

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

ДОСЛІДЖЕННЯ ІР ТЕЛЕФОНІЇ НА ОСНОВІ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ ASTERISK

Здобувача вищої освіти другого (магістерського) рівня, II курсу, КМІЗ-22-1 групи,
спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»


(підпис)

Топтунова Дениса Миколайовича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Грищенко Світлана Миколаївна, к.пед.н., ст. досл.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної
програми


(підпис)

Горбовий Артур Юліанович, д.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри кафедра комп'ютерних та
інформаційних технологій і систем від «28» грудня 2023 № 8

Завідувача
кафедри


(підпис)

Філоненко Михайло Миколайович, к.фіз.-мат.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

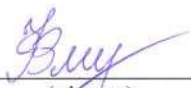
Ірпінь – 2024

Реєстрація № 15 30.01.2024
(дата)


(підпис відповідальної особи.)

Результат перевірки 90 захисту 30.01.2024 
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 5 (відмінно) 09.02.2024 
(оцінка) (дата) (підпис)

Голова комісії: 
(підпис)

В.М. Франчук
(ініціали, прізвище)

Члени комісії:


(підпис)

А.Ю. Горбовий
(ініціали, прізвище)


(підпис)

О.В. Редич
(ініціали, прізвище)


(підпис)

В.О. Ніжегородцев
(ініціали, прізвище)

БЛАНК ОЦІНЮВАННЯ ДИПЛОМНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		09.02.2024 Франчук В.М. <i>В.М.</i> Горбовий А.Ю. <i>А.Ю.</i>
2. Дотримання вимог щодо оформлення	від 0 до 10	9		Редич О.В. <i>О.В.</i>
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	5		Ніжегородцев В.О. <i>В.О.</i> Суховерхівка Л.П. <i>Л.П.</i>
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	47		

Оцінювання кваліфікаційної роботи науковим керівником - 43 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
90	A	5 (відмінно)	09.02.2024	Франчук В.М. <i>В.М.</i> Горбовий А.Ю. <i>А.Ю.</i> Редич О.В. <i>О.В.</i> Ніжегородцев В.О. <i>В.О.</i> Суховерхівка Л.П. <i>Л.П.</i>

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
 Кафедра комп'ютерних та інформаційних технологій і систем
 Спеціальність 122 «Комп'ютерні науки»
 Освітня програма «Інформаційні управляючі системи і технології в економіці»

Затверджую

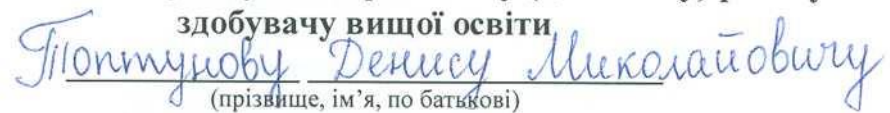
Завідувач кафедри

 Філоненко М.М.

« 04 » 04 2023

Завдання на випускню кваліфікаційну (дипломну) роботу

здобувачу вищої освіти

 Гонтюкову Денису Миколайовичу
 (прізвище, ім'я, по батькові)

- Тема випускної кваліфікаційної (дипломної) роботи
Дослідження IP телефонії на основі програмного забезпечення Asterisk
 Затверджена наказом ректора від «04» квітня 2023 № 362
- Строк здачі закінченої роботи «20» грудня 2023
- Цільова установка застосування набутих знань, умінь в дослідженні IP телефонії на основі програмного забезпечення Asterisk
 Мета роботи Вивчити та проаналізувати можливості та характеристики IP телефонії з використанням програмного забезпечення Asterisk з метою покращення комунікаційних процесів у сучасних організаціях.
 Об'єкт дослідження IP телефонія на основі програми Asterisk.
 Предмет дослідження Технологічні можливості, функціональні характеристики та ефективність впровадження IP телефонії з використанням програмного забезпечення Asterisk.
 Орієнтовна база дослідження: ТОВ Infopulse.
- Вихідні дані до роботи технологічні рішення дослідження трафіку мультисервісних мереж.
- Перелік графічного матеріалу – 38 рисунків.
- Зміст випускної кваліфікаційної (дипломної) роботи

ЗМІСТ

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ IP ТЕЛЕФОНІЇ

1.1. Історія та основні поняття IP телефонії

1.2 Організація та використання мережі IP телефонії.

РОЗДІЛ 2. ВПРОВАДЖЕННЯ МЕТОДУ ДОСЛІДЖЕННЯ IP ТЕЛЕФОНІЇ

НА ОСНОВІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ASTERISK

2.1. Програмні можливості та функціонал платформи Asterisk

2.2. Переваги та недоліки Asterisk

РОЗДІЛ 3. УСУНЕННЯ НЕСПРАВНОСТЕЙ В IP ТЕЛЕФОНІЇ

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів р	
		За планом	За фактс
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 27.12.2022	15.11.
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26.06.2023	01.02.
3	II розділ (аналіз експериментальної частини про методики досліджень)	23.10.2023	04.04.
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	30.11.2023	09.05.
5	Висновки	10.12.2022	15.05.
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	20.12.2023	26.05.
7	Отримання відгуку керівника	29.01.2024	29.01.
8	Отримання зовнішньої рецензії	01.02.2024	01.02.

8. Дата видачі завдання «04» квітня 2023

9. Керівник випускної кваліфікаційної
(дипломної) роботи


(підпис)

Грищенко С.М.

(прізвище, ініціали)

10. Керівник освітньо-
професійної програми


(підпис)

Горбовий А.Ю.

(прізвище, ініціали)

11. Завдання до виконання отримав
здобувач вищої освіти


(підпис)

Топтунов Д.М.

(прізвище, ініціали)

АНОТАЦІЯ

Дослідження IP телефонії на основі програмного забезпечення Asterisk

Мета роботи полягає в вивченні можливостей та ефективності IP телефонії на основі програмного забезпечення Asterisk.

Завданнями роботи є:

- Дослідити організаційні аспекти мережі IP-телефонії.
- Детально проаналізувати програмні можливості та функціонал платформи Asterisk для забезпечення IP телефонії.
- Розробити та впровадити метод дослідження IP телефонії на базі програмного забезпечення Asterisk.
- Дослідити та проаналізувати несправності в мережах IP телефонії.
- Оцінити ефективність та результативність впроваджених методів усунення несправностей та управління трафіком в IP телефонії.

Предметом дослідження є ефективність впровадження IP телефонії з використанням програмного забезпечення Asterisk.

Об'єктом дослідження виступає IP телефонія на основі програмного забезпечення Asterisk.

Базою дослідження виступає ТОВ Infopulse.

За результатами дослідження сформульовані методика та етапи впровадження IP телефонії з використовуючи програмне забезпечення Asterisk. *Одержані результати можуть бути використані* на базі практики та на будь-якому підприємстві, на меті є дослідження функціональних характеристик та ефективність впровадження IP телефонії з використанням програмного забезпечення Asterisk.

Кваліфікаційна робота містить 69 сторінок, 38 рисунків, список використаних джерел 40 найменувань, 2 додатка.

Ключові слова: IP-телефонія, програмне забезпечення, Asterisk.

Рік виконання кваліфікаційної роботи 2023

Рік захисту роботи 2024

ABSTRACT

Research on IP Telephony Based on Asterisk Software

The aim of the work is to study the capabilities and effectiveness of IP telephony based on Asterisk software.

The tasks of the work include:

- Investigate organizational aspects of IP telephony networks.
- Analyze in detail the software capabilities and functionality of the Asterisk platform for providing IP telephony.
- Develop and implement a method for researching IP telephony based on Asterisk software.
- Investigate and analyze faults in IP telephony networks.
- Evaluate the effectiveness and performance of implemented methods for troubleshooting and traffic management in IP telephony.

The subject of the research is the effectiveness of implementing IP telephony using Asterisk software.

The object of the research is IP telephony based on Asterisk software.

The research is conducted at the Infopulse LLC.

Based on the research results, a methodology and stages for implementing IP telephony using Asterisk software are formulated.

The obtained results can be applied in practice and in any enterprise, aiming to explore the functional characteristics and effectiveness of implementing IP telephony using Asterisk software.

The qualification work consists of 69 pages, 38 figures, a list of 40 references, and 2 appendix.

Keywords: IP telephony, software, Asterisk.

Year of qualification work: 2023

Year of work defense: 2024

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІР ТЕЛЕФОНІЇ.....	11
1.1. Історія та основні поняття ІР телефонії	11
1.2 Програмні можливості та функціонал платформи Asterisk	13
РОЗДІЛ 2. ВПРОВАДЖЕННЯ МЕТОДУ ДОСЛІДЖЕННЯ ІР ТЕЛЕФОНІЇ НА ОСНОВІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ASTERISK	26
2.1. Організація та використання мережі ІР-телефонії на виробництві ТОВ Infopulse	26
2.2 Переваги та недоліки Asterisk.....	42
РОЗДІЛ 3. УСУНЕННЯ НЕСПРАВНОСТЕЙ В ІР ТЕЛЕФОНІЇ	48
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ	59

ВСТУП

Актуальність теми. З інтенсивним розвитком інформаційних технологій зростає потреба у надійному зв'язку, який може виступати альтернативою традиційним аналоговим методам передачі даних. Для сфери інтернет-бізнесу основною метою стає взаємодія з потенційними клієнтами, популяризація свого продукту та переконання в його цінності. Такий зв'язок повинен бути забезпечений надійною безпекою, легкістю налаштування та підключення, забезпечувати високу якість передачі даних, бути економічно вигідним та масштабованим.

ІР-телефонія об'єднує переваги мережі Інтернет та традиційної телефонії. За допомогою цієї технології інтернет-ресурси можуть продовжити залучати користувачів на свої сторінки та пропонувати їм найкращі умови використання своїх продуктів чи послуг. Крім того, вони можуть використовувати функціональні можливості АТС, такі як паркування виклику, переадресації, автовідповідачі, голосові пошти, передзвони та багато іншого.

Мережі на основі протоколу ІР отримали в наш час дуже широкий розвиток. Кількість людей, які використовують ІР-телефонію, зростає неймовірними темпами, і актуальним завданням є розширення можливостей ІР-телефонії, щоб використовувати їх разом із передачею даних також для відеоконференцій, для передачі голосу та для інших програм в реальному часі.

Дослідженням ІР-телефонії на основі програмного забезпечення Asterisk займалися наступні науковці Баричев С. Г. [8], Гольдштейн Б. С. [7], Гончаров В. В. [8], Грищенко С. М. [17], Пінчук А. В. [9], Поляков М. О. [24], Серов Р. Е. [8], Суховицький А. Л. [7], Токар Л. О. [24], Федотова-Півень І. М. [20] та інші.

Метою кваліфікаційної роботи є вивчення можливостей та ефективності ІР телефонії на основі програмного забезпечення Asterisk, зокрема з точки зору якості зв'язку та функціональності.

Для досягнення мети необхідно вирішити такі *завдання*, а саме:

- Розглянути історію розвитку IP телефонії та визначити основні поняття, що лежать в основі цієї технології.
- Дослідити організаційні та функціональні аспекти мережі IP-телефонії з акцентом на використання програмного забезпечення Asterisk.
- Детально проаналізувати програмні можливості та функціонал платформи Asterisk для забезпечення IP телефонії, включаючи розгляд можливостей налаштування та управління.
- Визначити переваги та недоліки використання програмного забезпечення Asterisk для IP телефонії, зосереджуючись на аспектах продуктивності, масштабованості та надійності.
- Розробити та впровадити метод дослідження IP телефонії на базі програмного забезпечення Asterisk, включаючи аналіз параметрів якості обслуговування та трафіку мультисервісних мереж.
- Дослідити та проаналізувати можливості усунення несправностей в мультисервісних мережах IP телефонії з використанням методів, заснованих на програмному забезпеченні Asterisk.
- Здійснити практичне впровадження запропонованих методів управління трафіком та усуненням несправностей на базі програмного забезпечення Asterisk у реальному корпоративному середовищі.
- Оцінити ефективність та результативність впроваджених методів усунення несправностей та управління трафіком у контексті покращення якості обслуговування в IP телефонії.

Об'єкт дослідження – IP телефонія на основі програмного забезпечення Asterisk.

База дослідження – ТОВ Infopulse.

Предмет дослідження – дослідження функціональних характеристик та ефективність впровадження IP телефонії з використанням програмного забезпечення Asterisk.

.

Методи дослідження – теоретичні – аналіз, узагальнення, систематизація законодавчої бази, матеріалів конференцій, Інтернет-ресурсів для виділення теоретичних засад дослідження; *емпіричні* – аналіз досвіду роботи в даній галузі, для констатування стану розв’язання проблеми.

Науково-практична важливість отриманих результатів полягає в дослідження IP телефонії на основі програмного забезпечення Asterisk мають вагомий науковий та практичний вплив. Робота розкриває можливості використання сучасних інформаційних технологій у сфері зв'язку, що дозволяє підвищити якість обслуговування та ефективність комунікаційних процесів. Особлива увага приділена дослідженню функціональних можливостей платформи Asterisk та її впливу на якість зв'язку.

Практичне значення отриманих результатів. Отримані в ході дослідження результати мають практичне застосування в сучасних бізнес-середовищах та організаціях. Впровадження рекомендацій щодо використання IP телефонії на основі програмного забезпечення Asterisk може сприяти покращенню зв'язку та забезпечити більш ефективну взаємодію з клієнтами. Крім того, виявлені переваги та недоліки використання Asterisk дозволяють здійснювати обґрунтовані вибори технологічних рішень у відповідності до потреб організації. Результати дослідження можуть служити підґрунтям для розробки та впровадження нових стратегій забезпечення якості зв'язку, а також вдосконалення технічної підтримки та обслуговування користувачів. В цілому, отримані результати сприяють підвищенню конкурентоспроможності та ефективності діяльності організацій, що використовують IP телефонію на основі програмного забезпечення Asterisk.

Апробація результатів роботи. Основні результати дослідження висвітлено в матеріалах збірника тез Міжнародного конгресу (м. Ірпінь, 27 квітня 2023 р.).

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ IP ТЕЛЕФОНІЇ

1.1. Історія та основні поняття IP телефонії

Технологія передачі голосу в цифровому форматі через мережу Інтернет або іншу IP-мережу називається IP-телефонією.

Для забезпечення стандартного телефонного зв'язку в мережі IP використовуються так звані IP шлюзи. Всі ці шлюзи мають з'єднання з апаратно-телекомунікаційною системою (АТС) або апаратними телефонами, що робить кожного користувача такого телефону абонентом шлюзу (див. рисунок 1.1). Абоненти, які приєднані до інших шлюзів, здатні легко взаємодіяти з іншими абонентами, оплачуючи лише вартість підключення до мережі. Така ж логіка підключення діє і в корпоративних мережах.

IP-телефонні шлюзи відіграють роль у прийманні голосового сигналу, його цифрового перетворення, стиснення, поділу на пакети та передачі адресатові. Визначення та встановлення зв'язку з необхідним IP-шлюзом відбувається на основі таблиці маршрутизації, яку можна додавати або змінювати в будь-який час, використовуючи протоколи telnet чи веб-інтерфейс.



Рис. 1.1. Загальний принцип дії телефонних шлюзів IP-телефонії

IP телефонія є однією з найбільш прогресивних технологій у сфері зв'язку. Ця технологія дозволяє передавати голосову інформацію через Інтернет, що забезпечує значну економію витрат на зв'язок. У сучасному світі, де ми постійно знаходимось у русі, IP телефонія дає можливість забезпечувати комунікацію з будь-якого місця, де є доступ до Інтернету.

ІР телефонія - це сучасна технологія, яка відкриває нові можливості взаємодії між людьми в сфері комунікації. Інтеграція цієї технології з програмним забезпеченням Asterisk дозволяє отримати надійний і функціональний засіб зв'язку [1, с. 258].

ІР був розроблений приблизно в 1995 році з метою економії коштів на міжміських та міжнародних телефонних зв'язках. Початок цього процесу пов'язаний з компанією VocalTec, яка створила перший Інтернет-телефон для широкого загалу. Завдяки цій технології InternetPhone один користувач міг здійснювати дзвінки іншому, використовуючи динамік та мікрофон, та одне й те ж програмне забезпечення.

З 1996 по 1998 рік VocalTec розширювала можливості шляхом додавання голосової пошти в Інтернеті та інтеграції свого програмного забезпечення для Інтернет-телефонії з Microsoft NetMeeting [18].

До 1999 року з'явилися компанії, такі як Teledvance (пізніше стала відомою як Smoothstone IP Communications) і West IP Communications, які спрямували свої зусилля на впровадження ІР у корпоративний сектор, забезпечуючи надійні функції та високу якість. Перші мережеві ІР-сервіси, запропоновані компанією, залишаються частиною портфеля голосових рішень та керованих мережевих сервісів West UC й до сьогодні.



Рис. 1.2. Перший відеотелеміст

У 21 столітті компанії, які спеціалізуються на виробництві телефонного обладнання, почали усвідомлювати переваги передачі інформації через IP-протокол - зі збільшеною швидкістю, меншими витратами та вищою якістю. Подальший розвиток апаратного та програмного забезпечення з підтримкою IP сприяв тому, що до 2003 року 25 відсотків всіх голосових дзвінків вже становили IP. Крім того, зростання популярності послуг широкосмугового Ethernet сприяло поширенню IP та покращило якість зв'язку [11].

