МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ,
ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ ТА РАДІОЕЛЕКТРОННИХ СИСТЕМ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Роман ОДАРЧЕНКО
“_____”_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

Тема: «Забезпечення QoS в мережі передачі даних засобами ОС Linux»

Виконавець: _____ Олександр ЗЕЛІНСЬКИЙ
(підпис)

Керівник: _____ Віталій КУРУШКІН
(підпис)

Нормоконтролер: _____ Денис БАХТІЯРОВ
(підпис)

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки та телекомунікацій
Кафедра телекомунікаційних та радіоелектронних систем
Спеціальність 172 «Телекомунікації та радіотехніка»
Освітньо-професійна програма «Телекомунікаційні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри

Роман ОДАРЧЕНКО
“_____”_____2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Зелінського Олександра Олеговича
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи: «Забезпечення QoS в мережі передачі даних засобами ОС Linux»
затверджена наказом ректора від «29» березня 2023 р. № 421/ст
2. Термін виконання роботи: з 22.05.2023 р. по 25.06.2023 р.
3. Вихідні дані до роботи: операційна система Linux для завдань дослідження QoS
4. Зміст пояснювальної записки: операційна система Linux у телекомунікаціях, забезпечення QoS в ОС Linux, розрахунок якості обслуговування IP телефонії
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: слайди презентації за матеріалами кваліфікаційної роботи в програмному пакеті Microsoft PowerPoint 2021.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Розробити деталізований зміст розділів кваліфікаційної роботи	22.05.2023-24.05.2023	Виконано
2	Вступ	25.05.2023	Виконано
3	Операційна система Linux у телекомунікаціях	26.05.2023-29.05.2023	Виконано
4	Забезпечення QoS в ОС Linux	30.05.2023-07.06.2023	Виконано
5	Розрахунок якості обслуговування IP телефонії	08.06.2023-14.06.2023	Виконано
6	Усунення недоліків та захист кваліфікаційної роботи	15.06.2023-25.06.2023	Виконано

7. Дата видачі завдання: “19" травня 2023 р.

Керівник кваліфікаційної роботи _____ Віталій КУРУШКІН
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Олександр ЗЕЛІНСЬКИЙ
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Кваліфікаційна робота «Забезпечення QoS в мережі передачі даних засобами ОС Linux» містить 81 сторінка, 7 рисунків, 3 таблиці, 30 використаних джерел.

ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ (QUALITY OF SERVICE, QOS), МЕРЕЖА ПЕРЕДАЧІ ДАНИХ, ОПЕРАЦІЙНА СИСТЕМА LINUX, ТРАФІК МЕРЕЖІ, ПЕРЕДАЧА ДАНИХ, ПРІОРИТЕТИ ТРАФІКУ, КЕРУВАННЯ ПРОПУСКНОЮ ЗДАТНІСТЮ, РОЗДІЛЕННЯ ТРАФІКУ, ПЕРЕДАЧА ГОЛОСУ ТА ВІДЕО, СПРАВЕДЛИВЕ РОЗПОДІЛ РЕСУРСІВ, TRAFFIC SHAPING (ФОРМУВАННЯ ТРАФІКУ), BANDWIDTH MANAGEMENT (УПРАВЛІННЯ ПРОПУСКНОЮ ЗДАТНІСТЮ), NETWORK CONGESTION (МЕРЕЖЕВА ЗАТОРУ), PACKET SCHEDULING (ПЛАНУВАННЯ ПАКЕТІВ), DIFFERENTIATED SERVICES (РІЗНОВИДНІСТЬ СЕРВІСІВ), LINUX KERNEL (ЯДЕРНА СИСТЕМА LINUX), NETWORK PROTOCOLS (МЕРЕЖЕВІ ПРОТОКОЛИ), QUALITY METRICS (МЕТРИКИ ЯКОСТІ), LATENCY (ЗАТРИМКА), THROUGHPUT (ПРОПУСКНА ЗДАТНІСТЬ).

Мета кваліфікаційної роботи – збільшення розуміння та знань щодо забезпечення якості обслуговування в мережах на базі ОС Linux, а також виявлення можливостей та обмежень засобів QoS у цьому середовищі.

Об’єктом дослідження – є процес забезпечення якості обслуговування (QoS) в мережі передачі даних, з використанням засобів, які надає операційна система Linux.

Предметом дослідження – є специфічні засоби та механізми, які надає операційна система Linux для забезпечення якості обслуговування (QoS) у мережевому середовищі.

Практичне значення отриманих результатів. Отримані результати можуть бути використані як рекомендації для системних адміністраторів, мережевих інженерів та розробників програмного забезпечення для досягнення кращої продуктивності, ефективності та задоволення потреб користувачів в мережевому середовищі з використанням ОС Linux.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

QoS – Якість обслуговування.

S**haping** – (шейпінг) контролінг швидкості передачі. Тут іде процес згладжування трафіку. І шейпінг застосовується тільки для вихідного трафіку.

Scheduling – (шедулінг) Контроль черги відправлення пакетів. Завдяки зміні черговості пакетів, що стоять у черзі на відправлення, можна підвищити якість обслуговування інтерактивного трафіку без шкоди для передавання пакетного трафіку, нечутливого до затримок.

Policing – Лімітування трафіку. Ліквідування трафіку в разі перевищень цієї величини.

Dropping - Marking – Маркування пакетів.

Classification – Класифікація пакетів.

Qdiscs – Дисципліни обслуговування.

Classes – Класи обслуговування.

Filters – Фільтри. Їхнє завдання систематизувати трафік. Трафік, який прибуває знову, не має відношення до жодного з класів, і його систематизація проводиться за допомогою фільтрів. Існують низки різних фільтрів трафіку, що відрізняються здібностями і принципом роботи.

Широкомовна адреса – IP-адреса зі складовою вузла з одних одиниць.

Вузол – комп'ютер або інший пристрій у мережі TCP/IP.

Інтернет – глобальна сукупність з'єднаних між собою мереж із загальним діапазоном IP-адрес.

InterNIC – організація, що займається адмініструванням IP-адрес в Інтернеті.

IP – мережевий протокол, який використовується для надсилання мережевих пакетів мережею TCP/IP або Інтернетом.

IP-адреса – унікальна 32-розрядна адреса вузла в мережі TCP/IP або загальнодоступній мережі.

OS – Operating System (операційна система).

Linux – Лінукс (операційна система Linux).

ISP – Internet Service Provider (провайдер інтернет-послуг).

VLAN – Virtual Local Area Network (віртуальна локальна мережа).

LAN – Local Area Network (локальна область мережі).

WAN – Wide Area Network (широкообхватна мережа).

VoIP – Voice over Internet Protocol (голосова передача даних через інтернет).

ICMP – Internet Control Message Protocol (протокол керування повідомленнями інтернету).

QoS parameters – параметри якості обслуговування.

DSCP – Differentiated Services Code Point (кодова точка різновидних сервісів).

RSVP – Resource Reservation Protocol (протокол резервування ресурсів).

MTU – Maximum Transmission Unit (максимальна одиниця передачі).

BW – Bandwidth (пропускна здатність).

Пакет – одиниця даних, що передається через мережу TCP/IP або глобальну мережу.

RFC (Request for Comment) – документ, що визначає стандарти в Інтернеті.

Маршрутизатор – пристрій, що забезпечує обмін даними між різними IP-мережами.

TCP/IP – набір протоколів, стандартів і програм, що широко використовується в Інтернеті та інших великих мережах.

ВСТУП

Актуальність теми. Сучасне суспільство володіє величезними ресурсами, найважливішим з яких стала інформація. Найвище зростання обсягів інформаційних потоків поставило перед суспільством нові завдання, пов'язані зі зберіганням, опрацюванням, охороною і швидким передаванням інформації. Для вирішення цих завдань було використано комп'ютерні технології.

По всьому світу більшість комп'ютерів об'єднані в різні інформаційно-обчислювальні мережі: від малих локальних мереж в офісах до глобальних мереж на кшталт Інтернет.

Отже, актуальним завданням для будь-яких підприємств є під'єднання до тих чи інших інформаційних ресурсів, які дають змогу отримувати достовірну, найсвіжішу інформацію, необхідну в бізнес-процесах компанії.

Крім доступу до мережі Інтернет, обміну файлами та корпоративною поштою, широко стали застосовуватися і мультимедійні можливості широкосмугового доступу в мережах. Практично будь-яка компанія має можливість використовувати IP-телефонію і відеоконференції. Ці мережеві послуги досить специфічні і вимогливі до якості каналів зв'язку. Проте вони забезпечують відмінні комунікаційні можливості всередині компанії, прискорюють обмін інформацією, спрощують роботу співробітників.

У великих холдингах і компаніях середнього розміру IP-телефонія вже кілька років успішно використовується і розвивається. Але завжди залишається актуальним питання про забезпечення внутрішньокорпоративним телефонним зв'язком віддалених філій, що знаходяться в передмісті великих мегаполісів, таких як Київ. На жаль, ширина каналів доступу до Інтернет або VPN-мереж у цих віддалених районах залишає бажати кращого. Таким чином, для забезпечення прийнятного рівня якості VOIP-зв'язку необхідно використовувати програмно-апаратні механізми QoS, широко представлені різними виробниками програмного та апаратного забезпечення.

В умовах поточної економічної кризи спричиненої воєнними діями та нестабільної економічної ситуації загалом, серед широкого різноманіття таких рішень, вибір кінцевих користувачів зупиняється на системах, реалізованих на основі відкритого програмного забезпечення (архітектура Linux). Такі рішення дають змогу реалізовувати безпечну, потужну, бюджетну та надійну архітектуру IP-телефонії, що має багатий функціонал і подальші перспективи розвитку та масштабування.

Тема "Забезпечення QoS в мережі передачі даних засобами ОС Linux" є актуальною навіть на сьогоднішній день. Linux є однією з найпопулярніших операційних систем, використовуваних в мережевих оточеннях. Забезпечення якості обслуговування (QoS) є критичним аспектом для забезпечення надійної та ефективної передачі даних в мережі.

Засоби QoS в Linux дозволяють призначати пріоритети різним типам трафіку, контролювати пропускну здатність, обмежувати витрати ресурсів та керувати трафіком відповідно до встановлених правил. Це особливо важливо в сучасних мережах, де переважають різні типи даних, такі як голосовий трафік, відео, потокове відео, інтерактивні додатки тощо.

Застосування засобів QoS в Linux дозволяє забезпечити гарантовану якість обслуговування для важливих додатків, зменшити затримки та втрати пакетів, підвищити надійність та продуктивність мережі. Це стає особливо корисним у великих корпоративних мережах, дата-центрах, провайдерських мережах та інших критичних застосуваннях.

Отже, дослідження та використання засобів QoS в Linux є актуальними для забезпечення ефективної та надійної мережевої передачі даних [1-30].

Мета і завдання дослідження. ***Мета кваліфікаційної роботи*** – збільшення розуміння та знань щодо забезпечення якості обслуговування в мережах на базі ОС Linux, а також виявлення можливостей та обмежень засобів QoS у цьому середовищі.

Для досягнення поставленої мети вирішуються такі завдання.

- Вивчити основні принципи та концепції забезпечення якості обслуговування (QoS) в мережевих системах.

- Ознайомитись зі засобами та механізмами QoS, доступними в операційній системі Linux, зокрема у різних дистрибутивах.
- Дослідити можливості налаштування та управління пріоритетами, бандвідтом, затримками та іншими параметрами трафіку в Linux для досягнення бажаної якості обслуговування.
- Перевірити ефективність та ефективність застосування засобів QoS в Linux через експериментальні дослідження та вимірювання показників якості обслуговування.
- Зробити висновки щодо придатності та ефективності засобів QoS в Linux та їх застосування у реальних мережевих сценаріях.
- Запропонувати можливі рекомендації щодо оптимального використання засобів QoS в Linux та можливих напрямків подальших досліджень у цій області.

Об'єктом дослідження – є процес забезпечення якості обслуговування (QoS) в мережі передачі даних, з використанням засобів, які надає операційна система Linux.

Предметом дослідження – є специфічні засоби та механізми, які надає операційна система Linux для забезпечення якості обслуговування (QoS) у мережевому середовищі.

Методи досліджень. При дослідженні теми "Забезпечення QoS в мережі передачі даних засобами ОС Linux" використані наступні методи:

- Літературний аналіз: Проведення докладного перегляду наукових статей, книг, документації та інших джерел, що стосуються забезпечення QoS в мережах з використанням засобів Linux. Це дозволить отримати теоретичні знання про основні концепції, принципи та практики QoS в Linux.
- Експериментальні дослідження: Здійснення практичних експериментів для вивчення функціональності та ефективності засобів QoS в Linux. Це може включати налаштування різних параметрів QoS, вимірювання показників якості обслуговування та порівняння результатів з різними налаштуваннями.

- Випробування реальних сценаріїв: Аналіз та випробування засобів QoS в реальних мережевих сценаріях, таких як корпоративні мережі, провайдерські мережі або дата-центри. Це дозволяє оцінити придатність та ефективність засобів QoS в реальних умовах експлуатації.
- Моделювання та симуляція: Використання комп'ютерних програм або інструментів для моделювання та симуляції різних сценаріїв мережі з використанням засобів QoS в Linux. Це дозволяє провести дослідження та оцінку різних варіантів налаштувань QoS без необхідності в реальних мережевих середовищах.
- Аналіз даних та статистична обробка: Обробка отриманих даних з експериментів, випробувань та моделювання з використанням методів статистики та аналізу даних. Це дозволяє встановити залежності, виявити тенденції та зробити об'єктивні висновки з отриманих результатів дослідження.

Комбінація цих методів дозволяє детально дослідити та оцінити засоби QoS в Linux, їхню придатність у різних сценаріях мережі та ефективність у досягненні бажаної якості обслуговування.

Практичне значення отриманих результатів.

Отримані результати можуть бути використані як рекомендації для системних адміністраторів, мережевих інженерів та розробників програмного забезпечення для досягнення кращої продуктивності, ефективності та задоволення потреб користувачів в мережевому середовищі з використанням ОС Linux.

Апробація отриманих результатів. Основні положення роботи доповідалися та обговорювалися на таких конференціях:

- Науково-практична конференція «Проблеми експлуатації та захисту інформаційно-комунікаційних систем», м. Київ, 2023 р.

РОЗДІЛ 1
ОПЕРАЦІЙНА СИСТЕМА LINUX У ТЕЛЕКОМУНІКАЦІЯХ

Складання IT-інфраструктури будь-якої сучасної компанії зазвичай починається з підбору інфраструктурного програмного забезпечення, а саме з підбору операційної системи. Операційна система - це сукупність програм, що власне гарантує управління апаратними засобами ПК, утворює роботу з файлами (таким як пуск і регулювання виконанням програм), також здійснює взаємозв'язок з користувачем команд та вирішення результатів оброблення цих команд. Без операторної концепції ПК загалом ніяк не здатний працювати у вигляді такого. У цьому разі він являє собою не менше ніж ансамбль непрацюючих електричних пристосувань, неясно навіщо складених в одне ціле.

У реальний період операторними концепціями для ПК, що користуються найбільшою популярністю, є такі операторні системи, як Microsoft Windows і UNIX. Початкові ведуть свій родовід з операторної системи MS-DOS, якою забезпечувалися початкові індивідуальні ПК фірми IBM. Операційну систему UNIX розробила команда співробітників Bell Labs під керівництвом Денніса Річі, Кена Томпсона і Браяна Кернігана (Dennis Ritchie, Ken Thompson, Brian Kernighan) у 1969 р. Але в наші часи, коли говорять про операційну систему UNIX, загалом мають на увазі не конкретну ОС, а точніше - єдину сукупність UNIX-аналогічних операційних систем. А сам термін UNIX, як випливає, є зареєстрованою торговою маркою організації AT&T.

Операційні системи, що повторюють вигляд UNIX, навіть BSD, спочатку розроблялися з метою роботи у великих багатокористувацьких ПК - мейнфреймах. Але індивідуальні комп'ютери з плином часу підвищували міць особистого апаратного забезпечення, і в наш час вони наразі перевершують за здібностями ті мейнфрейми, для яких у 70-х роках розроблялася ОС UNIX. І ось, на початку 90-х років учень університету гельсінського ВНЗ Лінус Торвальдс (Linus Torvalds) розпочав пошуки UNIX-подібної ОС для IBM-сумісних персональних комп'ютерів [1].

У використанні Linux згідно з різними умовами мають всі можливості бути зацікавлені численні групи користувачів. Навести повний перелік нереально. Все ж, ось трохи зразків. Linux винятково абсолютна 32-х розрядна операційна система, що власне споживає ПК на всю потужність. Linux спрямовує індивідуальний ПК IBM PC у найреальнішу робочу станцію. Відповідно ціни персоналки, власне значно нижчі за розцінки робочої станції.

Прибуток у ціні дуже великий, таким чином, так само, як і не рахуючи економії в обладнанні, програмне забезпечення в Linux постачається з незалежною ліцензією, що дає безоплатне нелімітоване дублювання концепції. База, редактори, транслятори, СУБД, мережа, графічні інтерфейси, забави і величезна кількість інших програм об'ємом у тисячу мегабайт - безплатно і на законних підставах.

Для піратського ринку не виплачувати за програми - буденна справа. Але у випадку з Linux мало того, що за це ніхто переслідувати не буде, до того ж і всю документацію наддадуть!

Великий інтерес Linux представляє для споживачів і розробників прикладних систем. Уявіть собі фірму, що складається з декількох філій, територіально рознесених у різних районах міста і навіть у різних містах і країнах. На головному підприємстві працює сервер бази даних, клієнти - робочі місця у філіях - через мережу взаємодіють із сервером. Така система швидко, дешево і зручно робиться в Linux.

Припустимо, ви розробник платних програм для комерційних ОС, наприклад, SCO Unix, Solaris або DOS. Ви застосовуєте багатоплатформний компілятор, сильну багатовіконну систему налагодження, емулятори та системи сумісності. Усе без винятку таке є в Linux.

А в разі якщо ви науковий співробітник і пишете статті в міжнародні журнали, до ваших послуг у Linux видавнича система TeX, у форматі якої приймають статті в ці журнали. Статтю зможете надіслати електронною поштою безпосередньо з власної Linux-машини. Доки ви писали статтю, система здійснювала тривале складне обчислення, і для вас це зовсім ніяк не заважало. І масиви у вашій програмі зовсім не обмежені межею в 640 KB. Бажаєте масив у 50 мб? Будь ласка, зокрема й у разі, якщо у вас винятково 8 MB RAM, у такий спосіб, як у Linux функціонує віртуальна пам'ять.

Так само як ви вважаєте, чи приємно наштовхнутися із зависанням ОС, якщо аж до завершення тригодинної розплати залишилося 3 хвилинки. У Linux це для вас ніяк не загрожує.

Тут ніяк не дарма такого роду акцент зроблено на стабільність ОС. Користувач, практична діяльність якого урізана системами DOS і особливо MS Windows, звикся, чому зависання - непереборна особливість ОС. Справа йде якраз із точністю так само тільки Linux відкриває двері в гігантський світ відкритих систем, в якому існує величезна кількість засобів для вирішення найрізноманітніших завдань [2].

