

## Съдържание

<u>Съдържание.....</u>	<u>2</u>
<u>Увод.....</u>	<u>4</u>
<u>Глава Първа.....</u>	<u>6</u>
<u>Архитектура и основни функции на VoIP сървъра. Стратегия за реализиране на VoIP свързаност. Различни видове мрежови телефонни топологии.....</u>	<u>6</u>
<u>    Същност на IP телефонията.....</u>	<u>6</u>
<u>        Превю на една VoIP връзка.....</u>	<u>8</u>
<u>        1.2 Преобразуване от аналогов в цифров сигнал.....</u>	<u>9</u>
<u>        1.3 Алгоритъм за компресиране.....</u>	<u>9</u>
<u>        1.4 RTP - Real Time Transport Protocol.....</u>	<u>9</u>
<u>        1.5 RSVP.....</u>	<u>10</u>
<u>        1.6 Качество на Услугата (QoS).....</u>	<u>10</u>
<u>        1.7 H.323 Сигнален протокол.....</u>	<u>11</u>
<u>2. Приспособяване.....</u>	<u>16</u>
<u>    2.1 Потребителския пазар.....</u>	<u>16</u>
<u>    2.2 PSTN и мобилни мрежови доставчици.....</u>	<u>17</u>
<u>    2.3 Корпоративна употреба.....</u>	<u>18</u>
<u>3. Предимства и недостатъци.....</u>	<u>19</u>
<u>    3.1 Оперативни разходи.....</u>	<u>19</u>
<u>    3.2 Качество на обслужването.....</u>	<u>19</u>
<u>    3.3 Податливост от спиране на тока.....</u>	<u>20</u>
<u>    3.4 Защита.....</u>	<u>21</u>
<u>4. Необходим хардуер за изграждането на VoIP мрежа.....</u>	<u>21</u>
<u>    4.1 Медиа шлюзове.....</u>	<u>21</u>
<u>    4.2 Медиа шлюз контролери.....</u>	<u>23</u>
<u>    4.3 IP Мрежа.....</u>	<u>24</u>
<u>    4.4 Телефонни апарати.....</u>	<u>24</u>
<u>    Видове телефонни топологии.....</u>	<u>26</u>
<u>        6.1 Топология използваща само VoIP шлюз.....</u>	<u>26</u>
<u>        6.2 Топология използваща само и единствено IP телефонни апарати.....</u>	<u>27</u>
<u>        6.3 Топология от смесен тип.....</u>	<u>27</u>
<u>Глава Втора.....</u>	<u>28</u>
<u>Изграждане на проект използващ Интернет услугата VoIP за преминаване на Икономически Университет Варна към Интернет Телефония.....</u>	<u>28</u>
<u>    2.1 Подход и план на разработване на проекта.....</u>	<u>28</u>
<u>    2.2 Необходима техническа част за реализирането на проекта.....</u>	<u>29</u>
<u>    2.3 Необходима софтуерна част за реализирането на проекта.....</u>	<u>32</u>

<u>2.4 Инсталиране и конфигуриране.....</u>	<u>33</u>
<u>2.4.1 Инсталиране на операционната система и добавяне на хранилища.....</u>	<u>33</u>
<u>2.4.2 Инсталиране и конфигуриране на OpenVPN.....</u>	<u>33</u>
<u>2.4.3 Инсталиране на Asterisk.....</u>	<u>42</u>
<u>2.4.4 Конфигуриране на Asterisk.....</u>	<u>49</u>
<u>2.5 Администриране на Asterisk.....</u>	<u>60</u>
<u>Заклучение.....</u>	<u>61</u>
<u>Използвана литература.....</u>	<u>63</u>
<u>Интернет адреси:.....</u>	<u>63</u>
<u>Приложение 1.....</u>	<u>65</u>

## Увод

Voice Over Internet Protocol (VoIP) представлява актуална и модерна технология, позволяваща пренос на глас посредством използване на широколентова интернет връзка. Благодарение на тази услуга фирмите, които я поддържат, дават възможност на крайните си потребители да осъществяват както градски и междуградски, така и евтини международни разговори с близки и роднини в чужбина.

Според актуалните пазарни проучвания на TeleGeography, през последните години международният VoIP трафик нараства значително. За 2011 г. увеличението е с 35 %, като за 2012 г. прогнозите са за нарастване с 38 % (за Източна Европа – с 23 %). Ръстът се дължи основно на увеличеното използване на VoIP телефония в развиващите се страни и донякъде на стабилизирането на цените на услугата.

Съществуват два основни начина, по които може да осъществите VoIP обаждане – чрез стационарен телефон (набирайки специален код на вашия оператор или при наличието на специален адаптер) и чрез домашен компютър (разбира се, с интернет връзка и необходимото софтуерно и хардуерно обезпечение).

В момента Икономически Университет Варна използва Телефонна централа Alcatel. А във Втори корпус липсва каквато и да било телефонна централа, затова там ще трябва да започнем от нулата и да изградим телефонна мрежа. Но тъй като във Втори корпус има изградена интернет мрежа, може спокойно да използваме VoIP телефонията през нея, понеже едно от предимствата на VoIP е точно това, че гласа преминава през вътрешната интернет мрежа докато стигне до централата, където може да излезе извън пределите на корпуса. Останалите предимства ще разгледаме във втората глава на дипломната работа.

Целите, които ще трябва да изпълни дипломната работа са:

- Преминане от старата телефонна централа, обслужваща в момента Икономически Университет Варна в нова, интернет базирана телефонна централа.
- Втората цел, която дипломната работа трябва да разгледа е изграждане на телефонна система във Втори корпус на Икономически Университет Варна.
- Трета цел е телефонна комуникация между Първи и Втори корпус на Университета.

Задачите който трябва да бъдат изпълнени са следните:

- ✓ Свързване на основната сграда на Икономически университет – Варна и Втори корпус при същия университет, чрез изграждане на Виртуална Частна Мрежа<sup>1</sup> (VPN);
- ✓ Инсталиране и конфигуриране на VoIP Server;
- ✓ Регистрация на клиентските номера в телефонната централа;

---

<sup>1</sup> Виртуална частна мрежа или VPN (от английски *Virtual Private Network*) е компютърна мрежа, логически изградена чрез криптиране, използваща физическата и програмна инфраструктура на по-голяма обществена мрежа, най-често Интернет.

## Глава Първа

### Архитектура и основни функции на VoIP сървъра. Стратегия за реализиране на VoIP свързаност. Различни видове мрежови телефонни топологии.

#### Същност на IP телефонията

Ай Пи (IP) Телефонията е високотехнологична комуникационна услуга от следващо поколение, предоставяща възможности за провеждане на качествени телефонни разговори на ниска цена, посредством Интернет протокол. Обединява предимствата на традиционната телефонна услуга и на Интернет технологията, за да даде нови и функционални възможности на конкурентна цена.

Тази услуга е същата като телефонната услуга, с тази разлика, че гласът се цифровизира и посредством Интернет се пренася до желаното направление като на мястото на пристигане отново се трансформира в глас. Ниските цени, които се предлагат се дължат на по-ниската себестойност за пренасяне на глас през Интернет.

Voice over IP, или т.нар. IP телефония, е съвременно средство за неограничени от времето и разстоянието телефонни разговори на много по-ниски цени от досегашните. Това става възможно посредством нова технология, която пренася Вашия глас по Интернет до която точка на света желаете.

Цените не се формират на базата на отдалечеността на търсеното направление, а според състоянието на Интернет мрежата там, а това е предпоставка за бъдещото им понижаване.

Днес VoIP е популярен термин на пазара. Всъщност съвременната VoIP телефония не би трябвало да се различава по нищо като услуга от класическата POTS<sup>2</sup> телефония. За разлика от POTS, VoIP притежава няколко много добри качества, които ще бъдат описани в дипломната работа. Именно поради тях все повече фирми предпочитат да използват VoIP системи за изграждането на техните телефонни централи.

Като основен факт, който може да се изтъкне, е че Интернет вече е навсякъде около нас, а VoIP се уповава именно на него. Няма фирма, която да не използва в някаква степен интернет. Точно заради това VoIP става много удобен за използване и същевременно е невидим.

---

<sup>2</sup> POTS – Plain old telephone service, е аналогова телефонна услуга служеща за свързване на телефонната мрежа на малък или среден бизнес.

Един от положителните ефекти при използването на VoIP, е че ако преминаваме от стара аналогова телефонна централа към VoIP, разходите няма да бъдат големи, да не кажа, че ще бъдат почти никакви, в случай че имаме вече изградена интернет мрежа във фирмата, в която ще осъществяваме това преминаване.

Някой от основните предимства и плюсове, които притежава VoIP са:

- VoIP е модерен;
- След '90-те години бума на Интернет поражда търсене на начин за представяне на нови услуги, поради този факт се ражда и VoIP. Предимството тука е, че VoIP използва Интернет;
- В сравнение с останалите пакетни технологии VoIP е най-евтин, поради сложни пазарни причини. При разговори между две страни, които се намират на голямо разстояние, цената се свежда до цената, която се плаща на местния интернет доставчик, като по този начин се елиминират месечните такси и плащания за минутите разговор към PSTN<sup>3</sup>.
- Базовите познания за тази технология са същите като за Интернетта, ето защо на пазара може да се намери лесно група от хора, които да разбират от VoIP системи, отколкото от POTS.

Бъдещето на телефонията през интернет все още е неясно. Напредъка на технологията Voice Gateway в посока на подобряване на качеството на звука и съвместимостта между продуктите продължава. Цените падат постоянно. В зависимост от услугите на местния Интернет доставчик (ISP<sup>4</sup>), потребителите могат да гледат на VoIP като на алтернатива за осъществяване на повиквания на голямо разстояние. За много хора се оказва по-изгодно да плащат месечен абонамент при доставка на Интернет, а това води до нарастване на ползване на VoIP в средния спектър на потребление. Телефонията през Интернет е все още иновационна технология, но тя продължава да съсредоточава все по-голям интерес. Скоростите за връзка към Интернет нарастват, което води до нарастване и на качество и падане на цената на VoIP телефонията.

Интернет телефонията или съкратено VoIP (глас през интернет протокол) е високотехнологична услуга и представлява пренос на глас чрез използване на широколентова интернет връзка. Терминът VoIP може да се отнася до връзка между два компютъра, два телефонни апарата, или компютър и телефонен апарат, стига сигналът да се пренася в част от пътя си чрез IP пакети. Чрез нея може да провеждате както градски,

---

<sup>3</sup> PSTN (Public Switched Telephone Network) е телефонна мрежа от канално-комутируемите публични мрежи по света.

<sup>4</sup> ISP – Internet service provider – местен доставчик на интернет.

междуградски, така и международни разговори на много ниски цени. Гласовата услуга по интернет спестява на частния потребител и особено на фирмите огромни средства.

Преди много години хората открили, че могат да изпратят аналогов сигнал до отдалечено място като същия този сигнал могат да го дигитализират. Но преди да бъде изпратен, сигнала трябва да премине през ADC (analog to digital converter – преобразувател на аналогов в цифров сигнал), да се пренесе сигнала и накрая да се декодира от DAC (digital to analog converter – преобразувател на цифров в аналогов сигнал).

VoIP работи точно по този начин. Дигитализира гласа в пакети от данни, изпраща ги през мрежата и когато пристигнат при отсрещната страна преобразува същите пакети обратно в глас. Като цяло цифровия формат данни може да бъде по-лесно контролиран; може да бъде компресиран, рутиран, преобразуван в нов по-добър формат и т.н. Също така цифровия сигнал е по-толерантен към „шум“ отколкото аналоговия.

### Превю на една VoIP връзка

За да се осъществи една VoIP връзка са ни необходими:

1. Преобразувател на аналогов сигнал в цифрови битове (ADC<sup>5</sup>);
2. След това битовете трябва да се компресират в добър формат за транспортиране: съществуват много протоколи, които правят това;
3. След това трябва да вмъкнем нашите „гласови пакети“ в пакетите от данни, който се използват от времевите протоколи (по принцип RTP<sup>6</sup> използващ UDP<sup>7</sup> и IP<sup>8</sup>);
4. Имаме нужда от „сигнален протокол“ за да се обадим на потребителя отсреща: ITU-T<sup>9</sup> H.323 извършва това;
5. При получателя ние трябва да деасемблираме пакетите от данни, да изведем информацията от тях, да я преобразуваме в аналогов гласов сигнал и да я изпратим към звуковата карта или към слушалката на телефона;
6. Всичко това трябва да се случи веднага или в сегашно време защото не може да чакаме прекалено дълго.

---

<sup>5</sup> ADC – Analog to Digital Converter – преобразувател от аналогов в цифров сигнал.

<sup>6</sup> RTP – Real Time Transport Protocol – Протокол за пренос на данни в реално време.

<sup>7</sup> UDP – User Datagram Protocol – Минимален Транспортен Протокол.

<sup>8</sup> IP – Internet Protocol – Интернет Протокол.

<sup>9</sup> ITU-T – International Telecommunication Union - Telecommunications –  
Международна Телекомуникационна организация – подсектор Телекомуникации.

## 1.2 Преобразуване от аналогов в цифров сигнал

Това се прави от хардуера и по-точно от хардуерни карти наречени ADC. В днешно време всяка звукова карта има възможност да преобразува с 16 бита 22 050 Hz и като резултат да получи 176.4 kBytes/s за стерео поток. На VoIP обаче не му е необходим чак такава скорост за пренос на гласови пакети.

## 1.3 Алгоритъм за компресиране

След като вече имаме цифрова информация може да я преобразуваме в стандартен формат за по-добре пренасяне по мрежата. За тази цел се използва PCM<sup>10</sup> ITU-T G.711.

- Гласовата пропускателна способност е 4 kHz така, че пропускателната способност за взимаме на проби трябва да бъде 8 kHz (по Найквист<sup>11</sup>);
- Представяме всяка проба с 8 бита (имащи 256 възможни стойности);
- Пропускателната способност на типичната цифрова телефонна линия е 8000 Hz \* 8 bit = 64 kbit/s.

## 1.4 RTP - Real Time Transport Protocol

Вече имаме сурова информация и искаме да я инкапсулираме в TCP/IP стек. Ще следваме следната структура: VoIP пакети -> RTP -> UDP -> IP -> I, II слой

VoIP пакетите се съдържат в RTP (Протокол за пренос на данни в сегашно време) пакети, който от своя страна се намират в UDP-IP пакетите.

Първо, VoIP не използва TCP<sup>12</sup> защото е прекалено „тежък“ за приложения, които работят в „реално време“, така че вместо него VoIP използва UDP.

Второ, UDP не контролира подредбата под която пристигат пакетите при получателя или колко време им отнема за да пристигнат там. Тези двете са от голямо значение за качеството на гласа (това колко добре може да разберем какво казва другия човек, от другата страна на линията) и качеството на разговора (колко е лесно да се проведе разговор). RTP решава този проблем като позволява на получателя да сложи пакетите обратно в правилния ред и да не чака прекалено дълго за пакети, които или са се загубили по пътя или им отнема прекалено много време за да пристигнат (не се нуждаем от всеки един пакет по отделно, нуждаем се от продължителен поток от много пакети и да са подредени правилно). На Таблица 1 може да видим какво представлява RTP протокола:

<sup>10</sup> PCM – Pulse Code Modulation – метод използван за цифрово представяне на аналоговите сигнали.

<sup>11</sup> Хари Найквист – математик и физик, създател на теоремата Шум на Джонсън-Найквист.

<sup>12</sup> TCP – Transmission Control Protocol – мрежов протокол за управление на обема на информация.



1. TOS<sup>15</sup> поле в IP протокола за да се опише типа на услугата: високите стойности показват по-ниска спешност от пакета, докато по-ниските стойности се използват за пренос в реално време;
2. Методи за подреждане на пакетите в опашка:
  - a. FIFO<sup>16</sup> – най-простия метод позволяващ на пакетите да преминат в правилния ред;
  - b. WFQ<sup>17</sup> – състоящ се в справедливото преминаване на пакети (например FTP<sup>18</sup> не може да изконсумира всичката налична честотна лента) в зависимост от вида на потока от данни, обикновено един пакет за UDP и един за TCP преминават по справедлив начин;
  - c. CQ<sup>19</sup> – потребителите могат да преценят приоритета на преминаването на пакетите;
  - d. PQ<sup>20</sup> – има редица (обикновено от по 4) опашки, които притежават приоритетно ниво: първо, пакетите от първата опашка се изпращат, след това, (когато първата опашка е вече празна) започва изпращането от втората и т.н.;
  - e. CB-WFQ<sup>21</sup> - също като WFQ, но в допълнение има класова концепция (до 54) и лентовата честота се асоциира за все един пакет по отделно;
3. Пропускателна способност, където позволява да се постави лимит на използваната честотна лента в:
  - a. Сваляне;
  - b. Качване;
4. Избягване на „задръстванията“, чрез RED<sup>22</sup>;

### 1.7 H.323 Сигнален протокол

H.323 е препоръчан от ITU-T и описва протоколи, които доставят звуково-визуални комуникационни сесии във всяка мрежа, която използва пакети. Стандарта H.323 регистрира и контролира сигнала на обаждането, мултимедийния транспорт и контролира връзката при използването на честотната лента при схемата точка-до-точка или при конферентните обаждания и връзки. Той е широко имплементиран при използването на глас и оборудване за видео конферентни разговори. Също така се използва при много интернет приложения,

---

<sup>15</sup> TOS – Type of service – поле в IP протокола, описващо типа на услугата.

<sup>16</sup> FIFO – First in, First Out – Първи влязъл, първи излязъл.

<sup>17</sup> WFQ – Weighted Fair Queuing – Претеглено разпределяне на опашката.

<sup>18</sup> FTP – File Transfer Protocol – протокол за трансфер на файлове.

<sup>19</sup> CQ – Custom Queuing – потребителско подреждане на опашката.

<sup>20</sup> PQ – Priority Queuing – Приоритетно подреждане на опашката.

<sup>21</sup> CB-WFQ - Class Based Weighted Fair Queuing – Клас базирано WFQ.

<sup>22</sup> RED – Random Early Detection – Случайно ранно откриване;

които изискват пренос на данни в реално време и поради това се разработва от много доставчици на гласови и видео услуги в интернет. H.323 е част от семейството протоколи на ITU-T H.32x, които също регистрират мултимедийни връзка през ISDN<sup>23</sup>, PSTN или SS7<sup>24</sup>, и 3G мобилните мрежи.