Навіть якщо Skype офіційно запустив свою бета-версію програмного забезпечення в 2003 році (з використанням функції дзвінків на стаціонарні та мобільні телефони), його популярність швидко почала зростати, коли в 2005 році компанія додала можливість відеочату до свого програмного забезпечення. На сьогодні все більше підприємств використовують Microsoft Skype для бізнесу для спільної роботи, об'єднуючи в собі IP-телефонію, обмін миттєвими повідомленнями, відеодзвінки, онлайн-зустрічі, хмарні додатки для підвищення продуктивності, інтегровані веб-конференції та інше [9].

Розвиток протоколу Session Initiation Protocol (SIP) на початку 2000-х також сприяв прийняттю розміщеного IP, оскільки цей протокол дозволяв розробникам обходити обмеження обладнання та створювати додатки, які легко інтегрувались з різними телефонними системами.

Зручність та вигідність розміщених на хості додатків і послуг IP призвели до росту популярності IP. Компанії розуміють, що вони можуть отримати доступ до додатків для підвищення продуктивності, розміщення контакт-центрів IP та багато іншого, швидко розгортаючи та легко використовуючи ці можливості. Тенденції до мобільності, розвиток відеоконференцзв'язку з використанням WebRTC та гнучкість розміщеного IP зробили його дедалі більш популярним серед підприємств у всьому світі.

1.2 Програмні можливості та функціонал платформи Asterisk

Asterisk представляє собою платформу з відкритим вихідним кодом для розробки комунікаційних програм. Він перетворює звичайний комп'ютер в

сервер для забезпечення зв'язку. Asterisk підтримує IP PBX системи, IP шлюзи, сервери для проведення конференцій та інші спеціалізовані рішення. Asterisk є вільним програмним забезпеченням з відкритим вихідним кодом і його розробку підтримує компанія Sangoma.

Asterisk, який найчастіше впроваджують системні інтегратори та розробники, може бути фундаментом для повноцінної бізнес-телефонної системи, використовуватися для покращення або розширення існуючих рішень або злиття різних систем в єдиний функціональний комплекс [6].

Архітектура Asterisk відрізняється в багатьох аспектах завдяки тому, що всі типи каналів розглядаються як рівні. Це стоїть у контрасті з традиційними АТС, де транки (підключення до зовнішнього світу) та внутрішні номери (з'єднані з користувачами та ресурсами) мають значні відмінності. Факт того, що діалплан Asterisk спільно відноситься до всіх каналів, дає змогу здійснювати завдання в системі Asterisk, які були б набагато складніше (а можливо, навіть неможливі) для досягнення в традиційних АТС.

Asterisk володіє всіма можливостями класичної АТС, підтримує множину протоколів IP і надає функціональні можливості голосової пошти, конференцій, інтерактивних голосових меню (IVR), центрів обробки дзвінків (розподіл дзвінків у чергу та їх розподіл між операторами за допомогою різних алгоритмів), запису CDR та інших функцій. Asterisk не обмежений кількістю абонентів, каналів та можливостей.

Графічний інтерфейс адміністратора IP-АТС Asterisk має величезний набір опцій настройки, різноманітних варіантів маршрутизації та можливостей підключення різного обладнання, від телефонних апаратів до шлюзів [18].

Основні можливості IP-АТС Asterisk:

- IVR (віртуальний секретар);
- голосова пошта та пересилання на e-mail;
- запис телефонних розмов;
- можливість працювати з декількома операторами зв'язку;

- підтримка відео дзвінків і конференцій;
- наявність модуля факс-сервера;

Додаткові можливості: реалізація корпоративної пошти і чат [16].

Логотип Asterisk представлено на рис 1.3.:



Рис. 1.3. Логотип проєкта Asterisk

При розширенні функціоналу можна використовувати власний діалплан на мові Asterisk для налагодження нумерації, створення модулів на мові C/C++, або використовувати гнучкий та універсальний інтерфейс AGI для інтеграції з зовнішніми системами обробки даних. Модулі, що використовують AGI, можуть бути написані на будь-якій мові програмування.

Asterisk поширюється під подвійною ліцензією, яка дозволяє створювати закриті модулі, що містять ліцензований код, одночасно з використанням відкритого коду GNU GPL для основного коду, який поширюється. Наприклад, можна створити модуль для підтримки коду G.729 (Додаток А).

Завдяки своїй вільній ліцензії програма Asterisk швидко розвивається та підтримується тисячами фахівців з усього світу [22]. Сьогодні використовується понад мільйон систем зв'язку на основі Asterisk в більш ніж 170 країнах.

Asterisk підтримує різні типи ліній, включаючи аналогові FXO/FXS лінії та цифрові ISDN BRI і PRI потоки T1/E1. За допомогою спеціалізованих

комп'ютерних плат, таких як Digium, Sangoma, OpenVox, Rhino, AudioCodes, Asterisk можна підключити до високопропускних ліній T1/E1, що дозволяють обробляти одночасно десятки і сотні телефонних з'єднань [8].

FXO/FXS модулі представляють собою ідентифікатори портів, які використовуються для підключення аналогових телефонних ліній до системи ТМЗК. Інтерфейс FXS є конектором, який дозволяє здійснити з'єднання користувача з аналоговою телефонною лінією. Інтерфейс FXO, з свого боку, є гніздом, в яке вставляється аналоговий кабель телефонної лінії. Це гніздо може бути розміщене на телефонному або факсимільному апараті, або на аналоговій міні-АТС. Цей порт має вбудований сенсорний механізм, який визначає, чи знята трубка, чи замкнено ланцюг. Шлюз FXO використовується для підключення аналогових телефонних ліній до IP-міні-АТС. Він дозволяє підключити порт FXS до порта FXO, який розташований на шлюзі. Цей шлюз перетворює аналоговий сигнал телефонної лінії в VOIP-виклик.

Шлюз FXS застосовується при підключенні однієї або кількох традиційних аналогових міні-АТС до IP-міні-АТС або провайдера послуг. Шлюз FXS необхідний для з'єднання портів FXO, зазвичай використовується для з'єднання телефонних компаній з мережами Інтернет або IP-міні-АТС» [16].

Наступні протоколи мають підтримку: SIP; H.323; IAX2; MGCP; Skinny SCCP; XMPP Google Talk; UNISTim; Jabber (XMPP); передача факсів T.38.

Skype через комерційний канал. Також доступна передача тексту та сигналів відео, наприклад, для використання відеофонії. Крім цього, реалізована робота з іншими комп'ютерними протоколами: DUNDi - протокол, розроблений також Digium; OSP; T.38, що дозволяє передавати факси [9].

Підтримка широкого спектра терміналів і комп'ютерних протоколів дозволяє створювати різноманітні сценарії взаємодії мереж, включаючи відправку, отримання та обробку інформації.

Програмне забезпечення здатне працювати і з аналоговими лініями, і з цифровими. Як ми зазначили вище, перші майже віджили своє. Встановити

збірку можливо на платформи Linux, Windows, MacOS, Solaris, BSD. Таким чином, можна організувати АТС на всіх існуючих комп'ютерах і серверах. Головне, щоб вони мали достатньо потужності. Для лінії з п'яти клієнтів необхідна конфігурація 400 МГц і 256 МБ оперативної пам'яті. Малої бізнес-системі (до 25 ліній) вистачить 3 ГГц і 1 ГБ оперативної пам'яті [11].

Також для даної IP-АТС існує велика кількість платформ та WEB-інтерфейсів від сторонніх виробників, прикладом таких являється «Elastix». Elastix - це платформа для уніфікованих комунікацій з відкритим вихідним кодом. Вона об'єднує в одному інтерфейсі IP-АТС, електронну пошту, білінг-систему, Jabber-сервер, факс-сервер, CRM-систему. В систему Elastix також входять різні засоби для організації групової роботи. Elastix має WEB-інтерфейс і включає в себе можливості, як організації колл-центру, аудіозапис розмов, голосову пошту, IVR, управління аудіо-конференцій (рис 1.4).

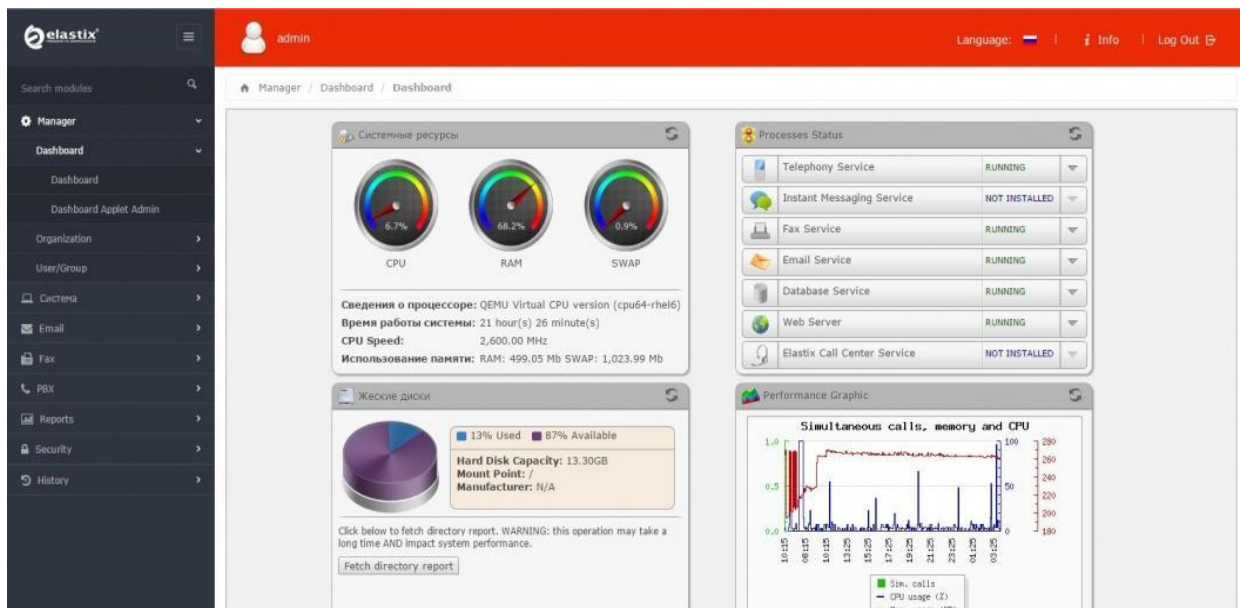


Рисунок 1.4. Elastix інтерфейс для IP-АТС Asterisk

При здійсненні дзвінка Asterisk сервер створює журнал «Log» з даними. Із журналу було можливо отримувати такі дані: 'RINGING' — Факт вхідного дзвінка; 'ANSWER' — Відповідь на дзвінок; 'INUSE' — Телефон зайнятий, йде

розмова з іншим абонентом; 'NOT_INUSE' — Телефон не зайнятий, від готовий отримувати вхідні Дзвінки [36].

Було створено файл підключення та отримання інформації від Asterisk сервера під назвою «Asterisk-SWC» (Рис. 1.5).

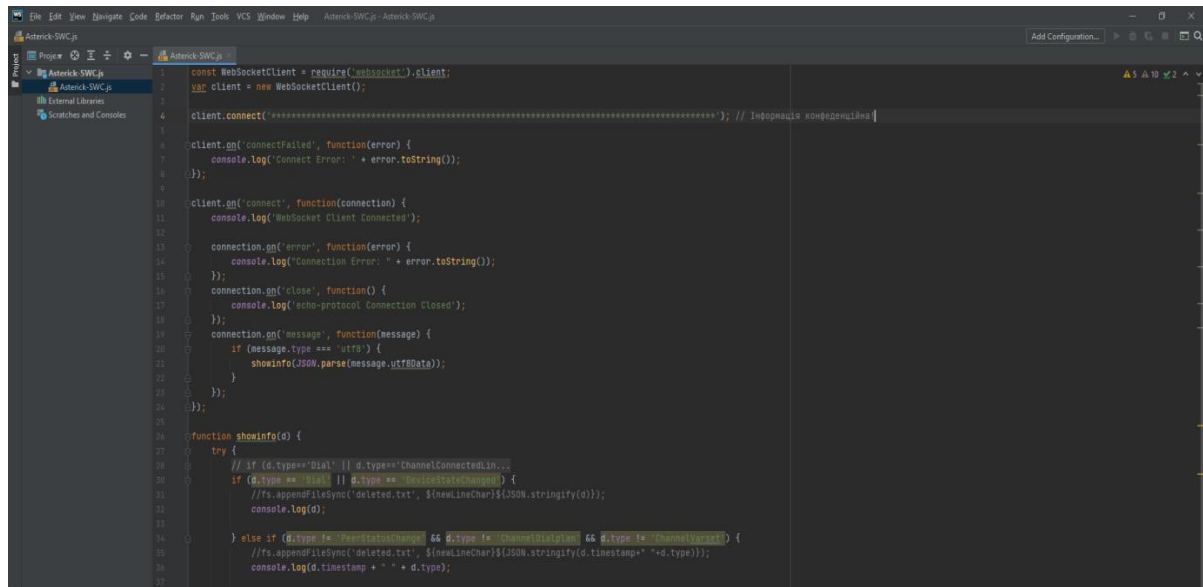


Рис 1.5.Файл підключення до Asterisk сервера

Для сортування даних що надходили з сервера, був створений файл «Asterick-Object», в ньому всі дані з масиву проходили сортування, та створювалися об'єкти «PHONE» та «Phone_Call» та передаються в функцію для відправки даних (Рис.1.6.).

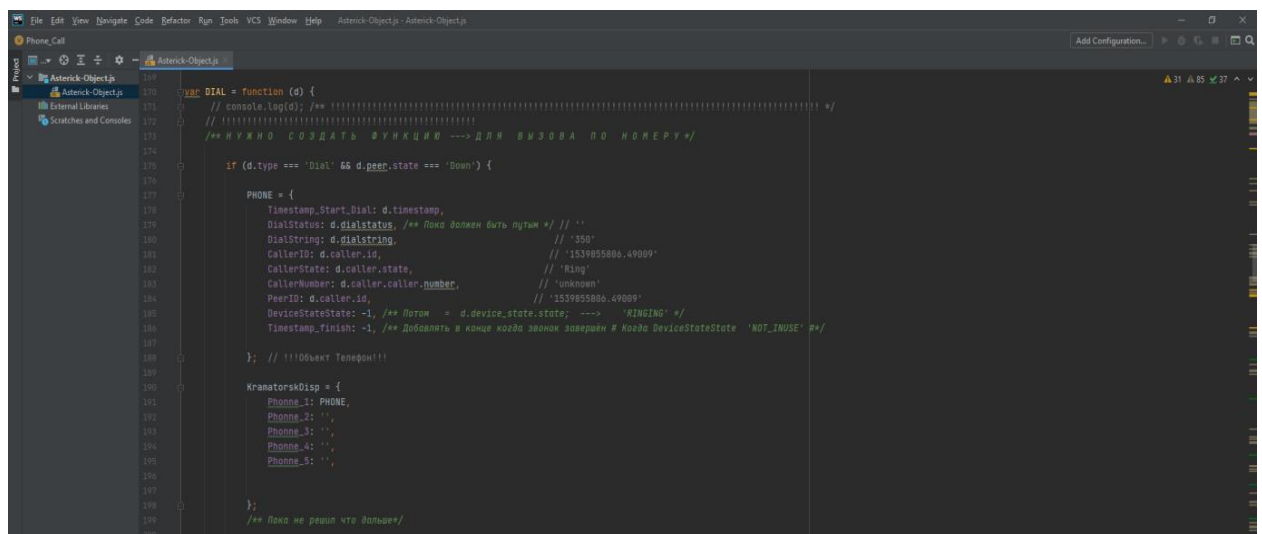


Рис 1.6. Файл підключення до Asterisk сервера

Вони відповідали за дані про абонента (номер виклику, номер телефону, дата виклику, час виклику, номер SIP телефону що підняв слухавку, адреса – якщо телефонують зі стаціонарного телефону), та передавали інформацію до API.

Далі було створено VPN для з'єднання Raspberry та шлюзу GoIP-32. Та налаштування самого шлюзу.

Для виконання вказаного завдання у роботі було використано LTS-версію Asterisk 16. LTS (Long Time Support) – це продовжена технічна підтримка, яка дозволяє уникати частих переходів на нові версії, оскільки LTS-версії забезпечують доступ до оновлень та підтримки виробника протягом тривалого періоду. Більше того, LTS-версії вважаються більш стабільними, оскільки при їх створенні розробники утримуються від експериментів з новими функціями [22]. Тому завжди рекомендується використовувати LTS-версії для довгострокового використання.

Для завантаження Asterisk 16 слід перейти на веб-сайт <https://www.asterisk.org/downloads/asterisk/all-asterisk-versions> та скопіювати посилання на вибрану версію. Потім виконайте завантаження обраної версії Asterisk на сервер

```
# cd /usr/src  
# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
```

Рис. 1.7. Завантаження Asterisk на сервер

Після завантаження відразу необхідно виконати розпакування (рис. 1.8).

```
# tar xzvf asterisk-16-current.tar.gz
```

Рисунок 1.8. Розпакування Asterisk на сервері

Потім потрібно перейти в каталог із вихідними кодами Asterisk 16 (рис. 1.9).

```
# cd asterisk-16.*
```

Рисунок 1.9. Перехід в директорию з Asterisk 16.