1.1. Основні характеристики ОС Linux

Унаслідок того, власне початкові коди Linux поширюються невимушено і доступні, до розвитку системи від самого початку долучилася величезна кількість незалежних творців. Завдяки цьому на нинішній епізод Linux - винятково прогресивна, стійка система, що швидко розвивається, яка практично одномоментно вбирає в себе найзаключніші науково-технічні новаторства.

Вона володіє всіма здібностями, які притаманні прогресивним повнофункціональним операційним системам на кшталт UNIX. Наведемо лаконічний перелік цих здібностей.

Характерні особливості Linux як ОС. Це [2]:

- багатозадачність: ймовірність робити чимало різних програм одразу;
- багатокористувацький режим: ймовірність діяти на однаковій автомашині майже всім користувачам одразу;
- захищений режим мікропроцесора (386 well-conserved wise);
- охорона пам'яті процесу; перебої програми не стануть спричиняти зависання системи;
- економне завантаження: ОС Linux автоматично зчитує тільки з диска ті частини програми, які дійсно використовуються для виконання;

- поділ сторінок за записом між екземплярами виконуваної програми. Це означає, власне процеси-екземпляри програми застосовують під час виконання однакову пам'ять. Щойно такий процес намагається зробити запис у пам'ять, то 4-х кілобайтна сторінка, у яку слідує запис, копіюється на вільне місце. Ця властивість нарощує швидкодію і зберігає пам'ять;
- віртуальна пам'ять зі сторінковою організацією (тобто на диск із пам'яті витісняють не весь неактивний процес, а лише необхідну сторінку); віртуальна пам'ять у самостійних сегментах диска та/або файлах файлової системи; розмір віртуальної пам'яті до 2 Гбайт; зміна обсягу віртуальної пам'яті під час проведення програм;
- єдина пам'ять програм і дискового кешу: вся вільна пам'ять вживається для буферизації обміну з диском;
- динамічні завантажувані подільні бібліотеки, що завантажуються;
- дамп програми для пост-мортем аналізу: дає змогу розглядати налагоджувальником не тільки ту, що виконується, а й програму, яка закінчилася аварійно;
- сертифікація за еталоном POSIX.1, порівнянність з еталонами Anschauung Trident і BSD на рівні початкових слів;
- через iBCS2-узгоджений емулятор порівнянність із SCO, SVR3, SVR4 за завантажуваними програмами;
- присутність початкового слова всіх програм, включно з текстами ядра, драйверів, засобів дослідження та додатків. Ці слова невимушено поширюються. Наразі деякі організації для Linux постачають низку платних програм за відсутності початкових слів, хоча все, що власне було незалежним, так і залишається вільним;
- управління завданнями в стандарті POSIX;
- емуляція співпроцесора в ядрі, тому додаток має можливість не турбуватися про емуляцію співпроцесора. Звісно, якщо співпроцесор у наявності, то він і вживається;

- підтримка державних алфавітів і узгоджень, у т.ч. для російської мови; ймовірність додавати свіжі;
- численні віртуальні консолі: на одному моніторі трохи одночасних незалежних сеансів роботи, які перемикаються з клавіатури;
- підтримка низки поширених файлових систем (MINIX, Xenix, файлові системи System V); наявність власної передової файлової системи обсягом до 4 Терабайт і з іменами файлів до 255 знаків;
- прозорий доступ до розділів DOS (або OS/2 FAT): розділ DOS має вигляд частини файлової системи Linux; підтримка VFAT (WNT, Windows 95);
- особлива файлова система UMSDOS, яка дає змогу у файлову систему DOS встановлювати Лінукс;
- доступ (тільки читання) до файлової системи HPFS-2 OS/2 2.1;
- підтримує всі стандартні формати CD ROM;
- підтримує мережі TCP/IP, приєднуючи ftp, telnet, NFS тощо [3] Реальна багатозадачність.

Усі процеси самостійні; жоден із них не може перешкоджати виконанню інших завдань. Щоб досягти бажаного результату ядро виробляє режим поділу часу центрального мікропроцесора, поперемінно підкреслюючи будь-якому процесу інтервали часу для виконання. Це значно відрізняється від режиму "витісняючої багатозадачності", реалізованої в перших версіях ОС Windows, коли процес зобов'язаний сам "поступитися" мікропроцесором іншим діям (і зможе дуже затримати їх виконання).

Linux - не тільки багатозадачна ОС, вона підтримує ймовірність одночасної роботи безлічі користувачів. При цьому Linux має можливість давати всі системні ресурси користувачам, які працюють із хостом через різні віддалені термінали.

Свопування оперативної пам'яті на диск. Свопування оперативки на диск дає змогу діяти за невеликого розміру фізичної оперативки; щоб досягти бажаного у результаті вміст деяких часток (сторінок) оперативки записуються у виділену область на жорсткому диску, яка трактується як добірна оперативка. Це трохи зменшує швидкість роботи, хоча дає змогу організувати роботу програм, що вимагають найбільшого розміру ОЗП, ніж практично є в наявності в комп'ютері [4].

Сторінкова організація пам'яті. Системна пам'ять Linux організована у вигляді сторінок розміром 4K. У разі якщо оперативка стовідсотково вичерпана, ОС стане знаходити здавна не використані сторінки пам'яті для їх переміщення з пам'яті на жорсткий диск. Коли деякі з цих сторінок стають необхідними, Linux відновлює їх із диска. Деякі старі Unix-системи і деякі прогресивні платформи (включно з Microsoft Windows) переносять на диск увесь вміст ВП, який належить до додатка, який на даний момент не працює, (тобто ВСІ сторінки пам'яті, які стосуються додатка, зберігаються на диску в разі нестачі пам'яті) власне найменш ефективно [4].

Завантаження виконуваних модулів "на вимогу". Ядро Linux підтримує виділення сторінок пам'яті на вимогу, за якого винятково потрібна частина коду виконуваної програми перебуває в оперативці, але частки, які не застосовуються на цей момент, залишаються на диску. Спільне використання виконуваних програм.

У разі якщо потрібно запустити відразу кілька копій деякого додатка (або 1 юзер запускає кілька схожих завдань, або різні юзери запускають однакове завдання), то в пам'ять завантажується виключно 1 копія виконуваного коду цього додатка, яка вживається всіма схожими завданнями, які одразу здійснюються.

Бібліотеки - набори операцій, що застосовуються програмами для обробки даних. Існує певна кількість звичайних бібліотек, що застосовуються відразу більш ніж одним процесом. У стареньких системах ці бібліотеки вбудовувалися в будь-який виконуваний файл, одночасне виконання яких призводило до непродуктивного застосування пам'яті. У новітніх системах (а саме, в Linux), підтримується робота з динамічно і статично поділюваними бібліотеками, що, власне, дає змогу зменшити об'єм окремих додатків.

Кешування диска - це впровадження частини оперативної пам'яті для збереження часто вживаних даних із диска, що власне значно прискорює доступ до часто вживаних програм і завдань. Linux використовує жваву систему кешування: пам'ять, зарезервована під кеш, зростає, коли пам'ять не використовується, і мініатюризується, у разі якщо системі або процесу користувача потрібно більше пам'яті [5].

100%-ва відповідність стандарту POSIX 1003.1. Часткова підтримка можливостей System V і BSD.

За допомогою емулятора iBCS2 система Linux дає змогу виконувати програми, що завантажуються, із систем SCO Unix, Xenix V/386, SVR3 generic, Wyse V/386, SVR4 (Unixware, USL, Dell), BSD/OS, FreeBSD. Наприклад, у Linux працюють такі додатки SCO Unix як CorelDraw, WordPerfect, Oracle.

У Linux (і назад) легко переносяться на рівні вихідних текстів програми із систем Unix System V і BSD.

Linux підтримує стандарти відкритих систем, у т.ч. POSIX. Світовий лідер з питань стандартизації інформаційних технологій і власник торгової марки UNIX компанія X/Open присвоїла ОС Linux сертифікат стандарту POSIX.1 FIPS151-2. Це означає офіційне визнання того факту, що практично всі Unix-додатки можуть бути без проблем перенесені в Linux. Не за горами сертифікація щодо POSIX.2, POSIX.4 і POSIX.7. Компанія Lasermoon, що випускає дистрибутив Linux- FT, має членство в X/Open.

POSIX 1003.1 (Portable Operating System Interface - інтерфейс мобільної операційної системи) задає стандартний інтерфейс Unix-систем, який описується набором процедур мови Сі. Зараз він підтримується всіма новими ОС. Microsoft Windows NT також підтримує POSIX 1003.1. Linux 100%-но відповідає POSIX. Додатково підтримуються деякі можливості System V і BSD для збільшення сумісності.

Linux використовує технологію IPC (InterProcess Communication) для обміну повідомленнями між процесами, використання семафорів і загальної пам'яті.

Можливість запуску виконуваних файлів інших ОС. Очевидно що ОС Linux не є першою в історії операційною системою. Для раніше розроблених ОС, включно з DOS, Windows, FreeBSD або OS/2, розроблено масу різного, зокрема дуже корисного і дуже непоганого програмного забезпечення. Для запуску таких програм під Linux розроблено емулятори DOS, Windows. Що цікаво, деякі компанії, зокрема фірма Vmware, розробили системи "віртуальних машин", які являють собою емулятор комп'ютера, де можна запустити будь-яку операційну систему. ОС Linux також здатна виконувати бінарні файли інших Intel-орієнтованих Unix-платформ, що відповідають стандарту iBCS2 (intel Binary Compatibility).

Linux підтримує велику кількість форматів файлових систем, включно з файловими системами Windows і OS/2, а також сучасними файловими системами з журналюванням. При цьому і власна файлова система Linux, яка називається Second Extended File System (ext2fs), дає змогу ефективно використовувати дисковий простір.

З Linux звичайним чином можна працювати з розділами жорстких дисків і дискетами, що містять файлові системи інших ОС, у т.ч. DOS, Windows, Minix, Xenix, Coherent, файлові системи System V. Файлові системи DoubleSpace, HPFS-2 (OS/2) і Amiga доступні в режимі тільки читання.

Файлові системи DoubleSpace/Stacked тощо стають доступними на читання і на запис у Linux під час роботи емулятора DOS.

Файлова система ОС Linux підтримує всі стандартні формати CD ROM.

Linux здатний бути як клієнтом, так і сервером мережевої файлової системи NFS. Linux підтримує протоколи NCP і SMB і може служити файлсервером або отримувати доступ до файлосерверів NetWare і Windows for Workgroups, Windows NT.

Linux можливо інтегрувати в будь-яку локальну мережу. Підтримуються всі служби Unix, включно з Networked File System (NFS), віддалений доступ (telnet, rlogin), робота в TCP/IP мережах, dial-up-доступ за протоколами SLIP і PPP, і так далі. Крім того, підтримується під'єднання Linux-машини як сервера або відвідувача для іншої мережі, а саме, діє спільне впровадження (sharing) файлів і віддалений друк у Macintosh, NetWare і Windows.

Хоча ОС Linux спочатку була розроблена для ПК на базі Intel 386/486, на даний момент працює на всіх версіях Intel-івських процесорів, починаючи з 386 і закінчуючи багатопроцесорними системами на Core i3-7 від Intel. Аналогічно вдало Linux працює на різних клонах Intel від інших виробників; в Інтернеті зустрічаються звістки про те, що на мікропроцесорах від AMD Linux діє навіть краще, ніж на Intel. Крім усього цього, розроблено версії для інших типів мікропроцесорів - ARM (ОС Android), DEC Alppha, SUN Sparc, M68000 (Atari і Amiga), MIPS, PowerPC та інших [1].

1.2. Архітектура операційної системи

Відкритий процес розробки є найбільшою перевагою Linux. Будь-хто може вносити зміни, які стануть доступними всім, оскільки вихідний код ядра вільно доступний для кожного.

Ядро Linux написано мовою С і асемблером. Між цими двома мовами існує звичайний компроміс: код на С більш переносимий і простий у підтримці, тоді як код на асемблері сприяє великій швидкості виконання. У загальному випадку асемблер у ядрі використовується тільки в тих місцях, де найбільш критичним показником є швидкість, або там, де потрібна реалізація коду, специфічного для конкретної платформи.

Хоч і в своїй основі ядро Linux монолітне, воно не є чистим монолітним ядром. Внесення змін і доповнень не викликає особливих труднощів, оскільки монолітний проєкт ядра досить модульний всередині.

Спрощена архітектура Linux-подібної ОС, з усіма низькорівневими деталями, які тільки можуть знадобитися для платформно-незалежної ОС. Варто вказати дві характерні особливості ядра:

- ядро відокремлює прикладні програми від апаратних засобів;
- одна частина ядра враховує специфіку архітектури та апаратури, тоді як інша частина ядра є переносною [6].

Архітектура ядра ОС Linux представлена на Рис. 1.1.

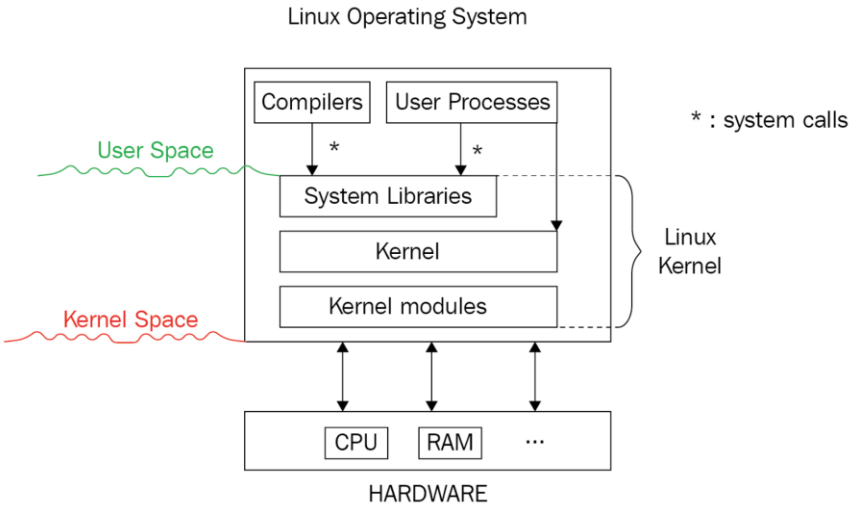

Рис. 1.1. Архітектура ОС Linux

Ядро досягає переносимості, частково за рахунок того, що воно застосовує по відношенню до себе ті ж самі трюки, що і по відношенню до призначених для користувача додатків. Це означає, що певна частина ядра забезпечує відокремлення частини ядра, що залишилася, від апаратних засобів так само, як ядро відокремлює призначені для користувача додатки від тієї самої апаратури. І додатки, і частина ядра, стають переносними завдяки такому поділу.

Інтерфейс низькорівневої, залежної від архітектури частини в загальному випадку визначається незалежною від архітектури частиною коду.

Таким чином, перенесення ядра на нову платформу зводиться до ідентифікації можливостей, на зразок розглянутих вище, і реалізації їх, як того вимагає нова платформа.

Перенесення користувацьких додатків далі підтримується за допомогою шару між додатками і ядром - стандартною бібліотекою C (libc). Додатки взаємодіють з ядром тільки через libc, але ніяк не безпосередньо.

Спосіб взаємодії з ядром через libc не залежить від архітектури, причому libc оберігає користувацький код від зайвої деталізації. Унаслідок того, що існує вищезгаданий механізм, усі користувацькі додатки, і навіть більша частина бібліотеки C, взаємодіють із ядром за допомогою способу, який не залежить від архітектури.

Операційна система, яка складається з ядра Linux, майже повністю ізольована від усіх прикладних програм. Ядро функціонує в режимі ядра - захищеному режимі процесора. Web-оглядачі, поштові клієнти, ігри та інші користувацькі програми запускаються поза режимом ядра в користувацькому режимі. Ядро має прямий, неконтрольований доступ до системних ресурсів, таких як пам'ять, процесор, периферія. За допомогою системних викликів інтерфейсу syscall запити користувацьких додатків перехоплюються ядром. Цей інтерфейс перевіряє дані, що надходять від користувацьких програм, перш ніж передати їх у ядро. Таким чином здійснюється захист від краху ядра від некоректно написаних програм [7].

Крім відмінностей між режимами, ядро і користувацькі програми займають різні області пам'яті. Кожен процес має власний віртуальний адресний простір чо-

тири гігабайти, здебільшого ядро займає тільки один гігабайт пам'яті, тоді як програма отримує три гігабайти. Програми користувацького режиму не мають прав доступу до пам'яті ядра напряму.

До підтримуваних апаратних платформ належать:

- x86 основна 32-бітна архітектура Intel, до цієї категорії відносяться процесори AMD та інших виробників, однак на комп'ютери з процесорами Intel 386 або 486 Red Hat Enterprise Linux не може бути встановлений. Залежно від версії WS, AS або ES, підтримується від 2 і вище процесорів, і оперативну пам'ять від 256 мегабайт;
- 64-бітні процесори фірми Intel. AMD64. Це 64-бітна архітектура від фірми AMD. Підтримується від 1 процесора і більше та від 512 мегабайт оперативної пам'яті.

IBM архітектура. Red Hat Enterprise Linux підтримує сервери S/390 і сервери eServer серій zSeries, iSeries, pSeries від компанії IBM.

У системах із симетричним мультипроцесорним опрацюванням процесори повинні розподілити роботу таким чином, щоб не заважати один одному, і при цьому вони не повинні витрачати на цю координацію дуже багато часу, що веде до того, що додаткова продуктивність процесора буде майже повністю витрачена.

Ядро Linux від версії 2.4 підтримує модель NUMA, для якої час доступу до різних ділянок пам'яті може варіюватися залежно від різновидів процесора [8].

Підсумовуючи сказане, слід зазначити, що ізоляція ядра від призначених для користувача додатків значно сприяє збільшенню надійності роботи системи і розширюваності.

Операційні системи Linux - це системи з відкритим кодом, однак для того, щоб адміністратори могли самостійно виправити дірки в безпеці або поліпшити продуктивність системи, вони повинні володіти знаннями, принаймні, не меншими, ніж розробники корпорації Red Hat. З огляду на те, що під час подальшого встановлення на систему нових патчів або сервіс-паків, зміни затираються, або призводять до порушення працездатності, відкритість кодів ядра або основних сервісів не є перевагою.

Крім цього, можливі проломи в безпеці та стабільності роботи системи можуть бути спричинені зміною коду системних програм.

Є ймовірність інтегрування Linux у будь-яку локальну мережу. Фактично будь-який дистрибутив Linux застосовують, як серверну операційну систему або звичайну робочу станцію. Web-сервер (Apache), сервер електронної пошти (Sendmail), ftp-сервер, файловий сервер Samba або сервер IP телефонії мають усі шанси бути легко встановленими в операційній системі Linux. Усі служби Unix, включно з Networked File System (NFS), віддаленим доступом (Telnet, Rlogin), роботою в TCP/IP мережах (рис. 1.2), dial-up-доступом за протоколами SLIP, L2TP і PPP підтримуються в цій ОС. Використання ОС у вигляді Інтернет-шлюзу вважається однією з найпоширеніших версій експлуатації ОС Лінукс, оскільки дає змогу містити всередині себе проксі-сервер, міжмережевий екран, поштовий сервер, DHCP і DNS сервери та інші мережеві утиліти. Інтернет-шлюз має можливість працювати як на одному з комп'ютерів мережі, так і на окремому сервері [9].

