

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ДОПУЩЕНО ДО ЗАХИСТУ**

Завідувач кафедри \_\_\_\_\_ Є. О. Давиденко  
*підпис*

«17» лютого 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**Програмна система контролю, обліку та доступу в Інтернет абонентів  
оператора телекомунікацій**

Спеціальність «Інженерія програмного забезпечення»

121–КРМ.1–608м мз.221508001

***Студент***

\_\_\_\_\_ К.О. Кірей  
*підпис*

«17» лютого 2023 р.

***Керівник***

канд. техн. наук, доцент \_\_\_\_\_ Г. В. Горбань  
*підпис*

«17» лютого 2023 р.

***Консультант***

\_\_\_\_\_ С. М. Братякін  
*підпис*

«17» лютого 2023 р.

***Консультант***

д-р біол. наук, професор \_\_\_\_\_ Л. І. Григор'єва  
*підпис*

«17» лютого 2023 р.

**Миколаїв – 2023**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Зав. кафедри \_\_\_\_\_ Є. О. Давиденко  
*підпис*  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи магістра**

Видано студенту групи 608мз факультету комп'ютерних наук

Кірей Катерині Олександрівни

*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи

Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

Затверджена наказом по ЧНУ від 10 січня 2023 р. № 1/1

2. Строк представлення кваліфікаційної роботи 24.02.2023 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Працездатна програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій для ТОВ Телекомунікаційна компанія «НЕОН»

Вимоги до програмної системи з боку ТОВ Телекомунікаційна компанія «НЕОН»

4. Перелік питань, що підлягають розробці предметна галузь та аналогічні програмні системи зі схожим функціоналом; технології для розроблення програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій; база даних; показники системи щодо складних умов експлуатації.

5. Перелік графічних матеріалів

презентація

6. Завдання до спеціальної частини

Вплив електромагнітного випромінювання на людину

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
провідний інженер-електронік С. М. Братякін	ДП НВКГ «Зоря» - «Машпроект»	фахова частина
професор Григор'єва Л. І.	кафедра екології ЧНУ ім. Петра Могили	спеціальна частина з охорони праці

Керівник роботи \_\_\_\_\_ доцент Горбань Гліб Валентинович

*(посада, прізвище, ім'я, по батькові)*

\_\_\_\_\_  
*(підпис)*

Завдання прийнято до виконання \_\_\_\_\_ Кірей Катериною Олександрівною

*(прізвище, ім'я, по батькові студента)*

\_\_\_\_\_  
*(підпис)*

Дата видачі завдання 03.10.2022 р.

## КАЛЕНДАРНИЙ ПЛАН

### виконання кваліфікаційної роботи

Тема: «Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРМ	03.10.2022	05.10.2022	виконано
2.	Огляд літератури за темою роботи	10.10.2022	21.10.2022	виконано
3.	Складання календарного плану КРМ	24.10.2022	25.10.2022	виконано
4.	Аналіз предметної області	26.10.2022	04.11.2022	виконано
5.	Розробка проектних рішень	07.11.2022	18.11.2022	виконано
6.	Моделювання та конструювання ПЗ	21.11.2022	02.12.2022	виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування	05.12.2022	23.12.2022	виконано
8.	Проведення дослідження	26.12.2022	14.01.2023	виконано
9.	Розробка спеціальної частини з охорони праці	16.01.2023	28.01.2023	виконано
10.	Відгук керівника КРМ	30.01.2023	03.02.2023	виконано
11.	Оформлення КРМ та презентації	06.02.2023	08.02.2023	виконано
12.	Попередній захист	13.02.2023	13.02.2023	виконано
13.	Рецензування	14.02.2023	15.02.2023	виконано
14.	Завершення оформлення КРМ та презентації	16.02.2023	16.02.2023	виконано
15.	Захист кваліфікаційної роботи			

Розробив студент Кірей Катерина Олександрівна  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

«\_\_» \_\_\_\_\_ 2023 р.

Керівник роботи доцент Горбань Гліб Валентинович  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

«\_\_» \_\_\_\_\_ 2023 р.

## АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій»

Студент 608мз гр.: Кірей Катерина Олександрівна

Керівник: канд. техн. наук, доцент Горбань Г. В.

Магістерська кваліфікаційна робота присвячена розробці, програмної реалізації та дослідженню системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій.

**Об'єктом кваліфікаційної роботи** є процеси контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій.

**Предметом кваліфікаційної роботи** є метод організації контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій із використанням PPPoE-з'єднання та RADIUS протоколу.

**Метою кваліфікаційної роботи** є поліпшення організації доступу в Інтернет абонентів оператора телекомунікацій ТОВ Телекомунікаційна компанія «НЕОН».

**Методами дослідження** є теоретичні: аналіз джерел інформації за проблемою дослідження (наукової та технічної літератури, керівництв користувачів, документації тощо), аналіз технологій та протоколів, що використовуються для авторизації, автентифікації та обліку доступу абонентів в Інтернет; студіювання вітчизняного та зарубіжного досвіду проектування і побудови аналогічних систем за темою дослідження. Емпіричні: налагодження та тестування розробленої системи; спостереження за роботою системи в реальних умовах та аналіз одержаної системної інформації.

Кваліфікаційна робота складається з фахової частини і спеціальної частини з охорони праці. Фахова частина складається зі вступу, п'яти розділів, висновків та додатків. У першому розділі описано предметну галузь, основні поняття та специфікацію вимог. У другому розділі описано проектування системи, наведено діаграми варіантів використання, діяльності та розгортання; описано інтерфейс користувача. У третьому розділі описано вибір інструментальних засобів та технологій розробки та описано етапи програмної реалізації системи. У четвертому розділі описано налагодження та тестування розробленої системи. У п'ятому розділі описано результати дослідження працездатності системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій у складних умовах експлуатації. У спеціальній частині з охорони праці розкрито питання впливу електромагнітного випромінювання на людину.

Магістерська кваліфікаційна робота містить: 96 сторінок, 43 рисунка, 5 додатків, 30 джерел посилання.

Ключові слова: *автоматизована система розрахунків, мережа, Інтернет, RADIUS, Point-to-Point Protocol over Ethernet, послуга доступу до мережі Інтернет, оператор електронних комунікацій.*

## **ABSTRACT**

of the Master's Thesis

«Software billing system for a telecommunication operator»

Student of group 608mz: Kirei Kateryna Oleksandrivna

Supervisor: Ph.D. Sc., Associate Professor Horban H. V.

The Master's Thesis is devoted to the development, software implementation and research of the system of control, accounting and Internet access of subscribers of a telecommunications operator.

The object of research is the processes of control, accounting and Internet access of subscribers of a telecommunications operator.

The subject of research is the method of organizing control, accounting and internet access of subscribers of a telecommunications operator using PPPoE connection and RADIUS protocol.

The purpose of Master's Thesis is to improve the organization Internet access of subscribers of the telecommunications operator Ltd Telecommunications company "NEON".

Methods of research are theoretical: analysis of sources of information on the Research Problem (Scientific and technical literature, user manuals, documentation, etc.), analysis of technologies and protocols used for authorization, authentication and accounting of subscriber access to the Internet; study of domestic and foreign experience in designing and building similar systems on the research topic. Empirical: setting up and testing the developed system; monitoring the operation of the system in real conditions and analyzing the received system information.

The Master's Thesis consists of a professional part and a special part on labor protection. The professional part consists of an introduction, five sections, conclusions and appendices. The first section describes the subject area, basic concepts, and requirements specification. The second section describes the design of the system, provides diagrams of use cases, activities, and deployments, and describes the user interface. The third chapter describes the choice of development tools and technologies and describes the stages of software implementation of the system. The fourth section describes debugging and testing the developed system. The fifth chapter describes the results of a study of the operability of the system for monitoring, accounting and internet access of telecommunications operator subscribers in difficult operating conditions. In the special part on labor protection, the issue of the influence of electromagnetic radiation on humans is revealed.

The Master's Thesis contains: 96 pages, 43 figures, 5 appendices, 30 link sources.

*Keywords: Automated Payment System, Network, Internet, RADIUS, Point-to-Point Protocol over Ethernet, Internet Access, Operators of Electronic Communications.*

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ. СПЕЦИФІКАЦІЯ ВИМОГ .....	7
1.1 Опис предметної галузі .....	7
1.2 Point-to-Point Protocol over Ethernet мережевий протокол передачі кадрів через Ethernet .....	11
1.3 RADIUS протокол автентифікації, авторизації та збору даних про використовувані ресурси мережі.....	14
1.4 Огляд та аналіз аналогів .....	19
1.4.1 Автоматизована система розрахунків (АСР) .....	19
1.4.2 Наявні АСР .....	20
1.5 Специфікація вимог .....	26
Висновки до розділу 1 .....	29
2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ .....	31
2.1 Діаграма варіантів використання .....	31
2.2 Діаграма діяльності.....	33
2.3 Діаграма розгортання.....	35
2.4 Інтерфейс користувача .....	37
Висновки до розділу 2 .....	42
3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ.....	43
3.1 Вибір найбільш доцільної операційної системи.....	43
3.2 Мови програмування та середовище розробки.....	45
3.3 Створення модуля розширення RADIUS-сервера.....	48
3.4 Створення процедури перевірки прийняття Угоди.....	55
3.5 Схема бази даних .....	58
Висновки до розділу 3 .....	61
4 НАЛАГОДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	62
4.1 Налагодження системи .....	62
4.2 Тестування системи .....	64

Висновки до розділу 4 .....	69
<b>5 ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ В СКЛАДНИХ УМОВАХ ЕКСПЛУАТАЦІЇ .....</b>	<b>70</b>
Висновки до розділу 5 .....	76
<b>ВИСНОВКИ.....</b>	<b>77</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	<b>79</b>
ДОДАТОК А Конфігурація модуля rlm_isp .....	82
ДОДАТОК Б Масив структур CONF_PARSER .....	85
ДОДАТОК В Структура isp_user.....	87
ДОДАТОК Г Скрипт перевірки прийняття Угоди check_agreement.py .....	88
ДОДАТОК Д Акт впровадження.....	91



## **ПЕРЕЛІК СКОРОЧЕНЬ**

АСР – Автоматизована система розрахунків

БД – база даних

ГОМ – головна обчислювальна машина

ОС – операційна система

СКБД – система керування базами даних

AAA – Authorization, Authentication and Accounting

ADCCP – Advanced Data Communication Control Procedures (or Protocol)

CHAP – Challenge Handshake Authentication Protocol

CPN – Customer Premises Network

HDLC – High-Level Data Link Control

LAN – Local Area Network

LCP – Link Control Protocol

LLC – Logical Link Control

MAC – Media Access Control

MAN – Metropolitan Area Network

NAS – Network Access Server

NCP – Network Control Protocols

PAP – Password Authentication Protocol

PPP – Point-to-Point Protocol

PPPoA – Point-to-Point Protocol over ATM

PPPoE – Point-to-Point Protocol over Ethernet

RADIUS – Remote Authentication Dial-In User Service

UI – User Interface

UUCP – Unix-to-Unix CoPy

WAN – Wide Area Network

## ВСТУП

Одним із покликань людства є комунікація. Про це свідчить розвиток засобів комунікації впродовж усієї історії людства від появи писемності й до сучасних технічних засобів, таких як смартфони, комп'ютери тощо. Значний стрибок у цьому процесі був зумовлений появою і розвитком мережі Інтернет. Нині можна стверджувати, що мережа Інтернет є потужним джерелом інформації. Отже, усе більше й більше людей прагнуть отримати доступ до Мережі, а ще більша частина людства вже не уявляє свого існування без Мережі. В Україні діяльність у сфері телекомунікацій регламентується Законом України «Про електронні комунікації» та низкою нормативних актів. Організацією доступу абонентів до мережі Інтернет займаються оператори електронних телекомунікацій (оператори). Зокрема, до їхніх обов'язків входить організація доступу абонентів до мережі Інтернет. Організація доступу інтегрує в собі процеси авторизації, автентифікації та тарифікації абонентів і має задовольняти певним технічним, користувацьким та законодавчим вимогам. З огляду на те, що ці вимоги можуть змінюватися організація доступу є актуальним завданням.

**Об'єктом кваліфікаційної роботи** є процеси контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій.

**Предметом кваліфікаційної роботи** є метод організації контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій із використанням PPPoE-з'єднання та RADIUS протоколу.

**Актуальність теми кваліфікаційної роботи** полягає в створенні програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій, що буде відповідати сучасному розвитку засобів телекомунікацій та задовольняти технічним, користувацьким та законодавчим вимогам.

**Метою кваліфікаційної роботи** є поліпшення організації доступу в Інтернет абонентів оператора телекомунікацій ТОВ Телекомунікаційна компанія «НЕОН».

Для досягнення мети треба вирішити такі завдання:

- провести аналіз предметної галузі та аналогічних програмних систем зі схожим функціоналом;
- обрати технології для розроблення програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій;
- спроектувати та реалізувати базу даних;
- реалізувати програмну систему;
- дослідити її характеристики щодо складних умов експлуатації.

**Інновація** полягає:

- у розробленні вдосконаленого алгоритму тарифікації пакетів абонентів оператора телекомунікацій;
- у розробленні додаткових можливостей щодо керування доступом абонентами до мережі Інтернет;
- відмовостійкості системи під час пікових навантажень на сервера обробки даних.

**Апробація результатів КРМ.** Результати роботи було представлено на Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія» (7-10 лютого 2023 р., ЧНУ ім. Петра Могили).

**Публікації.** Результати роботи представлено тезами доповіді: Кірей К. О., Горбань Г. В. Вдосконалення системи контролю, обліку та доступу в інтернет абонентів оператора телекомунікацій. *Інформаційні технології та інженерія* : тези доп. Всеукр. наук.-практ. конф. молодих вчених, аспірантів і студентів. Миколаїв, 7–10 лютого 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023. С. 97–99.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ. СПЕЦИФІКАЦІЯ ВИМОГ

### 1.1 Опис предметної галузі

Нині ми не уявляємо життя без глобальної мережі Інтернет. Сервіси Інтернету широко використовуються майже у всіх галузях діяльності людей, вони спрямовані на задоволення потреб бізнесу, суспільства, школярів, студентів тощо. Окрім цього розвиток самого Інтернету сприяє розвитку багатьох технологій передачі та обробки даних. Отже, можемо стверджувати, що Інтернет став однією з необхідних складових життя людини.

Формування глобальної мережі відбувалося поступово, проходячи певні етапи, пов'язані з розвитком тих чи інших технологій [1]. За малим винятком, перші комп'ютери підключалися прямо до терміналів і використовувалися окремими користувачами, зазвичай, у тій же будівлі або приміщенні. Такі мережі стали відомі як локальні обчислювальні мережі (LAN). Мережі, що виходять за рамки локальних, відомі як глобальні обчислювальні мережі (WAN), з'явилися в 1950-х роках і були введені у 1960-х. Згодом з'явилися ще міські обчислювальні мережі (MAN). Доменна адресація в сучасному вигляді розвивається як міжнародний стандарт з 1987 року.