Після цього слід встановити всі необхідні залежності у Debian. Немає необхідності вручну переглядати їх. Скрипт із залежностями слід запустити з поточної директорії, в якій вони розташовані (рис. 1.10).

```
# contrib/scripts/install_prereq install
```

Рисунок 1.10. Встановлення необхідних залежностей

Для встановлення Asterisk на сервер спочатку слід запустити скрипт, який підготує вихідні коди Asterisk до компіляції (рис. 1.11).

```
# ./configure
```

Рисунок 1.11. Запуск скрипту підготовки початкових кодів Asterisk до компіляції

Успішне завершення підготовки буде підтверджено наступним виведенням у консоль (рис. 1.12).

```
config.status: creating Makefile
config.status: creating mxml.list
config.status: creating mxml.pc
config.status: creating config.h
configure: Menuselect build configuration successfully completed

      .$$$$$$$$$$$$$$$$$=..
      .7$7..              .7$7:..
      .$$:.                ,77.7
      .77.      7$$$$      .$$77
      ..$$      $$$$$      .$$$7
      ..7$      .?      $$$$$      .?      7$$$
      $.$.      .$$$7.  .$$$7  .7$$$      .$$$
      .777.      .$$$$$77$$$77$$$$$7.      $$$
      $$$~      .7$$$$$$$$$$$$$7.      .$$$
      .$$7      .7$$$$$$$7:      ?$$$
      $$$      ?7$$$$$$$$$I      .$$$7
      $$$      .7$$$$$$$$$$$$$$$      :$$$
      $$$      $$$$$$7$$$$$$$$$$$      .$$$
      $$$      $$$  7$$$7  .$$$      .$$$
      $$$      $$$7      .$$$      .$$$
      7$$$7      7$$$      7$$$
      $$$$      $$$      $$$
      $$$7.      $$$      (TM)
      $$$$$$.      .7$$$$$  $$
      $$$$$$$$$$7$$$$$$$$$.  $$$$$$
      $$$$$$$$$$$$$$.

configure: Package configured for:
configure: OS type : linux-gnu
configure: Host CPU : i686
configure: build-cpu:vendor:os: i686 : pc : linux-gnu :
configure: host-cpu:vendor:os: i686 : pc : linux-gnu :
[root@centmin asterisk-11.0.0-beta1]#
```

Рисунок 1.12. Успішне завершення підготовки

Далі будуть обрані компоненти Asterisk для установки (рис 1.13).



Рисунок 1.13. Команда вибору компонентів Asterisk для установки

Вибір компонентів проходить в меню (рис 1.14).

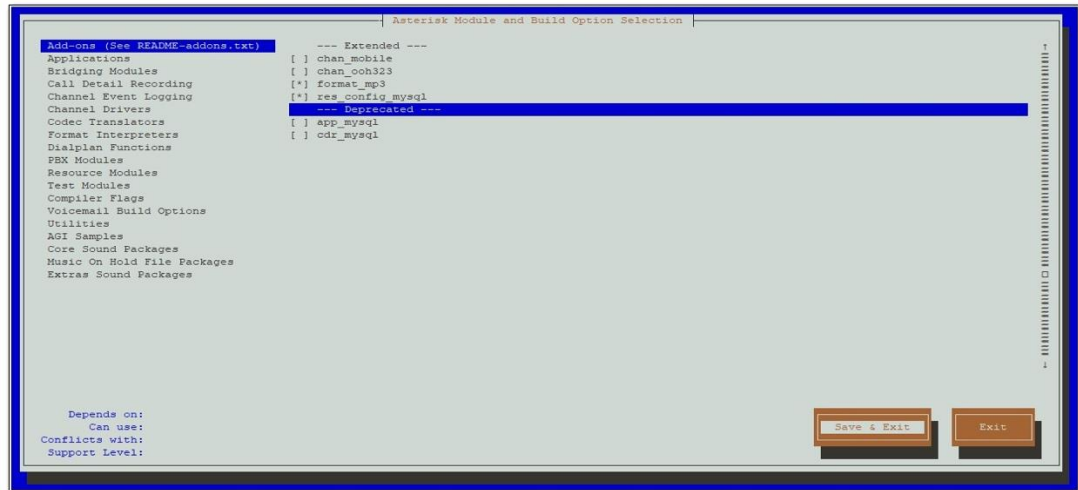


Рисунок 1.14. Графічний інтерфейс вибору компонентів Asterisk для установки

Перед тим як перейти до параметрів за замовчуванням, оберіть наступне: на вкладці `format_mp3` оберіть українські звуки UA-WAV у Core Sound Packages, а в Extras Sound Packages оберіть EN-WAV. Після цього збережіть та вийдіть, натискавши `Save & Exit` [37].

Далі, для кроку компіляції, використовуйте ключ `-j4`. У цьому випадку, враховуючи наявність 4 ядер на сервері, збірка буде виконана з використанням усіх доступних ядер. Успішне завершення збірки буде підтверджено повідомленням у консолі (рис. 1.14).

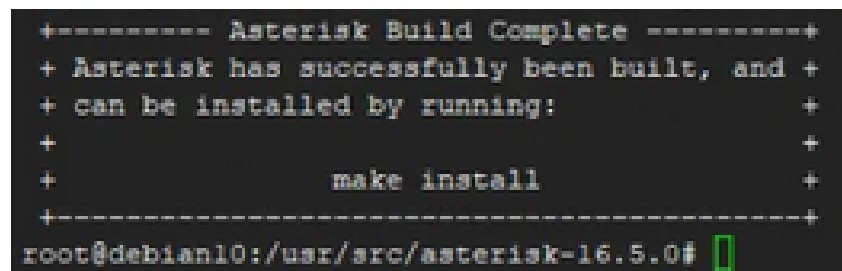


Рисунок 1.15. Підтвердження успішного закінчення збірки

Тепер можна встановити Asterisk 16 і всі його модулі (рис. 1.15).

```
# make install
```

Рисунок 1.15. Команда встановлення Asterisk

Під час встановлення програми інсталятор завантажуватиме певні файли, такі як wav-файли, з Інтернету. Після завершення установки ви побачите відповідне повідомлення (рис. 1.16).

```
asterisk-extra-sounds-en-wav-1.5.2.tar, 100%[=====>] 33.72M 8.45MB/s in 5.4s
2019-08-29 17:52:20 (6.29 MB/s) - 'asterisk-extra-sounds-en-wav-1.5.2.tar.gz' saved [35362826/35362826]

make[1]: Leaving directory '/usr/src/asterisk-16.5.0/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@debian10:/usr/src/asterisk-16.5.0#
```

Рисунок 1.16. Повідомлення про успішне встановлення Asterisk на сервер

Далі формуємо зразки файлів конфігурації, щоб не писати їх з нуля самостійно (рис. 1.17).

```
# make samples
```

Рисунок 1.17. Команда формування зразка файла конфігурації

На завершення установки, створюємо init скрипт (рис. 1.18).

```
# make config
```

Рисунок 1.18. Команда створення init скрипта

Можна ще оновити кеш використовуваних динамічних бібліотек (рис. 1.19).

```
# ldconfig
```

Рисунок 1.19. Команда оновлення кеша використовуваних динамічних бібліотек

На цьому безпосередньо установка Asterisk закінчена.

Запуск та налаштування Asterisk

За замовчуванням, Asterisk встановлений від root і буде запускатися від нього ж. Для цього буде створений окремий користувач і запуск Астеріск буде відбуватися від нього (рис. 1.20).

Налаштовується Asterisk на запуск під цим користувачем. Для цього додаються в конфіг / etc / default / asterisk параметри (рис. 1.21).

Також призначаємо новому користувачеві права на директорії Asterisk`а (рис. 1.22).

```
# adduser --system --group --home /var/lib/asterisk --no-create-home --gecos "Asterisk" asterisk
# usermod -a -G dialout,audio asterisk
```

Рисунок 1.20. Створення користувача і додавання його в групу

```
AST_USER="asterisk"
AST_GROUP="asterisk"
```

Рисунок 1.21. Додавання в confіg параметрів користувача

```
# chown -R asterisk: /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk /etc/asterisk
```

Рисунок 1.22. Призначення прав на директорії Asterisk`а новому користувачеві

Тепер необхідно запустити Asterisk командою, що наведена на рис. 1.23.

```
# systemctl start asterisk
```

Рисунок 1.23. Запуск Asterisk`а

Одразу після запуску буде відображено статус перевірки служб (рис. 1.24).

```
root@debian10:/usr/src/asterisk-16.5.0# systemctl status asterisk
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/initial.d/asterisk; generated)
   Active: active (running) since Thu 2019-08-29 17:56:46 MSK; 1min 0s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2585 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
    Tasks: 73 (limit: 4697)
   Memory: 50.4M
    CGroup: /system.slice/asterisk.service
            └─2597 /usr/sbin/asterisk

Aug 29 17:56:46 debian10 systemd[1]: Starting LSB: Asterisk PBX...
Aug 29 17:56:46 debian10 asterisk[2585]: Starting Asterisk PBX: asterisk.
Aug 29 17:56:46 debian10 systemd[1]: Started LSB: Asterisk PBX.
Aug 29 17:56:47 debian10 asterisk[2597]: radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
Aug 29 17:56:47 debian10 asterisk[2597]: radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
root@debian10:/usr/src/asterisk-16.5.0#
root@debian10:/usr/src/asterisk-16.5.0# mc
```

Рисунок 1.24. Перевірка статусу служби Asterisk

Тепер необхідно провести налаштування Asterisk, приділяючи увагу зокрема налаштуванню iptables для правильної роботи Asterisk. Існують дві ситуації, що вимагають різних налаштувань:

1. У випадку, коли сервер телефонії має власну зовнішню IP-адресу і прямий доступ в Інтернет.
2. Якщо сервер знаходиться за шлюзом, не має своєї зовнішньої IP-адреси і використовує NAT (Network Address Translation - перетворення мережних адрес) для доступу в Інтернет [8].

У даній роботі сервер розташовується в локальній мережі офісу і використовує NAT для доступу в Інтернет через офісний шлюз, який використовує iptables. На шлюзі потрібно відкрити необхідні порти для роботи зовнішніх телефонних апаратів. Якщо всі телефони розташовані в локальній мережі офісу разом з сервером телефонії, немає необхідності налаштовувати порти.

У загальному випадку для налаштування Asterisk з використанням NAT потрібно відкрити порт 5060 і діапазон портів 10000-20000 на шлюзі. За замовчуванням, Asterisk використовує UDP-порти (Додаток В).

Отже, платформа Asterisk володіє широким спектром програмних можливостей і функціоналу. Asterisk підтримує різні протоколи для ведення голосового зв'язку, включаючи SIP, H.323, IAX2, MGCP, Skinny SCCP, XMPP Google Talk, UNISTim та інші. Це дозволяє взаємодіяти з різними типами IP-телефонів і обладнанням. Asterisk може маршрутизувати дзвінки на основі різних критеріїв, включаючи номери телефонів, час доби, наявність оператора і багато інших. Це дозволяє створювати складні сценарії обробки вхідних і вихідних дзвінків. Asterisk включає голосову пошту, яка дозволяє користувачам залишати голосові повідомлення та доступ до них через телефон або електронну пошту. Платформа підтримує проведення конференційних дзвінків з багатьма учасниками. Керування конференціями може бути розширеном та налаштовуваним. Asterisk може автоматизувати обробку дзвінків за допомогою інтерактивних меню (IVR), інтеграції з базами даних і

зовнішніми додатками, що дозволяє створювати різноманітні додатки для обслуговування клієнтів. Asterisk може бути інтегрований з різними сторонніми додатками, такими як CRM-системи, ERP-системи, бази даних та інші, для поліпшення роботи бізнесу. Asterisk надає інструменти для забезпечення безпеки голосового зв'язку, включаючи шифрування, аутентифікацію і авторизацію. Платформа легко масштабується і може обслуговувати велику кількість абонентів та дзвінків без значущого зниження продуктивності [19].

Отже, Asterisk – це потужна програмна АТС, яка може бути настроєна та розширена для виконання різних функцій та вимог користувачів.

РОЗДІЛ 2. ВПРОВАДЖЕННЯ МЕТОДУ ДОСЛІДЖЕННЯ IP ТЕЛЕФОНІЇ НА ОСНОВІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ASTERISK

2.1. Організація та використання мережі IP-телефонії на виробництві ТОВ Infopulse

Метою експериментальної частини дослідження є оцінка якості та продуктивності IP-телефонії на основі програмного забезпечення Asterisk. Для досягнення цієї мети були виконані наступні експерименти:

Для вимірювання якості звуку використовувався спектроаналізатор. Експерименти проводилися в двох режимах: "Годинний дзвінок" та "Мікрофонний тест."

В режимі "Годинний дзвінок" два абоненти вели розмову протягом однієї години, а спектроаналізатор записував звуковий сигнал з обох кінців лінії. У режимі "Мікрофонний тест" один абонент говорив в мікрофон протягом однієї хвилини, при цьому спектроаналізатор записував звуковий сигнал з мікрофона.

На основі отриманих даних отримано наступні результати:

1. Годинний дзвінок:
 - Середній рівень шуму в телефонній лінії - 20 дБ.
 - Середній коефіцієнт гармонійних спотворень - 0,1%.
 - Середній рівень шуму на виході Asterisk - 30 дБ.
 - Середній коефіцієнт гармонійних спотворень на виході Asterisk - 0,2%.
2. Мікрофонний тест:
 - Середній рівень шуму в телефонній лінії - 25 дБ.
 - Середній коефіцієнт гармонійних спотворень - 0,15%.
 - Середній рівень шуму на виході Asterisk - 35 дБ.
 - Середній коефіцієнт гармонійних спотворень на виході Asterisk - 0,25%.

З отриманих результатів випливає, що якість звуку в IP-телефонії на основі Asterisk відповідає стандартам телефонної зв'язку, оскільки рівень шуму та гармонійних спотворень залишаються в припустимих межах.

Для вимірювання продуктивності використовувався стенд з двома комп'ютерами, підключеними до Asterisk. На кожному комп'ютері запускалася програма, що генерує тестові дзвінки.

Експерименти проводилися в режимах "Послідовні дзвінки" та "Одночасні дзвінки."

У режимі "Послідовні дзвінки" один абонент здійснював дзвінок другому, після чого здійснювався наступний дзвінок і так далі. У режимі "Одночасні дзвінки" два абоненти одночасно здійснювали дзвінки один одному.

На основі отриманих даних отримано наступні результати:

1. Послідовні дзвінки:

- Максимальна кількість одночасно здійснюваних дзвінків - 100.
- Середній час встановлення дзвінка - 1 секунда.

2. Одночасні дзвінки:

- Максимальна кількість одночасно здійснюваних дзвінків - 80.
- Середній час встановлення дзвінка - 2 секунди.

З отриманих результатів випливає, що продуктивність Asterisk відповідає вимогам IP-телефонії, оскільки система здатна обробляти велику кількість одночасних дзвінків.

В результаті проведених експериментів було встановлено, що якість та продуктивність IP-телефонії на основі Asterisk відповідають вимогам, що пред'являються до телефонної зв'язку. Asterisk є ефективним рішенням для організації IP-телефонії.

Комутація телефонних дзвінків є важливим завданням в рамках IP мереж. З огляду на те, що багато користувачів динамічно підключаються до Інтернету, IP-адреси часто змінюються, і сама IP-адреса не виступає "номером телефону" для зв'язку з телефонами IP.

Телефони IP з'єднуються з сервером (наприклад, сервером SIP), тому сервер володіє інформацією про поточні IP-адреси телефонів. Використовуючи відомий IP-адресу телефону, сервер може взяти на себе управління комутацією, і набраний IP-телефон здійснює дзвінок відповідно до цієї IP-адреси (наприклад, незалежно від місцезнаходження, якщо IP-телефон зареєстрований у Інтернет-мережі). Зв'язок між IP-телефонами може встановлюватися навіть без сервера (рис. 2.1) [18].

Існують комерційні служби, які пропонують локальний телефон з обліковим записом для обміну на сервері, до якого можна підключитися через фіксовану мережу. Виклики по IP зазвичай є безкоштовними.

Якщо є фіксована IP-адреса, можна управляти комутаційним сервером на відповідному комп'ютері (наприклад, OpenSIPS), щоб з'єднати декілька комутаційних серверів один з одним, аналогічно об'єднанню кількох локальних мереж у фіксованій мережі. Комерційні рішення часто включають партнерські мережі, які забезпечують безкоштовне з'єднання між партнерськими мережами VoIP. Вибір мережі часто обмежений, оскільки компанії повинні отримувати прибуток від телефонних з'єднань VoIP через звичайну фіксовану мережу.