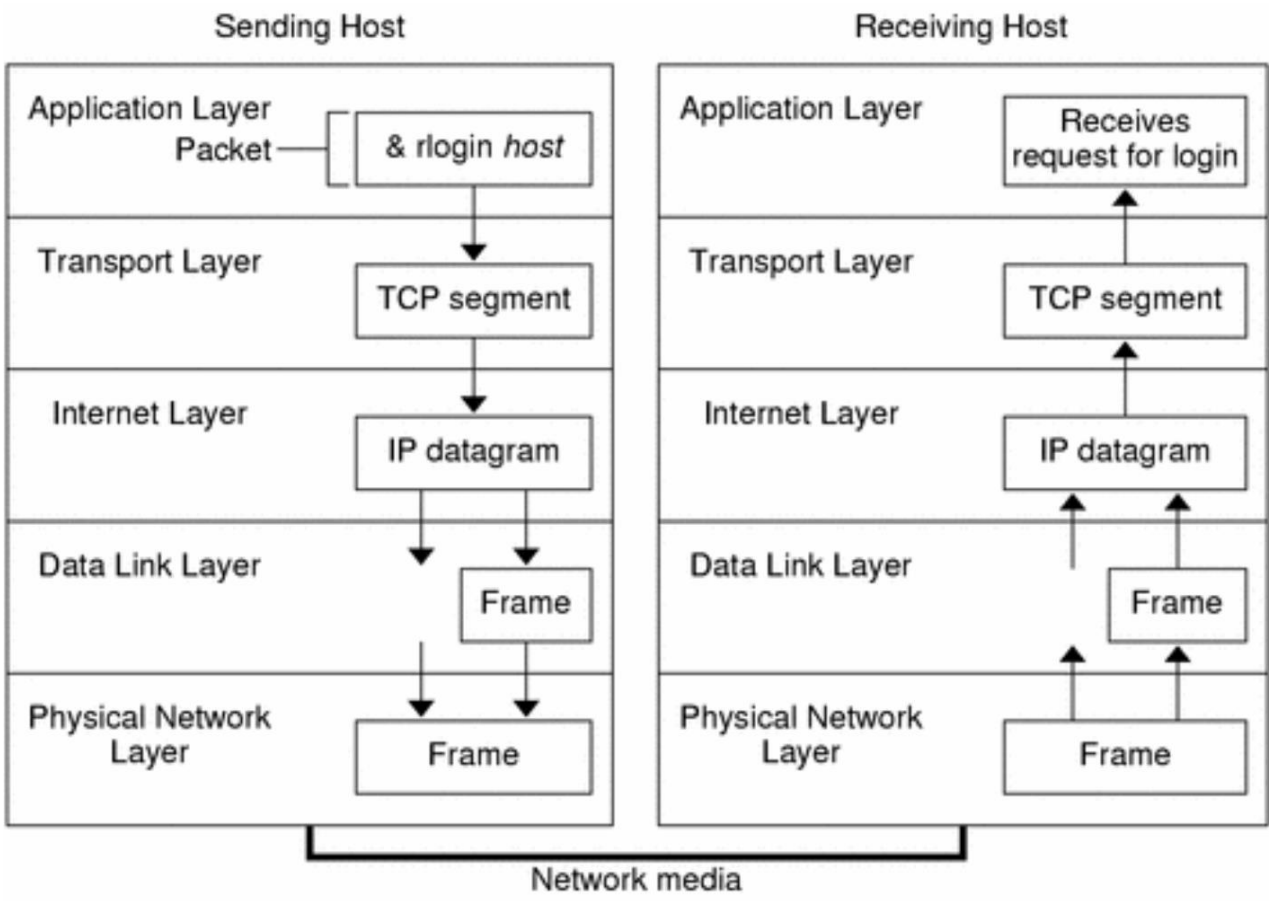

Рис. 1.2. Архітектура стека протоколів TCP/IP ОС Linux.

TCP/IP стек у Linux відповідає всім стандартам, а також перевершує реалізацію TCP/IP в інших ОС за багатьма своїми можливостями. Підтримка TCP/IP охоплює просунуту маршрутизацію (policy routing, QoS and Fair Quering), traffic shaping, пакетну фільтрацію (firewalling), multicasting, підтримку "прозорого" проксі, masquerading, тунелінг, aliasing та ін. Крім IPv4, у ядро Linux входить підтримка IPv6. Стандартно в поставку Linux входять: Apache - http-сервер, який користується найбільшою популярністю в Internet, Sendmail - програмка передавання електронної пошти (Mail Transfer Agent), FTP, POP3/IMAP, сервер доменних імен DNS, проксі-сервер Squid, який кешує http проксі-сервер Squid, засоби, що забезпечують пакетну фільтрацію, файрвол Iptables. Крім того підтримується підключення Linux-машини як сервера або відвідувача для іншої мережі, а саме, діє єдине впровадження (sharing) файлів і віддалений друк у Macintosh, NetWare і Windows. У Linux вірусів немає! Бо самі основи побудови операційної системи ліквідують імовірність роботи вірусів. Linux вважається багатокористувацькою операційною системою. Усіх користувачів поділяють на 2 види: звичайні користувачі та адміни. Таким чином, поділ прав призводить до того, власне звичайний користувач не має доступу до системних файлів, а перебудову операційної системи йому не класифікують як можливу.

У цій роботі мною буде спроєктовано шлюз для використання в ЛОМ компаній, розташованих у містах-супутниках м. Києва, і тих, що мають низькошвидкісне підключення до мережі Інтреннет. Передбачається використання цього сервера як шлюзу IP телефонії (Asterisk) і шлюзу виходу в Інтернет [10].

1.3. VoIP в ОС Linux і мережеві можливості. IP телефонія

На даний момент поняття "IP-телефонія" стало дуже відомим. Використання IP-телефонії обіцяє економію коштів, якість зв'язку і безліч додаткових сервісів.

Існує дві основні технології передачі даних - комутація каналів і комутація пакетів. Перша - це класична телефонія: встановлюють з'єднання, виділяють канал "від пункту А до пункту Б", і цей канал зайнятий увесь час розмови - незалежно від того, передається по ньому що-небудь, чи ні. Друге - основа сучасних комп'ютерних мереж:

дані розбиваються на невеликі пакети, кожен з яких передається окремо; кожен пакет може бути переданий своїм шляхом. У першому випадку ми отримуємо гарантовану якість (безперервність звукового потоку). Під час комутації пакетів нам необхідно докладати багато зусиль для забезпечення гарантованої якості зв'язку, зате плюс у тому, що канали використовуються істотно оптимальніше і гнучкіше.

Велика вартість комерційних АТС (ціна може становити кілька тисяч доларів), що вирішується застосуванням Asterisk; - велика вартість цифрових апаратів і шлюзів, що згодом перестає бути правдою (24 внутрішні аналогові порти від D-Link коштують близько 750$, SIP-телефон коштує 80-90$).

IP-телефонія - це цифрова телефонія. Залежно від налаштувань обладнання та пропускної здатності каналу може забезпечуватися якість зв'язку як на рівні цифрової телефонії, так і нижча - якщо є бажання заощадити на пропускній здатності каналу.

IP-телефонія - це не "дешева заміна звичайної телефонії". Таке ставлення рівносильне тому, що "електронна пошта - це дешева альтернатива поштового листування". Так, звісно, електронна пошта дешевша за паперову - через більш ефективні технології. Але можливості електронної пошти перед паперовою не обмежуються дешевизною - це абсолютно інша технологія, зі своїми особливостями. Точно так само і IP-телефонія. IP-телефонія надає можливості, або принципово недоступні раніше, або доступні з неприйнятними витратами.

Переваги:

- можливість створювати складні сервіси (голосові меню, складні правила обробки дзвінків, автоматизоване приймання факсів);
- можливість "об'єднувати офіси", тим самим заощаджуючи на телефонних переговорах;
- простота обслуговування і розширення;
- безпека - трафік IP-телефонії можна направляти через VPN, тим самим захищаючи його від перехоплення [11].

Під IP-телефонією мається на увазі голосовий взаємозв'язок, що виконується по мережах передачі даних, а саме по IP-мережах (IP - Internet Protocol). На цей час IP-

телефонія більше тіснить класичні телефонні мережі за допомогою легкості розгортання, невисокої ціни дзвінка, простоти конфігурації, чудових споживчих якостей взаємозв'язку та порівняльної захищеності з'єднання. Далі я хочу викласти основи еталонної моделі OSI (Open Systems Interconnection basic reference model) і розповідати "знизу-вгору", починаючи з фізичного та канального значень і завершуючи рівнями даних.

Принципи IP-телефонії. Під час здійснення дзвінка голосовий сигнал перетворюється на стислий пакет даних (докладніше цей процес піддасться розгляду в розділах "Імпульсно кодова модуляція" і "Кодеки"). Далі відбувається пересилання цих пакетів поверх мереж з комутацією пакетів, а саме, IP-мереж. При досягненні пакетами одержувача, вони декодуються в унікальні голосові сигнали. Ці процеси вірогідні завдяки великій кількості запасних протоколів, частину з яких буде розглянуто далі.

У цьому контексті, протокол передавання даних - певна мова, що дає змогу двом абонентам зрозуміти один одного і забезпечити якісне пересилання даних між двома пунктами.

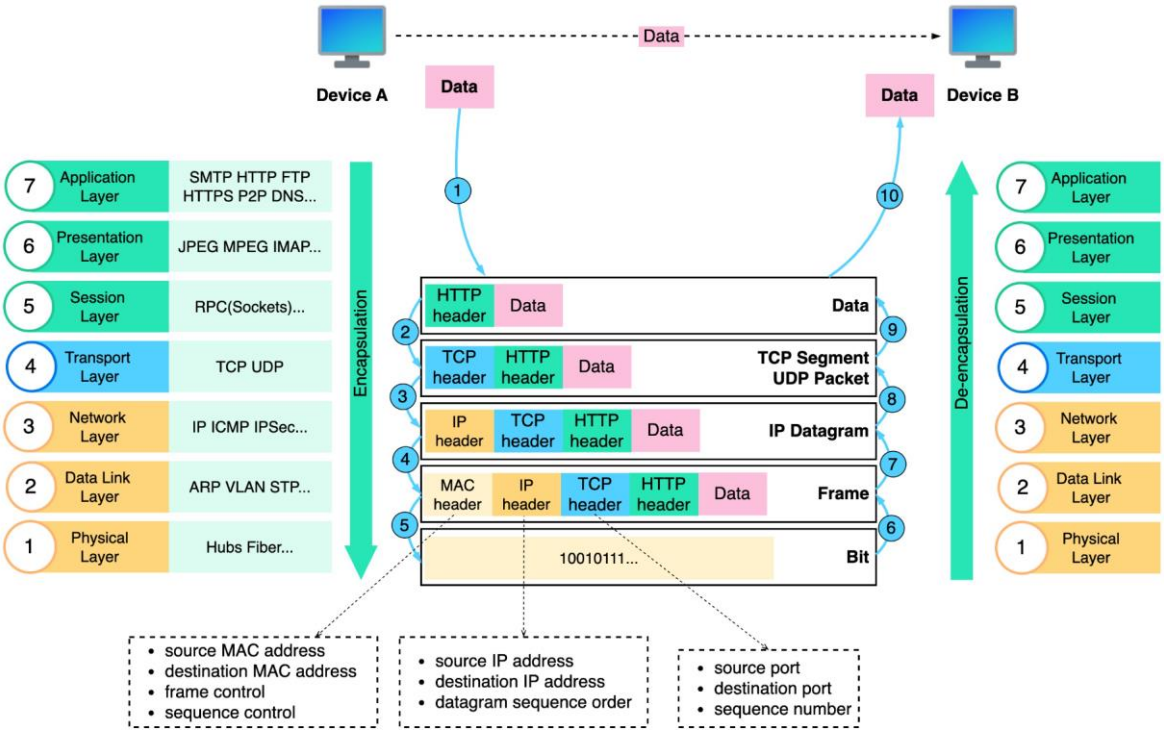

Рис. 1.3. Модель OSI та інкапсуляція даних

Відмінність від традиційної телефонії. У класичній телефонії встановлення з'єднання трапляється за допомогою телефонної станції і має на меті тільки бесіду. Тут голосові сигнали передаються по телефонних лініях, через виділене підключення. У разі ж IP-телефонії, стислі пакети цих надходять у масштабну або локальну мережу з явною адресою і передаються на базі цієї адреси. При цьому вживається вже IP-адресація, з усіма властивими їй відмінними рисами (зокрема маршрутизація).

При цьому IP-телефонія виявляється дешевшим рішенням як для оператора, так і для абонента. Відбувається це завдяки тому, що:

- класичні телефонні мережі володіють зайвою продуктивністю, тоді як IP-телефонія використовує технологію стиснення голосових пакетів і дає змогу стовідсотково використовувати ємність телефонної лінії;
- зазвичай, на теперішній момент доступ до мережі інтернет є у всіх охочих, що дає змогу зменшити витрати на підключення або зовсім ліквідувати їх;
- дзвінки в локальній мережі застосовують внутрішній сервер і відбуваються за відсутності ролі зовнішньої АТС.

Спільно з перерахованим вище, IP-телефонія дає змогу зробити кращою якість взаємозв'язку. Досягається це, знову ж таки, завдяки 3 головним моментам:

- Телефонні сервери постійно поліпшуються, і методи їхньої роботи стають найбільш стійкими до затримок або інших дилем IP-мереж;
- В особистих мережах їхні власники володіють повним контролюванням над обстановкою і мають всі шанси змінювати такі характеристики, як ширина смуги пропускання, кількість абонентів на одній смузі, і, як наслідок, значення затримки;
- Мережі з комутацією пакетів розвиваються, і раз на рік запроваджують новітні протоколи та технології, що дають змогу зробити кращою якість взаємозв'язку (наприклад, протокол резервування смуги пропускання RSVP).

Завдяки IP-телефонії доволі розкішно вирішується проблема лінії, що займається, тому що переадресація або переведення в режим очікування мають всі шанси бути здійснені кількома командами в конфігураційному файліку на АТС [12-14].

Фізичний рівень (Physical Layer). На фізичному рівні виконується надання потоку бітів у фізичному середовищі через відповідний інтерфейс. IP-телефонія фактично стовідсотково спирається на тісніше наявну інфраструктуру мереж. У вигляді середовища передавання інформації використовуються, як правило, кручена пара категорії 5 (UTP5), одномодове або багатомодове оптичне волокно, або коаксіальний кабель. Тим самим повною мірою реалізується принцип конвергенції телекомунікаційних мереж.

Канальний ступінь (Data Link Layer). Згідно зі специфікацією IEEE 802 канальний рівень ділиться на 2 підрівні:

- MAC (Media Access Control) - забезпечує взаємодію з тілесним рівнем;
- LLC (Logical Link Control) - обслуговує мережевий рівень.

На канальному рівні діють комутатори - прилади, що забезпечують з'єднання кількох вузлів комп'ютерної мережі та розміщення фреймів між хостами на базі фізіологічної (MAC) адресації.

Необхідно згадати пристосування віртуальних локальних мереж (Virtual Local Area Network). Ця розробка дозволяє творити закономірну топологію мережі, не зважаючи ні на що на її фізичні характеристики. Досягається це тегуванням трафіку, власне ретельно описано в стандарті IEEE 802.1Q.

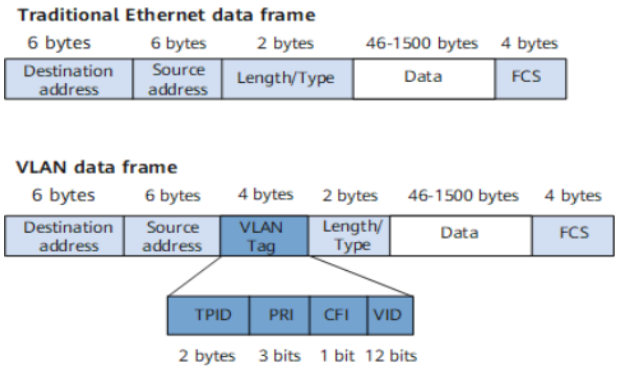

Рис.1.4. Формат фрейма

У контексті IP-телефонії підкреслимо Voice VLAN, що широко застосовується для ізоляції голосового трафіку, що генерується IP-телефонами, від інших даних. Її введення доцільне за 2 факторами:

- збереження. Створення окремої голосової VLAN зменшує ймовірність перехоплення і розбору голосових пакетів.
- збільшення характеристики передавання. Пристосування VLAN дає змогу встановити завищене значення голосовим пакетам, і, як наслідок, збільшити властивість взаємозв'язку.

Мережевий рівень (Network Layer). На мережевому рівні трапляється маршрутизація, відповідно до цього ключовими пристроями мережевого значення вважаються маршрутизатори (Router). Саме тут орієнтується, яким методом дані досягнуть одержувача з якимось Айпішником. Основний маршрутизований протокол - IP (Internet Protocol), на базі якого і побудовано IP-телефонію, а також глобальну мережу Інтернет. Крім того, присутня величезна кількість динамічних протоколів маршрутизації, найпопулярніший серед яких OSPF (Open Shortest Path First) - внутрішній протокол, що базується на поточному стані каналів взаємозв'язку;

На нинішній епізод є особливі VoIP-шлюзи (Voice Over IP Gateway), що забезпечують включення звичайних аналогових телефонних апаратів до IP-мережі. Зазвичай, вони мають і інтегрований маршрутизатор, що дає змогу вести облік трафіку, авторизувати користувачів, автоматом видавати айпішники, керувати смугою пропускання. Серед стандартних функцій VoIP-шлюзів:

- функції безпеки (створення списків доступу, авторизація);
- підтримка факсимільного зв'язку;
- підтримка голосової пошти;
- підтримка протоколів H.323, SIP (Session Initiation Protocol).

Для боротьби з імовірними затримками передавання IP потрібно доповнювати додатковими засобами, наприклад протоколами встановлення послідовності (щоб голосові дані не змагалися зі звичайними).

Зазвичай, з цією метою на маршрутизаторах використовується послідовність з невеликою затримкою (LLQ - Low-Latency queuing), або зважена організація черг на

базі класів (CBWFQ - Class-Based Weighted Fair Queuing). Також, необхідні схеми маркування із завданням пріоритетів для розгляду голосових даних, як більш принципових для передачі [15].

Транспортний рівень (Transport Layer). Для транспортного значення властиві:

- сегментація даних додатків верхнього рівня;
- забезпечення наскрізного з'єднання;
- запорука надійності даних.