Спочатку свого розвитку система, яка згодом перетворилася в Інтернет, головно призначалася для використання урядом і державними органами. Тим не менш цікавість до комерційного використання Інтернету незабаром став широко обговорюваною темою. У 1980-х роках дослідження британського вченого Тіма Бернерса-Лі в ЦЕРН у Швейцарії призвели до створення Всесвітньої павутини в результаті з'єднання гіпертекстових документів в інформаційну систему, доступну з будь-якого вузла мережі [2, С.2]. Перший вебсайт Бернерса-Лі представив у 1991 році [3].

У результаті наприкінці 1980-х років були сформовані перші інтернет-оператори. Були створені компанії, такі як PSINet, UUNET, Netcom і Portal Software, для того щоб надавати послуги регіональним дослідним мережам, а

також надавати громадськості альтернативний доступ до мережі, ґрунтованої на протоколах UUCP електронній пошті та Usenet News. Першим комерційним інтернет-провайдером у США був The World, створений в 1989 році [4].

У 1992 році Конгрес США прийняв Закон про підтримку доступу до комп'ютерних мереж (розділ 42 Кодексу США, 1862 (g) [5]), який дав змогу Національному науковому фонду підтримувати доступ дослідних і освітніх спільнот до комп'ютерних мереж, які не використовувалися виключно для дослідних і навчальних цілей, що дало змогу NSFNET налагодити взаємозв'язок із комерційними мережами.

У 1994 році опорна мережа Інтернету була приватизована: комерційні компанії взялися доставляти інтернет-трафік на далекі відстані. Це дало змогу відмовитися від мережі NSFNet, яку фінансував уряд США. Чотирма найбільшими приватними провайдерами мережевого зв'язку на далекій відстані стали UUNetruen, AT&T, Sprint і Level 3 [1].

Нині можна спостерігати постійне збільшення кількості інформації, що передається на високих швидкостях (1 Гбіт/с, 10 Гбіт/с і вище [14]) через оптоволоконні мережі. Захоплення Інтернетом сфери глобального зв'язку за історичними мірками було майже миттєвим: в 1993 році Мережею передавалося лише 1% інформації, 51% – у 2000 році й понад 97% дистанційної інформації – у 2007 році [6]. Нині Інтернет продовжує розвиватися, про що свідчить стрімке збільшення інтернет-хостів у всьому світі за порівняно короткий період (рис. 1.1).

До кінця 2010-х років розвиток Інтернету поставив безліч глобальних питань, пов'язаних із захистом приватності, інтернет-цензурою, кібертероризмом тощо. У квітні 2022 року США, ЄС і ще 32 країни оприлюднили «Декларацію майбутнього Інтернету» [8]. Ініціатори декларації мають намір домагатися доступності, мережевого нейтралітету, видалення нелегального контенту без обмеження свободи самовираження, а також чинити опір зростаючому цифровому авторитаризму.

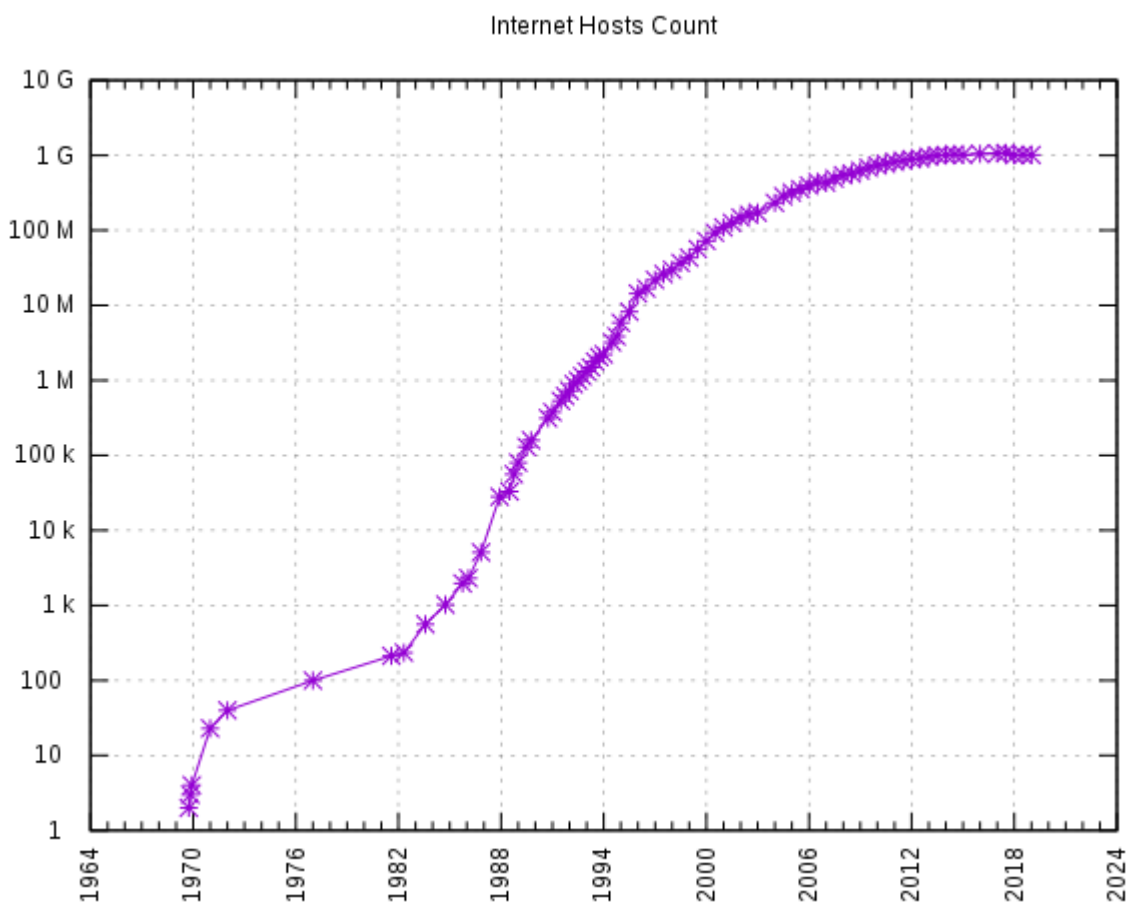


Рисунок 1.1 – Кількість інтернет-хостингів у всьому світі в 1970-2015 роках [7]

Отже, майже кожна людина у світі, зокрема, і в Україні прагне бути підключеною до світової Мережі та отримати доступ до її сервісів. В Україні діяльність у сфері телекомунікацій регламентується Законом України «Про електронні комунікації» та низкою нормативних актів. Організацією доступу абонентів до мережі Інтернет займаються оператори електронних телекомунікацій (оператори). Згідно із цим законом оператор електронних комунікацій (оператор) – це суб’єкт господарювання, який володіє, здійснює експлуатацію та управління електронними комунікаційними мережами та/або пов’язаними засобами [9]. Зокрема, до їхніх обов’язків входить організація доступу абонентів до мережі Інтернет.

Законом України «Про електронні комунікації» також визначено, що:

- послуга доступу до мережі Інтернет – це електронна комунікаційна послуга, що забезпечує доступ до мережі Інтернет і можливість логічного з'єднання з кінцевими точками мережі Інтернет незалежно від технології, що застосовується в електронній комунікаційній мережі, і кінцевого (термінального) обладнання, що використовується;
- доступ – це надання відповідно до цього Закону права та можливості доступу до електронних комунікаційних мереж, інфраструктури, засобів та/або послуг іншим постачальникам електронних комунікаційних мереж та/або послуг з метою надання ними електронних комунікаційних мереж та/або послуг, у тому числі для розповсюдження програм телерадіомовлення [9].

Отже, перед операторами постає питання: «Як організувати доступ користувачів до мережі Інтернет?». Протоколи, що відповідають за це містяться на другому рівні семирівневої моделі OSI – каналному [10]. Цей рівень являє собою рівень протоколу, який передає дані між вузлами в сегменті мережі через фізичний рівень. Канальний рівень передачі даних надає функціональні та процедурні засоби для передачі даних між мережевими об'єктами, а також може надавати засоби для виявлення і, можливо, виправлення помилок, які можуть виникнути на фізичному рівні.

Канальний рівень може взаємодіяти з одним або декількома фізичними рівнями, контролюючи цю взаємодію і керуючи нею. Специфікація IEEE 802 поділяє цей рівень на 2 підрівня: MAC регулює доступ до поділюваного фізичного середовища, LLC забезпечує обслуговування мережного рівня. На цьому рівні працюють комутатори, мости й мережеві адаптери [11].

Прикладами стандартів та протоколів каналу передачі даних є Ethernet, двоточковий протокол PPP, HDLC і ADCCP. У пакеті інтернет-протоколів TCP/IP функціональність каналного рівня міститься в каналному рівні, найнижчому рівні описової моделі, який вважається незалежним від фізичної інфраструктури.

Ethernet – сімейство дротових комп'ютерних мережевих технологій, які зазвичай використовуються в локальних обчислювальних мережах (LAN),

міських мережах (MAN) і глобальних мережах (WAN) [12]. Стандарт Ethernet було комерційно представлено в 1980 році і вперше стандартизовано в 1983 році, відоме під назвою IEEE 802.3. Відтоді Ethernet було удосконалено щодо підтримки більш високих швидкостей передачі даних, більшої кількості вузлів і більш довгих відстаней з'єднання, проте збережено велику зворотну сумісність. Із часом Ethernet значною мірою замінив конкуруючі технології дротової локальної мережі, такі як Token Ring, FDDI і ARCNET. Станом на 2016 рік приблизно 85% усіх комп'ютерів у світі були підключені до комп'ютерних мереж за технологіями Ethernet. Часто використовуваним протоколом для підключення за стандартом Ethernet є різновид протоколу PPP – PPPoE.

## **1.2 Point-to-Point Protocol over Ethernet мережевий протокол передачі кадрів через Ethernet**

Наприкінці 1980-х років мережа Інтернет почала відчувати різке зростання кількості ГОМ, що використовують TCP/IP. Переважна частина цих ГОМ була приєднана до LAN різних типів, причому найбільш популярними були технології Ethernet. Велика частина інших ГОМ з'єднувалися через WAN, такі як загальнодоступні мережі передачі даних PDN типу X.25. Порівняно невелике число ГОМ були підключені до каналів зв'язку двоточною сполукою. Проте, канали зв'язку з двоточковим з'єднанням належать до числа найстаріших методів передачі інформації і майже кожна ГОМ підтримувала двоточкове з'єднання.

Однією з причин малого числа каналів зв'язку IP з безпосереднім з'єднанням був брак стандартного протоколу формування пакету даних. Протокол PPP мав вирішити цю проблему. Крім вирішення зазначеної проблеми в каналах із безпосереднім з'єднанням, PPP також мав вирішити інші проблеми, у тому числі: присвоєння і керування адресами IP; асинхронне (старт/стоп) і синхронне біт-орієнтоване формування пакету даних; мультиплексування протоколу мережі; конфігурація та перевірка якості каналу зв'язку; виявлення помилок; погодження адресу мережевого рівня та погодження компресії інформації. PPP вирішив ці



питання через забезпечення розширюваного протоколу керування каналом LCP і сімейства протоколів керування мережею NCP, які дали змогу погоджувати параметри конфігурації і різні можливості. Нині PPP не втратив актуальності та широко використовується операторами електронних комунікацій.

Згідно з Вікіпедією в комп'ютерних мережах протокол PPP є протоколом передачі даних канального рівня між двома маршрутизаторами безпосередньо без будь-якого хоста або іншої мережі між ними. Він може забезпечити аутентифікацію за циклічним з'єднання, шифрування передачі і стиснення даних [13].

PPP використовується в багатьох типах фізичних мереж, включно з послідовним кабелем, телефонним лініями, магістральними лініями, стільниковим телефонним зв'язком, спеціалізованими радіолініями, волоконно-оптичними лініями тощо. Оскільки IP-пакети не можуть передаватися через модемні лінії самостійно без будь-якого протоколу каналу передачі даних, який може визначити, де починається і де закінчується переданий кадр, оператори електронних комунікацій використовують PPP для віддаленого доступу клієнтів до Інтернету.

Зокрема, часто використовуваними є такі підвиди протоколу PPP:

- PPPoE – використовується для підключення за технологіями Ethernet, а іноді й DSL;
- PPPoA – використовується для підключення за технологіями ATM Adaptation Layer 5, які є основною альтернативою PPPoE для DSL.

Протокол PPPoE – це мережевий протокол для інкапсуляції кадрів протоколу PPP всередині кадрів Ethernet [15]. Він з'явився в 1999 році в контексті технологій DSL, як рішення для тунелювання пакетів через DSL-з'єднання в IP-мережу оператора, а звідти і в іншу частину Інтернету. Типове використання PPPoE включає в себе використання засобів PPP для автентифікації користувача з допомогою імені користувача та пароля, переважно через протокол PAP і рідше через CHAP. Приблизно у 2000 році PPPoE також почав замінювати метод зв'язку

з модемом, підключеним до комп'ютера або маршрутизатора LAN Ethernet, витісняючи більш старий метод, яким був USB.

PPPoE розроблено UUNET, Redback Networks і RouterWare та описано в RFC 2516 [16]. У стандарті, зокрема, зазначено та обґрунтовано функціональні можливості цього протоколу. Автори стандарту RFC 2516 звертають увагу на те, що вимоги до сучасних технологій доступу деякою мірою суперечливі. Бажано підключити безліч хостів віддаленого сайту через один пристрій доступу, що розташовується на цьому сайті. Бажано також забезпечити контроль доступу і функціональність, подібні наданням послуг через комутовані канали на базі PPP. Здебільшого економічно найефективнішим є метод підключення багатьох хостів до одного пристрою доступу на стороні абонента через мережу Ethernet. Вельми бажано також забезпечити невисоку ціну для пристрою доступу, укупі з простотою налаштування або можливістю обійтися взагалі без налаштування конфігурації цього пристрою.

PPPoE використовується для організації підключення CPN через один простий пристрій доступу (маршрутизатор) до віддаленого NAS (рис. 1.2). У цій моделі кожен хост використовує свій стек PPP і користувач представляється зі звичайним інтерфейсом. Контроль доступу, облік роботи (billing) і тип обслуговування можуть здійснюватися для окремих користувачів, а не для сайту загалом.