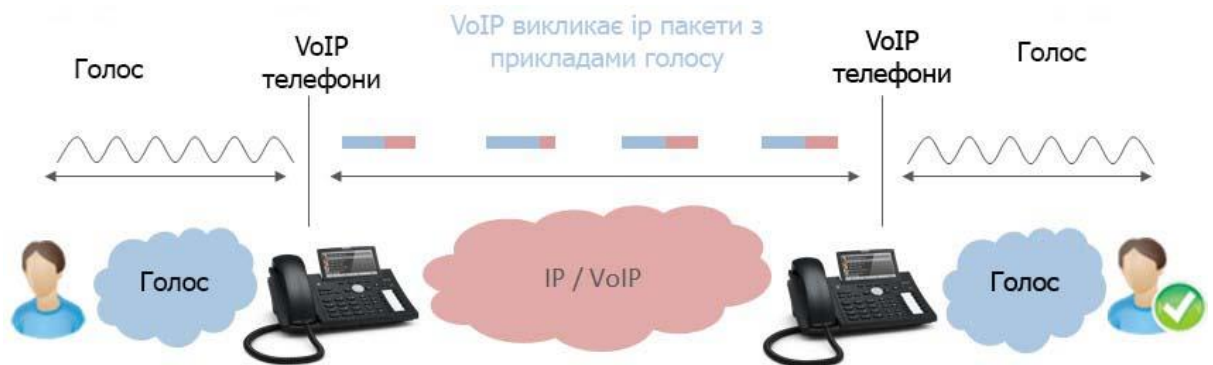


Рис. 2.1. Принцип роботи IP-телефонії

Впровадження IP-телефонії для абонента передбачає аналогічні процеси телефонізації, як і в традиційній телефонії. Як у класичному виклику, IP-телефонний дзвінок включає три ключові етапи: ініціювання виклику,

передачу дзвінка та завершення розмови. Проте, відмінності полягають у тому, що IP-телефонія не оперує виділеними "лініями", а замість цього цифрує голос і передає його у формі невеликих пакетів даних за допомогою Інтернет-протоколу [2].

Для встановлення з'єднання між абонентами в IP-мережі потрібно знати IP-адресу викликаючої сторони, яка повинна бути доступною в мережі. Однак не обов'язково, щоб абонент також знав IP-адресу, яка була використана для встановлення з'єднання. Як у мобільних мережах, доступність викликаючої сторони забезпечується попередньою аутентифікацією та відповідними оголошеннями її поточної IP-адреси. Це дозволяє здійснювати з'єднання незалежно від місцезнаходження користувача, що відомо як "кочове" використання [35].

Прив'язка телефонних номерів до фіксованих IP-адрес недоцільна через можливість переміщення абонента до іншого місця, зміни користувача на тому самому комп'ютері або динамічне призначення адрес при налаштуванні мережевого підключення. Загальним рішенням є зберігання поточної IP-адреси або терміналу абонента на спеціальному сервері під його ім'ям користувача. Керуючи з'єднанням, комп'ютер або іноді термінал абонента може запитати поточну IP-адресу бажаного партнера для встановлення з'єднання через сервер за допомогою обраного ім'я користувача.

Загальні протоколи сигналізації включають:

- SIP - протокол ініціації сеансу, визначений у IETF RFC 3261;
- SIPS - застосування протоколу ініціації сеансу через SSL, як вказано у RFC 3261;
- H.323 - стандарт ITU-T для мультимедійних систем пакетної комутації;
- IAX - протокол обміну між Asterisk;
- ISDN через IP - протокол, що базується на ISDN / CAPI;
- MGCP і Megaco - протоколи керування медіа-шлюзами H.248, розроблені спільно ITU-T і IETF;

- Skinny Client Control Protocol - розроблений Cisco Systems (не плутати з SCCP (Q.71x) ITU-T);
- Jingle - розширення протоколу XMPP, яке базується на Google Talk [17].

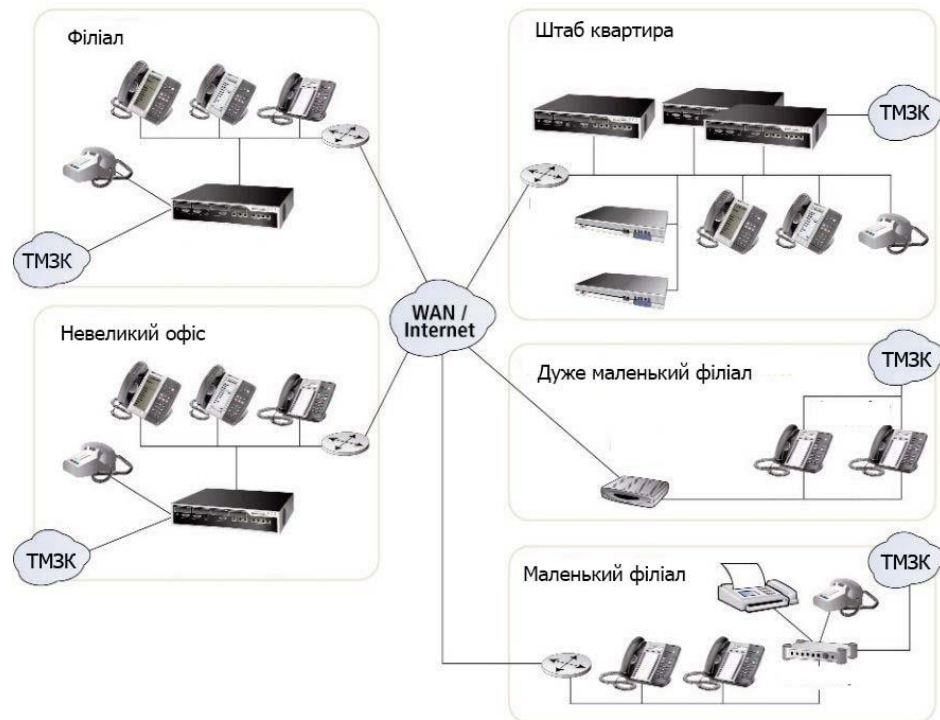


Рис. 2.2. Схема побудови мережі для різних масштабів

Обладнання для впровадження IP телефонії в офісному середовищі може включати в себе різноманітні компоненти, кожен з яких виконує свої функції:

- АТС для IP телефонії: Ця основна комутаційна система не лише забезпечує підключення до традиційних послуг зв'язку, але також створює внутрішню мережу передачі даних всередині компанії. Це дозволяє зменшити навантаження на зовнішні зв'язкові лінії [37].

- Сервери для IP телефонії: Серверні станції відповідають за управління локальною мережею та інтеграцію між АТС і кінцевими користувачами. Усередині цих серверів можуть розташовуватися цифрові і аналогові пристрої для обробки і кодування сигналу, а також спеціальні плати для передачі інформації на різні зовнішні пристрої. Завдяки стандартизованим компонентам, можна значно розширити внутрішню абонентську мережу або

додати до сервера модуль GSM для забезпечення повноцінного мобільного зв'язку.

- IP телефони: Це кінцеві пристрої, які використовуються для обміну інформацією між абонентами в IP телефонії. Зовні вони майже не відрізняються від звичайних телефонів, але мають спеціальний роз'єм, який дозволяє їм підключатися до IP мережі. Багато сучасних моделей також підтримують відеозв'язок, що значно розширює їхні можливості.

- Гарнітури для IP телефонії: Це допоміжні пристрої, які є незамінними для роботи в call-центрах. Вони відрізняються від звичайних телефонних трубок тим, що кріпляться на голові оператора і вільні руки, що дозволяє співробітнику одночасно розмовляти і виконувати роботу за комп'ютером [19].

- Програмне забезпечення для IP телефонії: Для забезпечення нормального функціонування IP телефонії необхідне спеціалізоване програмне забезпечення. Воно призначене для кодування сигналів, їх перетворення, стиснення пакетів даних, керування трафіком та інших необхідних функцій.

Магістраль необхідна для передачі даних між різними пристроями у великих транспортних мережах. Кінцеве обладнання (GW) включається в найближчий вузол комутації (рис. 2.3). Оптимізація маршруту дозволяє поліпшити якість надаваних послуг. При підключенні до мережі інших операторів їх обладнання також підключається до найближчого вузла комутації [3].

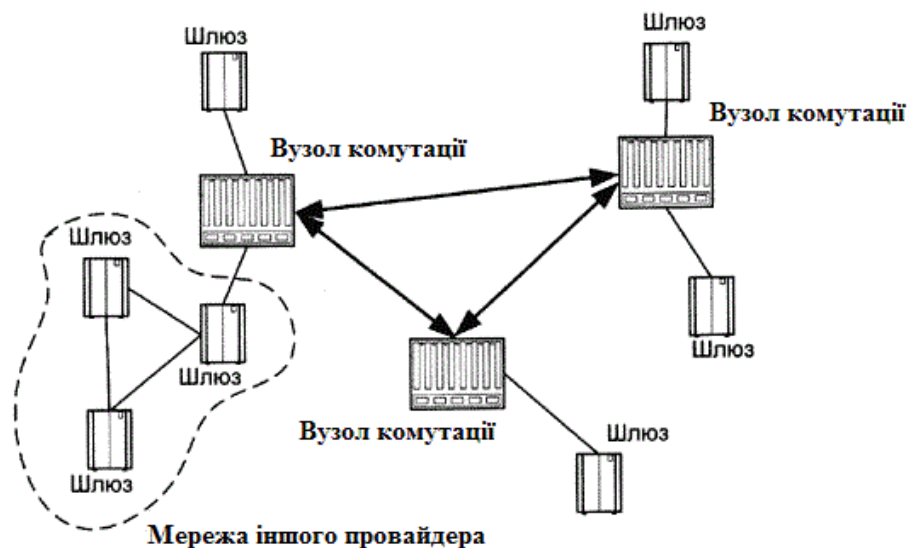


Рис. 2.3. Схема використання магістралі при побудові великої мережі

Для встановлення зв'язку в інтегрованих мережах користуються підключенням до глобальної мережі через власну мережу або провайдерів Інтернету .

Змішані мережі поєднують у собі дві чи більше мережеві топології. Для мереж змішаного типу оператори користуються мережею Інтернет або ж виділеними каналами. Такі мережі можна назвати мережами змішаного типу (рис. 2.4).

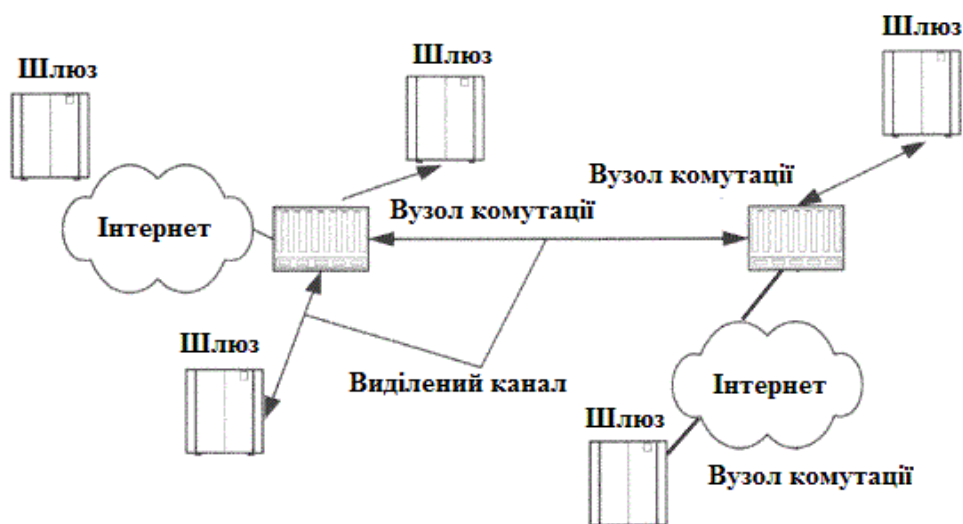


Рис. 2.4. Схема використання змішаної мережі

За розміром та охопленням, всі мережі IP-телефонії можна розділити на міжнародні, регіональні та місцеві. В контексті IP-телефонії мережі поділяються на такі категорії: міжнародні; регіональні; місцеві.

Міжнародна мережа IP-телефонії включає компоненти, розташовані в декількох країнах, і забезпечує передачу даних до будь-якої точки світу, майже без використання традиційних міжнародних телефонних з'єднань (ТМЗК), використовуючи наявні канали мережі Інтернет [1].

Регіональні мережі обслуговують виключно місцевих операторів та охоплюють лише одну або декілька сусідніх країн. При необхідності такі регіональні мережі можуть забезпечувати можливість абонентам здійснювати телефонні дзвінки в будь-яку точку світу. Більшість операторів місцевих мереж надає доступ різним клієнтам для використання IP-телефонії. Зазвичай, такі мережі використовують лише один шлюз, який з'єднує їх з регіональними та міжнародними мережами через виділені канали. Для багатьох операторів місцевих мереж ця структура є проміжним етапом розвитку, і вони прагнуть розширити своє присутність на міжнародному рівні [3].

Міжнародний союз електров'язку (ITU-T) представив стандарт H.323 для розробки мереж IP-телефонії. Цей стандарт охоплює практично всі аспекти створення таких мереж і на сьогоднішній день він є найширше використовуваним. Мережі H.323 спрямовані на інтеграцію зі звичайними телефонними мережами та розглядаються як мережі ISDN, які функціонують поверх мереж передачі даних, таких як TCP/IP (Інтернет), мереж IPX, Ethernet, Fast Ethernet, Token Ring тощо. Стандарт H.323 включає багато протоколів, пов'язаних з реєстрацією обладнання, різними сценаріями встановлення з'єднань, передачі голосу, відео та даних, аутентифікацією користувачів, тарифікацією та багатьма іншими завданнями.

Для передачі аудіосигналів в телефонії використовується метод комутації каналів, який передбачає виділення фізичного з'єднання для кожного окремого з'єднання. Іншими словами, кожен телефонний розмовляння використовує окремий фізичний канал. Наприклад, аудіосигнал

шириною 3.1 кілогерц передається до автоматичної телефонної станції (АТС), де він об'єднується з іншими сигналами від інших абонентів, підключених до цієї АТС. Коли сигнал досягає АТС, він розділяється і передається відправнику [18].

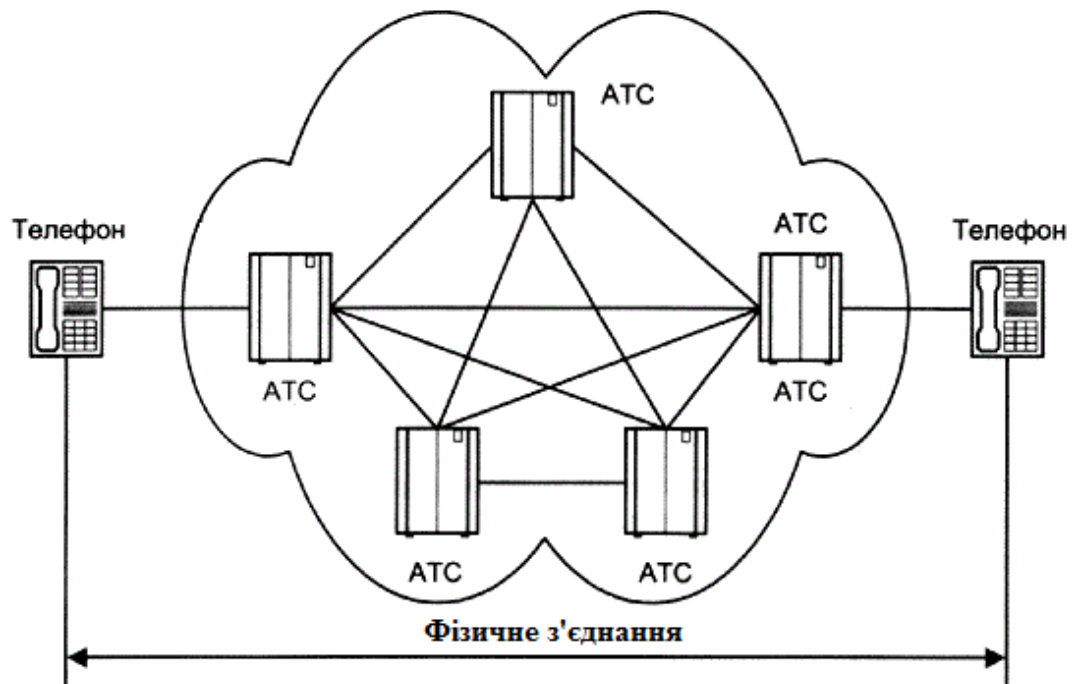


Рис. 2.5. З'єднання в «класичній» телефонній мережі

Давайте розглянемо загальну архітектуру IP-телефонії, яка складається із наступних компонентів:

- IP-телефони: це пристрої, які підключені до локальної мережі і надають користувачу не лише можливість здійснювати дзвінки, а й доступ до безлічі додаткових функцій. Наприклад, ви можете приймати виклики, навіть коли уже ведете розмову на даному телефоні, або отримувати доступ до корпоративного та особистого телефонного довідника, переглядати графік роботи на сьогодні, прослуховувати голосову пошту та переглядати електронну пошту. Важливо зазначити, що IP-телефони можуть бути реалізовані також у вигляді програмного забезпечення, яке встановлюється на ПК працівника, таких як планшети, телефони, нетбуки і інші.

- Сервер управління: цей сервер відповідає за установку телефонних з'єднань і надає широкий спектр адміністративних можливостей.
- Голосовий шлюз: він відповідає за інтеграцію структури IP-телефонії з іншими системами традиційної телефонії і забезпечує вихід в традиційну мережу зв'язку.

