

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra telekomunikační techniky



Diplomová práce

Spolupráce mezi systémy Cisco Unified Call Manager a Asterisk

Bc. Jan Kropáček

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jan Kropáček**

Studijní program: Komunikace, multimédia a elektronika
Obor: Komunikační systémy

Název tématu: **Spolupráce mezi systémy Cisco Unified Call Manager a Asterisk**

Pokyny pro vypracování:

Prostudujte do dostatečné hloubky oba systémy uvedené v názvu práce a zhodnoťte možnosti vzájemné spolupráce tak, aby bylo možno např. prostřednictvím Asterisku řešit náhradu některých funkcí CUCM, např. pokud je licenční a cenová politika Cisca nepříznivá k nákupu originálních funkcí přímo v tomto produktu, nebo pro možnost úzké vzájemné spolupráce již existujících instalací obou systémů. Experimentálně ověřte své závěry na instalacích obou systémů.

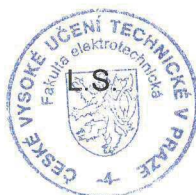
Seznam odborné literatury:

- [1] Finke, J.; Hartmann, D.: *Implementing Cisco Unified Communications Manager (Part 1)*. Cisco Press 2010. ISBN: 978-1-58720-418-0.
- [2] Olsen, Ch.: *Implementing Cisco Unified Communications Manager. Pt.2*, Cisco Press 2011.
- [3] Online dokumentace systému Asterisk [on-line]

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2016/2017

prof. Ing. Boris Šimák, CSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 12. 2015

Abstrakt

Tato práce se zabývá možností spojením telefonních ústředen Cisco Unified Call Manager (CUCM) a Asterisk. Spojení provedeno pomocí protokolu SIP. Po tomto spojení byly realizovány služby voicemail a callcentrum. Práce také obsahuje popis protokolu SIP.

Klíčová slova

Cisco Unified Call Manager (CUCM), Asterisk, SIP, voicemail, callcentrum

Abstract

This thesis deals with the possibility of connecting Cisco Unified Call Manager (CUCM) and Asterisk PBXs. The connection is made using the SIP protocol. On this connection was implemented a voicemail and a callcenter services. The work also contains a description of the SIP protocol.

Keywords

Cisco Unified Call Manager (CUCM), Asterisk, SIP, voicemail, callcentr

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu. A že jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

.....
Bc. Jan Kropáček
9. 1. 2018

Poděkování

Chtěl bych poděkovat panu Ing. Pavlu Trollerovi, CSc., za jeho vedení a pomoc při tvorbě této práce. Dále bych chtěl poděkovat Ing. Jánmu Kučerákovi a kolegovi Ing. Petru Šimberovi, za půjčení potřebného hardwaru pro fyzickou realizaci této práce. A také bych chtěl poděkovat rodičům a přátelům za jejich pomoc a podporu při tvorbě této diplomové práce.

Obsah

Obsah	1
1 Úvod	3
2 Historie telefonie	4
2.1 Generační vývoj telefonních ústředen	5
3 Asterisk	6
3.1 Telefonní ústředna Asterisk	7
3.1.1 Hardware	7
3.1.2 Lokální operační systém	8
3.2 Architektura Asterisku	9
3.2.1 Jádru (core) Asterisku	9
3.2.2 Aplikační rozhraní (API)	10
4 Cisco Unified Call Manager	11
4.1 Infrastruktura	12
4.2 Cisco Unified Communications Manager	13
4.3 Aplikační vrstva	15
5 Protokol SIP	16
5.1 Signalizace	18
5.1.1 Žádosti	18
5.1.2 Odpovědi	19
5.2 Sestavení a ukončení hovoru protokolem SIP	23
6 Voicemail	25
6.1 Řešení společnosti Cisco	25
6.2 Řešení Asterisk	25
7 Callcentrum	27
7.1 Řešení společnosti Cisco	27
7.2 Řešení Asterisku	27
7.2.1 IVR	27
7.2.2 Fronta hovorů	27
7.2.3 Agent	29
8 Instalace	31
8.1 Seznam otestovaných nefunkčních konfigurací serverů	31
8.2 Fungující řešení	32
8.3 Virtuální počítač pro CUCM	33
8.4 Virtuální počítač pro Asterisk	34

8.5	Ostatní hardware.....	34
8.6	Navrhnuté řešení.....	34
9	Nastavení CUCM	35
9.1.1	Nastavení SIP Trunku pomocí GUI.....	35
9.1.2	Nastavení Voicemailu.....	37
10	Nastavení Asterisku	38
10.1	Nastavení SIP.....	38
10.2	Nastavení voicemailu	38
10.3	Nastavení callcentra	39
11	Závěr	41
	Literatura	43
	Seznam použitých zkratk.....	45

1 Úvod

Touha lidí komunikovat na velkou vzdálenost je stará jako lidstvo samo. V prvních pokusech se jednalo o systémy založené na předávání zpráv pomocí posílů či signálu. Postupem času se z těchto metod vyvinula telefonie.

Jako telefonie se označuje obousměrný způsob přenosu lidského hlasu na velkou vzdálenost v reálném čase. Telefonie se zajišťuje pomocí telefonní technologie. Ta zahrnuje koncová zařízení (telefony) a přenosové prostředí (telefonní síť).

V této práci si budu zabývat možností propojení dvou rozdílných systémů. Tak aby navzájem správně komunikovali a doplňovali se. Jedná se o spojení Cisco Unified Communications Manageru a Asterisku.

Cílem této práce je v praxi zprovoznit některou službu, která chybí v základní instalaci Cisco Unified Communications Manageru za pomoci Asterisku.

Po seznámení s Cisco Unified Communications Managerem a Asteriskem jsem se rozhodl pro zprovoznění služeb voicemail a callcentrum. Jde o služby, které v Cisco Unified Communications Manageru chybí. Sice je lze od společnost Cisco dokoupit, ale jejich cena bude akceptovatelná spíše pro větší firmy.

2 Historie telefonie

První zmínky o pokusech přenosu hlasu na větší vzdálenost jsou z 10. století. Šlo o různé vedení hlasu pomocí tvarovaných trubek. Například spojení můstku lodi se strojovnou. Pokusy s vedením zvuku pomocí elektrického signálu se datují do druhé poloviny 19. století. Pokusy s přenosem hlasu se zabývalo více vynálezců jako třeba italský vědec Antonio Meucci a jeho telefon z roku 1857. První telefon byl ovšem patentován v roce 1876 americkým vynálezcem Alexandrem Grahamem Bellem. A proto bývá považován za jeho vynálezce. Telefon se signalizací (zvonkem) spatřil světlo světa až o několik let později.

Telefony se začaly uplatňovat nejdříve v oblastech dopravy (především na železnici) a průmyslu, kde nahrazovaly telegrafní spojení. Se vzrůstající oblibou telefonů se postupně začaly zavádět podnikové a městské telefonní sítě.

V České republice se první telefony začaly objevovat po první světové válce. Nejdříve na železnici a ve velkých továrnách, později ve větších městech pod křídly Správy pošt a telegrafů.

2.1 Generační vývoj telefonních ústředen

Ústředny 1. generace jsou elektromechanické. Měly decentralizované, dekoncentrované a většinou synchronní řízení.

Ústředny 2. generace se objevují od 70. let 20. století. Jejich řízení se realizovalo pomocí registrů a určovatelů a bylo asynchronní, centralizované a částečně koncentrované. Ve spojovacím poli byly obsaženy křížové spínače, které byly využívány i pro připojování řídicích prvků.

Ústředny 3. generace se objevují od 80. let 20. století. Byly už polo-elektronické a elektronické ústředny. Křížové spojovací pole obsahovalo křížové a kódové voliče, jazýčková relé, doutnavky, diskretní polovodičové prvky (diody, tyristory, tranzistory) a později i integrované obvody. Řízení bylo centralizované a provedeno pomocí počítače.

Ústředny 4. generace objevují od 90. let 20. století. Účastnická sada je již digitalizována a signál se přenáší buď v nekomprimované podobě (PCM) nebo v komprimované podobě (Alaw či μ -law). Spojování ve spojovém poli probíhá pomocí čtyř drátů (2 vodiče tam a 2 zpátky). Spojování probíhá pomocí S článků (prostorově) i T článků (časově). Používá se signalizace SS7.

Ústředny 5. generace využívající IP telefonie (VoIP). Účastnický okruh již není vystavěn po celou dobu hovoru (jako u předchozích generací), ale hovor je přenášen v paketech (malé kousky dat) skrze IP síť. Většinou už neobsahují spojovací pole, protože ho nahrazují IP síťe. Hlavní funkcí ústředny je především zajištění signalizace (SIP, H.323, IAX a MGCP). Přenos hovoru nemusí být vždy realizován přes ústředny.