Головні протоколи автотранспортного значення - TCP (Transmission Control Protocol), UDP (User Datagram Protocol), RTP (Real-time Transport Protocol). Конкретно в IP-телефонії використовуються протоколи UDP і RTP, причому основна їхня відмінність від TCP полягає в тому, що вони не забезпечують надійність доставки даних. Це є найбільш застосовним варіантом, ніж втілення контролювання за доставкою (TCP), тому що телефонний взаємозв'язок дуже залежний від затримок передачі, хоча найменш чутливий до втрат пакетів.

Значення даних (Data Layers). 3 заключних значення моделі OSI розглянемо разом. Таке об'єднання припустиме, тому що процеси, що відбуваються на цих рівнях, щільно сплетені один від одного, і змальовувати їх безвідносно поділу на підрівні стане логічніше.

H.323. Спершу потрібно описати стек протоколів H.323, створений 1996 року. Цей еталон має опис обладнання, мережевих служб і термінальних пристосувань, створених для втілення аудіо- та відеозв'язку в мережах із комутацією пакетів (Веб). Для хоч якого пристрою стандарту H.323 невідворотна підтримка обміну голосовою інформацією.

Рекомендації H.323 мають на увазі [16]:

- Платформену автономію;
- Еталони кодування аналогових даних;
- Керування смугою пропускання;
- Гнучкість і сумісність.

Зазначимо доволі важливий факт: у настановах не визначено фізичне середовище передавання, транспортний протокол і мережевий інтерфейс. Це означає, що пристосування, які підтримують стандарт H.323, мають всі шанси діяти в будь-яких наявних зараз мережах з комутацією пакетів.

Згідно з H.323 чотирма ключовими складовими VoIP-з'єднання вважаються:

- термінал;
- шлюз;
- контролер зони;
- контролер управління багатоточковою конференцією (MCU - Multipoint Control Unit).

SIP (Session Initiation Protocol). SIP - протокол сигналізації, створений для організації, конфігурації та закінчення сеансів взаємозв'язку. SIP незалежний від транспортних технологій, але при встановленні з'єднання бажано застосовувати UDP. Для передачі самої голосової та відеоінформації рекомендується використовувати RTP, хоча ймовірність застосування інших протоколів не виключена [17].

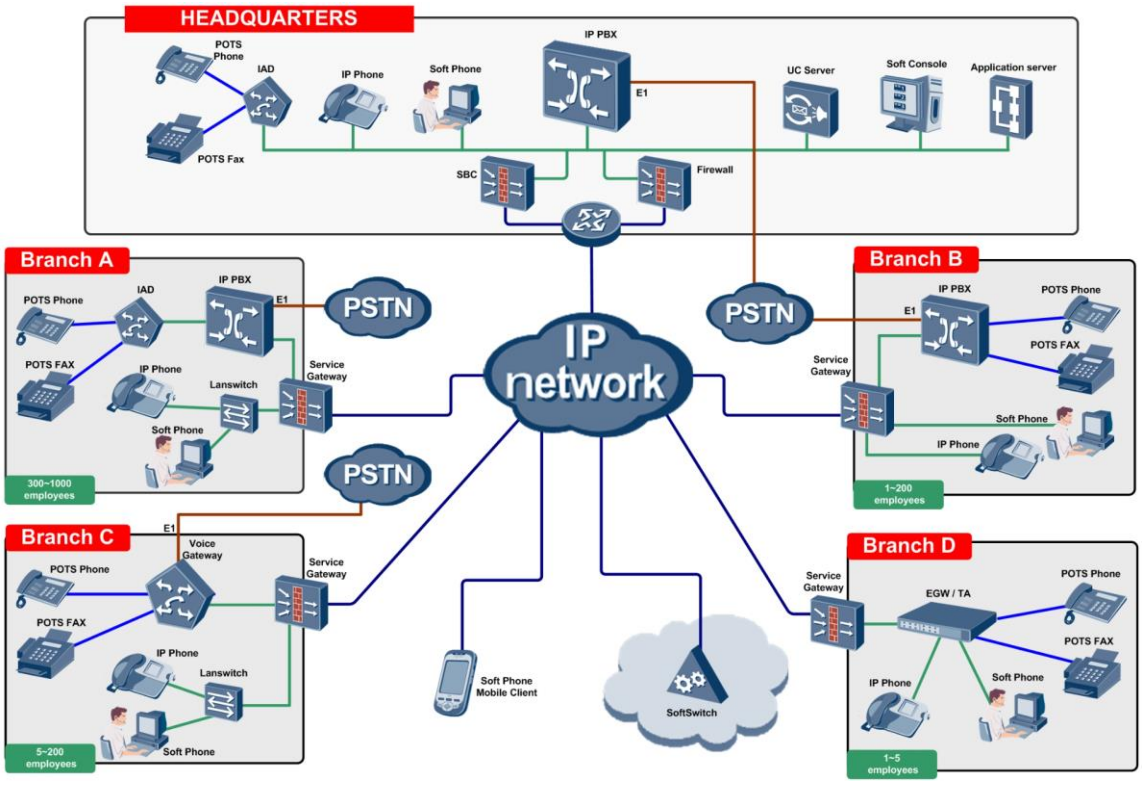

Рис. 1.5. Приклад структурної схеми мережі в IP-телефонії

У SIP віднесено два на подобі сигнальних повідомлень - запит і відповідь. І ще присутні шість процедур [18]:

- INVITE (запрошення) - кличе користувача брати участь у сеансі взаємозв'язку (працює для встановлення нового з'єднання; має можливість містити характеристики для узгодження);
- BYE (роз'єднання) - закінчує злиття між 2-ма користувачами;
- OPTIONS (функції) - вживається для передання інформації про підтримувані властивості (це передання має можливість виконуватися безпосередньо між 2-ма агентами користувача;
- ACK (підтвердження) - використовується для підтвердження отримання звістки або для позитивної відповіді на команду INVITE;
- CANCEL (скасування) - закінчує пошук юзера;
- REGISTER (реєстрація) - передає інформацію про місцезнаходження користувача на сервер SIP, який має можливість передавати її на сервер адрес (Location Server).

Кодеки. Аудіокодеком називають програму або метод, який стискає, або розтискає цифрові звукові дані, що дає змогу знизити вимоги до пропускної спроможності каналу передавання даних. В IP-телефонії на сьогодні більш поширене перетворення за допомогою кодека G.729, також стиснення G.711 за A-законом (alaw) і μ-законом (ulaw). G.729 вважається кодеком, який стискає початковий сигнал із втратою даних. Головна ідея, закладена в G.729 - надання далеко не оцифрованого сигналу, а його характеристик (спектральної властивості, чисельності переходів через нуль), необхідних для наступного синтезування на приймаючій стороні. При цьому всі головні властивості голосу, в тому числі амплітуда и тембр зберігаються. Пропускна спроможність каналу, на яку розрахований цей кодек, - 8 кбіт/с. Протяжність кадру оброблюваного G.729 - 10 мс, частота дискретизації - 8 кГц. Для будь-якого з таких кадрів орієнтуються характеристики математичної моделі, які в подальшому і передаються в канал у вигляді кодів [19].

При застосуванні кодування G.729 затримка становить 15 мс, з яких 5 мс витрачається на заповнення підготовчого буфера. Зауважимо крім того, що кодек G.729 висуває досить високі вимоги до ресурсів процесора.

G.711 - голосовийкодек, який не передбачає практично ніякого стиснення, крім компандування - методу скорочення результатів каналів з урізаним динамічним спектром. У базі цього способу лежить принцип зменшення чисельності значень квантування сигналу у сфері найвищої гучності, зберігаючи при цьому якість звуку. Дві широко застосовувані в телефонії схеми компандування alaw иulaw. Сигнал у цьому кодеку надано потоком величиною 64 кбіт/с.

Частота дискретизації - 8000 кадрів по 8 біт за секунду. Якість голосу суб'єктивно набагато краща, ніж при використанні кодека G.729.

Властивості профілактики в IP-телефонії (Quality of Service - QoS).

У мережах на базі стека TCP/IP відмінна якість обслуговування трафіку, чутливого до затримок передавання, не підтримується за дефлотом. При застосуванні протоколу TCP наявна гарантія достовірної доставки інформації, хоча її перенесення може виконуватися з непередбачуваними затримками. Для UDP властива мінімізація затримок, хоча гарантія правильної доставки пакета відсутня.

У той момент добротність мовного трафіку дуже залежить від властивості передавання, і в мережі, де не реалізовано механізми, які гарантують відповідну якість, реалізація IP-телефонії може не задовольняти домаганням користувачів.

Головними ознаками властивості профілактики вважаються пропускна здатність мережі та затримка передавання. Затримка при цьому орієнтується як проміжок часу, що минув з моменту відправлення пакета, до моменту його приймання.

Крім іншого є такі властивості, як готовність мережі та її надійність (оцінюються за наслідками контролювання значення профілактики протягом тривалого часу, або за коефіцієнтом використання). Для вдосконалення властивості взаємозв'язку вживають такі механізми [16-18]:

- Перемаршрутизація. У разі перевантаження одного з каналів взаємозв'язку дозволяє виконати доставку за допомогою запасних маршрутів.

- Резервування ресурсів каналу взаємозв'язку тимчасово з'єднання.
- Пріоретизація трафіку. Дозволяє позначати пакети згідно з рівнем їхньої значущості та створювати сервіс на базі міток.
- Як говорилося раніше, голосовий трафік дуже чутливий до затримок передачі. Найбільший час затримки не може перевершувати 400 мс (сюди входить і тривалість обробки інформації на остаточних станціях). Розпізнають два головних на подобі затримок:
- Затримка під час кодування інформації в голосових шлюзах або термінальному обладнанні. Мініатюризується методом вдосконалення алгоритмів оброблення і перевлаштування голосу. Затримка, яку вносить мережа передавання. Зменшується шляхом поліпшення мережевої інфраструктури, зокрема, скороченням кількості маршрутизаторів і використанням високошвидкісних каналів [17].

Фуррор TCP/IP у вигляді мережевого протоколу Інтернету значною мірою пояснюється його можливістю поєднувати мережі різних обсягів і системи різних типів. Ці мережі довільно поділяються на 3 головні класи (і кілька неосновних) із заздалегідь явними об'ємами, будь-який з яких системні адміни можуть розбити на найдрібніші сабмережі. Маска підмережі ділить Айпішник на 2 частини. 1 частина ідентифікує вузол, інша - мережу, до якої він належить.

IP-адреси: мережі та вузли. IP-адреса являє собою 32-розрядний номер, який неповторно ідентифікує вузол (комп'ютер або пристосування, приміром, принтер або маршрутизатор) у мережі TCP/IP.

IP-адреси традиційно подано у вигляді 4-х розрядів, розбитих крапками, наприклад, 192.168.123.132. Щоб усвідомити використання масок підмереж для визначення вузлів, сіток і підмереж, зверніть увагу на IP-адресу в двійковому позначенні.

Наприклад, IP-адреса 192.168.123.132, що повторює вигляд розрядів, розбитих крапками, - це (у двійковому позначенні) 32-розрядний номер 11000000010100011110111000100. Такий номер важко інтерпретувати, тому розбийте його на 4 частки по 8 двійкових символів.

Ці 8-розрядні секції називаються "октети". У тих випадках ця IP-адреса матиме вигляд: 11000000.10101000.01111011.10000100. Цей номер ненабагато зрозуміліший, тому майже завжди слід перетворювати двійкову адресу у формат розбитих крапками розрядів (192.168.123.132). Десяткові кількості, розбиті крапками, і є октети, перевтілені з двійкового в десяткове позначення.

Щоб мережа інтернет TCP/IP діяла ефективно як сукупність сіток, маршрутизатори, що забезпечують обмін пакетами між цими мережами, не уявляють чіткого місця розташування вузла, для якого призначено пакет. Маршрутизатори уявляють виключно, до якої мережі належить вузол, і використовують відомості, що зберігаються в таблицях маршрутизації, щоб привезти пакет у мережу вузла призначення. Щойно пакет доставлено в потрібну мережу, його доставляють у відповідний вузол.

Для втілення цього процесу IP-адреса складається з 2-ух часток. 1-ша частина IP-адреси означає адресу мережі, заключна частина - адресу вузла. Якщо розглянути IP-адресу 192.168.123.132 і розбити її на ці 2 частки, то вийде наступне: 192.168.123. Мережа 132. вузол або 192.168.123.0 - адреса мережі 0.0.0.0.132 - адреса вузла

Маска підмережі. Наступний складник, необхідний для роботи протоколу TCP/IP, - це маска підмережі. Протокол TCP/IP використовує маску підмережі, щоб знайти, у якій мережі розташовуватися вузол: у локальній підмережі або віддаленій мережі.

У протоколі TCP/IP частки IP-адреси, що застосовуються у вигляді адреси мережі та вузла, не помічені, як наслідок, позначені вище адреси мережі та вузла нереально знайти за відсутності присутності додаткових відомостей. Ці відомості можна знайти з іншого 32-розрядного номера з ім'ям "маска сабмережі" [14]. У цьому разі маскою сабмережі вважають 255.255.255.0. Значення цього номера зрозуміле, якщо знати, що власне кількість 255 у двійковому позначенні відповідає кількості 1111111111; таким чином, маскою сабмережі вважають номер:

11111111.11111111.11111111.0000000

Розташувавши таким чином IP-адресу і маску підмережі, можна виділити складові мережі та вузла:

11000000.10101000.01111011.10000100 - IP-адреса (192.168.123.132)
1111111111.1111111111.1111111111.1111111111.00000000 - маска підмережі (255.255.255.0).

Перші 24 розряди (кількість одиниць у масці підмережі) розпізнаються як адреса мережі, а останні 8 розрядів (кількість нулів, що залишилися в масці підмережі) - адреса вузла. Таким чином, отримуємо таке:

11000000.10101000.01111011.00000000 - адреса мережі (192.168.123.0)
00000000.00000000.00000000.00000000.10000100 - адреса вузла(000.000.000.000.132)

З цього прикладу з використанням маски підмережі 255.255.255.0 видно, що код мережі 192.168.123.0, а адреса вузла 0.0.0.132. Коли пакет із кінцевою адресою 192.168.123.132 доставляють у мережу 192.168.123.0 (із локальної підмережі або віддаленої мережі), комп'ютер отримає його з мережі й обробить.

Майже всі десяткові маски підмережі перетворюються на двійкові числа, представлені одиницями ліворуч і нулями праворуч. Ось ще деякі поширені маски підмережі:

Десяткові	Двійкові
255.255.255.192	1111111.11111111.1111111.11000000
255.255.255.224	1111111.11111111.1111111.11100000

Класи мереж. Інтернет-адреси розподіляються організацією InterNIC, яка адмініструє Веб. Ці IP-адреси розподілені за класами. Найпоширеніші класи A, B і C. Класи D і E існують, але традиційно не вживаються остаточними користувачами.

Будь-який із класів адрес має власну маску сабмережі за дефлотом. Знайти клас Айпішника можливо за його першим октетом. Нижче описано інтернет-адреси класів A, B і C зі зразком адреси для будь-якого класу.

Мережі класу A за дефлотом вживають маску сабмережі 255.0.0.0.0 і мають цінності від 0 до 127 у першому октеті. Адреса 10.52.36.11 вважається адресою класу A. Першим октетом вважається кількість 10, що входить у спектр від 1 до 126 включно.

Мережі класу B за дефлотом вживають маску сабмережі 255.255.0.0 і мають у першому октеті значення від 128 до 191. Адресок 172.16.52.63 вважається адресою класу B. Першим октетом вважається кількість 172, що входить у спектр від 128 до 191 включно.

Мережі класу C за дефлотом вживають маску сабмережі 255.255.255.0 і мають у першому октеті значення від 192 до 223. Адресок 192.168.123.132 вважається адресою класу Yards. У першому октеті кількість 192, яка присутня між 192 і 223 включно.

У деяких варіантах значення маски сабмережі за дефлотом не підходить потребам організації у зв'язку з фізичною топологією мережі або оскільки кількість сіток (чи вузлів) не відповідає лімітуванням маски сабмережі за дефлотом. У наступному розділі йдеться, як можна розпланувати мережі за допомогою масок сабмережі.

Підмережі. TCP/IP-мережу класу A, B або C зможе ще бути розбита на сабмережі сисадміном. Утворення субмереж, можливо, потрібне під час узгодження закономірної текстури адреси Веба (абстрактний світ айпішників і субмереж) із фізичними мережами, які застосовують у теперішньому світі. Сисадмін, який видалив блок Айпішників, можливо, адмініструє мережі, санкціоновані не належним їм чином. Наприклад, є в наявності мережа інтернет зі 150 вузлами в 3-х мережах (у різних містечках), з'єднаних маршрутизатором TCP/IP. У кожній із цих 3-х сіток 50 вузлів. Виділяємо мережу класу Yards 192.168.123.0. (Для прикладу, по суті, ця адреса із серії, не розміщеної у Вебі.) Це означає, власне адреси з 192.168.123.1 по 192.168.123.254 застосовують для цих 150 вузлів.

Дві адреси, які неможливо застосовувати в цьому випадку, - 192.168.123.0 і 192.168.123.255, тому що двійкові адреси з компонентом вузла з 1 одиниць і нулів неприпустимі. Адреса з 0 неприпустима, оскільки її використовують для визначення

мережі за відсутності вказівки вузла. Адресок із кількістю 255 (у двійковому позначенні адреса вузла, що складається з 1 одиниці) вживається для доставки звістки на будь-який вузол мережі. Йде просто з'ясувати, власне 1-у і заключну адресу в будь-якій мережі і сабмережі не має можливості бути присвоєно.

Зараз залишилося віддати Айпішники 254 вузлам. Це неважко, коли всі 150 комп'ютерів вважаються часткою однієї мережі. Але в цьому випадку 150 комп'ютерів діють у 3-х окремих фізичних мережах. Замість запиту на найбільшу чисельність адресних блоків для будь-якої мережі мережу розбивають на сабмережі, що, власне, дає можливість використовувати 1 блок адрес у кількох фізичних мережах.

У цьому разі мережу розбивають на 4 сабмережі за допомогою маски сабмережі, яка нарощує адресу мережі та зменшує ймовірний спектр адрес вузлів. Інакше кажучи, ми "позичаємо" трохи розрядів, традиційно застосовуваних для адреси вузла, і візьмемо на озброєння їх для компоненти мережі в адресі.

Маска сабмережі 255.255.255.192 дозволяє зробити 4 мережі з 62 вузлами в кожній. Це може бути, оскільки в двійковому позначенні 255.255.255.192 - це ж, власне, і 1111111.11111111.1111111.1111111.11000000. 1-ші дві цифри останнього октету стають адресами мережі, тому з'являються додаткові мережі 00000000 (0), 01000000 (64), 10000000 (128) і 11000000 (192). (Деякі адміни використовують винятково 2 із цих підмереж, використовуючи номер 255.255.255.192 як маску сабмережі. RFC 1878.) У цих 4 мережах заключні 6 двійкових цифр застосовують у вигляді адрес вузлів.