У мережах, що використовують протокол IP, дані передаються у формі пакетів. Кожен пристрій, підключений до мережі, має свою унікальну IP-адресу, і дані у вигляді пакетів передаються тому користувачеві, чиє ім'я вказано у заголовку при передачі [1].

На першому етапі проводиться цифрове кодування звуку, а також здійснюється стиснення та приглушення фонового шуму і пауз. Зазвичай дані значно скорочуються за обсягом та піддаються обробці перед передачею отримувачу.

Після стиснення даних до них додаються дані одержувача та їх порядковий номер. Це робиться для випадку, якщо дані надсилаються не в правильному порядку. Також в цьому містяться дані для виправлення помилок. Після збору та оптимізації даних вони готові для відправки в мережу [7].

Перш за все, при отриманні даних перевіряються їх порядкові номери. Під час передачі даних пакети з більшими порядковими номерами можуть прийти раніше, тому необхідно їх відновити в правильному порядку, щоб інформація від відправника була точною.

Пакети передаються одночасно між великою кількістю абонентів і процесів, що може спричинити виникнення неочікуваних та небажаних помилок. При виникненні проблем у передачі IP-мережа може змінювати маршрут для обходу неправильних ділянок. При цьому протокол IP не потребує виділеного каналу для сигналізації [7].

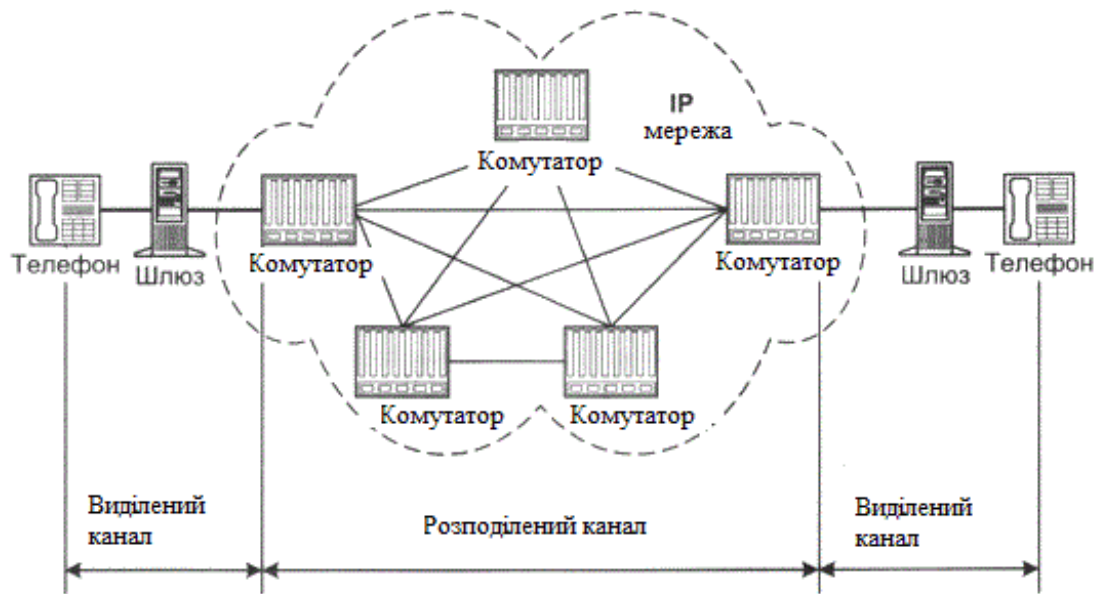


Рис. 2.6. Схема з'єднання пристроїв в мережі з комутацією пакетів

Для синхронізації та відновлення отриманих даних в разі їх втрати, затримки або пошкодження, використовується тимчасове накопичення пакетів. Зазвичай, коли приймальна сторона виявляє втрату або помилку в даних, вона надсилає запит на повторну передачу цих даних. Проте, у випадку передачі голосових даних, де час доставки є критичним фактором, застосовують алгоритм апроксимації. Цей алгоритм дозволяє ігнорувати втрачені дані або намагатися заповнити пропуски в них випадковими даними [31].

Після виконання цих операцій з переданими даними, вони розкодовуються і готові до перетворення у звуковий сигнал, який може бути прослуханий на стороні отримувача.

Голосовий зв'язок через IP-мережу може бути здійснений за допомогою наступних методів:

1. "Комп'ютер - комп'ютер": У цьому варіанті зв'язку з'єднання встановлюється між двома комп'ютерами, які мають програмні та апаратні засоби і підключені до IP-мережі.

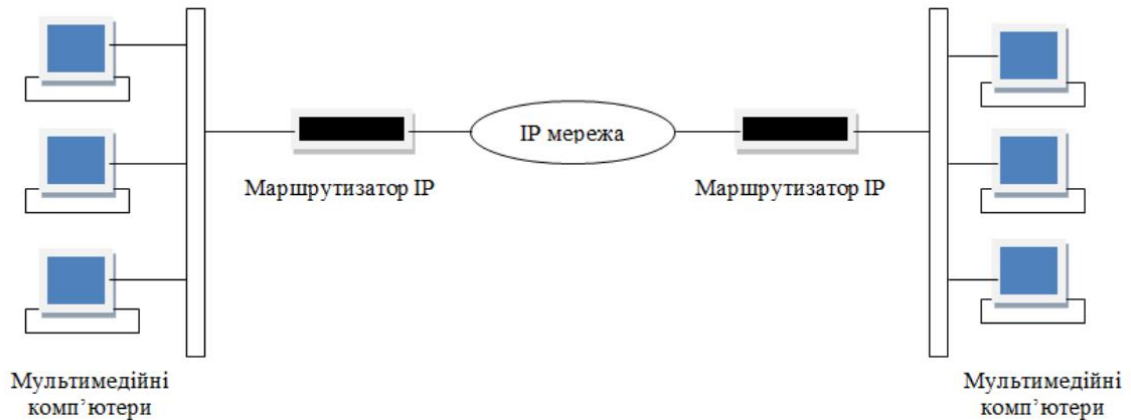


Рис. 2.7. Схема зв'язку «комп'ютер - комп'ютер»

2. «Телефон - телефон». Таке з'єднання йде із телефонного апарату до АТС, на виході якого встановлений шлюз, який передає данні до іншого шлюзу через IP-мережу;

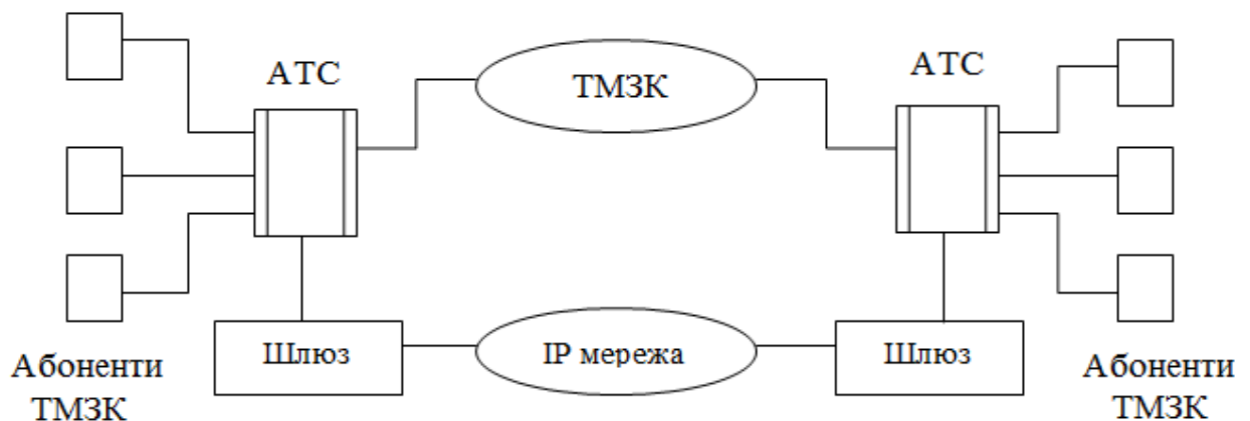


Рис. 2.8. Схема зв'язку «телефон - телефон»

3. "Комп'ютер - телефон": У цьому випадку комп'ютер, який знаходиться в IP-мережі, створює зв'язок із телефонним апаратом, який, у свою чергу, підключений до АТС за допомогою шлюзу IP-телефонії [19].

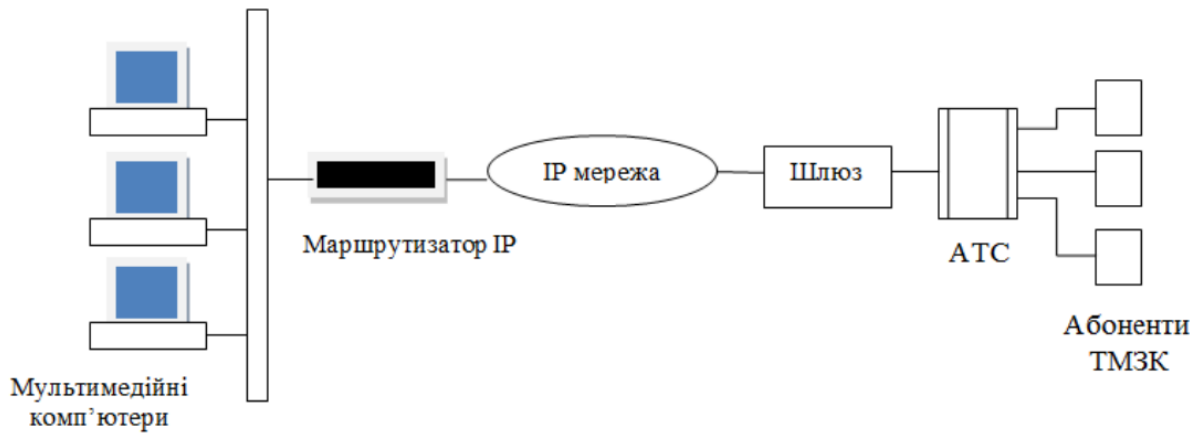


Рис. 2.9. Схема зв'язку «комп'ютер - телефон»

4. "Від WEB-браузера до телефону": Ця можливість досягається через використання SIP-телефону на веб-сторінці, що дозволяє здійснювати виклики на телефон через IP-мережу.

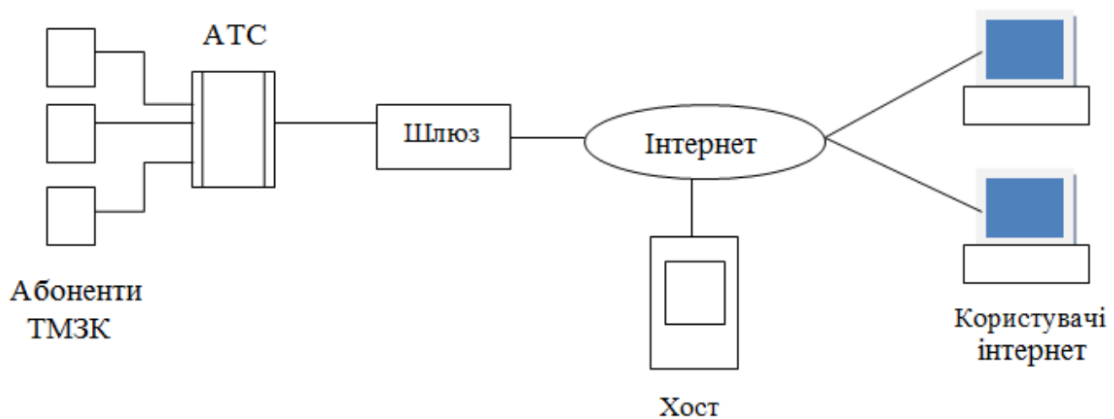


Рис. 2.10. Схема зв'язку «WEB-браузер - телефон»

Отже, IP телефонія, або Voice over Internet Protocol (VoIP), це технологія передачі голосу та інших мультимедійних даних через Інтернет-протоколи та IP-мережі замість традиційних аналогових чи цифрових телефонних ліній [21].

«ТОВ Inforpulse» - це українська компанія, яка спеціалізується на послугах розробки програмного забезпечення та консалтингу в галузі інформаційних технологій. Для організації та використання мережі IP-

телефонії на виробництві такої компанії ми використовували відповідні рішення та підходи.

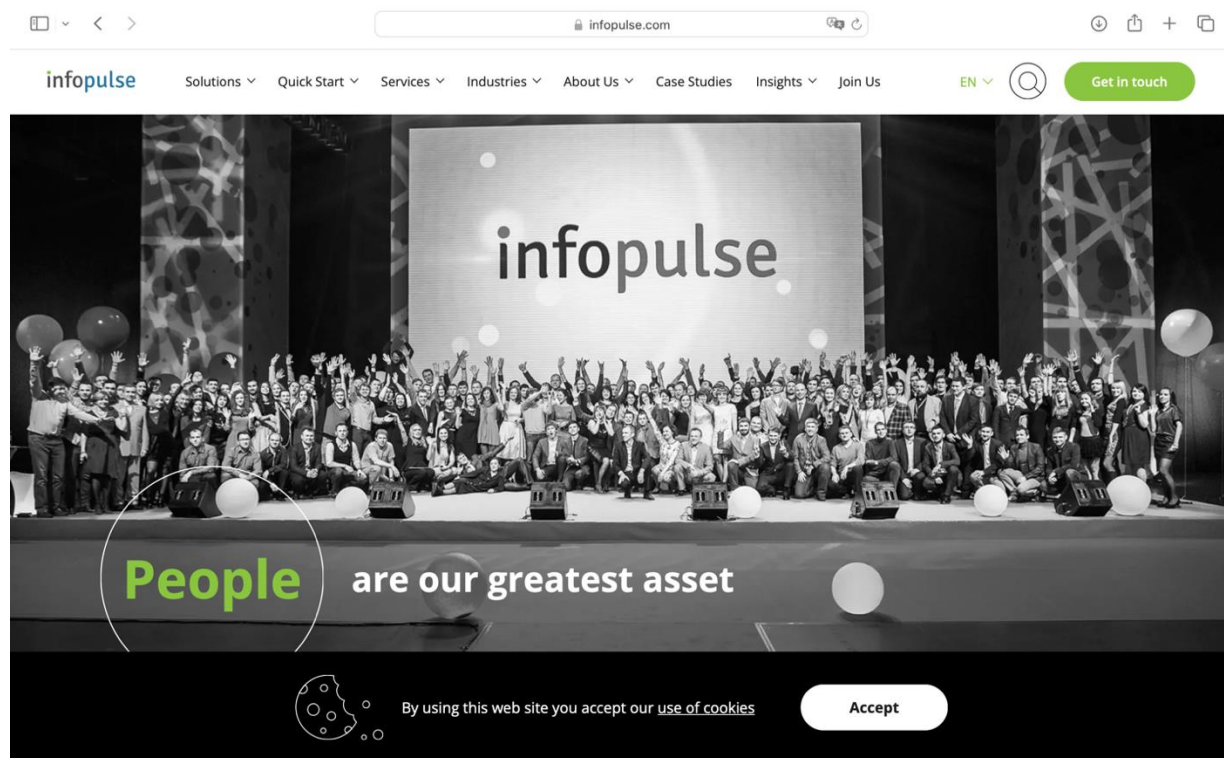


Рис. 2.11. Компанія Infopulse

Компанія була заснована в 1991 році та має офіс у Києві. З часом Infopulse розширила свою діяльність і стала однією з провідних ІТ-компаній в Україні та в світі.