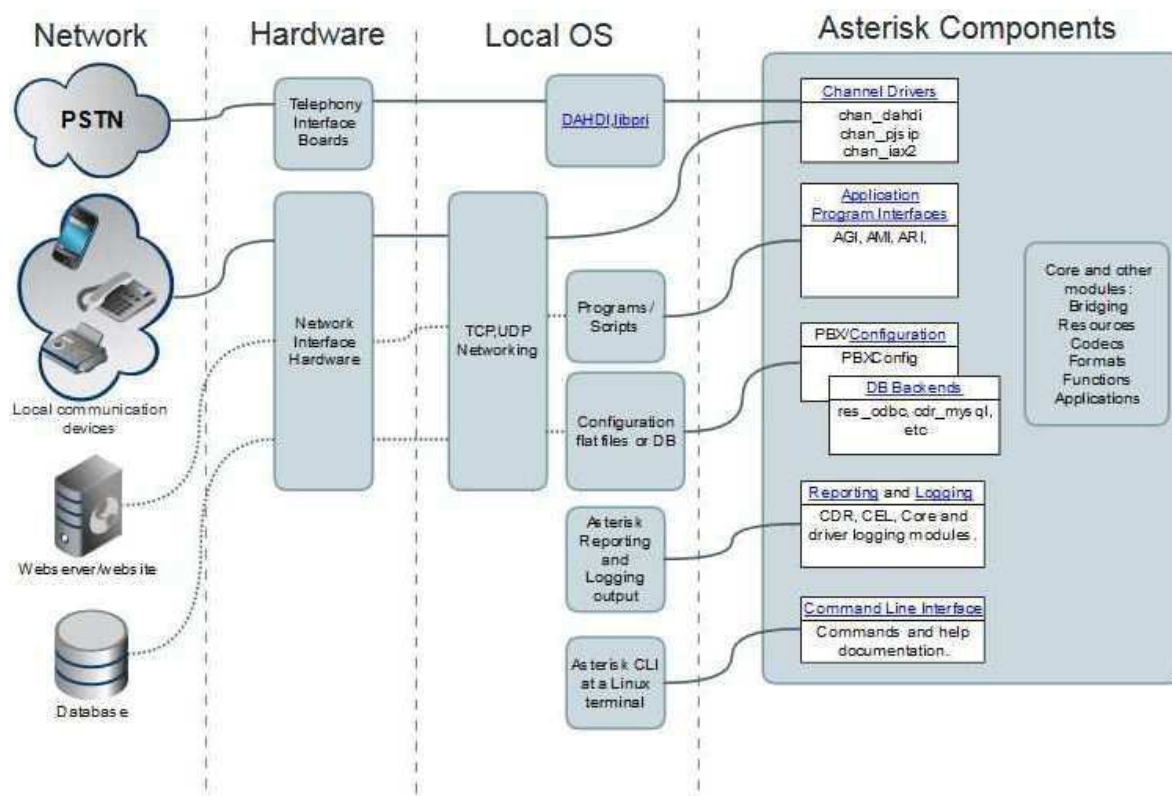
3 Asterisk

Asterisk je linuxová open-source softwarová pobočková telefonní ústředna (PBX). Tudiž se dá provozovat na široké škále hardwaru, lze si ji programováním přizpůsobit pro individuální potřeby a protokoly a její pořizovací cena je nízká.

O její vznik se roku 1999 zasloužil Mark Spencer, který podle svých slov neměl peníze na komerční produkt PBX ústředny a tak si vytvořil vlastní. Asterisk je nabízen pod podmínkami GPL (General Public Licence). To znamená, že je volně přístupná na internetu a je možné zdrojové kódy (v programovacím jazyku C) upravovat k vlastní potřebě. Na celý projekt dohlíží firma Digium, která tedy nevydělává na vývoji ústředny, ale na její podpoře a prodeji certifikovaného hardwaru.

3.1 Telefonní ústředna Asterisk

Ústředna (server) Asterisk se skládá ze 3. částí. Je to samotný software Asterisk, lokální operační systém a hardware.



Obrázek 1: Zjednodušené schéma Asterisku

3.1.1 Hardware

Slouží k propojení ústředny s okolními sítěmi. Pro analogové spojení se využívají dva druhy hardwaru.

3.1.1.1 ZapTel hardware

Asterisk nativně podporuje ZapTel hardware. ZapTel hardware je pseudo TDM rozhraní. Originální TDM (Time-Division Multiplex) hardware je patentován a je tudíž nákladný. Proto firma ZapTel začala vyrábět vlastní verze TDM rozhraní. Vyznačuje se kvalitou a real-time zpracováním jako originální TDM hardware.

3.1.1.2 Non-Zaptel hardware

Jde o rozhraní bez podpory TDM. Např.: ISDN4Linux, OSS/Alsa, LTI - Linux Telephony Interface.

3.1.1.3 Packet voice

K připojení ústředny Asterisk do paketových sítí (IP nebo Frame relay) není potřeba žádný speciální hardware. Asterisk má svůj vlastní protokol a to IAX. Ten se stará o přenášení signalizace k sestavení hovoru i o transport hlasových dat mezi komunikujícími stranami. Díky snaze o co nejuniverzálnější pojetí, jsou implementovány další často používané signalizace jako SIP, H.323, MGCP, Skinny/SCCP.

3.1.2 Lokální operační systém

Stará se o komunikaci mezi Asteriskem a hardwarem, pro uložení konfigurace či ukládání dat.

3.2 Architektura Asterisku

Architektura systému je velmi jednoduchá. Asterisk je v podstatě středový prvek spojující telefonní technologie (VoIP nebo TDM technologie) s telefonními aplikacemi (např.: konference, hlasovou poštu a další).

Kolem systému centrálního jádra PBX jsou definovány různé aplikační rozhraní (API). Toto pokročilé jádro ovládá vnitřní propojení specifických protokolů, kodeků a hardwarového rozhraní telefonních aplikací. Toto zaručuje případnou rozšiřitelnost o další nová rozhraní do budoucna.

3.2.1 Jádro (core) Asterisku

Jádro spravuje tyto bloky:

1. *PBX přepojování (PBX Switching)* - Hlavní funkcí Asterisku je samozřejmě přepojovací systém pobočkové ústředny, spojování volání mezi různorodými uživateli a automatizovanými úlohami. PBX přepojovací jádro transparentně spojuje příchozí volání na různých hardwarových a softwarových rozhraních.
2. *Spouštěč aplikací (Application Launcher)* - Spouští aplikace zajišťující služby jako jsou například hlasová pošta, přehrání souboru nebo výpis adresáře.
3. *Překladač kodeků (Codec Translator)* - Používá moduly kodeků pro kódování a dekódování zvukových kompresních formátů používaných v telefonním prostředí. Množství dostupných kodeků je vhodné pro různorodé potřeby a docílení rovnováhy mezi zvukovou kvalitou a šířkou pásma.
4. *Plánovač a I/O manažer (Schedule and I/O manager)* - Ovládání rozvrhování nízkourovňových úloh a systémového řízení pro optimální výkon podle stavu zatížení.

3.2.2 Aplikační rozhraní (API)

Jsou definovány čtyři API, které usnadňují oddělení hardwaru a protokolů. Z toho vyplývá, že jádro Asterisku se nemusí zabývat tím, jak se daný účastník připojil, jaké používá kodeky atd..

Čtyři API jsou:

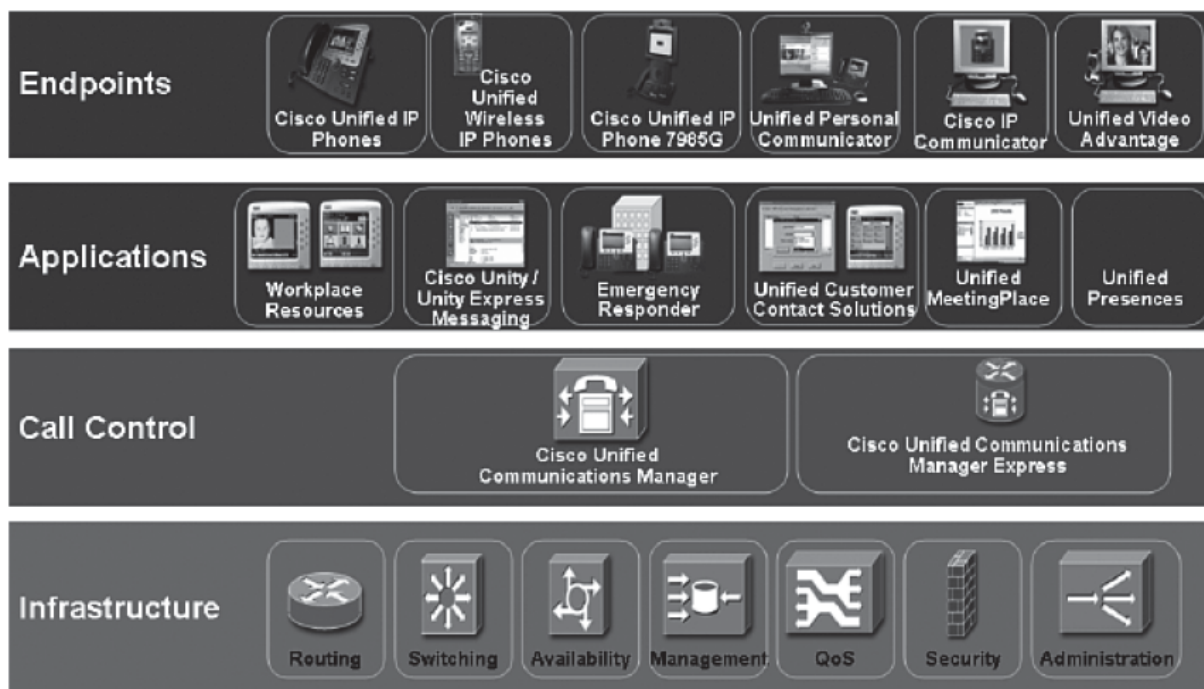
1. *Kanálové API* - Ovládá typ spojení příchozího volání, tedy jedná-li se o VoIP spojení, ISDN, PRI nebo nějakou jinou další technologii. Dynamické jednotky jsou zavedené pro ovládání detailů nižších vrstev těchto spojení.
2. *Aplikační API* - Ovládá různorodé jednotky úkolů, které by PBX systém mohl vykonávat (teď nebo v budoucnu). Například funkce jako konferenční hovor, výpis adresáře, hlasová pošta, přímý přenos dat a další.
3. *API překladače kodeků* - Zavádí moduly kodeků pro podporu různých audio formátů jako G.722, G.726, GMS, μ -law, A-law, MP3 .
4. *API souborových formátů* - Provádí zápis a čtení různých souborových formátů a prokládání dat v souborovém systému.