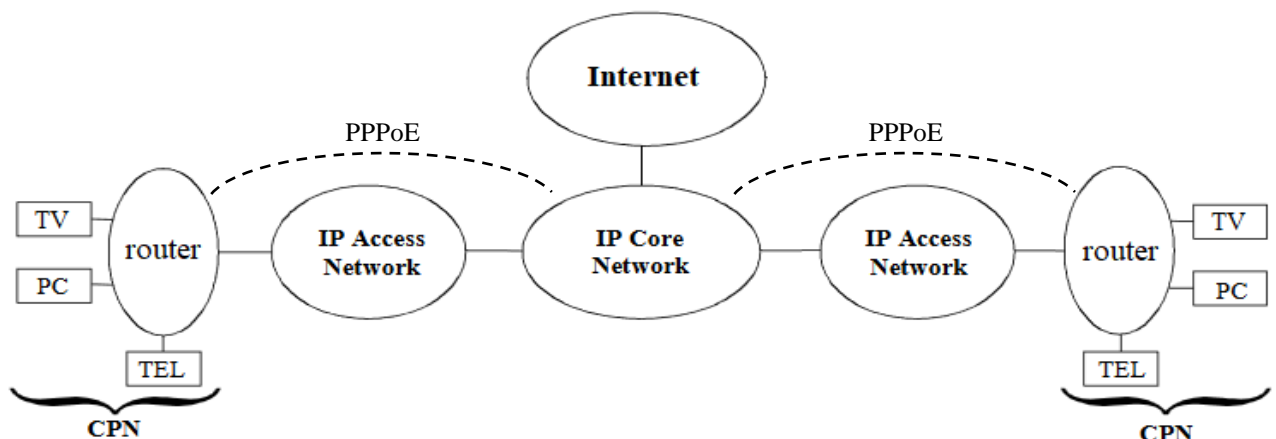


Рисунок 1.2 – Схема підключення користувача до мережі Інтернет

Для забезпечення двоточкового з'єднання через мережу Ethernet кожна сесія PPP має дізнатися Ethernet-адрес віддаленого партнера, а також створити унікальний ідентифікатор сесії. PPPoE також включає протокол виявлення (Discovery) для вирішення цих питань. Отже, робота PPPoE включає два різних етапи – виявлення (Discovery) і сеанс PPP. Хост, який бажає організувати сеанс PPPoE, має виконати етап виявлення для визначення MAC-адресу партнера та створення ідентифікатора сеансу PPPoE SESSION\_ID. Хоча PPP визначає рівноправну взаємодію (peer-to-peer relationship), етап виявлення використовує відносини «клієнт-сервер». У процесі виявлення хост (клієнт) відшукує концентратор доступу (сервер). Залежно від топології мережі в ній може бути присутнім більше ніж один концентратор доступу, з яким може взаємодіяти хост. Етап виявлення дає змогу хосту виявити концентратори доступу та вибрати один із них. Після успішного завершення етапу виявлення хост і обраний ним концентратор доступу мають інформацію, яка потрібна для організації двоточкового з'єднання через мережу Ethernet.

Етап виявлення не задає будь-якого стану (stateless), поки не буде організовано сеанс PPP. Після організації сеансу PPP хост і концентратор доступу мають виділити ресурси для віртуального інтерфейсу PPP.

Більш детально про формат пакетів; формат кадрів Ethernet; етапи процесу виявлення (Discovery) та пакети цього процесу, такі як PADI, PADO, PADR, PADS, PADT; етап організації сеансу PPP Session; керування каналом LCP; захист від DoS-атак та інше описано в стандарті RFC 2516 [16].

### **1.3 RADIUS протокол автентифікації, авторизації та збору даних про використовувані ресурси мережі**

RADIUS (служба віддаленої автентифікації користувачів, що додзвонюються) – мережевий протокол, призначений для забезпечення централізованої автентифікації, авторизації і обліку (AAA) користувачів, які

підключаються до різних мережевих служб [17]. Цей протокол описано в стандартах RFC 2865 і RFC 2866.

Концепція RADIUS networking народилася на початку 90-х років. Некомерційна організація Merit Network, що надає якісні мережеві послуги освітнім, державним медичним установам, запросила рішення, яке скоротило б їхні системи автентифікації, авторизації і обліку. У відповідь компанія Livingston Enterprises розробила першу версію служби віддаленої автентифікації для користувачів [18]. Спочатку RADIUS був тільки протоколом AAA для користувачів із комутованим доступом. Пізніше, з розвитком методів доступу користувачів, RADIUS було адаптовано до інших методів доступу, таких як доступ через Ethernet. Він надає доступ за автентифікації та авторизації, а також збирає і записує дані використання мережевих ресурсів користувачами для стягування плати. Отже, RADIUS набув широке застосування в різних галузях промисловості. Зокрема, цей протокол застосовується операторами для системи тарифікації використаних ресурсів конкретним користувачем/абонентом.

RADIUS використовує два типи пакетів для управління повним процесом AAA [17]: запит доступу, який управляє автентифікацією і авторизацією; запит обліку, який управляє обліком. Автентифікація і авторизація визначено в RFC 2865, у той час як облік описано в RFC 2866.

Незважаючи на те, що є безліч способів побудови мереж із використанням RADIUS, загальна структура може бути представлена у вигляді, представленому на рис. 1.3.



Рисунок 1.3 – Структура мережі з RADIUS сервером

Місце NAS може займати VPN-сервер, PPPoE-сервер, RAS (Remote Access Server), мережевий комутатор тощо. RADIUS-сервер може бути частиною виключно локальної мережі або ж мати доступ до мережі Інтернет. Бази автентифікації зберігають інформацію про користувачів (абонентів) та права їхнього доступу до різних сервісів. Термін «база» в цьому випадку є збірним, оскільки дані можуть зберігатися як локально, у текстових файлах і різного роду БД, так і на віддалених серверах (SQL, Kerberos, LDAP, Active Directory тощо).

Механізми автентифікації і авторизації, як уже зазначалося вище, описані в RFC 2058. Вони протікають за такою схемою (рис. 1.4): RADIUS-клієнт відправляє RADIUS-серверу запит доступу (Access Request), у якому містяться дані автентифікації (логін і пароль, сертифікат доступу). Після порівняння отриманих даних із наявними в базі, сервер може відправити одну з декількох відповідей [19, С.10-15]:

- Access-Accept – доступ отримано, можна починати використання ресурсів. Пакет, що несе цю відповідь, також може містити додаткову інформацію: IP, що видано користувачеві, допустиму тривалість сесії, максимальний обсяг переданого трафіку тощо;
- Access-Challenge – потрібне введення додаткових даних (PIN, додаткового пароля). Використання цієї відповіді дає змогу проводити процедури складної автентифікації в рамках захищеного мережевого тунелю, встановленого безпосередньо між користувачем і серверними машинами (для уникнення «осідання» даних на сервері доступу);
- Access-Reject – доступ заборонено через невірно вказані дані або брак користувача в базі.

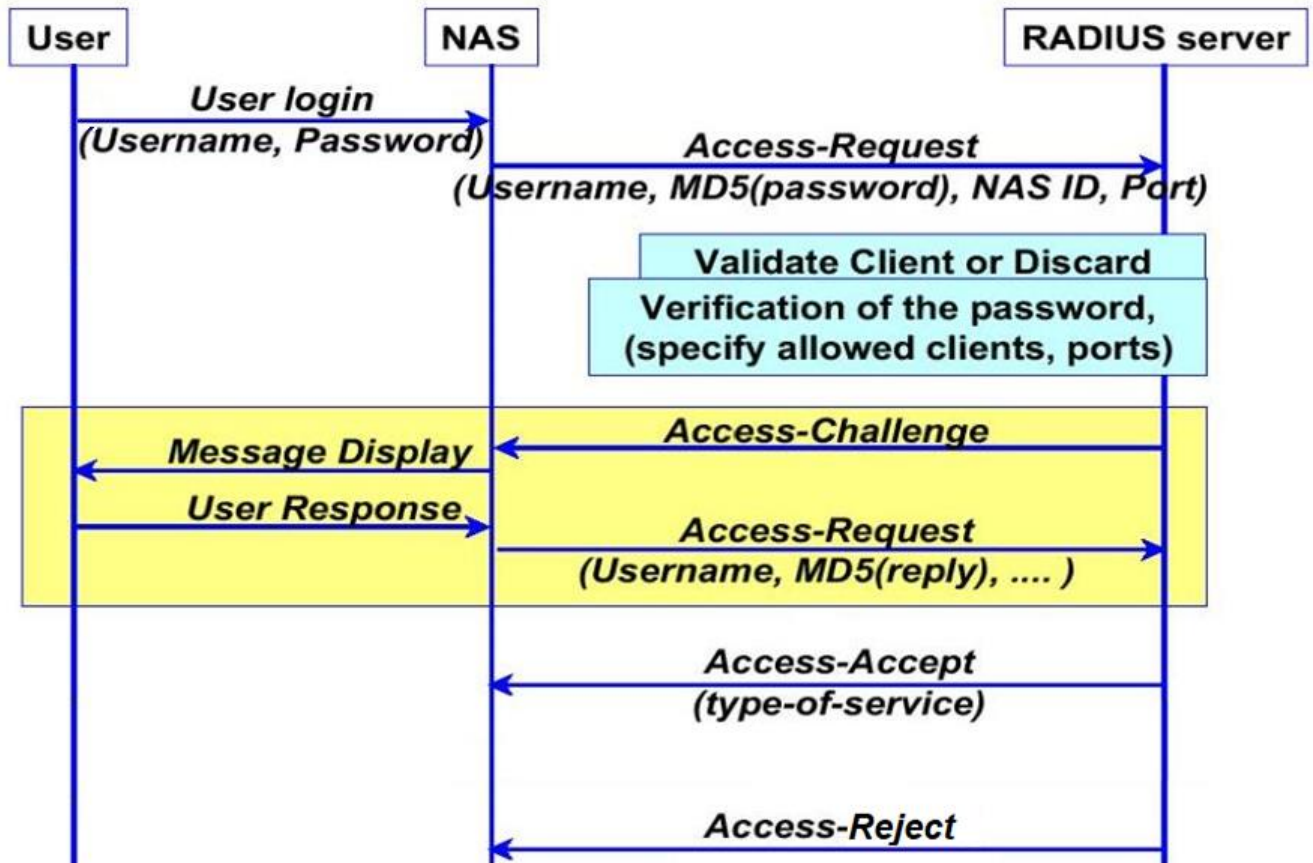


Рисунок 1.4 – Механізми автентифікації і авторизації

Правильність даних автентифікації може перевірятися за різними схемами автентифікації: PAP, EAP або CHAP. Процедuru обміну в рамках операцій обліку описано в стандарті RFC 2086. У разі підключення клієнт посилає серверу запит обслуговування (Accounting-Request), що містить параметр `acct_type=start`. Це є сигналом до початку надання послуг обліку й контролю. У відповідь на цей запит клієнту надсилається його унікальний ідентифікатор у мережі, ідентифікатор сесії і мережевий адрес.

Під час роботи клієнт періодично відправляє на сервер запит зі значенням `acct_type=interim_update`, що є сигналом до того, що клієнт усе ще використовує ресурс і операції обліку необхідно продовжувати. Цей запит зазвичай містить поточну тривалість сесії і обсяг переданих даних. У разі закінчення роботи в мережі, після відключення клієнта від NAS, на сервер надсилається запит із параметром `acct_type=stop`, що означає припинення роботи й закінчення надання

послуг обліку. Цей пакет запиту може містити в собі такі дані, як тривалість сесії, кількість переданих пакетів і обсяг пересланих даних. У відповідь на кожен запит подібного роду сервер надсилає підтвердження (Accounting-Response), що гарантує подальший доступ клієнта до ресурсу. Якщо запит не отримує підтвердження з першого разу, через деякий час буде зроблена спроба повторного запиту – і так аж до отримання відгуку або прийняття рішення про недоступність сервера.

Треба зазначити, що RADIUS забезпечує певну захищеність каналів обміну інформацією. Паролі від NAS до RADIUS-сервера не пересилаються у відкритому вигляді. Для шифрування паролів використовується принцип «поділу секрету» й хеш-функції MD5. Однак, через часткову реалізацію цих функцій, на практиці необхідно використання додаткових заходів, таких як застосування IPsec або фізичного захисту корпоративних мереж. Це дає змогу в подальшому захистити трафік між сервером доступу і RADIUS-сервером. Крім того, дані, що передаються, піддаються захисту лише частково: захищені логін і пароль, у той час, як інші дані, можливо, є секретними або приватними, не захищені.

З більш детальною інформацією про RADIUS можна ознайомитися в стандартах RFC 2865 Remote Authentication Dial In User Service (RADIUS) та RFC 2866 RADIUS Accounting. Також RADIUS має стосунок до таких стандартів [17]: RFC 2548 Microsoft Vendor-specific RADIUS Attributes, RFC 2607 Proxy Chaining and Policy Implementation in Roaming, RFC 2618 RADIUS Authentication Client MIB, RFC 2619 RADIUS Authentication Server MIB, RFC 2620 RADIUS Accounting Client MIB, RFC 2621 RADIUS Accounting Server MIB, RFC 2809 Implementation of L2TP Compulsory Tunneling via RADIUS, RFC 2867 RADIUS Accounting Modifications for Tunnel Protocol Support, RFC 2868 RADIUS Attributes for Tunnel Protocol Support, RFC 2869 RADIUS Extensions, RFC 2882 Network Access Servers Requirements: Extended RADIUS Practices, RFC 3162 RADIUS and IPv6, RFC 3575 IANA Considerations for RADIUS, RFC 3576 Dynamic Authorization Extensions to RADIUS тощо.

## **1.4 Огляд та аналіз аналогів**

### **1.4.1 Автоматизована система розрахунків (АСР)**

Для тарифікації послуг, що надаються оператором електронних комунікацій абонентам, необхідно мати спеціалізовану АСР (Billing).

АСР (білінг, billing) – у деяких видах бізнесу, зокрема, в телекомунікаціях – автоматизована система обліку наданих послуг, їхньої тарифікації і виставляння рахунків для оплати. У сфері телекомунікацій білінг офіційно іменується «Автоматизована Система Розрахунків» [20].

Також білінгом називають системи обліку й оплати інтернет трафіку та супутніх послуг (сервісів). Такі системи часто суміщені з технічними засобами, що автоматично обмежують доступ до послуг у разі нульового балансу рахунку і дають можливість надати деталізацію витрачених коштів.

Функції АСР сфери телекомунікацій групуються у два основні блоки: розрахункові операції та інформаційне обслуговування [21]. Розрахункові операції поділяють на два різновиди: передоплата (Prepaid) та післяплата (Postpaid).

У блоці розрахункових операцій виділяються такі функції, як визначення споживання (наприклад, обробка даних із комутаційного обладнання про споживання трафіку, отримання даних із системи медіації), оцінка споживання (визначення розрахункових характеристик даних про споживання), агрегація оцінок і формування нарахувань абонентам, розрахунок податків, знижок, додаткових нарахувань, коригувань, випуск рахунків для оплати, забезпечення доставки або ознайомлення абонентів із рахунками для оплати, управління особовими рахунками абонентів. Реалізація розрахункового блоку може суттєво відрізнитися як для різного типу комунікацій, так і в різних моделях взаємини з абонентами.

Передоплата (Prepaid). Це модель розрахунків з абонентами й агентами, що передбачає попереднє внесення коштів на свій особистий рахунок оператора послуг зв'язку, які згодом витрачаються на оплату наданих послуг. Prepaid-



системи розрахунків зазвичай ведуть облік у реальному часі безпосередньо керуючи процесом надання послуг. За умови досягнення нижнього порога кількості коштів на рахунку контрагента режим надання послуг може бути змінено або повністю припинено. Кількість коштів на рахунку, тривалість збереження додатного балансу, розмір і регулярність надходжень можуть служити параметром тарифікації в процесі розрахунку вартості, якості та кількості послуг, що надаються. Функціональні можливості з підтримки обмеження доступу до послуг зв'язку в реальному часі з передплатною моделлю розрахунків іноді називають гарячим білінгом.

Післяплата (Postpaid). Це модель розрахунків, за якою оператор спочатку надає послуги абоненту в рамках укладеного з ним договору, а потім проводить тарифікацію та виставляння рахунків для оплати. Процес тарифікації і надання рахунків є планово-регулярним і зазвичай охоплює обумовлений у договорі календарний проміжок часу. Контрагент зобов'язаний сплатити суму виставленого рахунку протягом обумовленого в контракті проміжку часу, у разі несвоєчасної оплати до нього застосовуються зазначені в договорі методи роботи з боржником, у рамках процесів стягнення дебіторської заборгованості.