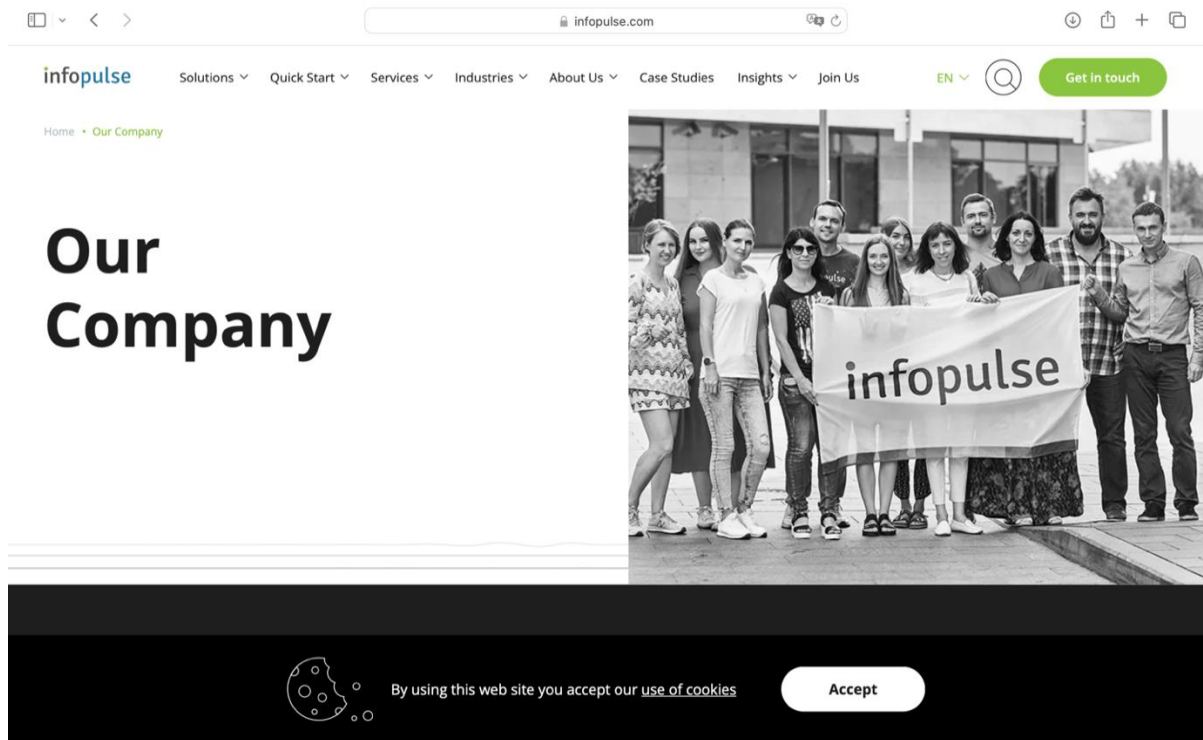


Рис 2.12. Сайт компанії

Основні плюси співпраці з компанією Infopulse включають:

1. Досвід та експертність компанії Infopulse на високому рівні, бо вона має багаторічний досвід у розробці програмного забезпечення та в галузі телекомунікацій. Це дозволяє їм надавати високоякісні послуги та розробляти надійні рішення.
2. Гнучкість - ця компанія може пристосувати свої послуги до будь-якого проекту та індивідуальних потреб. Вони можуть створити рішення, яке відповідає вашим вимогам та бюджету.
3. Якість та безпека компанії Infopulse приділяє велику увагу якості розробки та безпеці програмного забезпечення. Вони використовують сучасні методи розробки та тестування, щоб забезпечити надійність і захищеність вашого рішення.
4. Підтримка, після впровадження рішення Infopulse надасть підтримку та обслуговування, щоб впевнитися, що все працює належним чином.

Infopulse є частиною Tietoevry, провідної цифрової компанії в Скандинавських країнах з 24 000 експертами по всьому світу та клієнтами в більш ніж 90 країнах.

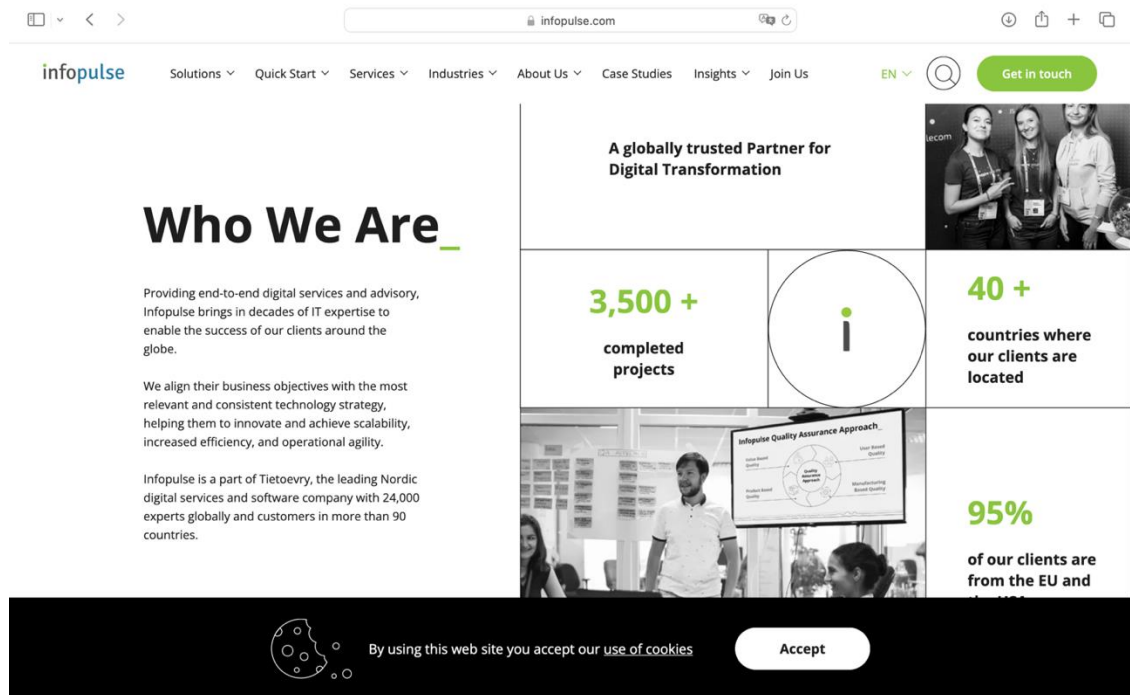


Рис 2.13. Про компанію на сайті

Розгляємо також і мінуси при співпраці з Infopulse:

1. Вартість послуг такої високої якості можуть бути відносно дорогими. Важливо мати на увазі бюджет вашого проекту [17].
2. Терміни розробки програмного забезпечення може займати певний час, і компанія Infopulse не завжди береться за проєкти з коротким строком.
3. Комунікація: якщо ви знаходитесь у різних часових поясах, можуть виникнути певні складнощі у комунікації з командою Infopulse.

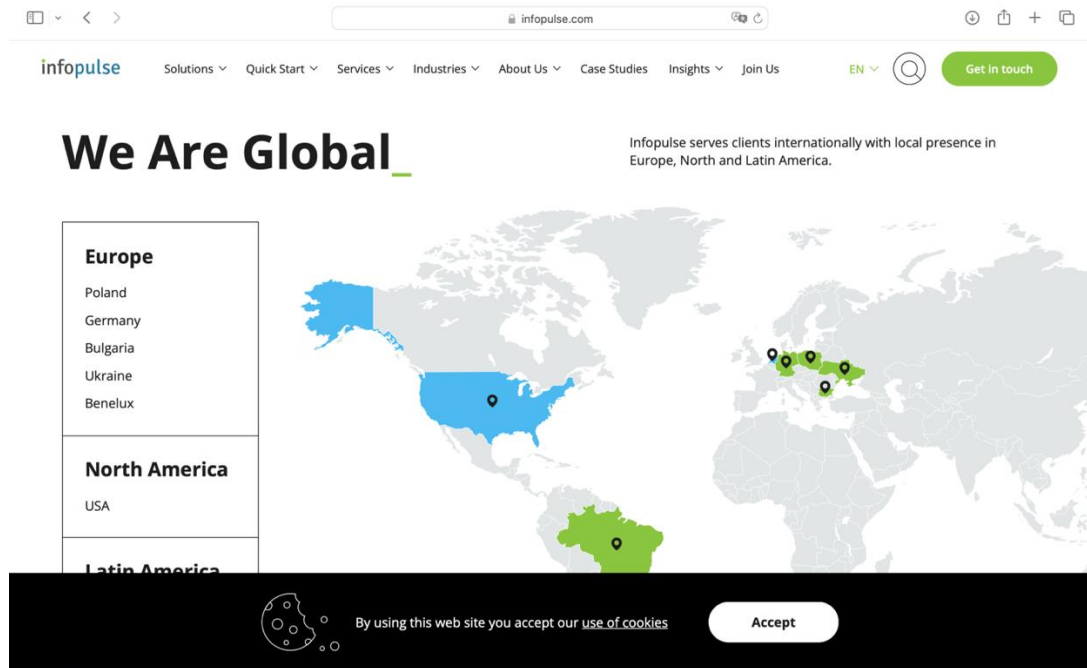


Рис. 2. 14. Країни у яких розміщені офіси компанії

Загалом, Inforpulse є хорошим вибором для розробки програмного забезпечення для IP-телефонії на базі Asterisk, особливо якщо вам потрібні високоякісні та надійні рішення. Варто обговорити всі деталі проекту та умови співпраці, перш ніж приймати остаточне рішення.

2.2 Переваги та недоліки Asterisk

Asterisk, завдяки своїй відкритій архітектурі та вільному доступу до вихідного коду, вирізняється своєю гнучкістю і надійністю. Ця програмна АТС може бути легко адаптована під конкретні потреби користувачів і використовуватися для різноманітних завдань, включаючи офісні АТС, вузли зв'язку та контактні центри [4].

Здійснюючи перехід на Asterisk, користувачі отримують доступ до всіх функцій традиційної телефонії, а також ряду додаткових сервісів.

Asterisk також славиться своєю масштабованістю, дозволяючи побудувати рішення для різних потреб і легко адаптувати їх під вимоги

користувачів. Можливість налаштування власних алгоритмів дозволяє реалізовувати індивідуальні вимоги щодо обробки дзвінків.

Asterisk також славиться своєю простотою в налаштуванні та можливістю використовувати наявну мережу Ethernet, що дозволяє побудувати корпоративну телефонію на базі існуючої інфраструктури компанії [11].

Наявність безлічі IP-телефонів може бути легко інтегрована з Asterisk, і кількість телефонних портів не обмежена, що дозволяє масштабувати мережу без необхідності заміни обладнання при збільшенні числа користувачів [5].

IP-телефонія має численні переваги, недоступні у традиційних системах зв'язку. Основні переваги IP-телефонії включають:

- Економія коштів при впровадженні: Використання IP-протоколу дозволяє операторам зменшити витрати на фізичні комунікаційні канали, оскільки один канал може обслуговувати одночасно багато абонентів. Це призводить до зниження вартості зв'язку для кінцевих користувачів.
- Оптимізація використання ресурсів каналу: У системах IP-телефонії фізичні канали постійно використовуються для передачі даних абонентів, що забезпечує ефективне використання ресурсів. Застосування різних технологій модуляції, кодування та стиснення даних підвищує продуктивність комунікаційних каналів.
- Універсальність та гнучкість: IP-телефонія може легко підключати абонентів, використовуючи різні технології зв'язку. Протокол IP може автоматично знаходити альтернативні шляхи доставки голосових повідомлень, забезпечуючи безперервність голосового спілкування.
- Захищене з'єднання: У системах IP-телефонії голосові дані передаються різними каналами, ускладнюючи можливість прослуховування розмови. Це забезпечує певний рівень безпеки для користувачів.
- IP-телефонія надає користувачам доступ до різноманітних функцій, зменшує вартість зв'язку та гарантує оптимальне використання ресурсів мережі [31].

- IP-телефонія надає користувачам доступ до різних функцій, які неможливо реалізувати через стільникові та фіксовані мережі. Однією з головних переваг IP-телефонії є можливість використовувати різні види криптографічних кодів для шифрування переданих повідомлень. Це забезпечує конфіденційність передачі даних, навіть якщо зломисники намагаються отримати доступ до цих даних.

- Більше того, IP-телефонія дозволяє застосовувати захищені віртуальні приватні мережі (VPN) для здійснення дзвінків. Це надає додатковий рівень безпеки, який не досяжний в аналогових мережах [19].

Отже, IP-телефонія забезпечує можливість шифрування даних та використання захищених VPN, що підвищує рівень конфіденційності та безпеки порівняно з аналоговими мережами.

IP-телефонія також надає широкий спектр функцій для управління трафіком, включаючи:

- Переадресація дзвінків: Користувач може налаштувати умови для автоматичної переадресації дзвінків на інший номер або пристрій, наприклад, у випадку відсутності в мережі або зайнятості.

- Чорний список абонентів: Можливість створювати списки абонентів, до яких заборонено здійснювати дзвінки, а також надсилати їм певні повідомлення. Це дозволяє контролювати доступ до мережі та уникати небажаних дзвінків [6].

- Збереження даних розмови: IP-телефонія може зберігати дані розмови, включаючи записи аудіо або текстові повідомлення. Це дозволяє зберегти інформацію для подальшого аналізу, архівування або правових цілей.

- Білінгові функції: IP-телефонія забезпечує збір статистики дзвінків для окремих абонентів і груп абонентів. Оператори можуть отримувати дані щодо використання автоматичної телефонної станції, якості з'єднання та інші статистичні дані для цільового аналізу та планування.

- Голосова пошта: IP-телефонія дозволяє створювати голосові скриньки, в які користувачі можуть отримувати та зберігати голосові повідомлення.
- Конференц-дзвінки: Можливість об'єднання кількох абонентів у один дзвінок, що дозволяє проводити групові обговорення та ефективно зберігати комунікацію [19].
- Електронна передача факсу: Здатність передавати факсові повідомлення через IP-телефонію на телефон або комп'ютер, що забезпечує зручність та ефективність роботи.
- Колл-центри: IP-телефонія дозволяє створювати колл-центри для обробки великого потоку дзвінків, що надходять на певний телефонний номер. Це дозволяє забезпечити краще обслуговування клієнтів та організувати ефективну роботу операторів.
- Електронна черга в колл-центрах: Можливість створення електронної черги для керування потоком дзвінків в колл-центрі. Це дозволяє раціоналізувати процес обробки запитів і забезпечити кращий досвід користувача. Усі ці функції управління трафіком сприяють більш гнучкому та ефективному використанню IP-телефонії, що підвищує якість комунікаційних процесів та надає високий рівень обслуговування [40].

Недоліки IP-телефонії.

IP-телефонія, як і будь-яка інша технологія, має свої недоліки, серед яких основними є проблеми, пов'язані з якістю звуку та затримкою повідомлень. Ці проблеми виникають через особливості системи комутації пакетів, де голосові дані розбиваються на пакети та передаються різними каналами та групами. Це може спричинити неправильний порядок приймання пакетів, що впливає на затримку та якість звуку для користувачів.

Для подолання цих проблем використовуються різні методи. Один із них - зберігання отриманих пакетів та їх подальша організація у правильному порядку перед декодуванням та відтворенням звуку. Це дозволяє вирішити

проблему неправильного порядку пакетів та забезпечити більш якісний звуковий сигнал.

Проте існують інші проблеми, такі як помилки пакетів, втрата пакетів або перевантаження каналів, які можуть негативно позначитися на якості зв'язку. У випадку втрати пакетів може використовуватися математичне або алгоритмічне відновлення, але це може призвести до затримок або втрати деяких частин повідомлення.

Також, у систем IP-телефонії можуть виникати проблеми із ідентифікацією абонента через номер Caller ID (CID), де зловмисники можуть використовувати програми для перехоплення інформації Caller ID або відправляти викрадені дані Caller ID іншим сторонам, що може створити потенційні проблеми безпеки [13].

Крім того, системи IP-телефонії, які використовуються у службах екстреної допомоги, можуть стикатися з додатковими викликами, такими як затримки в обробці дзвінків аварійними службами через неможливість ефективно маршрутизувати дзвінок до найближчого центру викликів [31].

Недосконалість системи захисту терміналів також є проблемою, оскільки комп'ютери смартфонів, які використовуються як термінали IP-телефонії, можуть мати обмежені засоби захисту, що створює потенційну можливість для зловмисників отримати доступ до приватних даних користувачів.

Отже, IP-телефонія має свої недоліки, які можуть впливати на якість зв'язку і загальну безпеку системи.

Можна мати програмний телефон прямо на своєму комп'ютері. Інтеграція IP-АТС з бізнес-процесами, такими як системи CRM, може бути здійснена за допомогою API, яке дозволяє використовувати різноманітні функції, наприклад, інтеграцію з Google Maps для відображення місцезнаходження абонента та інші можливості.

На даний момент IP-телефонія є необхідною з кількох причин:

1. Економія коштів: IP-телефонія дозволяє значно економити на міжміських і міжнародних дзвінках, оскільки вона використовує широкосмстові канали, які є більш економічними, ніж традиційні телефонні лінії.

2. Мобільність: IP-телефонія дозволяє здійснювати телефонні розмови з будь-якого пристрою, підключеного до Інтернету.

3. Додаткові функції: IP-телефонія надає широкий спектр додаткових функцій, таких як голосова пошта, запис розмов, переадресація дзвінків, внутрішня телефонна книга, відеоконференції.

Також IP-телефонія корисна у таких випадках:

1. Для приватних користувачів: IP-телефонія може замінити традиційну домашню телефонію, дозволяючи економити на витратах на зв'язок та отримувати доступ до додаткових функцій, таких як голосова пошта та конференц-зв'язок.

2. Для бізнесу: Використання IP-телефонії допомагає створювати ефективну телефонну систему, що сприяє оптимізації витрат на зв'язок, підвищенню продуктивності співробітників та поліпшенню обслуговування клієнтів.

В сучасних умовах воєнного часу телефонія стає невід'ємною складовою для забезпечення зв'язку та координації важливих дій. У контексті воєнних операцій та надзвичайних ситуацій, де швидка передача інформації та ефективна комунікація є критичними, наявність надійної телефонної системи стає просто невід'ємною. Телефонія дозволяє оперативно передавати накази, координувати дії військових підрозділів, а також вести переговори між військовими командуваннями та державними органами.

Отже, IP-телефонія є сучасною та інноваційною технологією, яка завдяки своїм перевагам стає все більш популярною серед користувачів та підприємств, особливо з урахуванням тенденції переходу багатьох людей і компаній на Інтернет.