Modulární forma dovoluje systému hladkou integraci a to jak aktuálně implementovaného hardwaru přepojované telefonie, tak rychle rostoucí hlasové technologie na bázi paketů. Ještě důležitější je tato schopnost u modulů s kodeky. Umožňuje přidat jakýkoli kodek. To šetří náklady při přechodu na nový kodek, stejně tak lze zvětšovat či zmenšovat kvalitu přenášeného zvuku. Aplikační API dovoluje vyvinout a připojit libovolné služby.

4 Cisco Unified Call Manager

Cisco Unified Call Manager, dnes již Cisco Unified Communications Manager zkráceně CUCM nebo i UCCM, je řešení firmy Cisco pro telefonii. Tato ústředna spadá do programu (řešení) Cisco Unified Communications (Cisco UC). Jde o řešení zapadající do sítě navržené a provozované nad zařízením od firmy Cisco, které se zabývá skloubením přenosu zvuku, videa a dat v jedné síti. CUCM má v tomto řešení odpovídat za zpracování hovorů, ovládání zařízení, správu plánu čísel a ostatních funkcí.

Toto řešení se díky své velikosti a cenou hodí spíše pro střední a velké podniky.



Obrázek 2: Schéma programu Cisco Unified Communications

4.1 Infrastruktura

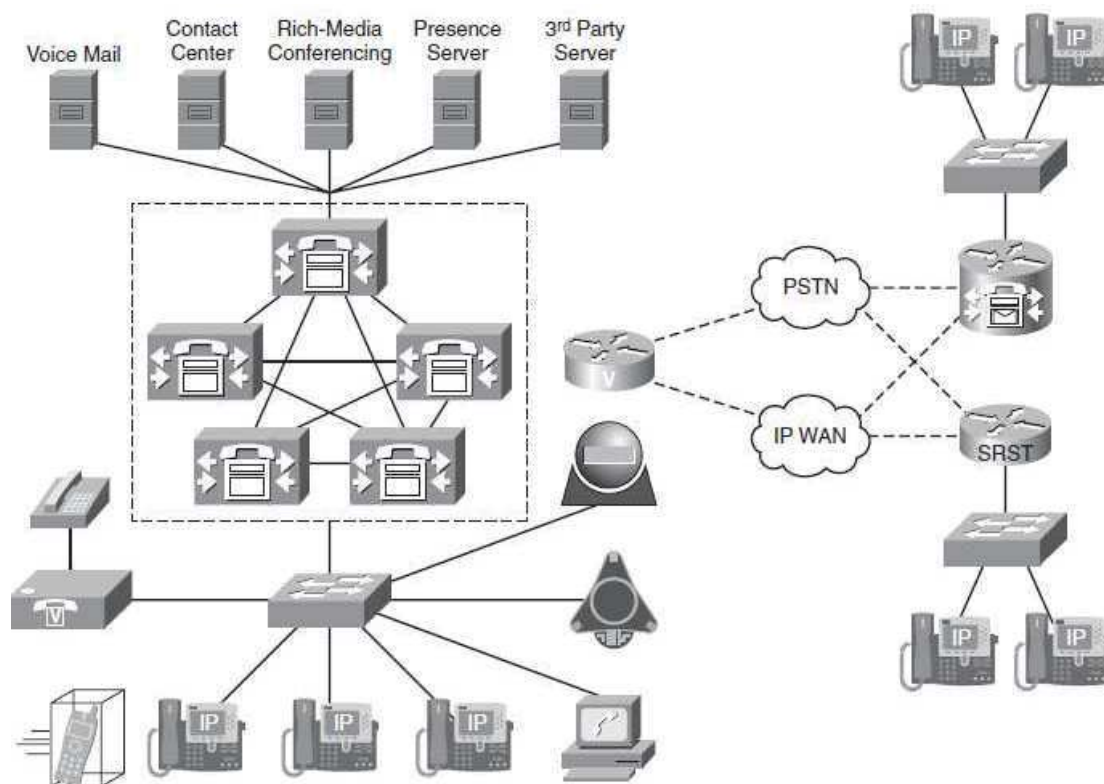
Nejdůležitější prvek této vrstvy z pohledu telefonie je hlasová brána (voice gateway). Pomocí ní se propojí IP síť se sítí telefonní. V architektuře Cisco však není úplně povinná. Pokud jsou všichni uživatelé připojeni přes VoIP (IP telefony) není hlasová brána potřeba. Komunikace s mimo firemními uživateli bude zasílána rovnou operátorovi ve formě datového proudu přes IP síť. Jako hlasové brány se používají buď speciální hlasové brány (např.: VoIP GSM gateway) nebo i klasické Cisco routery (například série 2800 či 3800). Analogová zařízení jako fax, modem a analogový telefonní přístroj se také připojují na hlasovou bránu. Nevýhoda tohoto přístupu je špatná (nákladná) změna popřípadě problém s přidání analogových portů.

Tím, že je CUMC plně integrován do řešení UC, se tato vrstva postará i o dostupnost, kvalitu (QoS) a bezpečnost sítě.

4.2 Cisco Unified Communications Manager

CUMC je nasazen na speciálně certifikované servery. Tyto servery mohou běžet jak pod operačním systémem Linux tak i Windows, záleží na verzi CUMC. Servery dodávají firmy IBM a HP.

Pro komunikaci používá Cisco primárně svůj protokol *Skinny Client Control Protocol* (SCCP), ale podporuje také (hlavně u nových verzí) H.323, SIP, MGCP. Dále se pro získání informací nebo ovládání CUMC můžeme použít protokoly JTAPI či SOAP.



Obrázek 3: Ciscem doporučené zapojení CUCM

Cisco Unified Communications Manager nabízí:

1. *Zpracování hovorů* – Skládá se ze všech aspektů jako: vytáčení, směrování, ukončení hovoru, ale i účtování a sběru statistik.
2. *Signalizace a ovládání zařízení*
3. *Správa plánu čísel* - Jde o seznam jednotlivých uživatelů a jejich čísel.
4. *Správu uživatelů* – Pro snadnou správu uživatelů.
5. *Telefonní funkce* - CUCM rozšiřující služby, jako je přidržení hovoru, přesměrování, konferenční hovory, rychlé vytáčení, opakované vytáčení a mnoho dalších funkcí pro IP telefony
6. *Programovatelné rozhraní pro externí aplikace* - Poskytuje programovatelné rozhraní pro externí aplikace, jako je Cisco IP softphone, Cisco IP Communicator, Cisco Unified IP Interactive Voice Response (IP IVR), Cisco Personal Assistant, Cisco Unified Personal Communicator, a CUCM Attendant Console.
7. *Nástroje pro zálohování a obnovení* - Poskytuje systém obnovy po havárii (DRS) se zálohováním a obnovou konfigurační databázi CUCM. Systém DRS také zálohuje CDR (Call Details Records – podrobnosti o hovoru), CMR (Call Records Management) a databázi analýzy CDR a reporting CAR.

4.3 Aplikační vrstva

Je nezávislá na hovorové (call-control) vrstvě a tím umožňuje nasadit aplikace kamkoliv do sítě.

Umožňuje například:

1. *Hlasovou poštu, textovou komunikaci (IM nebo unified messaging application)*
2. *Kontaktní centrum (Cisco Unified Contact Center)*
3. *Přednášky a videokonference (Cisco Unified MeetingPlace)*
4. *Přednostní volání do integrovaného záchraného systému (Cisco Emergency Responder)* – Dovede identifikovat volajícího i jeho polohu.
5. *Rozhraní pro standardní protokoly* - jako Telephony Application Programming Interface (TAPI), Java Telephony Application Programming Interface (JTAPI), Simple Object Access Protocol (SOAP), Q.SIG, H.323, Media Gateway Control Protocol (MGCP) a SIP. Ty jsou k dispozici pro podporu aplikací třetích stran.