Інформаційне обслуговування включає функції підтримки операційної інформації про абонентів, про продукти й послуги, включно з їхніми тарифами, обмеженнями, можливими комбінаціями, а також конфігураційних даних про АСР загалом.

#### **1.4.2 Наявні АСР**

На ринку телекомунікацій є пропозиції щодо готових АСР інтернет-абонентів оператора телекомунікацій. Найбільш цікавими з них є: ABills (Advanced Billing Solutions), NoDeny, Stargazer, ETBill.

##### **АСР ABills (Advanced Billing Solutions)**

Офіційний сайт: [abills.net.ua](http://abills.net.ua)

На офіційному сайті зазначено, що ABills (Advanced Billing Solutions) – надійна конвергентна білінгова система, призначена для обліку та тарифікації

всього спектру послуг, що надаються операторами зв'язку (VPN, Hotspot, IPTV тощо). Система є багатофункціональною, модульною АСР з відкритим програмним кодом. Можливості системи є вельми широкими.

Система є кроссплатформною і може функціонувати на будь-якій Unix-подібній ОС, а також може бути портована на MS Windows. Для управління системою використовується вебінтерфейс. Для менеджерів відділу продажів передбачено MS Windows інтерфейс (ABM). Для користувачів передбачено вебінтерфейс, а також MS Windows кабінет користувача.

Для зберігання даних система використовує СКБД MySQL. Для високопродуктивних рішень є можливість побудови кластера. Система може обслуговувати з'єднання: VPN, PPPoE, а також з'єднання за протоколом IPoE (DHCP).

Послуги, що надаються АСР ABills: робота за передплатою або післяплатою, Dialup/VPN (PPPoE, PPTP, L2TP, IPSec, OpenVPN), Radius, NetFlow, Hotspot, Squid Monitoring, VoIP (GNU gatekeeper, Asterisk), CMTS (BSR1000 monitoring), Mail management (Postfix, Spamassassin, Courier IMAP/POP), управління комутаторами з допомогою SNMP (D-Link, Cisco, Zyxel, Patton, Mikrotik), Cisco ISG (Services, Turbo Mode), IPTV (VoD, nVoD), Antivirus (Dr. Web, AV Desk), маркетингові звіти, платіжні системи (Webmoney, Pegas, 24 Non Stop, SMS Proху тощо), управління DHCP, система масової розсилки повідомлень абонентам, мультидоменна структура дає можливість вести кілька незалежних провайдерів в одному білінгу, карткова платформа з можливістю роботи з дилерами.

ABills підтримує широкий спектр тарифікації (за часом, за вхідним або вихідним трафіком, тарифи з передплаченим трафіком тощо), лімітів (денний, тижневий або місячний трафік, час, гроші тощо).

ABills має систему керування швидкістю (шейпер), систему моніторингу. Також ABills підтримує різні типи й методи авторизації (PAP, CHAP, MsChar,

MsChar V2 шифрування трафіку; IEEE 802.1x; користувач + пароль; користувач + пароль + IP; користувач + пароль + MAC тощо).

Окрім цього ABills має модулі фінансового обліку та особистий кабінет користувача, що надає багато функціональних можливостей абоненту, зокрема, це статистика за наданими послугами; статистика платежів і відрахувань; поповнення пластиковими картками Visa/Master Cards або іншими способами онлайн оплати (Webmoney); форма технічної підтримки; виписка бухгалтерських документів; зміна тарифного плану, кредиту, пароля, персональної інформації; управління e-mail ящиком.

На офіційному сайті є документація ([abills.net.ua:8090](http://abills.net.ua:8090)). Окрім цього за додаткову плату можна отримати розширений технічний супровід системи. Представлені три пакети: Abills Base, Abills Standart, Abills Extra. Ціна залежить від кількості абонентів оператора, кількості білінгових серверів, кількості функціоналу АСР.

До недоліків цієї системи можна віднести дуже складну систему розгортання АСР. Широкий функціонал можливостей, з одного боку, є перевагою, проте, функціонал який не буде використовуватися перевантажує інтерфейс оператора й робить його не дуже зручним та наочним. Також треба відмітити брак такого важливого функціоналу щодо адміністрування мережі оператора та швидкого пошуку локалізації проблеми, як моніторинг усіх активних сесій абонентів.

### **АСР NoDeny**

Офіційний сайт: [nodeny.com.ua](http://nodeny.com.ua)

Система працює під керуванням ОС FreeBSD.

На офіційному сайті зазначено, що АСР NoDeny виконує всі типові дії, які потрібні оператору: облік даних абонентів із будь-яким набором полів, ведення рахунків, підключення тарифів, автоматичне блокування доступу, управління швидкістю на різних типах обладнання, підрахунок трафіку, побудова

різноманітних графіків та мапи з вузлами та комунікаціями, картки оплати, платіжні системи, особистий кабінет тощо. Є українська локалізація.

Умовно NoDeny можна розділити на вебінтерфейс і серверну частину. Вебінтерфейс доступний, абонентам, адміністраторам, працівникам компанії. Клієнтський і адміністративний інтерфейс має одну точку входу. Адміністратор має можливість зайти в особистий кабінет абонента від його імені, не змінюючи поточну сесію. Це корисно, коли адміністратор хоче точно знати, що бачить абонент і чому в нього щось не виходить. До того ж адміністратор може виконати деякі дії від імені абонента, наприклад, активувати картку поповнення рахунку. У будь-якому випадку, у логах і базі даних реєструється подія, що дію виконав саме адміністратор.

Дані облікового запису абонента можуть мати будь-який набір полів, можна створювати поля залежно від потреб оператора. У будь-який момент часу можна додати/видалити поле й це відобразиться для всіх уже наявних абонентів.

Послуги надаються абоненту за передоплатою. Приклад послуги – тариф, який дає доступ в Інтернет на певній швидкості на заданий період. Параметр послуги «автоматичне продовження» вказує яка послуга буде автоматично підключена після завершення поточної. Автоматичне продовження може бути відключене для конкретної послуги й тоді після її завершення не буде підключена інша. Можна встановити тривалість послуги, збільшити швидкість Інтернету на певний проміжок часу. Блокування за балансом виконує ядро NoDeny.

До облікового запису можна підключити один або кілька IP-адрес. Також допускається динамічна видача IP в момент авторизації. Можна комбінувати, тобто одним абонентам видати статичні адреси, інші будуть отримувати динамічні. У будь-якому разі, попередньо список IP-адрес має бути створено.

Система NoDeny підтримує різні види авторизації абонентів у мережі: PPPoE, DHCP з прив'язкою за мак-адресом.

Окрім цього ACP NoDeny надає можливість збору статистики за трафіком: загальної і деталізованої. Остання – це повна інформація

(ір/порти/протоколи/байти) за всіма сесіями. Трафік можна класифікувати за «напрямами», наприклад, напрямом UA-IX, пірінг із сусідньою мережею, Інтернет тощо та будувати графіки за цими напрямками для різних абонентів або всієї мережі на будь-який день.

NoDeny підтримує різні варіанти підключення абонентів навіть у межах одного протоколу. Наприклад, DHCP на Mikrotik за Radius, DHCP на Mikrotik з управлінням по SSH, DHCP на Linux/FreeBSD з реєстрацією через веб тощо.

На офіційному сайті є документація ([nodeny.com.ua/wiki/index.php/Содержание](http://nodeny.com.ua/wiki/index.php/Содержание)). Проте, дані про вартість системи, можливості технічної підтримки відсутні.

До недоліків цієї системи можна віднести як і в попередній АСР брак моніторингу всіх активних сесій клієнтів. А також брак схеми або опису технічного супроводу, що є вельми важливою складовою забезпечення працездатності системи в майбутньому в разі довгострокової експлуатації.

### **АСР Stargazer**

Офіційний сайт: [stg.net.ua](http://stg.net.ua)

На офіційному сайті зазначено, що АСР Stargazer призначена для авторизації і обліку трафіку в локальних мережах. У процесі написання цієї системи перед розробниками було поставлено завдання створення продукту, який відповідав би вимогам більшості локальних мереж для обліку трафіку й коштів абонентів, а також їхньої безпечної авторизації.

Система побудована за клієнт-серверною технологією. У якості сервера може бути машина з ОС Linux або FreeBSD з архітектурою x86/x86\_64/SPARC. У якості клієнтів можуть використовуватися машини сімейства ОС Windows, так і клієнти з ОС Linux або FreeBSD. Так само клієнтом може виступати будь-яка ОС, у якої є підтримка мережесих протоколів TCP/IP і веббраузер.

Система має модульну структуру, що дає змогу змінювати її конфігурацію і функціональність. Основні можливості системи:

- контроль над клієнтами мережі, їхнє додавання, видалення, поточне коригування;
- авторизація клієнта з дозволом або заборонаю доступу в Інтернет;
- підрахунок трафіку за попередньо заданими напрямками і правилами;
- підрахунок витрачених клієнтом коштів і автоматичне вимкнення в разі їхнього повного витрачання;
- зберігання додаткової інформації про клієнта, такої як адрес місця проживання, телефон тощо;
- автоматичний пінг усіх клієнтів мережі та вивід результатів на екран;
- ведення історії для всіх клієнтів;
- журналювання дій адміністраторів системи;
- оперативне надання клієнту інформації про його трафік і наявність коштів на особистому рахунку;
- формування докладних звітів про стан клієнтів за їхніми обліковими записами;
- модулі.

На офіційному сайті є документація ([stg.net.ua/doc/index.html](http://stg.net.ua/doc/index.html)). Проте, дані про технології AAA, що підтримуються системою Stargazer; про вартість системи; можливості технічної підтримки відсутні.

До недоліків системи Stargazer можна віднести, по-перше, недостатній функціонал для надання повного спектру послуг оператором. Наприклад, вимога з боку розробників використання спеціального авторизатора для отримання доступу в Інтернет і брак можливості використання Wi-Fi маршрутизаторів для отримання такого доступу, робить систему досить вендорозалежною. Пропонований моніторинг активних сесій є неефективним.

Як і в АСР NoDeny у Stargazer відсутня схема або опис технічного супроводу. Цей недолік ускладнюється ще і браком повної технічної документації за системою.

## **АСР за телекомунікаційні послуги ЕТBill**

Офіційний сайт: [enrantelecom.com/ru/projects/billing/](http://enrantelecom.com/ru/projects/billing/)

АСР за телекомунікаційні послуги ЕТBill розроблено ТОВ науково-виробничою фірмою «Енран Телеком». Як вказує розробник система є повнофункціональною розрахунковою системою оператора телекомунікацій, що надає оператору облікові функції та необхідний для цього функціонал, а також функції обслуговування населення щодо прийому оплати за надані послуги і продаж супутніх товарів.

На офіційному сайті зазначено, що АСР ЕТBill має сертифікат відповідності від 11 травня 2011 року, виданий органом сертифікації. За своїми можливостями і функціональними особливостям АСР відповідає вимогам ГСТУ 45.028-2004 – «Автоматизовані системи розрахунків за телекомунікаційні послуги. Класифікація. Загальні технічні вимоги». Система має трьох рівневу архітектуру з використанням сервера застосунків – розробки компанії, а також можливостей сервера БД Oracle (версія 11g) і кластера Oracle RAC.

На офіційному сайті відсутні документація; дані про технології AAA, що підтримуються системою ЕТBill; дані про вартість системи; можливості технічної підтримки. Розробник системи ЕТBill надав дуже стислу інформацію щодо свого продукту, що не дає можливості зробити певні висновки щодо ефективності та обсягу функціоналу, технологій, тарифів за використання цієї системи. Система є закритою і це ускладнює прийняття рішення щодо доцільності її впровадження.

### **1.5 Специфікація вимог**

На основі аналізу предметної галузі, відповідних технологій та дослідження наявних АСР сформовані вимоги щодо програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій. Сформовано специфікацію вимог щодо програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій, що є описом функціональних вимог та функцій розроблюваної АСР.

## ПРИЗНАЧЕННЯ АСР

АСР має забезпечувати організацію контролю, доступу та обліку в Інтернет абонентів оператора телекомунікацій із використанням PPPoE-з'єднання та RADIUS протоколу.

### ФУНКЦІЇ СИСТЕМИ

*Функція системи* – Контроль

*Функціональні вимоги*

Контроль поділяється на пасивний та активний.

Пасивний контроль:

- моніторинг усіх активних сесій із використанням фільтрів;
- формування фінансової звітності;
- моніторинг розподілу IP-адресного діапазону;
- журналювання процесу авторизації абонентів;
- журналювання дій менеджерів;
- моніторинг проблемних з'єднань;
- моніторинг сесії абонента.

Активний контроль

- формування кабінету абонента;
- пошук абонентів у БД;
- примусове відключення активної сесії абонента.

*Функція системи* – Доступ

*Функціональні вимоги*

Авторизація абонента в системі за протоколом PPPoE згідно з діючими тарифними планами та налаштуваннями АСР.

*Функція системи* – Облік

Функції обліку можна умовно поділити на глобальні та абонентські правила.

Глобальні правила:

- формування тарифних планів;



- налаштування IP-адресних діапазонів;
- налаштування систем оплати.

Абонентські правила:

- формування карток абонентів (створення, редагування, видалення).

## ВИМОГИ ДО АСР

### *Мова й технологія розробки*

Систему АСР можна поділити на дві складові: backend та frontend. До backend належить рівень AAA, до frontend – вебінтерфейс.

Backend:

- FreeRADIUS – сервер із відкритим вихідним кодом, з відповідними модулями;
- Eclipse – відкрите інтегроване середовище розробки модульних кросплатформних застосунків;
- MySQL – реляційна СКБД із відкритим вихідним кодом;
- мова C та компілятор GCC, мова Python.

Frontend:

- Apache HTTP Server;
- мова PHP.

*Вимоги:*

Система АСР буде працювати під керуванням ОС Linux Ubuntu.

## ВЛАСТИВОСТІ АСР

### *Горизонтальна масштабованість*

У разі збільшення абонентської бази і зростання навантаження на NAS-сервери передбачена можливість підключення додаткових NAS-серверів.

### *Швидкість авторизації абонентів*

Час авторизації всіх абонентів оператора не мусить перевищувати 10 хв.

### *Стабільність роботи*

АСР залишається працездатною в разі масового підключення абонентів із від'ємним балансом на рахунку.

### *Розширюваність*

Можливість розширення функціоналу АСР завдяки використанню додаткових модулів.

### *Зручність використання*

Функціонал АСР має бути доступний через вебінтерфейс.

## **Висновки до розділу 1**

Аналіз предметної галузі дав змогу зробити такі висновки.

Мережа Інтернет перебуває в стані активного розвитку, про це свідчить зростання кількості інтернет-сервісів; збільшення кількості інформації, що передається, зокрема, на високих швидкостях (1 Гбіт/с, 10 Гбіт/с і вище) через оптоволоконні мережі; стрімке збільшення інтернет-хостів у всьому світі за порівняно короткий період. Отже, майже кожна людина у світі, зокрема, і в Україні прагне бути підключеною до світової Мережі та отримати доступ до її сервісів.