H.323 се основава на препоръката на ITU-T, която е от протокол Q.931 и е подходящ за предаване на обаждания и глас през мрежи използващи смес от IP, PSTN, ISDN и QSIG през ISDN. Модела, който използва и описва H.323 е подобен на модела на ISDN, но в същото време е лесен за вграждане на IP телефония във вече съществуващи мрежи, базирани на ISDN PBX<sup>25</sup>, включително и IP базираните PBX системи.

Първата версия на H.323 е публикувана от ITU през Ноември 1996 г. с акцент да се даде възможност за видео конферентна връзка през локална мрежа, но бързо бе приет от индустрията, като средство за предаване на гласова комуникация през различни IP мрежи, включително WANs<sup>26</sup> и Интернет. С течение на годините H.323 е бил ревизиран и публикуван отново с необходимите подобрения от към по-добра гласова и видео функционалност при преноса им през PSTN, като всяка нова версия е обратно съвместима с предишната такава. Една от силните страни на H.323 е относителното ранно наличие на набор от стандарти, които не само определят базови модели на осъществяване на повикване, но също така дефинират и услуги, необходими за справянето с очакваната бизнес комуникация.

H.323 е протокол използван, например, от Microsoft Netmeeting за VoIP обаждания. Този протокол притежава няколко елемента свързани един с друг:

1. Терминали, клиенти, които създават VoIP връзката. Също така терминалите могат да си комуникират един с друг и притежават допълнителни елементи за по-добра скалируема визия;
2. Многоточков контрол на единиците за предоставяне на конференция;
3. Шлюзове - отправни точки за преобразуване на TCP/IP в PSTN;
4. „Gatekeepers“, които по същество действат по следния начин:
  - a. Притежават преводачески услуги, за да използват имена вместо IP адреси;
  - b. Също притежават контрол за разрешаване или забраняване на някои потребители или потребителски станции;

---

<sup>23</sup> ISDN - Integrated Services Digital Network – Цифрова мрежа за интегрирани услуги.

<sup>24</sup> SS7 – Signaling System No. 7 – набор от телефонни сигнали и протоколи, които се използват от PSTN.

<sup>25</sup> PBX – Private Branch Exchange – частен клонов обмен или с други думи телефонна обмяна на услуги в бизнес офис.

<sup>26</sup> WAN – Wide Area Network – мрежа преминаваща през някакви териториални граници – били те на компании, на области или на държави.

с. Управление на трафика;

5. Прокси сървъри също се използват;

Следват обяснителни бележки, за всяка една от горните подточки.

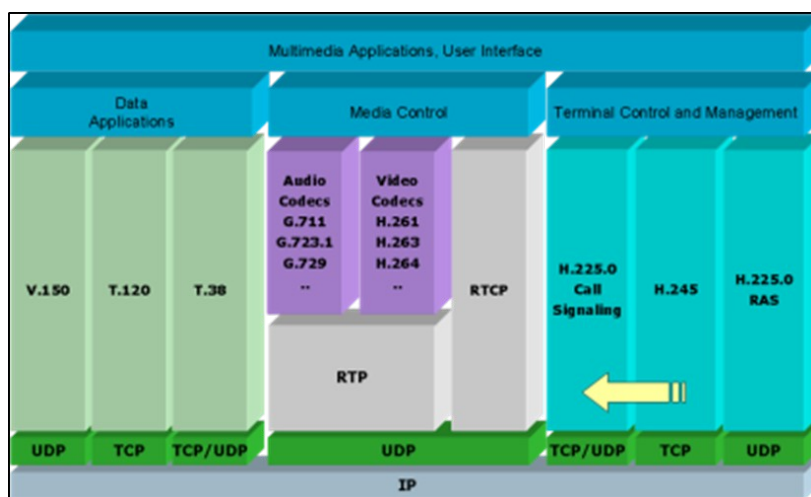
### 1.7.1 Терминалите

Терминалите в H.323 мрежа са най-основните елементи във всяка H.323 система, тъй като това са устройства, които потребителите обикновено срещат. Те могат да съществуват под формата на прости IP телефони или мощна система за видеоконферентна връзка с висока разделителна способност.

Във вътрешността на един H.323 терминал се намира нещо, което сочи към Протоколния стек, който от своя страна изпълнява функции определени от цялата H.323 система. Протоколния стек включва изпълнението на основния протокол дефиниран от препоръка на ITU-T – H.225.0 и H.245.

Фигура 1 изобразява пълната функционалност, която се осъществява при един видео или звуков разговор. В действителност в повечето H.323 системи не се прилага такъв широк спектър от възможности.

Фиг. 1



### **1.7.2 Многоточков контрол на единиците (MCU<sup>27</sup>)**

Многоточковия контрол на единиците (Multipoint Control Unit) е отговорен за управлението на многоточковите конференции и се състои от две логически лица, посочени като Multipoint контролер (Multipoint Controller) и Multipoint процесор (Multipoint Processor). От практическа гледна точка многоточковият контрол се изразява в мост между изградената конферентна връзка, но не прилича на моста, който би бил изграден от една PSTN връзка. Най-значимата разлика, обаче е, че един такъв контрол на единица в H.323 може да ни осъществи и видео връзка или превключване на видео връзка в допълнение на нормалната аудио връзка, която се осъществява. Някой такива контролери също така могат да осигурят и многоточково прехвърляне на данни. Предимството в този случай е, че потребителя може да види всички останали участници в една конферентна връзка, не само да чува гласовете им или в определен момент да вижда само един от участниците.

### **1.7.3 Шлюзове**

Шлюзовете са устройства, които позволяват комуникация между H.323 мрежи и други мрежи, като PSTN или ISDN мрежи. Ако една от страните в разговора използва терминал, който не е H.323, тогава разговора трябва да премине през един такъв шлюз за да даде възможност на двете страни да общуват.

Шлюзовете са широко използвани днес, за да се даде възможност на старите PSTN мрежи да могат да се свържат с мрежите H.323, които в момента се предоставят от доставчиците на интернет и други услуги. Шлюзовете се използват и в рамките на едно предприятие, за да се даде възможност на потребителите, които използват IP телефоните да могат да общуват с други потребители използващи PSTN услугите.

Шлюзовете се използват също и с цел да се даде възможност на устройствата предоставящи видео конферентни връзки базирани на H.320 и H.324, да общуват с H.323 системи. Повечето 3G мобилни мрежи днес използват H.324 протоколи и са в състояние да комуникират с H.323 терминали в корпоративните мрежи, като връзката между 3G и корпоративните мрежи преминава именно през такива устройства – шлюзове.

### **1.7.4 “Gatekeepers”**

Gatekeepers са допълнителни компоненти в една H.323 мрежа, която предоставя редица услуги към терминалите, шлюзовете и многоточковия контрол на устройствата. Тези услуги включват регистрация на крайната точка, адресиране, контрол на допускането, потребителска автентикация и т.н. От всички тези функции, които се изпълняват от

---

<sup>27</sup> MCU – Multipoint Control Unit – Многоточков контрол на единиците.

„Gatekeepers“ най-важна се оказва адресацията, тъй като дава възможност на две крайни точки да се свържат помежду си без никоя от тях да се налага да знае IP адреса на другата точка.

Всеки един от така наречените „Gatekeepers“ може да бъде проектиран да работи в един от двата режима – директно пренасочваме или пренасочваме през друг „gatekeeper“. Режима на директно пренасочване е най-ефективния и най-широко използвания. В този режим крайните точки използват RAS<sup>28</sup> протокола, за да научат IP адреса на отдалечената крайна точка и разговора се осъществява директно със отдалеченото устройство. При другия режим сигнала за звънене винаги минава през друг „gatekeeper“. При този режим, устройството, което използваме за „gatekeeper“ трябва да има по-мощен процесор.

Н.323 крайните точки използват RAS протокола, за да общуват с „Gatekeepers“. На същия принцип и „Gatekeepers“ използват RAS протокола за да общуват с други „Gatekeepers“.

Колекция от крайни точки, които са регистрирани само към един „Gatekeeper“ в една Н.323 мрежа се нарича зона. Тази колекция от устройства не е задължително да е свързана в физическа топология (всеки да има достъп до всеки физически). Зоната се определя от мрежовия администратор. Негова е задачата да осъществи достъп на отделните крайни устройства и да ги регистрира в зоната.

### **1.7.5 Прокси сървъри, гранични елементи и Реег елементи**

Граничните елементи и Реег елементите са незадължителни устройства също като „Gatekeepers“, но те предоставят някои услуги, които не са описани в RAS протокола. Ролята на граничните и реег елементите се разбира чрез определението за „административен домейн“.

Административния домейн е колекция от всички зони, които са под контрол на едно лице или организация, като доставчик на услуги. В рамките на доставчика на мрежови услуги може да има стотици или хиляди устройства от тип шлюз, телефони, видео терминали или други мрежови компоненти, които са елементи от Н.323. Доставчикът на услуги може да организира устройствата в „зони“, които му дават възможност най-добре да управлява всички свои устройства, като например може да раздели зоните по административни градове. Взети заедно всички зони, които има един доставчик на услуги ще се появят при друг доставчик, но вече ще се разглеждат като административен домейн. Това създава една йерархия между доставчиците и територията, която всеки от тях покрива.

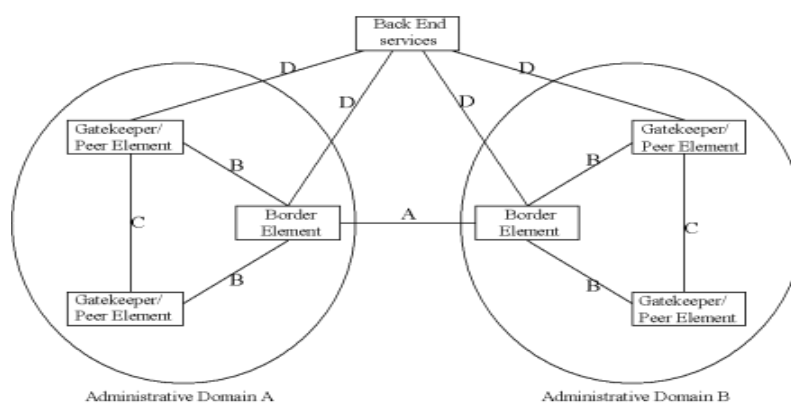
---

<sup>28</sup> RAS – Remote Access Service – Услуга за отдалечен достъп.

Граничните елементи, обикновено са устройства, които стоят на края на административния домейн и комуникират с други гранични елементи от друг домейн. Тази комуникация може да включва неща като: информация за достъпа и автентикацията на потребителите, цени на обажданията, или друга важна информация необходими за осъществяване на връзка между двата административни домейна.

Peer елементите са устройства и съоръжения в рамките на административната област, които повече или по-малко помагат разпространението на събраната, от граничните елементи, информация. Такава архитектура е предназначена да позволи мащабни внедрявания в рамките на операторските мрежи.

Фигура 2 илюстрира административна област с граничните елементи, партньорските елементи и „Gatekeepers“.



Фиг. 2

H.323 освен, че позволява VoIP, също така може да пренася както видео така и друг вид информация. По отношение на VoIP, H.323 може да изпълнява и аудио кодеци G.711, G.722, G.723, G.728 и G.729, а за видео поддръжка H.261 и H.263.

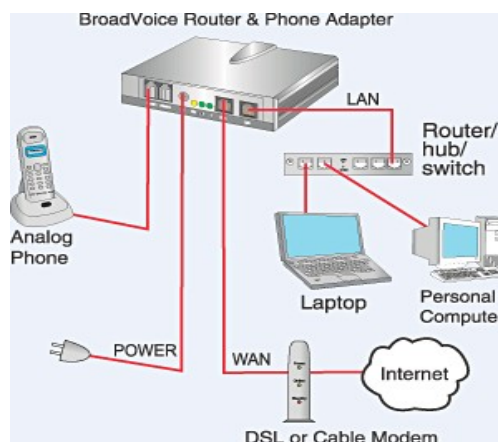
## 2. Приспособяване

### 2.1 Потребителския пазар

Основното развитие, което започна през 2004 година беше въвеждане на услугата за пренос на глас през интернет на масовия пазар, чрез използването на съществуващия вече ширококолов интернет достъп, като по този начин абонатите получават телефонни обаждания по същия начин, както биха получили през обществената комутируема телефонна мрежа (PSTN). Появиха се компании, които осигуряват пълно обслужване на входящи и изходящи директни и недиректни линии. Много от тях предлагат неограничение вътрешни разговори като потребителите заплащат фиксирана абонаментна месечна такса. Това понякога включва и международни разговори към определени държави. Телефонните

разговори между абонати на един и същи доставчик, обикновена са свободни. Когато тази услуга не е възможна VoIP телефона трябва да се свърже с доставчика на VoIP услуги. Това може да се реализира по няколко начина:

- Специализирани VoIP телефони се свързват директно към IP мрежата, чрез използването на технологии като кабелна (Ethernet<sup>29</sup>) или безжична (Wi-Fi<sup>30</sup>). Те обикновено са проектирани в стила на традиционните бизнес цифрови телефони.
- Аналоговият телефонен адаптер е устройство, което се свързва с мрежата и имплементира електронни импулси за да управлява аналоговия телефон, който е закачен чрез модулен телефонен куплунг. Някои жилищни интернет портали и кабелни модеми имат тази функция вградена в тях.
- Софтуерния телефон е приложен софтуер, инсталиран на компютър свързан към мрежата, и оборудван с микрофон и високоговорители или слушалки. Този софтуер обикновено наподобява екран и клавиатура, по която може да се избират различни опции с помощта на мишката или на реалната клавиатура (Фиг. 3).



Фиг. 3

## 2.2 PSTN и мобилни мрежови доставчици

Става все по-често за телекомуникационните доставчици да използват VoIP телефонията през публичните IP мрежи, за да се свържат отделни центрове и да имат свързаност и с други центрове на други доставчици на телефонни услуги. Този модел също така се нарича „IP свързаност“.

<sup>29</sup> Ethernet – понятие, което се отнася до цялата фамилия LAN устройства, които покрива стандарта IEEE 802.3.

<sup>30</sup> Wi-Fi – технология на безжична мрежа (WLAN) базирана на спецификациите от серията IEEE 802.11.

„Умните“ телефони или още наричани Смартфони и Wi-Fi мобилните телефони могат да имат вграден SIP<sup>31</sup> клиент или да имат възможността да си инсталират приложението, което да използва SIP.

### 2.3 Корпоративна употреба

Поради по-високата си широколентова ефективност и по-ниските разходи, които VoIP технологиите ни предлагат, много бизнес компании решават да мигрират от телефонните системи използващи медни връзки, към VoIP системи, които ще им намалят месечните телефонни разходите. През 2008 г., 80% от новоинсталираните PBX системи бяха VoIP системи.

VoIP решенията, насочени към бизнеса са се превърнали в „унифицирани комуникационни“ услуги, които обслужват цялата кореспонденция – телефонни разговори, факсове, гласови пощи, електронни пощи и други – като отделни единици, които да бъдат доставени чрез всякакви средства, включвае и мобилни телефонни апарати. Два вида конкуренти се появяват на пазара: единия се фокусира да предоставя VoIP услуги на големите и средни бизнес компании, докато другия покрива предприятия от малкия и среден бизнес.

VoIP позволява и двете: гласови и други пакети от данни да бъдат пренесени през мрежата, което значително намалява разходите за инфраструктурата.

Цените на разширения VoIP са по-ниски, отколкото за PBX или други системи. VoIP комутатори могат да работят на обикновен хардуер, като например персонални компютри или Линукс системи. Те не са устройства със затворена архитектура, точно обратното, те разчитат на стандартни интерфейси.

VoIP устройствата имат прост, интуитивен потребителски интерфейс, така че потребителите често могат да правят прости промени в конфигурацията на системата. Телефоните, които поддържат двоен режим позволяват на потребителите да продължат своя разговор докато се движат и преминават от една мрежова клетка в друга и след това преминават във вътрешната интранет Wi-Fi мрежа, така че вече не е необходимо потребителите да носят две устройства – обикновен телефон и клетъчен телефон. Поддръжката на VoIP системите е по-лесна тъй като има по-малко устройства да се наблюдават.

---

<sup>31</sup> SIP – Session Initiation Protocol – Протокол за инициране на сесия. Представлява протокол, опериращ на нивото на приложния слой от OSI модела, който се използва при провеждането на IP и Интернет конферентни разговори, както и за изпращане на кратки съобщения.

### 3. Предимства и недостатъци

#### 3.1 Оперативни разходи

VoIP може да бъде от полза за намаляване на разходите за комуникация и инфраструктура. Примерите включват:

- Телефонни обаждания през съществуващи мрежи за данни, за да се избегне необходимостта за отделни мрежи за глас и данни;
- Способност за предаване на повече от едно обаждане през една широколентова връзка;
- Защитени повиквания през стандартните протоколи. Повечето от трудностите за създаване на сигурна телефонна връзка в сравнение с традиционните телефонни линии, като цифровизацията и цифровото предаване, са вече осъществими посредством VoIP. Необходимо е само да се шифрова и идентифицира съществуващия поток от данни.

#### 3.2 Качество на обслужването

Комуникация по IP мрежата е по-малко надеждна в сравнение с PSTN, тъй като не предоставя мрежов-базиран механизъм, за да гарантира, че пакетите с данни не са изгубени и се доставят в последователен ред. Това е типа на мрежата без да се използва услугата „Качество на услугата“ (QoS). Следователно VoIP имплементации могат да се сблъскат с проблеми като намаляване на латентността на трептене.

По подразбиране, мрежовите рутери се справят с трафика на принципа „първи дошъл, първи обслужен“. Мрежови рутери с висок обем на трафик могат да въведат латентност, която надвишава допустимите прагове за VoIP. Фиксираните закъснения не могат да бъдат контролирани, тъй като те са причинени от физическите разстояния, които пакетите трябва да изминат, но латентността може да бъде сведена до минимум чрез маркиране на гласовите пакети от данни като чувствителни към закъснения.

VoIP пакет обикновено трябва да изчака текущия за да завърши предаването, въпреки, че е възможно да изпревари по-малко важен пакет в средата на предаването. Това обикновено не се прави, особено за високоскоростни връзки, при които времето за предаване на пакети, дори с максимален размер, е значително малко. Една алтернатива, която по-бавните DSL връзки предлагат е намаляването на максималното време за предаване на единица пакет.

ADSL модемите осигуряват жична връзка (или жична връзка през USB) към нашето мрежово оборудване, но всъщност те представляват модеми, които предават връзката

асинхронно. Те използват ATM Adaptation Layer 5<sup>32</sup> за да сегментират всеки пакет в серия от 53 байтови пакети за да се предадат по мрежата от изпращача до получателя и обратно.