5 Protokol SIP

Vývoj protokolu Session Initiation Protocol (SIP) byl zahájen pracovní skupinou MMUSIC (Multiparty Multimedia Session Control) v rámci IETF (Internet Engineering Task Force) roku 1996. V roce 1999 byl uvolněn ve formě navrhovaného standardu v RFC 2543. Téhož roku na popud IETF vznikla nová pracovní skupina, nazvaná příznačně SIP. Právě tato skupina převzala vývoj hlavního jádra protokolu. Toto úsilí vyústilo v nový standard RFC 3261.

Pro přenos hlasu přes datové sítě v současné době jsou používána převážně dvě řešení:

1. Rodina protokolů implementujících doporučení ITU-T H.323. Využívá binární formát založený na ASN.1, což představuje problém při potřebě sledování a ladění provozu aplikací.
2. Protokol SIP od IETF. Prezentuje data textově.

Koncepce SIP protokolu je podobná jako u HTTP (Hypertext Transfer Protocol - využívaný službou World Wide Web) nebo u SMTP (Simple Mail Transfer Protocol - protokol pro přenos elektronické pošty). Protokol SIP má za úkol řízení navázání, průběhu a ukončení spojení s jedním nebo více účastníky v síti založených na technologii IP (Internet Protocol). Mezi relace mohou zařadit multimediální konference, hovory přes IP síť, sdílení multimediálních dat. Účastníci spolu komunikují přes skupinové vysílání (multicast - všesměrové vysílání, kde kopie paketu jsou doručeny určené skupině koncových uzlů), nebo více spojů typu bod-bod (unicast – vysílání pouze mezi dvěma body). Možná je i kombinace obou možností.

SIP je aplikační protokol. Jeho činnost proto závisí na protokolech nižších vrstev. Jde hlavně o protokol RTP (Real-time Protocol), adresářové služby spojené s LDAP (Lightweight Directory Access Protocol), signalizační protokoly pro komunikaci s telefonní sítí například SS7 (Signaling System Number 7), garanci kvality řečového signálu pomocí RSVP (Resource Reservation Protocol), autentizaci uživatelů apod.

SIP přesně nespecifikuje užití protokoly. Příkladem je třeba výše uvedený RTP; ten může pracovat jako spojově orientovaný (nad TCP) nebo jako nespojovaný (nad UDP), podle použitého typu protokolu transportní vrstvy (TCP - Transmission Control Protocol nebo UDP - User Datagram Protocol). Díky této volnosti specifikace SIP máme možnost výběru použitých protokolů. Většinou se používá protokol UDP, který má jednoduchou

implementaci a u přenosu zaměřeném na doručení dat v reálný čas nezáleží příliš na spolehlivosti přenosu, jako spíše na rychlosti doručení.

Protokol SIP také není pevně svázán s žádnými konkrétními protokoly pro přenos multimediálních dat. Uvnitř zprávy protokolu SIP při navázání spojení je proto zapouzdřena zpráva jiného protokolu, který specifikuje použitá kódování pro multimediální data, jejich parametry a čísla portů, na kterých mají být data vysílána nebo přijímána. Nejčastěji se pro tento účel používá SDP (Session Description Protocol), který je rovněž textový.

Jednotlivé zprávy se sestávají z posloupnosti textových hlaviček. Vytváření a rozpoznávání těchto zpráv na straně odesílatele resp. příjemce je jednodušší než u binárních zpráv. Dekódování a práce s textovou hlavičkou ulehčuje použití některého z vyšších programovacích jazyků. Výhodou proto je i to, že protokol může být snadno rozšiřován přidáváním nových hlaviček, specifikovaných jako samostatné normy RFC.

5.1 Signalizace

Protokol SIP zná dva druhy zpráv:

1. Žádosti - říká se jim též metody.
2. Odpovědi – říká se jim též návratové kódy.

5.1.1 Žádosti

Základní žádostí jsou:

INVITE - žádost o navázání spojení nebo o změnu parametrů již existujícího spojení.

BYE - žádost o ukončení spojení

ACK - žádost, kterou klient potvrzuje, že obdržel odpověď na žádost INVITE

REGISTER - žádost o registraci klienta u registračního serveru

CANCEL - žádost o zrušení probíhající žádosti INVITE

OPTIONS - žádost o zaslání přehledu funkcí podporovaných serverem

INFO - znamená přenos informací během hovoru

UPDATE - dovoluje klientovi aktualizovat parametry spojení

PRACK - dočasné potvrzení

SUBSCRIBE - znamená přijímání/odběr událostí (textové zprávy)

NOTIFY - informuj (uvědom) účastníky

MESSAGE - zpráva

5.1.2 Odpovědi

Návratový kód je třímístné číslo kódující výsledek žádosti obdobně, jak je tomu třeba v protokolu HTTP. Dají se rozdělit do šesti skupin podle první číslice. Uvedu zde tyto skupiny a pár kódů které obsahují:

1xx - žádost přijata, pokračuji ve zpracování žádosti

100 Zkouším to dál (*Trying*). Žádost byla přijata serverem a byly provedeny kroky k jejímu splnění.

180 Vyzvánění (*Ringin*g). Uživatelský agent přijal žádost INVITE a nyní se snaží upozornit uživatele na příchozí hovor.

181 Volání je přesměrováno (*Call Is Being Forwarded*). Server oznamuje přesměrování hovoru na některý jiný server.

182 Zařazen ve frontě (*Queued*). Volaná strana je dočasně nedostupná a server se rozhodl zařadit žádost do fronty místo zamítnutí volání. V okamžiku, kdy je volaný opět dostupný, vyšle se vhodná finální odpověď dle situace (odmítnout/přijmout).

2xx - znamená úspěšné provedení žádosti

200 OK. Žádost byla v pořádku zpracována. Informace přenesená v odpovědi závisí na typu žádosti, která tuto odpověď vyvolala.

3xx - označuje přesměrování (odpověď od redirect serveru)

300 Více možností (*Multiple Choices*). SIP adresa uvedená v žádosti byla analyzována a rozložena na několik možných lokací volaného. Volající uživatel (potažmo UA) může upřednostnit jednu z lokací (koncových bodů) a pak přesměrovat svůj požadavek na zvolený koncový bod. V těle odpovědi by samozřejmě měl být uveden seznam, podle kterého se volající rozhodne.

301 Trvale přesunut (*Moved Permanently*). Volaný uživatel už není na cílové adrese k nalezení. Volající by měl zopakovat svou žádost na adresu získanou z odpovědi v hlavičce "Contact:". Stejně tak by měl nahradit záznam v adresáři

("seznam kontaktů") aktualizovanou informací.

302 Běžné přesměrování redirect serverem (*Moved Temporarily*). Je možné specifikovat dobu, po kterou je adresa, na kterou se přesměrování provede, platná.

380 Poskytnutí náhradní služby (*Alternative Service*). Volání nebylo úspěšné, ale server nabízí možnost náhradní služby. Typ služby je popsán v těle odpovědi.

4xx - chyba způsobená klientem (chybný formát žádosti)

400 Poškozená metoda (*Bad Request*). Žádost nemůže být zpracována díky zjevným syntaktickým chybám. Například chybí hlavička "Call-ID:"

401 Pokus o neautorizovaný přístup (*Unauthorized*). Žádost si vynucuje autentizaci uživatele. Tato odpověď je generována UAS a registračními servery. Proxy servery používají odpověď 407 (viz níže).

403 Zakázaná žádost (*Forbidden*). Server rozuměl žádosti, ale odmítá jí vyhovět/zpracovat. V RFC se uvádí, že by neměla být opakována.

404 Volaný uživatel se nenachází na daném serveru (*Not Found*). Server má konečnou informaci, že volaný uživatel v zadané doméně neexistuje. Stejně tak se tato odpověď zašle po žádosti obsahující doménu odlišnou od těch, které daný SIP server spravuje.

405 Žádost není povolena (*Method Not Allowed*). Server správně rozpoznal žádost, ale její zpracování je pro adresu ze které žádost přišla zakázáno. Aby klient věděl, které žádosti má povoleny, server je zašle v těle odpovědi (toto chování je u všech serverů povinné).

406 Nelze přijmout (*Not Acceptable*). V žádosti je uveden seznam charakteristik očekávaného datového toku, které kolidují s možnostmi serveru.

407 Nutná autentizace u proxy (*Proxy Authentication Required*). Klient se musí nejprve autentizovat u proxy serveru. Odpověď je podobná s "Unauthorized" výše.

