

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ КОМП'ЮТЕРНОЇ СИСТЕМИ
IP-ТЕЛЕФОНІЇ НА ОСНОВІ ASTERISK

Виконав: студент 2 курсу, групи КНТ-513м
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерні системи та мережі»

КАЗАНСЬКИЙ О.А.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С. Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЄФИМЕНКО М.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти магістерський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) «Комп'ютерна системи та мережі»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Зав. кафедри Кудерметов Р.К.
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

КАЗАНСЬКИЙ Олексій Аркадійович
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження ефективності комп'ютерної системи
IP-телефонії на основі Asterisk
керівник проєкту (роботи) к. т. н., доцент, СКРУПСЬКИЙ Степан Юрійович
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від “18” жовтня 2024 року № 149

2. Строк подання студентом проєкту (роботи) 10 грудня 2024 року
3. Вихідні дані до проєкту (роботи) існуючі методи та алгоритми підвищення
продуктивності комп'ютерної системи IP-телефонії на основі Asterisk

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1) Аналіз предметної області;
2) Встановлення та налаштування Asterisk;
3) Дослідження розробленої системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	СКРУПСЬКИЙ С.Ю., доцент		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання 01.10.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз наявних методів та алгоритмів розробки ІР-телефонії, перегляд отриманих результатів, виділення недоліків та переваг	18.10.2024 р.	
2	Встановлення та налаштування програмного забезпечення	20.10.2024 р.	
3	Аналіз технології Asterisk	26.10.2024 р.	
4	Розробка програмних компонентів системи	12.11.2024 р.	
5	Дослідження розробленої системи	24.11.2023 р.	
6	Оформлення отриманих результатів у ПЗ	01.12.2024 р.	
7	Оформлення графічного матеріалу	02.12.2024 р.	
8	Оформлення допоміжного матеріалу	08.12.2024 р.	

Студент (ка) Олексій КАЗАНСЬКИЙ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи) Степан СКРУПСЬКИЙ
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 61 с., 48 рис., 1 формула, 1 табл. 15 джерел.

ASTERSIK, VOIP, IP-ТЕЛЕФОНІЯ, LINUX, UBUNTU, SIP

Метою магістерської роботи є дослідження ефективності комп'ютерної системи IP-телефонії на основі Asterisk, для забезпечення оптимальної продуктивності в умовах різної кількості одночасних дзвінків та використання різних кодеків.

Предмет дослідження – аналіз конфігурації та продуктивності системи Asterisk з точки зору впливу різних кодеків на навантаження системи при обробці одночасних дзвінків.

Об'єкт дослідження – комп'ютерна система телефонії на базі програмного забезпечення Asterisk та її функціонування в умовах IP-телефонії.

Перший розділ магістерської роботи був присвячений ознайомленню з предметною областю, визначенню необхідного функціоналу комп'ютерної системи, а також аналізу переваг Asterisk у порівнянні з його аналогами.

Другий розділ розглядає питання встановлення та налаштування основних конфігурацій Asterisk.

Третій розділ присвячений тестуванню працездатності системи та порівнянню отриманих результатів.

У результаті виконання дипломної роботи було проведено детальний аналіз, проектування та дослідження розробленої комп'ютерної системи.

ABSTRACT

PP: 61 p., 48 figures, 1 formula, 1 table. 15 sources.

ASTERISK, VOIP, IP-TELEPHONY, LINUX, UBUNTU, SIP

The purpose of the master's thesis is to study the effectiveness of a computer-based IP telephony system based on Asterisk to ensure optimal performance in conditions of different numbers of simultaneous calls and the use of different codecs.

The subject of the study is to analyze the configuration and performance of the Asterisk system in terms of the impact of different codecs on the system load when processing simultaneous calls.

The object of the study is a computer telephony system based on Asterisk software and its operation in IP telephony.

The first chapter of the master's thesis was devoted to familiarization with the subject area, determination of the required functionality of the computer system, and analysis of the advantages of Asterisk in comparison with its analogues.

The second section deals with the installation and configuration of the basic Asterisk configurations.

The third chapter is devoted to testing the system's performance and comparing the results.

As a result of the thesis, a detailed analysis, design and research of the developed computer system was carried out.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Історія IP-телефонії.....	9
1.2 Принципи пакетної передачі мовлення	9
1.3 Шлюзи IP-телефонії.....	14
1.4 Способи підключення до IP-телефонії.....	15
1.5 Протокол SIP в IP-телефонії	16
1.6 Особливості програмного пакета Asterisk для створення IP-телефонів.....	17
1.6.1 Переваги IP-телефонії на базі Asterisk	18
1.6.2 Обґрунтування вибору Asterisk як АТС.....	18
1.6.3 Аналіз прив'язки до певного серверного обладнання	19
1.6.4 Аналіз системи ліцензування системи Asterisk	20
1.6.5 Аналіз методу інтеграції з CRM-системами та іншими АТС.....	20
1.6.6 Обслуговування та адмініструючий персонал	21
1.6.7 Цілі та завдання програмної АТС на базі Asterisk.....	21
1.7 Висновки за розділом.....	22
2 Встановлення та налаштування Asterisk.....	23
2.1 План встановлення та налаштування Asterisk.....	23
2.1.1 Встановлення необхідних залежностей	23
2.2 Встановлення Asterisk.....	26
2.3 Створення Linux користувача та запуск Asterisk.....	32
2.4 Структура Asterisk.....	33
2.5 Підключення користувачів.....	35
2.6 Створення SIP користувача.....	36
2.7 Створення та застосування шаблону.....	38
2.8 Маршрутизація викликів	41
2.8.1 Структура файлу extensions.conf	42

2.9 Процес виклику	44
2.10 Шаблони номерів	45
2.11 Висновок по розділу	45
3 Дослідження розробленої системи	46
3.1 Сучасні варіанти дослідження системи	46
3.2 Методологія	47
3.3 Тестування	50
3.4 Результат	53
3.5 Висновок по розділу	58
Висновки	59
Перелік джерел посилання	60

ВСТУП

Сучасні інформаційні технології значно впливають на різні аспекти комунікації, зокрема на телекомунікаційні системи. Однією з найважливіших технологій, яка отримала широке поширення в останні роки, є IP-телефонія. IP-телефонія — це технологія, яка дозволяє передавати голосові дані в реальному часі через Інтернет-протоколи, забезпечуючи високу якість зв'язку.

На відміну від традиційних стаціонарних телефонів, які обмежені фізичними лініями, IP-телефонія надає користувачам можливість здійснювати дзвінки через Інтернет, використовуючи різні пристрої: комп'ютери, смартфони, планшети або звичайні телефони. Це відкриває нові можливості для організації внутрішніх і зовнішніх комунікацій, підвищуючи гнучкість та ефективність взаємодії.

Однією з головних переваг IP-телефонії є її здатність інтегруватися з іншими інформаційними системами, такими як CRM або контакт-центри, що робить її особливо привабливою для корпоративних користувачів. Більше того, IP-телефонія дозволяє суттєво оптимізувати витрати на міжміські та міжнародні дзвінки, що є актуальним для компаній з розгалуженою мережею офісів і співробітників у різних містах і країнах.

Актуальність теми дослідження полягає в тому, що багато підприємств все ще використовують застарілі аналогові телефонні мережі, які вже не можуть задовольнити сучасні потреби в швидкості та якості зв'язку. У зв'язку з цим IP-телефонія пропонує інноваційне рішення, здатне відповідати вимогам сучасного ринку телекомунікацій, забезпечуючи гнучкість, масштабованість і високу продуктивність.

Таким чином, дослідження можливостей IP-телефонії, її впровадження в корпоративні системи та аналіз переваг перед аналоговими рішеннями є актуальним та важливим завданням для подальшого розвитку телекомунікаційних технологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія IP-телефонії

IP-телефонія – це сучасна телекомунікаційна технологія, яка може використовуватися з будь-якими мережами зв'язку, включно з мобільними телефонами, локальними мережами та інтернетом. Здебільшого ця технологія використовується для організації телефонних мереж великих компаній з метою економії коштів на міжнародних і міжміських дзвінках.

IP-телефонія охоплює різноманітні комунікаційні протоколи, технології їх побудови та методи організації, які дозволяють здійснювати традиційні телефонні дзвінки, а також проводити аудіо та відеоконференції. Передача сигналів відбувається у вигляді пакетів даних, завдяки чому інформація може бути стиснена, що сприяє зменшенню навантаження на канали передачі даних.

IP-телефонію розробили наприкінці 80-х років минулого століття; піонером IP-телефонії стала ізраїльська компанія Vocal Technologies, якій до 1995 року вдалося об'єднати досягнення в галузі цифрового оброблення сигналів (DSP), кодеків, комп'ютерів і протоколів маршрутизації. З 1995 року в IP-телефонії використовують два методи стиснення мови - GSM і True Speech від DSP Group, який сильно стискає вихідний голосовий сигнал і знижує втрати якості під час декомпресії. Крім алгоритмів стиснення/декомпресії мови і стандартних протоколів, IP-телефонія також постійно вдосконалює способи боротьби із затримками в Інтернеті.

1.2 Принципи пакетної передачі мовлення

Класична телефонна мережа базується на технології комутації каналів, яка вимагає, щоб кожному телефонному дзвінку було призначено фізичне з'єднання.

Таким чином, телефонний дзвінок – це фізичне з'єднання з телефонним каналом. У цьому випадку аналоговий сигнал 3,1 кГц надсилається до найближчої АТС, де він мультиплексується з сигналами інших користувачів, підключених до АТС, за допомогою технології часового розділення. Потім груповий сигнал передається міжстанційною каналною мережею; коли він досягає адреси АТС, сигнал демультиплексується і надсилається за призначенням.

Головний недолік телефонної мережі з комутацією каналів полягає в нераціональному використанні пропускної здатності — під час пауз у розмові канал залишається без корисного навантаження.

Перехід від аналогових до цифрових технологій став важливим кроком у становленні сучасних цифрових мереж зв'язку. Одним з таких кроків у розвитку цифрової телефонії став перехід до комутації пакетів. У мережах з комутацією пакетів канал зв'язку несе інформаційні одиниці, підключені до фізичного середовища. Ці одиниці, пакети даних, кадри або комірки (залежно від протоколу), передаються не тільки в загальних мережах, але й по незалежних віртуальних каналах, в залежності від фізичного середовища [1].

Кожен пакет має заголовок, який містить дані про канал, включаючи інформацію про джерело (відправника) та призначення (одержувача). Цей заголовок допомагає забезпечити правильну маршрутизацію пакета через мережу до його кінцевого пункту.

У IP-мережах будь-яка інформація — голос, текст, відео, програми або інші дані — передавалася у вигляді пакетів. Кожен комп'ютер або термінал мав унікальну IP-адресу, а пакети даних доставлялися одержувачу на основі адреси, зазначеної в заголовку. Передача даних могла відбуватися паралельно між кількома користувачами або процесами на одній лінії. У разі виникнення проблем IP-мережа була здатна перенаправити пакети, оминаючи несправні ділянки. Протокол IP не вимагав виділеного каналу для сигналізації.

Процес передачі голосу через IP-мережі складався з кількох етапів.

На першому етапі голос оцифровувався. Потім оцифровані дані аналізувалися та оброблялися, щоб зменшити фізичний обсяг даних, що

передавалися до отримувача. Як правило, на цьому етапі пригнічувалися небажані паузи, фоновий шум та проводилася компресія.

На наступному етапі отримані дані розбивалися на пакети, до яких додавалася протокольна інформація: адреса отримувача, порядковий номер пакету (на випадок, якщо пакети доставлялися в неправильному порядку), а також додаткові дані для виправлення помилок. Це дозволяло тимчасово накопичувати обсяг інформації, необхідний для створення пакету, перед його передачею в мережу.

Витяг переданої голосової інформації з отриманих пакетів даних також відбувався в кілька етапів. Коли голосовий пакет надходив на приймальний термінал, спочатку перевірявся порядок проходження пакетів: оскільки IP-мережі не гарантували час доставки, пакети даних з вищими порядковими номерами могли надходити раніше і мати різні часові інтервали прийому. Пакети даних тимчасово зберігалися, щоб відновити вихідну послідовність і синхронізацію. Однак деякі пакети могли бути повністю втрачені під час доставки або затримка доставки могла перевищити допустимий діапазон поширення. За звичайних обставин одержувач запитував повторну передачу невірних або відсутніх даних. Проте, оскільки передача голосу була надто критичною до часу доставки, можна було або використовувати алгоритм апроксимації, який приблизно відновлював втрачені дані на основі отриманих пакетів, або просто ігнорувати ці втрати та заповнювати прогалини даними.

В результаті послідовність даних стискається і перетворюється безпосередньо в мовний сигнал, і мовна інформація передається одержувачу.

В результаті отримана інформація, ймовірно, не відповідає вихідній інформації (спотворення) і виникають затримки (проміжне сховище необхідне як для обробки відправником, так і для обробки одержувачем).

Однак такі втрати до певної міри терпимі завдяки надлишковості голосових повідомлень.

Оператори мереж з комутацією пакетів в основному отримують вигоду від спільного використання комунікаційної інфраструктури.

Коротше кажучи, згідно зі статистичним аналізом мережі, вони можуть продавати більше доходів, ніж отримують насправді. Оскільки не передбачається, що користувачі використовуватимуть весь платний спектр 24 години на добу, магістральну інфраструктуру можна розширити, щоб обслуговувати більше користувачів. І продажі, і прибуток збільшаться.

Іншими словами, абоненти, які платять за пропускну здатність 64 кбіт/с, у середу використовували лише 25 відсотків каналів. Це дозволяє операторам продавати вчетверо більше доступних ресурсів вчетверо більшій кількості користувачів, не перевантажуючи свої мережі. Це вигідно як клієнтам, так і постачальникам, оскільки оператори можуть зменшити витрати і збільшити доходи за рахунок зниження абонентської плати. Це успішне рішення отримало визнання в галузі передачі даних і тепер використовується на ринку телефонії.

В даний час існує два основних способи, за допомогою яких IP-телефонія передає пакети голосу і даних через IP-мережі:

- через глобальну мережу Інтернет (Інтернет-телефонія);
- за допомогою виділеної лінійної мережі передачі (IP-телефонія).

У першому випадку пропускну здатність безпосередньо залежить від завантаження інтернет-пакетів, що містять дані, голос, графіку тощо, що може призвести до дуже різних затримок при комутації пакетів. Використання каналів, виділених для голосових пакетів даних, гарантує постійну (або майже постійну) швидкість передачі даних. У зв'язку з широким розповсюдженням Інтернету особлива увага приділяється використанню систем Інтернет-телефонії, але слід визнати, що в цьому випадку якість телефонної передачі не може бути гарантована.

