretKey=cecil
// Povolení dialogového okna, které se nás před vypnutím terminálu
// zeptá, zda jsme si jisti, že to myslíme vážně
showExitDialog=true
// Authentication Management Field
AMF=0000
// Povolení zvukových efektů
autoPlaySound=true
// Sekvenční číslo přiřazené uživateli Cecil
SQN=000000000000
// Určení, do které domény uživatel patří
realm=open-ims.test
// Privátní identita uživatele Cecil
privateIdentity=cecil@open-ims.test
// Adresa SIP proxy serveru
proxyCSCF=pcscf.open-ims.test\:4060/UDP
// Veřejná identita uživatele Cecil
publicIdentities=sip\:cecil@open-ims.test
// Jméno, které bude zobrazováno u uživatele Cecil
displayName=Cecil
// Povolení použití autentizačního klíče
useAK=true
// Zakázání automatického přijetí hovoru
autoAnswer=false
// Povolení automatické registrace uživatele po spuštění terminálu
autoSignIn=true
// Terminál podporuje všechny potřebné IMS požadavky
earlyIMS=false
```

Výpis 11: Konfigurační soubor klientského programu OpenIC_Lite pro uživatele Cecil

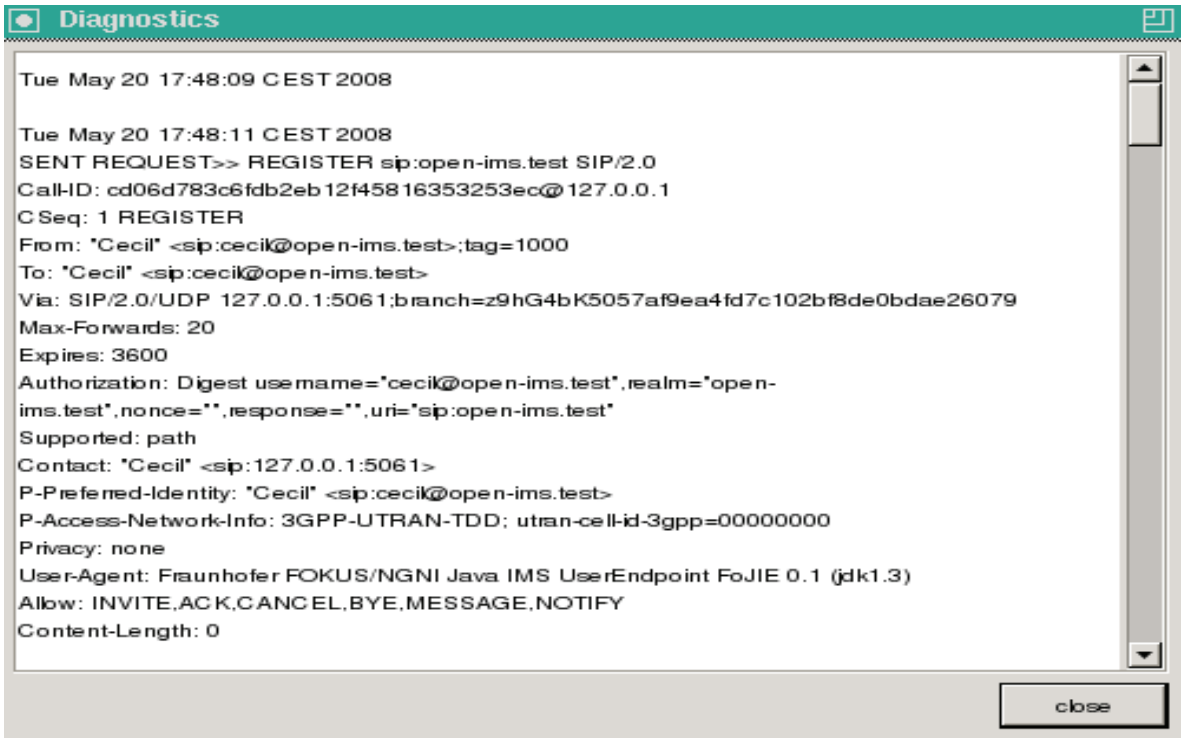
4.4 Zachytávání komunikace

Pro zachytávání komunikace v námi vytvořené síti se nám opět nabízí hned několik možností. Asi nejkomplexnější služby nabízí program Wireshark, který umožňuje jak zachytávání veškerých dat proudících v síti, tak jejich pozdější analýzu. Pakety zachycené pomocí programu Wireshark můžeme vidět na obr.18. Jak je však na tomto obrázku vidět, je protokol u všech zachycených paketů označen *unknown* tedy jako neznámý, a to i přesto, že databáze protokolů programu Wireshark obsahuje jak protokol SIP, tak protokol Diameter. Z toho důvodu bylo zapotřebí rozlišovat získaná data podle obsahu hlaviček zachycených paketů.