По-голямата част от доставчиците на DSL използват само един виртуален кръг за всеки клиент, дори и за тези, които използват VoIP услугата. Всеки мрежови пакет трябва да бъде напълно изпратен преди да се започне изпращането на друг такъв пакет.

Гласът, заедно с всичката друга информация пътува през IP мрежата в пакети с фиксирана максимална големина. Тази система може да бъде по-податлива на задръствания и DoS<sup>33</sup> атаки в сравнение с традиционните верижно свързани системи.

Фиксираните закъснения не могат да бъдат контролирани, тъй като те са причинени от физическото разстояние, което пакетите трябва да преминават. Те са особено проблематични, когато са включени сателитни вериги, защото разстоянието от геостанциите до сателита и обратно предполага забавяне на връзката с около 400-600 милисекунди.

Механизма на качество на услугата кара VoIP да използва UDP, а не TCP поради факта, че ако пакета пристигне в мрежа, където има прекалено много трафик, и пакета бъде изхвърлен да не се изисква повторно препращане на същия пакет, защото това повторно препращане обикновено предизвиква много латентност.

### 3.3 Податливост от спиране на тока

Телефоните, които обикновено се използват при аналоговите услуги, директно са свързани към телефонните линии на компанията доставчик на телефонната услуга. Същата компания предоставя и директно свързване на телефона към токовата мрежа, която е независима от основната токова мрежа, предоставена на домакинствата.

IP телефоните и VoIP телефоните са свързани към рутери или кабелни модеми, които обикновено зависят от електрическата мрежа на домакинството или предприятието. Някои VoIP доставчици предоставят модеми с допълнителна батерия, като по този начин може да осигури непрекъснато обслужване до няколко часа в случай на токова авария. Така, че в този случай аналоговите телефони имат предимство.

Някои доставчици на VoIP услуги, предлагат маршрутизиране на повикванията към други клетъчни телефони. Това се прави в случай, че устройството с което искаме да се свържем е недостъпно.

---

<sup>32</sup> ATM Adaptation Layer 5 – AAL5 – метод използван от VoIP, за подобряване качеството на услугата.

<sup>33</sup> DoS – Denial of Service – Атака чрез отказа на услугата.

### 3.4 Защита

VoIP телефонните системи са податливи на атаки, какво всички интернет-свързани устройства. Това означава, че кракерите, които знаят уязвимостите на системата (като несигурни пароли и т.н.) могат да започнат атаки като: отказа от услугата, събиране на информация, запис на данни, и да влязат в пощенските кутии на потребителите, като по този начин ще си осигурят достъп до всички данни и гласови записи.

Друго предизвикателство е маршрутизирането на VoIP трафик през защитни стени и през така наречения NAT<sup>34</sup>. За да се даде възможност на VoIP разговори да се провеждат към и от защитни стени се използват частни сесиини контролери. Например Skype използва собствен протокол за да провежда разговори през маршрутизаторите до други Skype връзки в мрежата, който протокол му позволява да преминава през симетрични NAT и защитни стени. Други методи за преминаването през NAT включват използването на протоколи като STUN<sup>35</sup> или „Създаване на интеративна връзка“.

Много от потребителските VoIP решения не поддържат криптиране на връзката и все пак осигуряването на защитен телефонен разговор е по-лесно реализуем с VoIP отколкото с обикновените телефони. Като резултат от това е относително лесно да се подслуша един VoIP разговор и дори да му бъде променено съдържанието. Нападателят снабден със софтуер за подслушване на пакети, може да прихване вашето VoIP обаждане, ако не сте в защитен VLAN<sup>36</sup>.

По-нататъшни изследвания показват, подслушването на фибро-оптична оптика без детектор е невъзможно. Това означава, че ако гласът вече е в такава мрежа подслушването става невъзможно.

## 4. Необходим хардуер за изграждането на VoIP мрежа

### 4.1 Медна шлюзове

През 1995 г. е представена първата технология за телефония през Интернет. Предложеният тогава телефон е примитивен в сравнение с наличните технологии днес. Софтуерът е оригинално замислен да работи на компютър 486 с 33 MHz процесор или по-висок, чрез който потребителя ще разговаря с друг потребител, използвайки модема на компютъра, звуковата карта, (високоговорител и микрофон). В процеса на трансфер, софтуерът трансформира (компресира) гласа, който е говорен в микрофона. След това

---

<sup>34</sup> NAT – Network Address Translator – Преобразуване на мрежови адреси. Технология, при която адресите на получателя и подателя в IP пакета биват пренаписани от маршрутизатора или защитната стена.

<sup>35</sup> STUN – Session/Simple Traversal Utilities for NAT – протокол осъществяващ връзка на устройства стоящи зад NAT стени.

<sup>36</sup> VLAN – Virtual LAN – Виртуална частна мрежа.

компресираният глас се транспортира чрез IP пакети във формат на стандартна интернет сесия. С тази технология обаче разговорът е ограничен само до два компютърни потребителя (компютър към компютър).

Около година по-късно, през март 1996 г. компанията VocalTec обявява, че ще работи с други фирми за производство на хардуер наречен Voice Gateway, който да позволява аудио връзки между интернет телефон и телефон от публичната комутируема мрежа (PSTN). Все пак остава едно предизвикателство, а именно как да се адресира и достигне до потребител на компютър, разположен където и да е по света. За тази цел потребителят трябва да знае IP-адреса на отдалечения компютър, а той не е лесно откриваем, ако не е имало предшестваш контакт с него. Voice Gateway търси друг такъв, в който да е съхранен телефонният номер на търсения потребител. Телефонен номер се намира по-лесно от IP-адрес. По този начин функционалността на Voice Gateway се справя едновременно с препятствията на свързването на мрежата и адресирането.

Как работи един Voice Gateway – при получаване на аналогов глас (стандартен глас) от телефон, Voice Gateway първо дигитализира сигнала и компресира новия цифров сигнал във вид на стандартни блокове данни, известни като IP пакети. Те биват изпратени през Интернет към вход на Voice Gateway, където процесът се обръща. С тази технология е възможно да се правят три различни видове повиквания: компютър към компютър, компютър към телефон и телефон към телефон.

Преносът на интернет телефония започва от повикващата страна – системата, компютър, микрофон и слушалки, искаща да се свърже с повикваната страна – телефон от обществената телефонна мрежа. Доставчикът на интернет на повикващата страна имайки нужният софтуер се свързва с повиканата страна и предоставя телефонния номер на интернет доставчика, който предоставя услугата VoIP. Използвайки микрофон, обаждачата се страна тогава говори с него и гласовият сигнал се прехвърля на Voice Gateway, където се дигитализира (ако сигнала не е цифров). От тук IP пакетите се прехвърлят през Интернет по път, който е определен от Voice Gateway на доставчика, докато достигнат отдалечения Voice Gateway. Той от своя страна превръща IP пакетите в гласов сигнал и прехвърля гласът на местния PSTN на повикваната страна. От тук, телефонът на повикваната страна ще сигнализира постъпващо повикване. Двете страни могат да проведат напълно двупосочен (дуплексен) разговор. Със същия пример лесно могат да бъдат описани и връзките компютър към телефон и компютър към компютър.

Медиа шлюзовете също така могат да изпълняват и други функции освен изброените горе, функции като събиране на статистическа информация, премахване на ехото,

непредаване на паузите по време на разговор и др. За всяко обаждане се създава отделна RTP сесия, като задачата на медия шлюза е създаването на пакетите пренасящи частите на речта. Интерфейсът към PSTN, освен връзка с традиционните телефонни мрежи, е и алтернативен път за гласовия трафик, в случай на претоварване на VoIP мрежата, както и в случаите на прекъсване на някоя VoIP връзка или отпадане на мрежови елемент.

Медиа шлюзовете могат да съществуват под различни форми. Възможно е те да бъдат както отделни телекомуникационни хардуерни елементи, така и ролята им да бъде изпълнявана от персонален компютър с работещ на него VoIP софтуер. Медиа шлюзовете могат да изпълняват функциите на един или няколко от изброените по-долу видове:

- Магистрален шлюз, служещ за интерфейс между традиционната телефонна и VoIP мрежа, управляващ голям брой цифрови връзки. Той предоставя различни видове интерфейси към PSTN.
- Жилищен шлюз, осигуряващ стандартен аналогов интерфейс на VoIP мрежата. Примери за това са кабелните модеми, xDSL и широколентовите безжични устройства. Този вид шлюз дава достъп на крайните потребители до VoIP мрежата.
- Шлюз за достъп представляващ аналогов или цифров интерфейс между обикновена телефонна централа и VoIP.

#### 4.2 Медиа шлюз контролери

Медиа шлюз контролерите осигуряват сигнализация и контрола върху обажданията и всички предлагани услуги във VoIP мрежата. Други техни функции са съпоставянето на телефонни номера и мнемонични имена в IP адреси, намиране на адреси/потребители, управление на ресурсите, авторизация и автентикация на потребителите. При наличие на връзки към PSTN, медия шлюз контролерите преобразуват сигнализацията SS7<sup>37</sup>, използвана в традиционната телефония, в конкретния за VoIP мрежата протокол за сигнализация. Сигнализацията във VoIP мрежата служи за управление на връзката между двете крайни точки и уговаряне на нейните параметри. След изпълнение на тази задача, шлюз контролера не изпълнява други функции по време на разговора, до момента в който някой от двете крайни точки изпрати съобщение за смяна на статуса на връзката или шлюз контролера уведоми някой от тях за наличие на изчакващо обаждане или смяна на конфигурацията на връзката. Пакетите, съдържащи гласови данни, не минават през медия шлюз контролерите, а се обменят директно между участниците в разговора или преминават през медия шлюз в случаите, в които има нужда от преобразуване на сигнала.

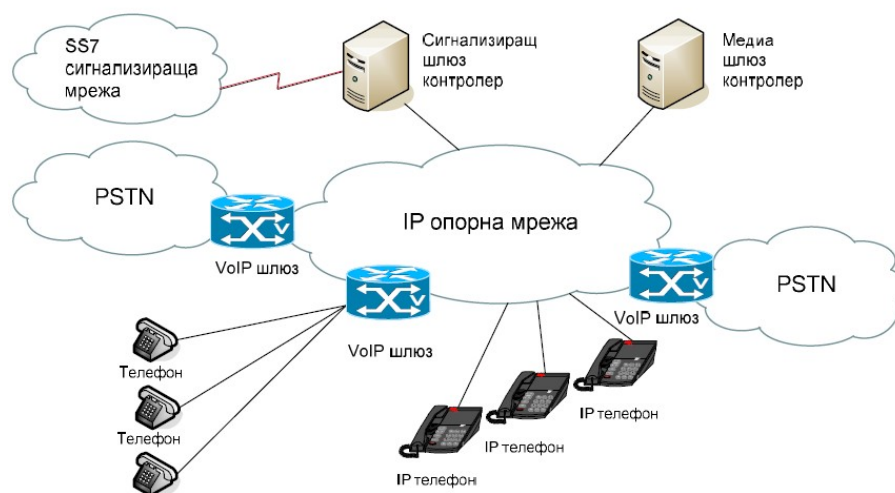
---

<sup>37</sup> SS7 – Signal system #7 – набор от телефонни протоколи за сигнализация.

Възможно е ролята на шлюз контролера да бъде разделена на сигнализиращ шлюз контролер и медиа шлюз контролер. За всички обаждания, с начална и крайна точка в пределните на VoIP мрежата, е достатъчна функционалността на медиа шлюз контролера. В случаите, когато е необходима връзка между PSTN и VoIP мрежите, се налага използването на сигнализиращи шлюз контролери. Техните функции са съсредоточени върху преобразуване на служебни съобщения и сигнализацията, което е необходимо за свързване на двата вида мрежи.

### 4.3 IP Мрежа

Възможно е цялата VoIP мрежа да бъде разгледана като един логически комутатор, осигуряващ връзките между разпределените системи (Фиг. 4).



Фиг. 4


IP инфраструктурата трябва да гарантира безпроблемно доставяне на гласовите данни и сигнализиращия трафик. За това е необходимо да се установят изискванията за допълнителна честотна лента, с оглед бъдещото добавяне на гласовия трафик и при нужда да се увеличи капацитетът на връзките. Дори след осигуряване на необходимия капацитет е препоръчително използването на механизми, осигуряващи качество на услугите, за да се гарантира приоритет на гласовия трафик и сигнализиращите протоколи. Изискване към IP инфраструктурата е поддържането на тези механизми. Поради високата чувствителност на гласовите данни към закъснение и загуба на пакети, времето за възстановяване на функционалността на IP мрежата при отпадане на връзка или устройство трябва да бъде сведено до минимум.

### 4.4 Телефонни апарати

В момента на пазара цари изобилие от IP телефони и хибридни телефони поддържащи както VoIP така и PSTN. Цената на един VoIP телефон зависи от функционалността, която

той може да предложи. За нашата цел ще разгледаме няколко телефона, които отговарят на нуждите на потребителите от университета.

Първият телефон от тях е модел:

Cisco IP Phone 7911G	
Дисплей:	LCD графичен, монохромен
Функционални бутони:	4
Брой линии:	1
Интегриран комутатор:	Вграден Ethernet комутатор
Високоговорител:	не
Захранване:	PoE/локално
Поддържани протоколи:	SCCP, DHCP, SRTP, CDP, SIP
Поддържане на XML приложения:	да
Интерфейси:	не
Цена:	220 – 280 лв.

Другия телефон е модел:

Cisco IP phone 7931G	
Дисплей:	LCD графичен, монохромен, 192x64
Функционални бутони:	4 + 24
Брой линии:	24
Интегриран комутатор:	10/100 Base-T Ethernet Switch
Високоговорител:	да
Захранване:	PoE/локално
Поддържани протоколи:	SCCP, DHCP, SRTP, CDP, SIP
Поддръжка на XML приложения:	да
Интерфейси:	стерео жак за микрофон и слушалки
Други:	24 тонове за звънене
Цена:	383 – 403 лв.

На пазара също така се предлагат и по-евтини телефони, на не толкова известни производители. Горните модели обаче отговарят на изискванията, а фирмата производител е една от разработчиците на VoIP като услуга.

Друго интересно устройство, което се предлага на пазара е следната мултимедийна клавиатура:

Мултимедийна клавиатура с IP телефон KIP-900 (Изобр. 1). Комплекта съдържа: Интернет клавиатура с IP телефон, модел: KIP-900;



Конектор: USB

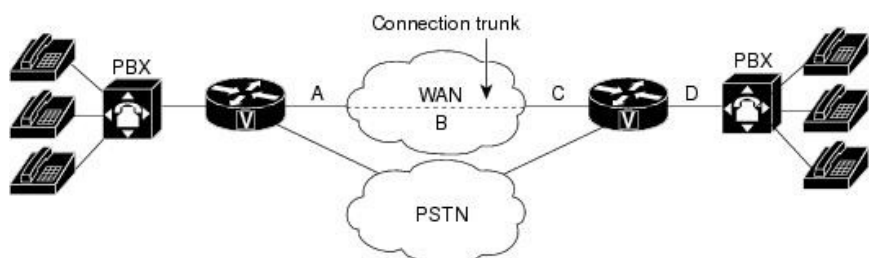
Спецификации: Съвместима с Skype, MSN, AOL, Yahoo и всички други провайдери на интернет, ултра тънък дизайн, вградени високоговорители, двупортов USB 2.0 порт, вход за микрофон, изход за слушалки. Работи под всички версии на Microsoft Windows след Windows 95. Цента на продукта на пазара е между 33 и 45 лв.

## Видове телефонни топологии

### 6.1 Топология използваща само VoIP шлюз

На Фиг. 5 (за легенда виж. Приложение 1) можете да видите как изглежда топологията на мрежа използваща само VoIP шлюз. Както се вижда от фигурата имаме два физически отдалечени обекта (бизнес сгради, офиси и т.н.), които могат да бъдат разположени в отделни метрополиси. При

така зададената схема всеки от двата обекта разполага с вътрешна телефонна централа, която обслужва телефонните постове. При тази схема, ако се поставят



Фиг. 5

два VoIP шлюза на мястото за изход извън вътрешната мрежа на двата обекта може да се изгради връзка, която да използва Интернет като комуникация между двете PBX мрежи. Също така VoIP шлюзовете могат да играят ролята и на маршрутизатори между вътрешните линии и комутируемата публична телефонна мрежа.

Как би се осъществило едно обаждане, ако разглеждаме горната схема: абонат от единия офис вдига телефона и набира вътрешна линия, която физически се намира в другия офис. Обаждането първо стига до телефонната централа, от там централата прехвърля обаждането на VoIP шлюза, който според конфигурацията си може да пренасочи обаждането да премине през Интернет мрежата докато стигне до отсрещния VoIP шлюз или обаждането да премине през комутируемата публична мрежа за телефонни обаждания. След като вече повикването/обаждането е пристигнало до отсрещния шлюз процеса е в обратна посока: шлюза препраща повикването на телефонната централа, а тя на търсения абонат.

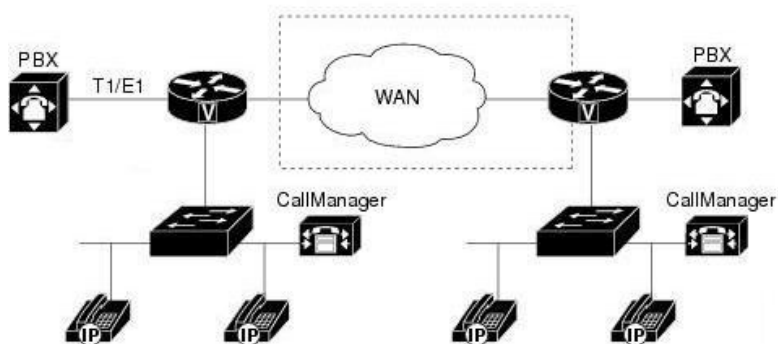
Представената схема е подходяща в случаите, където имаме изградена вече телефонна свързаност и просто желаем да обединим два или повече бизнес обекта. Схемата обаче не е подходяща за целта на дипломната работа понеже в основната сграда на Икономически Университет Варна имаме изградена телефонна свързаност, но във Втори корпус при същия

университет нямаме такава система. Ако и там имаше поставена телефонна централа с вътрешни и външни линии тогава най-добрият вариант за реализиране на VoIP относно двете сгради щеше да бъде горната схема.

### 6.2 Топология използваща само и единствено IP телефонни апарати

Нека да разгледаме схемата в която участват само и единствено IP телефонни апарати (Фиг. 6).