Щоб використовувати телефонні сервери для міжміського (міжнародного) зв'язку, організація або провайдер послуг повинні встановити сервер у місці або місцях, звідки планується здійснювати дзвінки. Вартість таких з'єднань набагато нижча, ніж вартість дзвінків через звичайні телефонні лінії. Ця різниця особливо важлива у випадку міжнародних дзвінків.

Загальний принцип роботи телефонного сервера VoIP: з одного боку, сервер під'єднано до телефонної лінії, і його можна під'єднати до будь-якого телефону у

світі. З іншого боку, сервер під'єднано до Інтернету, і його можна під'єднати до будь-якого комп'ютера у світі. Сервер отримує стандартний телефонний сигнал, оцифровує його (якщо це не цифровий сигнал), сильно стискає, розбиває на пакети даних і відправляє їх у пункт призначення в Інтернеті за допомогою протоколу IP. Для пакетів даних, що передаються з мережі на телефонний сервер або телефонну лінію, операція виконується у зворотному порядку. Ці дві операції (введення сигналу в телефонну мережу і виведення сигналу з телефонної мережі) відбуваються практично одночасно, що дає змогу вести повно дуплексні переговори. На основі цих базових операцій можуть бути побудовані різні конфігурації. Наприклад, телефон-комп'ютер і комп'ютер-телефон можуть бути забезпечені одним сервером телефонії. Для зв'язку між телефонами (факсами) потрібно два сервери.

Основною перешкодою для поширення IP-телефонії є відсутність в IP-протоколі механізму, що гарантує якість обслуговування, що є не найнадійнішим способом голосового зв'язку. Сам IP-протокол не гарантує доставку пакетів даних та їх час, що призводить до таких проблем, як переривання голосового зв'язку та невдалі дзвінки. Це може призвести до таких проблем, як переривання голосового зв'язку і невдалі дзвінки. Сьогодні ці проблеми вирішуються. Органи стандартизації розробляють нові протоколи, виробники випускають нові пристрої, але на цьому рівні Інтернет операбельність і стандартизація вже не поширюються на мови пакування. Деградація якості передавання голосу прийнятна на середньому рівні в корпоративних мережах із високим навантаженням на ресурси, але більш серйозна в мережах загального користування.

Оскільки оператори надають послуги і стягують за них плату, вони зобов'язані гарантувати їхню якість. Навіть якщо клієнти час від часу миряться з низьким рівнем якості (що малоймовірно на конкурентному ринку телекомунікацій), вони все одно можуть поскаржитися, якщо виникнуть серйозні або довгострокові проблеми. Проте оператори змушені стежити за якістю послуг, що вимагає відповідного обладнання та програмного забезпечення в разі великомасштабного надання, яке дуже дороге і доступне не у всіх районах.

З точки зору масштабованості (якщо не брати до уваги проблему неконтрольованого зниження навантаження на мережу) IP-телефонія – ідеальне рішення. По-перше, тому що з'єднання на базі IP можуть ініціюватися (і завершуватися) з будь-якої точки мережі, від користувача до магістралі. Тому IP-телефонія може бути розгорнута в мережі на кожному конкретному об'єкті, а міграцію можна здійснювати згори донизу, знизу догори або іншим чином. Рішення IP-телефонії характеризуються певною модульністю. Кількість і потужність різних шлюзових вузлів, гейткіперів (серверів обробки викликів у термінах VoIP), можуть бути збільшені практично самостійно відповідно до поточних потреб. Зрозуміло, при цьому не враховується збільшення ресурсів самої мережевої інфраструктури, оскільки самі мережеві вузли можуть бути як незалежними від системи IP-телефонії, так і об'єднаними за функціями.

1.3 Шлюзи IP-телефонії

VoIP-шлюзи можуть мати різні аналогові або цифрові інтерфейси. Аналогові інтерфейси FXS і FXO повинні бути підключені до аналогового телефону, АТС або аналогової телефонної лінії (PSTN), тоді як цифровий інтерфейс Е1 призначений для передачі даних зі швидкістю 64 кбіт/с. Деякі моделі мають роз'єми для альтернативних аналогових лінійних підключень.

VoIP-шлюз підключається до вільного порту АТС, зовнішньої завантажувальної лінії або вільного внутрішнього номера АТС. Далі встановлюються і задаються правила маршрутизації. Відповідно до цих правил частина викликів спрямовується безпосередньо на АТС, а частина – в IP-мережу через VoIP-шлюз [2].

Класична організація офісної телефонії виглядає наступним чином: АТС підключена до локальної мережі (з обмеженою кількістю зовнішніх ліній) і підтримує певну кількість внутрішніх користувачів. Дзвінки між користувачами є

внутрішніми і безкоштовними. Вихідні дзвінки управляються АТС і тарифікуються відповідно до обраних тарифів. Міжміські та міжнародні дзвінки обробляються Федеральним оператором зв'язку. При використанні шлюзів та ІР-телефонії обмеження обмежується кількістю зовнішніх ліній; інші обмеження є умовними і залежать від вибору ІР-АТС [3].

1.4 Способи підключення до ІР-телефонії

Підключення до ІР-телефонії, що надається телефонними операторами, зазвичай здійснюється двома способами:

- підключення до VoІР-телефону за допомогою SIP-телефону (рис. 1.1);
- підключення до VoІР-телефону за допомогою VoІР-шлюзу (рис. 1.2).

Такий спосіб підключення жодним чином не заважає замінити свій SIP-телефон і VoІР-шлюз на ІР-АТС і прокласти до нас лінії зв'язку за власним бажанням. Провайдер телефонії також не визначає, що кінцевим пристроєм є телефонний сервер, а не телефон користувача.

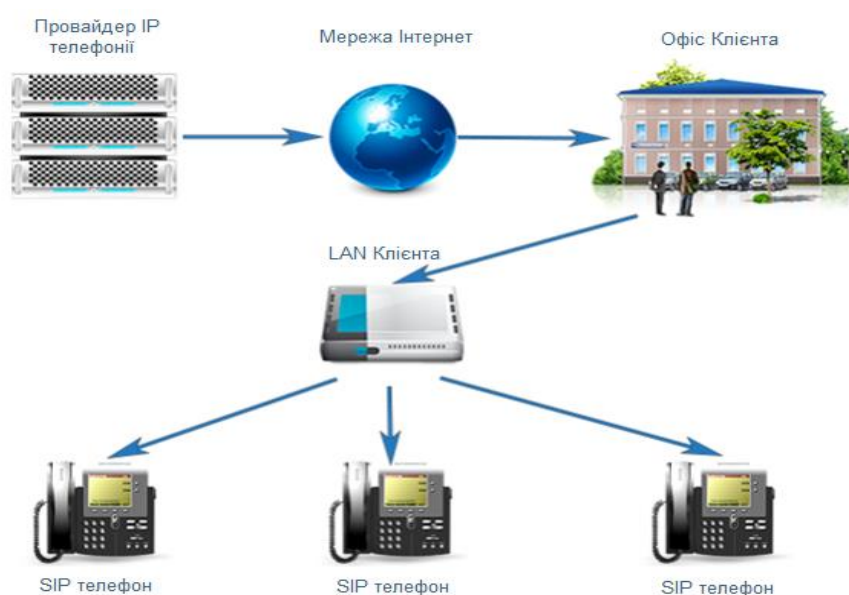


Рисунок 1.1 – Телефонні з'єднання VoІР з використанням SIP-телефонії

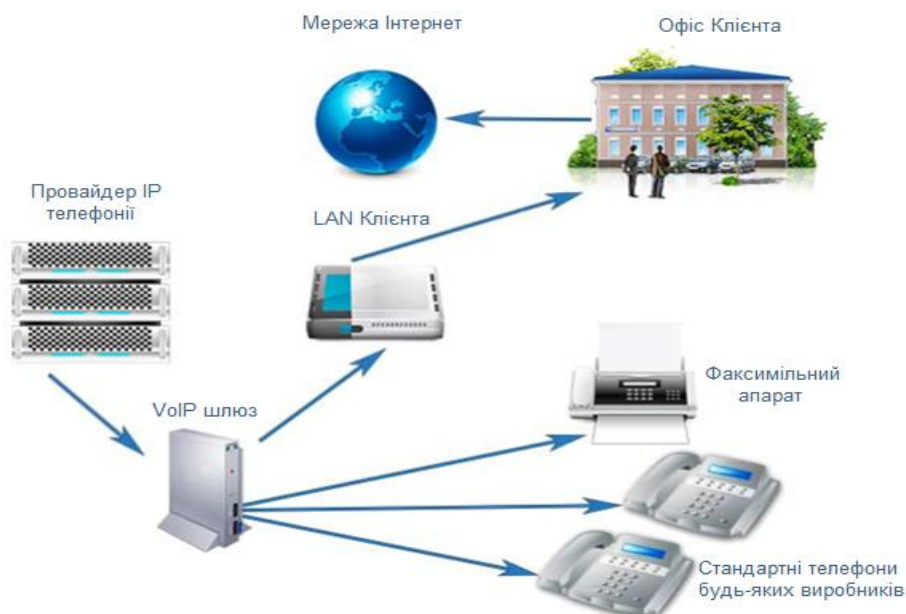


Рисунок 1.2 – Телефонні з’єднання VoIP з використанням шлюзів VoIP

1.5 Протокол SIP в IP-телефонії

Протокол ініціювання сеансу (SIP) – це протокол передавання даних, що описує порядок встановлення і завершення користувацької інтернет-сесії, включно з обміном мультимедійним контентом (IP-телефонія, аудіо- та відеоконференції, обмін миттєвими повідомленнями) [4].

Цей протокол описує, як клієнтська програма використовує унікальне ім’я для запиту з’єднання з іншим клієнтом, який може бути фізично віддаленим у тій же мережі. Цей протокол визначає спосіб досягнення домовленості між клієнтами про відкриття каналу зв’язку на основі інших протоколів (наприклад, RTP), які можуть бути використані для безпосереднього надсилання інформації.

Такі канали можуть додаватися або видалятися під час встановленої сесії, а додаткові клієнти можуть підключатися або відключатися (тобто брати участь в обміні між двома або більше сторонами телеконференції). Протокол визначає процедуру завершення сеансу.

1.6 Особливості програмного пакета Asterisk для створення IP-телефонів

Програмне забезпечення Asterisk (ПК) може використовуватися організаціями з особливими вимогами до безпеки даних, оскільки воно відокремлює Asterisk від Інтернету і дозволяє створити локальну телефонну мережу з усіма можливостями, не пов'язаними з Інтернетом.

Поширюється і розробляється під ліцензією GPL. Для підвищення безпеки з'єднання між користувачем і IP-АТС Asterisk на сервері працює сервер автентифікації на транспортному рівні (TLS), видаються ключі шифрування і є можливість конфігурування телефону.

Особливості АТС на основі Asterisk:

- можливість підключення до будь-якого провайдера IP-телефонії без додаткових витрат;
- підключення до традиційного телефонного оператора за допомогою інтерфейсної карти або шлюзу;
- необмежена кількість користувачів через VoIP;
- функція запису дзвінків, детальна статистика та можливість виставлення рахунків;
- підтримка голосового меню та голосової пошти;
- відправлення та отримання факсів через електронну пошту;
- підключення віддалених або мобільних користувачів з будь-якої точки світу;
- повна переадресація дзвінків, у тому числі на мобільні телефони;
- об'єднання необмеженої кількості філій в єдиний план нумерації;
- інтеграція з веб-сайтами та CRM-системами для управління взаємодією з клієнтами;
- використання програмного телефону на ПК, VoIP-телефону або фіксованого VoIP-телефону;
- інтеграція з корпоративними програмними рішеннями.

1.6.1 Переваги IP-телефонії на базі Asterisk

Переваги IP-телефонії полягають у її гнучкості та економічності. Вона дозволяє легко та оперативно змінювати конфігурацію віртуальної АТС, додаючи або виключаючи абонентів. Це дозволяє створювати телефонні мережі будь-якого масштабу без необхідності встановлювати фізичне обладнання у кожному офісі чи філії. Використання віртуальних каналів для обміну даними в єдиній корпоративній мережі значно спрощує управління зв'язком.

Основна перевага IP-телефонії перед традиційною полягає в суттєвій економії коштів. Витрати на зв'язок помітно зменшуються вже в перший місяць після впровадження, а широкий вибір тарифів дає змогу обрати оптимальне рішення для будь-якого бізнесу. Співробітники можуть залишатися на зв'язку незалежно від місця роботи — в офісі, віддалено чи у відрядженні, без додаткових витрат на дзвінки між філіями, оскільки всі номери компанії об'єднуються в єдину мережу.

IP-телефонія також надає розширені можливості, такі як проведення конференцій, автоматична переадресація дзвінків, доступ до статистики та контроль витрат. Завдяки використанню SIP-протоколу, підключення не вимагає прокладання додаткових ліній, що також знижує витрати.

Хоча IP-телефонія має мінімальні недоліки, її переваги роблять її очевидним вибором для багатьох компаній.

1.6.2 Обґрунтування вибору Asterisk як АТС

Asterisk — це гнучке програмне рішення, яке не потребує прив'язки до конкретного апаратного забезпечення. Воно підтримує роботу в таких операційних системах, як Linux, FreeBSD, OpenBSD і Solaris, що робить його універсальним вибором для різних ІТ-інфраструктур.

Це рішення дозволяє інтегрувати різноманітне обладнання для забезпечення всіх функцій класичної АТС, підтримує кілька VoIP-протоколів і пропонує широкі можливості для керування дзвінками. Серед основних функцій:

- голосова пошта;
- організація телефонних конференцій;

- IVR (інтерактивне голосове меню);
- центр обробки викликів із підтримкою різних алгоритмів черги та розподілу дзвінків;
- детальні записи дзвінків.

Для додавання нових функцій можна створювати план нумерації за допомогою власної мови програмування Asterisk, писати модулі на С або використовувати AGI — універсальний інтерфейс для інтеграції з зовнішніми системами обробки даних. Модулі через AGI можуть бути написані будь-якою мовою програмування.

Asterisk випускається за подвійною ліцензією, що дозволяє використовувати як відкриті модулі під ліцензією GNU GPL, так і приватні модулі з ліцензійним кодом. Наприклад, модулі, які підтримують кодек G.729.

1.6.3 Аналіз прив'язки до певного серверного обладнання

В умовах сучасної економіки компанії часто перш за все звертають увагу на вартість впровадження та обслуговування телефонних систем. Asterisk пропонує значну перевагу, оскільки він може працювати практично на будь-якому сервері з операційною системою Linux. Основний критерій вибору обладнання — це продуктивність системи, яка залежить від кількості користувачів, одночасних дзвінків, черг, голосових меню та інших процесів, що навантажують сервер. Чим більша система, тим потужніше повинно бути обладнання, але користувачі не обмежені конкретним виробником серверів, що дає додаткову гнучкість.