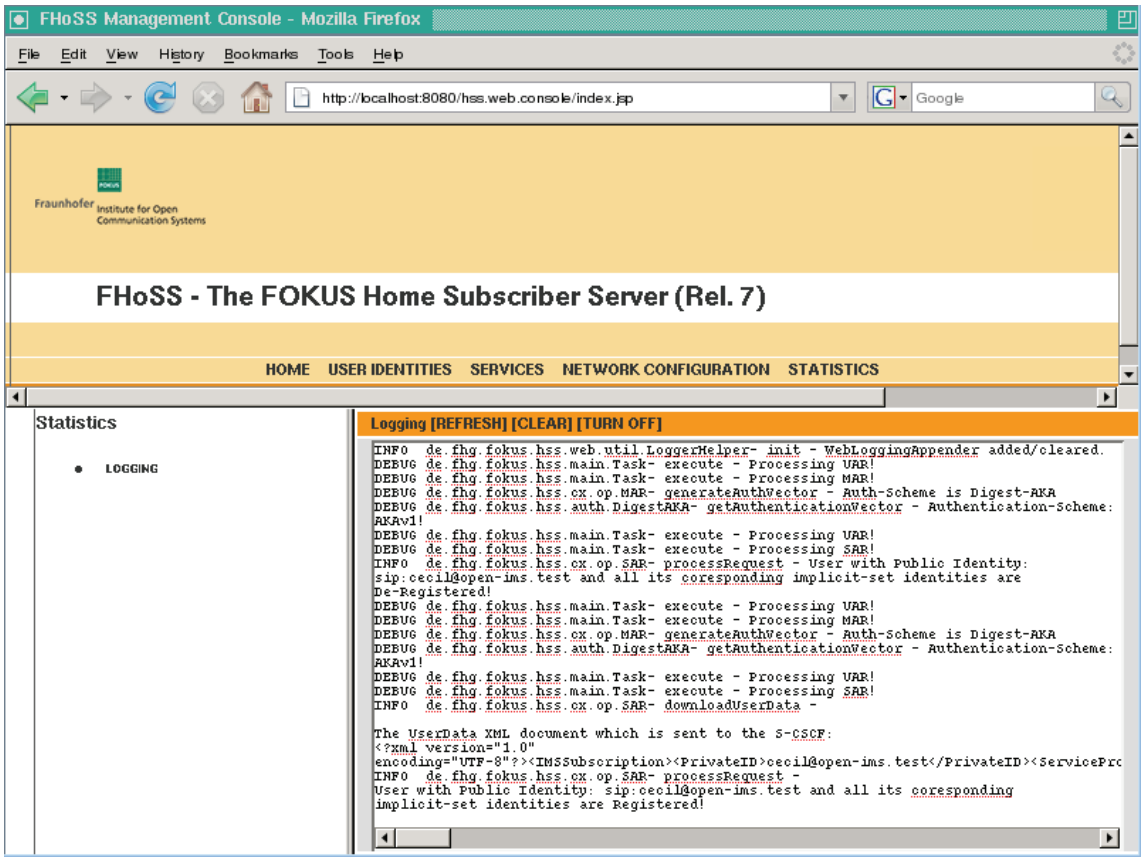
Obr. 18: Zachycené pakty SIP zpráv INVITE a 100 Trying v programu Wireshark

Zachytávání dat však nabízí, jak bylo zmíněno výše, také klientský program OpenIC_Lite. Ten zachytává data na rozhraní mezi terminálem a Proxy-CSCF a zobrazuje obsah zachycených paketů. Ukázka SIP zprávy *REGISTER* zachycené při registraci uživatele Cecil je vidět na obr.19.