- 408** Překročení doba odezvy (*Request Timeout*). Server není schopen reagovat v rozumném čase, například nestihne lokalizovat uživatele. Klient může zopakovat svou žádost beze změn.
- 410** (*Gone*). Žádaný zdroj není dále k dispozici a navíc není známa jeho stávající adresa pro přesun žádosti. Zde se předpokládá trvalý stav. Avšak, pokud server sám o sobě nemá možnost zjistit, je-li situace opravdu trvalá, potom by se se místo "Gone" měla generovat odpověď 404 - "Not Found".
- 413** Překročení délky zprávy (*Request Entity Too Large*). Tělo zprávy je delší než je server schopen (nebo ochoten) zpracovat.
- 414** Dlouhá adresa cíle (*Request-URI Too Large*). Server odmítl poskytnout službu z důvodu příliš dlouhé hlavičky "Request-URI".
- 415** Neznámý formát zprávy (*Unsupported Media Type*). Zde server opět odmítá žádost, nyní však z důvodu nemožnosti analýzy těla zprávy. Jednoduše - formát zprávy není na serveru podporován. Server musí naoplátku vrátit účastníkovi seznam podporovaných formátů.
- 416** Neznámý formát URI (*Unsupported URI Scheme*). Podobný případ jako předchozí.
- 423** Krátký interval (*Interval Too Brief*). Požadavek nelze splnit v žádané době. Použití této odpovědi spadá do oblasti registračního serveru. Žádost je zamítnuta.
- 480** Dočasně nedostupný (*Temporarily not available*). Bylo úspěšně navázáno spojení s koncovým bodem, ten však v té chvíli není dostupný. To znamená například, že není nalogován v systému, má nastaveno "nerušit", nebo je nalogován ale zrovna není ve stavu přijmout hovor. Odpověď by měla specifikovat čas pro další pokus o volání. Jiné použití je u proxy a redirect serverů, které touto odpovědí dávají najevo situaci, kdy volaný uživatel nemá v dané chvíli definovanou doménu kam hovor přesměrovat.

5xx - chyba způsobená serverem

500 Vnitřní chyba serveru (*Internal Server Error*)

501 Požadovaná funkce není implementována (*Not Implemented*)

503 Služba není dostupná (*Service Unavailable*)

504 Vypršel čas pro odpověď (*Server Time-out*)

505 Použitá verze SIP není podporována (*SIP Version not supported*)

513 Příliš dlouhá zpráva (*Message Too Large*)

6xx - obecná chyba, žádost nemůže být akceptována ani jiným serverem

600 (*Busy Everywhere*)

603 (*Decline*)

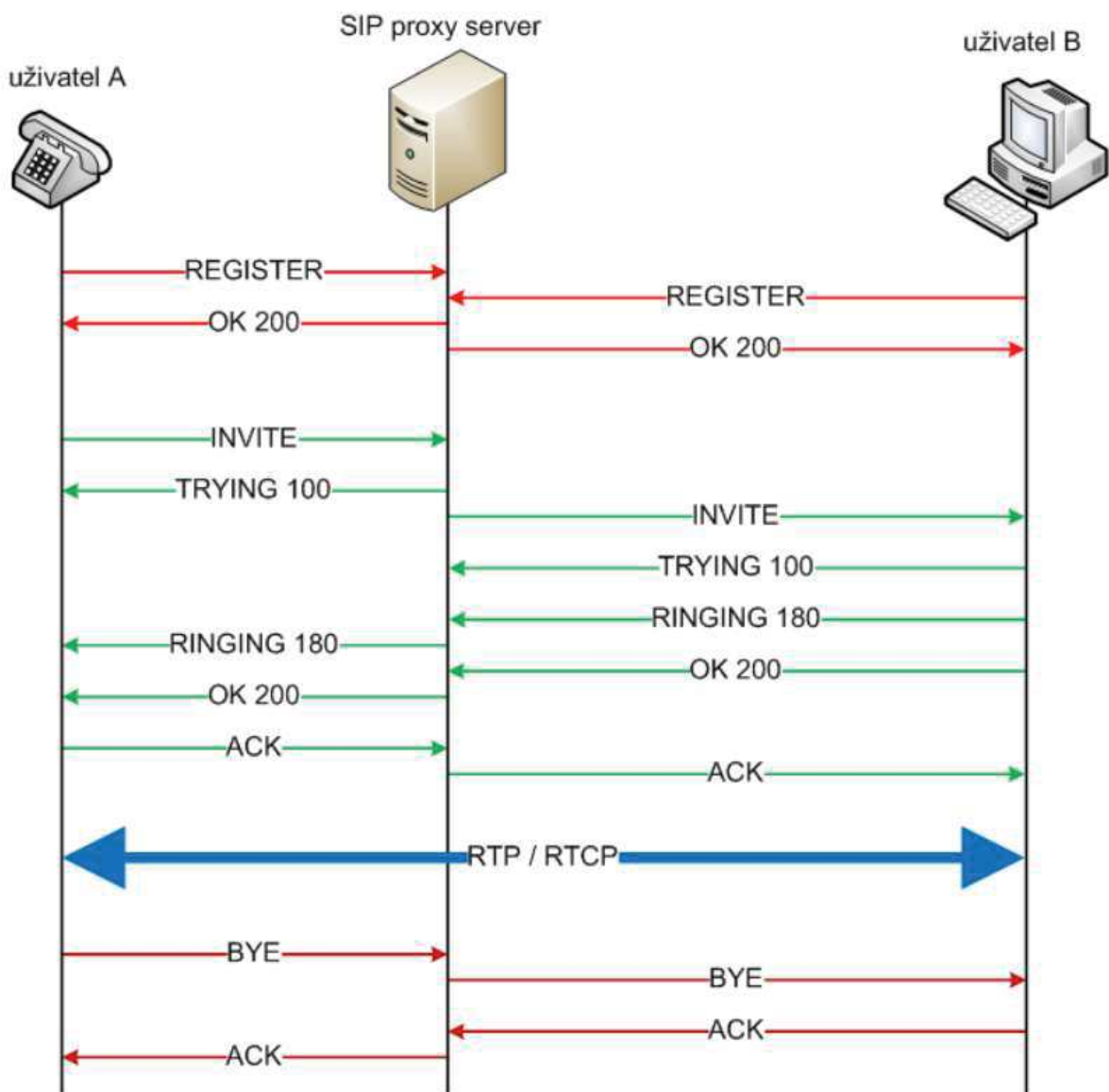
604 (*Does not exist anywhere*)

606 (*Not Acceptable*)

Nejběžnější odpovědi jsou "200 OK", "302 Moved Temporarily" a "404 Not Found".

5.2 Sestavení a ukončení hovoru protokolem SIP

Ukážeme si jednoduchý příklad základní funkce protokolu SIP. Příklad ilustruje schéma základní funkce protokolu na obrázku 4. V praxi se k celému procesu obvykle ještě přidává registrace a odchozí požadavek s autentizací nebo třeba více proxy serverů v cestě požadavku.



Obrázek 4: Sestavení a ukončení hovoru protokolem SIP

Nejprve musí každý koncový bod registrovat na vlastním registrar serveru (pomocí zprávy REGISTER), aby ostatní účastníci sítě byli schopni tento koncový bod na základě jeho názvu ho lokalizovat. Uživatel A odešle požadavek na spojení s jiným koncovým bodem na svůj proxy server (pomocí zprávy INVITE). Proxy server odpoví zprávou 100 - TRYING. Následně se pokusí lokalizovat koncový bod specifikovaný v INVITE zprávě. Po úspěšné lokalizaci předá INVITE požadavek cílovému koncovému bodu. Ten nejprve odpoví zprávou 100 - TRYING a následuje oznámení o vyzvánění 180 - RINGING, čímž informuje server o čekání na zahájení hovoru od uživatele B. Tato zpráva je předána na počáteční bod spojení, který uživateli A signalizuje vyzvánění na přístroji uživatele B. Přijetí hovoru uživatelem B je z jeho strany oznámeno zprávou 200 - OK, která se předá postupně na počáteční bod. Obdržení této zprávy potvrdí počáteční bod požadavkem zprávou ACK, která je předána cílovému uzlu. V tuto chvíli mezi sebou oba body navážou RTP relaci, která není závislá na proxy serveru. Toto může být doplněno o RTCP spojení sloužící k monitorování primární RTP relace.

Když se některý z uživatelů rozhodne hovor ukončit, odešle jeho klient požadavek zprávou BYE. Zpráva je předána opět prostřednictvím proxy serveru druhému bodu, který ho zase potvrdí zprávou ACK. Ve chvíli, kdy bod, který vyvolal ukončení, obdrží potvrzení, dojde k rozpojení RTP relace a hovor je ukončen.

6 Voicemail

Jde o službu telefonních ústředen, kdy v případě naplnění určitých podmínek se hovor přepojí do voicemailu (hlasového záznamníku pro dané číslo). Podmínek může být víc za sebou, ale vždy půjde o sérii nezvednutí hovoru po určitém časovém intervalu. Tato zpráva je potom uložena na serveru a připravena pro vyzvednutí od oprávněného uživatele.