Впровадження маски сабмережі 255.255.255.192 конвертує мережу 192.168.123.0 у 4 мережі: 192.168.123.0, 192.168.123.64, 192.168.123.128 і

192.168.123.192. Ці 4 мережі стануть мати наступні чинні адреси:

192.168.123.1-62

192.168.123.65-126

192.168.123.129-190

192.168.123.193-254

Не забувайте, що двійкові адреси вузлів з одними лише одиницями і нулями недійсні, тому неможливо брати на озброєння адреси з такими кількостями в заключному октеті: 0, 63, 64, 127, 128, 191, 192 або 255.

Не пропустіть зі свого поля зору наступні 2 адреси вузлів: 192.168.123.71 і 192.168.123.133. Якщо застосовувати по дефлоту маску сабмережі класу C 255.255.255.0, дві адреси опиняться в мережі 192.168.123.0. Однак, якщо використовувати маску підмережі 255.255.255.192, вони опиняться в різних мережах: 192.168.123.71 - у мережі 192.168.123.64, тоді як 192.168.123.133 - у мережі 192.168.123.128.

Основні шлюзи. Взаємозв'язок між TCP/IP-комп'ютером і вузлом з іншої мережі традиційно виконується через прилад, іменований маршрутизатором. З огляду на переконання TCP/IP маршрутизатор, позначений на вузлі, що зв'язує сабмережу вузла з іншими мережами, іменується головним шлюзом. У цьому розділі йдеться про те, яким чином протокол TCP/IP описує, надсилати чи ні пакети цих пакетів на головний шлюз, щоб зв'язатися з іншим комп'ютером або пристосуванням у мережі. Під час спроби встановлення взаємозв'язку між вузлом та іншим пристосуванням за допомогою протоколу TCP/IP вузол порівнює явну маску сабмережі та айпішник призначення з маскою сабмережі та власним своїм айпішником. Унаслідок цього порівняння комп'ютер з'ясує, для якого з вузлів призначений цей пакет - локального або віддаленого.

Якщо внаслідок цього процесу призначенням вважається локальний вузол, то комп просто посилає пакет у локальну сабмережу. У разі якщо внаслідок порівняння з'ясовується, власне призначенням вважається віддалений вузол, комп направляє пакет на головний шлюз, особливий в властивості TCP/IP. Таким чином, конкретно маршрутизатор відповідає за надсилання пакета у правильну сабмережу [19-20].

Усунення неполадок. Передумовою проблем, пов'язаних із протоколом TCP/IP, нерідко працює некоректне налаштування 3-х головних частин у TCP/IP-властивостях PC. Усвідомлюючи вплив похибок у налаштуванні TCP/IP на функціонування мережі, можна вирішити майже всі найпоширеніші труднощі протоколу TCP/IP.

Неправильна маска сабмережі. Якщо мережа вживає маску сабмережі, відмінну від маски за дефлотом, для власного класу адрес, а у відвідувача ще налаштовано маску сабмережі за дефлотом для класу адрес, взаємозв'язок із прилеглими мережами стане неможливим, але це не стосується віддалених мереж. Наприклад, якщо зробити 4 сабмережі (як у зразку про підмережі), хоча брати на озброєння неправильну маску сабмережі 255.255.255.0 під час налаштування протоколу TCP/IP, вузли не зможуть знайти, чи власне якісь комп'ютери перебувають в інших підмережах. Тоді пакети, створені для вузлів у різних фізичних мережах, які є часткою 1-ї адреси класу Yards, не стануть надіслані на головний шлюз для доставки. Єдиним показником цього вважається обстановка, як скоро комп зможе вести взаємодію з вузлами у своїй локальній мережі та може зв'язатися з усіма віддаленими мережами, крім тих, що розташовані поруч і мають ту саму адресу класу A, B або C. Для усунення цієї проблеми вкажіть вірну маску підмережі в налаштуванні TCP/IP для цього вузла.

Неправильна IP-адреса. Коли помістити комп'ютери з айпішниками, які повинні бути в окремих підмережах, спільно в локальну мережу, вони не зуміють встановити взаємозв'язок разом. Вони намагатимуться надсилати один одному пакети через маршрутизатор, який не зуміє навести ці пакети належним чином. Симптомом цих труднощів вважається ситуація, коли комп зможе встановити взаємозв'язок із вузлами у віддалених мережах, хоча не має змоги вести взаємодію з деякими або всіма комп'ютерами у власній локальній мережі. Для усунення цієї проблеми переконайтеся, що всі комп'ютери в одній фізичній мережі мають IP-адреси в одній і тій самій IP-підмережі. Коли витрачено всі IP-адреси в окремому секторі мережі, є інші рішення, які в цій замітці не описуються. Неправильний головний шлюз. PC з помилково налаштованим головним шлюзом зуміє взаємодіяти з вузлами в особистому секторі мережі, але не зуміє встановити взаємозв'язок з вузлами в деяких або у всіх віддалених мережах. Якщо 1 фізична мережа має більше ніж 1-й маршрутизатор і неправильний маршрутизатор налаштований як ключовий шлюз, вузол зуміє вести взаємодію з деякими віддаленими мережами, хоча не з усіма. Ця проблема нерідко з'являється, якщо в організації 1 маршрутизатор з'єднаний із внутрішньою мережею TCP/IP, а інший - із Вебом.

Протоколи динамічної маршрутизації і ще можливо розбити на класові та безкласові протоколи. Розглянемо, чим вони відрізняються.

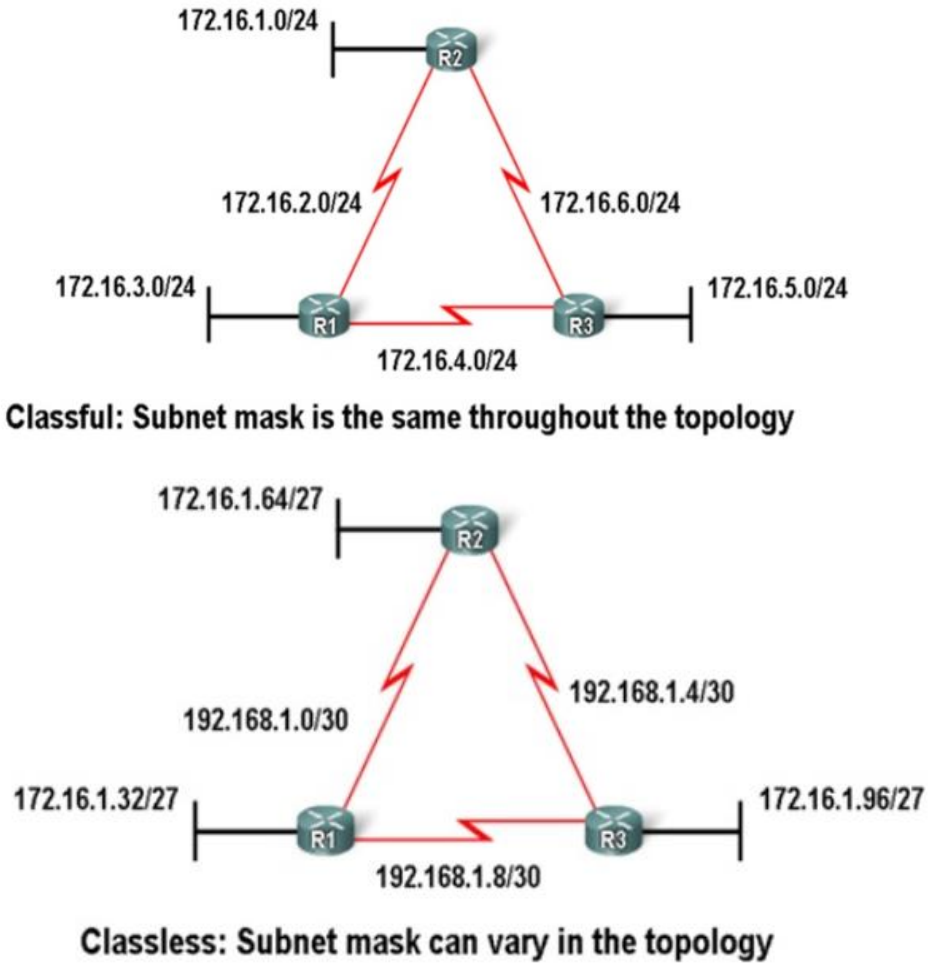

Рис. 1.6. Схема Класової та Безкласової маршрутизацій

Класові протоколи маршрутизації не надсилають інформацію по масці підмережі в маршрутних оновленнях. 1-ші протоколи маршрутизації, зокрема RIP, були класовими. Це було на той момент, як тільки мережеві адреси відрізнялися на базі класів A, B або C. Протокол маршрутизації не міг включати маску сабмережі в маршрутне оновлення, оскільки мережеву маску можна було визначити, базуючись на першому октеті мережевої адреси.

Класові протоколи маршрутизації мають усі шанси досі вживатися в деяких із нинішніх сіток, хоча через те, що вони не включають маску сабсети, вони не в змозі вживатися у всіх обстановках. Класові протоколи маршрутизації не мають змоги вжи-

ватися, коли мережа ділиться на сабмережі, використовуючи більш ніж 1 маску сабмережі, іншими словами класові протоколи маршрутизації не підтримують маски сабмережі змінної довжини (VLSM).

Є інші обмеження до класових протоколів маршрутизації, включно з їхньою нездатністю підтримувати несуміжні мережі. Класові протоколи маршрутизації, несуміжні мережі та VLSM обговорюватимуться в пізніших рубриках.

Класові протоколи маршрутизації включають RIPv1 і IGRP.

Безкласові протоколи маршрутизації включають маску сабмережі з мережевою адресою в маршрутних оновленнях. Нинішні мережі більше не відрізняються на базі класів, і маска сабмережі не має можливості бути визначена значенням першого октету. Безкласові протоколи маршрутизації знадобляться в безлічі сіток зараз унаслідок їхньої допомоги VLSM, несуміжних сіток та інших функцій, що стануть обговорені в найпізніших рубриках.

На (рис. 1.6) зауважте, що безкласова версія мережі використовує маски підмережі /30 і /27 у тій самій топології. Також зауважте, що ця топологія використовує несуміжний дизайн.

Безкласовими протоколами маршрутизації є RIPv2, EIGRP, OSPF, IS-IS, BGP.

Шлюз (Gateway) - це мережевий прилад, створений для з'єднання 2-х сіток (передавання між ними користувацького трафіку), які володіють різними рисами, використовують різні протоколи або технології. Gateway має можливість діяти на кожному з 7 значень моделі взаємодії відкритих систем (OSI). Мережі, що сполучаються, мають усі шанси володіти різними швидкостями передавання, затримками, вправами збереження. Крім усього цього мають всі шанси вживатися різні протоколи (TCP і UDP), технології (ATM і Ethernet), і навіть середовища передавання (оптичне волокно і літаюча пара). І ще знайшли широке використання стовідсотково бездротові шлюзи, які, наприклад, застосовують технологію WiFi на рівні доступу, а для взаємозв'язку із зовнішніми мережами - стільникові системи взаємозв'язку.

Одним із найпоширеніших способів застосування Gateway є забезпечення доступу з локальної мережі (LAN) у зовнішню мережу, наприклад Інтернет. При цьому в LAN може використовуватися одна технологія, а в зовнішньому з'єднанні - інша:

Ethernet - xDSL, PDH - Ethernet, SDH - ATM тощо. Також шлюз може виконувати завдання брандмауера, бути точкою початку VPN або бути сервером аутентифікації.

Служба загального доступу (sharing). Комп'ютерна мережа за визначенням являє собою розподілену систему. Її призначення - спільна робота користувачів. Така робота передбачає доступ до мережевих ресурсів, як-от файли і каталоги, принтери тощо. Для користувача звернення до мережевих ресурсів має бути прозорим, тобто:

- віддалені ресурси мають мати такий вигляд, немов вони є локальними і звернення до них із застосунків відбувається однаково.
- клієнту має бути байдуже, яка платформа використовується як сервер загального доступу.

При цьому потрібно враховувати й те, що не кожен користувач має мати доступ до конкретного ресурсу, і не кожен ресурс має бути доступний усім. Тобто необхідно забезпечити можливості управління загальним доступом як на рівні користувачів, так і на рівні окремих ресурсів.

Зазначені можливості надають спеціальні протоколи загального доступу. Найпоширенішими з них є SMB/CIFS для ОС Windows і NFS, який використовується в UNIX-подібних системах [22].

1.4. Asterisk

Asterisk - це повноцінна програмна АТС. У неї є можливість діяти на таких операційних системах, як Linux, BSD, Windows і OS X, та надає Вам усі здібності, які є у звичайної міні АТС, причому навіть більше. Функціонування Asterisk базується на протоколах, які гарантують передавання голосу через мережі, що базуються на IP-протоколі (VOIP), і, завдяки цьому, ця АТС має змогу діяти фактично з будь-яким устаткуванням для IP-телефонії, що вживають стереотипні протоколи для VOIP, при цьому використовуючи відносно недороге апаратне забезпечення. Asterisk надає функції голосової пошти (Voicemail), конференцій, інтерактивного голосового раціоналу (IVR), центру обробки викликів та їх обробки (Call Queuing). Він, окрім іншого,

має підтримку таких сервісів, як переведення викликів іншому абоненту, сервіс визначення і передавання абоненту, що викликається, номера абонента, що викликає (callerID), протоколи ADSI, SIP, H. 323 (як у режимі терміналу, так і в режимі гейтування), MGCP (лише для call manager) та SCCP/Skinny (не стовідсотково) [29].

Asterisk не потребує додаткового апаратного забезпечення для реалізації передавання голосу через IP мережі (VOIP). Застосовують єдиного (або декількох) провайдерів VOIP для вхідних і/або викликів (вхідні та вихідні виклики мають всі шанси оброблятися різними інтернет-провайдерами Веб і/або звичайної телефонії).

Для забезпечення комунікацій між цифровим і аналоговим телефонним обладнанням, Asterisk має підтримку деяких апаратних приладів, які можливо позначити, як обладнання, вироблене спонсором проєкту Asterisk - Digium. Digium виробляє одинарні та зчетверенні T1 і E1 інтерфейсні плати для комунікацій з каналами PRI і channel banks. На додаток, виконуються інтерфейси, від 1-го до 4 портів на плату, для комунікації з аналоговими телефонними рисами FXO і FXS, які затребувані при побудові невеликої АТС. Інтерфейсні пристосування інших виробників мають всі шанси бути застосовані для організації каналів BRI (ISDN2) з чотирма і вісьмома портами BRI, якщо вони сумісні з CAPI або базуються на чипсеті HFC [8].

Asterisk - це конвергований перрон для телефонії з відкритим початковим кодом, розроблений, переважно, для виконання на Linux. Більш ніж 100-річний навик телефонного взаємозв'язку дозволив зробити вірний пакет тісно вбудованих телекомунікаційних додатків. Міць Asterisk - у її настроюваній природі в поєднанні з співвідношенням стандартів, що не має аналогів. Не кажучи вже про те, що, Asterisk має можливість вбудовуватися з іншими бізнес-технологіями такими методами, про які закр вузькоспеціалізовані офісні АТС можуть тільки мріяти [29].

Огляд функцій Asterisk [30]:

- ACD Features - сервіси механічного розосередження сигналів;
- Advanced User Interface - інтерфейс для користувача з функціями: Yield I, Abide by I, Coherent Messaging, Angelus recordings, Voicemail і записом ситуації дзвінків;

- Analog Phone Support - підтримка аналогової телефонії методом пропуску сигналів через VoIP-шлюз;
- ANI Routing (Automatic Number Identification) - відображення інформації про дзвінок, що надійшов, і маршрутизація вхідних викликів на базі ANI;
- Announcement Interface - інтерфейс привітань IP PBX, механічне привітання під час вхідного дзвінка стане різним для робочого часу, а також для неробочого і weekenda/урочистих днів;
- Auto call forwarding - перенаправлення дзвінка, функція відповідає за переадресацію дзвінка на інший номер телефонного апарата (вручну або автоматом), залежно від того, де реєструється місцеперебування абонента - в кабінеті або за його межами;
- BYO Phones - додавання свіжих номерів до вашого переліку телефонних апаратів;
- Call Barge - перегляд на телефоні панелі управління для відстеження інтенсивних дзвінків;
- Call Parking - паркування дзвінка, переведення виклику на віртуальну телефонну лінію (з постановкою на утримання) для того, щоб трохи пізніше відвідувача опрацював цей самий або інший оператор. Щоб прийняти запаркований виклик, оператору потрібно на телефонному агрегаті набрати номер віртуальної смуги;
- Call Return - автоматичне повернення переведеного дзвінка в разі невідповіді - назад секретарю;
- Call Screening - екранування вхідних дзвінків: система просить абонента представитися, потім ставить вхідний дзвінок на утримання, і доповідає оператору про вхідний дзвінок;
- Caller ID Customization - зміна власного Visitor Vital impulse (номера телефонного апарата, що визначається) під час вихідних дзвінків;
- Caller Position Notification - сповіщення абонента про номер у черзі в переліку надії;
- Cell Phone Integration - інтеграція мобільників в IP PBX;

- Channel Bank Support - долучення аналогових ліній до високошвидкісного каналу E1 шляхом перетворення голосу в цифрову форму і мультиплексування з поділом часу;
- Chan Witness - часткове прослуховування конкретних поточних дискусій операторів;
- Conference Rooms - створення та управління віртуальними конференціями;
- CRM Integration - інтеграція IP АТС з CRM системою;
- Custom Routing Rules - завдання верховодил для маршрутизації дзвінків;
- Custom VoIP Provider - включення до IP-провайдера;
- Dial by Name Directory - набір телефонного апарата за прізвищамиDID's (Direct Inward Dialing) - прямі номери, підключені до IP АТС через операторів зв'язку;
- DNIS Routing - маршрутизація дзвінків, заснована на номері призначення;
- Enhanced Mobility - поліпшені мобільні функції, наприклад, впровадження бездротових Bluetooth-гарнітур;
- Extension Call Recording - запис вхідних/вихідних дискусій для різних внутрішніх номерів, транків;
- Extension Groups - категорії операторів з єдиним внутрішнім номером;
- Extension Range Flexibility - спектр внутрішніх номерів;
- Fax PDF Support - підтримка пов'язаних із передаванням факсів у форматі .pdf;Fax Support - підтримка приймання факсів;
- Fax to Email - надсилання факсу на електричну пошту;
- Find Me / Follow Me - перенаправлення дзвінка на різні номери телефонних апаратів працівника, наприклад, на офісний телефонний апарат, потім на мобільний, потім на побутовий, потім дзвінок на голосову пошту;
- Inbound Call Description - опис характеристик з'єднання;
- Inbound Call Whisper - видача звістки оператору, який приймає дзвінок, про те, хто телефонує і звідки;

- Integrated Paging - інтеграція текстових звісток;
- Interactive Voice Response (IVR) - інтерактивне голосове меню, що дає змогу абоненту зробити вибір натисканням клавіш телефону, позбавляючи операторів рутинної роботи;
- Intuitive VoIP Ready - ймовірність включення до різних провайдерів, незалежно від їхніх форматів передавання цих, а також під'єднання телефонних ліній на величезну кількість вихідних дзвінків;
- Least Cost Routing (LCR) - маршрутизація телефонних викликів за аспектом меншої ціни дзвінка;
- Multiple Music on Hold - ймовірність брати на озброєння не стільки 1 композицію, скільки кілька різновидів музики в період надії, прямо до включення музичної бібліотеки або потокового інтернет-радіо;
- NIS Routing - маршрутизація на основі інформації мережевої інформаційної служби (маршрутизація за протоколом NIS);
- Operator Console - консоль оператора;
- Outbound Dial Map - карта вихідної маршрутизації;
- Outlook Integration - інтеграція IP АТС із поштовим відвідувачем Outlook;
- Overhead Paging - оголошення для працівників компанії через телефонні апарати, що мають функцію звукового взаємозв'язку;
- Powerful Reporting - збір звітів про ідеальні дзвінки;
- PSTN or Digital Failover - перемикання на дублювальні канали, на телефонну мережу сукупного доступу або цифрові канали;
- Remote Linked Servers - взаємозв'язок Reference mark з віддаленими серверами;
- Remote Phones - віддалений телефонний доступ до системи;
- Speed Dial Numbers - клавіші жвавого комплекту номера;
- System Diagnositcs - діагностика системи і цих про користувачів;
- Time of Day Routing -переадресація дзвінка залежно від часу доби;

- Unlimited Voicemail Boxes - величезна кількість скриньок для голосової пошти;
- Voicemail Blast Groups - обрати перелік потрібних номерів і зробити з їхньої категорії категорію, щоб можна було кинути голосове сповіщення одразу на кілька номерів (усій групі);
- Voicemail Bypass - перемикання на оператора в обхід голосової пошти;
- Voicemail Web Access - доступ до голосової пошти через веб-інтерфейс;
- VoIP Compression - стиснення голосу під час передавання за протоколом VoIP, перебуває залежно від застосовуваного кодека;
- VoIP Ready - функція ввімкнення до хоч якого активного VoIP-провайдера з переліку головних [10].