Obr. 19: Obsah SIP zprávy REGISTER zachycené během registrace uživatele pomocí klientského programu OpenIC_Lite

Poslední možností je využití již zmíněného webového rozhraní databázového serveru HSS. Tímto způsobem může být zaznamenáván příchozí provoz na HSS. Můžeme tak zde vidět pakety, které si HSS vyměňuje s I-CSCF a S-CSCF. Zachytávání na webovém rozhraní spustíme pomocí položky *TURN ON – DEBUG* v menu *STATISTICS*. Náhled takto zachycených paketů je zobrazen na obr.20.

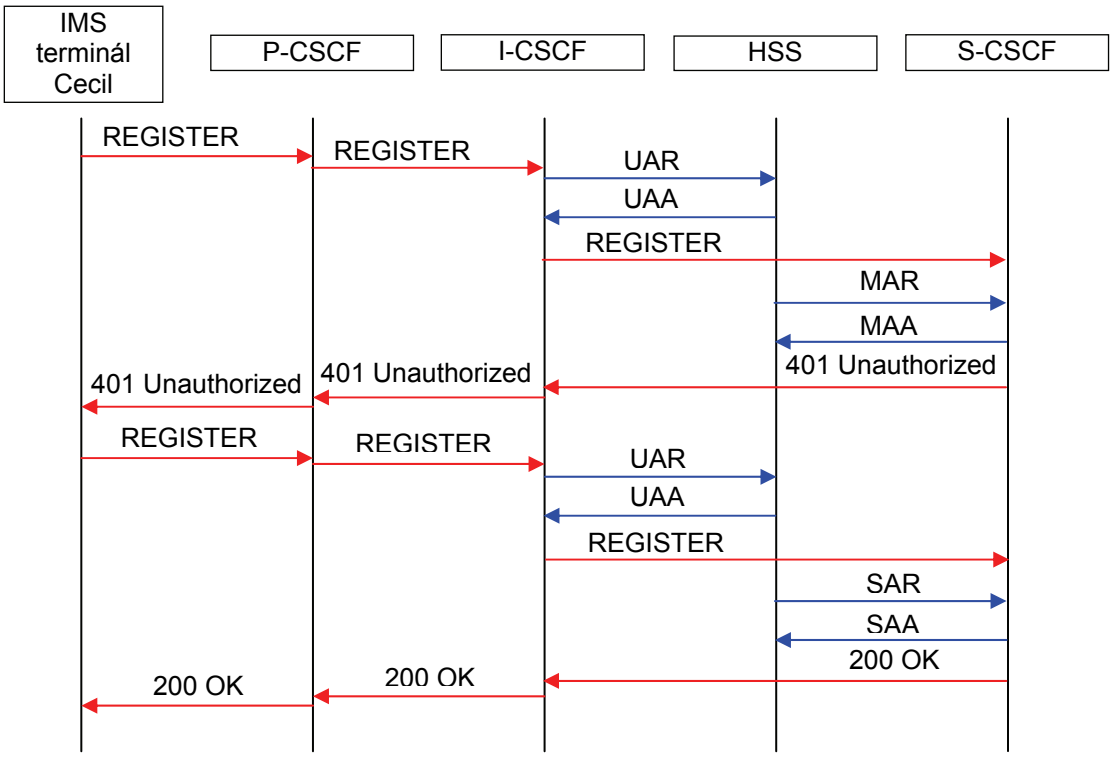


Obr. 20: Zachycené zprávy protokolu Diameter přicházející do HSS během registrace uživatele

V následujícím textu budou ukázány a popsány stavové diagramy sestavené na základě zachycené komunikace mezi dvěma uživateli Alice a Cecil. Při tvorbě těchto diagramů se z převážné části vycházelo z informací, které o provozu v síti poskytl program Wireshark. Doplňující informace pak byly získány z klientského programu OpenIC_Lite a webového rozhraní databázového serveru HSS. Těmito programy byl zachycen provoz v síti při procesu registrace uživatele a při navazování a ukončování spojení mezi dvěma účastníky. Pro všechny následující diagramy platí, že zprávy označené červenými šipkami patří protokolu SIP, a protokolu Diameter patří naopak zprávy označené šipkami modrými.