Както се вижда на схемата отново имаме два физически отделени обекта – офис сгради, отделни институции и т.н. Но тука ситуацията е по-различна отколкото в предишния пример. Тука нямаме вътрешна телефонна централа. Единственото, което имаме са IP



Фиг. 6

базирани телефони, които използват вътрешната мрежа, мрежови комутатори, който свързват IP телефоните с VoIP маршрутизатора (шлюза) и VoIP шлюз при двата отдалечени обекта. И също така по един CallManager, чрез който може да се настройват потребителските настройки на телефоните. Като цяло CallManager не е задължителен, защото телефоните могат да се настройват и индивидуално.

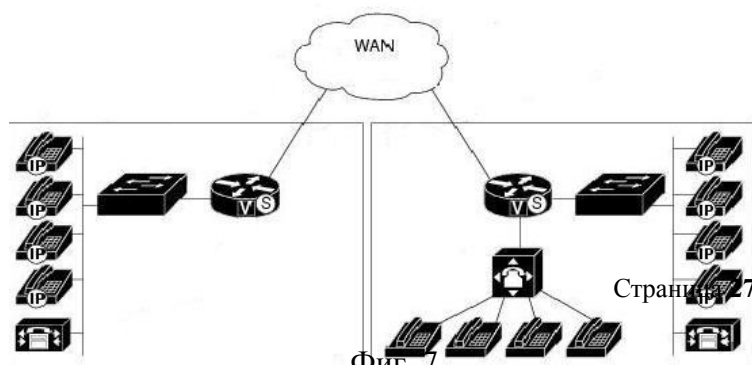
Двата VoIP шлюза комуникират един с друг посредством Интернет и също така могат да бъдат свързани към телефонна централа за да се извършват обаждания и към обикновени телефони свързани към PSTN.

Как се извършва едно телефонно обаждане: абонат от единия офис или населено място набира абонат от друг офис/населено място. Самото обаждане преминава първо през мрежовия комутатор и след това се предава на VoIP шлюза. От там през Интернет единия VoIP шлюз изпраща обаждането на другия, който го приема и го препраща към търсения номер по IP мрежата.

Тази система е напълно реализуема и най-добрия вариант, ако двата ни бизнес обекта нямат вътрешна телефонна централа и са процес на изграждане на вътрешната IP мрежа.

### 6.3 Топология от смесен тип

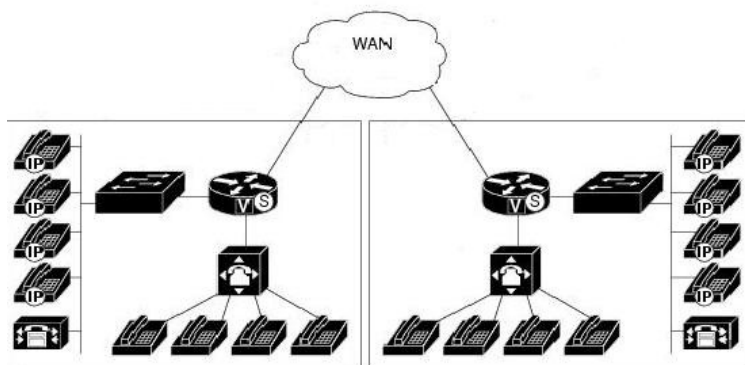
Топологията от смесен тип представлява и IP телефони, и цифрови телефони свързани



Фиг. 7

посредством телефонна централа. На Фиг. 7 може да видите схема на една такава топология. Разбира се на обекта в дясно е добавен мрежов комутатор и към него са прибавени IP телефонни апарати и телефонен мениджър за управление на IP номерата и конфигурацията на телефоните, но това не е задължително.

Отново връзката между двата обекта става посредством Интернет и е напълно безплатна. Разбира се ако предположим, че в ляво е схемата на Втори корпус тогава и там може да имаме класическа телефонна централа, която да е свързана към VoIP шлюза (Фиг. 8). Но на този етап знаем, че във Втори корпус няма никаква телефонна свързаност, за това бихме предпочели мрежата, която ще обслужва телефоните да бъде IP базирана и да не добавят допълнителни разходи за PSTN централа.



Фиг. 8

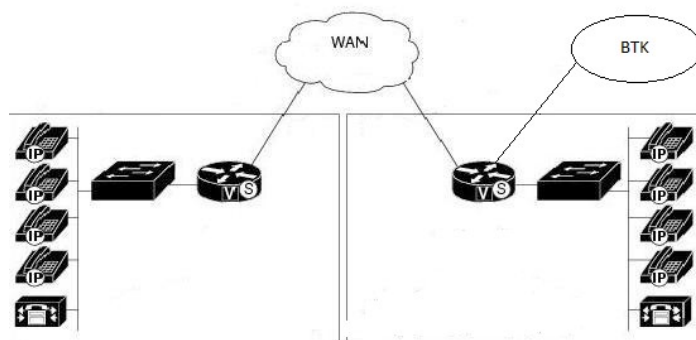
При така устроената схема номерата от Втори корпус ще могат да излизат извън централата чрез номерацията, която е зададена в централата, която се намира в Първи корпус на университета.

## Глава Втора

### Изграждане на проект използващ Интернет услугата VoIP за преминаване на Икономически Университет Варна към Интернет Телефония

#### 2.1 Подход и план на разработване на проекта

Схемата по която избираме да работим е същата като на Фиг. 9. Като се вижда, тя много прилича на Фиг. 7, но ние сме премахнали изцяло старата телефонна централа защото целта ни е цялостно преминаване на двата корпуса към IP телефония. Като при тази схема съществуват няколко варианта за изходящи обаждания. Този на който съм се спрял е VoIP сървър да е свързан към външните линии



Фиг. 9

посредством хибридна платка и през нея да се преминава, когато ни се налагат изходящи повиквания. При тази схема още повече може да оценим всички положителни страни на IP телефонията:

- В единия корпус имаме телефонна централа, която посредством VPN е свързана към втория корпус;
- При евентуално разпадане на VPN връзката втори корпус ще има входящи и изходящи повиквания само в рамките на корпуса;
- Всички IP телефони ще използват изградената вече интернет мрежа на университета – т.е. няма да се налага да се пускат допълнителни кабели, стига до мястото където искаме да имаме телефон да има компютър или поне да има място за компютър;

Недостатък на схемата е че при разпадане на връзката между централата, втори корпус няма да може да набира външни линии, но това може лесно да се реши чрез закупуване на отделна платка за изходящи повиквания и нов номерационен план, който да обслужва само втори корпус.

## 2.2 Необходима техническа част за реализирането на проекта

За реализирането на проекта ще имаме нужда от два отделни компютър-сървър, които всъщност ще играят ролята на Сървъри осигуряващи виртуалната частна мрежа между двете отделни сгради, а също така и VoIP шлюзове.

След направени проучвания и постъпили предложения от различни фирми предлагащи компютърен хардуер (фирми като: Stemo Ltd., Most Bulgaria, CNsys) се спрях на следното предложение (Таблица 2):

### 1. Сървърна с-ма Dell PowerEdge R710:

<i>Компонент</i>	<i>Марка и модел</i>	<i>кол-во</i>
<b>Base</b>	2U Rack Chassis, Up to 6x 3.5" HDDs, No Internal TBU Support	x1
<b>Processor</b>	Intel Xeon E5606, 4C, 2.13GHz, 8M Cache, 4.80GT/s, 80W TDP, DDR3-1066MHz	x2
<b>Memory</b>	4GB Memory for 2 CPUs, DDR3, 1333MHz	x1
<b>Primary Hard Drive</b>	300GB, SAS 6Gbps, 3.5-in, 15K RPM Hard Drive (Hot Plug)	x4
<b>Controller</b>	PERC H700 Integrated RAID Controller, 512MB Cache	x1
<b>Optical Drive</b>	16X DVD+/-RW Drive SATA	x1
<b>PSU</b>	High Output Power Supply, Redundant (2 PSU), 870W, Performance BIOS Setting	x1
<b>Accessories</b>	Riser with 2 PCIe x8 + 2 PCIe x4 Slots iDRAC6 Express Server Management Card Embedded Gigabit Ethernet NIC with 4P TOE, 2U Rack Bezel, Sliding Ready Rails	x1

Таблица 2

На горната таблица са изброени основните хардуерни части, с които трябва да разполагат нашите сървърни машини за да изпълнява функциите описани в дипломната работа. Цената на изброения хардуер е 4742 лв. (без ДДС) за единична бройка. На следващата таблица (Таблица 3) има изброени допълнителни опции, които ще допълнят сървърната конфигурация за да стане по-издръжлива на натоварване. Това е постигнато с добавянето на още един процесор, допълнително добавяне на повече твърди дискове за повече свободно място<sup>38</sup> и пет годишна гаранция (по време на проучванията се получиха различни времеви интервали относно гаранционния период, така, че времето на гаранцията зависи изцяло от доставчика и фирмата от която смятаме да закупим сървърните машини).

#### 1. Опции за PowerEdge R710:

Компонент	Марка и модел	кол-во	Ед. цена
<b>Processor</b>	2 x Intel Xeon E5620, 4C, 2.40GHz, 12M Cache, 5.86GT/s, 80W TDP, Turbo, HT, DDR3-1066MHz	x1	662лв
<b>Memory</b>	8GB Memory for 2 CPUs, DDR3, 1066MHz	x1	227лв
<b>Warranty</b>	5Yr Basic Warranty - Next Business Day	x1	364лв
<b>Primary Hard Drive</b>	4x600GB, SAS 6Gbps, 3.5-in, 15K RPM Hard Drive (Hot Plug)	x1	562лв

Таблица 3

Новата цена на сървърната машина стана 6557 лв. (без ДДС).

Друг вариант, значително по-евтин, но и не с чак толкова добри параметри е следната конфигурация (Таблица 4):

#### Сървърна с-ма HP DL180G5 E5405 (2 GHz):

Компонент	Марка и модел	кол-во
Base	HP DL180G5 E5405 (2 GHz) 2x1GB 2x160G HP-LFF-SATA DVD R-RW	x1
<b>Processor</b>	Quad-Core Intel Xeon processor E5405 (2.00 GHz, 80 W, 1333 MHz FSB)	x2
<b>Memory</b>	4GB Memory for 2 CPUs, DDR3, 1333MHz	x1
<b>Primary Hard Drive</b>	2x160G HP-LFF-SATA	X2
<b>Controller</b>	PERC H700 Integrated RAID Controller, 512MB Cache	x1
<b>Optical Drive</b>	16X DVD+/-RW Drive SATA	x1
<b>PSU</b>	High Output Power Supply, Redundant (2 PSU), 870W, Performance BIOS Setting	x1
<b>Accessories</b>	Riser with 2 PCIe x8 + 2 PCIe x4 Slots iDRAC6 Express Server Management Card Embedded Gigabit Ethernet NIC with 4P TOE,	x1

<sup>38</sup> По принцип сървърната машина, която ще служи за VoIP шлюз няма нужда от повече от 120GB твърд диск. В конфигурацията обаче предвиждаме и пускането на Voicemail, който има нужда от място за пазене на гласовите съобщения.

	2U Rack Bezel, Sliding Ready Rails	
--	------------------------------------	--

Таблица 4

Този вариант на пазара може да се намери за 1959лв (с включен ДДС). Въпрос на избор и на бюджет е кой вариант да се избере.

След като вече имаме представа от какви машини имаме нужда, сега трябва да определим и каква платка трябва да закупим за да може да провеждаме изходящи повиквания. Избираме платката да е хибридна, поради причината, че може да получим обаждане от цифрова система по цифрова линия или по аналогова такава. Платката която ни удовлетворява е модел: 1TDM2460EF на фирма Digium, но тази платка е само за 8 изходящи външни линии, а те не са ни достатъчни. Затова ще се наложи закупуването на две такива платки.

Равносметка при закупуване на най-скъпия хардуер (Таблица 5):

Таблица 5

Брой	Стока	Единична цена	Крайна цена
250	Cisco IP phone 7931G	403 лв.	100 750 лв.
2	Dell PowerEdge R710	7 868 лв.	15 736 лв.
2	1HA8-0008BF	1 448 лв.	2 896 лв.
Крайна сума:			119 382 лв. (с ДДС)

Същата равнoсметка, но с ограничен бюджет би изглеждала така при минимални изисквания към хардуера (Таблица 6):

Таблица 6

Брой	Стока	Единична цена	Крайна цена
250	Cisco IP Phone 7911G	220 лв.	55 000 лв.
2	HP DL180G5 E5405	1 959 лв.	3 918 лв.
2	1HA8-0008BF	1 448 лв.	2 896 лв.
Крайна сума:			61 814 лв. (с ДДС)

Разликите в цените са огромни поради факта, че към проблема може да се подходи по няколко начина – в горните примери съм извадил цените за максималния и минималния план.

Вариант за реализирането на проекта без закупуването на отделен сървър също съществува. Ако вече разполагаме с компютърна машина, която действа като сървър и ако тя разполага с необходимите ресурси да поеме още две услуги (VPN и VoIP) може да използваме нея. За този вариант най-удачното решение е да се използва виртуална машина, която да играе ролята на VoIP шлюз и да работи само за тази услуга. По този начин може да спестим сумата, която трябва да дадем за отделен сървър.

### *2.3 Необходима софтуерна част за реализирането на проекта*

За целите на проекта ще използваме следния софтуер:

- ❖ Операционна система – Ubuntu Server (No GUI) v11.10;
- ❖ Софтуер за отдалечено влизане в системата – OpenSSH Server v5.8;
- ❖ Софтуер за криптиране на данните и създаване на публични и частни ключове – OpenSSL v1.0.0e
- ❖ VoIP софтуер – Asterisk v1.8.8.1;
- ❖ База данни – PostgreSQL v8.4.8;
- ❖ Софтуер за изграждането на виртуална частна мрежа – OpenVPN v2.2.2;
- ❖ Клиент за отдалечена връзка между сървъра и Windows базиран потребителски компютър – Putty v0.61;

След като вече сме определили необходимия ни софтуер, може да преминем към инсталирането, конфигурирането и пуска на системата. Тъй като двата сървъра ще играят една и съща роля ще разгледаме инсталирането и конфигурирането само на единия от тях.

## 2.4 Инсталиране и конфигуриране

### 2.4.1 Инсталиране на операционната система и добавяне на хранилища

Смятам да пропусна частта с инсталирането на операционната система, защото тя не е обект на дипломната работа. Само ще спомена, че инсталацията е стандартна, без добавянето на какъвто и да било софтуер, който се предлага от инсталатора. Хранилищата, които използвам са достъпни от официалния сайт на Ubuntu, и все пак ще ги добавя тук:

```
root@ubuntu:/home/voip# cat /etc/apt/sources.list
deb http://us.archive.ubuntu.com/ubuntu/ oneiric-updates main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/oneiric-updatesmain restricted

deb http://us.archive.ubuntu.com/ubuntu/ oneiric universe
deb-src http://us.archive.ubuntu.com/ubuntu/ oneiric universe
deb http://us.archive.ubuntu.com/ubuntu/ oneiric-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ oneiric-updates universe

deb http://us.archive.ubuntu.com/ubuntu/ oneiric multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ oneiric multiverse
deb http://us.archive.ubuntu.com/ubuntu/ oneiric-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ oneiric-updates multiverse
```

Хранилищата са места в интернет, от които Линукс операционните системи свалят и инсталират софтуер.

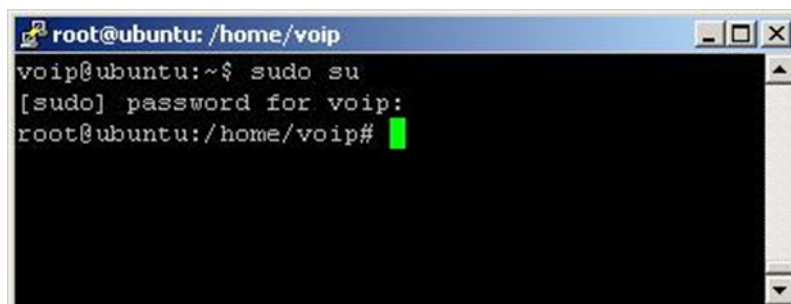
### 2.4.2 Инсталиране и конфигуриране на OpenVPN

Въпреки, че дипломната работа е ориентирана към VoIP и неговата интеграция в ИУ Варна, не може да пропуснем инсталирането и конфигурирането на OpenVPN софтуера поради факта, че реших да използвам виртуална частна мрежа, която да свързва 1-ви с 2-тори корпус на университета.

В следващите стъпки ще покажем и опишем подобно инсталацията и пуска на сървъра, който ще можем да използваме като VPN сървър.

След като влезем в системата (локално или отдалечено) трябва да придобием права на администратор за да може да инсталираме какъвто и да било софтуер. В Линукс администраторския потребител се казва root. При нашата операционна система (Ubuntu Server 11.10) може да придобием правата на администратор чрез командата: `sudo su` (Фиг. 10).

На Фиг. 10 се вижда как чрез командата `sudo su` придобиваме правата на администратора, след като сме били попитани за парола. Можем да познаем дали сме с права на обикновен потребител или с



```
root@ubuntu: /home/voip
voip@ubuntu:~$ sudo su
[sudo] password for voip:
root@ubuntu: /home/voip#
```

Фиг. 10

права на администратор по инициала, който се намира в края на пропта. Ако в края имаме знака долар (\$) значи работим като обикновен потребител, ако края на пропта е със знака диес (#) значи имаме права на root потребител.

Друг начин да инсталираме софтуер е само чрез командата `sudo` последвана от команда, която изисква root права. Тогава интерактивно се взимат правата на root, инсталира се програмата и след това правата за прекъсват. В примерите по-долу ще използваме и двата начина за инсталиране и конфигуриране на системата.

След като вече сме влезли като root ще започнем с инсталирането на OpenVPN. Ще инсталираме софтуера чрез командата:

```
# apt-get install openvpn openssl liblz2-2
```

Нека да обясним какво направихме току що: `apt-get install` е команда в Линукс, която сваля от интернет и инсталира софтуера, посочен като име след нея. Местата от където `apt-get` търси софтуера, са хранилищата описани в предходната точка.

След тази команда сме изброили три различни софтуера: `openvpn`, `openssl`, `liblz2-2`. Следователно `apt-get` ще инсталира първо `openvpn`, след това `openssl` и накрая `liblz2-2`. Предимството на `apt-get` е че ако исканата програма изисква допълнителни библиотеки, `apt-get` ще ми предложи да свали и инсталира и тях заедно с желаната от нас програма.