Asterisk базується на відкритих телефонних стандартах і використовує загальновизнані протоколи, такі як SIP і IAX2, для підключення серверів і кінцевого обладнання. Це дозволяє використовувати широкий спектр телефонів і пристроїв, включно з «залізними» телефонами, програмними телефонами або шлюзами fxs/fxo, які забезпечують підтримку внутрішніх аналогових телефонів та зовнішніх аналогових ліній. Це надає значні переваги в умовах кризи, оскільки дозволяє заощаджувати на закупівлі обладнання, використовуючи будь-який сумісний SIP-телефон.

На відміну від Asterisk, багато рішень від постачальників, таких як Panasonic, Samsung або LG, передбачають використання їх власного обладнання, що може значно збільшити витрати. Це робить Asterisk більш гнучким та економічно вигідним вибором для компаній, особливо в періоди економічної невизначеності.

1.6.4 Аналіз системи ліцензування системи Asterisk

Типові вимоги до систем телефонії зазвичай включають такі функції, як підключення внутрішніх користувачів із можливістю спілкування за короткими додатковими номерами, підключення кількох SIP-провайдерів, організація черг дзвінків, налаштування голосових привітань для вхідних викликів, можливість переадресації під час вхідного дзвінка ззовні та голосову пошту. Всі ці функції в Asterisk надаються абсолютно безкоштовно, що є значною перевагою в порівнянні з пропрієтарними рішеннями від вендорів.

Для прикладу, вартість аналогічної функціональності у традиційних системах телефонії може бути надзвичайно високою. Наприклад, плата Panasonic KX-TDA0194X, яка підтримує лише 4 канали голосової пошти, коштує значних грошей. Якщо ж потрібно реалізувати таку функціональність для 100 користувачів, витрати на обладнання будуть дуже великими.

Asterisk дозволяє уникнути цих витрат завдяки своїй відкритій архітектурі та підтримці стандартів SIP, забезпечуючи економічно ефективне рішення для малого, середнього та великого бізнесу.

1.6.5 Аналіз методу інтеграції з CRM-системами та іншими АТС

Системи уніфікованих комунікацій, які об'єднують CRM, телефонні системи, чати та статуси присутності, стають все більш популярними. Використовуючи Asterisk, ви отримуєте можливість інтегрувати всі ці компоненти завдяки спеціально розробленим каналам, які є частиною Asterisk і розповсюджуються безкоштовно. Наприклад, для підтримки пропрієтарного протоколу Cisco SCCP в Asterisk існує окремий канал (`chan_sccp`).

При використанні інших АТС можуть виникнути складнощі з інтеграцією, яка або зовсім неможлива, або потребує дорогих ліцензій. Це також стосується інтеграції з CRM-системами, такими як SugarCRM, VTigerCRM, AMoCRM та інші.

Для Asterisk вже існують безкоштовні модулі інтеграції з цими системами, що робить його більш доступним рішенням.

1.6.6 Обслуговування та адмініструючий персонал

Для управління IP-телефонами на базі Asterisk, що працює на ПК, дійсно потрібен кваліфікований адміністратор, який має досвід роботи з цією системою. Однак важливо пам'ятати, що рішення щодо впровадження телекомунікаційних систем також приймаються постачальниками, які часто є спеціалістами у своїй вузькій сфері. Ті, хто вже працює з Linux та іншими рішеннями з відкритим кодом, зазвичай можуть не тільки налаштовувати Asterisk, але й вирішувати нові виклики в IT-інфраструктурі, замість того, щоб покладатися на стандартні шаблони налаштування.

Хоча Asterisk не є повністю безкоштовним рішенням, його загальна вартість залежить в основному від витрат на налаштування і роботу адміністратора, а не від програмних обмежень. Це робить Asterisk ідеальним вибором для корпоративних IP-АТС, оскільки ви оплачуєте лише ресурси та послуги адміністрування, а не ліцензійні обмеження.

1.6.7 Цілі та завдання програмної АТС на базі Asterisk

Asterisk — це надзвичайно гнучка телефонна система, яка легко інтегрується з існуючими системами в різних виробничих відділах чи управліннях без додаткових витрат. Незважаючи на велику кількість користувачів і дзвінків у системі, для її ефективного управління достатньо одного кваліфікованого спеціаліста [5].

Основне завдання модернізації телефонної системи — перехід на пакетну передачу даних, що підвищує захищеність інформації, особливо якщо використовуються віртуальні приватні мережі (VPN). Це також спрощує розширення, підтримку та резервне копіювання системи в майбутньому.

Ще однією важливою функцією Asterisk є організація аудіо- та відеоконференцій, що можна реалізувати на одній платформі. Система ефективно маршрутизує дзвінки, записує їх, а також вирішує інші завдання, пов'язані з управлінням дзвінками та взаємодією користувачів.

Крім того, Asterisk може інтегруватися з іншими системами, такими як системи пожежної безпеки, для автоматизованих екстрених сповіщень, або використовуватися для контролю присутності працівників на робочому місці та розрахунку заробітної плати.

Для малих і середніх підприємств Asterisk є особливо вигідним рішенням, оскільки основні витрати зводяться лише до придбання телефонів і серверів, а саме програмне забезпечення розповсюджується безкоштовно. Система може обслуговувати до 2000 користувачів, а підключення філій до основної АТС дозволяє значно заощадити на міжміських та міжнародних дзвінках завдяки використанню інтернет-технологій.

Окрім цих функцій, Asterisk може бути адаптований до різноманітних завдань і потреб, забезпечуючи додаткові або вузькоспеціалізовані рішення для бізнесу.

1.7 Висновки за розділом

У цьому розділі розглядається розвиток IP-телефонії, починаючи з 1980-х років, як сучасної технології для передачі голосу через інтернет-протоколи. Відзначається роль компанії Vocal Technologies у впровадженні цифрових протоколів та методів стиснення мови, що дозволило зменшити обсяг переданих даних і зберегти якість зв'язку.

Також аналізуються принципи пакетної передачі голосу, що замінила традиційну комутацію каналів, та переваги цієї технології — ефективніше використання пропускну здатності та зниження витрат. Розглянуто роль VoIP-шлюзів і програмних рішень, таких як Asterisk, що забезпечують гнучкість, масштабованість та інтеграцію з іншими бізнес-системами.

Розділ підкреслює важливість IP-телефонії як економічно вигідного й ефективного інструменту для корпоративного зв'язку.

2 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ ASTERISK

2.1 План встановлення та налаштування Asterisk

Щоб створити ефективну телефонну систему на основі Asterisk, передбачено такі основні кроки:

- підготовка системи — оновлення пакунків і встановлення залежностей для сумісної роботи Asterisk із сервером;
- встановлення DAHDI та LibPRI — для підключення аналогових та цифрових телефонних ліній, а також ISDN-з'єднань;
- інсталяція Asterisk — завантаження та збирання Asterisk з вихідного коду;
- налаштування конфігурацій — створення та налаштування файлів `pjsip.conf` і `extensions.conf` для підключення користувачів і маршрутизації дзвінків;
- забезпечення безпеки — створення окремого користувача для Asterisk та застосування шаблонів для зручного управління групами користувачів;
- маршрутизація дзвінків — налаштування діалплану для управління викликами;
- тестування та оптимізація — перевірка стабільності та працездатності системи.

Ці етапи забезпечать безпечну та функціональну телефонну систему з гнучкими налаштуваннями для зручного управління.

2.1.1 Встановлення необхідних залежностей

Перед початком роботи необхідно переконатися, що всі пакети Ubuntu встановлені та оновлені на сервері за допомогою такої команди показано на рисунку 2.1:

```

root@king-VirtualBox:~# apt-get update -y
Hit:1 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done

```

Рисунок 2.1 – Оновлення пакетів

Наступні залежності потрібно встановити для налаштування сервера Asterisk. Усі їх можна встановити за допомогою такої команди показано на рисунку 2.2:

```

root@king-VirtualBox:~# apt-get install build-essential git autoconf wget
subversion pkg-config libjansson-dev libxml2-dev uuid-dev libsqlite3-dev
libtool -y

```

Рисунок 2.2 – Встановлення залежностей

Після встановлення всіх пакетів необхідно встановити DAHDI для забезпечення зв'язку Asterisk з аналоговими та цифровими телефонами. Спочатку завантажується остання версія DAHDI [6], після чого вона компілюється за допомогою таких команд показаних на рисунках 2.3 та 2.4:

```

root@king-VirtualBox:/opt# git clone -b next git://git.asterisk.org/dahdi/linux dahdi-linux
Cloning into 'dahdi-linux'...
remote: Counting objects: 9853, done.
remote: Compressing objects: 100% (2145/2145), done.
remote: Total 9853 (delta 6397), reused 9853 (delta 6397)
Receiving objects: 100% (9853/9853), 6.90 MiB | 1.02 MiB/s, done.
Resolving deltas: 100% (6397/6397), done.
root@king-VirtualBox:/opt# cd dahdi-linux
root@king-VirtualBox:/opt/dahdi-linux# make

```

Рисунок 2.3 – Клонування DAHDI

```
#####
###
### DAHDI installed successfully.
### If you have not done so before, install the package
### dahdi-tools.
###
#####
```

Рисунок 2.4 – Успішне встановлення DAHDI

На рисунках 2.5 та 2.6 завантажуються інструменти DAHDI, після чого їх встановлюють і налаштовують:

```
root@king-VirtualBox:/opt# git clone -b next git://git.asterisk.org/dahdi/tools dahdi-tools
Cloning into 'dahdi-tools'...
remote: Counting objects: 2161, done.
remote: Compressing objects: 100% (764/764), done.
remote: Total 2161 (delta 1374), reused 2161 (delta 1374)
Receiving objects: 100% (2161/2161), 1.36 MiB | 668.00 KiB/s, done.
Resolving deltas: 100% (1374/1374), done.
root@king-VirtualBox:/opt# cd dahdi-tools
root@king-VirtualBox:/opt/dahdi-tools# autoreconf -i
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'auxdir'.
libtoolize: copying file 'auxdir/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'm4'.
libtoolize: copying file 'm4/libtool.m4'
libtoolize: copying file 'm4/ltoptions.m4'
libtoolize: copying file 'm4/ltugar.m4'
libtoolize: copying file 'm4/ltversion.m4'
libtoolize: copying file 'm4/lt-obsolete.m4'
configure.ac:69: installing 'auxdir/compile'
configure.ac:82: installing 'auxdir/config.guess'
configure.ac:82: installing 'auxdir/config.sub'
configure.ac:15: installing 'auxdir/install-sh'
configure.ac:15: installing 'auxdir/missing'
Makefile.am: installing 'auxdir/depcomp'
root@king-VirtualBox:/opt/dahdi-tools#
```

Рисунок 2.5 – Клонування інструментів DAHDI

```
#####
###
### DAHDI tools installed successfully.
### If you have not done so before, install init scripts with:
###
###   make install-config
###
#####
```

Рисунок 2.6 – Успішне встановлення інструментів DAHDI

Далі завантажуються LibPRI, щоб забезпечити зв'язок Asterisk із з'єднаннями ISDN показано на рисунку 2.7.

```

root@king-VirtualBox:/opt/libpri# make install
mkdir -p /usr/lib
mkdir -p /usr/include
install -m 644 libpri.h /usr/include
install -m 755 libpri.so.1.4 /usr/lib
#if [ -x /usr/sbin/sestatus ] && ( /usr/sbin/sestatus | grep "SELinux status:" | grep -q "enabled"); then /sbin/restorecon -v /usr/lib/libpri.so.1.4; fi
( cd /usr/lib ; ln -sf libpri.so.1.4 libpri.so )
install -m 644 libpri.a /usr/lib
if test $(id -u) = 0; then /sbin/ldconfig -n /usr/lib; fi

```

Рисунок 2.7 – Запуск скриптів LibPRI

2.2 Встановлення Asterisk

За замовчуванням пакет Asterisk не доступний у стандартному репозиторії Ubuntu 20.04. Тому необхідно завантажити та зібрати його з вихідного коду [7]. Для цього його завантажують з GitHub за допомогою такої команди (рис. 2.8):

```

root@king-VirtualBox:/opt# git clone -b 18 https://gerrit.asterisk.org/asterisk asterisk-18
Cloning into 'asterisk-18'...
remote: Counting objects: 351165, done
remote: Finding sources: 100% (351165/351165)
remote: Total 351165 (delta 266900), reused 347277 (delta 266900)
Receiving objects: 100% (351165/351165), 179.89 MiB | 746.00 KiB/s, done.
Resolving deltas: 100% (266900/266900), done.

```

Рисунок 2.8 – Клонування Asterisk

На рисунку 2.9 показано встановлення необхідних залежностей. Далі встановлення Asterisk показано на рисунку 2.10:

```

root@king-VirtualBox:/opt/asterisk-18# contrib/scripts/get_mp3_source.sh
A   addons/mp3
A   addons/mp3/MPGLIB_README
A   addons/mp3/common.c
A   addons/mp3/huffman.h
A   addons/mp3/tabinit.c
A   addons/mp3/Makefile
A   addons/mp3/README
A   addons/mp3/decode_i386.c
A   addons/mp3/dct64_i386.c
A   addons/mp3/MPGLIB_TODO
A   addons/mp3/mpg123.h
A   addons/mp3/layer3.c
A   addons/mp3/mpglib.h
A   addons/mp3/decode_ntom.c
A   addons/mp3/interface.c
Exported revision 202.