Voicemail byl vytvořen Televoice International (pozdější Voicemail International, nebo VMI) v jejich prvním *US-wide Voicemail service* v roce 1980. Termín se nakonec začal používat obecně na prakticky všechny automatické hlasové služby využívající telefon. Oblíbenost hlasové pošty pokračuje dnes s internetovými telefonními službami, jako jsou Skype, Google Voice a ATT, které integrují hlasové, hlasové a textové služby pro tablety a smartphony.

Systémy hlasové pošty byly vyvinuty v pozdních sedmdesátých letech pro výměnu hlasových zpráv (VMX). Hlasová pošta se stala populární na začátku osmdesátých let, když byla zpřístupněna pro užití v PC. V září 2012 časopisy USA Today a Vonage prohlásily, že hlasová pošta je na ústupu. Zpráva uvádí, že počet hlasových zpráv klesl oproti roku 2011 o 8 procent.

6.1 Řešení společnosti Cisco

Pro fungování voicemailu je potřeba integrovat do Cisco Unified Communications Managera integrovat část která se jmenuje Cisco Unity Connection. Toto řešení pak řeší celou i celou oblast messagingu. Cisco má vlastní Cisco Jabber. Toto řešení lze snadno integrovat do celé sítě i do Cisco Unified Communications Managera. Nevýhodou je potřeba dalšího virtuálního stroje s dostatečnou velikostí pevného disku a paměti RAM.

6.2 Řešení Asterisk

Asterisk Voicemail službu nativně podporuje a má pro ni vlastní aplikační modul s názvem `app_voicemail.so`. Tento modul se konfiguruje pomocí konfiguračního souboru `voicemail.conf`. Tento soubor může obsahovat tři části. První je *general* v ní je základní nastavení pro celý voicemail.

Může obsahovat následující volby:

Format = gsm | wav – v jakém formátu se voicemail uloží

Maxsilence = - říká, po kolika sekundách ticha je zpráva považována za ukončenou

Druhá sekce *zonemessages* nemusí být uvedena.

Následuje definice kontextu v hranatých závorkách a následně specifikace samotných hlasových schránek. Definice vypadá takto:

Mailbox = heslo, jméno [,email [,pager_email [,nastaveni]]]

Uvádím příklad souboru voicemail.conf

```
[general]
format=wav
maxsilence=5
```

```
[internal]
7000 => 1,Pokus
```

7 Callcentrum

Callcentrum je služba pro využití telefonní ústředny pro potřeby podniku komunikovat se svými zákazníky či zaměstnanci. Třeba formou zákaznické linky či technické podpory (helpdesku).

7.1 Řešení společnosti Cisco

V případě řešení společnosti Cisco jde o řešení pro firemní zákazníky jmenující se Cisco Unified Contact Center Enterprise. Jde o komplexní řešení všech možných komunikačních kanálů ke komunikaci se zákazníky. Obsahuje kromě telefonních služeb, video hovoru, instant messagingu či e-mailové komunikace i nástroje pro vyhodnocování řešených dotazů i sledování jednotlivé komunikace a na ni navázaného reportingu. Bohužel se jedná o zvláštní produkt, který je firmou Cisco patřičně zpoplatněn.

7.2 Řešení Asterisku

Z pohledu Asterisku jde pouze o správu hlasových funkcí.

V rámci Asterisku je tato služba rozdělena na tři funkce:

1. *Interaktivní hlasovou odezvu* IVR (Interactive Voice Response)
2. *Fronta hovorů*
3. *Agent*

7.2.1 IVR

IVR je funkce, která zjistí požadavky volajícího a poté ho zařadí do požadované fronty hovorů. K zjištění do které fronty se má daný volající zařadit se typicky užívají menu nebo hlasové automaty.

7.2.2 Fronta hovorů

Systém front je základní funkce pro řešení callcentra. Umožňuje volajícímu čekat dokud se pro něj neuvolní odpovídající agent. Front samozřejmě může být více, třeba podle počtu prodáváných produktů.

Volající je po projití IVR zařazen do fronty, kde vyčkává na dobu než na něj přijde řada. Tedy uvolní se odpovídající agent, který jeho požadavek následně vyřídí.

Fronty se konfigurují v souboru *queues.conf* která vypadá následovně:

Musiconhold => default – nastaví se která hudba se bude čekajícímu přehrávat. Lze nastavit v souboru *musiconhold.conf*

Strategy = ringall | leastrecent | fewestcalls | rrmemory – uvádí jakým způsobem se budou hovory přidělovat jednotlivým agentům.

- *Ringall* – zavolá se na všechny dostupné agenty
- *Leastrecent* – zavolá se agentovy s nejdelším dobou od posledního uskutečněného hovoru
- *Fewestcalls* – zavolá se agentovy s nejmenším počtem uskutečněných hovorů
- *Rrmemory* – budou se cyklicky postupně obvolávat agenti podle seznamu agentů

Joinempty = strict - nelze uložit do fronty volajícího pokud není aktivní aspoň jeden agent

Leavewhenempty = strict – pokud se z fronty odhlásí poslední agent, jen volající z fronty odstraněn a předán zpět aplikaci *queue*, která ho dle priority předá do nové fronty

Member =>typ / jméno – určí komu bude hovor z fronty směřován. Jsou možné 3 varianty

- *Agent / číslo agenta* – členem se stane daný agent
- *Agent / @1* – členem se stane skupina 1
- *SIP / uživatel* – členem se stane daný uživatel

Příklad takového nastavení fronty

```
[fronta1]
musicclass=default
strategy=ringall
context=C_Prodej
joinempty=strict
leavewhenempty=strict
member => Agent/@1
```

7.2.3 Agent

Agent je operátor, který, pokud je přihlášen, přijme daný telefonní hovor a zpracuje ho. Agenti se mohou hlásit z jakéhokoli účtu: local nebo třeba SIP. Agenti se hlásí pomocí funkce *AgentLogin()* (slouží pro běžné přihlášení agenta) případně *AgentCallbackLogin()* (slouží pro přihlášení agenta s klapkou). Poté na ně mohou být přesměrovány hovory z front, kde jsou registrováni. Každý agent může být registrován ve více frontách.

Agenti se spravují a zakládají pomocí *souboru agent.conf*.

V sekci *general* je důležitá volba **persistentmembers = yes / no** – pro opětovné připojení po startu či restartu ústředny.

Sekce agents:

maxlogintries = * – počet pokusů pro přihlášení klienta. Standardně se nastavuje na 3.

autologoff = * - doba v sekundách po kterou bude agentovi zvonit telefon než bude považován za nedostupného a automaticky odhlášen.

ackcall = yes / no – zda bude po přihlášení pomocí funkce *AgentCallbackLogin()*, agent potvrzovat svoji volbu pomocí "#".

wrapuptime = * - určuje prodlevy mezi jednotlivými hovory agenta v milisekundách.

musiconhold => agent – určuje hudbu na pozadí.

groupe = * - určuje do jaké skupiny agent patří. Může patřit do více skupin, pak je potřeb jednotlivá čísla oddělit čárkou.

recordagentcall = yes / no – zda má být agent nahrán.

recordformat = wav / gsm / wav49 – formát v kterém má být hovor nahrán.

Poté následuje definice jednotlivých agentů:

Agent => AgentId, heslo, jméno

Příklad definice agentů:

```
[agents]
musiconhold = agent
autologoff = 20
wrapuptime = 10000
ackcall = no
```

```
;agenti pro frontu 1
group = 1
agent => 501,1,Agent1_fronta1
agent => 502,1, Agent2_fronta1
; agenti pro frontu 2
group=2
agent => 601,1, Agent1_fronta2
agent => 602,1, Agent2_fronta2
```

8 Instalace

Pro praktickou část diplomové práce jsem získal testovací verzi Cisco Unified Communications Manageru verze 11.0 (CUCM v11.0) s omezenou dobou užití na 60 dnů. Pro Asterisk jsem užil aktuální verzi staženou ze stránek www.asterisk.org (AsteriskNOW 10.13).

První problémy nastali s instalací Cisco Unified Communications Manageru. Dlouho jsem se snažil danou verzi nainstalovat přímo na nějaký server. Zde bych ještě jednou chtěl poděkovat mému vedoucímu diplomové práce Ing. Pavlu Trollerovi, CSc., Ing. Jánmu Kučerákovi a kolegovi Ing. Petru Šimberu za možnosti zkusit instalaci na větší počet různých počítačů a serverů. Bohužel tato snaha vždy skončila na chybové hlášení oznamující, že daný stroj nesplňuje hardwarové požadavky.