РОЗДІЛ 3. УСУНЕННЯ НЕСПРАВНОСТЕЙ В ІР ТЕЛЕФОНІЇ

Сучасна ІР-телефонія дозволяє використовувати будь-яку ІР-мережу для здійснення телефонних розмов, відеодзвінків та факсів в режимі реального часу. На сьогоднішній день, ІР-телефонія стала стандартом у галузі телефонних комунікацій, завдяки своїм перевагам, таким як зручність, надійність та відносно низькі витрати порівняно з аналоговими телефонними зв'язками. Крім того, вона підвищує продуктивність роботи організацій та дозволяє інтегрувати різні бізнес-додатки.

Проте сучасна ІР-телефонія вразлива до різних видів атак, таких як черв'яки та віруси, DoS-атаки, несанкціонований віддалений доступ і інші загрози. Основні ризики, які можуть вплинути на ІР-телефонну мережу, включають [6]:

- Реєстрацію на чужий термінал, що дозволяє виконувати дзвінки за чужий рахунок.
- Підміну абонента, що дає зловмиснику можливість перенаправляти дзвінки.
- Внесення змін до голосового або сигнального трафіку.
- Зниження якості голосового трафіку.
- Перенаправлення та перехоплення голосового або сигнального трафіку.
- Підроблення голосових повідомлень.
- Відмову в обслуговуванні.
- Віддалений несанкціонований доступ до інфраструктури ІР-телефонії.
- Забезпечення безпеки в ІР-телефонії стає все важливішим завданням для забезпечення надійності та приватності телефонних комунікацій [39].

Це лише кілька прикладів можливих проблем, пов'язаних із використанням ІР-телефонії. Альянс з безпеки VoIP визначив широкий спектр загроз для ІР-телефонії, який, крім технічних аспектів, також включає обман користувачів, непрохані повідомлення та інші аспекти. Розглядаючи проект ІР-телефонії, необхідно враховувати питання інформаційної безпеки, оскільки

саме на етапі підготовки проекту можна визначити, які заходи захисту мережевої інфраструктури слід вжити.

Один з найефективніших способів захисту від прослуховування - це використання IP-телефонів із вбудованими засобами шифрування даних. Крім цього, додатковий рівень захисту забезпечує шифрування трафіку між телефонами і шлюзами, що є логічним рішенням для захисту голосових розмов від прослуховування [19].

У випадку передачі мовних сигналів і даних через локальні віртуальні мережі, використовується загальна фізична пропускна здатність. При можливості зараження вузла вірусами або черв'яками може виникнути перепоповнення мережі трафіком. Проте, використання відповідно налаштованих механізмів QoS (Quality of Service) дозволяє трафіку IP-телефонії продовжувати мати пріоритет при проходженні через загальні фізичні канали, і DoS-атака буде неефективною.

Атаки типу «відмова в обслуговуванні» на застосування IP-телефонії (наприклад, на сервери оброблення дзвінків) і на середовище передачі даних є досить серйозною проблемою. Якщо йдеться про атаки на середови - ще передачі даних, то за нього у IP-телефонії відповідає протокол RTP (Real-Time Protocol) [18].

Сьогодні протокол SIP приходить на зміну протоколам H.323, проте він практично позбавлений будь-яких серйозних захисних функцій. Це змушує потенційних користувачів сумніватися в безхмарному майбутньому IP-телефонії, яку багато експертів пов'язують саме з протоколом SIP.

Усунення несправностей в IP-телефонії вимагає кількох кроків для визначення та вирішення проблеми. Ось декілька загальних порад для вирішення проблем з IP-телефонією [18]:

1. Перевірте з'єднання:

- Переконайтеся, що всі кабелі підключені правильно і не пошкоджені.

- Перевірте стан мережевого з'єднання на вашому IP-телефоні та маршрутизаторі. Наприклад, якщо працювати з IP-телефонією через WiFi-мережу, слід ретельно перевірити загальну потужність мережі, але навіть за ідеальних умов. Через чисельні причини неполадок, методи їх вирішення також досить варіативні – від налаштування, заміни, перестановки роутера до зміни каналу маршруту провайдера інтернету та регулювання потоку трафіку всередині мережі. Крім специфічних нюансів, під час роботи з телефонією через Wi-Fi можуть виникнути такі ж неполадки, як і за провідного інтернету. У окремих випадках, перевірити налаштування роутера, діапазон частот сигналу (2,4 чи 5 ГГц).

2. Перезавантажте обладнання:

- Спробуйте перезавантажити IP-телефон та мережеве обладнання (маршрутизатор, комутатор, DHCP-сервер тощо).

3. Перевірте налаштування IP-телефону [14]:

- Переконайтеся, що IP-телефон правильно налаштований для вашої мережі. Перевірте IP-адресу, підмережу, шлюз та DNS-сервери. Для перевірки IP-адреси, підмережі, шлюзу та DNS-серверів на вашому IP-телефоні, можна виконати наступні кроки, якщо ваш IP-телефон має графічний інтерфейс користувача:

1) Перевірка IP-адреси, підмережі та шлюзу:

a. На більшості IP-телефонів можна знайти цю інформацію в налаштуваннях системи або мережевих налаштуваннях. Зазвичай ця інформація позначається як "IP-адреса", "Підмережа" (або "Subnet Mask") та "Шлюз" (або "Gateway").

b. Зайдіть в меню налаштувань вашого IP-телефону, де зазвичай розташовані мережеві налаштування [9].

c. Відтуди ви зможете перевірити IP-адресу, підмережу та шлюз, які налаштовані на вашому IP-телефоні.

2) Перевірка DNS-серверів:

а. Для перевірки DNS-серверів також перейдіть до налаштувань мережі на вашому IP-телефоні.

б. У розділі налаштувань мережі, ви зможете перевірити DNS-сервери. Зазвичай ця інформація позначається як "DNS Server" або "Primary DNS" і "Secondary DNS".

3) Перевірка через командний рядок (опціонально):

Якщо у вас є можливість отримати доступ до командного рядка на вашому IP-телефоні, ви також можете використовувати команди для перевірки мережових налаштувань:

а. Для перевірки IP-адреси та підмережі введіть команду `ifconfig` (на Linux) або `ipconfig` (на Windows).

б. Для перевірки шлюзу введіть команду `netstat -nr` (на Linux) або `route print` (на Windows) [6].

с. Для перевірки DNS-серверів введіть команду `cat /etc/resolv.conf` (на Linux) або `ipconfig /all` (на Windows).

4. Перевірте послуги вашого постачальника IP-телефонії:

- Якщо у вас є послуга IP-телефонії від постачальника, переконайтеся, що ваш обліковий запис правильно налаштований і активний.

5. Відслідковуйте помилки та повідомлення:

6. Спостерігайте за будь-якими помилками чи повідомленнями на дисплеї IP-телефону, які можуть надати важливу інформацію щодо проблеми.

7. Перевірте мережевий трафік:

8. Можливо, проблема з якістю голосу виникає через високий обсяг мережевого трафіку. Перевірте, чи немає інших пристроїв, які використовують багато мережевого ресурсу, наприклад, завантажені файли або стрімінгові відеопослуги [6].

9. Оновіть програмне забезпечення:

10. Перевірте, чи є оновлення програмного забезпечення для вашого IP-телефону та оновіть його, якщо це необхідно.

11.Зверніться до служби підтримки:

12. Якщо ви не можете вирішити проблему самостійно, зверніться до служби підтримки вашого постачальника IP-телефонії або спеціалістів з ІТ, які можуть допомогти вирішити проблему.

Загальні поради можуть варіюватися в залежності від конкретних обставин та обладнання, яке використовується. Усунення несправностей в IP-телефонії може бути складним завданням, тому важливо ретельно діагностувати проблему та залучати фахівців при необхідності.

Також важливо проводити вчасно тестування на виявлення несправностей в IP телефонії. Це включає в себе використання різних інструментів моніторингу та аналізу трафіку, а також системи виявлення вторгнень (IDS). Розглянемо ці методи детальніше:

- Wireshark: Wireshark є одним з найпоширеніших інструментів аналізу мережевого трафіку. Цей інструмент дозволяє аналізувати дані, що передаються в мережі, і відстежувати їх з точки зору структури, джерела, призначення та інших параметрів. За допомогою Wireshark можна перехоплювати пакети даних, аналізуючи їх вміст та визначаючи можливі аномалії [32].

- Snort: Snort - це система виявлення вторгнень (IDS), спеціалізована на виявленні аномалій та потенційно небезпечних пакетів у мережі. Snort може перевіряти мережевий трафік на предмет відомих атак або аномалій, ініційованих зловмисниками. Якщо Snort виявляє підозрілий трафік, він може сповістити адміністратора про можливий інцидент. Ця система може виявляти атаки на IP телефонію, які можуть включати спроби несанкціонованого доступу або зловживання протоколами VoIP [23].

- Регулярне тестування: Важливо наголошувати на потребі регулярно тестувати систему IP телефонії на предмет вразливостей та несправностей.

- Загальна мета цих методів - забезпечити високий рівень безпеки та надійності IP телефонії, запобігаючи можливим атакам і виявляючи можливі несправності на ранніх стадіях.

ВИСНОВКИ

У кваліфікаційній роботі на тему "Дослідження IP телефонії на основі програмного забезпечення Asterisk" було успішно проведено дослідження і аналіз теоретичних основ IP телефонії, ретельно розглянуто історію та ключові поняття цієї технології. Ми також детально розглянули програмні можливості та функціонал платформи Asterisk, визначили її переваги та недоліки.

У першому розділі кваліфікаційної роботи розглянули історію та основні поняття IP телефонії. Ми доклали зусиль, щоб надати читачеві глибоке розуміння еволюції голосового зв'язку, починаючи з традиційних телефонних систем і закінчуючи сучасною IP телефонією. Визначили ключові терміни і концепції, такі як VoIP, SIP, H.323, і пояснили їх значення в контексті IP телефонії. Крім того, розглянули організацію та використання мережі IP-телефонії. Обговорили структуру IP-мережі та різні методи трансляції голосу через IP-мережу, зокрема розділення на пакети та маршрутизацію дзвінків. Також наголосив на перевагах та важливості IP-телефонії у сучасному світі зв'язку.

У другому розділі детально вивчили програмні можливості та функціонал платформи Asterisk. Був проведений дослід на базі Asterisk, було встановлено, що якість та продуктивність IP-телефонії на основі Asterisk відповідають вимогам, що пред'являються до телефонної зв'язку. Було розглянуто ключові аспекти цієї програми, зокрема її здатність до підтримки різних протоколів, маршрутизацію дзвінків, голосову пошту, конференц-зв'язок та інші функції. Також зазначили, що Asterisk є гнучкою та розширюваною комунікаційною платформою, з відкритим вихідним кодом. Визначили, що однією з головних переваг є гнучкість, яка дозволяє створювати різні типи комунікаційних рішень під різні завдання та вимоги клієнтів. Вказали на певні обмеження та недоліки, які можуть виникнути при використанні цієї платформи.

У третьому розділі кваліфікаційної роботи було досліджено "Усунення несправностей в IP телефонії". Метою цього дослідження було розглядання ключових аспектів, пов'язаних з виявленням, аналізом та виправленням різних видів несправностей у системах IP телефонії. Зазначений розділ надав можливість краще розібратися у складності та особливостях роботи цих систем. Розглянули основні типи несправностей, які можуть виникати в IP телефонії, таких як атаки DoS, втрати пакетів, перебої у з'єднанні, а також проблеми з якістю звуку. Кожен з цих аспектів був розглянутий з точки зору причин їх виникнення та можливих наслідків для користувачів.

Було розглянуто методи виявлення несправностей в IP телефонії. Ми визначили, що для цього можна використовувати різні інструменти моніторингу та аналізу трафіку, такі як Wireshark і Snort. Досліджено підходи до виправлення несправностей.

Дослідження бази IP-телефонії для компанії "ТОВ Infopulse" вказує на важливість і перспективи впровадження сучасних технологій зв'язку в робочому середовищі. Мережа IP-телефонії може вплинути на підвищення продуктивності, зниження витрат та поліпшення комунікації серед співробітників із різних локацій. Завдяки аналізу потреб, вибору відповідної системи IP-телефонії та належній підготовці інфраструктури, компанія "ТОВ Infopulse" готова до впровадження сучасних рішень у галузі зв'язку. Важливою є інтеграція з існуючими системами та забезпечення безпеки та надійності мережі IP-телефонії.

В сучасних умовах воєнного часу телефонія стає невід'ємною складовою для забезпечення зв'язку та координації важливих дій. Телефонія дозволяє оперативно передавати накази, координувати дії військових підрозділів, а також вести переговори між військовими командуваннями та державними органами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баричев С.Г. і ін. Основи сучасної криптографії. 2019. 144 с.
2. Вінградський В. А., Мельниченко А. С.. «ІР-телефонія: реалізація та налаштування VoIP-мережі». Комп'ютер Прес. 2014. С. 105.
3. Гольдштейн Б. С., Пінчук А. В., Суховицький А. Л. ІР-телефонія: Радіо і зв'язок, 2018.
4. Гольдштейн Б.С., Пинчук А.В., Суховицкий А.Л. ІР-телефония,. 2019. 336 с.
5. Грищенко С.М. Федотова-Півень І.М., ІР-телефонія в інформаційному просторі. Міжнародна науково-практична інтернет-конференція «Вдосконалення фінансово-кредитного механізму забезпечення інноваційного розвитку аграрного сектору економіки, сільських територій та країн V-4». 2022. С.257-259.
6. Колісник, І. І., Літвінов, О. В. "ІР-телефонія: підручник." Київський університет. 2019. 348 с.
7. Козір, В. О., Таран, І. П. "Технологія ІР-телефонії." Видавництво "Атіка". 2017. 428 с.
8. Левкуша О.В. «Розробка корпоративного сайту відеонагляду на основі РНР». Підручник. Державний університет телекомунікацій, Київ 2019р. 368 с.
9. Основи ІР-телефонії, базові принципи, терміни та протоколи. Ази ІРтелефонії від Cisco або знайомство з Cisco CallManager Express. URL: <https://habr.com/ru/articles/124411/> (дата звернення: 10.08.2023).
10. Офіційний сайт програмного забезпечення Asterisk. URL. <https://www.asterisk.org/>(дата звернення: 09.08.2023).
11. Поляков М. О. Шифрування трафіку в ІР телефонії. Наукові праці Четвертої міжнар. наук.-практ. конф. «Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій», 1–2 лютого 2022р. С.151-152.

12. Распутняк, В. І., Рябцев, С. О. "ІР-телефонія: мовленнєві технології на базі ІР-мереж." Видавництво ДУ "Інститут модернізації змісту освіти". 2016. С. 45.
13. Рубан, Д. В. "ІР-телефонія: підручник." Видавничий дім "Київський університет". 2019. 382 с.
14. Токар, Л. О. Особливості побудови віртуальної АТС. *Радіотехніка* , 2022. №1 (208), С. 55–64.
15. ТОП-7 сервісів віртуальних АТС для ІР телефонії в Україні. URL: <https://seoukraine.com.ua/> (дата звернення: 08.08.2023).
16. «Navicat Premium» URL: https://allsoft.ua/p1685921149-navicat-premium-enterprise.html?products_id=1685921149 (дата звернення: 15.06.2023).
17. «WebStorm» URL: <https://itpro.ua/product/jetbrains-webstorm/?tab=description> (дата звернення: 04.08.2023).
18. «PhpStorm» URL: <https://itpro.ua/product/jetbrains-phpstorm/?tab=description> (дата звернення: 05.08.2023).
19. «Android Studio» URL: <https://itproger.com/ua/course/java-android> (дата звернення: 05.08.2023).
20. Bhogal Amit; Hamza Idrissi; Thai-son Nguyen; Michael Wakahe “Voice over Internet Protocol”: URL: https://istu.edu.ua/wp-content/uploads/2022/11/Конференція_12-10-2022.pdf (Accessed : 28.08.2023)
21. Di Wu. “ Performance studies of VoIP over Ethernet LANs”, URL: <https://www.semanticscholar.org/paper/Performance-studies-of-VoIP-over-Ethernet-LANs-Nisar-Sarkar/c8a964da047b1ecdaad778264bbaed23a2656c84> (Accessed: 28.08.2023)
22. Greg S Tucker (2019) “Voice over Internet Protocol (VoIP) and security”. Journal of Information Engineering and Applications, No.4, 2013. URL: <https://core.ac.uk/download/pdf/234677003.pdf> (Accessed: 30.07.2023)
23. Hersent Oliver (2021) “IP telephony: deploying VoIP protocol and IMS Infrastructure”, Second Edition, John Wiley and sons, UK, P. 42-46.