```

Рисунок 2.9 – Встановлення залежностей для Asterisk

```

Setting up libcfg-dev:amd64 (3.0.3-2ubuntu2.1) ...
Setting up libspandsp-dev:amd64 (0.0.6+dfsg-2) ...
Setting up freetds-dev (1.1.6-1.1) ...
Setting up po-debconf (1.0.21) ...
Setting up libjack-jackd2-dev:amd64 (1.9.12~dfsg-2ubuntu2) ...
Setting up libical-dev:amd64 (3.0.8-1) ...
Setting up portaudio19-dev:amd64 (19.6.0-1build1) ...
Setting up libcpg-dev:amd64 (3.0.3-2ubuntu2.1) ...
Setting up libnewt-dev:amd64 (0.52.21-4ubuntu2) ...
Setting up libneon27-dev (0.30.2-4) ...
Setting up dh-autoreconf (19) ...
Setting up odbcinst1debian2:amd64 (2.3.6-0.1build1) ...
Setting up unixodbc-dev:amd64 (2.3.6-0.1build1) ...
Setting up odbcinst (2.3.6-0.1build1) ...
Setting up debhelper (12.10ubuntu1) ...
Setting up vpb-driver-source (4.2.61-1build1) ...
Processing triggers for libgl1:amd64 (2.64.6-1~ubuntu20.04.4) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Setting up libgmime-2.6-dev (2.6.23+dfsg1-4) ...
Setting up libgmime-3.0-dev:amd64 (3.2.7-1) ...

#####
## install completed successfully
#####

```

Рисунок 2.10 – Успішне встановлення Asterisk

Далі на рисунку 2.11 показано Asterisk налаштовується:

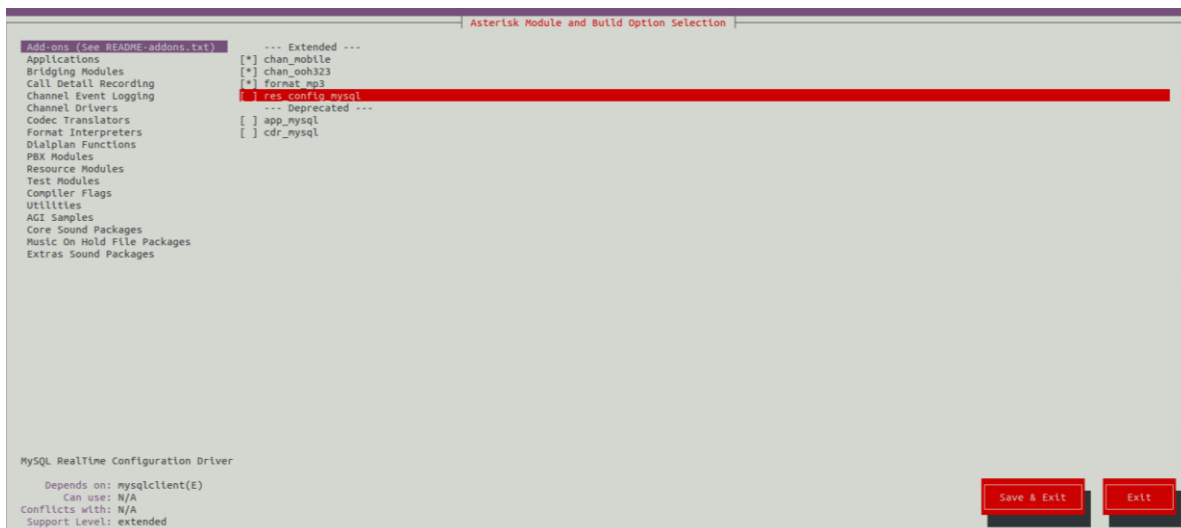


Рисунок 2.13 – Вибір аддонів

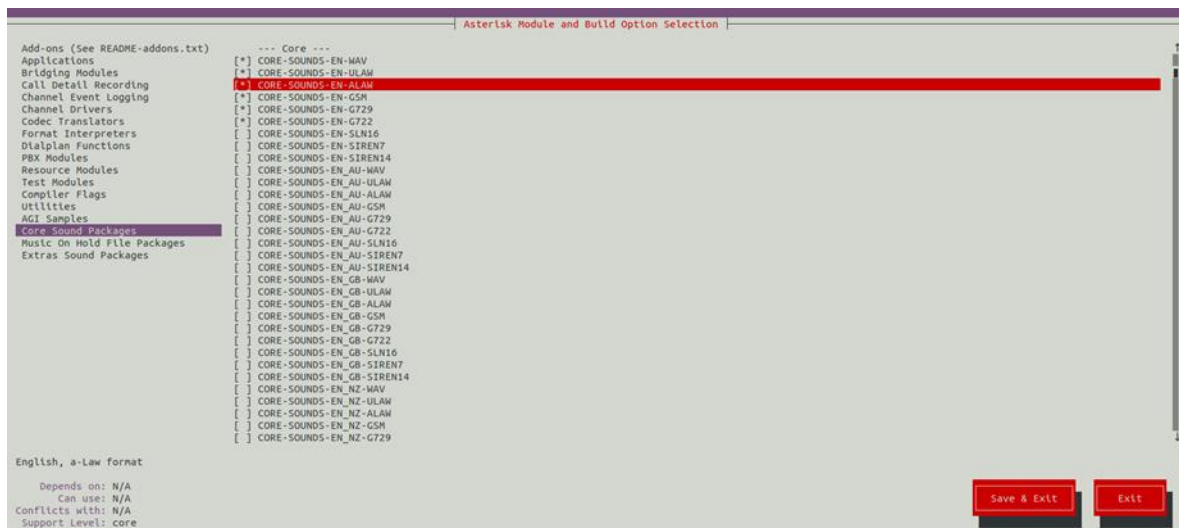


Рисунок 2.14 – Вибір звукових модулів



Рисунок 2.15 – Включення МОН пакетів

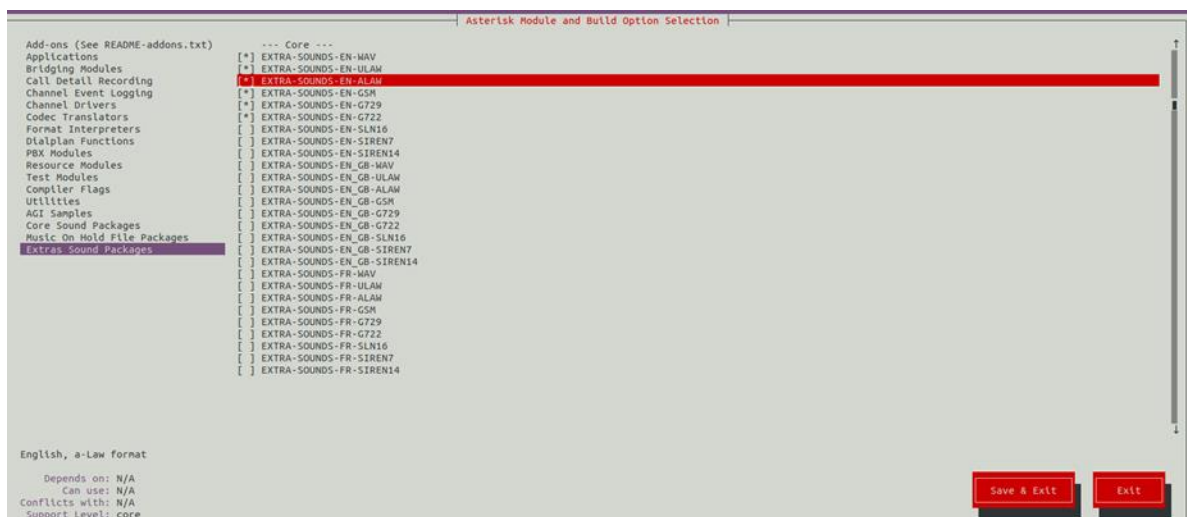


Рисунок 2.16 – Включення звукових пакетів

Потім Asterisk компілюється, на рисунку 2.17 показана команда для компіляції та на рисунку 2.18 показана успішна компіляція:

```
root@king-VirtualBox:/opt/asterisk-18# make -j2
```

Рисунок 2.17 – Команда для компіляції Asterisk.

```

Building Documentation For: third-party channels
+----- Asterisk Build Complete -----+
+ Asterisk has successfully been built, and +
+ can be installed by running:             +
+                                           +
+               make install                +
+-----+

```

Рисунок 2.18 – Успішна компіляція

На рисунку 2.19 встановлення модулів Asterisk:

```

make[1]: Leaving directory '/opt/asterisk-18/sounds'
find rest-api -name "*.json" | while read x; do \
    /usr/bin/install -c -m 644 $x "/var/lib/asterisk/rest-api"
done
+---- Asterisk Installation Complete -----+
+                                           +
+   YOU MUST READ THE SECURITY DOCUMENT   +
+                                           +
+ Asterisk has successfully been installed. +
+ If you would like to install the sample  +
+ configuration files (overwriting any     +
+ existing config files), run:             +
+                                           +
+ For generic reference documentation:      +
+   make samples                           +
+                                           +
+ For a sample basic PBX:                  +
+   make basic-pbx                         +
+                                           +
+----- or -----+
+                                           +
+ You can go ahead and install the asterisk +
+ program documentation now or later run:   +
+                                           +
+               make progdocs               +
+                                           +
+ **Note** This requires that you have     +
+ doxygen installed on your local system   +
+-----+
root@king-VirtualBox:/opt/asterisk-18# █

```

Рисунок 2.19 – Успішне встановлення модулів Asterisk

Запуск сценарію ініціалізації Asterisk здійснюється за допомогою такої команди показаною на рисунку 2.20:

```
root@king-VirtualBox:/opt/asterisk-18# make config
```

Рисунок 2.20 – Команда для запуску сценарію ініціалізації Asterisk

На рисунку 2.21 оновлюються спільні бібліотеки:

```
root@king-VirtualBox:/opt/asterisk-18# ldconfig
```

Рисунок 2.21 – Оновлення бібліотек Asterisk

2.3 Створення Linux користувача та запуск Asterisk

З міркувань безпеки рекомендується запускати Asterisk під окремим користувачем. Тому створюється новий користувач Asterisk на рисунку 2.22:

```
root@king-VirtualBox:/# adduser --system --group --home /var/lib/asterisk --no-create-home --gecos "Asterisk PBX" asterisk
Adding system user 'asterisk' (UID 127) ...
Adding new group 'asterisk' (GID 134) ...
Adding new user 'asterisk' (UID 127) with group 'asterisk' ...
Not creating home directory '/var/lib/asterisk'.
```

Рисунок 2.22 – Створення нового користувача

Потім користувач додається до групи телефонного виклику та аудіо показано на рисунку 2.23:

```
root@king-VirtualBox:/# nano /etc/default/asterisk
root@king-VirtualBox:/# usermod -a -G dialout,audio asterisk
```

Рисунок 2.23 – Додавання користувача до групи викликів та аудіо

Також встановлюються належні права доступу на рисунку 2.24 та дозволи на всі файли та каталоги Asterisk:

```
root@king-VirtualBox:/# chown -R asterisk: /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk /etc/asterisk
root@king-VirtualBox:/# chmod -R 750 /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk /etc/asterisk
```

Рисунок 2.24 – Надання прав доступу

На цьому етапі Asterisk було встановлено та налаштовано. Після запуску перевіряється його стан роботи показано на рисунку 2.25:

```
root@king-VirtualBox:/# systemctl status asterisk
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/init.d/asterisk; generated)
   Active: active (running) since Sun 2024-11-10 01:08:37 EET;
     Docs: man:systemd-sysv-generator(8)
    Tasks: 35 (limit: 9468)
   Memory: 34.6M
    CGroup: /system.slice/asterisk.service
            └─89829 /usr/sbin/asterisk -U asterisk -G asterisk

ноя 10 01:08:37 king-VirtualBox systemd[1]: Starting LSB: Asterisk
ноя 10 01:08:37 king-VirtualBox asterisk[89814]: * Starting Asterisk
ноя 10 01:08:37 king-VirtualBox asterisk[89814]: ...done.
ноя 10 01:08:37 king-VirtualBox systemd[1]: Started LSB: Asterisk
root@king-VirtualBox:/#
```

Рисунок 2.25 – Стан роботи Asterisk

2.4 Структура Asterisk

Налаштування Asterisk здійснювалося шляхом редагування конфігураційних файлів, розташованих у директорії /etc/asterisk/ [9]. Основні файли для початкового налаштування включали:

- pjsip.conf — файл налаштувань мультимедійної бібліотеки PJSIP, що містив налаштування реєстрації, транків та користувачів;

- `extensions.conf` — файл для налаштування діалплану, який визначав параметри обробки дзвінків, переадресації, мелодій на утриманні, автосекретаря та інші функції;
- `asterisk.conf` — основний конфігураційний файл, що містив шляхи до папок, параметри запуску Asterisk тощо;
- `sip.conf` — файл для налаштувань застарілої бібліотеки ``chan_sip``, який не використовували;
- `cdr.conf` — налаштування звітування про дзвінки (Call Detail Record);
- `rtp.conf` — файл для налаштувань протоколу RTP (порти, підтримка ICE тощо);
- `features.conf` — файл для налаштувань перехоплення і паркування викликів, часу набору, DTMF кодів абонентських функцій;
- `modules.conf` — містив інформацію про модулі, які слід завантажувати при старті або вимикати;
- `musiconhold.conf` — файл для налаштувань музики на утриманні;
- `iax.conf` — використовували для налаштування з'єднань за протоколом IAX, зазвичай для об'єднання кількох АТС;
- `voicemail.conf` — детальне налаштування параметрів голосової пошти;
- `logger.conf` — файл для налаштувань журналювання роботи Asterisk.

Тут перераховано лише основні конфігураційні файли Asterisk, хоча їх набагато більше. У повсякденній роботі зазвичай використовувалися переважно два файли: `extensions.conf` для налаштування діалплану та `pjsip.conf` для налаштувань мультимедійної бібліотеки PJSIP.

Частину функцій та повторне завантаження даних з конфігураційних файлів виконували за допомогою консолі Asterisk. Для входу в консоль використовували таку команду:

```
asterisk -r
```

Для більш детального виведення даних можна було додати ключ `-v` (зазвичай використовували `-vvvvvv`). Чим більше літер "v" вказано, тим детальніше

відображалася інформація в консолі. Запуск у режимі підвищеної деталізації відбувався за допомогою команди:

```
asterisk -rvvvvvv
```

2.5 Підключення користувачів

Додавання нових користувачів здійснювалося у файлі `pjsip.conf`. Цей конфігураційний файл складався із секцій, імена яких зазначалися у квадратних дужках. Назви секцій могли бути довільними, за винятком імен `ENDPOINT` і `AOR`, які мали відповідати заголовку SIP URI.

Можливі типи секцій:

- `ENDPOINT` визначав опції SIP протоколу і взаємодіяв з модулями `AOR`, `AUTH` і `TRANSPORT`. Секція `ENDPOINT` обов'язково була пов'язана з однією або кількома секціями `AOR`. У ній задавалися основні параметри SIP підключення, такі як телефон, софтфон чи транк;
- `AOR` (Address of Record) вказував, як встановити зв'язок з `ENDPOINT`. Секція `ENDPOINT` не могла функціонувати без `AOR`. Тут визначали тривалість реєстрації, відповідність голосовій пошті, `MWI`, та налаштовували періодичність SIP повідомлень для моніторингу стану пристрою. Запит на реєстрацію пристрою спочатку оброблявся через `AOR`, і проводився пошук відповідності SIP за заголовком `To`;
- `TRANSPORT` описував налаштування транспортного рівня за допомогою протоколів `UDP`, `TCP`, `WebSockets` і шифрування `TLS/SSL`. Одна транспортна секція могла використовуватися кількома точками `ENDPOINT`, або, за потреби, створювався унікальний транспортний рівень для конкретної точки;
- `System` визначав загальні системні налаштування `PJSIP`;
- `Global` налаштовував глобальні параметри, включаючи можливість увімкнення дебага, зміну рядка `User Agent` і параметр `max forwards`;

- AUTH використовувався для аутентифікації та містив налаштування для вхідних і вихідних реєстрацій. До цієї секції могли бути прив'язані ENDPOINT та REGISTRATIONS. Різні ENDPOINT та REGISTRATIONS за потреби могли використовувати спільну секцію AUTH;
- REGISTRATION зазвичай застосовувався для реєстрації транку від провайдера;
- DOMAIN_ALIAS використовувався для вказання псевдонімів доменів;
- ACL контролював вхідні SIP з'єднання;
- IDENTIFY вказував IP-адреси кінцевих точок ENDPOINT;
- CONTACT дозволяв використовувати короткий контакт замість SIP URI.