8.1 Seznam otestovaných nefunkčních konfigurací serverů

Server1:

- Processor Intel Xeon 6100 Skylake
- Operační paměť 32 GB
- 2x 500GB 7200rpm pevné disky v Raid 0
- 1 Gbit Ethernet rozhraní

Server2:

- Processor Intel Xeon 4100 Skylake
- Operační paměť 12 GB
- 250GB 7200rpm pevný disk
- 1 Gbit Ethernet rozhraní

Server3:

- Processor Intel Xeon E3 - 1231 v3 Haswell
- Operační paměť 32 GB
- 2x 500GB 7200rpm pevné disky v Raid 0
- 1 Gbit Ethernet rozhraní

Server4:

- Processor Intel Xeon E3 - 1275 v5 Skylake
- Operační paměť 16 GB
- 2x 250GB 5400rpm pevné disky v Raid 0
- 1 Gbit Ethernet rozhraní

Server5:

- Processor Intel Xeon E5 - 2630 v4 Haswell
- Operační paměť 16 GB
- 2x 100GB 5400rpm pevné disky bez Raidu
- 1 Gbit Ethernet rozhraní

Server6:

- Processor Intel Xeon E5 - 2600 v3 Haswell
- Operační paměť 12 GB
- 500GB 7200rpm pevný disk
- 1 Gbit Ethernet rozhraní

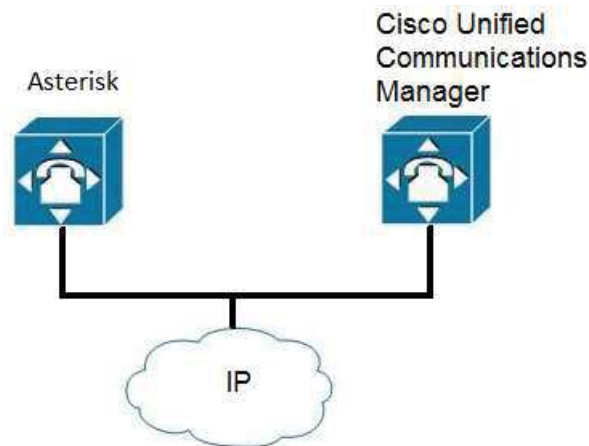
Server7:

- Processor Intel Xeon E3 - 1225 v5 Skylake
- Operační paměť 64 GB
- 2x 1TB 7200rpm pevné disky v Raid 0
- 1 Gbit Ethernet rozhraní

8.2 Fungující řešení

Po delším pátrání jsem na stránkách společnosti Cisco našel parametry potřebné pro spuštění Cisco Unified Communications Manageru ve virtuálním prostředí VMware. Proto jsem změnil přístup a vytvořil server s prostředím VMware a na něm jsem vytvořil dva virtuální servery. Pro každou instalaci jeden. Firma Cisco má dokonce připravené konfigurační soubory .ova pro příslušné verze Cisco Unified Communications Manageru ke stažení a použití. Právě tento konfigurační soubor mi významně pomohl. Po jeho užití se Cisco Unified Communications Manager v pořádku rozběhl na serveru, pro který v předchozích pokusech o instalaci hlásil problém s nedostačujícím hardwarem.

Stroj s Asteriskem plnil i funkci NTP a DHCP serveru pro Cisco Unified Communications Manager. Schéma propojení obou zařízení je na vyobrazeno na obrázku 5.



Obrázek 5: Propojení Cisco Unified Communications Manageru a Asterisku

Server pro VMware:

- Processor Intel Xeon E5 - 2670 v3 Ivy Bridge
- Operační paměť 128 GB
- 2x 4TB 7200rpm pevné disky v Raid 0
- 1 Gbit Ethernet rozhraní

8.3 Virtuální počítač pro CUCM

Minimální požadavky deklarované firmou Cisco je:

- Dva procesory Intel Xeon Haswell taktu 2GHz
- Operační paměť 4GB
- Pevný disk velikosti 80GB
- 1 Gbit Ethernet rozhraní

Použité nastavení:

- Procesor – pevně čtyři lokální jádra
- Operační paměť – dynamická alokace paměti až do 32 GB
- Pevný disk – dynamicky se zvětšující až do 250 GB

8.4 Virtuální počítač pro Asterisk

Minimální požadavky operačního systému Linux Centos 7

- Procesor – jedno jádra
- Operační paměť – 2 GB
- Pevný disk – 6 GB

Použité nastavení:

- Procesor – pevně čtyři lokální jádra
- Operační paměť – dynamická alokace paměti až do 64 GB
- Pevný disk – dynamicky se zvětšující až do 500 GB

8.5 Ostatní hardware

Dále jsem použil dva notebook s operačním systémem Windows 7 s Cisco IP SoftPhone pro simulaci telefonního provozu.

8.6 Navrhnuté řešení

Po pokusech s nastavením jednotlivých funkcí, které nabízí Cisco Unified Communications Manageru ve zkušební verzi, jsem se rozhodl připojit k němu Asterisku pomocí protokolu SIP. Pro možnost komunikace mezi oběma zařízeními jsem se rozhodoval mezi třemi možnostmi. Buď použít protokol H.323, který využívá binární formát, protokol SCCP (Skinny Client Control Protocol) nebo použít protokol SIP, který využívá textový formát.

Po pokusech se všemi protokoly jsem se rozhodl pro využití protokolu SIP. Jeho nastavení mi přišlo nejjednodušší a funkčnost byla bezproblémová.

9 Nastavení CUCM

Nebudu se zde rozepisovat o základním nastavení Cisco Unified Communications Manageru. Toto nastavení je podrobně popsáno v příručkách, které jsou bezplatně k dispozici na stránkách společnosti Cisco [1].

Nastavení Cisco Unified Communications Manageru může být pomocí příkazové řádky nebo pomocí webového rozhraní (GUI).

9.1.1 Nastavení SIP Trunku pomocí GUI

1. Přihlásíme se do Cisco Unified Communication Manager.

2. Vytvoření SIP Trunk Security Profile

V záložce *System > Security* zvolíme *SIP Trunk Security Profile*

Klikneme na *Add New*

Povolíme šifrování **TLS**:

- V *Device Security Mode* vybereme záložku *Encrypted*
- V poli *Incoming Transport Type and Outgoing Transport Type* vybereme typ **TLS**
- V položce *X.509 Subject Name field* vyplníme název certifikátu. V našem případě **X.509**
- Do pole *Incoming Port* zadáme číslo portu. Výchozí hodnota pro TLS je **5061**

Nastavení uložíme kliknutím na tlačítko *Save*

3. Nastavíme typ zařízení na druhé straně

V záložce *Device > Device Settings* zvolíme *Common Device Configuration*

Klikneme na *Add New*

Vyplníme jméno zařízení v položce *name*

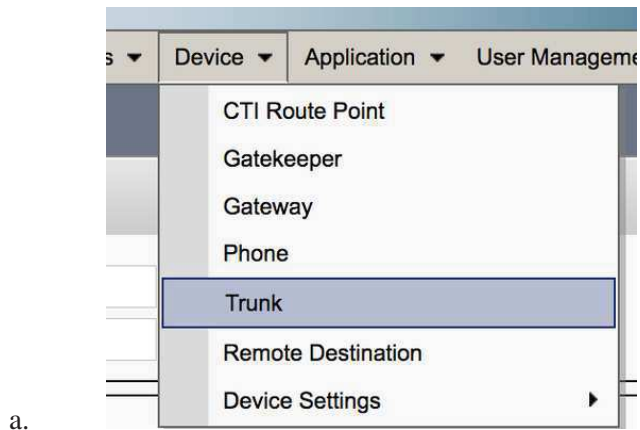
V položce *IP Addressing Mode* necháme výchozí hodnotu **IPv4 and IPv6 (Default)**

A v položce *IP Addressing Mode for Signaling* vybereme hodnotu **IPv4**

Nastavení uložíme kliknutím na tlačítko *Save*

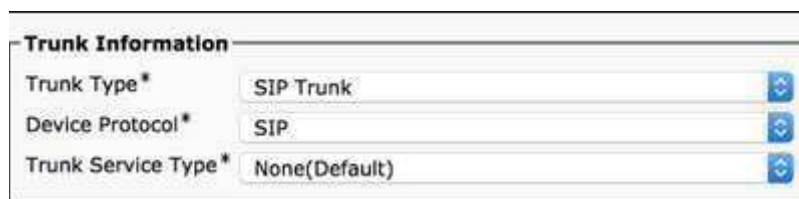
4. Nastavíme samotný SIP Trunk

V záložce *Device -> Trunk* klikneme na *Add a New Trunk*



Obrázek 6: Postup nastavení CUCM I

Nastavíme: *Trunk Type* - **SIP Trunk**
Device Protocol - **SIP Trunk**
Trunk Service Type - **None (Default)**
 a kliknout na *Next*



Obrázek 7: Postup nastavení CUCM II

Na tomto listě nastavíme:

Device Name na **Trunk-Asterisk**

Description – **Trunk na Asterisk**

Device Pool – Vybrat odpovídající Device pool, v našem případě **Tel**

SIP Information – IP adresa Asterisku **10.10.1.2**

Port - **5061**

Vybrat *SIP Trunk Security Profile* – Nastavíme náš profil z bodu 2. **SIP Asterisk**

A zaškrtneme **SRTP Allowed**

Postupně kliknout na *Save, Apply a Reset*

9.1.2 Nastavení Voicemailu

Nastavíme pravidla pro voicemail. Nejrychleji se vytvoří přes příkazovou řádku.