Нека да кажем по няколко думи за трите програми, които инсталирахме току що:

- ❖ `openvpn` – името на OpenVPN софтуера, който ще използваме за изграждането на виртуална частна мрежа;
- ❖ `openssl` – е софтуер, който се грижи за криптирането на данните, когато минават през интернет пространството с цел да не бъдат компрометирани.
- ❖ `liblz2-2` – е преносима библиотека, която е необходима за компресирането на данните. Тя предлага бързо компресиране и още по-бързо декомпресиране. Декомпресирането не изисква никаква виртуална памет. За да може да използваме `liblz2-2` библиотеката трябва да имаме инсталирана една от основните библиотеки, които използват всички Линукс дистрибуции, а именно `libc6`.

След като изчакаме инсталатора да приключи сме готови с настройка на системата. Следващата стъпка, която трябва да предприемем е създаването на директория с име `scripts`, която трябва да се намира в `/etc/openvpn`. Това може да направим чрез следната команда:

```
mkdir --parents /etc/openvpn/scripts
```

Тази директория ще съдържа скриптове за добавяне на клиент или изтриване на съществуващ вече клиент. Малко по-късно ще добавим файловете, които ще извършват тази услуга.

При вече инсталирания софтуер, трябва да добавим символно устройство, което ще бъде използвано от VPN сървъра, а също така и трябва да заредим необходимите модули. Модула, който трябва задължително да е зареден в ядрото на операционната система е с име `tun`. TUN (съкращението идва от `tunnel`) осигурява приемане и предаване на потребителски програми. Той може да се разглежда като обикновено устройство тип точка-до-точка (`point-to-point`<sup>39</sup>) или Ethernet устройство, което вместо да получава пакети от физическа преносна среда (мрежов кабел, модем, оптичен кабел), получава пакетите от потребителска програма и вместо да ги препраща обратно във физическата преносна среда ги изпраща към потребителска програма.

Когато програма отвори едно такова `tun` устройство, драйвер създава и регистрира съответния мрежови `tunX` или `tapX`. След приключването на програмата и затварянето на тези устройства, отново драйвера автоматично ще изтрие създадените `tunX` или `tapX` устройства и всички мрежови пътища водещи до тях.

Добавянето на символно устройство може да направим чрез следните команди:

```
modprobe tun
mkdir /dev/net
mknod /dev/net/tun c 10 200
```

Първата команда ще зареди в ядрото модула `tun`, втората команда ще създаде директория `net`, която ще е поддиректория на `/dev` и накрая ще създадем символно устройство посредством `mknod`, което ще кръстим `tun`. Добра практика е да създаваме такива устройства в поддиректорията `/dev` (`devices` – в превод: устройства). Ако не сме успели да заредим в ядрото `tun` модула е добре да инсталираме пакета: `module-init-tools`. Това може да направим чрез следната команда:

```
apt-get install module-init-tools
```

За да сме сигурни че при рестарт модула `tun` отново ще бъде зареден трябва да го добавим във файла: `/etc/modules`

```
echo tun >> /etc/modules
```

Следващата стъпка е осигуряване на сигурността при свързване с VPN сървъра. OpenVPN използва OpenSSL за да криптира връзките между сървъра и потребителя.

---

<sup>39</sup> Point-To-Point - Вид мрежова топология, в която всеки от възлите в мрежата е свързан до централен възел чрез 'point to point' връзка. Цялата информация, която се предава между възлите в мрежата, се предава до централния възел, който обикновено е някакъв тип устройство, което препредава информацията до всички други възли в мрежата.

Клиентската ауторизация е базирана на двойката публичен / частен ключ. Тези ключове са в основата на OpenVPN мрежите. Така, че трябва да внимаваме когато ги създаваме.

За улеснението при създаването на ключовете може да използваме примерите, които идват заедно с инсталацията на OpenVPN. Трябва да копираме скриптовете за бързо създаване на ключове в директорията `/etc/openvpn` за да може да ги редактираме според нашите нужди. Това става по следния начин:

```
cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0 /etc/openvpn
mv /etc/openvpn/2.0/ /etc/openvpn/easy-rsa
```

Втората команда ще преименува директорията 2.0 на `easy-rsa`, това го правим с цел по-голямо удобство за нас, а и да спазим йерархията на директорията от която сървърът ще черпи информация относно потребителите и ключовете им.

Следващата стъпка е да свалим от интернет модифицирани версии на два от файловете за лесно създаване на ключове. Ще използваме готови модифицирани вече файлове.

```
wget http://howto.landure.fr/gnu-linux/debian-4-0-etch/installer-et-
configurer-openvpn-sur-debian-4-0-etch/vars --output-document /etc/openvpn/easy-
rsa/vars
wget http://howto.landure.fr/gnu-linux/debian-4-0-etch/installer-et-
configurer-openvpn-sur-debian-4-0-etch/openssl.cnf --output-document
/etc/openvpn/easy-rsa/openssl.cnf
```

Вече може да продължим с параметрите на VPN сървърът. За целта трябва да редактираме файла `/etc/openvpn/easy-rsa/vars`. По-долу можете да видите кратки описания на променливите, на които трябва да им се зададе стойност:

- ❖ `OPENVPN_SERVER` – DNS<sup>40</sup> запис на машината, на която е инсталиран OpenVPN софтуера;
- ❖ `OPENVPN_CLIENTS` – лист, който съдържа имената на потребителите, които ще използват VPN сървърът. Имената трябва да са разделени посредством интервал (шпация);
- ❖ `OPENVPN_IPRANGE` – първите три цифри от IP адреса на VPN сървърът. Тези цифри трябва да са различни от локалните IP адреси;
- ❖ `OPENVPN_LOCALDOMAIN` – ако VPN сървърът е част от локален домейн, на тази променлива се задава името на домейна;

---

<sup>40</sup> DNS - Системата за имената на домейните (Domain Name System) - представлява разпределена база от данни за компютри, услуги или други ресурси свързани към Интернет или частни мрежи, с чиято помощ се осъществява преобразуването на имената на хостовете в IP-адреси. Това улеснява работата на потребителите на Интернет услуги. Вместо да въвежда IP-адрес (комбинация от цифри) за да достигне до даден ресурс в мрежата, потребителят може просто да въведе неговото име (домейн).

- ❖ KEY\_COUNTRY – кода на страната в която се намираме. В нашия случай то ще бъде “BG”;
- ❖ KEY\_PROVINCE – кода на провинцията. В нашия пример сме избрали числото 9000;
- ❖ KEY\_CITY – името на града;
- ❖ KEY\_ORG – името на VPN сървъра (може и да не бъде променяно);
- ❖ KEY\_EMAIL – електронната поща на администратора на сървъра;
- ❖ KEY\_SIZE – по подразбиране тука стойността е 1024, но ако искаме по-голяма сигурност може да променим тази стойност на 2048. Това ще забави договарянето между двете страни (клиент / сървър), поради по-големия ключ, който ще трябва да бъде проверен;

Ето как би изглеждал файла след добавените промени (Фиг. 11):

```

root@ubuntu: /home/voip
export D="/etc/openvpn"

export KEY_CONFIG="/etc/openvpn/easy-rsa/openssl.cnf"

export KEY_DIR=$D/keys

echo NOTE: when you run ./clean-all, I will be doing a rm -rf on $KEY_DIR

export KEY_SIZE="2048"

export OPENVPN_SERVER="vpn.us-varna.bg"
export OPENVPN_CLIENTS="korpus1 korpus2"
export OPENVPN_IPRANGE="10.8.142"
export OPENVPN_LOCALDOMAIN="vpndomain.vpn"

export KEY_COUNTRY="BG"
export KEY_PROVINCE="9000"
export KEY_CITY="VARNA"
export KEY_ORG="$OPENVPN_LOCALDOMAIN Server"
export KEY_EMAIL="admin@email.org"
  
```

Фиг. 11

Създаване на сертифициращ орган – сертифициращия орган е двойка публичен / частен ключ служещ за подписване на други публични ключове. За да създадем нашия сертифициращ орган ще изпълним следните команди:

```

source /etc/openvpn/easy-rsa/vars

export KEY_COMMONNAME="ca.$OPENVPN_SERVER"

/etc/openvpn/easy-rsa/clean-all

/etc/openvpn/easy-rsa/build-ca
  
```

След като сме променили горните променливи може да продължим със създаването на нашите сертификати. Тях ще създадем чрез следните команди:

```
source /etc/openvpn/easy-rsa/vars
export KEY_COMMONNAME="$OPENVPN_SERVER"
/etc/openvpn/easy-rsa/build-key-server server
```

При създаването на сертификатите openssl ни задава въпроси, на които стойностите по подразбиране са записани във файла vars, така че на всички въпроси просто натискаме Enter и продължаваме докато не ни попита дали да подпише сертификата на който отговаряме с “y”:

```
Sign the certificate? [y/n]: y
```

Може да продължим с подsigуряването на защита на нашия сървър като създадем TLS<sup>41</sup> ключ, който ще ни помогне при евентуални атаки. Това правим с командата:

```
openvpn --genkey --secret /etc/openvpn/keys/ta.key
```

За да продължим с конфигурирането на VPN сървъра трябва да копираме още един от примерните файлове и след това да го разархивираме:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
gunzip /etc/openvpn/server.conf.gz
```

След като разархивираме архива, виждаме че в него един единствен файл server.conf. Този файл трябва да бъде редактиран според нашите нужди. В него трябва да бъде посочен пълния път до сертификатите на сървъра, също така и IP адреса на VPN сървъра, както и още допълнителни опции. За по-бързо ще използваме командния инструмент sed. SED претърсва файл по даден шаблон и заменя намерените стойности с други. Ето как би изглеждала командата, която ще ни попълни файла server.conf с правилните пътища и настройки:

```
sed -i \
-e "s/^ca ca\.crt/ca \/etc\/openvpn\/keys\/ca\.crt/" \
-e "s/^cert server\.crt/cert \/etc\/openvpn\/keys\/server\.crt/" \
-e "s/^key server\.key/key \/etc\/openvpn\/keys\/server\.key/" \
-e "s/^dh[\t ]*dh1024.pem/dh \/etc\/openvpn\/keys\/dh$KEY_SIZE.pem/" \
-e "s/^server[\t ].*$/server $OPENVPN_IPRANGE\.0 255\.255\.255\.0/" \
```

---

<sup>41</sup> TLS - Transport Layer Security и неговият предшественик Secure Sockets Layer (SSL) са криптографски протоколи, които осигуряват сигурност на комуникацията по Интернет. TLS и SSL криптиране са сегменти на мрежови връзки над Transport Layer, използвайки асиметрична криптография за личния message authentication code за надеждността на съобщението.

```

-e 's/^;\(tls-auth \)\(ta.key.*\)$/\1/etc/openvpn/keys/\2/' \
-e 's/^;\(.*# Triple-DES\)$/\1/' \
-e 's/^\(status \).*\/\1/var/log/openvpn-status.log/' \
/etc/openvpn/server.conf

```

Ще обясня само първия ред на командата: `sed` търси ред във файла, започващ с `ca.crt` и когато намери такъв ред го заменя с `ca /etc/openvpn/keys/ca.crt`. Останалите редове са аналогични регулярни изрази, които търсят стринг по дадения шаблон и след това `sed` го заменя с друг стринг.

След като редактирахме и този конфигурационен файл следва да определим правата с които ще се изпълнява `OpenVPN` сървърът. Разбира се от гледна точка на защитата искаме нашият сървър да се изпълнява с минимални права, за да се избегне всякакъв опит за компрометиране на информацията. За това първо ще дадем права само за четене на нашите ключове:

```
chmod go+rx /etc/openvpn/keys
```

След това ще определим процеса на `openvpn` да бъде стартиран от потребител `nouser` и група `nogroup`. Това ще стане като разкомментираме редовете в които се оказва потребителя и групата от които ще бъде стартиран сървърът. Отново ще използваме `sed`:

```

sed -i \
-e 's/^;\(user[ \t]*.*\)/\1/' \
-e 's/^;\(group[ \t]*.*\)/\1/' \
/etc/openvpn/server.conf

```

Една от най-важните части в настройката на `VPN` сървърът е да позволим на клиентите да комуникират един с друг. Това е важно, защото по подразбиране тази опция е спряна и тогава клиента може да се свързва само и единствено със сървърът. Позволяването на клиентите да си „говорят“ един с друг отново става като се редактира `server.conf` файла. Това ще направим чрез следната команда (отново ще използваме `sed` за да намерим реда, който е коментиран посредством точка и запетая (;) и да го разкомментираме):

```

sed -i -e 's/^\;client-to-client/client-to-client/' \
/etc/openvpn/server.conf

```

Предпоследната стъпка, която трябва да направим е да добавим мениджъра на отменените сертификати:

```

echo "
# Revoked certificate list
crl-verify /etc/openvpn/keys/crl.pem" >> /etc/openvpn/server.conf

```

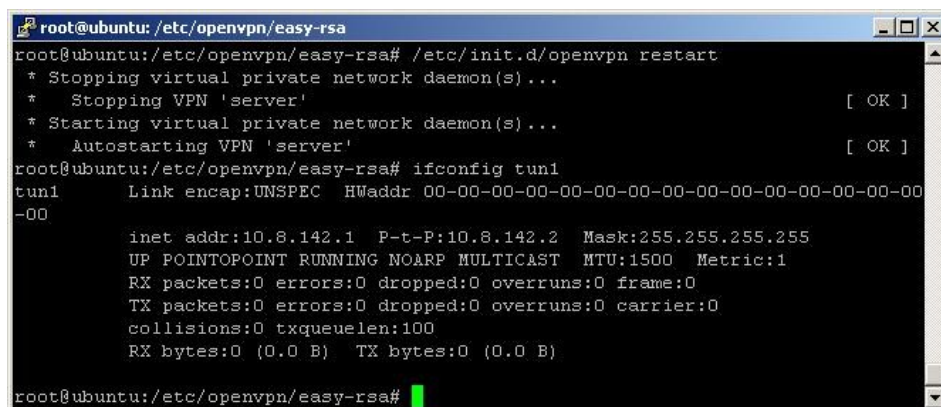
И да създадем празен сертификат:

```
chmod +x /etc/openvpn/easy-rsa/make-crl  
  
/etc/openvpn/easy-rsa/make-crl /etc/openvpn/keys/crl.pem
```

Накрая трябва да рестартираме сървъра за да влязат в сила промените, който до сега правихме:

```
/etc/init.d/openvpn restart
```

Ако всичко е наред екрана ни трябва да изглежда по следния начин (Фиг. 12):



```
root@ubuntu: /etc/openvpn/easy-rsa  
root@ubuntu:/etc/openvpn/easy-rsa# /etc/init.d/openvpn restart  
* Stopping virtual private network daemon(s)...  
* Stopping VPN 'server' [ OK ]  
* Starting virtual private network daemon(s)...  
* Autostarting VPN 'server' [ OK ]  
root@ubuntu:/etc/openvpn/easy-rsa# ifconfig tun1  
tun1 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  
-00  
inet addr:10.8.142.1 P-t-P:10.8.142.2 Mask:255.255.255.255  
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1  
RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:100  
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)  
  
root@ubuntu:/etc/openvpn/easy-rsa#
```

Фиг. 12

За да работи правилно нашия VPN сървър и да може да приема връзки от клиенти, трябва да разрешим на защитната стена да изпълнява това. Тъй като до сега не сме конфигурирали защитна стена на сървъра, ще направим това, като тя ще съдържа само едно правило – ще „маскира“ адресите от локалната мрежа с тези на мрежата, която създава VPN сървъра. Важно е да споменем, че схемата която описва дипломната работа не съдържа никакви защитни стени. Ако решим да използваме такива (за по-голяма сигурност) трябва да се уверим, че в тях е зададено правило да пропуска заявките на клиентите към VPN сървъра. Това най-често става, като се разреши използването от Интернет на порт 1194. За реализирането на правилата на защитната стена първо трябва да създадем скрипта `ip-up.d`. Този скрипт ще бъде стартиран всеки път когато сървъра стартира своите мрежови настройки или с други думи всеки път когато сървъра активира мрежовата си карта. За създаването на скрипта ще използваме следния програмен код, който се стартира в терминала, проверява дали файла съществува, ако не съществува го създава и го прави изпълним:

```
if [ ! -e /etc/network/if-up.d/iptables ]; then  
    echo '#!/bin/sh  
# IpTables rules.' | /usr/bin/tee /etc/network/if-up.d/iptables  
fi  
  
/bin/chmod +x /etc/network/if-up.d/iptables
```

Следващата стъпка е да позволим NAT правилата да работят в нашата система:

```
sed -i -e 's/[# ]*\(net\.ipv4\.conf\.default\.forwarding=\).*\/\1/g'
etc/sysctl.conf
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Чрез следващата команда ще заредим NAT да работи за нашия VPN сървър:

```
iptables -t nat -A POSTROUTING -s $OPENVPN_IPRANGE.0/24 -o eth0 -j
MASQUERADE
```

И накрая трябва да добавим горния код в скрипта, който вече създадохме, за да може да се зарежда със зареждането на системата и на мрежовите настройки:

```
echo "iptables -t nat -A POSTROUTING -s $OPENVPN_IPRANGE.0/24 -o eth0 -j
MASQUERADE" \
| /usr/bin/tee -a /etc/network/if-up.d/iptables
```

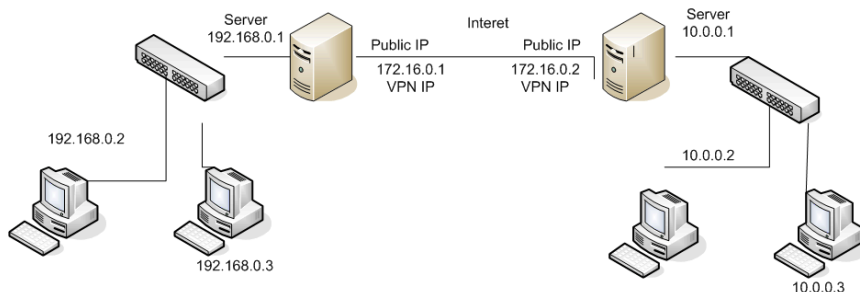
За да позволим на клиентите да имат достъп до локалната мрежа в която оперира сървъра трябва първо да се уверим, че клиентите ни не използват IP номерация от същата мрежа, в която се намира и сървъра. Това е така поради факта, че клиентите се свързват към сървъра отдалечено и сървъра изгражда тунел между него и тях като за целта използва локални IP адреси. Ако искаме клиентите да достигат ресурсите на локалната мрежа в която се намира сървъра, трябва да преминават през маршрут определен от самия VPN сървър. Ако всичко е наред и адреса от IP на клиентите е различен от този на локалната мрежа може да добавим следната конфигурация в конфигурационния файл на сървъра – `server.conf`:

```
ifconfig eth0 | grep "inet " | \
sed -e 's/.*:\([0-9\.]*\)\ [0-9]\{1,3\} .*:\([0-9\.]*\) .*:\([0-9\.]*\)
.*\/push "route \10 \3"/g' \
>> /etc/openvpn/server.conf
```

При така зададените конфигурации вече имаме работещ VPN сървър, който може да бъде достъпван от клиенти от целия свят, стига да им предоставим реален публичен Интернет адрес (IP адрес) и също така и двойката подписани от сървъра публичен и частен ключ. Разбира се сървъра може да се настрои така, че клиентите да използват потребителско име и парола, но този вариант не е никак за предпочитане. В момента, както е конфигуриран VPN сървъра клиентите, които се намират във Втори корпус на Университета, ако искат да се свържат с някого, който се намира в първи корпус на университета, по телефона трябва задължително преди да се обадят да са се свързали към VPN сървъра, за да могат да осъществят локално обаждане. Не възникват проблеми дори, ако се използва софтуер за осъществяване на обаждането, тъй като потребителите ще имат VPN клиентски софтуер

инсталиран на компютъра и ще могат да провеждат обаждания от Втори корпус, към Първи и обратно.