У кожній секції була опція type, яка визначала призначення секції в загальній конфігурації.

2.6 Створення SIP користувача

Спочатку створювався обліковий запис першого користувача Asterisk. Користувач міг підключатися за допомогою SIP телефону або софтфону показано на рисунку 2.26. Налаштування відбувалися у файлі pjsip.conf. Додавалася інформація про нового користувача з додатковим номером 1000, при цьому використовувалися чотиризначні внутрішні номери для можливого подальшого зростання кількості абонентів.

```

[tr-udp]
type=transport
protocol=udp
bind=0.0.0.0
allow_reload=true

[1000]
type=endpoint
context=from-internal
disallow=all
allow=allow
transport=tr-udp
auth=auth1000
aors=1000

[auth1000]
type=auth
auth_type=userpass
password=parol
username=1000

[1000]
type=aor
max_contacts=2

```

Рисунок 2.26 – Налаштування SIP користувача

Було додано необхідну інформацію для реєстрації користувача на сервері Asterisk, зокрема:

[tr-udp] – це транспортна секція, про що свідчить запис type=transport:

- protocol – тип протоколу, що використовується клієнтом;
- bind – IP-адреса клієнта. Можна вказати конкретну IP-адресу, щоб заборонити доступ з інших адрес. При необхідності також можна задати порт;
- allow_reload – дозволяє переналаштування транспортного рівня без перезавантаження Asterisk. За замовчуванням, якщо цей параметр не вказано, він має значення true, і після внесення змін до налаштувань Asterisk потрібно перезареєструвати.

[1000] – секція ENDPOINT. Як зазначалося раніше, назва секції ENDPOINT відповідає назві секції AOR:

- type – тип секції; у даному випадку це ENDPOINT;
- context – контекст діалплану, який вказує, куди будуть направлені вхідні дзвінки для цього ENDPOINT;

- `disallow` – визначає, які кодеки необхідно заборонити; у даному випадку вказано "all" – всі кодеки;
 - `allow` – кодеки, які дозволено використовувати. В попередньому рядку заборонено всі кодеки, а в цьому дозволено тільки `alaw`;
 - `transport` – вказує, яку конфігурацію слід використовувати для транспортування. У нашому випадку це `tr-udp`, який ми створили раніше;
 - `auth` – об'єкт аутентифікації, що відноситься до цього ****ENDPOINT****. У нашому випадку це `auth1000`;
 - `aors` – об'єкт AOR для цього ENDPOINT.
- [`auth1000`] – секція аутентифікації:
- `type` – тип використовуваної секції;
 - `auth_type` – тип аутентифікації, що використовується. У моєму випадку – логін та пароль;
 - `password` – пароль для облікового запису;
 - `username` – логін облікового запису.
- [`1000`] – секція AOR:
- `type` – тип секції;
 - `max_contacts` – максимальна кількість контактів, які можуть використовуватися з AOR.

Після збереження конфігурації у файл, застосуємо її до Asterisk.

2.7 Створення та застосування шаблону

Використання шаблонів також є корисним для розподілу клієнтів із різних відділів, які мають різні рівні доступу. Назва шаблону зазначається в квадратних дужках [], після чого слідує знак (!) показано на рисунку 2.27.

```
[marketing-endpoint](!)
type=endpoint
transport=tr-udp
context=from-internal
disallow=all
allow=allow

[pass-auth](!)
type=auth
auth_type=userpass

[two-contact-aor](!)
type=aor
max_contacts=2
```

Рисунок 2.27 – Налаштування шаблону

У шаблоні можна зазначати ті самі параметри, що й при звичайному створенні користувача. Вище наведено приклад простого шаблону з мінімальним набором налаштувань. Однак можливе використання значно більшої кількості параметрів, таких як використовувані кодеки, необхідність використання NAT, контекст, а також дозволені та заборонені мережі для реєстрації абонентів. Далі розглянемо, як застосовувати шаблон для створення користувачів.

Створюємо трьох користувачів рисунок 2.28, використовуючи шаблон. Для застосування шаблону необхідно додати його назву праворуч від назви секції в круглих дужках.

Використовуючи шаблон, потрібно вказувати лише ті параметри, які потрібно змінити. Усі параметри, не зазначені в кожній секції, братимуться з шаблону.

```

;=====Добавочный номер 2000
[2000](marketing-endpoint)
auth=auth2000
aors=2000

[auth2000](pass-auth)
password=2000
username=2000

[2000](two-contact-aor)

;=====Добавочный номер 2001
[2001](marketing-endpoint)
auth=auth2001
aors=2001

[auth2001](pass-auth)
password=2001
username=2001

[2001](two-contact-aor)

;=====Добавочный номер 2002
[2002](marketing-endpoint)
auth=auth2002
aors=2002

[auth2002](pass-auth)
password=2002
username=2002

[2002](two-contact-aor)

```

Рисунок 2.28 – Створення користувачів за шаблоном

Команди для виведення користувачів, зареєстрованих на сервері показано на рисунку 2.29, 2.30 та 2.31:

```

root@king-VirtualBox:~# asterisk -rx "pjsip show aors"

  Aor: <Aor.....> <MaxContact>
  Contact: <Aor/ContactUri.....> <Hash....> <Status> <RTT(ms)...>
=====
  Aor: 2000                                2
  Aor: 2001                                2
  Aor: 2002                                2
  Aor: dcs-aor                             0
  Contact: dcs-aor/sip:sip.digiumcloud.net 05b40e4cbd NonQual nan
Objects found: 4

```

Рисунок 2.29 – Команда pjsip show aors

```

root@king-VirtualBox:~# asterisk -rx "pjsip show auths"

I/OAuth:  <AuthId/UserName.....>
=====

Auth:  auth2000/2000
Auth:  auth2001/2001
Auth:  auth2002/2002

Objects found: 3

```

Рисунок 2.30 – Команда pjsip show auths

```

root@king-VirtualBox:~# asterisk -rx "pjsip show endpoints"

Endpoint:  <Endpoint/CID.....>  <State.....>  <Channels.>
I/OAuth:  <AuthId/UserName.....>
Aor:  <Aor.....>  <MaxContact>
Contact:  <Aor/ContactUri.....>  <Hash.....>  <Status>  <RTT(ms)..>
Transport:  <TransportId.....>  <Type>  <cos>  <tos>  <BindAddress.....>
Identify:  <Identify/Endpoint.....>
Match:  <criteria.....>
Channel:  <ChannelId.....>  <State.....>  <Time.....>
Exten:  <DialedExten.....>  CLCID:  <ConnectedLineCID.....>
=====

Endpoint:  2000                                Unavailable  0 of inf
  InAuth:  auth2000/2000
  Aor:  2000                                2

Endpoint:  2001                                Unavailable  0 of inf
  InAuth:  auth2001/2001
  Aor:  2001                                2

Endpoint:  2002                                Unavailable  0 of inf
  InAuth:  auth2002/2002
  Aor:  2002                                2

Endpoint:  dcs-endpoint                        Not in use   0 of inf
  Aor:  dcs-aor                                0
  Contact:  dcs-aor/sip:sip.digiumcloud.net  05b40e4cbd NonQual  nan

Objects found: 4

```

Рисунок 2.31 – Команда pjsip show endpoints

2.8 Маршрутизація викликів

Для здійснення дзвінків необхідно внести зміни до файлу `extensions.conf` — основного робочого файлу. У цьому файлі налаштовується маршрутизація викликів для Asterisk. У контексті телефонії, до якого належить і Asterisk, це називається DIALPLAN.

2.8.1 Структура файлу `extensions.conf`

Файл містить дві обов'язкові секції: `general` та `globals`. Ці секції також вказуються в квадратних дужках. Після цих двох обов'язкових секцій розміщуються секції вашого діалплану.

Параметри розділу `[general]`:

- `static` — якщо встановлено значення `no` або параметр не вказано, `pbx_config` перезапише файл при зміні `extensions`. Усі коментарі в цьому файлі будуть втрачені;
- `writeprotect` — якщо параметр `static` має значення `yes`, а `writeprotect` дорівнює `no`, можна зберегти налаштування діалплану через командний рядок Asterisk за допомогою команди «`dialplan save`»;
- `autofallthrough` — якщо параметр має значення `yes` і можливості вашого діалплану вичерпані, дзвінок буде завершено. Це є стандартною поведінкою. Якщо значення буде `no` або параметр не вказано, Asterisk чекатиме дій від користувача;
- `extenpatternmatchnew` — встановивши цей параметр у значення `true` або `yes`, буде використовуватися новий алгоритм, який шукатиме кращі збіги патерну. Швидкість обробки за новим алгоритмом зростає зі збільшенням кількості додаткових номерів, що обробляються. За замовчуванням цей параметр вимкнено;
- `clearglobalvars` — якщо параметр `clearglobalvars` встановлено, глобальні змінні будуть очищені та повторно оброблені при перезавантаженні діалплану або перезапуску Asterisk. Коли параметр не вказано або має значення `no`, глобальні змінні зберігатимуться під час перезавантаження, і навіть при видаленні з `extensions.conf` або одного з файлів, що його включають, використовуватиметься попереднє значення.

Перейдемо до параметрів секції `[globals]`.

В Asterisk можна використовувати як глобальні змінні, так і змінні каналу. Існують також змінні оточення, які забезпечують доступ до змінних оточення Unix з Asterisk.

Глобальні змінні можуть бути визначені в секції `[globals]` або за допомогою команди `SetGlobalVar` у плані набору номерів. Відмінною рисою глобальної змінної

є те, що вона діє для файлу `extensions.conf`. Змінні каналу використовуються лише в межах того діалплану, в якому вони визначені.

У глобальних змінних зручно зберігати довгі додаткові номери, робочий час та інші змінні, які часто використовуються. Ось приклад секції `[globals]` показано на рисунку 2.32.

```
[globals]
MSK_time=08:00-18:00
MSK_week=mon-fri
```

Рисунок 2.32 – Глобальні налаштування робочих днів та часу

У наведеному прикладі показано дві змінні, що вказують час роботи та робочі дні.

Діалплан в Asterisk будується на основі контекстів. У кожному контексті визначається набір додатків із відповідними параметрами. Наприклад, щоб користувачі могли телефонувати на додатковий номер 1000 рисунок 2.33.

```
[from-internal]
exten = 1000,1,Dial(PJSIP/1000)
```

Рисунок 2.33 – Створення контексту

У контексті з назвою `from-internal` вказано, що при дзвінку на додатковий номер 1000 буде здійснено виклик цього номера через інтерфейс PJSIP. Під час створення користувача Asterisk у файлі `pjsip.conf`, у розділі налаштувань `endpoint`, зазначається поле `context=from-internal`. Це означає, що під час виклику даний користувач дотримуватиметься правил, визначених у контексті `[from-internal]` файлу `extensions.conf`.

2.9 Процес виклику

Під час дзвінка абонент шукатиме у своєму контексті правила для набору зазначеного номера. У кожному контексті може бути кілька додаткових номерів, і правила для набраного номера застосовуватимуться послідовно.

Формат запису такий: `exten = ім'я, пріоритет, додаток (параметри)`, показано на рисунку 2.34.

```
[from-internal]
exten = 1000,1,Answer()
    same = n,Playback(demo-congrats)
    same = n,Dial(PJSIP/1000)
```

Рисунок 2.34 – Створення діалплану

Використовуючи `same` замість `exten`, можна уникнути повторного запису додаткового номера 1000 у кожному рядку, але в першому рядку `exten` потрібно вказувати обов'язково.

Системні додаткові номери:

- `s` — якщо дзвінок надходить у контекст без зазначеного додаткового номера, використовується номер `s` (від слова "start");
- `i` — якщо абонент набирає неіснуючий додатковий номер, виклик переходить на додатковий номер `I`;
- `t` — коли абонент довго не натискає кнопку після запуску програми `WaitExten()`, дзвінок переходить на додатковий номер `t`. Час очікування за замовчуванням — 10 секунд;
- `h` — цей додатковий номер використовується при завершенні дзвінка, коли медіаканал уже закрито.

2.10 Шаблони номерів

Для зручності додаткові номери зіставляються за шаблонами. Усі шаблони в діалплані починаються зі знака підкреслення _:

- X — будь-яка цифра від 0 до 9;
- Z — будь-яка цифра від 1 до 9;
- N — будь-яка цифра від 2 до 9;
- . — будь-який символ;
- [184-7] — цифри 1, 8, 4, 5, 6, 7.

Шаблон для дзвінків всередині організації на всі додаткові номери з 4 цифр показано на рисунку 2.35, виглядатиме наступним чином:

```
[from-internal]
exten = _XXXX,1,Dial(PJSIP/${EXTEN})
```

Рисунок 2.35 – Створення шаблону

Де _XXXX — будь-які 4 цифри, \${EXTEN} — змінна, що передає набраний додатковий номер.

2.11 Висновок по розділу

У цьому розділі було описано процес встановлення та налаштування Asterisk на сервері, з акцентом на підготовку системи та встановлення залежностей. Важливим аспектом стало правильне налаштування компонентів, таких як DAHDI та LibPRI, для забезпечення телефонних з'єднань.

Також розглянули клонування Asterisk з репозиторію, підготовку конфігураційних файлів і діалплану, підкреслюючи значення структурованого підходу у конфігурації. Окрему увагу приділили безпеці, зокрема створенню користувача для Asterisk та налаштуванню маршрутизації викликів у файлі `extensions.conf`.

Отже, розділ підкреслив ключові етапи встановлення та налаштування Asterisk для створення ефективної телефонної системи, наголошуючи на важливості ретельної конфігурації для забезпечення безпеки та функціональності.