Jak můžeme vidět na obrázku 21, IMS terminál zahajuje svou registraci zasláním SIP zprávy *REGISTER* proxy serveru P-CSCF. Smyslem této žádosti je sdělit aktuální polohu uživatele. Zpráva *REGISTER* obsahuje informaci o aktuální IP adrese a portu, na kterém může být uživatel zastižen. Registrace je časově limitována a z toho důvodu je nutné její periodické opakování. P-CSCF

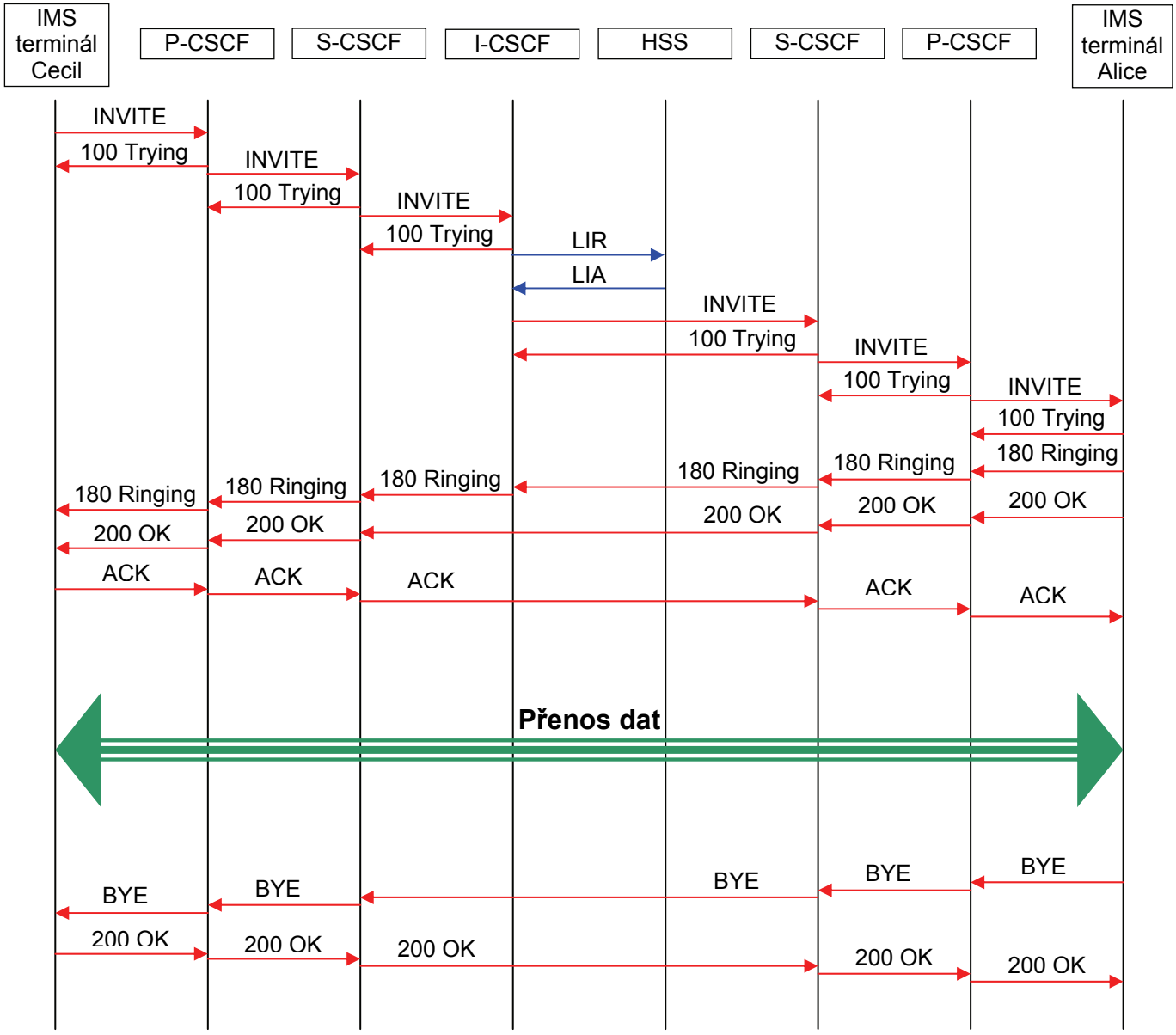
poté zašle tuto zprávu serveru I-CSCF. Ten následně provede, pomocí zpráv UAR a UAA (User Authorization Request/Answer) protokolu Diameter, výměnu autorizačních informací uživatele mezi I-CSCF a HSS. Zpráva UAR obsahuje informace o uživatelské identitě, veřejné identitě uživatele nutné pro registraci, o aktuálně připojených sítích a o typu registrace. Zpráva UAA, která je zasílána HSS jako odpověď na UAR, obsahuje, v závislosti na stavu uživatele, název S-CSCF nebo informaci pro jeho výběr. [3] Nyní když zná I-CSCF jméno registračního serveru S-CSCF, zasílá na něj požadavek o registraci uživatele, opět ve formě SIP zprávy *REGISTER*. Registrační server si po jejím obdržení vymění autentizační informace s databázovým serverem HSS. Toto se provádí pomocí zpráv MAR a MAA (Multimedia Authentication Request/Answer) protokolu Diameter. Zpráva MAR je zasílána z HSS na S-CSCF a obsahuje uživatelskou identitu pro autentizaci, a pokud je to umožněno, tak také autentizační parametry. Dále obsahuje jméno S-CSCF, po kterém je autentizace uživatele žádána. Odpovědí na tento požadavek je pak zpráva MAA, která vrací výsledek autentizace, a pokud je o to požádána, také autentizační parametry pro kontrolu uživatele. [3] Jak