Можливості моніторингу Asterisk. Підтримка SNMP для збору статистики. IP АТС Asterisk має підтримку SNMP-протоколу, що дає змогу здійснювати моніторинг кількості з'єднань, навантаження на систему та мережеві інтерфейси, доступність VoIP-пірів, а також повідомляти та реагувати на проблеми.

Панель оператора. Використання Панелі оператора для візуалізації під'єднань користувачів і дзвінків, що здійснюються, перехоплення, керування конференціями, прослуховування розмов, числа тих, хто очікує в черзі, підключених операторів, припаркованих викликів.

Безпека. Використання паролів для обмеження доступу та авторизації. Практично будь-яку функцію АТС можна поставити "під пароль" у вигляді PIN-коду, наприклад, міжнародні та міжміські дзвінки, вихід до іншого офісу, участь у конференції, заняття "резервної лінії" тощо. Під час дзвінка з міста абонент може вибрати в меню опцію, що дає йому змогу авторизуватися за PIN-кодом і віртуально переміститься в контекст свого офісного телефона, тобто здійснювати дії "від імені" свого телефона, як-от перевірка голосової пошти, внутрішні та зовнішні дзвінки, керування режимами переадресації тощо [29].

Мережеві можливості. Міжмережевий екран, маршрутизатор і VPN сервер. Оскільки Asterisk працює на ОС Linux і FreeBSD, що зарекомендували себе як надійні та високопродуктивні мережеві операційні системи. Це дає змогу поєднати функції

АТС і мережевого маршрутизатора з функціями NAT, сервера доступу в інтернет або віддаленого доступу з підтримкою PAP/CHAP авторизації, створити на базі Asterisk PBX віртуальну приватну мережу з використанням протоколів IPSec, LT2P, OpenVPN [30].

1.4. Концепція клієнт/сервер і мережа

Прогресивні комп'ютерні технології націлені на ідеологію клієнт/сервер у масштабах мережі. Широко поширене уявлення, власне сервер - це основний комп'ютер, що традиційно діє десь за стіною, далеченько; а замовник - це традиційно ми самі та наша автомашина, вочевидь.

У цьому контексті мається на увазі щось інше. Сервер - це програма, яка дає сервіс, тобто отримує запит і виділяє відповідь. Клієнт - це програма, яка дає запит і отримує відповідь. Так би мовити, випадок - це ролі, якими програмки мають всі шанси обмінятися в процесі справи. Одна й та сама програма у взаємодіях із різними партнерами може бути одразу сервером і відвідувачем. Сервер має можливість одразу обслуговувати трохи відвідувачів. Сучасне розуміння мережі містить три рівні:

- комп'ютер як мережа;
- локальна мережа;
- глобальна мережа.

Комп'ютер, забезпечений справжньою ОС із мережевою підтримкою протоколів TCP/IP (як Linux, наприклад), володіє чудовою властивістю. Можна створити систему замовник/сервер у масштабах 1-го компа, а використовувати цю систему як на окремому комп'ютері, так і в локальній мережі та в мережі інтернет. Заявимо, систему, в якій сервер керує прикладною інформаційною базою, а програми-клієнти дають користувачеві прикладний інтерфейс для доступу до бази через запити до сервера по мережі.

Це виходить завдяки тому, що ОС для взаємодії відвідувача і сервера біля однієї машини вживає такі самі протоколи, власне і для взаємодії відвідувача і сервера через масштабну мережу.

Для DOS і MS Windows продані мережеві пакети, які підтримують протоколи TCP/IP. Щоб влаштувати справжній сервер цього замало, хоча в ролі відвідувачів програмки DOS і MS Windows мають усі шанси вбудовуватися в мережу.

Локальна мережа складається з низки автомашин, які вживають сукупну трасу цих. Наприклад, Ethernet. Можливі й інші варіанти, наприклад, включення автомашин через почергові порти. Локальна мережа, зазвичай, обмежується межами будівлі.

Локальні мережі зв'язуються одна від одної, утворюючи масштабну мережу. Найбільша мережа інтернет - це Internet. Вона з'єднує воєдино воєдино

45 000 субмереж.

Робити мережу - так справжню. У себе на підприємстві користувач будує мережу в стандартах Internet, у вигляді сукупності локальних мереж по корпусах, потім йде підключення до Internet і зв'язується мережею зі спорідненими підприємствами. Необхідно звернути увагу на те, що при цьому програма-клієнт працює з програмою-сервером абсолютно однаково, що на машині, де встановлено сервер, що з іншого корпусу, що з іншого міста і навіть іншої країни.

1.5. Роль Linux у створенні сучасної мережі підприємства

Linux займає значну роль у такій розробці. Linux має всі необхідні інструменти, він вільний і комфортний. Це може вважатися чудовим полігоном для обкатки рішень. І якщо ви потім вирішите перейти на іншу платформу (наприклад, потужний сервер Sun і ОС Solaris), то це буде зроблено не просто так, тому що разом із поточною платформою будуть захоплені напрацювання.

Це досягається за рахунок кількох переваг Linux. По-перше, Linux має здатність підтримувати стандарти відкритих систем.

По-друге, величезна кількість інструментальних пакетів існує в Linux, за допомогою цих пакетів реалізується прикладна система клієнт-сервер. Це СУБД, побудовники графічних інтерфейсів тощо. Ці пакети є вільними, вони поставляються у вихідних текстах. Вони генеруються з вихідних текстів як для Linux, так і для десятків інших платформ, у т.ч. комерційних - Solaris, SCO, BSD*.

Наприклад, СУБД Exodus, Postgres тощо. Пакет для розробки інтерфейсу користувача Tcl/Tk. Останній може бути використаний як інтерпретатор і як бібліотека програмних викликів. Tcl може генеруватися на десятки інших платформ: Linux, BSDI, SunOS, Solaris, Ultrix, IRIX, AIX, OSF/1, HPUX, SCO Unix, Xenix, Dynix, UMAX, Unicos, A/UX, QNX, VMS, NextStep, Convex, DOS, MS Windows, OS/2, ...

Якщо вам необхідно зробити перенесення прикладної системи на іншу платформу, вам з легкістю вдасться зробити це, оскільки ви маєте не тільки вихідні тексти вашої прикладної системи, а й вихідні тексти використаних інструментальних пакетів, які ви можете вільно прихопити з собою на законній основі.

Особливо виділяється роль Linux як платформи для знайомства розробників-початківців і користувачів зі світом відкритих систем. По-перше, Linux є комфортною і повноцінною 32-х розрядною ОС, за якістю порівнянною, а в низці аспектів перевершує комерційні ОС. По-друге, користувач економить на витратах, пов'язаних з освоєнням Linux, і витрачається у багато разів менше, ніж у випадку з комерційною системою. І не треба платити за додаткові підсистеми і пакети програм, немає обмежень на кількість користувачів. При цьому доступ до вихідних текстів! І не у вигляді прикладу з підручника, а у вигляді тексту реально працюючої програми.

Результат - дорогоцінний досвід за мінімальних витрат, який вбереже від дорогих помилок у сфері комерційних систем.

Особливість X Window System як мережевої системи.

X Window System (X Windows) є віконно-графічною системою клієнт/сервер.

Сервер X Windows виконується на машині, тут необхідно відображати інформацію, у найчастішому випадку до однієї машини під'єднано один дисплей, але буває і кілька. Можлива робота кількох серверів на одній машині, кожен із них обслуговує свій дисплей. Дисплей це клавіатура, мишка, планшет тощо у зв'язці з монітором або кількома моніторами (наприклад, звичайним і великим графічним).

Клієнт є програмою, яка посилає дані серверу для того, щоб вони були відображені у вікні (вікнах) на моніторі. Також, приймає дані від сервера, наприклад, рядок, введений з клавіатури. Клієнт має можливість виконуватися як на тій самій машині, що й X сервер, так і на іншій машині Internet.

Припустимо, ви працюєте на машині my_host і на одній із консолей запустили X Window System, тобто X сервер. Потім, працюючи на іншій консолі або у вікні, ви завантажилися у віддалену машину remote_host і хочете виконати X додаток, тобто прикладну програму, призначену для роботи в X Windows. Для того, щоб повідомити X застосунку, що ви хочете бачити його віконця у себе на моніторі, достатньо ввести у віддалену машину команду

```
export DISPLAY=my_host:0.0
```

Якщо ви викличте X додаток, то він відкриє через мережу канал зв'язку з вашим X сервером на машині my_host, а не з X сервером машини remote_host, і відобразить усі свої вікна на вашому екрані. Таким чином, дані передаються каналом зв'язку між клієнтом (X додатком) і X сервером так само, як якщо б вони працювали на одній машині. Виникає ілюзія, що робота відбувається на одній машині, оскільки мережа прозора. :0.0 це просто конкретні значення :<номер дисплея>.<номер монітора>, характерні для випадку машини з одним монітором.

Windows X реалізована на величезній кількості машин різної архітектури в різних операційних системах. Таким чином, існує можливість виконувати застосунок на потужному (супер) комп'ютері через мережу, а отримувати результат і керувати роботою застосунку на скромній персоналці. Існує можливість одночасно відкрити кілька вікон на своєму моніторі - один додаток на машині в Сан-Франциско, другий - у Сіднеї, третій - у Парижі, - і працювати з ними одночасно, достатньо тільки переміщати мишкою (або програмно) дані між ними.

Спеціальні пристрої створені для роботи в мережах із системою X Windows - X-термінали. X-термінал є спеціалізованим комп'ютером, це - апаратне втілення X сервера з вбудованим мережевим інтерфейсом. Попри те, що він досить дорогий, це високоякісний пристрій із чудовою графікою.

Linux з X Windows на персональному комп'ютері являє собою гідну уваги заміну X-терміналу.

По-перше, це недороге рішення порівняно з іншими. Навіть якщо в якості дисплея буде використано хороший 17-дюймовий монітор.

По-друге, це щось більше, ніж просто X-термінал. Він універсальний, його одночасно можна використовувати для вирішення широкого спектра завдань.

Існують й інші програмні реалізації X-терміналів для DOS та MS Windows. Вони є комерційними системами, проте всі вони значно поступаються за швидкістю дій і надійністю системі X Windows в Linux.

Встановивши Linux у розділ DOS і викинувши все зайве, з'являється можливість отримати програмний X-термінал, який завантажується командою в DOS прямо з файлової системи DOS. На диску це займе близько 10 MB.

1.6. Завантаження Linux через мережу

Мережевий завантажувач Linux викликається таким чином:

- з постійної пам'яті (ПЗП) мережевої плати на бездисковій машині;
- з флоппі диска;
- як програма DOS, що виконується відразу ж після завантаження.

Для того, щоб можна було використовувати мережевий завантажувач, необхідно в мережі запустити відповідний сервер, який взаємодіє із завантажувачем. Linux, завантажений по мережі, працює з файловою системою через мережу.

Взаємодія Linux з DOS і MS Windows. У системах DOS і MS Windows існує підтримка протоколів TCP/IP, що дає змогу виконувати програми-клієнти, які через мережу TCP/IP взаємодіють з Linux і дають змогу завантажуватися в Linux, обмінюватися файлами, електронною поштою та новинами, редагувати файли через мережеву файлову систему NFS (комерційні PCNFS, PCTCP, WATTCP, WinQVT, WINARCH, EINet winWAIS, вільні WinVn, WS_FTP, NCSA Telnet, NCSA Mosaic). У Linux існують відповідні програми-сервери.

Сервер Samba, який є в Linux, дає змогу отримати доступ до файлової системи Linux і принтера, що працює в Linux програмам-клієнтам через протокол SMB (Session Message Block). До таких клієнтів належать LanManager для DOS, Windows

for Workgroups, Windows NT, OS/2, Pathworks і багато інших. До пакета Samba входить і програма-клієнт, яка дає змогу отримати з Linux доступ до файлів і принтера в Windows for Workgoups, OS/2.

Протокол SMB виконується поверх протоколу TCP/IP [23-25].

РОЗДІЛ 2
ЗАБЕЗПЕЧЕННЯ QoS В ОС LINUX

Якість обслуговування, або, як найчастіше кажуть, **QoS (Quality of Service)** - характеристика, що визначає якість і клас обслуговування з передавання потоку даних, надаваного користувачеві через мережу, яка є критичною за часом. Тут немає певних жорстких норм, однак, заведено вважати, що постійний плавний перебіг бесіди можливий тільки тоді, коли на доставлення звуку, який виробляється мовцем, до вуха слухача витрачається 150 мс. У разі коли затримка перевищує 300 мс, учасники бесіди починають перебивати один одного. Нормальна розмова не буде можливою, якщо затримка перевищить 500 мс.

Крім виконання вимог щодо часу, необхідно щоб була гарантія, що інформація не пошкоджується під час передавання. Відновлення дискретизованого аудіосигналу на дальньому кінці не можливе при втраті величезної кількості пакетів, отже, прогалини в даних чутимуться як перешкоди або, в більш важких випадках, пропуски цілих слів або речень. Вважається, що навіть втрата 5% пакетів може сильно пошкодити мережі VoIP [13].

Існує ключова проблема, яка потребує розв'язання, на сьогоднішній день Інтернет має можливість забезпечити лише негарантовану доставку даних (best effort service), що не передбачає надання будь-яких гарантій, які стосуються часу і самого факту прибуття пакета в пункт призначення [14].

2.1. Транспортні протоколи в мережах передавання даних

TCP, UDP і SCTP. Для передавання даних мережею, яка працює за IP-протоколом, використовується один із трьох обговорюваних тут транспортних протоколів.

Протокол керування передачею. Transmission Control Protocol (TCP) практично не застосовується для VoIP, оскільки, попри те, що він має механізми забезпе-

чення гарантованої доставки, вони фактично не використовуються. TCP має схильність створювати більше проблем, ніж знаходити їхні рішення, у багатьох з'єднаннях і може бути застосований тільки в з'єднаннях між двома кінцевими точками з надзвичайно малим часом очікування. Гарантувати доставку пакетів є головним призначенням TCP. Декілька механізмів реалізуються для цього, таких як нумерація пакетів (для відновлення блоків даних), підтвердження доставки та повторний запит загублених пакетів. Швидка доставка пакетів у кінцеву точку є першочерговим завданням у світі VoIP, але за 20 років використання стільникового зв'язку вже звично, що є нестача кількох пакетів.

Такі характеристики як ретельне опрацювання, управління станом і підтвердження доставки роблять TCP чудовим протоколом для передавання великих обсягів даних, однак він недостатньо ефективний для передавання медіаданих у реальному масштабі часу.

User Datagram Protocol. Порівняно з TCP, User Datagram Protocol (UDP) не пропонує жодних гарантій доставки. Пакети передаються по дротах дуже швидко, як це можливо, і випускаються у світ без будь-якої інформації про їхню доставку або не доставку до свого звичайно призначеного пункту. Оскільки UDP не надає жодних гарантій доставки даних, його ефективність забезпечується дуже невеликими витратами на транспортування.

Протокол передачі потокового контролю. Схвалений IETF як пропонований стандарт у RFC 2960, SCTP є порівняно новим транспортним протоколом. У самому

На початку він розроблявся як протокол, позбавлений недоліків TCP і UDP, він був призначений насамперед для сервісів, які зазвичай надаються комутованими телефонними мережами. Деякими з цілей SCTP були: кращі техніки запобігання перевантаженню (зокрема, запобігання атакам типу "відмова в обслуговуванні"), суворо впорядкована доставка даних, значно нижча затримка для поліпшення передачі в режимі реального часу.

Розробники SCTP мали надію створити надійний протокол для передачі SS7 та інших типів сигналів PSTN через IP-мережі, позбавлений основних недоліків TCP і UDP [24].

2.2. Механізми QoS

Нині дедалі більше рішень у сфері комунікацій найрізноманітнішого масштабу використовують IP-протокол і локальні мережі Ethernet. При цьому є можливість мінімізувати витрати завдяки використанню вже наявного обладнання, наявним програмним рішенням і навіть набутим знанням і досвіду місцевих фахівців. Спрощує роботу включення в загальний проєкт уже наявних комунікацій. IP-трафік, що забезпечує роботу програм, передаватиметься каналами LAN і шлюзами, що з'єднують мережеві сегменти. Як приклад можна навести варіант організації інформаційного повідомлення між рознесеними офісами шляхом реалізації VPN.

Одна з проблем IP-мереж - це ефективний контроль трафіку. Методи розв'язання цієї проблеми ґрунтуються на застосуванні політики, що дає змогу гармонійно розподілити інформаційний потік. Провідні розробники мережевого обладнання намагаються втілити ідею використання політики контролю трафіку в технічні рішення. Як правило, реалізуються методи QoS (Quality of Service) і CoS (Class of Service), які, втім, пов'язані між собою.