3 ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Сучасні варіанти дослідження системи

Існує кілька власних рішень для тестування SIP, які мають перевагу у зрозумілості сценаріїв тестування для користувачів. Однак у реальному світі є й недоліки, такі як висока вартість і можливі несумісності результатів, оскільки кожна компанія фокусується на різних основних сферах інтересів. З іншого боку, IETF опублікував кілька проектів, які мають методологію та показники тестування інфраструктури SIP як основну тему (рис. 3.2, 3.3 та 3.4). Ці проекти прагнуть визначити основні терміни для порівняльного аналізу SIP, а також часові вимірювання, які є важливими для отримання відповідних результатів. Оскільки ці чернетки перебувають на ранній стадії розробки, наразі не існує програмного чи апаратного забезпечення для порівняльного аналізу SIP, яке б використовувало ці чернетки.

На шляху до створення відповідного та загальноприйнятого методу тестування існує модель порівняльного аналізу SIP від компанії Transnexus, яка може стати джерелом натхнення (рис. 3.1). Ця компанія розробила ефективний метод для порівняння інфраструктури SIP за допомогою генератора трафіку SIPp з відкритим кодом. Щоб розробити метод, який відображатиме основні концепції

проектів IETF, доцільно вдосконалити процедуру Transnexus. Це дозволить отримати достатньо результатів для визначення ефективності системи, максимальної навантажувальної здатності, а також динамічних характеристик системи.

3.2 Методологія

Для виконання тестування SIP моделюються обидва кінці діалогу SIP, щоб перевірити основну частину інфраструктури SIP — SIP-сервер. SIP-сервер складається з набору серверів, які завжди включають реєстратор SIP та SIP-проксі або B2BUA (Back to Back User Agent). Останнє є найпоширенішим рішенням у корпоративному середовищі, як для малих і середніх підприємств, так і для великих підприємств. На рисунку 3.1 представлено його базову конфігурацію тестового обладнання.

Ця загальна конфігурація не відображає деякі апаратні та програмні обмеження, проте чудово описує два важливі елементи тестування. Перший — це спеціальний комп'ютер для тестування RTP-потоків, який дозволяє використовувати більш складні інструменти для захоплення мережевого трафіку, не впливаючи на взаємодію RTP і SIP у тестах.

Друга ідея полягає в тому, що всі комп'ютери повинні бути з'єднані один з одним лише за допомогою одного комутатора. Це забезпечує можливість відтворення тестування. Кожне вимірювання для окремого випадку перекладу кодека складається з кількох етапів. Кожен етап триває 16 хвилин, протягом яких генеруються 60-секундні дзвінки з визначеною частотою викликів. Після цього слідує 60-секундна пауза, протягом якої незавершені дзвінки припиняються. Цей процес повторюється для кожного конкретного етапу швидкості виклику. Кожен дзвінок включає стандартний діалог SIP та медіа RTP. Оскільки навантаження не є постійним, а повільно збільшується на початку тесту (протягом перших 60 секунд)

і зменшується в кінці (останні 60 секунд), результати, отримані після цього початкового періоду та перед закінченням, є єдиними, які вважатимуться дійсними. Результати фіксуються в двох місцях: за допомогою елементів і вибраних параметрів. На рисунку 3.2 детально показано значення затримок RRD і SRD.

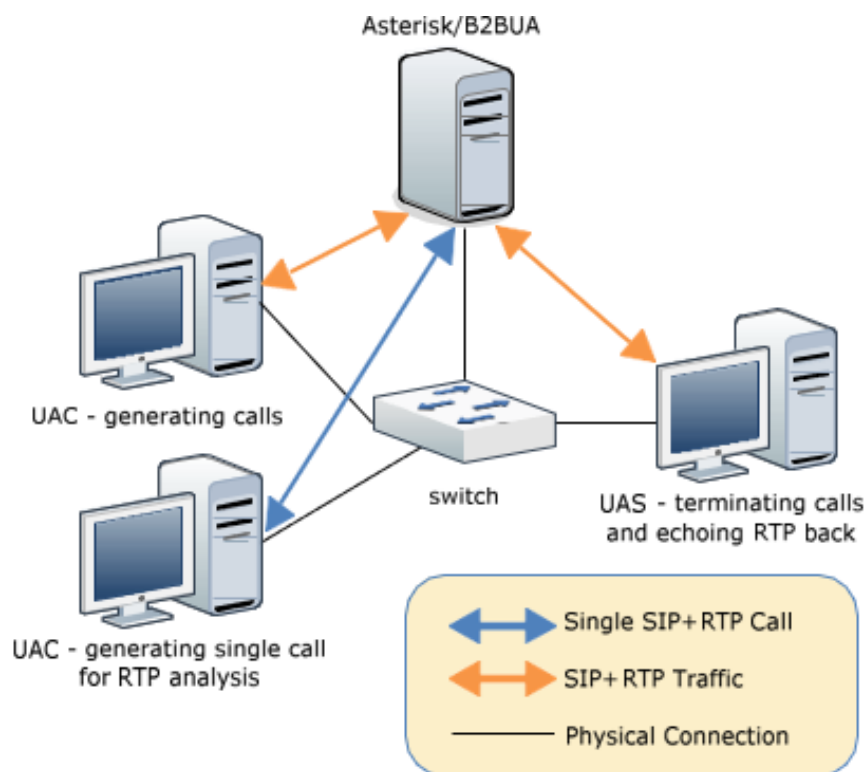


Рисунок 3.1 – Спрощена конфігурація тестової системи

Елементи:

- SIP-сервер: тут вимірювалися навантаження на CPU, використання пам'яті та обсяг мережевого трафіку;
- UAC: у цьому місці фіксувалася кількість (не) успішних викликів, тривалість обміну повідомленнями, перерозподіли викликів, а також відбиралися вибірки RTP для аналізу (на окремому UAC).

Вимірювані параметри:

- завантаження CPU;
- кількість (не) успішних дзвінків (для контролю завершення тесту);

- затримка запиту на реєстрацію (RRD) — час між першим запитом реєстрації та відповіддю 200 OK зображено на рисунку 3.2;
- затримка запиту сеансу (SRD) — час між першим запитом Invite та повідомленням 180 дзвінка показано на рисунку 3.2;
- середній джиттер та максимальна затримка пакету RTP.

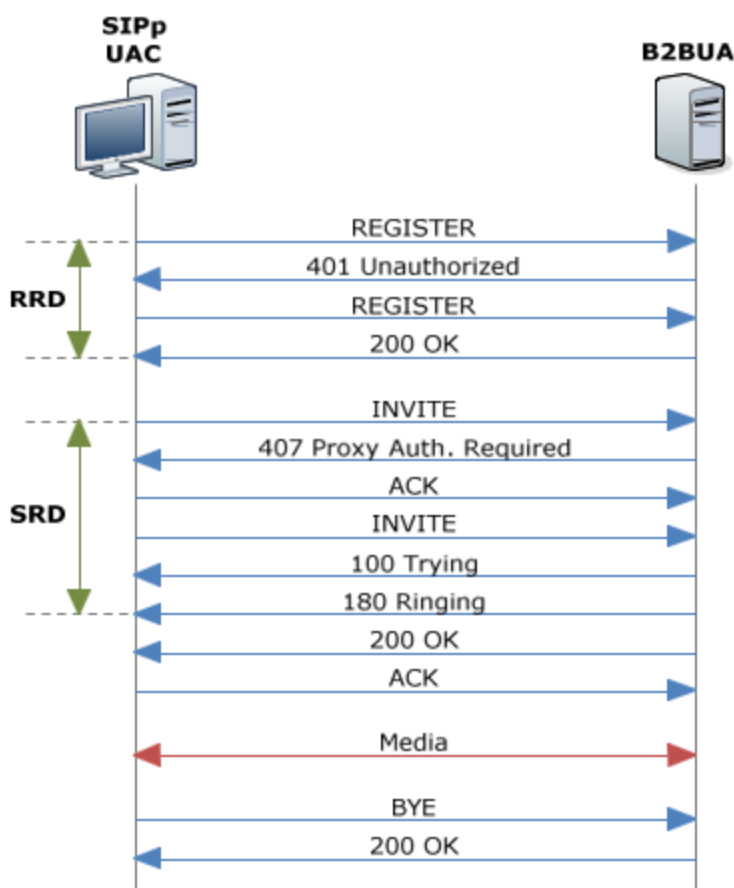


Рисунок 3.2 – Затримка запиту реєстрації та затримка запиту сеансу у діалоговому вікні SIP

Оскільки увага була зосереджена на тестуванні ефективності та швидкості трансляції кодеків, вдалося визначити максимальне навантаження, яке сервер SIP міг обробляти з точки зору протоколів SIP або RTP. Однак ці результати стосувалися лише конкретної машини або платформи, тому додавали ще один етап до аналізу даних. Та сама процедура тестування, описана вище, виконувалася на машині, налаштованій для пропуску медіа лише через SIP-сервер. Результати цього

тесту слугували основою для порівняння всіх інших результатів. Відношення виражалося як коефіцієнт продуктивності. Коефіцієнт оцінки продуктивності (PRF) визначався як відношення кількості викликів із трансляцією кодека (PCT) до загальної продуктивності (P) без перекодування.

$$P_{RF} = \frac{P_{CT}}{P} \times 100 \quad (3.1)$$

Цей крок дозволяв порівнювати результати незалежно від апаратного забезпечення та платформи.

3.3 Тестування

Для моделювання як UAC, так і UAS, планувалося використовувати інструмент для тестування продуктивності SIP під назвою SIPr. Ця утиліта з відкритим кодом здатна моделювати одночасні SIP-дзвінки, а також дозволяла вимірювати ключові часові показники, зокрема ті, що визначені у проєктах IETF. SIPr виконував виклики на основі користувацьких сценаріїв, описаних мовою XML. Ці XML-сценарії поширювалися на кожному комп'ютері, а запуск SIPr виконувався через сценарій bash і SSH. Один з комп'ютерів працював як клієнт SSH і керував усім тестом, надсилаючи команди іншим комп'ютерам (SSH-серверам) через SSH. Потоки виклику повідомлень, які обмінювалися між пов'язаними екземплярами UAC та UAS SIPr, були показані рисунку 3.3.

Ключові показники використання обладнання на B2BUA вимірювалися за допомогою системного звіту про активність (SAR) кожні 10 секунд протягом 60 вимірювань, тобто в середньому за 10 хвилин тесту, коли кількість одночасних дзвінків залишалася сталою. Носій складався з 60-секундної музичної композиції, записаної у файл G711u рсар, який використовувався UAC. UAS налаштовувалися на використання кодеків G711u-law, G711A-law, G726-32 і GSM. Asterisk PBX

здійснював трансляцію кодеків. Потоки RTP можна було захоплювати та аналізувати за допомогою Wireshark, який надавав складні засоби для аналізу RTP. Однак генерація потоків RTP на стороні клієнта вимагала значних ресурсів CPU, що означало необхідність обмежити кількість викликів, генерованих однією машиною. Це призвело до збільшення кількості ПК, на яких працювала схема UAS.

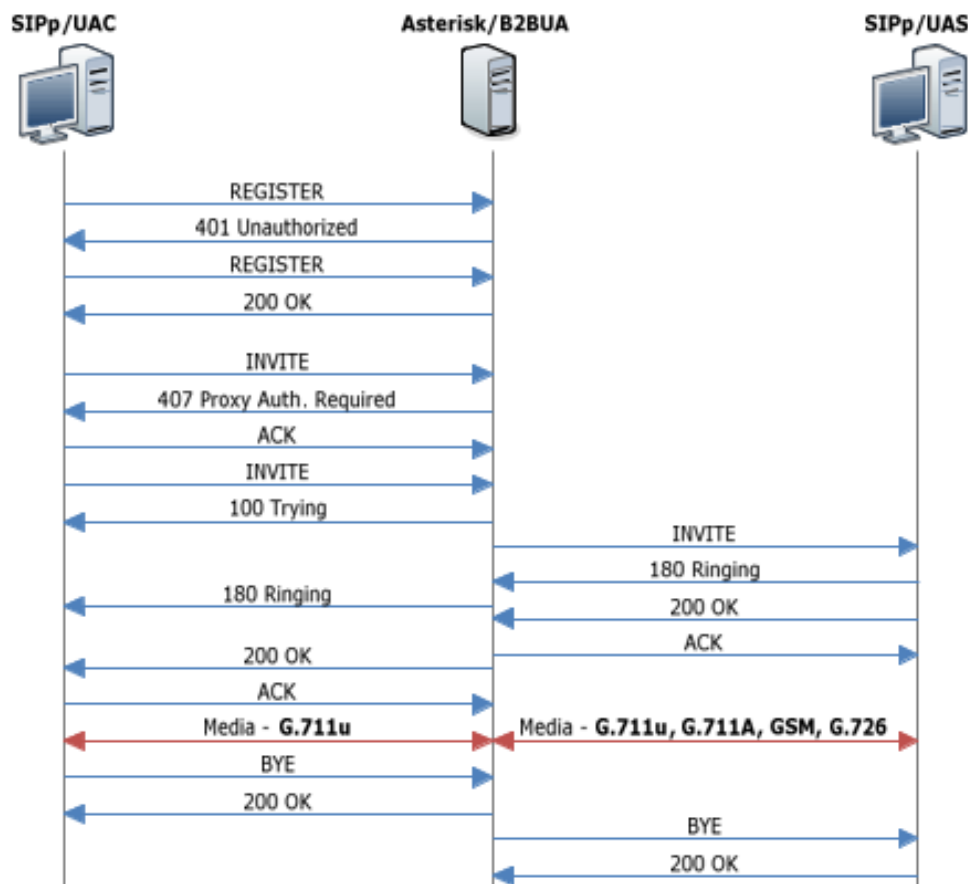


Рисунок 3.3 – Потік SIP повідомлень що проходять через B2BUA

Загальну кількість комп'ютерів можна визначити на основі оціненого максимального навантаження на сервер SIP. Оскільки в даному випадку SIP-сервер – це ПК з чотирьохядерним процесором, загальна кількість одночасних викликів не перевищить 2000. Кожен комп'ютер із такою апаратною конфігурацією може генерувати близько 400 викликів. Тому кількість клієнтів має становити не менше чотирьох. У нашому випадку для проведення тесту достатньо чотирьох. Сервери

здатні витримувати подвоєне навантаження, тому їх буде всього два. Апаратна та програмна конфігурація зазначени у таблиці 1.

Таблиця 1 – Апаратна та програмна конфігурація

Конфігурація	Процесор	Оперативна пам'ять	Операційна система	Програмне забезпечення
UAC	Core 2 Duo E7200	4 ГБ DDR	Ubuntu 20.04 x64	—
UAS	Core 2 Duo E7200	2 ГБ DDR	Ubuntu 20.04 x64	—
B2BUA	Core 2 Quad Q6600	4 ГБ DDR3	Ubuntu 20.04 x64	Asterisk версії 18

Пристрої підключалися до комутатора, а всі ПК – до портів швидкого Ethernet. Цього було достатньо, оскільки навантаження на трафік розподілялося, проте B2BUA використовував гігабітний порт комутатора [10]. Топологія мережі була зображена на рисунку 3.4.