můžeme vidět na obr.21, protože se v IMS sítích provádí ověřování registrace, odpovídá registrační server uživateli zprávou *401 Unauthorized*, která obsahuje výzvu pro doplnění potřebných údajů. Po doplnění hlavičky o tyto údaje zasílá uživatel novou zprávu *REGISTER*. Nyní již vše proběhne v pořádku a uživatel je pomocí zpráv SAR a SAA (Server Assignment Request/Answer) protokolu Diameter úspěšně registrován. Zpráva SAR obsahuje identity pro registraci nebo případnou deregistraci závisející na typu operace požadované po S-CSCF. V odpovědi SAA na tento požadavek jsou pak obsažena uživatelská data. Úspěšné zakončení procesu registrace ja nakonec potvrzeno SIP zprávou *200 OK*.



Obr. 21: Stavový diagram registrace uživatele v IMS

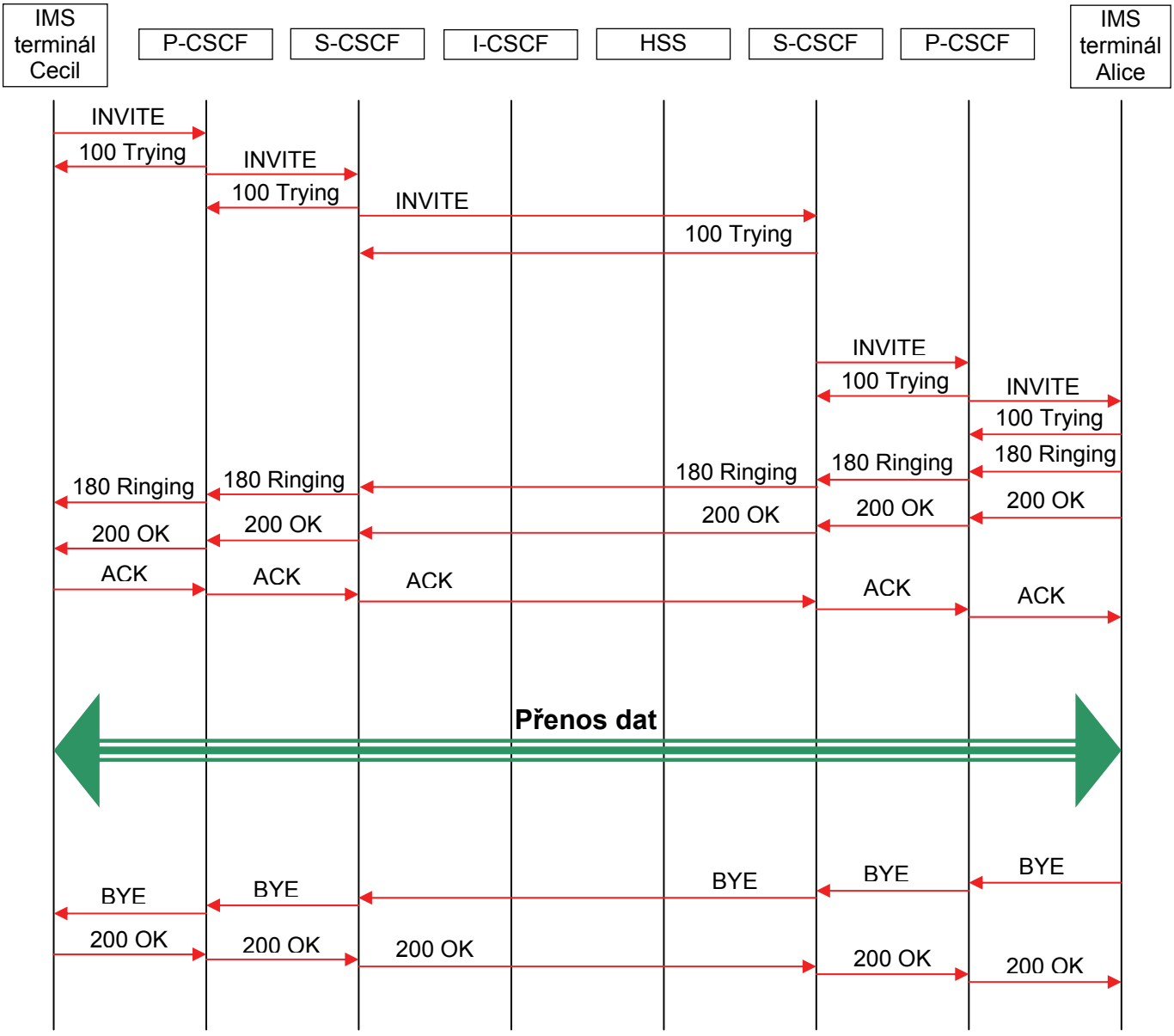
Na dalším obrázku můžeme vidět stavový diagram pro proces sestavování a ukončování spojení v sítích založených na principu IMS. Z diagramu je zřejmé, že volající, v tomto případě uživatel Cecil, zasílá pro zahájení hovoru SIP zprávu *INVITE*. Tato zpráva je zaslána na proxy server P-CSCF a obsahuje důležitá pole jako je *Call ID*, které jednoznačně identifikuje dané spojení. Dále se jedná o položky *From*, *To* a *Via*. Položka *From* identifikuje volajícího a v rámci spojení se nemění. To znamená, že zůstává stejná i ve zprávách, které zasílá volaný volajícímu. Položka *To* identifikuje volaného a *Via* obsahuje vedle verze SIPu a použitého transportního protokolu také IP adresu a port původce zprávy. Každý server, který zprávu odesílá dál vloží do hlavičky další záznam *Via* se svou IP adresou a portem. Tyto záznamy pak mohou být použity pro detekci