Потрібно звернути увагу, що протокол IP, створений для передавання даних в інтернеті, не обіцяє доставку даних. Цей же протокол, працюючи в локальній мережі Ethernet, не набув достатньої якості, також покладаючись на принцип best effort service (негарантована доставка даних). Якщо ж мережа компанії використовується лише для обміну даними, то, можливо, помітного постійного наростання трафіку в ній не помічається, а можливі затримки під час роботи в мережі або непомітні користувачам, або не важливі. Якщо навіть у разі додавання нових користувачів у групу потрібно тільки пропорційно зростити пропускну здатність каналів, щоб забезпечити потрібну якість мережевих сервісів.

Абсолютно інша ситуація складається під час використання додатків, для роботи яких критичні затримки і швидкість потоку даних. Розширення пропускного каналу мережі зробить швидкість обміну даними лише вищою, але не запобіжить несподіваному збільшенню до критичного значення потоку пакетів. Не дивлячись на

це може спостерігатися переповнення буфера роутера або комутатора. У такому положенні відбувається руйнування частини мережевих пакетів. Також є очікування втрати пакетів другорядних програм, а не додатків міжмережевого обміну, наприклад e-mail. Для того щоб забезпечити ефективну роботу всіх додатків, потрібно змінити логіку функціонування мережі. Слід відмовитися від негарантованої доставки даних, що не дає жодних привілеїв програмам. У випадках, коли обсяг трафіку або затримки в локальній мережі можуть збільшуватися в межах 10-30%, правильно буде задіяти QoS/CoS-політику як спосіб контролю трафіку. Якщо ж організація думає використовувати відеоконференцію або систему групової роботи, в локальній мережі потрібно закріпити потрібний ресурс за певними додатками. До речі, розробники пропонують кілька схем, що дають змогу ефективно використовувати схему QoS/CoS у локальній мережі [24-27].

Легше застосувати схему пріоритезації мережевого трафіку. IEEE та Internet Engineering Task Force підтримують вісім однакових рівнів пріоритету IEEE у специфікації 802.1p (802.1d), а IETF застосовує спеціальні біти в IP-заголовку. Компанія 3COM однією з перших здійснила підтримку цієї специфікації у своїх комутаторах і роутерах під фірмовою назвою 3Com Transcend Policy Manager. Важливо, що провідні виробники мережевого обладнання, зокрема 3COM і Cisco, працювали з розробниками основних ОС для досягнення сумісності обробки бітів пріоритету в заголовках IP-пакета.

Вона, наприклад, зможе дізнатися, власне пакети, що належать до додатків CRM, мають перевагу в доставці перед пакетами інших типів - обміну файлами або інтернет-пейджерів. Якщо несподівано трапиться ситуація, коли частина пакетів унаслідок перевантаження стане не потрібна, то насамперед це трапиться з пакетами з низьким пріоритетом.

У робочих групах, де потрібно забезпечити особливий рівень затримок мережевих передач, гарним стане використання схеми QoS, виготовленої в еталоні ATM і RSVP (Resource Reservation Protocol). Дана схема підсобляє знайти контроль над пропускною можливістю, затримками і точністю дотримання цінностей (над тим, які пакети стануть викинуті в ситуації переповнення черги).

Так, RSVP здатний обіцяти те, власне затримка мережевого передавання не перевершить особливого максимуму, а ATM, крім усього іншого, рекомендує контроль за тремтінням - хитанням затримок і часом переміщення пакетів від однієї критичної точки LAN до іншої.

Якщо фірма не має потреби в системі управління трафіком на цей момент, можливо, власне ця потреба з'явиться надалі. Необхідно звертати увагу на те, щоб придбане мережеве обладнання, приміром комутатори, які встановлюють на кордоні мережі, підтримували 2-ий рівень і застосовували схему пріоритезації за еталоном 802.1p (802.1d) [25].

Із завданням формалізації зазначених проблем і рішень у 1998 році IETF зробила робочу категорію зі створення диференційованих послуг (DiffServ). Основне завдання диференційованих послуг - скооперувати вимоги щодо якості сервісу цих із принципом негарантованої доставки пакетів. У цій моделі трапляється поділ трафіку методом його розбивки на класи з різною цінністю.

Пріоритезація, іменована і ще CoS, ділить трафік за нижченаведеними категоріями: найвища, центральна і невисока, також винятково невисока, пакети якої мають усі шанси бути видалені. E-mail і веб-трафік часто відносять до особисто невисокої групи, і в разі впевненості завантаження мережі вони стануть відкинуті в першу чергу.

Сервіси, націлені на доставку пакетів, діють згідно з CoS, інакше кажучи, можна виокремити ті пакети, які варто привезти спочатку, вдруге тощо. Ця ідентичність не випадкова, адже CoS можна розглядати як підклас QoS. Хоча на відміну від неї, пріоритезація трафіку не дає гарантію того, чи власне ці пакети переміщені в конкретний час.

Зараз виключно ATM має всі механізми допомоги QoS. Для локальних сіток присутній RSVP. Він вважається IETF-стратегією допомоги QoS у мережах TCP/IP і є чимось більшим, ніж пріоритизація і CoS. RSVP передбачає розосередження смуги пропускання та її резервування під явний вид трафіку. Це досить непростий для реалізації протокол. Є й інший спосіб реалізації QoS в IP - MPLS (Multiprotocol Achievement Switching).

Зараз лише ATM має всі механізми допомоги QoS. Для локальних сіток присутній RSVP. Він вважається IETF-стратегією допомоги QoS у мережах TCP/IP і є чимось більшим, ніж пріоритизація і CoS. RSVP передбачає розосередження смуги пропускання та її резервування під явний вид трафіку. Це досить непростий для реалізації протокол. Є й інший спосіб реалізації QoS в IP - MPLS (Multiprotocol Label Switching).

Напевно, зараз одним з оптимальних шляхів управління трафіком (що вимагає істотно найменших витрат, ніж реалізація політичні діячі QoS) має можливість працювати поділ смуги пропускання на трохи часток для певних справ. У разі, як тільки зростає перевантаження, пакети починають буферизуватися мережевим обладнанням, що власне призводить до затримок. Коли буфер маршрутизатора (комутатора) переповнюється, деякі пакети взагалі мають всі шанси бути втрачені.

За появи проблем з буферизацією схема пріоритезації може допомогти вислати першочергові пакети або пакети, сприйнятливі до затримок, раніше за інших. Водночас потрібно, щоб ці пакети були правильно класифіковані та поміщені в чергу з відповідним рівнем профілактики. Щоб досягти бажаного результату, порт мережевого приладу зобов'язаний уміти систематизувати пакети, також читати мітку класу вхідного пакета і поміщати його в чергу стека, котра відсортована за значущістю пакетів. При цьому пакети, що перебувають на початку черги, будуть оброблені швидше.

Пріоритезація може бути задіяна і ще в мультипротокольних маршрутизаторах і комутаторах, щойно систематизація стане проводитися за принципом протоколу. Наприклад, ці, що відносяться до SNMP-протоколу, стануть визначені до класу з найвищою цінністю. Є й інші схеми маркування CoS:

- пріоритезація на основі MAC-адреси;
- VLAN-мітки та 802.1p. IEEE 802.1Q - стандарт, який описує спосіб, що дозволяє вставляти мітку в IEEE MAC-кадр. Ця мітка служить для визначення приналежності до тієї чи іншої VLAN. Три біти такої мітки можуть нумерувати вісім певних рівнів пріоритету. Таке поєднання бітів буде сприйнято мережевими пристроями як припис віднести пакет до будь-якого класу обслуговування;

- TOS. Заголовок IP-пакета має поле, яке називається TOS (Type Of Service). Це поле наразі перевизначено для використання в контексті IETF-диференціювання послуг (Differentiated Services - Diff-Serv). Diff-Serv класифікує і маркує пакети таким чином, що за ними закріплюється певний маршрут у ланцюжку мережевих пристроїв. Встановлення біта TOS відбувається на основі певної політики контролю і надалі зчитуватиметься мережевими пристроями. Розроблений для інтернету протокол IP сприймається в різних мережах, а отже, Diff-Serv може забезпечити CoS у LAN, extranet або Intranet.

Сьогодні робляться спроби стандартизувати опис класів обслуговування і поступово позбутися використання поля TOS на користь DSCP (Diff-Serve Code Point). Фактично поле DSCP являє собою розширення поля IP-пріоритету завдяки використанню бітів, що визначають типи обслуговування. Схематично трансляція заголовка має такий вигляд:

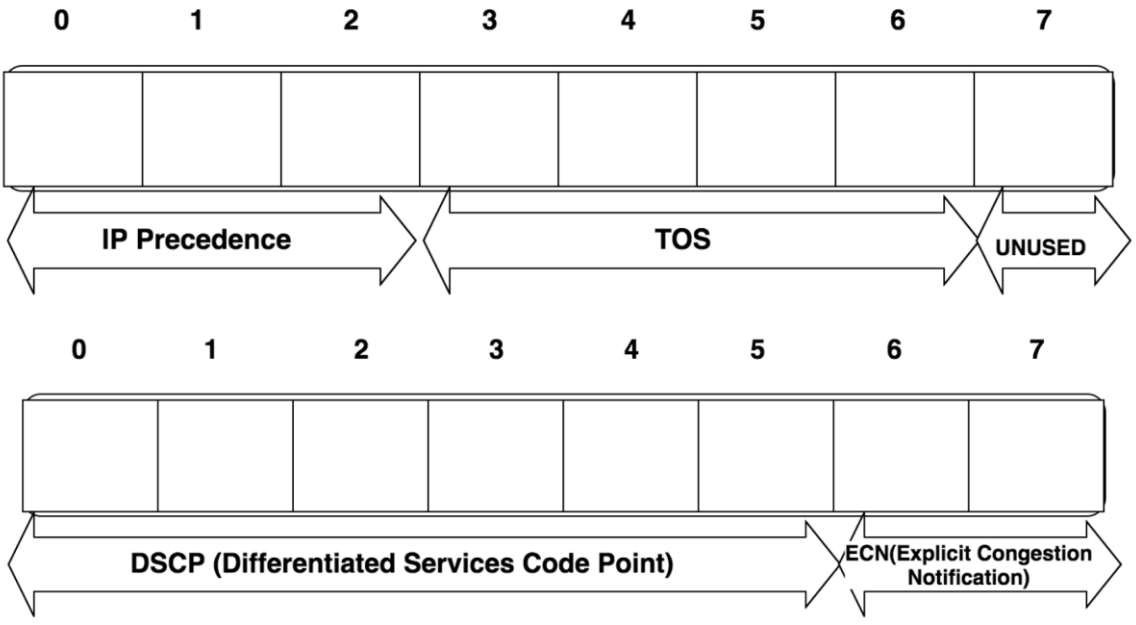

Рис. 2.1. Трансляція заголовка

Систематизація пакетів може бути зроблена різними методами. Найідеальніше, якщо це стане виготовлено користувацьким додатком. Хоча комп'ютерні програми,

зазвичай, не підтримують усі схеми систематизації, тому це завдання здебільшого передбачають робити багаторівневим комутаторам, розміщеним на кордоні мережі [26].

Диференційоване обслуговування. Диференційоване обслуговування, або DiffServ, - не стільки механізм QoS, скільки метод, за допомогою якого можна маркувати трафік і забезпечувати йому спеціальне обслуговування. Очевидно, що DiffServ може сприяти забезпеченню QoS, надаючи перевагу певним типам пакетів над іншими.

Під час використання різними додатками спільно однієї й тієї самої мережі, можливо відповідати вимогам QoS, проєктуючи мережу так, щоб обробка всього трафіку йшла відповідно до вимог або була диференційованою обробкою, що ґрунтується на класі трафіку. Така диференційована обробка вимагає, щоб кожен пакет був позначений. Цього можливо досягти, використовуючи поля ToS у стандартному IP-заголовку. Збільшена потреба в QoS мотивувала відновлення інтересу до методів використання цих бітів у заголовку пакета [28].

Хоча, безсумнівно, це підвищує шанси VoIP-пакета швидко пройти через усі з'єднання, але не дає твердих гарантій.

Гарантоване обслуговування. Гарантовану якість і клас послуг, що надаються, забезпечує PSTN. Для кожної розмови використовується виділений тільки під цей дзвінок канал зі швидкістю передавання даних 64 Кбіт/с; пропускна здатність гарантується. Так само протоколи, що пропонують гарантоване обслуговування, можуть забезпечити виділення під обслуговуване з'єднання необхідної смуги пропускання. Як для будь-якої мережевої технології з пакетуванням, ці механізми зазвичай найкраще працюють в умовах, коли обсяг трафіку нижчий за максимально допустимі рівні. Якщо з'єднання досягає своїх граничних значень, практично неможливо уникнути погіршення якості обслуговування.

MPLS. Multiprotocol Label Switching (MPLS) - це метод розробки та управління моделями мережевого трафіку незалежно від таблиць маршрутизації третього (мережевого) рівня. Суть роботи протоколу полягає у присвоєнні мережевим пакетам коротких міток (кадрів MPLS), які потім використовуються маршрутизатором для пере-

силання пакетів на вихідний маршрутизатор MPLS та зрештою - до їхнього остаточного місця призначення. Традиційно маршрутизатори приймають незалежне рішення про пересилання на підставі пошуку в IP-таблиці під час кожного переходу в мережі. У мережі MPLS такий пошук виконується тільки один раз, коли пакет входить в MPLS-хмару на вхідному маршрутизаторі. Після цього пакету призначається потік, званий Label Switched Path (LSP) та ідентифікований за міткою. Мітка використовується як індекс пошуку в таблиці пересилання MPLS, і пакет проходить по LSP незалежно від рішень маршрутизації третього рівня. Це дає змогу адміністраторам великих мереж тонко налаштовувати рішення з маршрутизації та використовувати мережеві ресурси з максимальною ефективністю. Крім того, з міткою може бути пов'язана інформація, що визначає пріоритетність пакета під час пересилання.

RSVP. У MPLS немає методу для динамічного встановлення LSP, але для цього в поєднанні з MPLS можна використовувати Reservation Protocol (RSVP). 247 RSVP - це протокол обміну сигналами, який використовується для спрощення завдань зі встановлення LSP і передання інформації про проблеми, що виникають, на вхідний маршрутизатор MPLS. Перевага використання RSVP у поєднанні з MPLS - скорочення витрат на адміністрування. Якщо не використовувати RSVP з MPLS, доведеться вручну конфігурувати всі мітки і кожен шлях на всіх маршрутизаторах. Застосування RSVP робить мережу більш динамічною за рахунок передачі функції управління мітками маршрутизаторам. Таким чином, мережа стає більш чутливо реагує на умови, що змінюються, і може бути налаштована на зміну шляхів, виходячи з певних умов, наприклад, якщо якийсь із шляхів недоступний (можливо, через вихід з ладу маршрутизатора). У цьому разі конфігурація маршрутизатора зможе використовувати RSVP для розподілу нових міток серед маршрутизаторів MPLS-мережі без жодного втручання людини (або за мінімального втручання).

Негарантоване обслуговування. Найпростіший і найдешевший підхід до QoS - не надавати якості послуг взагалі. Це називається негарантованим обслуговуванням. Ймовірно, звучить не дуже добре, але цей метод може дуже непогано працювати.

Будь-який виклик VoIP, що проходить відкритою мережею Інтернет, практично напевно буде викликом із негарантованим обслуговуванням, оскільки механізми QoS у цьому середовищі ще не набули широкого поширення [28].

2.3. Забезпечення якості в Asterisk

Правильний вибір інструментів для певної галузі впровадження QoS вважається складним завданням. Ми маємо певні методи формування трафіку, механізми попередження перевантаження і досить велику кількість алгоритмів планування пакетів. У будь-якому певному випадку рекомендовано ознайомитися з усіма RFC пов'язаних з QoS [29].

По-перше, згадане поле TOS (Type of service) тепер називається DSCP (Differentiated Services Code Point), воно замінює поле TOS і вказується для IPv4 і IPv6 (для довідки RFC2474 є специфікацією).

Таким чином підтримується зворотна сумісність із полем TOS. Більшість мереж використовують такі класи трафіку:

- PHB за замовчуванням - це, як правило, трафік з найкращими зусиллями;
- Expedited Forwarding (EF) PHB - призначена для трафіку з низькими втратами і низькою латентністю;
- Assured Forwarding (AF) PHB - дає гарантію доставки за встановлених умов;
- Class Selector PHBs - які підтримують зворотну сумісність із полем IP Precedence.

Таким чином EF, CS3 і AF41 означають, тільки загальний спосіб сигналізації, що наш пакет має важливе значення, що не так вже й багато. Але тільки помічені пакети нам не допоможуть. Необхідно правильно класифікувати і визначити пріоритети пакетів на сервері Asterisk.

Для контролювання трафіку в Linux є чудовий інструмент tc для конфігурування та налаштування політики QoS. З його допомогою ми можемо налаштувати різні види черг дисциплін і класів. Ці черги діють безпосередньо на мережеві пристрої, тож ми маємо налаштувати їх на нашому сервері Asterisk.

TC дозволяє конфігурувати класові та безкласові дисципліни, кожна з яких підтримує різні алгоритми планування. Після того, як наші черги налаштовані, ми повинні повідомити системі IPTables порядок використання черг.

Ми налаштуємо максимальну пропускну спроможність у даному випадку в 1024 Кбіт висхідної лінії зв'язку, якій ми збираємося додати політику QoS. Таблиця 2.1 ілюструє політику QoS, необхідну для роботи в мережі. Оскільки ми використовуємо асиметричне з'єднання, ми будемо обмежувати пропускну здатність завантаження до 95% від номінальної швидкості [30].

Таблиця 2.1

Політики QoS

Клас	Номінальна швидкість	Максимальна швидкість	Пріоритет	Пакети
Даний час	47,5 кбіт/с	95кбіт/с	0	ICMP, SYN, RST, ACK
Вищий	522,5 кбіт/с	950 кбіт/с	1	EF і пакети CS3
Регулярний	190 кбіт/с	950 кбіт/с	2	Регулярний трафік, HTTP, SSH тощо
Основний	190 кбіт/с	950 кбіт/с	3	

РОЗДІЛ 3
РОЗРАХУНОК ЯКОСТІ ОБСЛУГОВУВАННЯ IP ТЕЛЕФОНІЇ

3.1. Розрахунок інтенсивності навантаження на сервер

Для того щоб розрахувати інтенсивність навантаження на сервер, нам необхідні такі дані (таблиця 3.1), отримані шляхом статистичного аналізу за місяць.

Таблиця 3.1

Статистика середньої кількості викликів у робочий час

Час дзвінка	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Кіль-кість дзвінків на годину	7	10	15	20	10	17	20	17	9	5

Разом виходить, у середньому, навантаження на сервер Asterisk, становить 130 викликів на день.

Для розрахунку інтенсивності навантаження використовуємо формулу:

$$A = \lambda \cdot T, \tag{3.1}$$

де T - випадкова величина, середня тривалість обслуговування одного виклику, вимірювана в одиницях часу.

λ - середня частота надходження викликів, що вимірюється числом викликів за одиницю часу, виклик/год. Є загальною характеристикою випадкового потоку викликів [21].