Увесь процес тестування продуктивності вимагав кілька комп'ютерів для генерування SIP-трафіку. Щоб успішно виконати тест, весь процес повинен був бути автоматизованим. Тому всі комп'ютери отримували команди від головного UAC через SSH за допомогою набору сценаріїв bash. На головному UAC для виконання цього завдання викликався сценарій bash. На першому етапі головний UAC підраховував кількість викликів, які кожен комп'ютер повинен генерувати за одну секунду. Потім він наказував UAS зареєструватися та почати прослуховування на порту UDP 5060, що також виконувалося за допомогою сценарію bash. По-друге, SIPp на всіх UAC викликався для створення трафіку. Останнім кроком викликався sar через 2,5 хвилини, щоб впевнитись, що стабільне навантаження вже досягнуто. Результати містили статистику використання ЦП, пам'яті та мережі та зберігалися у файлі data_callrate.sar у двійковому форматі.

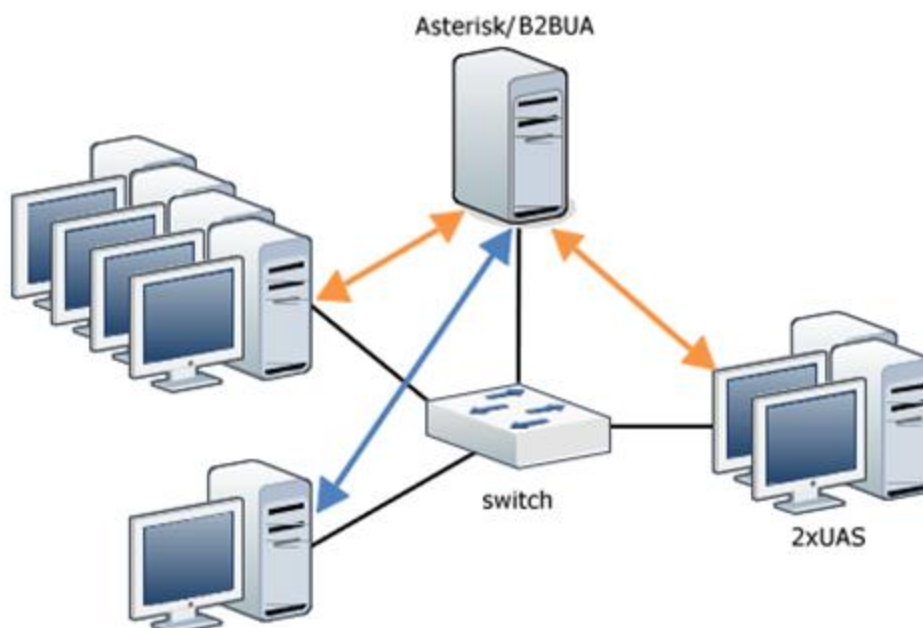


Рисунок 3.4 – Топологія тестової системи

3.4 Результат

Для кожної категорії представлені дві різні діаграми. Перша діаграма показувала результати для випадку без трансляції кодеків і була пофарбована в синій колір. Друга діаграма демонструвала нормовані значення для випадків із трансляцією кодека, пофарбовані у три різні кольори.

Перша діаграма ілюструвала просту залежність між кількістю одночасних викликів, що проходять через B2BUA, і використанням його ЦП. Пік, пов'язаний із 600 дзвінками, ймовірно, був спричинений коефіцієнтом питомого поглинання, який періодично вимірював дані. Таким чином, можливо, що він брав вибірки з меншою функцією використання ЦП, яка не є постійною, але періодично змінюється.

Друга діаграма показувала, що, як і очікувалося, трансляція кодеків з G711u на G711A споживала приблизно на 24% більше потужності процесора, ніж простий

G711u без перекодування. З іншого боку, найбільш вимогливим виявився кодек G726-32bit. Проте найменше навантаження давало найцікавішу інформацію: при завантаженні в 120 дзвінків різниця в споживанні потужності процесора для GSM і G726 була найбільшою в порівнянні з відсутністю трансляції кодеків. При більших навантаженнях ця різниця починала швидко зменшуватися.

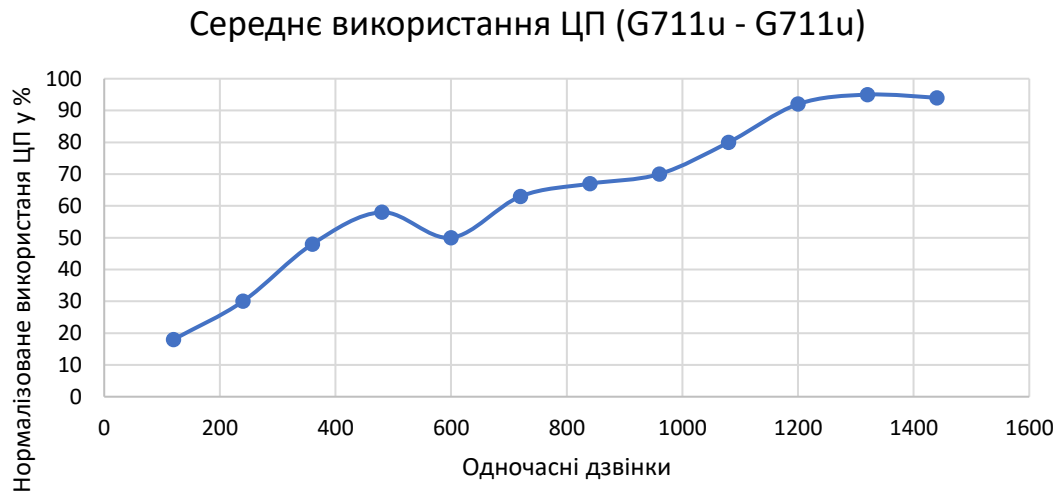


Рисунок 3.5 – Середнє використання ЦП без перекодування

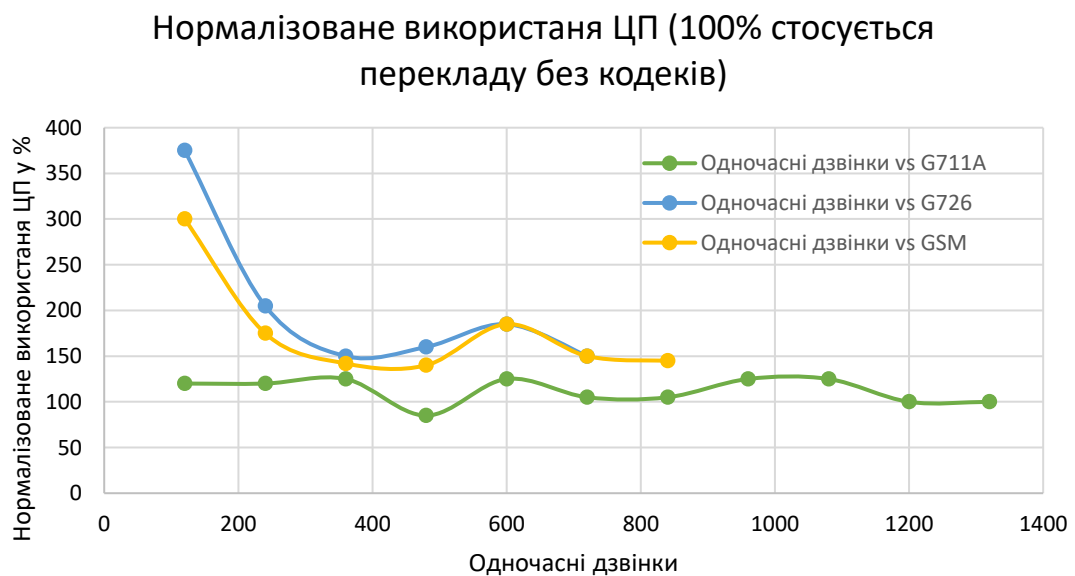


Рисунок 3.6 – Середнє використання ЦП та відповідні нормовані значення

Діаграми на рисунках 3.7 та 3.8 наочно демонструють, що дзвінок встановлювався ще швидше, коли використовувався переклад кодека і навантаження становило менше 240 одночасних викликів. Проте, коли використання ЦП збільшувалося, затримки ставали значно тривалими. Останнє значення G711a для обох діаграм було настільки низьким через швидке збільшення затримок у випадках G711u до G711u між 1200-1320 одночасними викликами [11].

Нормовані значення середнього джиттера та максимальної затримки пакетів підтвердили очікуваний результат, оскільки значення, пов'язані з невеликим навантаженням, були дуже схожими на основні значення у випадку без трансляції кодеків. Дуже швидке зниження обох нормованих значень для G711A було викликане збільшенням основних значень у випадку без перекодування та значною кількістю невдалих викликів у цьому сценарії. На рисунку 3.9 було показано середній джиттер і максимальну затримку пакетів RTP. Рисунок 3.10 демонстрував середній джиттер разом із відповідними нормованими значеннями. На рисунку 3.11 було відображено максимальну затримку пакету RTP та її нормовані значення.