Pokud voláš telefonní číslo 1000 a nikdo to 30s nevezme, přesměruj hovor na telefonní číslo 7000 (voicemail).

```
CUCM(config)#ephone-dn 1 ;number 1000  
CUCM(config-ephone-dn)#call-forward noan 7001 timeout 30
```

Pro usnadnění práce s voicemailem je vhodné nastavit i tlačítko rychlé volby na telefonní číslo pro výběr voicemailu. To se provede v záložce *Device > Device Settings > Phone Button Template* kde pro dané tlačítko vyplníme požadované telefonní číslo.

10 Nastavení Asterisku

Ani v této části se nebudu rozepisovat o základním nastavení Asterisku. Návod jak Asterisk nastavit je podrobně popsán například ve wikipédii na stránkách Asterisku [2].

Konfigurace Asterisku probíhá pomocí příkazové řádky nebo přímým upravováním modulů v textovém editoru nebo přes webové rozhraní.

10.1 Nastavení SIP

Pro správné nastavení SIP trunku je potřeba upravit modul *sip.conf* následujícím způsobem:

```
[general]
tlsenable=yes
tlscertfile=/etc/asterisk/asterisk.pem
```

```
[cucm]
Type = friend
Host = 10.10.1.1
Port = 5061
context=internal
secret=test
dtmfmode=rfc2833
disallow=all
allow=ulaw
Encryption = yes
```

10.2 Nastavení voicemailu

Nastavit voicemail na telefonní číslo 7000 a 7001. To se provede v *extensions.conf*:

```
[users]
exten => 7000,1,Answer(500)
exten => 7000,n,Playback(hello-world)
exten => 7000,n,Hangup()

exten => 7001,1,Dial(SIP/demo,20)
exten => 7001,n,VoiceMail(7001@vm-demo,u)
```

10.3 Nastavení callcentra

Nastavení v souboru *sip.conf*

```
[101]
Username = Volba_A
Secret = pass
Host = dynamic
Type = friend
Disallow = all
Allow = ulaw
Allow = h263
Allow = gsm
Allow = alaw
Context = moi-ivr
Callerid = Volba_A
Videosupport = yes
```

```
[102]
Username = Volba_B
Secret = pass
Host = dynamic
Type = friend
Disallow = all
Allow = ulaw
Allow = g722
allow = gsm
allow = alaw
context = moi-ivr
callerid = Volba_B
```

Dále přidáme kontext

```
[moi-incoming] ; Začátek
exten =>s,1,Verbose(1, Caller ${CALLERID(all)} has entered IVR menu)
same => n,Answer()
same => n,Set(TIMEOUT(digit)=2)
same => n,Wait(2) ; počká dvě sekundy
```

; Nastavení pracovní doby. Pokud volá v pracovní dobu pak přepojí do kontextu moi-ivr.
Pokud volá mimo pracovní dobu, tak přepojí do kontextu offices-closed.
same => n,GotIrfTime(9:00-16:00,mon-fri,*,*?moi-ivr,s,1:offices-closed)
same => n,GotIrfTime(9:00-12:00,sat,*,*?moi-ivr,s,1:offices-closed)

```
[offices-closed]
exten =>s,1,Answer()
same => n,wait(2)
same => Background(basic-pbx-ivr-main)
same => n,Hangup()
```

```

[moi-ivr] ; samotné menu
exten =>s,1,Answer()
same => n,wait(2)
same => n,Set(TIMEOUT(digit)=10) ; ; this sets the inter-digit timer
same => n,Set(TIMEOUT(response)=10)
same => n,Background(basic-pbx-ivr-main)
same => n,WaitExten()
same => n,Background(silence/5)

; stiskem 1 vytočíme 101 – přejdeme od ivr-Volba_A
; stiskem 2 vytočíme 102 – přejdeme od ivr-Volba_B
exten => 1,1,Goto(ivr-Volba_A,s,1)
exten => 2,1,Goto(ivr-Volba_B,s,1)

; pokus špatně zadá přehraje se chybová zpráva
exten => i,1,Playback(option-is-invalid)
same => n,Goto(moi-ivr)

; prodloužení timeoutu
exten => t,1,Playback(still-there)
exten => t,Goto(moi-ivr,s,5)

; definice kontextů jednotlivých položek menu
[ivr-Volba_A]
exten =>s,1,Verbose(1, Caller ${CALLERID(all)} has entered Customer care support center)
same => n,Answer()
same => n,Dial(SIP/101,10,r)
same => n,Hangup()
same => n,Playback(unavail)
same => n,Hangup()

[ivr-Volba_B]
exten =>s,1,Verbose(1, Caller ${CALLERID(all)} has entered Customercare support center)
same => n,Answer()
same => n,Dial(SIP/102,10,r)
same => n,Hangup()
same => n,Playback(unavail)
same => n,Hangup()

```

Do souboru *extensions.conf* přidáme řádek

```
include => moi-incoming
```


11 Závěr

Po zprovoznění a otestování navržené varianty spojení Cisco Unified Communications Manageru a Asterisku pomocí protokolů SIP. Můžu prohlásit, že spojení funguje, je jednoduché a dostatečně funkční pro provoz voicemailu a callcentra.

Výhodou Asterisku je jeho GPL forma distribuce a jeho nezávislost na hardwaru na kterém běží. Další velkou výhodou je možnost sám si přidat nebo upravit potřebné aplikace. Mezi další výhody patří jeho nezávislost na hardwaru.

Výhodou Cisco Unified Communications Manageru je jeho plná integrace do řešení Cisco Unified Communications. Všechna hardware je na pro jeho potřeby certifikována. To ovšem přináší i nevýhody. Hůře se instaluje do sítí s prvky od jiných výrobců. A samozřejmě i jeho velká pořizovací cena a cena jednotlivých funkčních rozšíření. Další slabinou může být orientace na IP síť a jeho horší (těžší a dražší) připojení pro analogová koncová zařízení.

Z popisu vyplývá, že oba systémy umí splnit službu telefonní ústředny samy o sobě. Proč je tedy spojovat? Jde o možnost rozšířit komerční (nákladné) řešení firmy Cisco o bezplatné aplikace produktu Asterisk. Hlavním důvodem pro tuto volbu bude asi snaha ušetřit prostředky a přesto nepřijít o výhody užití technologie firmy Cisco v sítích vystavěných nad technologií této firmy.

Řešení voicemailu i callcentra jen pomocí produktů od společností Cisco je možné. Toto řešení bude jednoduché na integraci do sítě v níž Cisco Unified Communications Manager již funguje. V případě callcentra to bude trochu těžší, protože se jedná o komplexní řešení komunikace se zákazníkem. Je také finančně dosti náročné. Nejde jen o nákup softwaru, ale i potřebného serveru a toto řešení se vyplatí jen pro opravdu velké firmy, které mají denně stovky hovorů. Cenu tohoto řešení odhaduji v řádech milionů korun. Proto se vyplatí menším či středním firmám najít jiné řešení. V případě voicemailu nebude ona úspora tak výrazná, protože cena tohoto řešení se pohybuje v desítkách tisíc korun.

Jistě se najde spousta dalších možností jak využít spojení Cisco Unified Communications Manageru a Asterisku. A vzhledem k cenové politice firmy Cisco bude toto řešení výrazně levnější.

Literatura

- [1] Webové stránky společnosti Cisco, <http://www.cisco.com>
- [2] Webové stránky společnosti Asterisk, <http://www.asterisk.org>
- [3] Schulzrinne, H., Rosenberg, J. et al.: *SIP: Session Initiation Protocol*. Request for Comments 3261. Internet Engineering Task Force, červen 2002. Proposed Standard

Seznam použitých zkratek

API	Application Programming Interface
CAR	Committed Access Rate
CDR	Call Details Records
CMR	Call Records Management
CUCM	Cisco Unified Call Manager či Cisco Unified Communications Manager
DHCP	Dynamic Host Configuration Protocol
DRS	Disaster Recovery System
GSM	Groupe Spécial Mobile
GPL	General Public Licence
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JTAPI	Java Telephony API
IAX	Inter-Asterisk eXchange
IETF	Internet Engineering Task Force
IM	Instant Messaging
IP	Internet Protocol
IP IVR	IP Interactive Voice Response
ISDN	Integrated Services Digital Network
LDAP	Lightweight Directory Access Protocol
MGCP	Media Gateway Control Protocol
MMUSIC	Multiparty Multimedia Session Control
NTP	Network Time Protocol
PBX	Private Branch eXchange
PCM	Pulse-Code modulation
QoS	Quality Of Service
RSVP	Resource Reservation Protocol
RTP	Real-Time Protocol
SCCP	Skinny Client Control Protocol
SIP	Session Initialization Protocol
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SDP	Session Description Protocol
SS7	Signaling System Number 7
TAPI	Telephony Application Programming Interface

TCP	Transmission Control Protocol
TDM	Time-Division Multiplex
UC	Unified Communications
UCCM	Unified Cisco Call Manager
UDP	User Datagram Protocol
VoIP	Voice Over Internet Protocol