smyček. Další důležitou položkou obsaženou ve zprávě *INVITE* je *Contact*. Zde můžeme nalézt URI pomocí kterého lze účastníka kontaktovat přímo. Zpráva *INVITE* také obsahuje informace o použitém protokolu a portu, na kterém bude volající vysílat. [9] Každé úspěšné přenesení zprávy *INVITE* je potvrzeno zprávou *100 Trying*. Poté co se *INVITE* dostane na I-CSCF server, který plní funkci kontaktního bodu, získá tento server pomocí zpráv LIR a LIA (Location Info Request/Answer) protokolu Diameter z HSS adresu S-CSCF serveru volaného uživatele. Zpráva LIR obsahuje veřejnou identitu uživatele, ke které je žádána adresa S-CSCF. Odpověď LIA vrací adresu S-CSCF registrovaného uživatele nebo serveru. [3] Poté, když už I-CSCF server zná adresu S-CSCF serveru volaného účastníka, zašle zprávu *INVITE* přímo na tento server. Odtud se pak zpráva dostane přes příslušný proxy server až k volanému uživateli. Poté co volaný účastník obdrží zprávu *INVITE*, tak ji buďto odmítne nebo, jak můžeme vidět na obr.22, vyšle volajícímu zprávu *180 Ringing*, kterou oznamuje, že signalizuje příchozí hovor a čeká na jeho přijetí nebo zamítnutí. Ve chvíli, kdy uživatel Alice hovor přijme je k volajícímu účastníkovi Cecil vyslána zpráva *200 OK*, která, stejně jako zpráva *INVITE*, obsahuje informaci o použitém protokolu a číslu portu, na kterém bude stanice vysílat. V našem případě byl dohodnut protokol UDP/RTP (Real-time Transport Protocol) a port 8000 pro volajícího a 8001 pro volaného. Po obdržení zprávy *200 OK* potvrdí volající sestavení spojení tím, že volanému odešle zprávu *ACK* (Acknowledgement). Potom už může být realizován vlastní hovor.