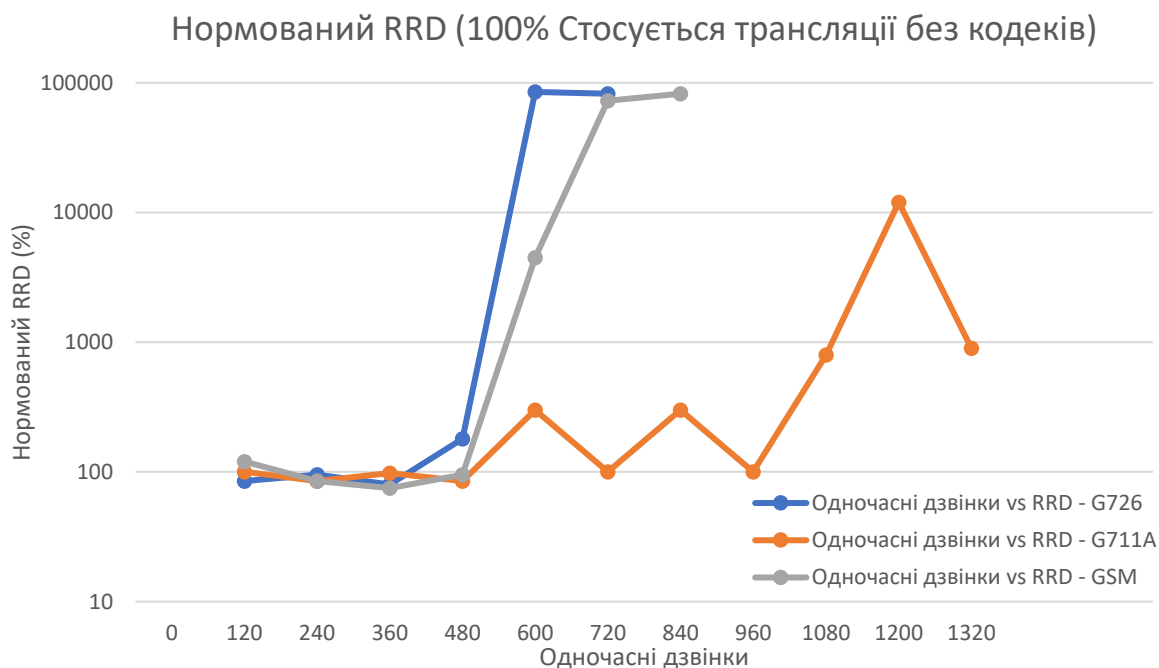


Рисунок 3.7 – RRD та відповідні нормовані значення

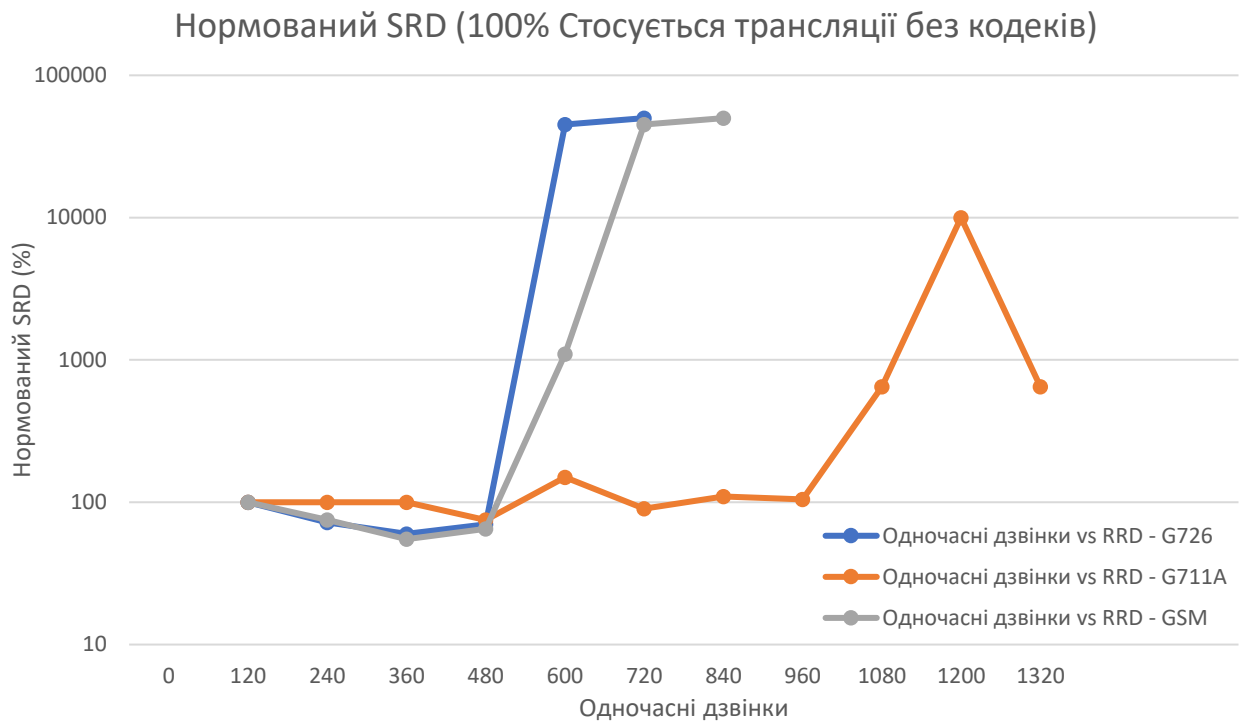


Рисунок 3.8 – SRD та відповідні нормовані значення

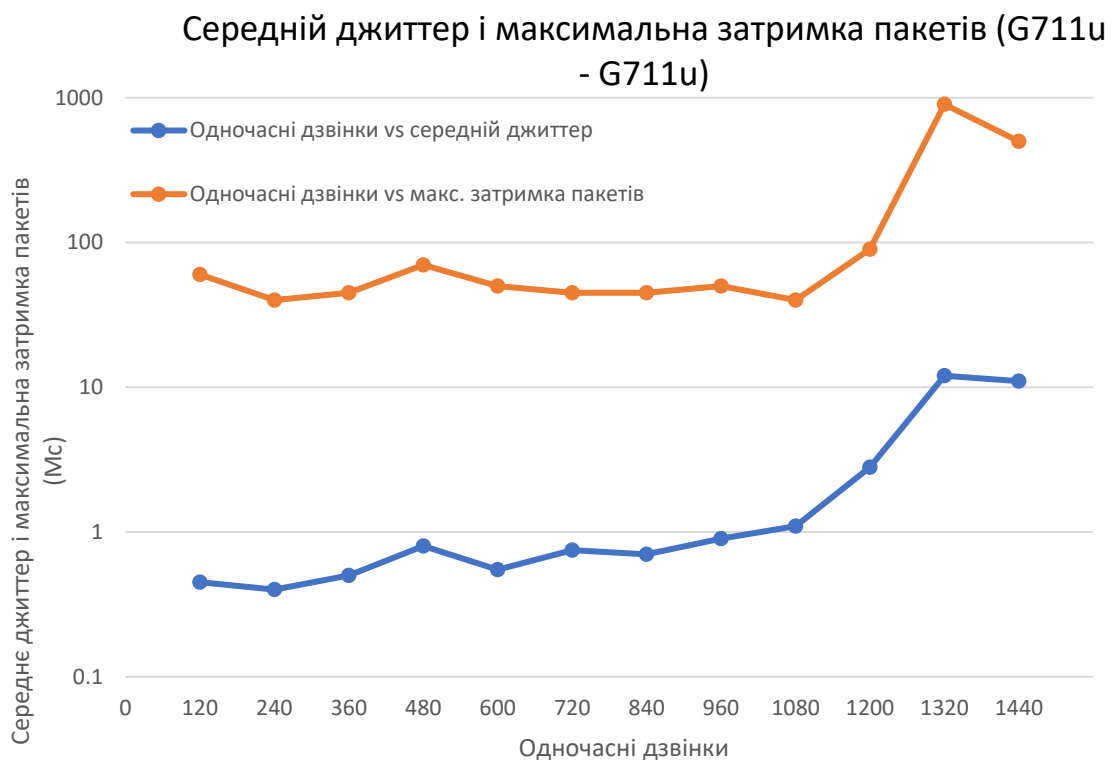


Рисунок 3.9 – Середній джиттер і максимальна затримка пакетів RTP

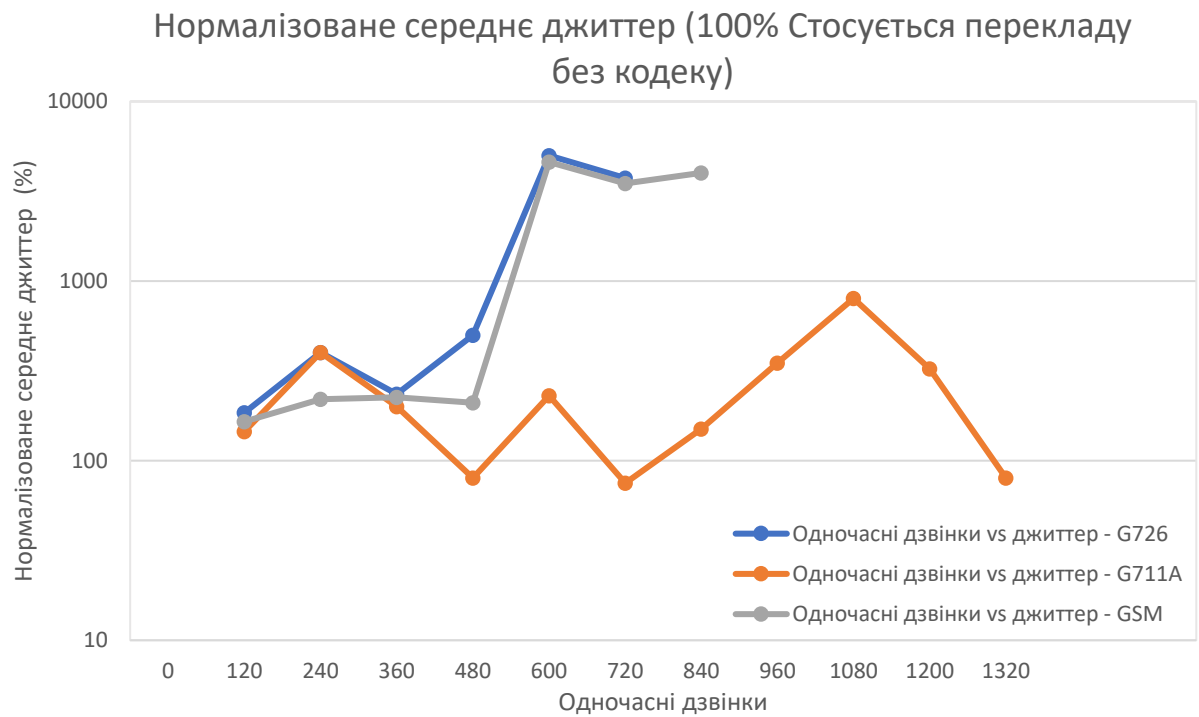


Рисунок 3.10 – Середній джиттер та відповідні нормовані значення

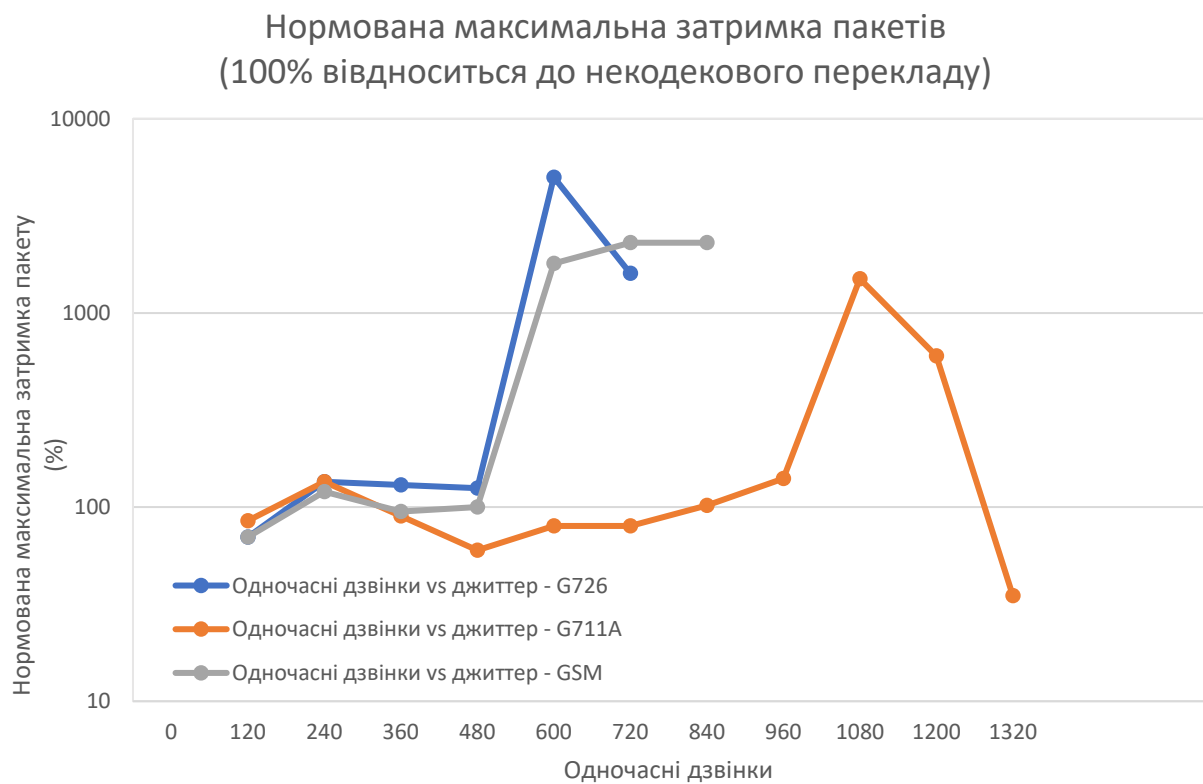


Рисунок 3.11 – Максимальна затримка пакету RTP та відповідні нормовані значення

Метод тестування та порівняльного аналізу інфраструктури SIP був призначений для порівняльного аналізу VoIP – інфраструктури на основі SIP. Він дозволяв визначити максимальне навантаження на систему та демонстрував динамічно змінювані характеристики системи, такі як час відгуку та затримка пакетів.

3.5 Висновок по розділу

У цьому розділі було проведено аналіз сучасних методів тестування SIP, підкреслюючи їх переваги та недоліки. Незважаючи на зрозумілі сценарії, практичні проблеми, такі як висока вартість і несумісність результатів, залишаються актуальними. Методологія тестування базувалася на моделюванні обох кінців діалогу, що дозволило виявити ключові параметри продуктивності SIP-сервера, зокрема навантаження на CPU і затримки.

Використання інструменту SIPr для моделювання одночасних дзвінків дало цінні дані про продуктивність залежно від різних кодеків. Результати показали, що трансляція кодеків збільшувала навантаження на процесор і впливала на затримки. Дослідження визначило максимальне навантаження системи та продемонструвало динаміку змін характеристик, що є важливим для оптимізації VoIP-інфраструктури.

ВИСНОВКИ

У магістерській роботі було розроблено, налаштовано та протестовано комп'ютерну систему телефонії на базі Asterisk, що дозволило отримати розгорнутий огляд ключових аспектів IP-телефонії та можливостей сучасних систем VoIP. У рамках дослідження було проведено ознайомлення з основними поняттями й термінологією, розглянуто переваги й недоліки Asterisk у порівнянні з іншими альтернативами, що дало можливість обрати цю платформу як основну для реалізації системи телефонії. Було визначено необхідний функціонал та здійснено детальну настройку й адаптацію системи, що забезпечило виконання специфічних вимог для практичного використання.

Особлива увага була приділена аналізу конфігураційних файлів Asterisk, які дозволяють гнучко налаштовувати параметри системи для забезпечення оптимальної продуктивності й надійності зв'язку. Проведене тестування працездатності системи з використанням різних аудіокодеків, таких як G711u-law, G711A-law, G726-32 та GSM, дало можливість оцінити вплив кодеків на навантаження системи. Зокрема, було виявлено, що перекодування з G711u на G711A збільшує навантаження на процесор на 24% у порівнянні з використанням G711u без перекодування. Найбільш ресурсомістким виявився кодек G726-32, тоді як GSM створює найменше навантаження на процесор.

При моделюванні сценарію з 120 одночасними дзвінками було зафіксовано значну різницю в навантаженні між кодеками GSM і G726, що дозволяє зробити висновок про суттєвий вплив вибору кодека на продуктивність системи. Це підтверджує, що Asterisk здатний забезпечити заявлену функціональність і стабільну роботу навіть при високих навантаженнях, а також дозволяє ефективно адаптувати систему до вимог продуктивності завдяки вибору відповідних кодеків. Таким чином, результати дослідження в рамках магістерської роботи можуть бути використані як практичні рекомендації для оптимізації VoIP-системи Asterisk та її подальшого застосування в різних середовищах телефонії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Джим В. М. Астеріск. Майбутнє телефонії [Текст] / Джим Ван Меггелен, Ліф Мадсен, Джаред Сміт - М.: Символ-Плюс, 2015. - 656 с.
2. Що таке VoIP - енциклопедія lanmarket.ua. Lanmarket. URL: <https://lanmarket.ua/ua/entsiklopediya/telekommunikatsionnye-tekhnologii/voip.html>.
3. Посібник із впровадження IP-АТС. А-КОМ. URL: https://a-kom.ua/files/product_1359748/instr_p1359748_2.pdf.
4. Що таке SIP-телефонія, і навіщо вона потрібна?. Блог Ringostat. URL: <https://blog.ringostat.com/uk/shcho-take-sip-telefonii-i-navishcho-vona-potribna-biznesu>.
5. IP телефонія для кол-центру: переваги та застосування. VoIPTime. URL: <https://www.voiptime.net/uk/ip-telephony-for-call-center.html>.
6. Terratel author. How to configure IP PBX Asterisk from scratch | TERRATEL | Page 1000. TERRATEL. URL: <https://www.terratel.eu/how-to-configure-ip-pbx-asterisk-from-scratch.html/1000>.
7. PJSIP-pjproject - asterisk documentation. Home - Asterisk Documentation. URL: <https://docs.asterisk.org/Getting-Started/Installing-Asterisk/Installing-Asterisk-From-Source/Prerequisites/PJSIP-pjproject/>.
8. Terratel author. How to configure IP PBX Asterisk from scratch | TERRATEL. TERRATEL | Solutions for VoIP and TDM networks integration. URL: <https://www.terratel.eu/how-to-configure-ip-pbx-asterisk-from-scratch.html>.
9. Creating a basic IP PBX with asterisk. Open Source For You. URL: <https://www.opensourceforu.com/2014/11/creating-a-basic-ip-pbx-with-asterisk/>.
10. Asterisk definitive guide. LanSystems. URL: https://www.lansys.com.ua/wpcontent/uploads/2017/10/Asterisk_Definitive_Guide_en.pdf.
11. The ultimate guide to IP PBX and voip systems. the best free IP pbxs for businesses. Cisco Networking, VPN Security, Routing, Catalyst-Nexus Switching,

Virtualization Hyper-V, Network Monitoring, Windows Server, CallManager, Free Cisco Lab, Linux Tutorials, Protocol Analysis, CCNA, CCNP, CCIE. URL: <https://www.firewall.cx/tools-tips-reviews/ip-pbx-unified-communications/how-ip-pbx-work-best-free-voip-systems-advanced-unified-communications.html?highlight=WzFd>.

12. Home ★ asterisk. Asterisk. URL: <https://www.asterisk.org/>.
13. Meggelen J. V. Asterisk. 2nd ed. Sebastopol : O'Reilly Media, Inc., 2008. 602 p.
14. PBX systems for IP telephony. New York : McGraw-Hill, 2002. 487 p.
15. Казанський. Інтернет-конференції НУБіП України. URL: <http://econference.nubip.edu.ua/index.php/grpi/grpi24/paper/view/3554>.