В Україні діяльність у сфері телекомунікацій регламентується Законом України «Про електронні комунікації» та низкою нормативних актів. Організацією доступу абонентів до мережі Інтернет займаються оператори електронних телекомунікацій (оператори). Зокрема, до їхніх обов'язків входить організація доступу абонентів до мережі Інтернет. Отже, питання, що постає перед операторами: «Як організувати доступ користувачів до мережі Інтернет?» й досі є актуальним.

Аналіз технологій організації доступу користувачів до мережі Інтернет дав можливість виокремити технології, що дають змогу організувати доступ абонентів в Інтернет у контексті сучасних вимог:

- Ethernet сімейство дротових комп'ютерних мережевих технологій, що описано в стандарті IEEE 802.3;
- PPPoE – мережевий протокол для інкапсуляції кадрів протоколу PPP всередині кадрів Ethernet, описано в стандарті RFC 2516;

- RADIUS – мережевий протокол, призначений для забезпечення централізованої автентифікації, авторизації і обліку (AAA) користувачів, які підключаються до різних мережних служб. Цей протокол описано в стандартах RFC 2865 і RFC 2866.

Для тарифікації послуг, що надаються оператором електронних комунікацій абонентам, необхідно мати спеціалізовану АСР (Billing). На ринку телекомунікацій є пропозиції щодо готових АСР інтернет-абонентів оператора телекомунікацій. Найбільш цікавими з них є: ABills (Advanced Billing Solutions), NoDeny, Stargazer, ETBill. Їхній детальний аналіз дав можливість виявити поряд із перевагами та вдалим рішеннями ще і слабкі сторони. Зокрема, усі розглядувані системи мали брак такого важливого функціоналу щодо адміністрування мережі оператора та швидкого пошуку локалізації проблеми, як моніторинг усіх активних сесій клієнтів. Окрім цього кожна система мала ще певні недоліки, які робили їхнє впровадження в компанії не дуже доцільним рішенням.

Отже, на основі аналізу предметної галузі, відповідних технологій та дослідження наявних АСР сформовані вимоги щодо програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій, що є описом функціональних вимог та функцій розроблюваної АСР.

## 2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Діаграма варіантів використання

Діаграма варіантів використання (Use Case Diagram) – демонструє набір варіантів використання і дійових осіб (які є спеціальним різновидом класів), а також їхній зв'язок [22, с. 40-41]. Діаграма варіантів використання описує функціональне призначення системи і є початковим концептуальним поданням або концептуальною моделлю системи в процесі її проєктування і розроблення.

Розроблення діаграми варіантів використання переслідує такі цілі:

- визначити загальні межі й контекст модельованої предметної галузі на початкових етапах проєктування системи;
- сформулювати загальні вимоги щодо функціональної поведінки системи;
- розробити початкову концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;
- підготувати початкову документацію для взаємодії розробників системи з її замовниками й користувачами.

В АСР оператора телекомунікацій можна виокремити такі групи користувачів: адміністратор, менеджер, абонент. Кожна група користувачів використовує систему за своїми потребами та наданим доступом. Загалом із системою можна виконувати такі дії:

- моніторинг (усіх активних сесій із використанням фільтрів, розподілу IP-адресного діапазону, процесу авторизації абонентів, дій менеджерів, проблемних з'єднань, сесії абонента);
- формування та перегляд фінансової звітності;
- примусове відключення активної сесії абонента;
- авторизація в системі;
- формування тарифних планів;
- налаштування IP-адресних діапазонів, систем оплати;
- формування карток абонентів (створення, редагування, видалення);

- перегляд особистої картки абонента;
- оплата послуги (оплата готівкою, картою передплати, банківській або довірчий платіж).

На рис. 2.1 наведено діаграму варіантів використання для розроблюваної системи АСР.

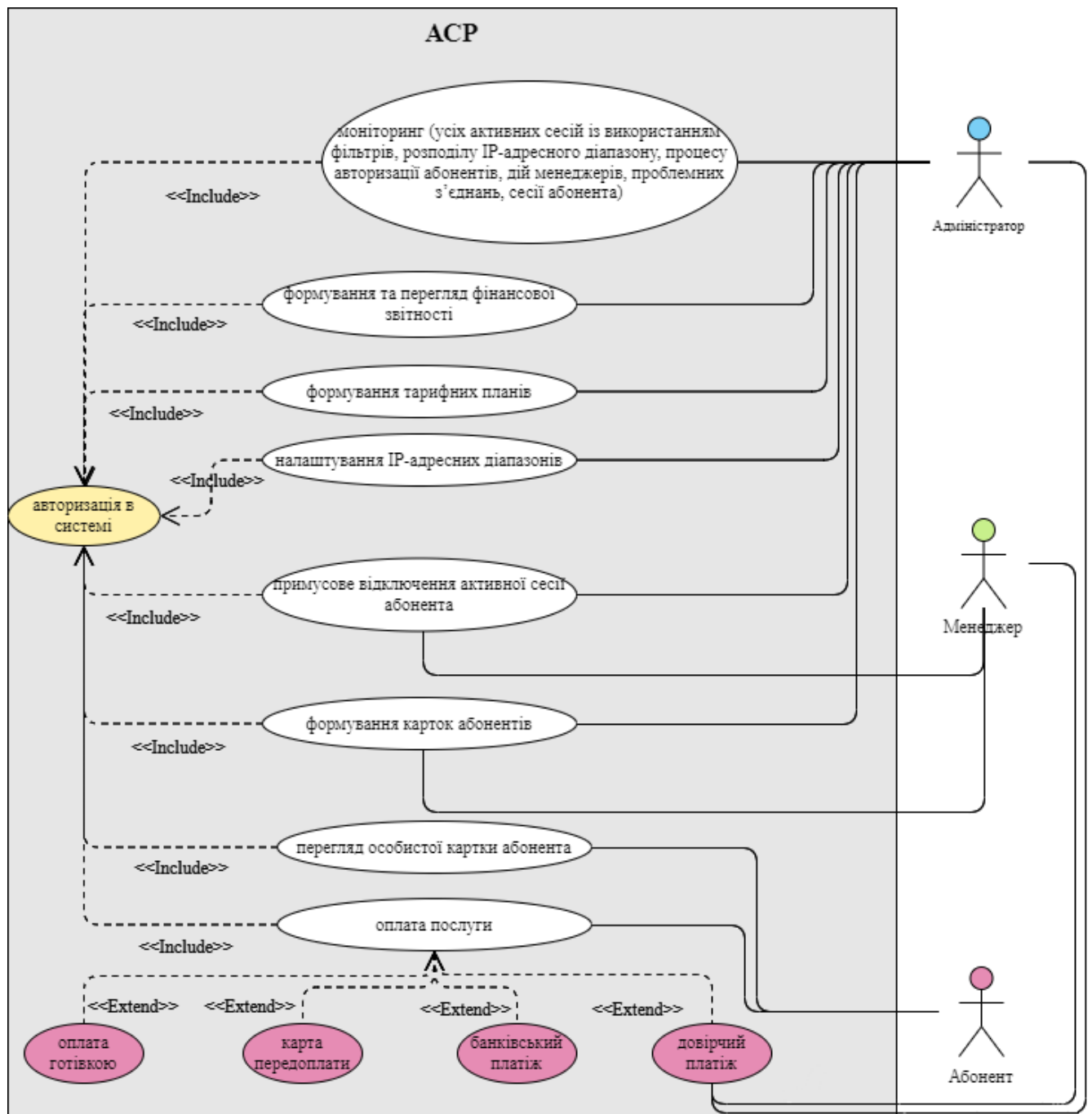


Рисунок 2.1 – Діаграма варіантів використання розроблюваної системи АСР

Адміністратор системи може виконувати всі зазначені дії крім оплати готівкою, картою передплати та банківським платежем. Ці дії виконує абонент із

допомогою своєї готівки або своєї банківської карти. Менеджер може виконувати тільки частину дій адміністратор, а саме формування карток абонентів та примусове відключення активної сесії абонента. Тобто в разі підключення до мережі оператора нового абонента менеджер створює запис у БД із відповідними даними про абонента та в процесі роботи може змінювати за зверненням абонента певні значення (тарифний план, особисті дані). Також менеджеру доступна функція примусового відключення активної сесії абонента, що необхідно для примусового перепідключення обладнання абонента для активації певних змін у налаштуваннях картки абонента.

Абонент має функцію перегляду своєї особистої картки абонента, тобто кабінету користувача. У якому він може дивитися статистику його сесій, платежів, а також виконати оплату послуги всіма зазначеними способами.

## 2.2 Діаграма діяльності

Діаграми діяльності (Activity Diagrams) – це графічні представлення робочих процесів покрокових дій і дій загалом [23]. Діаграма діяльності показує структуру процесу або інших обчислень як покроковий потік управління і даних. Діаграми діяльності описують динамічне представлення системи [22, с. 42]. Вони особливо важливі в процесі моделювання функцій системи і виділяють потік управління між об'єктами.

На рис. 2.2 наведено діаграму взаємодії клієнта із системою АСР. Процес загалом можна розділити на три складові: авторизація, автентифікація та облік. Авторизація – це валідація тільки логіну та паролю. Цей процес використовує мінімальний обсяг обчислювальних ресурсів. Далі виконується автентифікація за результатами якої створюється РРРoЕ-з'єднання згідно з налаштуваннями АСР (швидкість доступу, обмеження доступу тощо). Після цього з'єднання підтримується та керується функцією обліку.

Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

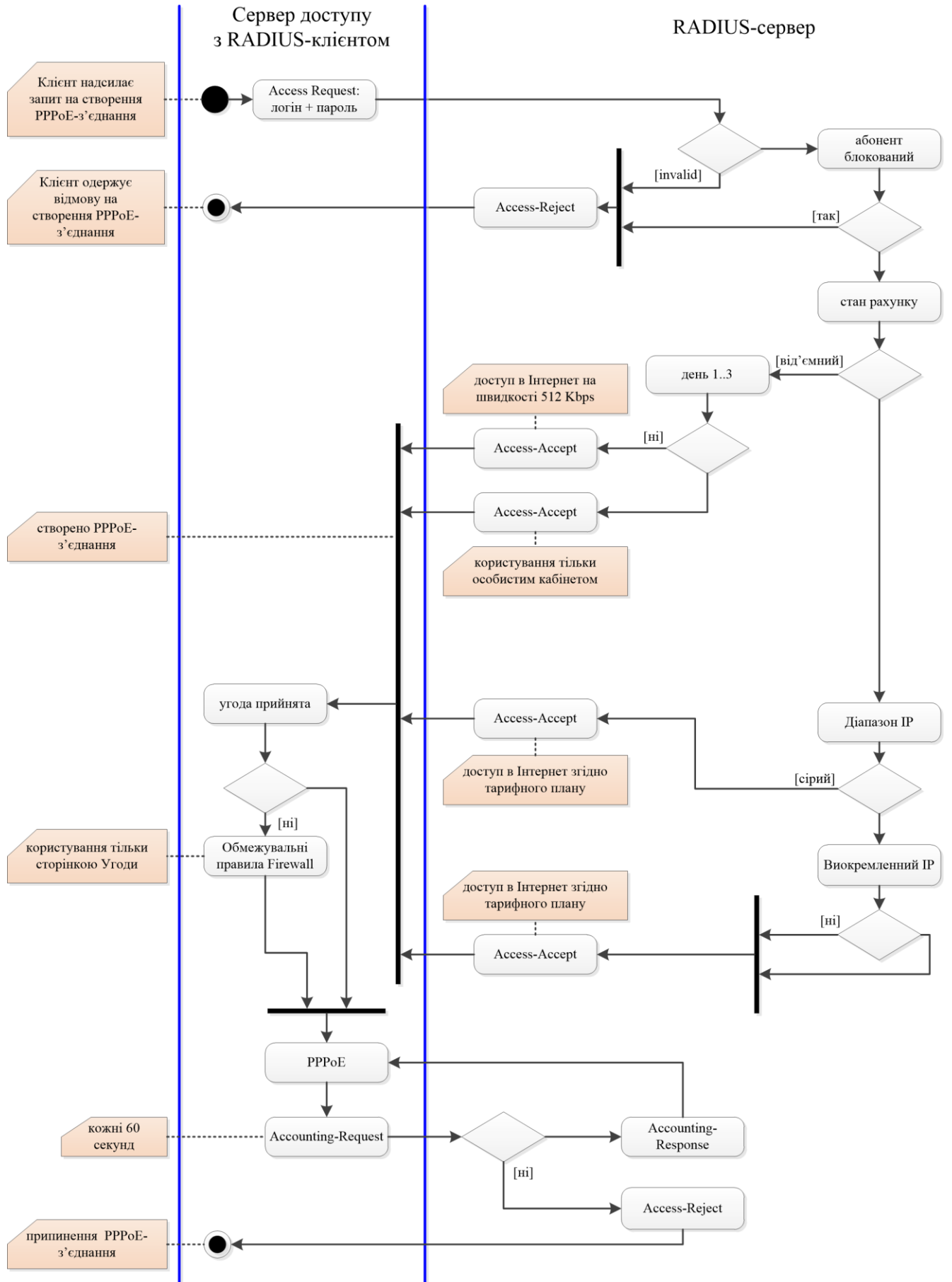


Рисунок 2.2 – Діаграма взаємодії клієнта із системою ACS

Для отримання доступу в Інтернет абоненту необхідно налаштувати своє обладнання на PPPoE-авторизацію. У процесі авторизації буде виконана низка правил згідно з налаштуваннями RADIUS-сервера. Однак не всі наявні налаштування RADIUS-сервера задовольняють вимогам оператора. Особливістю роботи інтернет-оператора, для якого призначена ця система АСР, є надання послуги за публічним договором приєднання. Угода за цим договором виконується в електронному вигляді на спеціально створеній вебсторінці, що відображено на діаграмі (рис. 2.2).

### 2.3 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) показує конфігурацію вузлів-процесів, а також розміщувані на них компоненти [22, С. 42]. Діаграми розгортання дають статичне уявлення про архітектуру системи. Вузли, зазвичай, містять один або кілька артефактів. Артефакт у мові UML – це фізична сутність на рівні платформи реалізації [22, С. 369]. Загалом під процесом побудови діаграми розгортання розуміється моделювання топології апаратних засобів, на яких виконується система. Діаграму розгортання розроблюваної системи АСР наведено на рис. 2.3. Ця діаграма відображає фізичну архітектуру системи з урахуванням вимог та наявного обладнання замовника (ТОВ Телекомунікаційна компанія «НЕОН»).