До тук описания вариант е доста евтин и лесен за реализация. Съществува обаче и друг вариант, който представлява следната схема (Фиг. 13):



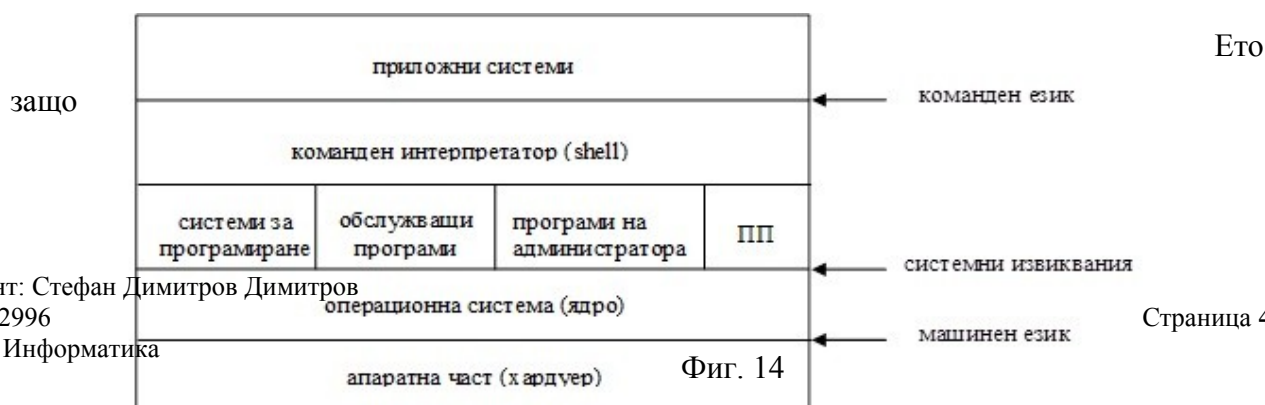
Фиг.13

На Фиг.13 виждаме два сървъра, свързани помежду си посредством VPN тунел. Предимството на този вариант, е че няма нужда клиентите (персоналните компютри) да се свързват всеки път към VPN сървъра, който се намира в Първи корпус (ако приемем, че до тук описанието и конфигурирането на сървъра беше за машина, която се намира в Първи корпус на Университета). Те, клиентите, ще са локално свързани за техния си сървър, който от своя страна е свързан чрез VPN за сървъра, намиращ се в Първи корпус на Университета.

### 2.4.3 Инсталиране на Asterisk

Преди да започнем инсталирането на самия софтуер, трябва да подготвим операционната система за работа като VoIP сървър.

Всяка операционна система в основата си има ядро, което ръководи и оперира всички процеси. Ядрото се грижи за абсолютно всички процеси, които се изпълняват, както и за комуникацията с наличните устройства. То осигурява работата на обвивката и на приложните програми. Обвивката служи за връзка между потребителя и ядрото. Тя може да бъде както графична, така и команден ред. ОС използва и друг вид системен софтуер, който обаче не е част от самата операционна система — драйверите. Те служат за връзка между ядрото на операционната система и съответните физически устройства. Самата ОС има вградени драйвери за определени устройства като процесор, временна памет, твърд диск и др., които осигуряват нейната работа (Фиг. 14).



имаме нужда от подготовка преди инсталирането на Asterisk софтуера. Подготовката се състои в това да сменим ядрото, което операционната система си е инсталирала в началото, с наше собствено ядро. Като нашето ядро ще се различава с няколко параметъра от стандартното.

Ще започнем с достъпването на `/usr/src` директорията. Това е основната директория в Линукс в която се намират така наречените `src` файлове:

```
cd /usr/src
```

След това ще свалим новото ядро от сайта: [www.kernel.org](http://www.kernel.org) – в този сайт са поместени всички излезли стабилни и тестови ядра. В нашия пример ще използваме последното (до написването на дипломната работа) тестово ядро. За да свалим ядрото използваме следната команда:

```
wget
http://www.kernel.org/pub/linux/kernel/v3.0/testing/linux-3.2.0-rc1.tar.bz2
```

След като ядрото бъде свалено се налага да инсталираме някои допълнителни програми с които ще може да построим новото ядро и да го заредим в операционната система. Приложните програми, от които имаме нужда са основни пакети при компилация на софтуер, както и пакети с които ядрото комуникира и една програма която служи за разархивиране на компресирани файлове. Програмите инсталираме отново от хранилищата с помощта на командата `apt-get install`:

```
apt-get install kernel-package libncurses5-dev fakeroot bzip2
build-essential
```

След като вече сме готови, може да разархивираме ново ядро:

```
tar xfv linux-2.6.26.8.tar.gz
```

Следващата стъпка е да направим символна връзка от директорията на новото ядро като също така за по-лесно преименуваме символната връзка:

```
ln -s /usr/src/linux-2.6.26.8 /usr/src/linux
```

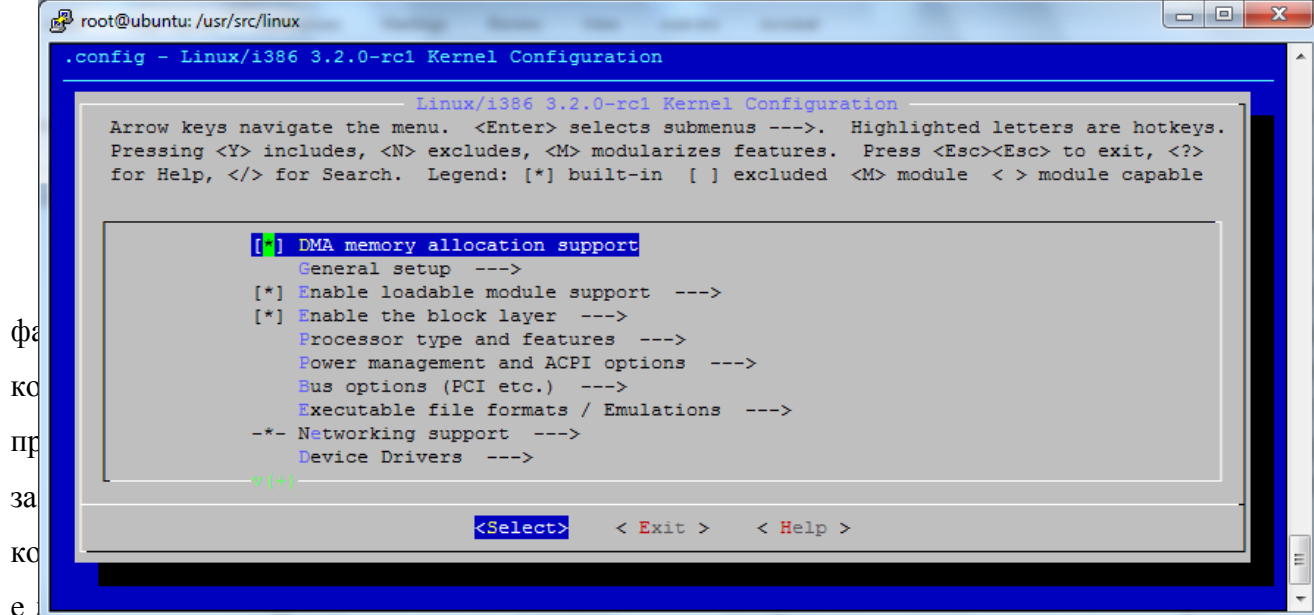
По принцип горната стъпка не е задължителна, но това е неписан стандарт създаден от един от разработчиците на Линукс – Ричард Столман<sup>42</sup>.

Продължаваме с влизане в директорията на архива, който току що разархивирахме и след това трябва да сме сигурни, че никакви изпълними файлове не са останали в архива.

Това може да направим чрез следите команди:

---

<sup>42</sup> [http://bg.wikipedia.org/wiki/Ричард\\_Столман](http://bg.wikipedia.org/wiki/Ричард_Столман)



ядро, защото знаем, че този конфигурационен файл е създаден при инсталиране на операционната система, следователно системата е разпознала наличния ни хардуер и си е изградила вече ядро, подходящо за работата на хардуера. Копирането на конфигурационния файл става чрез следната команда:

```
cp /boot/config-`uname -r` ./config
```

Вече може да започнем настройването на новото ядро. За целта стартираме командата:

```
make menuconfig
```

Както може да видим на Фигура 15 пред нас се появява потребителско меню, в което може да оперираме и да задаваме различни настройки.

Фиг. 15

Причините поради които се наложи да прекомпилираме цялото ядро са следните настройки:

```
Processor type and features >> [*]IRQ balancing
```

```
Processor type and features >> Timer frequency = 1000 Hz.
```

```
Processor type and features >> [*]High Resolution Timer Option
```

```
Processor type and features >> [*]HPET Timer Support
```

```
Device Drivers >> Character Devices >> [*]Enhanced Real Time Clock Support
```

```
Library Routines >> [*]CONFIG_CRC_CCITT
```

Първата опция с която задължително трябва да компилираме ядрото е така наречения „Балансиращ Демон на Заявките за Прекъсване“. Какво прави този демон – той разпространява прекъсвания върху процесора и процесорните ядра. Основната цел на демона

е да намира баланс между оптималната работа на процесора (и процесорните ядра) и най-икономичния режим на работа на процесора.

Следващата настройка е `Timer Frequency = 1000 Hz`, която представлява честотата на таймера, който изпраща прекъсванията към процесора. Тука по подразбиране честотата е `250 Hz`. Ние ще сменим тази честота на `1000 Hz`, поради факта, че за архитектура `i386` и ядро с версия по-голяма от `2.6` честотата на таймера в най-добрия случай достига до `1` милисекунда. В случай, че оставим таймера на `250 Hz` тогава неговата честота ще е от порядъка на `7 – 7,5` милисекунди.

`HPET Timer Support` е опцията, която включва „Високо Прецизния Таймер на Събития“ – което от своя страна представлява хардуерен таймер използван в персоналните компютри. Разработен съвместно от Интел<sup>43</sup> и Майкрософт<sup>44</sup> този таймер е внедрен в компютърните чипсети от `2005` година насам. `HPET` може да произвежда периодични прекъсвания на много по-висока резолюция от `RTC`<sup>45</sup> и често се използва за синхронизиране на мултимедийни потоци, осигуряващи плавно възпроизвеждане и намаляване на необходимостта от използване на други изчисления като `RDTSC`<sup>46</sup> инструкции на `x86`-базирани процесори.

След като обяснихме най-важните настройки, които трябва да включим в прекомпилацията на ядрото е ред на самата компилация. Следващите няколко команди ще компилират ядрото и ще създадат `.deb` пакет, който ще инсталираме и ще замени досегашното ни работещо ядро с новото, оптимизирано от нас ядро:

```
make-kpkg clean

fakeroot make-kpkg --initrd --append-to-version=-custom
kernel_image kernel_headers
```

Последната команда ще отнеме време, тъй като ще започне компилация на новото ядро. Следва инсталиране на вече готовия `.deb` пакет:

```
cd /usr/src

dpkg -i *.deb
```

---

<sup>43</sup> Интел (*Intel Corporation*) е базирана в град Санта Клара, щата Калифорния, САЩ мултинационална корпорация, известна най-вече с производството на микропроцесори и специализирани интегрални схеми.

<sup>44</sup> Корпорация „Майкрософт“ (на английски: *Microsoft Corporation*) е транснационална компания, развиваща дейност в областта на компютърните технологии и разработката на софтуер.

<sup>45</sup> `RTC` – Real Time Clock – Часовник в реално време - компютърен часовник (най-често под формата на интегрална схема), който следи текущото време.

<sup>46</sup> `Time Stamp Counter` е `64`-битов регистър на всички `x86` процесори след `Pentium`. Тя брои броя на циклите, докато се достигне до определено време когато се нулират.

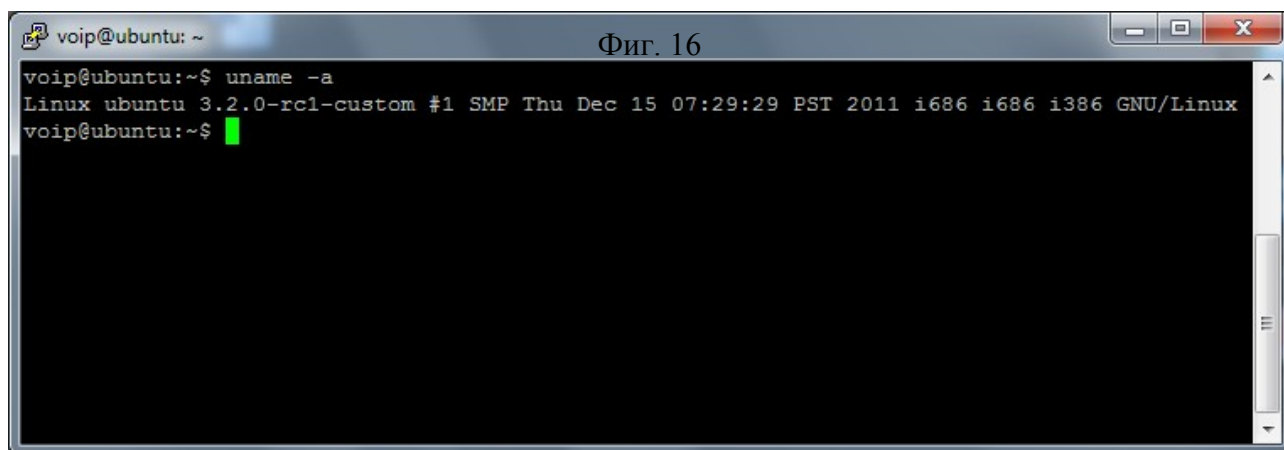
Последната стъпка е да рестартираме сървъра и да проверим дали операционната система ще зареди правилно и дали ще ни покаже някакви индикации за грешка. Ако всичко е наред, ще проверим версията на ядрото, което използваме:

```
reboot -t now!
```

На следващата фигура (Фиг. 16) виждаме, че операционната система е заредила и с помощта на командата:

```
uname -a
```

виждаме, че ядрото, което е заредено е нашето, прекомпилирано ядро.



Вече сме готови и имаме система напълно оптимизирана за най-добрите резултати с VoIP. Може да започнем подготвянето на машината за инсталирането на Asterisk. Изпълняваме следните команди, за да инсталираме допълнителни библиотеки за работата на Asterisk:

```
apt-get install build-essential libcurl3-dev libvorbis-dev  
libspeex-dev unixodbc unixodbc-dev libiksemel-dev
```

```
apt-get install flex xsltproc odbc-postgresql libusb-dev  
libnewt-dev libxml2-dev bison
```

```
apt-get install linux-headers-`uname -r` g++ libncurses5-dev  
libnewt-dev libusb-dev subversion git-core
```

```
apt-get install postgresql-8.1 postgresql-contrib-8.1  
postgresql-client-8.1 postgresql-dev
```

Нашата система е готова, сега трябва да изтеглим Asterisk от SVN<sup>47</sup> огледалото.

```
cd /usr/src
```

```
mkdir asterisk
```

```
cd asterisk
```

<sup>47</sup> Subversion (съкр. SVN) е програмен продукт за управление на софтуерните версии. Използва се главно за поддръжката на настоящи и минали версии на файлове за изходен код, веб страници и документация.

```
svn co http://svn.digium.com/svn/asterisk/trunk asterisk
svn co http://svn.digium.com/svn/asterisk-addons/trunk
asterisk-addons
svn co http://svn.digium.com/svn/dahdi/linux/trunk dahdi-
linux
svn co http://svn.digium.com/svn/dahdi/tools/trunk dahdi-
tools
svn co http://svn.digium.com/svn/libpri/branches/1.4 libpri
```

Сега да започнем с компилирането на DAHDI<sup>48</sup> модули за управление и контрол на Digium<sup>49</sup> устройства и други:

```
cd /usr/src/asterisk/dahdi-linux
make && make install
```

След като приключим с компилирането и инсталирането на DAHDI, следва да инсталираме и пакета с инструменти на DAHDI:

```
cd /usr/src/asterisk/dahdi-tools
./configure
make menuselect
make
make install
make config
```

Последната стъпка е да компилираме и инсталираме самият Asterisk:

```
cd /usr/src/asterisk/asterisk
./configure
make menuconfig
make
make install
make samples
make config
```

Ако нямаме съобщения за грешки и всичко е протекло нормално може да влезем в конзолния шел на Asterisk чрез следната команда:

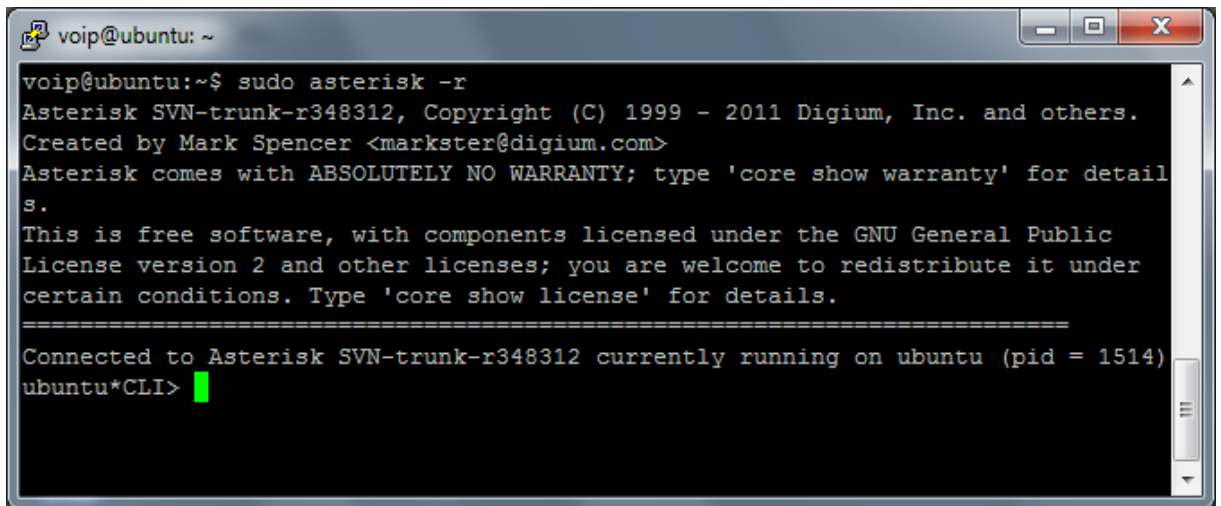
```
asterisk -r
```

Следва да ни се появи екрана от Фиг. 17:

---

<sup>48</sup> DAHDI (Digium / Asterisk Hardware Device Interface) е отворена технология източник, използван за контрол на Digium и други карти с телефония интерфейс.