Характеристики навантаження - середню частоту надходження викликів λ, трафік А - зазвичай оцінюють для години пік, тобто для годинного інтервалу в період найбільшого навантаження системи зв'язку.

З таблиці 1 розраховуємо середню кількість дзвінків на годину:

$$\lambda = \frac{N}{t_{роб}}, \tag{3.2}$$

де N - кількість вхідних і вихідних дзвінків у робочий час

$t_{роб}$ - година дзвінка.

$$\lambda = \frac{130}{10} = 13 \text{, вик/год}$$

Середня тривалість обслуговування одного виклику (Т) розраховується за формулою:

$$T = t_{рс} + t_{св} + n + t_{вст} + t_{пв/кпв} + t_{роз} + t_{звіл}, \tag{3.3}$$

де $t_{рс}$ = 0,1 с - час реакції системи комутації, визначений як проміжок часу від моменту посилки абонентом сигналу "заняття" на станцію до моменту отримання сигналу "відповідь станції";

$t_{св}$ = 3 с - середній час слухання сигналу "відповідь станції";

n - число знаків в абонентському номері, тоновий набір одного символу становить 0,5 с. Для дзвінка з міста-супутника в м.Київ набираються цифри: 8727XXXXXXXXXXXXXX - 11 символів, 0,5 *11=5,5 с;

$t_{вст}$ = 2 с - середній час встановлення з'єднання;

$t_{пв/кпв}$ = 7 - 8 с - середній час видачі сигналів "посилки виклику" і "контроль посилки виклику";

$t_{звіл}$ = 1с - середній час звільнення телефонного тракту для з'єднання, що закінчилося розмовою;

$t_{роз}$ - середня тривалість розмови.

За статистичними даними, середня тривалість розмови становить 214 секунд. Отримуємо

$$T = 0{,}1 + 3 + 5{,}5 + 2 + 8 + 214 + 1 = 233{,}6c$$

Переводимо секунди в години: 233,6 сек = 0,065 години. Тепер підставляємо дані у формулу 3.1:

$$A = 13 \cdot 0{,}065 = 0{,}845 \text{ Ерл}$$

Швидкість передавання інформації залежить значною мірою від швидкості її створення, способів кодування та декодування. Найбільша можлива в даному каналі швидкість передачі інформації називається його пропускною здатністю. Пропускна спроможність каналу, за визначенням, є швидкість передавання інформації за умови використання "найкращих" для даного каналу джерела, кодера і декодера, тому вона характеризує тільки канал. Оскільки середовищем передавання дзвінків є мережа Інтернет або будь-яка інша IP-мережа, потрібно прорахувати швидкість каналу, необхідну для якісного передавання голосу мережею. Для розрахунку необхідно знати: характеристики використовуваного кодека і середню кількість дзвінків на годину. Розрахуємо необхідну швидкість передачі даних по каналу за формулою:

$$C = \lambda \cdot R, \tag{3.4}$$

де, R - швидкість передачі кодека, кбіт/с.

За регламентом кодування корисної інформації за допомогою кодека G.729 становить 8 кбіт/с, насправді окрім корисної інформації, кодується службова інформація, швидкість передачі кодека становить близько 16-20 кбіт/с [22].

$$C = 13 \cdot 20 = 260 \text{ кбіт/с}$$

Під час проведення розрахунків було з'ясовано, що для передавання каналом зв'язку середньої кількості дзвінків на годину потрібен канал зв'язку швидкістю 260 кбіт/с. Цю швидкість буде вирізано із загальної швидкості, що надається провайдером для компанії. Виділена швидкість для каналу сервера становитиме 256 кбіт/с.

3.2. Розрахунок показника якості транспорту пакетів в IP-телефонії

Наступним кроком необхідно розрахувати показники якості транспорту пакетів - а саме, затримку, зумовлену стоянням пакета в черзі та часом, витраченим на передачу. Оскільки для обслуговування пакетів IP-телефонії та інших послуг використовуються різні черги, то розрахунок вестимемо окремо, виходячи з типового розміру пакета для кожного виду послуг. Нижче наведено детальний розрахунок, а для інших відділень результати розрахунків зведено в таблицю.

Для телефонії реальний розмір пакета визначено розрахунковим шляхом - 1900 біт. Пропускна здатність каналу - 1 Мбіт/сек. Отже, час, необхідний для передачі 1 пакета, визначається так

$$T_S = \frac{b_{реальн}}{B_{канала}},$$

$$T_S = \frac{1900}{1} = 1{,}9 мс.$$

58 , (3.5)

Оскільки розрахункове навантаження абонентської групи дорівнює A=0,845 Ерланг, а інтервал передавання пакетів кожним абонентом (час семпла) t_s=20 мс, то інтенсивність передавання пакетів λ від усієї групи буде розрахована так:

$$\lambda = \frac{A}{t_S},$$

$$\lambda = \frac{0{,}845}{20 \cdot 10^{-3}} = 42{,}25 \; пак / сек, \tag{3.6}$$

Звідки визначаємо коефіцієнт утилізації ρ - показник завантаженості каналу зв'язку, який впливає на час стояння в черзі

$$\rho = \lambda \cdot T_S,$$

$$\rho = 42{,}25 \cdot 1{,}9 \cdot 10^{-3} = 0{,}08, \tag{3.7}$$

З нього, згідно з [23], враховуючи той факт, що всі пакети телефонії мають однаковий розмір, визначаються такі показники якості. Загальна кількість пакетів r, що перебувають у системі:

$$r = \frac{\rho^2}{2 \cdot (1 - \rho)} + \rho,$$

$$r = \frac{0{,}08^2}{2 \cdot (1 - 0{,}08)} + 0{,}08 = 0{,}083, \tag{3.8}$$

З них у черзі на обслуговування перебувають w пакетів

$$\omega = \frac{\rho^2}{2 \cdot (1 - \rho)}, \tag{3.9}$$

$$\omega = \frac{0{,}08^2}{2 \cdot (1 - 0{,}08)} = 0{,}0035$$

Ця цифра може бути корисною при налаштуванні черг на обладнанні - в апаратурі можна вказувати максимальний розмір черги пакетів. У цьому випадку в системі на обслуговуванні менше 1 пакета, значення досить умовне; воно свідчить про те, що

система працює з великим запасом за продуктивністю. Але за подальшої оптимізації ця цифра буде вже набагато реальнішою.

Час стояння пакетів у черзі Tw.

$$T_w = \frac{\rho \cdot T_S}{2 \cdot (1-\rho)},$$

$$T_w = \frac{0{,}08 \cdot 1{,}9}{2 \cdot (1-0{,}08)} = 0{,}28 мс, \quad (3.10)$$

Повний час перебування пакета в системі Tr, включно з час обслуговування і час стояння в черзі

$$T_r = \frac{T_S \cdot (2-\rho)}{2 \cdot (1-\rho)};$$

$$T_r = \frac{1{,}9 \cdot (2-0{,}08)}{2 \cdot (1-0{,}08)} = 3{,}52 мс., \quad (3.11)$$

Як видно, цей час значно менший, ніж необхідний показник якості 150 мс для телефонії. Отже, можна зменшити пропускну здатність каналу зв'язку, і тим самим заощадити витрати.

3.3. Розрахунок показників якості в інтернет

Розрахунок показників якості для послуги обміну з інтернетом розраховуються подібним чином. Середню пропускну здатність, необхідну для комфортного завантаження сторінок і роботи електронної пошти, визначено емпірично в 512 Кбіт/сек, середній розмір пакета - теж емпірично, 350 байт (що дорівнює 2800 біт). З цих даних можна визначити середній інтервал передавання пакетів ts від одного абонента - за формулою:

$$t_S = \frac{b_{реальн}}{B_{абон}},$$

$$t_S = \frac{2800}{512} = 5{,}47мс, \tag{3.12}$$

Для пропускання навантаження в 3 Ерл (Оптимальне навантаження для Інтернет) буде потрібно 10 таких каналів:

$$B_{послуги} = B_{абонента} \cdot V, \tag{3.13}$$

де V - кількість каналів. Тоді:

$$B_{послуги} = 512 \cdot 10 = 5120 \text{ кбіт / сек}$$

Визначимо час обслуговування одного пакета:

$$T_S = \frac{2800}{5120} = 0{,}546мс$$

Тоді інтенсивність передавання пакетів визначається за формулою 6

$$\lambda = \frac{3}{5{,}47 \cdot 10^{-3}} = 548{,}44пак / сек$$

Коефіцієнт утилізації ρ - за формулою 7

$$\rho = 548{,}44 \cdot 0{,}546 \cdot 10^{-3} = 0{,}302$$

Потім визначаються показники якості; але, на відміну від телефонії, пакети мають неоднакову довжину. У першому наближенні можна вважати, що розмір пакетів

підпорядковується експоненціальному розподілу, хоча це і не зовсім так. Для розрахунку показників якості за експоненціального розподілу використовують такі формули [23, 26].

Кількість пакетів у системі:

$$r = \frac{\rho}{1-\rho},$$
$$r = \frac{0{,}302}{1-0{,}302} = 0{,}433 \tag{3.14}$$

Кількість пакетів у черзі:

$$\omega = \frac{\rho^2}{1-\rho},$$
$$\omega = \frac{0{,}302^2}{1-0{,}302} = 0{,}131, \tag{3.15}$$

Час стояння пакетів у черзі Tw

$$T_w = \frac{\rho \cdot T_S}{1-\rho},$$
$$T_w = \frac{0{,}302 \cdot 0{,}546}{1-0{,}302} = 0{,}115 мс, \tag{3.16}$$

Повний час перебування пакета в системі Tr, включно з час обслуговування і час стояння в черзі

$$T_r = \frac{T_S}{1-\rho},$$
$$T_r = \frac{0{,}546}{1-0{,}302} = 0{,}782 мс, \tag{3.17}$$

Звернемо увагу, що час знаходження пакета обміну з інтернету в системі набагато менший, ніж той самий час для телефонних пакетів, хоча вимоги до якості транспорту телефонії набагато вищі, ніж транспорту інтернету.

Резервів для економії пропускної спроможності тут набагато більше [22, 26]. Результати розрахунку наводяться в таблиці 3.2.

Таблиця 3.2

Розрахунок показників якості

Вид показника	Телефонія	Інтернет
Середня проп. здатність для послуги	1 мбіт/сек	512 кбіт/сек
Загальне навантаження групи	0,845 Ерл	0,845 Ерл
Середня інтенсивність передачі пакетів	42,25 пак/сек	548,44 пак/сек.
Час обслуговування пакета	1,9 мс	0,546 мс
Коефіцієнт утилізації	0,08	0,302
Кількість пакетів у системі	0,083	0,433
Кількість пакетів у черзі	0,0035	0,131
Час перебування пакета в системі	3,52 мс	0,782 мс
Час стояння в черзі	0,28 мс	0,115 мс

ВИСНОВКИ

У кваліфікаційній роботі мною було запропоновано варіант реалізації Інтернет і VoIP шлюзу для філіальної корпоративної мережі невеликої компанії, що забезпечує необхідну якість обслуговування на низькошвидкісній лінії зв'язку і дає змогу істотно заощадити бюджет компанії на апаратно-програмному забезпеченні, необхідному для його реалізації.

В основі проєкту лчгла безпечна і багатофункціональна операційна система з відкритим кодом Linux, що активно розвивається.

У хоті роботи було вивчено та розглянуто основні характеристики й технічні особливості цієї операційної системи, її мережеві можливості, зроблено огляд прикладного рішення Asterisk на базі вільного програмного забезпечення.

Детально було розглянуто питання забезпечення якості обслуговування засобами ОС Linux. Було здійснено аналіз наявних технологічних рішень цього завдання. Программною реалізацією розглянутих механізмів було обрано інструмент t_c - спеціальну утиліту для конфігурування та налаштування політик QoS. У результаті було запропоновано bash-скрипт (Додаток А), що використовує ієрархії Token Bucket (HTB) для пакетів, маркованих Астеріском, Stochastic Fairness Queuing (SFQ) і ланцюжки IPTables.

У розрахунковій частині були проведені обчислення інтенсивності мережевого навантаження на сервер. Оскільки проєктований сервер виконує функції Інтернет і VoIP-шлюзів, то було обчислено навантаження від послуги телефонії, яке склало 0,845 Ерл. Були проведені розрахунки швидкості каналу, необхідної для якісного передавання голосу і даних із мережі Інтернет. Повний час перебування пакета в системі склав 3,52 мс, що набагато менше необхідного рівня в 150 мс. Для Інтернет-пакетів час знаходження пакета в системі становив 0,782 мс, що свідчить про економію пропускної здатності і можливості звуження ширини каналу зв'язку для економії коштів підприємства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. R. Hoseinyfarahabady, Z. Tari and A. Y. Zomaya, "Disk Throughput Controller for Cloud Data-Centers," 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Gold Coast, QLD, Australia, 2019, pp. 404-409.
2. O. Reich, E. Hübner, B. Ghita, M. Wagner and J. Schäfer, "Bluetooth Performance Evaluation based on Notify for Real-time Body-Area Sensor Networks," 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 2020, pp. 516-520.
3. X. Jiang, Y. Zhang, W. Fu, X. Yang, Y. Sun and Z. Sun, "TASP: Enabling Time-Triggered Task Scheduling in TSN-Based Mixed-Criticality Systems," 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), Oslo, Norway, 2022, pp. 1-11.
4. S. Gallenmüller, J. Naab, I. Adam and G. Carle, "5G QoS: Impact of Security Functions on Latency," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-9.
5. D. Perić, V. Marinković, M. Perić and S. Usorac, "Analysis of minimum system requirements for running Android OS inside a Linux container," 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2022, pp. 239-241.
6. R. K. Boggavarapu and S. Jiang, "Deduplication-aware I/O Buffer Management in the Linux Kernel for Improved I/O Performance and Memory Utilization," 2020 12th International Conference on Knowledge and Smart Technology (KST), Pattaya, Thailand, 2020, pp. 70-74.
7. R. Karayat, M. Jadhav, L. S. Kondaka and A. Nambiar, "Web Application Penetration Testing & Patch Development Using Kali Linux," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1392-1397.

8. R. Karayat, M. Jadhav, L. S. Kondaka and A. Nambiar, "Web Application Penetration Testing & Patch Development Using Kali Linux," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1392-1397.
9. P. Cui, Z. Liu and J. Bai, "Linux Storage I/O Performance Optimization Based on Machine Learning," 2022 4th International Conference on Natural Language Processing (ICNLP), Xi'an, China, 2022, pp. 552-557.
10. B. Sachdeva, A. Kushwaha, A. Kumar and A. Tiwari, "Analysis of Linux Server Performance," 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1-4.
11. M. Dmitry and P. Elena, "Linux Privilege Increase Threat Analysis," 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russia, 2020, pp. 0579-0581.
12. B. Khan, C. Heinz and A. Koch, "DeLiBA: An Open-Source Hardware/Software Framework for the Development of Linux Block I/O Accelerators," 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL), Belfast, United Kingdom, 2022, pp. 183-191.
13. C. -C. J. Huang and C. -F. Yang, "An Empirical Approach to Minimize Latency of Real-Time Multiprocessor Linux Kernel," 2020 International Computer Symposium (ICS), Tainan, Taiwan, 2020, pp. 214-218.
14. N. Georgoulopoulos, A. Hatzopoulos, K. Karamitsios, I. M. Tabakis, K. Kotrotsios and A. I. Metsai, "Investigation and Simulation of Hardware Errors in Kernel Logs of Linux-based Server Systems," 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 2021, pp. 1-7.
15. R. A. Lara-Cueva, S. Pazmiño and F. Acosta, "Performance evaluation of an Asterisk PBX prototype Beaglebone Black based," 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 2022, pp. 1-5.

16. Y. Shu, J. Zhang, D. Zuo and Q. Z. Sheng, "Interval-Valued Skyline Web Service Selection on Incomplete QoS," 2022 IEEE International Conference on Web Services (ICWS), Barcelona, Spain, 2022, pp. 361-366.
17. X. Luo, J. Xie, L. Xiong, Z. Wang and C. Tian, "3-D Deployment of Multiple UAV-Mounted Mobile Base Stations for Full Coverage of IoT Ground Users with Different QoS Requirements," in IEEE Communications Letters, vol. 26, no. 12, pp. 3009-3013.
18. Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin and S. Mubeen, "QoS-MAN: A Novel QoS Mapping Algorithm for TSN-5G Flows," 2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Taipei, Taiwan, 2022, pp. 220-227.
19. J. Zhang, F. Li and F. Ye, "Network traffic clustering with QoS-awareness," in China Communications, vol. 19, no. 3, pp. 202-214.
20. H. Awane, Y. Ito and M. Koizumi, "Study on QoS Estimation of In-vehicle Ethernet with CBS by Multiple Regression Analysis," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 1425-1427.
21. M. D. Sazak and A. M. Demirtas, "UAV-BS Trajectory Optimization Under Coverage, Backhaul and QoS Constraints Using Q-Learning," 2022 International Balkan Conference on Communications and Networking (BalkanCom), Sarajevo, Bosnia and Herzegovina, 2022, pp. 157-161.
22. S. Huang, "Real-time QoS Analysis of Mental Health Monitoring Guidance by Cognitive Psychological Remote Data Transmission System," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1667-1671.
23. T. Mahboob, J. W. Lim, S. T. Shah and M. Y. Chung, "A Novel Deep-Learning-Enabled QoS Management Scheme for Encrypted Traffic in Software-Defined Cellular Networks," in IEEE Systems Journal, vol. 16, no. 2, pp. 2844-2855.

24. M. Beshley, V. Kochan, H. Beshley, M. Medvetskyi, I. Kahalo and Y. Shkoropad, "QoS-Coordinated Adaptive Spectrum Management Method for Coexistence 5G-U and Wi-Fi Networks with Short-Term Channel Failures," 2023 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Jaroslaw, Poland, 2023, pp. 22-26.
25. A. BenDahmene, O. Kazar, K. Rezeg and A. e. Merizig, "A planning framework for QoS based composition services in IoT environment," 2022 International Symposium on iNnovative Informatics of Biskra (ISNIB), Biskra, Algeria, 2022, pp. 1-5.
26. L. Magoula et al., "A Deep Learning Approach for Distributed QoS Prediction in Beyond 5G Networks," 2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Kyoto, Japan, 2022, pp. 1407-1412.
27. H. S. Yoo and W. E. S. Yu, "Building a QoS Testing Framework for Simulating Real-World Network Topologies in a Software-defined Networking Environment," 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, 2022, pp. 1-6.
28. S. Sivakumar, R. Anusuya, V. Nagaraju, L. P. Narendruni and R. Thamizhamuthu, "QoS Based Efficient Link and Consistent Routing in Wireless Sensor Network," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, pp. 1241-1246.
29. P. Kadam, M. Kulkarni and V. Gaikwad, "Bandwidth Management for VoIP Calling Through Asterisk," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-6.
30. F. Dong, B. Deng, H. Yu, W. Xie, H. Xu and Z. Gu, "An Asterisk-shaped Patch Attack for Object Detection," 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), Guilin, China, 2022, pp. 126-133.