Діаграма містить такі вузли:

- Border GW – прикордонний маршрутизатор, що забезпечує динамічну маршрутизацію протоколу TCP/IP між внутрішніми й зовнішніми вузлами Інтернету;
- Core network – фізична частина мережі, що відповідає за взаємодію вузлів ядра мережі;



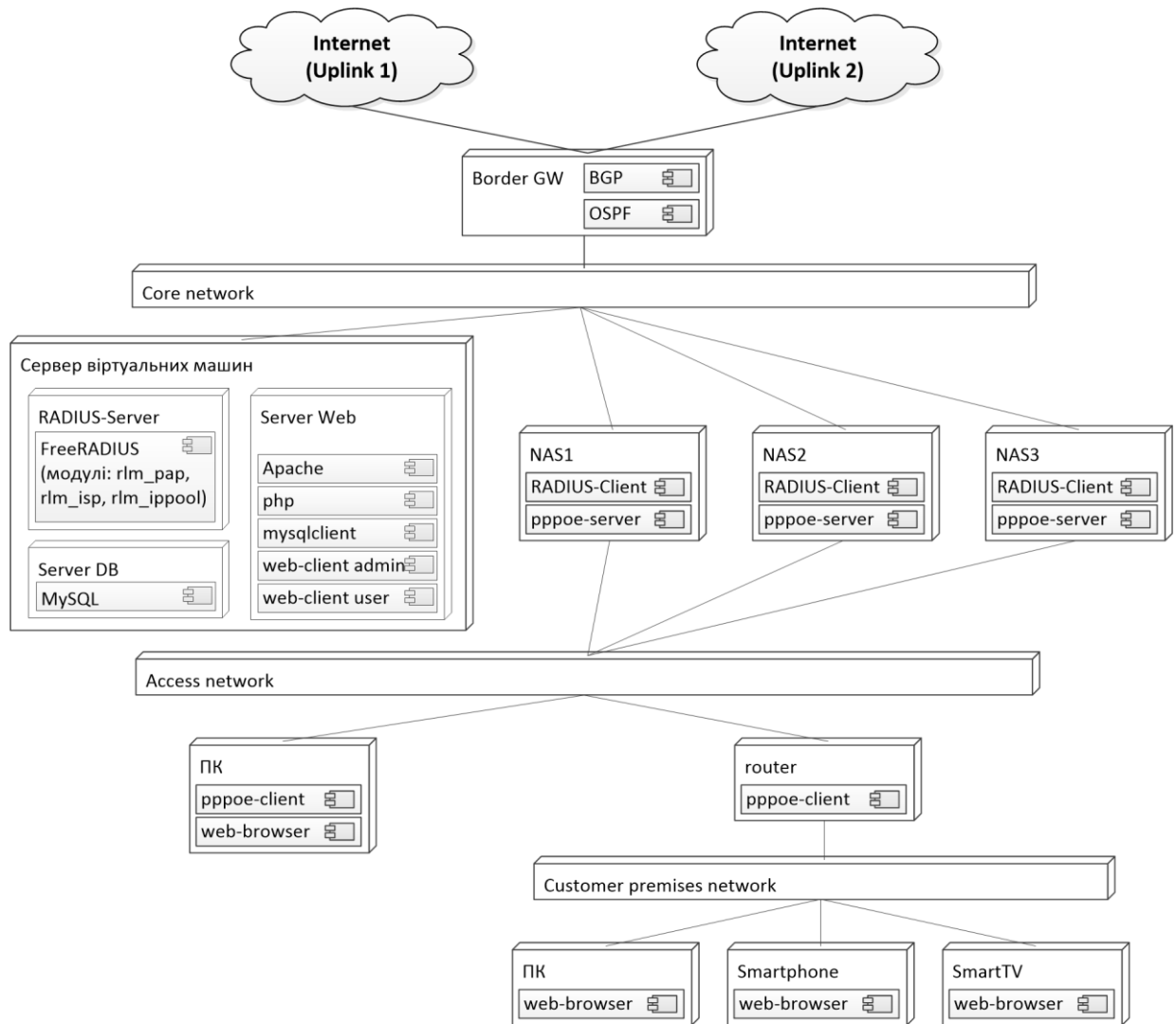


Рисунок 2.3 – Діаграма розгортання системи АСР оператора телекомунікацій з урахуванням вимог замовника (ТОВ Телекомунікаційна компанія «НЕОН»)

- Сервер віртуальних машин Linux KVM, на якому встановлені віртуальні машини. Застосування сервера віртуальних машин скорочує час для модернізації, портування та тестування серверів послуг, збільшує швидкість взаємодії між віртуальними серверами, а також дозволяє робити резервне копіювання системи у режимі «Hot Backup»;
- RADIUS-Server – сервер зі встановленою системою FreeRADIUS і модулями розширення;
- Server DB – сервер зі встановленою СКБД MySQL і БД;
- Server Web – сервер, що надає доступ до управління системою за протоколом HTTPS зі встановленими відповідними модулями і службами;

- NAS – мережевий сервер доступу, який забезпечує з'єднання клієнтів за протоколом PPPoE, зі встановленим RADIUS-клієнтом і іншими необхідними модулями;
- Access network – фізична частина мережі, що відповідає за з'єднання абонентів із їхнім безпосереднім постачальником послуг;
- ПК – комп'ютер абонента, який може підключатися без використання роутера, використовуючи клієнт PPPoE, або – з використанням роутера та без використання PPPoE. Для роботи з особистим кабінетом абонент мусить мати встановлений веббраузер;
- Router – програмно-апаратний пристрій, здатний підключатися через протокол PPPoE, що забезпечує доступ вузлів (ПК, smartphone, smartTV тощо) CPN до глобальної мережі Інтернет і до ресурсів оператора.

## 2.4 Інтерфейс користувача

Інтерфейс користувача (UI) – засіб зручної взаємодії користувача (людини) з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача. У графічних системах інтерфейс користувача втілюється багатовіконним режимом; змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон; їхнім розташуванням; сортуванням елементів вікон; гнучкими налагодженнями як самих вікон, так і окремих їхніх елементів (файли, каталоги, ярлики, шрифти тощо); доступністю багатокористувацьких налаштувань [24].

Система АСР має вебінтерфейс, що дає можливість адміністратору, менеджеру, абоненту користуватися відповідним функціоналом. Треба зазначити, що проектування дизайну вебінтерфейсу не є темою КРМ тому далі наведено тільки його опис у готовому вигляді.

Для адміністратора розроблено вебсторінку, що дає можливість отримати доступ до всіх функцій системи призначених для нього (рис. 2.4).

Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

Рисунок 2.4 – Головна сторінка порталу адміністратора

З головної сторінки порталу адміністратора можна виконувати такі дії: пошук із використанням фільтрів (рис. 2.4); переглядати поточну інформацію про тарифні плани та створити нові (рис. 2.5); сформуванню та переглянути фінансові звіти за певний період, надходження коштів та іншу фінансову інформацію; переглянути та виконати налаштування IP-адресних діапазонів (рис. 2.6), систем оплати; формування карток абонентів (створення, редагування, видалення); перегляд та створення особистої картки абонента (рис. 2.7); переглядати системну інформацію щодо роботи мережі оператора, наприклад, системний журнал RADIUS-серверу (рис. 2.8).

Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

Пакет	GID	Дії	Кількість абонентів	Білий IP	Примітка
<a href="#">Unlim-6M</a>	75	<a href="#">Зловити</a>			0.1. 6Мбіт/с, 140 грн/міс (сірий IP). Вига пара, 4.60 грн/день
<a href="#">Unlim-12M</a>	76	<a href="#">Зловити</a>			0.2. 12Мбіт/с, 180 грн/міс (сірий IP). Вига пара, 5.91 грн/день
<a href="#">Соціальний</a>	29	<a href="#">Зловити</a>			1.1. 10Мбіт/с, 50 грн/міс (сірий IP), для пенсіонерів
<a href="#">Домашній 25М</a>	38	<a href="#">Зловити</a>			2.1. 25Мбіт/с, 85грн/міс (сірий IP). Багатоповерхівки
<a href="#">Домашній 50М</a>	39	<a href="#">Зловити</a>			2.2. 50Мбіт/с, 130 грн/міс (сірий IP)
<a href="#">Домашній 80М</a>	40	<a href="#">Зловити</a>		+	2.3. 80Мбіт/с, 165 грн/міс (реальний IP)
<a href="#">Домашній 120М</a>	55	<a href="#">Зловити</a>		+	2.4. 120Мбіт/с, 200 грн/міс (реальний IP)
<a href="#">Домашній 250М</a>	56	<a href="#">Зловити</a>		+	2.5. 250Мбіт/с, 250 грн/міс (реальний IP)
<a href="#">Приватний 80М</a>	47	<a href="#">Зловити</a>			3.0. 80Мбіт/с, 165 грн/міс (сірий IP). Приватний сектор
<a href="#">Приватний-80М</a>	80	<a href="#">Зловити</a>		+	3.1. 80Мбіт/с, 165 грн/міс (реальний IP)
<a href="#">Приватний 120М</a>	48	<a href="#">Зловити</a>		+	3.2. 120Мбіт/с, 200 грн/міс (реальний IP)
<a href="#">Приватний 250М</a>	53	<a href="#">Зловити</a>		+	3.3. 250Мбіт/с, 250 грн/міс (реальний IP)
<a href="#">Приватний 500М</a>	54	<a href="#">Зловити</a>		+	3.4. 500Мбіт/с, 380 грн/міс (реальний IP)
<a href="#">Бізнес</a>	49	<a href="#">Зловити</a>		+	5.1. Бізнес. Швидкість і вартість пакета міняти в профілі користувача. GID для Бізнес-пакету не міняти!
<a href="#">Індивідуальний</a>	51	<a href="#">Зловити</a>			5.2. Індивідуальний. Швидкість і вартість пакета міняти в профілі користувача
<a href="#">NEON</a>	21	<a href="#">Зловити</a>		+	99.0. 100Мбіт/с, Офіс НЕОН, працівники, родичі
<a href="#">BLOCK-CYBER</a>	74	<a href="#">Зловити</a>			99.1. Пакет для заблокованих користувачів КиберНЕТ. GID не міняти!
<a href="#">BLOCK</a>	20	<a href="#">Зловити</a>			99.2. Пакет для заблокованих користувачів. GID не міняти!
<a href="#">PAY packet</a>	50	<a href="#">Зловити</a>			99.3. Спец. пакет для поповнення (Нічого не міняти!!! Не видаляти)
<b>Всього</b>					
Кількість заблокованих користувачів					

[Новий пакет](#)

Рисунок 2.5 – Сторінка перегляду та створення тарифних планів порталу адміністратора

Динамічні IP сірі		
N	Мережа	Адрес (V/I/3)
1	<a href="#">172.16.100</a>	251/165/225/26
2	<a href="#">172.16.101</a>	253/53/91/162
3	<a href="#">172.16.102</a>	253/138/199/54
4	<a href="#">172.16.103</a>	253/60/122/131
5	<a href="#">172.16.104</a>	253/49/108/145
6	<a href="#">172.16.105</a>	253/48/79/174
7	<a href="#">172.16.107</a>	253/84/150/103
8	<a href="#">172.16.108</a>	253/68/96/157
9	<a href="#">172.16.109</a>	253/157/224/29
10	<a href="#">172.16.110</a>	253/186/235/18
11	<a href="#">172.16.111</a>	253/159/214/39
Всього:		2781/1167/1743/1038

Динамічні IP білі		
N	Мережа	Адрес (V/I/3)
1	<a href="#">93.170.132</a>	249/53/247/2
2	<a href="#">93.170.133</a>	249/62/249/0
3	<a href="#">93.171.170</a>	155/42/155/0
4	<a href="#">93.171.171</a>	247/56/247/0
5	<a href="#">93.171.250</a>	253/58/224/29
6	<a href="#">93.171.251</a>	251/47/214/37
Всього:		1404/318/1336/68

Рисунок 2.6 – Сторінка перегляду та налаштування IP-адресних діапазонів

Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

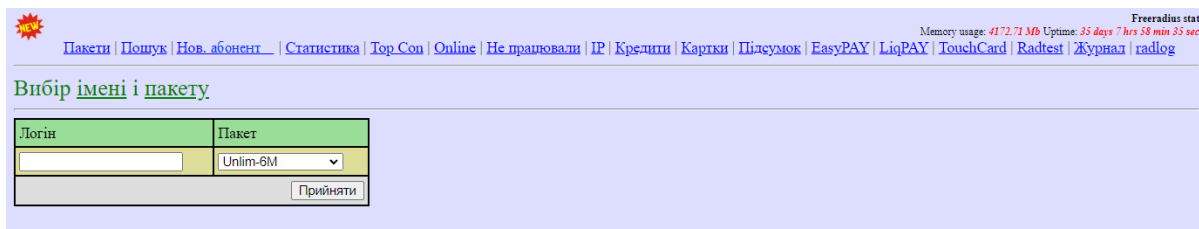


Рисунок 2.7 – Сторінка створення особистої картки абонента

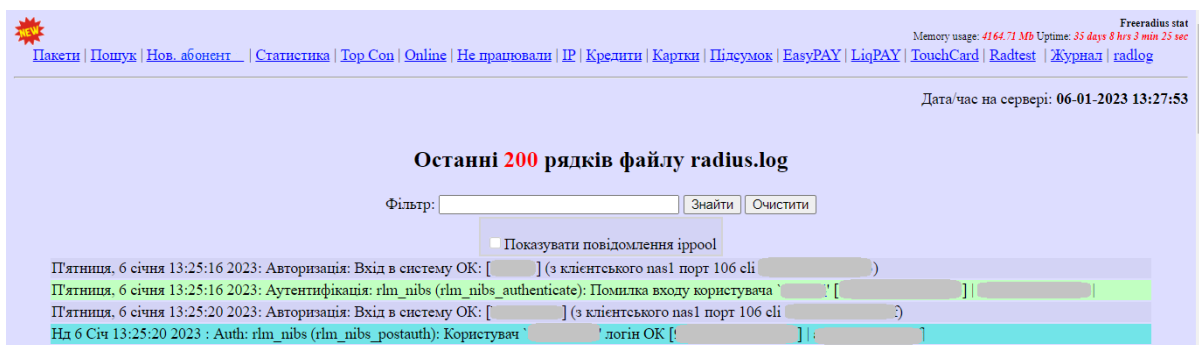


Рисунок 2.8 – Системний журнал RADIUS-серверу

Інтерфейс системи АСР, що призначено для менеджера (оператора) схожий з UI адміністратора (рис. 2.4, 2.9). Відмінність полягає в обмеженні доступного функціоналу згідно з відведеною роллю (рис. 2.1).

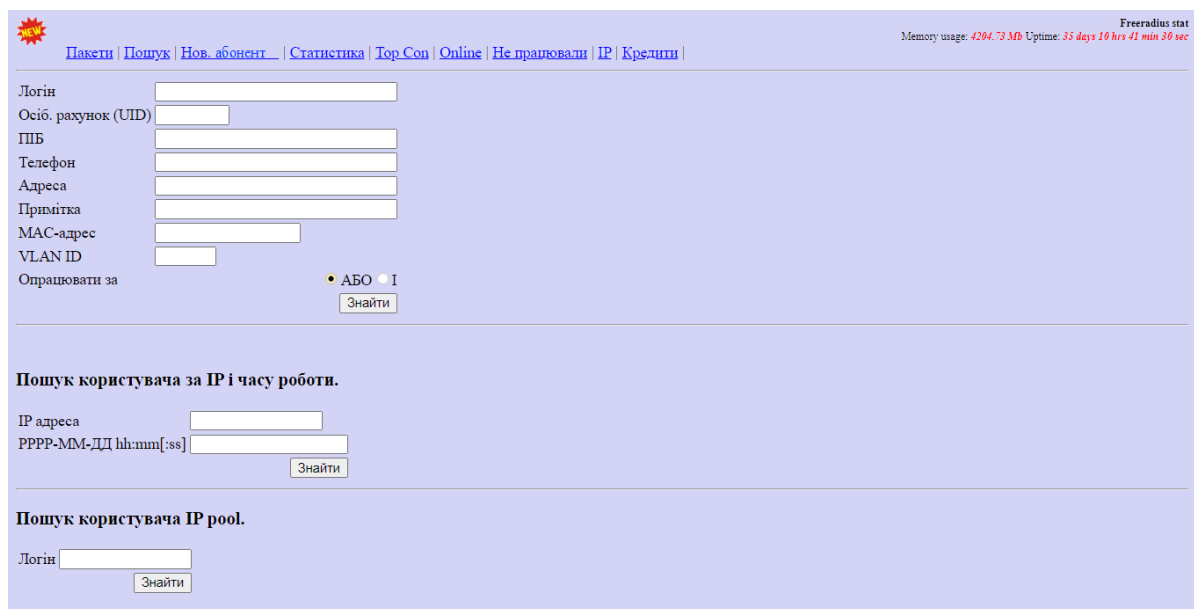


Рисунок 2.9 – Головна сторінка порталу менеджера

Абонент має доступ до відповідного функціоналу системи АСР з особового кабінету абонента. Доступ до кабінету можна дістати з корпоративного сайту компанії через авторизацію в системі (рис. 2.10).

Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

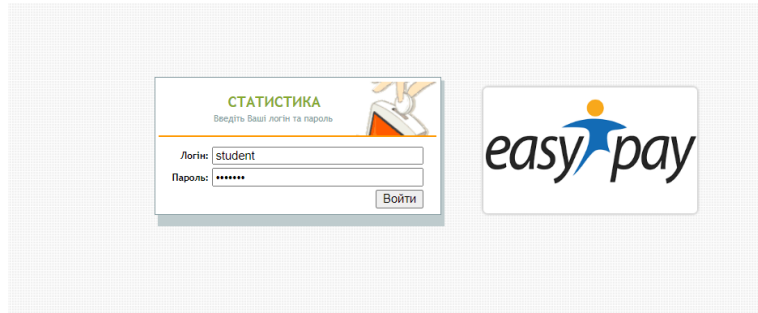


Рисунок 2.10 – Сторінка авторизації в системі APC

На рис. 2.11 наведено вигляд головної сторінки особового кабінету абонента. Тут він може переглянути загальну статистику або за обраний період: за з'єднаннями (підключень), за рухом коштів (2.11-12). Окрім цього абонент може здійснити платіж або активувати функцію довірчого платежу.

Статистика роботи в мережі Інтернет

Увага! Обнулення кредитів відбувається автоматично кожну середу о 12:00!

Логін	Особовий рахунок	Поточний тарифний план	ПІБ	Дата реєстрації	Останнє підключення	Швидкість з'єднання	Логін вкл./викл.	Стан рахунку	Довір. платіж
student	02847	NEON	РАХУНОК НЕ ПОПОВНОВАТИ	2020-05-07	2023-01-01	100000 Kbps	включений	-79.99 грн.	0.00 грн.

Код поповнення рахунку:

Примітка: у полі "Код поповнення рахунку" дефісні додаються автоматично.

Оплатити за допомогою VISA/MASTER/Приват24  грн.

Детальніше...

з: 2023 Січень групувати: НІМАЄ

за: 2023 Січень

З'єднань	Використано часу	Передає трафіку (Мбайт)	Отримано трафіку (Мбайт)	Знято грошей за обраний період	Кількість поповнень (Історія)	Поповнення рахунку
9	31:43:13	151.450	2686.922	0.00 грн.	0	0.00 грн.

Підключився	Відключився	Тривалість підключення	Передає трафіку (Мбайт)	Отримано трафіку (Мбайт)	MAC-адрес	До зняття (грн.)	Знято (грн.)	Залишок (грн.)
Дата			Знято системою: Цифрове ТВ "YouTV. Універсальний"			До зняття	Знято	Після зняття
2023-01-01 00:00:03						0.01	129.00	-128.99
Дата			Додано системою: Знижка за комбінування послуг			До поповнення	Добавлено	Після поповнення
2023-01-01 00:00:03						-128.99	49.00	-79.99
2023-01-01 00:00:20	2023-01-01 08:06:19	8:06:00	1.985	46.859	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-01 08:23:30	2023-01-01 21:08:00	12:44:30	69.516	1708.298	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-01 21:14:22	2023-01-02 05:54:21	8:40:00	66.484	653.153	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-02 07:34:50	2023-01-02 07:54:28	0:29:38	5.397	78.812	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-02 07:54:37	2023-01-02 09:31:58	1:37:30	7.265	189.099	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-02 10:37:20	2023-01-02 10:40:57	0:03:38	0.750	10.489	c0:25:2f:19:10:cd	-79.99	0.00	-79.99
2023-01-04 11:42:06	2023-01-04 11:42:10	0:00:04	0.004	0.001	58:8a:5a:24:ad:11	-79.99	0.00	-79.99
2023-01-04 11:43:06	2023-01-04 11:44:55	0:01:49	0.043	0.010	58:8a:5a:24:ad:11	-79.99	0.00	-79.99
2023-01-04 11:44:58	2023-01-04 11:45:02	0:00:04	0.005	0.001	58:8a:5a:24:ad:11	-79.99	0.00	-79.99

Рисунок 2.11 – Головна сторінка особового кабінету абонента

stat.neon.mk.ua :: Статистика роботи в мережі Інтернет - nEon ::

Історія змін рахунку користувача **СТУДЕНТ**  
період з 2023-01-01 0:00:00 за 2023-01-31 23:59:59

Дата	Стан рахунку до його зміни	Сума поповнення	Цільове призначення	Оператор
2023-01-01 00:00:03	0.01	129.00	Знято системою: Цифрове ТЕЛЕБАЧЕННЯ "YouTV. Універсальний"	'system'
2023-01-01 00:00:03	-128.99	49.00	Додано системою: Знижка для комбінування послуг	'system'

Рисунок 2.12 – Історія руху коштів абонента

У подальшому UI системи АСР може бути розширено через створення мобільного застосунку та удосконалено шляхом використання сучасних технологій розроблення вебінтерфейсів.

## **Висновки до розділу 2**

Етап проектування є дуже важливим етапом у створенні будь-якої системи. Моделювання дає уявлення про бажану структуру та поведінку системи. Окрім цього воно необхідне для візуалізації і керування її архітектурою. Отже, для кращого розуміння того, як має виглядати система АСР оператора телекомунікацій з урахуванням вимог замовника (ТОВ Телекомунікаційна компанія «НЕОН») було побудовано три діаграми: варіантів використання, діяльності та розгортання. Діаграма варіантів використання описує функціональне призначення системи АСР оператора телекомунікацій, де виокремлено такі групи користувачів: адміністратор, менеджер, абонент. Кожна група користувачів використовує систему за своїми потребами та наданим доступом. Діаграма діяльності демонструє логіку взаємодії клієнта із системою АСР. На діаграмі показано потік об'єктів та надано пояснення щодо певних етапів процесу ААА. Діаграма розгортання показує конфігурацію вузлів-процесів системи АСР оператора телекомунікацій з урахуванням вимог та наявного обладнання замовника, що дає уявлення про архітектуру системи.

Окрім моделей також наведено опис інтерфейсу користувача АСР оператора телекомунікацій, який розроблено з урахуванням вимог та побажань замовника. Окреслено подальший напрям його вдосконалення.

## 3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ

### 3.1 Вибір найбільш доцільної операційної системи

Вибір ОС впливає на процес розробки та роботи системи. Тому це є дуже важливим етапом створення готового продукту.

Операційна система – це базовий комплекс програм, що виконує управління апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем. ОС зазвичай складається з ядра ОС та базового набору прикладних програм. До складу ОС входять:

- ядро ОС, що забезпечує розподіл та управління ресурсами обчислювальної системи;
- базовий набір прикладних програм, системні бібліотеки та програми обслуговування.

Ядро ОС – це набір функцій, структур даних та окремих програмних модулів, які завантажуються в пам'ять комп'ютера під час завантаження ОС та забезпечують три типи системних сервісів:

- управління введенням-виведенням інформації (підсистема введення-виведення ядра ОС);
- управління оперативною пам'яттю (підсистема управління оперативною пам'яттю ядра ОС);
- управління процесами (підсистема управління процесами ядра ОС).

Найбільш розповсюдженими ОС згідно з багатьма рейтингами є: Android, Windows, IOS, OS X і Linux. Якщо подивитися статистику використання ОС, зокрема, для вебсерверів, то тут розподіл дещо інший. Так, за даними аналітичної компанії W3Cook можна побачити, що найбільш популярною є ОС Linux (рис. 3.1). Діаграма рис. 3.1 показує статистику використання та частку ринку ОС для вебсерверів. На цій діаграмі немає даних щодо ОС на ПК. Лідером є ОС Linux – безкоштовна ОС із відкритим вихідним кодом, що використовується приблизно



половиною вебсайтів. Ubuntu, CentOS, Debian – це дистрибутиви ОС Linux. ОС Windows від Microsoft не є безкоштовна ОС і її частка на ринку становить лише 1,9%.

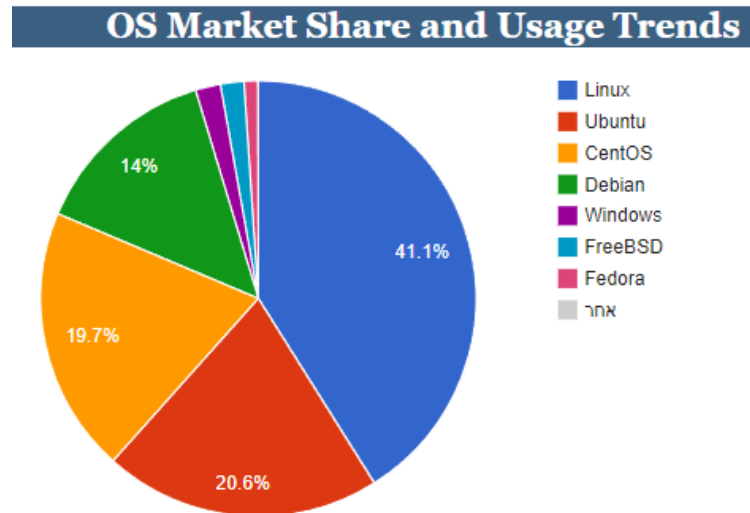


Рисунок 3.1 – Частка ринку ОС використовуваних вебсерверами [25]

За даними іншої аналітичної компанії, частка використання UNIX-подібних ОС для вирішення тих самих завдань складає 80,5%, ОС Windows – 19,8% [26]. Отже, розглянемо більш детально клас UNIX-подібних ОС.

Linux – загальна назва UNIX-подібних операційних систем на основі однойменного ядра. Здебільшого в дистрибутивів вихідний код є доступним усім для використання, зміни та вільного розповсюдження. Спершу ОС Linux розроблялася для використання окремими ентузіастами на своїх ПК, пізніше, завдяки підтримці таких компаній, як IBM, Sun Microsystems, HP, Novell та інших, система набула значного поширення як серверна ОС. Це можна побачити з наведеної вище статистики.

ОС Linux портовано на велику кількість апаратних платформ. ОС досить успішно використовується як на мейнфреймах та суперкомп'ютерах, так і вбудована в багато інших пристроїв (смартфони, планшетні ПК, маршрутизатори комп'ютерних мереж, пристрої автоматки, системи управління телевізорами та ігровими консолями тощо). Від середини 1990-х років Linux все частіше встановлюється і на ПК. Значна кількість спеціалізованих дистрибутивів Linux,

котрі розробляються та підтримуються різними спільнотами, дає широкі можливості вибору програмного забезпечення.

Найпопулярнішим дистрибутивом Linux є Ubuntu [25]. Ubuntu – це дистрибутив GNU/Linux, ґрунтований на Debian GNU/Linux. Основним розробником і спонсором є компанія Canonical. Проект активно розвивається і підтримується вільною спільнотою. За твердженнями Canonical, Ubuntu використовується приблизно 20 мільйонами користувачів у всьому світу. Цей дистрибутив є 1-м у списку найбільш популярних дистрибутивів GNU/Linux для вебсерверів.

Серед інших ОС як серверна ОС найбільш популярною та розповсюдженою є Windows Server – лінійка серверних операційних систем від компанії Microsoft. Більш детально із цією ОС можна ознайомитися на офіційному сайті ([www.microsoft.com/ru-ru/windows-server](http://www.microsoft.com/ru-ru/windows-server)). Досить потужна ОС, підтримує широкий функціонал, проте є досить вартісною з огляду фінансових обмежень наданих замовником.

Аналіз наявних ОС дав можливість прийняти рішення щодо вибору ОС. Було обрано дистрибутив Linux – Ubuntu. Ця ОС має весь необхідний функціонал, є гнучкою в налаштуванні і використанні, що дає можливість розробнику конфігурувати систему максимально зручно для своїх завдань.

### **3.2 Мови програмування та середовище розробки**

Згідно зі специфікаціями вимог (п. 1.5) систему АСР можна поділити на дві складові: backend та frontend. До backend належить рівень AAA, до frontend – вебінтерфейс. Для реалізації backend обрано: сервер із відкритим вихідним кодом FreeRADIUS; середовище розробки модульних кросплатформних застосунків Eclipse, реляційна СКБД із відкритим вихідним кодом MySQL, мову C та компілятор GCC, мову Python.

Одним із найпопулярніших RADIUS-серверів є FreeRADIUS.

Офіційний сайт: [freeradius.org](http://freeradius.org)

FreeRADIUS – RADIUS-сервер із відкритим вихідним кодом. Це альтернатива інших комерційних RADIUS-серверів, оскільки він модульний і функціональний. Крім того, він входить у п'ятірку RADIUS-серверів світу з погляду розгортання і кількості користувачів, яких цей сервер авторизує щодня.

Перевагою його є можливість ефективної роботи на вбудованих системах із невеликим обсягом пам'яті. FreeRADIUS швидкий, гнучкий, а також підтримує більше протоколів автентифікації, ніж більшість комерційних серверів. Сервер поставляється разом з інструментом управління через вебінтерфейс – dialupadmin, який написано на PHP.

Як уже було зазначено, FreeRADIUS має модульну архітектуру де кожен модуль відповідає за виконання певних завдань, наприклад, для авторизації користувачів за протоколом SMB використовується модуль `rlm_smb`, для роздачі IP-адрес із БД SQL використовується модуль `rlm_sqlipool`. До складу FreeRADIUS входять десятки різних модулів і якщо в його складі немає модуля необхідного для вирішення певного завдання, то модуль можна розробити самостійно, зокрема, з використанням мови C. Отже, для написання модулів АСР було обрано саме цю мову програмування.

Мова C – універсальна, процедурна, імперативна мова програмування загального призначення, розроблена в 1972 році Денісом Рітчі для того, щоб написати нею ОС UNIX. Хоча C і було розроблено для написання системного програмного забезпечення, наразі вона досить часто використовується для написання прикладного програмного забезпечення.

Мова C проєктувалася для машинно-незалежного програмування та є стандартизованою. Машинно-незалежна програма, написана мовою C, може компілюватися на великій кількості апаратних платформ та ОС із мінімальними змінами. Мова стала доступною для великої кількості платформ, від вбудованих мікроконтролерів до суперкомп'ютерів.

Як і більшість імперативних мов, ґрунтованих на традиції АЛГОЛ, C має можливості для структурного програмування і дає змогу здійснювати рекурсії, у

той час, як система статичної типізації даних запобігає виникненню багатьох непередбачуваних операцій. У мові С увесь виконуваний код міститься у функціях. Параметри функції завжди передаються за значеннями. Передача параметрів за вказівником реалізується через передачу значення вказівника. Гетерогенні сукупності типів даних (структури) дають можливість пов'язаним типам даних бути об'єднаними й маніпулювати ними, як єдиним цілим.