Proces ukončení spojení probíhá tak, že účastník, který chce spojení ukončit, zašle druhému účastníkovi SIP zprávu *BYE*. V případě situace zobrazené na stavovém diagramu na obr.22 ukončuje hovor uživatel Alice. Protější strana, tedy uživatel Cecil, potvrzuje po obdržení zprávy *BYE* ukončení spojení zasláním zprávy *200 OK*.



Obr. 22: Stavový diagram pro sestavení a ukončení spojení v IMS (teorie)

Jelikož v případě námi vytvořené sítě využívají uživatelé Cecil i Alice stejný S-CSCF server *sip:scscf.open-ims.test:6060*, nezjišťuje I-CSCF z tohoto důvodu adresu nového S-CSCF serveru. Proto není nutná ani výměna informací mezi I-CSCF a HSS pomocí zpráv protokolu Diameter LIR a LIA, jak je tomu v teoretické případě zobrazeném na obr.22. Zpráva *INVITE* protokolu SIP tak rovnou přechází na S-CSCF uživatele Alice. Stejně tak odpověď *180 Ringing* od Alice nejde jako v teoretické případě přes server I-CSCF, ale přechází z S-CSCF uživatele Alice rovnou na server S-CSCF uživatele Cecil. Tato situace je zobrazena stavovým diagramem na obr.23.



Obr. 23: Stavový diagram pro sestavení a ukončení spojení v IMS (náš případ)

5 Závěr

Tato diplomová práce je zaměřena na architekturu IMS a její možné využití. V první části je popsán vývoj IMS a jeho standardu. Poté je zde zmíněna struktura této architektury a stručně popsány její bloky. V dalším textu jsou funkce těchto bloků podrobněji rozebrány, spolu s protokoly, které IMS využívá. Dále je zde popsán vývoj, funkce a instalace open source platformy Open IMS Core. V poslední části dokumentu je popsána realizace jednoduché sítě na principu IMS, což bylo hlavní náplní této práce. V této kapitole je popsáno přidávání záznamů jednotlivých uživatelů na databázový server HSS, nastavení a funkce klientských programů, a také zachytávání provozu ve vytvořené síti.

Jak již bylo řečeno, cílem této diplomové práce bylo seznámit se s prostředím Open IMS Core, pomocí kterého lze simulovat podmínky ve skutečné architektuře IMS, a následně v tomto prostředí vytvořit jednoduchou síť založenou na principech této architektury. Nejdříve tedy bylo zapotřebí toto simulační prostředí nainstalovat. Pro možnost instalace se nabízí dva způsoby. První z nich je rychlá instalace, kterou lze provést pomocí sledu příkazů, ze kterých lze vytvořit skript. Tato metoda však vyžaduje, aby byly již dopředu přidány potřebné balíčky, a také nastavení DNS serveru Bind. Z těchto důvodů byla zvolena druhá možnost, a to instalace postupná. Při této metodě je nejprve zapotřebí vytvořit pro instalovanou aplikaci adresářovou strukturu, a poté stáhnout její zdrojový kód. Následně je třeba přidat potřebné balíčky a nastavit DNS server, který je však poté ještě třeba restartovat. Dalším krokem je nastavení uživatelských databází. Následně je již možné přejít k vlastní kompilaci zdrojového kódu. Při spouštění jednotlivých částí Open IMS Core však opět došlo k obtížím s nastavením Javy. Při pokusu o spuštění HSS aplikace hlásila chybně zadanou cestu k nástrojům Java. Jedná se o známou chybu, pro jejíž vyřešení radí instalační manuál znovu předefinovat proměnnou JAVA_HOME. Tento krok však v našem případě nepřinesl žádné zlepšení. Proto jsme se tuto chybu pokusili odstranit nahrazením balíčku java jdk 1.5 balíčkem java6 sdk. Přesto, že by měla pro správný běh aplikace stačit už java verze 1.5, vedl tento krok k úspěšnému spuštění HSS. Poté se však vyskytly další problémy s komunikací mezi HSS, I-CSCF a S-CSCF, které se bohužel nepodařilo odstranit ani po konzultacích s fraunhoferským institutem FOKUS. Z toho důvodu bylo nakonec, místo původně plánované instalace Open IMS Core na operační systém Ubuntu 7.10 – Gutsy Gibbon, použito ISO, určené pro VMware Player, s již nainstalovaným Open IMS Core na operační systém Gentoo. Pomocí něj pak byla vytvořena síť se třemi uživateli Alice, Bob a Cecil. Tito uživatelé spolu mohou navzájem sestavovat spojení a posílat si zprávy.