<sup>49</sup> <http://www.digium.com/en/>



```
voip@ubuntu: ~  
voip@ubuntu:~$ sudo asterisk -r  
Asterisk SVN-trunk-r348312, Copyright (C) 1999 - 2011 Digium, Inc. and others.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.  
s.  
This is free software, with components licensed under the GNU General Public  
License version 2 and other licenses; you are welcome to redistribute it under  
certain conditions. Type 'core show license' for details.  
=====  
Connected to Asterisk SVN-trunk-r348312 currently running on ubuntu (pid = 1514)  
ubuntu*CLI>
```

Фиг. 17

Кратка равностетка до тук: имаме конфигурирана операционна система, подходяща за VoIP сървър, имаме още инсталиран VoIP софтуер Asterisk и модул за VoIP устройства – DANDI.

Можем да спрем до тук и да започнем да конфигурираме Asterisk по наш избор – да въвеждаме потребители, да създаваме гласови пощенски кутии и т.н., но ще направим още една малка оптимизация. Ще пуснем Asterisk да работи в режим на Реално Време (Realtime). Причината за това е базата данни, която ще използваме. Ще свържем Asterisk с PostgreSQL база данни. Това се прави с цел потребителите да не се пазят в паметта на компютъра. Те само се зареждат в базата, когато един потребител иска да осъществи обаждане и след това се изтрива от там. Целта е кеширане и оптимизация. Да, колкото и странно да звучи, оптимизационния метод чрез използване на PostgreSQL е по-бърз от колкото, ако използваме паметта.

Сега ще конфигурираме една база данни, в която Asterisk ще записва потребителите. За целта трябва да изпълним следните команди:

```
cd /usr/src/asterisk  
wget http://coto.debianchile.cl/files/realtime_pgsql.sql  
su - postgres  
createuser -s -D -R -l -P -e asterisk  
createdb -O asterisk -e asteriskDB  
pgsql -U asterisk -h localhost -d asteriskDB <  
/usr/src/asterisk/realtime_pgsql.sql
```

Следващата стъпка е да позволим на Asterisk да се свърже с базата данни:

```
vim /etc/asterisk/cdr_pgsql.conf
```

[global]

```
hostname=localhost
port=5432
dbname=asteriskDB
password=password
user=asterisk
table=cdr

    vim /etc/asterisk/extconfig.conf

[settings]
extensions => pgsql,asteriskDB,extensions_conf
sipuser => pgsql,asteriskDB,sip_conf
sippeers => pgsql,asteriskDB,sip_conf
sipregs => pgsql,asteriskDB,sip_conf
voicemail => pgsql,asteriskDB,voicemail_users
queues => pgsql,asteriskDB,queue_table
queue_members => pgsql,asteriskDB,queue_member_table

    vim /etc/asterisk/res_pgsql.conf

[general]
dbhost=127.0.0.1
dbport=5432
dbname=asteriskDB
dbuser=asterisk
dbpass=password
requirements=warn
```

След всички тези промени, единственото нещо, което остана е да рестартираме Asterisk и сме готови с добавянето на базата от данни:

```
/etc/init.d/asterisk restart
```

#### 2.4.4 Конфигуриране на Asterisk

След всички стъпки горе, вече може да започнем конфигурацията на VoIP сървъра. Горните настройки бяха ориентирани предимно на работата му, докато в тази точка ще разгледаме как се създават потребители, как се настройва централата за оставяне на гласови съобщения, как се избират външни номера и т.н. Всички тези точки ще бъдат описани, и ще имат примери към тях, но като цяло пълната информация за всички възможни настройки, може да се намери в Интернет и в официалните ръководства на софтуера. Тука ще разгледаме само основните настройки на централата.

Ще започнем с конфигурационните файлове. Те се намират в директорията: /etc/asterisk/. Ето как изглежда командата (Фиг. 18):

```
ls /etc/asterisk
```

```
root@ubuntu: /home/voip
root@ubuntu:/home/voip# ls /etc/asterisk/
adsi.conf          codecs.conf        muted.conf
agents.conf        confbridge.conf   osp.conf
ais.conf           console.conf      oss.conf
alarmreceiver.conf dbsep.conf        phone.conf
alsa.conf          dnsmgr.conf       phoneprov.conf
amd.conf           dsp.conf          queuerules.conf
app_mysql.conf     dundi.conf        queues.conf
asterisk.adsi      enum.conf         res_config_mysql.conf
asterisk.conf      extconfig.conf    res_config_sqlite3.conf
calendar.conf      extensions.ael     res_config_sqlite.conf
ccss.conf          extensions.conf   res_curl.conf
cdr_adaptive_odbc.conf extensions.lua     res_fax.conf
cdr.conf           extensions_minivm.conf res_ldap.conf
cdr_custom.conf    features.conf     res_odbc.conf
cdr_manager.conf   festival.conf    res_pgsql.conf
cdr_mysql.conf     followme.conf    res_pktccops.conf
cdr_odbc.conf      func_odbc.conf   res_snmp.conf
cdr_pgsql.conf     gtalk.conf       res_stun_monitor.conf
cdr_sqlite3_custom.conf h323.conf        rpt.conf
cdr_syslog.conf    http.conf        rtp.conf
cdr_tds.conf       iax.conf         say.conf
cel.conf           iaxprov.conf     sip.conf
cel_custom.conf    indications.conf sip_notify.conf
cel_odbc.conf      jabber.conf      skinny.conf
cel_pgsql.conf     jingle.conf     sla.conf
cel_sqlite3_custom.conf logger.conf      smdi.conf
cel_tds.conf       manager.conf     telcordia-1.adsi
chan_dahdi.conf    meetme.conf      udptl.conf
chan_mobile.conf   mgcp.conf        unistim.conf
chan_oh323.conf    minivm.conf      usbradio.conf
cli_aliases.conf   misdncnf         users.conf
cli.conf           modules.conf     voicemail.conf
cli_permissions.conf musiconhold.conf  vpb.conf
root@ubuntu:/home/voip#
```

Фиг. 18

Както добре се вижда, централата позволява доста настройка и оптимизация. Всъщност Asterisk притежава 102 конфигурационни файла (Фиг. 19):

```
root@ubuntu: /home/voip
root@ubuntu:/home/voip# ls -la /etc/asterisk/ | wc -l
102
root@ubuntu:/home/voip#
```

Фиг. 19

Нека да хвърлим поглед върху някой от тях. Първият файл, който е един от най-важните е:

*extensions.conf* - в този конфигурационен файл се описват всички така наречени планове за звънене. Какво представлява един план за звънене – това всъщност е описание как Asterisk ще провежда входящите и изходящите обаждания, на къде ще ги насочва, как ще ги транслира. Също така този файл съдържа разширяващите номера в централата. Един план съдържа секции наречени контекст. Всеки контекст се състои от повече от едно разширение. Какво е разширение? Разширението е телефонен номер, който може да бъде числа, букви или и двете. Всяко разширение има приоритет и приложение. С помощта на контекстите ние

може да организираме нашите планове за звънене. В общи линии един план изглежда по следния начин:

```
[general]
```

→ Тука се описват основните настройки

```
[globals]
```

→ тука се описват дефинициите на глобалните променливи

```
[context1]
```

→ разширение 1, приоритет 1, приложение

→ разширение 1, приоритет 2, приложение

...

```
[context2]
```

→ разширение 9999, приоритет 1, приложение

→ разширение 9999, приоритет 2, приложение

**[general]** – първият контекст във файла `extensions.conf` е `[general]`. В него могат да се зададат 3 настройки:

`static = yes | no` – ако изберем опцията `yes` и заедно с това сме задали на втората опция `writeprotect = no`, може от командния ред на Asterisk да запазим нашия план: `save dialplan`

`writeprotect = yes | no` – тази опция се използва, ако искаме да имаме възможността да записваме нашия план от командния ред на програмата.

`autofallthrough = yes | no` – ако тази опция е настроена на `yes`, след приключването на нещата който извършва програмата, Asterisk ще затвори връзката с отсрещната страна. Ако е настроена на `no`, Asterisk ще чака за изпълнението на друго разширение. Силно се препоръчва тази опция да е настроена на `yes`.

**[globals]** – в контекста `[globals]` може да дефинираме наши, собствени променливи, които могат да бъдат използвани по-късно в други разширения. Променливите дефинирани в тази секция не са чувствителни към регистъра на буквите, който ще рече, че  `${MYVAR}` и  `${MyVAR}` са едно и също нещо. Начина на дефиниране на променливи става по следната схема:

```
име_на_променливата => стойност_на_променливата
```

Пример:

```
[general]
static=yes
writeprotect=no
```

```
[globals]
MyMusicOnHold => /mp3/Mozart.mp3
```

**[context1]** – с изключение на [general] и [globals] всичко друго в този файл се приема за контекст. Ето как изглежда един контекст:

```
[context_name]
exten => some_exten_number,priority,application(arg1,arg2,..)
exten => some_exten_number,priority,application,arg1|arg2...
exten => some_pattern,priority,application(arg1,arg2,...)
```

Но каква е ползата от контекстите? Отговора е че вътре в контекстите, ние може да създадем наше собствено Интегрирано Гласово Меню, използвайки разширенията. Също така може да дефинираме специфичен контекст за всеки отдел на Университета (счетоводство, поддръжка, ректорат и т.н.).

Използвайки контекстите, Счетоводството например, може да се свържи с техния главен счетоводител избирайки 123, а отдела по поддръжка с техния главен администратор отново избирайки 123.

Ако се питате, как от поддръжката ще могат да набират главния счетоводител – отново чрез контекстите, но този път поддръжката няма да набере 123, а ще избере 9 и след това 123, като се укаже, че ако някой извън контекста на счетоводството иска да се свържи с тях, трябва да набере 9 и след това желания номер.

Използвайки контексти може да се позволи на поддръжката да осъществяват изходящи обаждания, докато счетоводството, да бъдат ограничени от такива.

С помощта на контекстите е много лесно да се организира и ръководи всички регистрирани в централата телефони.

Пример:

```
[general]
; ...skip...
```

```
[globals]
; ...skip...
```

```
[Poddryjka]
exten01 => 600206,1,Dial(SIP/Petyr)
```

```
[Schetovodstvo]
exten01 => 600307,1,Dial(IAX2/Glaven_Schetovoditel)
```

Следващия конфигурационен файл който ще разгледаме е: `adsi.conf`

`adsi.conf` – `asdi` е съкращение от `Analog Display Services Interface` – или на български – Услуга за Аналоговите Екрани.

ASDI е протокол който позволява алтернативни гласови услуги и услуги за пренос на данни през аналогови телефонни мрежи, като например изпращане на данни до дисплея на аналогови телефони.

Други приложения на ASDI включват:

- Визуална гласова поща: дисплея на телефона показва меню с опциите за гласова поща както съобщенията;
- „Визуална“ директория: услуга, която позволява да намерите телефонния номер на физическо лице или бизнес контакт;
- Четене на електронна поща: позволява ви да изпращате и получавате електронни писма през ASDI устройство;

Един от недостатъците на тази опция на централата е че за да използваме тези услуги, трябва да имаме подходящ ASDI телефона, докато телефоните който избрахме за осъществяването на връзката между Първи и Втори корпус не са подходящи. Ето защо няма да продължавам с разглеждането на `adsi.conf` файла, просто го споменах за да се има предвид ако за в бъдеще бъдат закупени такива телефонни апарати.

Ще продължим с файла `agents.conf`.

`agents.conf` – най-общото приложение чрез заради което може да се използва този файл са опашките. Ето защо този файл предимно се използва заедно с файла `queues.conf`.

В `agents.conf` вие можете да създадете така наречените агенти. Това са потребители, който отговарят на входящите повиквания в потребителската опашка.

Агентите могат да бъдат разделени на отделни групи наречени контексти. По този начин имаме голяма гъвкавост при управляването на опашките, защото може да изберем кой агент с коя опашка да работи. Ще разгледаме един пример на агент.

Има два типа контексти. Първият е `[general]` контекст. Другия е `[agents]` контекст. Имената на контекстите в този файл не могат да бъдат променени. Ние дефинираме началото на контекста като започнем да пишем: `[general]`. Един контекст завършва когато започва нов контекст, а последния завършва с края на файла.

`[general]` – за този контекст са валидни следните опции:

`persistentagents = yes | no` – тази опция определя дали агентите трябва да се запазват в базата данни на Asterisk или не. Ако тази опция е включена, тогава агентите ще

могат да се запазват в базата от данни. Ако не е включена тогава агентите ще трябва да се презареждат когато се презарежда Astersk. По подразбиране тази опция е винаги `yes`.

`[agents]` – за агентите като контекст са валидни следните опции:

`autologoff` – с тази опция може да зададем колко време телефона ще звъни без да получи отговор. Вие трябва да зададете максималния период в секунди. По подразбиране тази опция е настроена на 15 секунди.

`askall` – ако тази опция е настроена на `yes`, агента се включва с едно приложение, което се казва `AgentCallbackLogin` след което потребителя трябва да потвърди включването на агента като натисне бутона „#“. По подразбиране тази опция е настроена на `no`.

`wrarturtime` – тази опция се отнася за времето след като разговора е приключил. С тази опция се задава минималния период на време, който трябва да мине преди да може да получите ново обаждане. Времето се задава в милисекунди и по подразбиране е настроено на 5000.

`musiconhold` – с тази опция можете да зададете мелодия по подразбиране на класа от агентите. Класовете на агентите който включват тази опция се намират във файла `musiconhold.conf`. В него файл се определят класовете, а в `agents.conf` само се записват имената им.

`updatecdr` – с тази опция определяме дали искаме да сменим канала на обаждания в един CDR запис или не. Така, че да знаем кой агент генерира обаждането. По подразбиране тази опция е изключена.

`group` – тази опция ви позволява да групирате няколко агента с цел по-лесно управление. По подразбиране не е зададена група. Начина на създаване на група е следния:

```
група = номер_на_групата
```

Пример:

```
group = 1
```

За да прибавим агент към групата трябва след номера на групата да изброим агентите, който искаме да са в тази група:

Пример:

```
group = 1
```

```
agent => 8822, 1122, user1
```

`recordagentcalls` – тази опция дава възможност за запис на разговорите. По подразбиране опцията е изключена.

`recordformat` – тази опция се отнася за формата на аудио файла. Файла в който ще бъде записан разговора. Позволените формати са: `.wav`, `.gsm`, и `.wav49`. По подразбиране тази опция е настроена на `.wav`.

`createlink` – тази опция добавя текст към името на записа. Който запис после може да бъде зареден от обикновен интернет адрес. Например:  
`urlprefix=http://localhost/calls/`

`savecallsin` – благодарение на тази опция може да променим директорията в която да се запазват записите. В противен случай, централата ще използва настройката по подразбиране която е директорията: `/var/spool/asterisk/monitor`.

`custom_beep` – с тази опция може да изберем наш тон който ще се използва когато някой агент се свърже с централата.

Нека да създадем един агент. Ето и формата по който се създава агент:

```
agent => agentnumber, agentpassword, name
```

`agentnumber` – е номера на агента;

`agentpassword` – е паролата на агента;

`name` – истинското име на агента. То може да бъде каквото ние пожелаем.

Пример:

```
[general]
```

```
persistantagents = yes
```

```
[agents]
```

```
autologoff = 15
```

```
ackcall = no
```

```
wrapuptime = 5000
```

```
musiconhold = default
```

```
recordagentcall = yes
```

```
recordformat = gsm
```

```
group1
```

```
agent => 101, 101, user1
```

```
agent => 102, 102, user2
agent => 103, 103, user3
agent => 104, 104, user4

group2
agent => 8887, 8887, operator
agent => 8899, 8899, ivan

group3
agent => 7771, 7771, petko
```

Нека сега да разгледаме гласовата поща и възможността, която ни предлага Asterisk. Настройките за гласова поща се намират във файла `voicemail.conf`, който също се намира както всички останали конфигурационни файлове в директорията: `/etc/asterisk`.

`voicemail.conf` – може да декларираме пощенска кутия в контекста `[default]` или да създадем наши контексти, в който да декларираме пощенските кутии. Важно е да знаем, че контекстите за гласовите пощи и тези контексти във файла `extensions.conf` нямат нищо общо помежду си.

Ето каква е схемата за създаване на пощенска кутия:

```
mailbox_number => password, name, email
```

`mailbox_number` – е номера, който сме използвали в `extensions.conf` за командата `VoiceMail()` и също така е същия номер, който сме описали във файла `sip.conf` или `iax.conf`;

`password` – е паролата която сме използвали във файла `sip.conf` или `iax.conf`;

`name` – това е името с което се асоциира пощенската кутия;

`email` – е електронната поща, на която ще се получават гласовите съобщения.

Пример (`voicemail.conf`):

```
[mailbox_petyr]

777 => 1234, petyr, petyr@ue-varna.bg
```

В горния пример създадохме контекст `[mailbox_petyr]` в който създадохме пощенска кутия на номер `777` с парола `1234`, притежание на `petyr` с електронна поща: [petyr@ue-varna.bg](mailto:petyr@ue-varna.bg)

Когато някой избира petyr, и той не отговори, отсрещната страна може да остави съобщение на petyr.