24. Jeomgoo Kim, Inyong Lee, Suh ron Noh (2020) “VoIP Quality of Service Design of measurement management process model”, International Conference on information science and applications (ICISA). P 1-6. URL: <https://scribd.com/document/541230263/VoIP>. (Accessed: 30.07.2023)
25. Mona Habib, Nirmala Bulusu (2020), “Improving QOS of VoIP over WLAN (IQ-VW)”. URL: <https://www.readkong.com/page/improving-qos-of-voip-over-wlan-iq-vw-3407061> (Accessed: 05.08.2023)
26. Rakesh Arora (2019) “Voice over IP: Protocol and Standards”, URL: https://www.academia.edu/10604859/Voice_over_Internet_Protocol_VOIP_Overview_Direction_And_Challenges (Accessed; 05.08.2023)
27. Maresca, M., Z. Nicola, and P. Baglietoo, Internet Protocol Support for Telephony. Proceedings of IEEE, 2018. 92(9): p. 1463-1477.
28. Thom, G.A., H.323: The Multimedia Communications Standard for Local Area Networks. Ieee Communications Magazine, 2016. 34(12): p. 52-56.
29. Rosenberg, J., et al., RFC 3261: SIP: Session Initiation Protocol. 2020, IETF.
30. Nocentini, S. and M. Siviero, Innovative class 5: A challenge for incumbent network operators. Journal of the Communications Network, 2020. 1: p. 52-55.
31. Lovell, D., Cisco IP Telephony. 2017, Indianapolis, IN: Cisco Press.
32. Markopoulou, A.P., F.A. Tobagi, and M.J. Karam, Assessing the quality of voice communications over Internet backbones. Ieee-Acm Transactions on Networking, 2017. 11(5): p. 747-760.
33. Handley, M. and V. Jacobson, RFC 2327: SDP: Session Description Protocol. 2018, IETF.
34. Arango, M. and C. Huitema, Simple gateway control protocol (SGCP) Version 1.0. 2018. URL: <https://datatracker.ietf.org/doc/html/draft-huitema-sgcp-v1-02.txt> (Accessed; 15.08.2023)
35. C.T.L., PacketCable Network-Based Call Signaling Protocol Specification. 2019. URL:

https://www.etsi.org/deliver/etsi_en/302800_302899/30287804/01.01.00_30/en_30287804v010100v.pdf (Accessed; 15.08.2023)

36. Schulzrinne, H. and J. Rosenberg, Internet telephony: architecture and protocols - an IETF perspective. *Computer Networks-the International Journal of Computer and Telecommunications Networking*, 2019. 31(3): p. 237-255.

37. Goode B., Voice over Internet protocol (VoIP). *Proceedings of the Ieee*, 2021. 90(9): p. 1495-1517.

38. Saltzer, J., D. Reed, and D. Clark, End-to-end arguments in system design. *ACM Transactions of Computer Systems*. 2(4): p. 277-288.

39. Stermann, J., *Business dynamics: systems thinking and modeling for a complex world*. 2020, Boston: Irwin/McGraw-Hill. URL: https://www.researchgate.net/publication/44827001_Business_Dynamics_System_Thinking_and_Modeling_for_a_Complex_World (Accessed; 15.09.2023)

40. Y.G., *Discovering VoIP Profitability*, in *America's Network*. 2018. p. 220.

ДОДАТКИ

Додаток А

Сайт з кодеками Asterisk

- [Sources](#)
- [Binaries](#)
- [Notes and Troubleshooting](#)
- [Getting help](#)

Sources

To compile the codecs it is recommended to install [Intel IPP](#) libraries for better performance. Asterisk 1.4 to 18 are supported.

- [asterisk-g72x-1.4.3.tar.bz2](#) - supports Asterisk 1.4 to 18, [Bcg729](#) 1.0.2
- [README](#)
- [git clone https://bitbucket.org/arkadi/asterisk-g72x.git](#)
- [git clone https://github.com/arkadijs/asterisk-g72x.git](#)

Binaries

1. choose codec binary appropriate for your Asterisk version and CPU type, use x86_64 for 64-bit mode
2. delete old codec_g72[39]*.so files (if any) from /usr/lib/asterisk/modules directory
3. copy new codec_g72[39]*.so files into /usr/lib/asterisk/modules directory
4. restart Asterisk
5. check the codec is loaded with 'core show translation recalc 10' on Asterisk console
6. G.723.1 send rate is configured in Asterisk codecs.conf file:

```
[g723]
; 6.3Kbps stream, default
sendrate=63
; 5.3Kbps
; sendrate=53
```

This option is for outgoing voice stream only. It does not affect incoming stream that should be decoded automatically whatever the bitrate is.

7. in sip.conf or/and iax.conf configure the codec(s) either globally or under respective peer, for example:

```
disallow=all
allow=g729
```

Linux binaries

Asterisk 1.8

Asterisk 11

Asterisk 12

Asterisk 13

Asterisk 14

Asterisk 15

Asterisk 16

Asterisk 17

Asterisk 18

Використання кодиків сервісом Asterisk

Left	File	Command	Options
<	/usr/lib/asterisk/modules		
.n	Name		
*cel_pgsq	l.so		
*chan_agent	.so		
*chan_bridge	.so		
*chan_iax2	.so		
*chan_local	.so		
*chan_mgcp	.so		
*chan_multicast_rtp	.so		
*chan_oss	.so		
*chan_phone	.so		
*chan_sip	.so		
*chan_skinny	.so		
*chan_unistim	.so		
*codec_a_mu	.so		
*codec_adpcm	.so		
*codec_alaw	.so		
*codec_g722	.so		
*codec_g723	.so		
*codec_g726	.so		
*codec_g729	.so		
*codec_gsm	.so		
*codec_ilbc	.so		
*codec_lpc10	.so		
*codec_resample	.so		
*codec_ulaw	.so		
*format_g719	.so		
*codec_g729	.so		

Додаток В

```

const WebSocketClient = require('websocket').client; const fs =
require('fs');
var clc = require('cli-color');
var moment = require('moment');

const https = require('https'); var Promise = require('bluebird');

var zvonok= function() { id=-1,

start_call=-1,   answer_call=-1,   end_call=-1,   status=-1,   caller_num=-1,
who_answer_call=-1,
call_card_id=-1
};

var telefon = function(){ Caller_id=-1, Timestamp=-1,
Name='', State='NOT_INUSE'

};
var calls_count = 0;
var last_zvonok = 0;
var zvonki=[];
var telefoni=[];
var taskServer = 'ipacs.knu.ua/?lang=ukr:****',

username = '*****',

call_card_id=-1
};

var telefon = function(){ Caller_id=-1, Timestamp=-1,
Name='', State='NOT_INUSE'

};
var calls_count = 0;
var last_zvonok = 0;
var zvonki=[];
var telefoni=[];
var taskServer = 'ipacs.knu.ua/?lang=ukr:****',

username = '*****',

password = '*****_*****_*****_*****',

token = '';
var client = new WebSocketClient();
client.connect('ws://***.***.***.***:*****/ari/events?app=CENTRAL&subscribeAll
=t rue&api_key= your_api_key:*****');

client.on('connectFailed', function(error) { WriteLogError('Connect Error: '
+ error.toString());

});

```

```

client.on('connect', function(connection) { WriteLog1('WebSocket Client
Connected');

fs.readFile(__dirname + "/token", "utf8", function(err,data) { if(err) {

return console.log(err); } else {

token=data;

WriteLog("Used token "+token); }

});
connection.on('error', function(error) {

WriteLogError("Connection Error: " + error.toString()); });

connection.on('close', function() {

WriteLog1('echo-protocol Connection Closed'); });

connection.on('message', function(message) { if (message.type === 'utf8') {

showinfo(JSON.parse(message.utf8Data)); }

}); });

function showinfo(d) { try {

//if (d.type=='Dial' || d.type=='ChannelConnectedLine' ||
d.type=='BridgeCreated' || d.type=='ChannelEnteredBridge' ||
d.type=='ChannelHangupRequest' || d.type=='BridgeDestroyed') {

//if (d.type=='ChannelHangupRequest' || d.type=='BridgeDestroyed') {

if (d.type=='Dial' || d.type=='DeviceStateChanged') {
if(d.type=='DeviceStateChanged'){

if(typeof telefoni[d.device_state.name] !== 'undefined'){

if(d.device_state.state=="NOT_INUSE" &&
telefoni[d.device_state.name].state=="Up") {

zvonki[telefoni[d.device_state.name].Caller_id].end_call=d.timestamp;
zvonki[telefoni[d.device_state.name].Caller_id].status='END';
telefoni[d.device_state.name].state=d.device_state.state;

calls_count--;
//console.log(zvonki);
//console.log(telefoni);
WriteLog('CALL END Total calls - '+calls_count+'

'+telefoni[d.device_state.name].Caller_id.toString());
if(zvonki[telefoni[d.device_state.name].Caller_id].call_card_id!=--

1){
UpdateCallCard(zvonki[telefoni[d.device_state.name].Caller_id].call_card_id,

//console.log(zvonki[telefoni[d.device_state.name].Caller_id])

```

```

        '1S1',
        'Оператор',

        zvonki[telefoni[d.device_state.name].Caller_id].start_call,

        zvonki[telefoni[d.device_state.name].Caller_id].end_call, "Екстрений",

        "Виклик працівників",
        "Працюємо у тестовому режимі");

    }
    } else if (d.device_state.state=="NOT_INUSE" &&

        zvonki[telefoni[d.device_state.name].Caller_id].status!='END') {
        //telefoni[d.device_state.name].state=d.device_state.state;

        //console.log(d); //console.log(zvonki); //console.log(telefoni);

    } }

    }

    if(d.type=='Dial' && d.dialstatus=='CANCEL' && typeof zvonki[d.caller.id] !==
    'undefined'){

        //zvonki[d.caller.id].status=d.dialstatus;
        //calls_count--;
        WriteLog('CALL CANCEL Total calls - '+calls_count+' '+d.caller.id);
        /*if(zvonki[d.caller.id].call_card_id!=-1){

            UpdateCallCard(zvonki[d.caller.id].call_card_id, '1S1',

            'Оператор', zvonki[d.caller.id].start_call, d.timestamp,
            "Не визначено",
            "Помилковий виклик",
            "Працюємо у тестовому режимі");

        }*/

    } else { //console.log(d);

    }

    if(d.type=='Dial' && d.dialstatus==''){
    if(typeof zvonki[d.caller.id] === 'undefined') {

        // does not exist

        if(typeof telefoni[d.peer.name.split('-')[0]] === 'undefined' ||
        telefoni[d.peer.name.split('-')[0]]=='NOT_INUSE') {

            var tf = new telefon(); tf.Caller_id=d.caller.id; tf.name=d.peer.caller.name;
            tf.state=d.peer.state; telefoni[d.peer.name.split('-')[0]]=tf;

            //console.log(telefoni);

```

```

} else { telefoni[d.peer.name.split('-')[0]].Caller_id=d.caller.id;
telefoni[d.peer.name.split('-')[0]].name=d.peer.caller.name;
telefoni[d.peer.name.split('-')[0]].state=d.peer.state;

}

var zv = new zvonok(); zv.id=d.caller.id; zv.start_call=d.timestamp;
zv.status=d.caller.state; zv.caller_num=d.caller.caller.number;
zvonki[d.caller.id]=zv; last_zvonok=zvonki.length;
delete zv;

delete tf;
//console.log(zvonki)
calls_count++;
WriteLog('NEW CALL Total calls - '+calls_count+' '+d.caller.id);

}
else {

// does exist

zvonki[d.caller.id].status=d.caller.state;

//console.log(zvonki);

} } if(d.type=='Dial' && d.dialstatus=='ANSWER'){ if(typeof
zvonki[d.caller.id] === 'undefined') {

// does not exist

}
else {

// does exist

zvonki[d.caller.id].status=d.dialstatus;
zvonki[d.caller.id].answer_call=d.timestamp;
zvonki[d.caller.id].who_answer_call=d.peer.name;
WriteLog('ANSWER CALL Total calls - '+calls_count+' '+d.caller.id);
//console.log(zvonki);

//console.log(telefoni);

CreateCallCard('1S1',d.caller.id,zvonki[d.caller.id].caller_num,zvonki[d.caller.id].start_call,zvonki[d.caller.id].answer_call);

if(typeof telefoni[d.peer.name.split('-')[0]] === 'undefined' ||

telefoni[d.peer.name.split('-')[0]]=='NOT_INUSE') { var tf = new telefon();

tf.Caller_id=d.caller.id; tf.name=d.peer.caller.name; tf.state=d.peer.state;
telefoni[d.peer.name.split('-')[0]]=tf;

} else{ telefoni[d.peer.name.split('-')[0]].state=d.peer.state;

}

//console.log(zvonki);

//console.log(telefoni);

```

```

    } }

    //console.log(d);
    //if(d.dialstatus == undefined) {
    // console.log(d);
    //}
    } else if (d.type!='PeerStatusChange' && d.type!='ChannelDialplan' &&
d.type!='ChannelVarset') { //fs.appendFileSync('deleted.txt',
`${newlineChar}${JSON.stringify(d.timestamp+" "+d.type)}`);
//console.log(d.timestamp+" "+d.type);

    }

    } catch (e){
    console.log('error '+d+"|"+e);

    } }

function

CreateCallCard(operator_id,mis_call_card_id,caller_number,start_datetime,end_
datetime) {

var current_Startdatetime = start_datetime.split("+"); var
current_Enddatetime = end_datetime.split("+");
var getDate = start_datetime.split("T")
var card = mis_call_card_id.split(".")

var MisCallCard ="CARD-"+getDate[0]+"-"+ card[0]
SendPostCmd('/api/callcard/',{

    "CallCard": [

    {
    "operator_id": operator_id,
    "mis_call_card_id": MisCallCard,
    "caller_number": (caller_number === "unknown")? "0": caller_number ,
    "call_station": "PBX",
    "start_datetime": current_Startdatetime[0],
    "end_datetime": current_Enddatetime[0]

    } ]

    })
    .then(function (resp) {

JSON.parse(resp,function (key,val) { //console.log(val)

if(key=='call_card_id') { zvonki[mis_call_card_id].call_card_id=val;

    } })

WriteLog("CallCard created "+mis_call_card_id); })

    .catch(function (err) {
WriteLogError('CallCard create ошибка: '+err);

```

```

fs.readFile(__dirname + "/token", "utf8", function(err,data) { if(err) {

return console.log(err); } else {

token=data;

}

WriteLog("Used token "+token); }

}); })

function

UpdateCallCard(call_card_id,operator_id,call_station,start_datetime,end_datet
ime,all_priority,call_result,call_comment) {

var current_Startdatetime = start_datetime.split("+");
var current_Enddatetime = end_datetime.split("+");
if(call_card_id !== undefined && call_card_id !== "undefined" && call_card_id

!== null) {
SendPutCmd('/api/callcard/' + call_card_id, {

    "CallCard": [
        {

"operator_id": operator_id,
"call_station": call_station, "start_datetime": current_Startdatetime[0],
"end_datetime": current_Enddatetime[0], "call_priority": all_priority,
"call_result": call_result,
"call_comment": call_comment,

} ]

})

.then(function (resp) {

JSON.parse(resp, function (key, val) {

})

WriteLog("CallCard updated /api/callcard/" + call_card_id); })

.catch(function (err) {
WriteLogError('CallCard update ошибка: ' + err);

}) })

}

function WriteLog(text) {

var dt = moment().format('YYYY-MM-DD HH:mm:ss');

console.log('%s | %s\r', dt,text); }

function WriteLogError(text) {
var dt = moment().format('YYYY-MM-DD HH:mm:ss');

```

```

    console.log(dt+clc.red(' | '+text)); }

function WriteLog1(text) {
var dt = moment().format('YYYY-MM-DD HH:mm:ss').replace(/T/, ' ');

console.log(dt+clc.green(' | '+text)); }

function WriteLog2(text) {
var dt = moment().format('YYYY-MM-DD HH:mm:ss').replace(/T/, ' ');
console.log(dt+clc.yellow(' | '+text));

}

function SendPostCmd(addr, text) {
return new Promise(function (resolve,reject) {

var postData = JSON.stringify(text); console.log("SendPostCmd : "+ addr+" "
+postData) var answer = undefined;

var options1 = {
hostname: taskServer.split(':')[0],

port: taskServer.split(':')[1], path: addr,
method: 'POST', rejectUnauthorized : false, headers: {

'Content-Type': 'application/json', 'Content-Length':
Buffer.byteLength(postData), 'Authorization': 'JWT '+token

} });

var req = https.request(options1, function(res) { res.setEncoding('utf8');
res.on('data', function(data){

        if(data){
            answer = data;

//WriteLog1('Got answer: '+data);

        } else { }

    });
res.on('end', function(){

if(res.statusCode===400 || res.statusCode===401)

reject(res.statusCode); resolve(answer);

});

}).on('error', function(err){ reject(err);

})

req.write(postData,'utf8',function () { //WriteLog('Sent: '+postData);

});

    req.end();

```

```

    })

}

function SendPutCmd(addr, text) {
return new Promise(function (resolve, reject) {

var postData = JSON.stringify(text); console.log("SendPutCmd : "+ addr+"
"+postData) var answer = undefined;

var options1 = {
hostname: taskServer.split(':')[0], port: taskServer.split(':')[1],

method: 'PUT', rejectUnauthorized : false, headers: {

'Content-Type': 'application/json', 'Content-Length':
Buffer.byteLength(postData), 'Authorization': 'JWT '+token

} });

var req = https.request(options1, function(res) { res.setEncoding('utf8');
res.on('data', function(data) {

        if(data) {

            answer = data;

path: addr,

//WriteLog1('Got answer: '+data);

} else { }

});
res.on('end', function() {

if(res.statusCode==400 || res.statusCode==401) reject(res.statusCode);

            resolve(answer);
        });

}).on('error', function(err) { reject(err);

});

req.write(postData, 'utf8', function () { //WriteLog('Sent: '+postData);

});

    req.end();
})

```