Pro přidání nového účastníka do sítě je nejprve zapotřebí vytvořit jeho uživatelský záznam v HSS. To lze provést buďto pomocí skriptu *add-imscore-user.sh* pro konzolové využití anebo přes webové rozhraní databázového serveru HSS. Námi bylo využito přidávání uživatelů přes webové rozhraní, které je oproti konzolovému uživatelsky příjemnější. Vlastní komunikace mezi účastníky pak probíhá pomocí klientských programů. Pro účely této práce byl použit program OpenIC_Lite, který nabízí využití jak v konzolovém, tak také v grafickém režimu. V poslední části dokumentu jsou uvedeny možnosti zachycení provozu ve vytvořené síti. Nejrozsáhlejší možnosti

nabízí program Wireshark, který umožňuje nejen zachycení paketů v síti, ale také jejich následnou analýzu. Jinou možností, kterou lze pro tento účel využít, je zachycení dat pomocí webového rozhraní HSS, které umožňuje zaznamenávat příchozí provoz na tento databázový server. Funkci pro zachytávání provozu mezi terminálem a proxy serverem P-CSCF a následné zobrazení obsahu zachycených paketů nabízí také grafické rozhraní klientského programu OpenIC_Lite. Za pomoci těchto programů byl zachycen provoz v síti při registraci uživatele, sestavování a ukončování spojení. Ani to však nebylo zcela bez problémů, neboť program Wireshark, přesto, že měl protokoly SIP i Diameter přidány do své databáze, označil protokol všech zachycených paketů jako *unknown*. Proto bylo nutné rozpoznávat tyto pakety podle obsahu jejich hlavičky. Ze získaných dat pak byly vytvořeny stavové diagramy.

Použitá literatura

[1] IMS. *IEEE Vehicular Technology magazine*. 2007, vol. 2, no. 1, s. 2-42.

[2] IMS Infrastructure and Services. *IEEE Communications magazine*. 2007, vol. 45, no. 3, s. 112-159.

[3] *Fundamentals of the IMS*. [s.l.] : [s.n.], [2007]. 253 s.

[4] *BerliOS The Open Source Mediator* [online]. 2007 [cit. 2007-12-08]. Dostupný z WWW: <www.berlios.de>.

[5] *UCT IMS Client* [online]. 2007 [cit. 2007-12-08]. Dostupný z WWW: <http://uctimsclient.berlios.de/openimscore_on_ubuntu_howto.html>.

[6] *FOKUS - Institut* [online]. 2007 [cit. 2007-12-08]. Dostupný z WWW: <http://www.fokus.fraunhofer.de>.

[7] *Protokol SIP ve zkratce - LUPA* [online]. 2004 [cit. 2007-12-08]. Dostupný z WWW: <http://www.lupa.cz/clanky/protokol-sip-ve-zkratce/>.

[8] HANUS, Stanislav. *Bezdrátové a mobilní komunikace*. 1. dotisk vyd. Praha : RadioMobil, 2003. ISBN 80 – 214 – 18. Zabezpečení informací proti zneužití, s. 97-100.

[9] HRUBÝ, Petr. Protokol SIP (I) - Architektura, typy zpráv a identifikace. *ISDN* [online]. 2003 [cit. 2008-05-20]. Dostupný z WWW: <www.isdn.cz>.

Seznam příloh

Příloha 1: CD s diplomovou prací ve formátu .pdf a .doc a použitými obrázky