Пример (extensions.conf):

```
exten2petyr => 1234,1,Dial(SIP/petyr, 30)
exten2petyr => 1234,2,VoiceMail(777@mailbox_petyr)
exten2petyr => 1234,3,PlayBack(vm-goodbye)
exten2petyr => 1234,4,HangUp()
```

Този пример ще се опита да набере SIP потребител Petyr с номер 1234 за 30 секунди и след това ако никой не вдигне ще се изпълни следващото разширение, т.е. Asterisk ще отвори пощенска кутия с номер 777, която се намира в контекста mailbox\_petyr и ще започне записването на съобщението. След приключване на записа ще се стартира автоматичния отговор за край на записването на съобщението и накрая централата ще приключи разговора като го прекъсне. Автоматичния отговор vm-goodbye се намира като звуков файл в директорията: /var/lib/asterisk/sounds/.

Записаните гласови съобщения се съхраняват в директория: /var/spool/asterisk/voicemail/<context>/<mailbox>/INBOX, и по-специално за нашия пример това ще е директорията: /var/spool/asterisk/mailbox\_petyr/777/INBOX/.

За да преслушаме нашите съобщения трябва да изпълним командата VoiceMailMain (в случай че искаме да ги слушаме на сървъра).

В случай, че използваме хардуерни телефони (точно какво ние решихме, че ще използваме и в двата корпуса) ние можем директно да прослушаме гласовите ни съобщения от телефона. Това се дължи на факта, че хардуерните телефони поддържат сесийния инициализиращ протокол (SIP), но задължително трябва да бъде активирана тази опция в sip.conf файла. Когато получим гласово съобщение на такъв телефон, индикатора за „Чакащи съобщения“ веднага ще ни извести за пристигнало такова съобщение.

Накрая ще разгледаме конфигурационния файл, който съдържа потребителите, заедно с настройките отнасящи се за тях за изходящи и входящи повиквания. Конфигурационните файлове за потребителите всъщност са два: iax.conf и sip.conf. В зависимост от това с какви устройства разполагаме решаваме кой от двата (или и двата) файла да настроим за нашите нужди. Тъй като в нашия план хардуерните телефони не поддържат iax стандарта, но за сметка на това поддържат SIP протокола, ще използваме файла sip.conf.

`sip.conf` – този файл съдържа параметри, свързани с конфигурацията на SIP потребителите и тяхната връзка с телефонната централа. Клиентите трябва да бъдат конфигурирани в този файл, преди те да имат възможността да използват Asterisk като сървър.

След този контекст, всички останали определят клиентските параметри. Параметри като потребителско име, парола, IP адрес по подразбиране (на клиента), IP адрес по подразбиране на клиенти, който не са регистрирани в централата и т.н. И в този файл контекстите са разделени чрез имена оградени в квадратни скоби, като всеки контекст описва един потребител. Както вече споменахме първия контекст се нарича `[general]` и той не може да се използва за потребителско име. `[general]` приема три атрибута:

`port` – това е порта на който Asterisk трябва да слуша за идващи SIP връзки. По подразбиране порта е с номер 5060. Ако все пак решим да сменим порта, трябва да се съобразим с това, че порта който трябва да използва Asterisk не трябва да бъде зает от други услуги.

`bindaddr` – това е IP адреса, на който централата трябва да слуша за идващи SIP връзки. Ако машината разполага с много публични IP адреса, както и виртуални IP адреса, с тази опция може да се определи на кой от всичките трябва работи централата. По подразбиране, ако тази опция не бъде изрично използвана, централата ще използва всички налични и активни мрежови устройства на който е зададен реален или виртуален IP адрес.

`context` – това е приветстващо съобщение, когато SIP потребител бъде свързан с централата. Това съобщение може изобщо да не бъде видяно от потребители поради различните модели софтуерни и хардуерни телефони.

Това са параметрите, който се настройват в контекста `[general]`. От него надолу всички останали контексти са отделни потребители на който също подлежат на аранжиране и настройка. Клиентските параметри са няколко. Ще разгледаме основните:

`type` – тази опция определя типа на класа за клиента. Тука класовете биват три вида:

- ❖ `peer` – устройство, което получава телефонни обаждания от централата;
- ❖ `user` – устройство, което провежда телефонни обаждания през централата;

❖ friend – устройство, което получава и провежда телефонни обаждания през централата

secret – задава паролата за клиента. Паролата може да съдържа както букви, така и цифри.

host – на тази опция се задава IP адреса на потребителя или името на устройството. Също така тази опция позволява да бъде настроена на 'dynamic', което означава, че потребителя може да се свържи с централата от всякакъв IP адрес. Тази опция най-много намира приложение, когато потребителите ни получават IP адрес от DHCP<sup>50</sup> сървър.

username – тази опция задава потребителското име с което Asterisk ще се опита да свържи пристигнало обаждане. Използва се предимно когато поради някаква причина стойността не е същата както стойността с която регистриран потребителя.

Сега ще регистрираме потребител в централата използвайки тези опции:

Пример:

```
[general]
port=5060
bindaddr=192.168.0.10
context=default
[petyr]
type=friend
secret=snom100
host=dynamic
defaultip=192.168.0.15
```

```
[Account_Manager]
type=friend
secret=mysecret
host=192.168.0.20
canreinvite=no
context=trusted
```

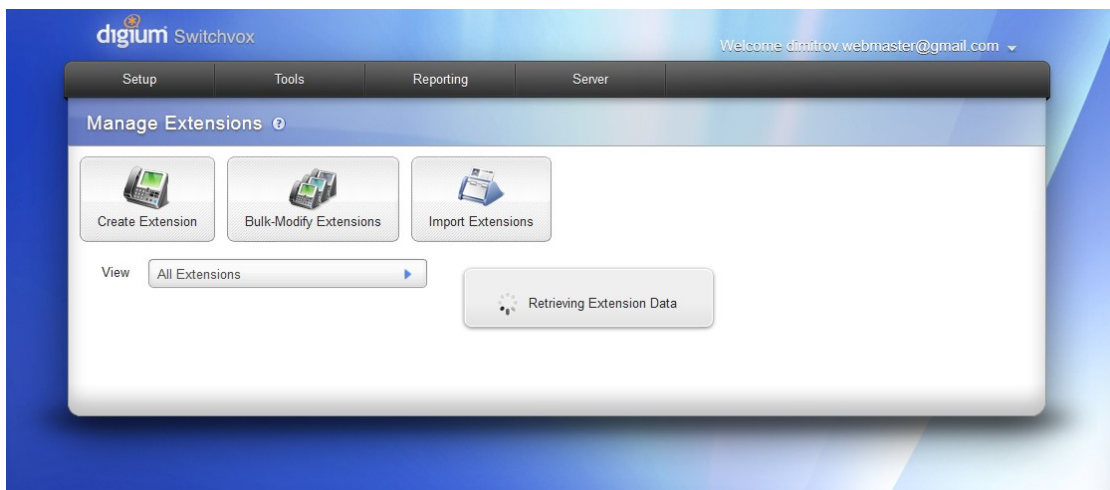
От тука на сетне единственото, което остава е да се въведат потребителите в централата.

---

<sup>50</sup> Dynamic Host Configuration Protocol (DHCP) -- Протокол за динамично конфигуриране на хостове - комуникационен протокол чрез който компютър, тип компютърно устройство, маршрутизатор или всякакъв друг вид устройство използващо IP адрес могат да заявят за Интернет адрес от сървър, който от своя страна притежава определено пространство от IP адреси за раздаване.

## 2.5 Администриране на Asterisk

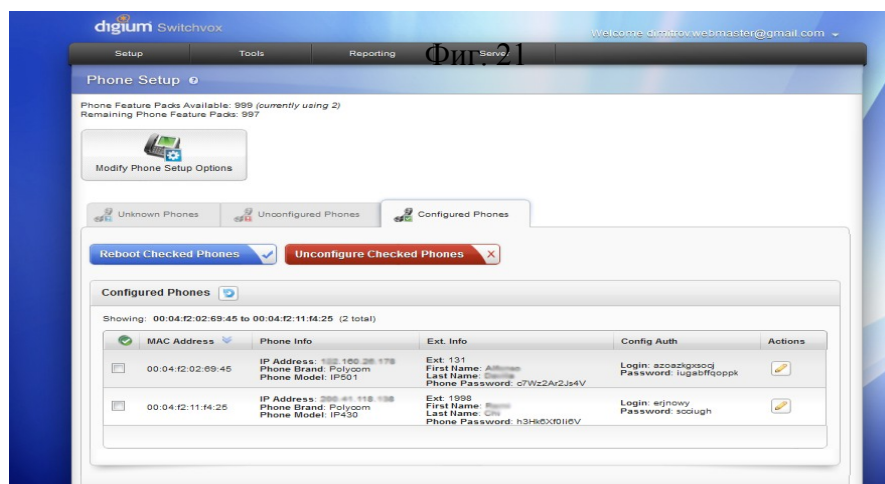
Asterisk позволява бързо и лесно администриране през удобен и лесен за използване web мениджър. Какво представлява този мениджър – той всъщност представлява приставка, която се инсталира отделно от основната инсталация на централата и тази приставка използва web страница за вход в централата. Веднъж влезли, вече можем да управляваме централата както пожелаем (стига да имаме достатъчно привилегии за изпълнение на желание от нас действия). На Фиг. 20 можете да видите как изглежда първоначалния екран след като въведем потребителско име и парола:



Фиг. 20

След това може да изберем какво действие да предприемем. Може да създаваме нови потребители, или да променяме вече създадени потребители. На следващата фигура, може да видим два въведени потребителя, с техните настройки, телефони, IP адреси и пароли за вход в централата (Фиг. 21).

фигури, са  
действаща  
центра  
като в  
виртуална  
нямах



Горните две  
взети от  
телефонна  
Asterisk, тъй  
процес на  
реализация

техническата възможност да закача IP телефон, за да мога да покажа как централата ще разпознае устройството.

## Заклучение

Главната цел на разработката беше да бъде изградена интернет телефонна свързаност между Първи корпус на Икономически Университет Варна и Втори корпус при същия университет. За изграждането на тази свързаност трябваше първо да бъде изградена виртуална частна мрежа, която да осигури локален достъп на всички потребители (и от двата корпуса) локално използване на ресурсите на университета (интернет споделяне, сървъри за комуникация на телефонните обаждания и т.н.). В дипломната работа беше разгледано подробно причините за една такава свързаност. Икономическата ползва от използването на интернет за пренасяне на глас е един от крайъгълните камъни, на дипломната работа.

Основен стимул за имплементиране на VoIP е възможността за понижаване на разходите. Финансовите отдели на компаниите все по-често поставят под въпрос нуждата от съществуването на две отделни устройства за гласови и мрежови комуникации, две отделни преносни среди и два различни отдела за тяхната поддръжка. В дипломната работа тези две устройства и среди бяха слети в едно като беше показано, че няма загуба на функционалност, а вместо това се спестиха разходи от допълнително хардуерно осигуряване.

Една от стъпките при внедряване на IP телефонна система е конфигурирането на мрежовите устройства за предаване на гласови данни и необходимите механизми, осигуряващи качество на услугите. В дипломната работа е разгледана примерна съществуваща мрежова инфраструктура и са представени конфигурациите на няколко типични схеми за междуофисна комуникация.

Подробно е описано и показано начина на свързаност между Първи и Втори корпус посредством софтуера OpenVPN, също така е показана и примерна конфигурация на виртуалния сървър.

След изграждането на виртуалната мрежа дипломната работа разглежда вече основната си цел и това е VoIP сървърите за управление на обаждания Asterisk. Отново са показани примери, които показват как телефонната централа работи и как може да бъде конфигурирана в зависимост от нуждите на потребителите и на университета.

Но както вече споменах, най-важен си остава факта, че с реализирането на един такъв проект и отделянето на средства за изграждането на телефонна свързаност бъдещите такси на университета относно телефонните обаждания значително ще паднат като цена. В дипломната работа никъде не се спомена свързаност между сегашната телефонна система и евентуално новоизградената такава и за това си има причина. Причината е допълнителното оскъпяване от страна на хардуера, чрез който Asterisk централата трябва да бъде включена

към досегашната PSTN централа. Освен това друга причина е постепенното преминаване на много фирми към IP телефонията.

Бъдещето не е в PSTN мрежите, а в Интернет. Така както фирмите масово вече използват Интернет за извършване на реклама, маркетинг, продажби, разплащане на сметки, така и комуникациите вече са насочени изцяло към Интернет. Ето защо социолози предвиждат премахването на PSTN мрежите (не физическо премахване, а спиране на тяхното използване) и преминаване на към VoIP.

Смятам че разработката беше успешна и че успях да постигна предварително поставените цели. Опитът ми като системен администратор с 3 годишен стаж (до написването на дипломната работа) ми позволи да реализирам този проект като единствената разлика от реалната ситуация е че използвах виртуална среда за изграждането на проекта. Но след като резултатите бяха отлични и дори и средата да беше виртуална клиентските машини от който се тестваше проекта си бяха напълно реални, това показва че проекта може да бъде реализиран и на реална машина.

Поддръжката на една такава система може спокойно да бъде извършвана от отдела, който поддържа и мрежовата инфраструктура на Икономическия Университет. Системните администратори на университета биха се справили с централата и без допълнителни курсове относно VoIP. Още повече че централата разполага с много допълнителни приставки, който не станаха обект на разглежданата дипломна работа, но тези приставки биха улеснили работата по централата – конфигурирането и администрирането ѝ.

Сигурността на телефонната свързаност между двата корпуса беше на второ място по важност в дипломната работа. На няколко места се изтъкнаха причини поради, които някой от настройките не бяха включване и също така именно заради сигурността и конфиденциалността на разговорите между отделите се взе решение логическата свързаност на двете отделни мрежи между първи и втори корпус да бъде посредством виртуална частна мрежа.

## Използвана литература







### Интернет адреси:

<http://bg.wikipedia.org/wiki/DHCP>  
[http://bg.wikipedia.org/wiki/Domain\\_Name\\_System](http://bg.wikipedia.org/wiki/Domain_Name_System)  
[http://bg.wikipedia.org/wiki/Network\\_address\\_translation](http://bg.wikipedia.org/wiki/Network_address_translation)  
<http://bg.wikipedia.org/wiki/RTP>  
<http://bg.wikipedia.org/wiki/SIP>  
[http://bg.wikipedia.org/wiki/Subversion\\_\(софтуер\)](http://bg.wikipedia.org/wiki/Subversion_(софтуер))  
<http://bg.wikipedia.org/wiki/TLS>  
<http://bg.wikipedia.org/wiki/WAN>  
<http://bg.wikipedia.org/wiki/Wi-Fi>  
<http://bg.wikipedia.org/wiki/Етернет>  
<http://bg.wikipedia.org/wiki/Интел>  
[http://bg.wikipedia.org/wiki/Интернет\\_телефония](http://bg.wikipedia.org/wiki/Интернет_телефония)  
<http://bg.wikipedia.org/wiki/Майкрософт>  
[http://bg.wikipedia.org/wiki/Мрежови\\_топологии](http://bg.wikipedia.org/wiki/Мрежови_топологии)  
[http://bg.wikipedia.org/wiki/Операционна\\_система](http://bg.wikipedia.org/wiki/Операционна_система)  
[http://bg.wikipedia.org/wiki/Протокол\\_за\\_трансфер\\_на\\_файлове](http://bg.wikipedia.org/wiki/Протокол_за_трансфер_на_файлове)  
[http://cio.bg/1044\\_sigurnost\\_voip\\_resheniya.2](http://cio.bg/1044_sigurnost_voip_resheniya.2)  
<http://delian.blogspot.com/2007/03/voip.html>  
<http://en.wikipedia.org/wiki/H.323>  
[http://en.wikipedia.org/wiki/Integrated\\_Services\\_Digital\\_Network](http://en.wikipedia.org/wiki/Integrated_Services_Digital_Network)  
<http://en.wikipedia.org/wiki/ITU-T>  
[http://en.wikipedia.org/wiki/Private\\_branch\\_exchange](http://en.wikipedia.org/wiki/Private_branch_exchange)  
<http://en.wikipedia.org/wiki/RAS>  
<http://en.wikipedia.org/wiki/RDTSC>  
[http://en.wikipedia.org/wiki/Real-time\\_clock](http://en.wikipedia.org/wiki/Real-time_clock)  
[http://en.wikipedia.org/wiki/Resource\\_reservation\\_protocol](http://en.wikipedia.org/wiki/Resource_reservation_protocol)  
[http://en.wikipedia.org/wiki/Signaling\\_System\\_7](http://en.wikipedia.org/wiki/Signaling_System_7)  
[http://en.wikipedia.org/wiki/Type\\_of\\_service](http://en.wikipedia.org/wiki/Type_of_service)  
[http://en.wikipedia.org/wiki/Voice\\_over\\_IP](http://en.wikipedia.org/wiki/Voice_over_IP)  
[http://en.wikipedia.org/wiki/VoIP\\_VPN](http://en.wikipedia.org/wiki/VoIP_VPN)  
<http://iptel.bg/voip.php>  
<http://kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/tuntap.txt>  
<http://nedelchev.net/Asterisk-VOIP>  
<http://orlovmost.net/ip-telephone.html>  
[http://plazmaxcomputers.com/product\\_info.php?manufacturers\\_id=14&products\\_id=1251](http://plazmaxcomputers.com/product_info.php?manufacturers_id=14&products_id=1251)  
<http://research.uni-sofia.bg/bitstream/10506/206/1/DiplomnaRabotaKBuykliev.pdf>  
<http://voip.start.bg>  
<http://www.asterisk.org/dahdi>  
<http://www.bertolinux.com/voip/english/VoIP-HOWTO-3.html>  
[http://www.cisco.com/en/US/docs/ios/solutions\\_docs/voip\\_solutions/CAC.html](http://www.cisco.com/en/US/docs/ios/solutions_docs/voip_solutions/CAC.html)  
<http://www.howtoforge.com/installing-and-configuring-asterisk-1.6-and-postgresql-to-manage-cdr-and-realtime-config-on-debian>  
<http://www.stemo.bg/jsp/bg.do?a=cHJvZHVjdC0tLS0wMDA2MzAxMDQ0OTE4>  
<http://www.stemo.bg/jsp/bg.do?a=cHJvZHVjdC1DSVNDTy0wMDA2MzAwMTY0OTIyLTAwMDYzMDEwNDQ1ODAtMDAwNjMwMTA0NDkyOA==#n.32.1.5.3>  
<http://www.voip-info.org/wiki/view/Asterisk>

<http://www.voip-info.org/wiki/view/STUN>

## Приложение 1

Легенда на Фиг. 5, 6, 7 и 8:

	Cisco мениджър на повикванията/обажданията
	VoIP Gateway – VoIP шлюз
 IP	IP телефон
 PBX	Телефонна централа
	Мрежов комутатор
	Обикновен телефон