Мова С також має такі специфічні властивості [27]:

- змінні можуть бути прихованими у вкладених блоках;
- слабка типізація, наприклад, символи можуть використовуватися, як цілі числа;
- низькорівневий доступ до оперативної пам'яті через перетворення машинних адрес у вказівники;
- вказівники на функції і дані підтримують динамічний поліморфізм;
- масив індексів як вторинне поняття, визначається в термінах арифметики вказівників;
- стандартизований препроцесор С для макровизначення, включення файлу з вихідним кодом, умовної трансляції тощо;
- комплексна функціональність, як то I/O, маніпуляція рядками, і делегування математичних функцій бібліотекам;
- порівняно невелика кількість зарезервованих слів;
- лексичні структури, які нагадують В (Бі) більше за ALGOL.

Середовищем розробки розширень RADIUS-серверу обрано Eclipse – відкрите інтегроване середовище розробки модульних кросплатформних застосунків. Це середовище має низку переваг. Основними з яких є такі. Eclipse має велику кількість розширень: будь-який розробник може розширити Eclipse своїми модулями. Вже є Java Development Tools (JDT), C/C++ Development Tools (CDT), розробляються модулі для мов Ada, COBOL, FORTRAN, PHP, X10 (X10DT) тощо. Безліч розширень доповнює середовище Eclipse диспетчерами для роботи з БД, серверами застосунків тощо. Також через безкоштовність і високу

якість, Eclipse у багатьох організаціях є корпоративним стандартом для розробки застосунків.

Для створення процедури перевірки прийняття Угоди, що описано у п. 2.2 обрано мову Python – інтерпретовану об'єктно-орієнтовану мову програмування високого рівня зі строгою динамічною типізацією. Її розроблено в 1990 році Гвідо ван Россумом. В останнє десятиліття ця мова є стандартною скриптовою мовою для Linux систем і встановлюється автоматично.

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі для всіх основних платформ. Усе це доступно з вебсайту Python [28] і може вільно розповсюджуватись. Також тут міститься документація, дистрибутиви й посилання на багато безплатних сторонніх модулів Python, програми та інструменти.

У мові Python підтримується кілька парадигм програмування, зокрема, об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована. Перевагою її використання для написання сценаріїв у порівнянні зі спеціалізованими засобами інтерпретації скриптів (Bash, Sh тощо) є:

- більш зрозумілий та простий синтаксис;
- засоби стандартних та інших бібліотек Python є більш зручними та функціональними ніж консольні утиліти.

Для реалізації frontend обрано: Apache HTTP Server та мову PHP. Треба зауважити, що опис реалізації frontend не увійшов в обсяг КРМ.

### **3.3 Створення модуля розширення RADIUS-сервера**

Треба зауважити, що через те, що ця система АСР розроблялася на замовлення ТОВ ТК «НЕОН», лістинг його коду підпадає під дію законодавства України у сфері захисту авторських прав, і він не може бути повністю викладений у цій роботі. Для ознайомлення зі структурою коду у п. 3.3 та додатках буде

наведено частину лістингу коду, що не несе інформації, яка не підлягає розповсюдженню.

Для реалізації логіки взаємодії клієнта із системою АСР було розроблено відповідний модуль розширення RADIUS-сервера – `rlm_isp`. Модуль має таку структуру:

- ім'я модуля FreeRADIUS має відповідати певному синтаксису – `rlm_XXX`, тому було задано ім'я **`rlm_isp`**;
- каталог із файлами модуля має розташовуватися в структурі вихідних файлів серверу FreeRADIUS в каталозі **`src/modules`**;
- основний код міститься у файлі **`rlm_isp.c`**, а визначення змінних, констант, прототипи функцій – у файлі **`rlm_isp.h`**.

Файл `rlm_isp.c`. У цьому файлі обов'язково має бути присутня структура з ім'ям модуля. Для розроблюваної системи АСР структуру модуля наведено в ліст. 3.1.

### Лістинг 3.1 – Структура модуля `rlm_isp`

```
module_t rlm_isp = {
    "ISP",
    RLM_TYPE_THREAD_SAFE,
    rlm_isp_init,          /* initialization */
    rlm_isp_instantiate,  /* instantiation */
    {
        rlm_isp_authenticate, /* authenticate */
        rlm_isp_authorize,    /* authorize */
        NULL,                 /* preaccounting */
        rlm_isp_accounting,   /* accounting */
        rlm_isp_checks simul, /* checksimul */
        NULL,                 /* pre-proxy */
        NULL,                 /* post-proxy */
        rlm_isp_postauth,     /* post-auth */
    },
    rlm_isp_detach,        /* detach */
    rlm_isp_destroy,      /* destroy */
};
```

У структурі `rlm_isp` оголошено функції, які реалізує модуль. У коментарях описано призначення кожної з них. Щодо реалізації призначення системи АСР основними функціями є: `rlm_isp_authenticate` (`authenticate`), `rlm_isp_authorize` (`authorize`), `rlm_isp_accounting` (`accounting`).

Сервер FreeRADIUS одержує запит доступу Access-Request від сервера доступу, який містить ім'я користувача, адрес та порт сервера доступу, CHAP-Challenge та CHAP-Password (рис. 3.2) і починає його обробляти. Сервер FreeRADIUS виконує оброблення поетапно проходячи по ланцюгу модулів, що вказано в налаштуванні RADIUS-сервера.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	93.171.170.240	93.171.170.254	RADIUS	195	Access-Request id=1
2	0.041737	93.171.170.254	93.171.170.240	RADIUS	104	Access-Accept id=1
3	0.067181	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=1
4	0.106415	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=1
5	60.106753	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=2
6	60.112103	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=2
7	120.106738	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=3
8	120.111808	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=3
9	180.106734	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=4
10	180.112275	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=4
11	240.106744	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=5
12	240.111719	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=5
13	259.933848	93.171.170.240	93.171.170.254	RADIUS	272	Accounting-Request id=6
14	259.939339	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=6

```

> Frame 1: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits)
> Ethernet II, Src: RealtekU_e0:b6:bb (52:54:00:e0:b6:bb), Dst: RealtekU_b4:c5:e3 (52:54:00:b4:c5:e3)
> Internet Protocol Version 4, Src: 93.171.170.240, Dst: 93.171.170.254
> User Datagram Protocol, Src Port: 36469, Dst Port: 1812
▼ RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0x1 (1)
  Length: 153
  Authenticator: 50beb10fd7b989fe1b66f01c575ea0b1
  [The response to this request is in frame 2]
▼ Attribute Value Pairs
  > AVP: t=User-Name(1) l=11 val=Test_user
  > AVP: t=NAS-Identifier(32) l=11 val=accel-ppp
  > AVP: t=NAS-IP-Address(4) l=6 val=93.171.170.240
  > AVP: t=NAS-Port(5) l=6 val=0
  > AVP: t=NAS-Port-Id(87) l=6 val=ppp0
  > AVP: t=NAS-Port-Type(61) l=6 val=Virtual(5)
  > AVP: t=Service-Type(6) l=6 val=Framed(2)
  > AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
  > AVP: t=Calling-Station-Id(31) l=19 val=52:54:00:9e:1d:25
  > AVP: t=Called-Station-Id(30) l=19 val=52:54:00:e0:b6:bb
  > AVP: t=CHAP-Challenge(60) l=18 val=50beb10fd7b989fe1b66f01c575ea0b1
  > AVP: t=CHAP-Password(3) l=19 val=0191789ac0b8a225e8bbf2c6a341a1b8e3

```

Рисунок 3.2 – Результат виконання Access-Request

Коли оброблення доходить до модуля rlm\_isc, то починають виконуватися його функції: rlm\_isc\_authenticate (authenticate) та rlm\_isc\_authorize (authorize). rlm\_isc\_authenticate (authenticate) – виконує валідацію імені користувача та його паролю. У разі, якщо такий користувач є і його пароль вірний, функція повертає код RLM\_MODULE\_OK – запит дозволено або модуль успішно обробив запит.

Відповідь FreeRADIUS буде визначатися обробкою більш пізніх модулів. Якщо це останній модуль, то відповіддю буде ОК. Коди значень, що повертаються модулем, описані в старих версіях FreeRADIUS у файлі заголовка `src/include/modules.h` або `src/include/radiusd.h` [29] – у новіших версіях.

Функція `rlm_osp_authenticate` (`authenticate`) – задає параметри користувача, наприклад, IP-адрес, адреси DNS серверів, швидкість трафіку тощо. Ці параметри повертаються у відповіді `Access-Accept` від FreeRADIUS у вигляді пар значень.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	93.171.170.240	93.171.170.254	RADIUS	195	Access-Request id=1
2	0.041737	93.171.170.254	93.171.170.240	RADIUS	104	Access-Accept id=1
3	0.067181	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=1
4	0.106415	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=1
5	60.106753	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=2
6	60.112103	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=2
7	120.106738	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=3
8	120.111808	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=3
9	180.106734	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=4
10	180.112275	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=4
11	240.106744	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=5
12	240.111719	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=5
13	259.933848	93.171.170.240	93.171.170.254	RADIUS	272	Accounting-Request id=6
14	259.939339	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=6

```

> Frame 2: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
> Ethernet II, Src: RealtekU_b4:c5:e3 (52:54:00:b4:c5:e3), Dst: RealtekU_e0:b6:bb (52:54:00:e0:b6:bb)
> Internet Protocol Version 4, Src: 93.171.170.254, Dst: 93.171.170.240
> User Datagram Protocol, Src Port: 1812, Dst Port: 36469
v RADIUS Protocol
  Code: Access-Accept (2)
  Packet identifier: 0x1 (1)
  Length: 62
  Authenticator: b4bc08a3947452fbfaf9706971951130
  [This is a response to a request in frame 1]
  [Time from request: 0.041737000 seconds]
v Attribute Value Pairs
  > AVP: t=Acct-Interim-Interval(85) l=6 val=60
  > AVP: t=Service-Type(6) l=6 val=Framed(2)
  > AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
  > AVP: t=Framed-Compression(13) l=6 val=Van-Jacobson-TCP-IP(1)
  > AVP: t=PPPD-Downstream-Speed-Limit(231) l=6 val=120000
  v AVP: t=PPPD-Upstream-Speed-Limit(230) l=6 val=120000
    Type: 230
    Length: 6
    PPPD-Upstream-Speed-Limit: 120000
  v AVP: t=Framed-IP-Address(8) l=6 val=93.171.171.150
    Type: 8
    Length: 6
    Framed-IP-Address: 93.171.171.150

```

Рисунок 3.3 – Результат виконання `Access-Accept`

Третя функція процесу AAA – це `rlm_osp_accounting` (`accounting`). Призначення цієї функції в модулі `rlm_osp` полягає в записі даних і сесії користувача в БД. Сервер доступу відправляє кожні 60 секунд запит RADIUS-



серверу Accounting-Request, який містить пари значень, наприклад, ID сесії, її тривалість, обсяг отриманих або відправлених даних тощо (рис. 3.4). Частина отриманих даних зберігається в БД.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	93.171.170.240	93.171.170.254	RADIUS	195	Access-Request id=1
2	0.041737	93.171.170.254	93.171.170.240	RADIUS	104	Access-Accept id=1
3	0.067181	93.171.170.240	93.171.170.254	RADIUS	266	Accounting-Request id=1
4	0.106415	93.171.170.254	93.171.170.240	RADIUS	62	Accounting-Response id=1

```

> Frame 3: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits)
> Ethernet II, Src: RealtekU_e0:b6:bb (52:54:00:e0:b6:bb), Dst: RealtekU_b4:c5:e3 (52:54:00:b4:c5:e3)
> Internet Protocol Version 4, Src: 93.171.170.240, Dst: 93.171.170.254
> User Datagram Protocol, Src Port: 57197, Dst Port: 1813
▼ RADIUS Protocol
  Code: Accounting-Request (4)
  Packet identifier: 0x1 (1)
  Length: 224
  Authenticator: 01bfb37123da9c8b7d50552fc30dfad6
  [The response to this request is in frame 4]
▼ Attribute Value Pairs
  > AVP: t=User-Name(1) l=11 val=Test_user
  > AVP: t=NAS-Identififier(32) l=11 val=accel-ppp
  > AVP: t=NAS-IP-Address(4) l=6 val=93.171.170.240
  > AVP: t=NAS-Port(5) l=6 val=0
  > AVP: t=NAS-Port-Id(87) l=6 val=ppp0
  > AVP: t=NAS-Port-Type(61) l=6 val=Virtual(5)
  > AVP: t=Service-Type(6) l=6 val=Framed(2)
  > AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
  > AVP: t=Calling-Station-Id(31) l=19 val=52:54:00:9e:1d:25
  > AVP: t=Called-Station-Id(30) l=19 val=52:54:00:e0:b6:bb
  > AVP: t=Acct-Status-Type(40) l=6 val=Start(1)
  > AVP: t=Acct-Authentic(45) l=6 val=RADIUS(1)
  > AVP: t=Acct-Session-Id(44) l=18 val=26d6cc6512e699df
  > AVP: t=Acct-Session-Time(46) l=6 val=0
  > AVP: t=Acct-Input-Octets(42) l=6 val=0
  > AVP: t=Acct-Output-Octets(43) l=6 val=0
  > AVP: t=Acct-Input-Packets(47) l=6 val=0
  > AVP: t=Acct-Output-Packets(48) l=6 val=0
  > AVP: t=Acct-Input-Gigawords(52) l=6 val=0
  > AVP: t=Acct-Output-Gigawords(53) l=6 val=0
  > AVP: t=Framed-IP-Address(8) l=6 val=93.171.171.150
  > AVP: t=Framed-Interface-Id(96) l=10 val=c43734190647a304
  > AVP: t=Framed-IPv6-Prefix(97) l=20 val=2001:678:9cc:a001::/64

```

Рисунок 3.4 – Результат виконання запитів Accounting-Request в процесі роботи PPPoE-сесії

У всі три функції (`rlm_isp_authorize`, `rlm_isp_authenticate`, `rlm_isp_accounting`) передається два параметри. Перший – вказівник типу `void`, який надалі приводиться до типу `sql_inst`. Тип `sql_inst` – це структура, яка містить параметри для підключення до бази даних, що беруться з файлу конфігурації (додат. А), її оголошення міститься у файлі `rlm_isp.h`. Структура `sql_inst` наведено

в ліст. 3.2. Значеннями може бути, наприклад, параметри підключення до бази й запити.

### Лістинг 3.2 – Структуру sql\_inst

```
typedef struct sql_inst {
    time_t          connect_after;
    SQLSOCK        *sqlpool;
    SQLSOCK        *last_used;
    ISP_CONFIG     *config;
    lt_dlhandle     handle;
    rlm_nibs_module_t *module;
} SQL_INST;
```

Другий параметр – це вказівник на структуру REQUEST, яка містить різні дані для підключення клієнта до RADIUS-сервера. Структура об'єкта REQUEST оголошено у файлі include/radiusd.h та наведено в ліст. 3.3. А також структуру RADIUS\_PACKET, що містить передані RADIUS-серверу дані, такі як RADIUS-пари атрибут-значення. У ліст. 3.4 наведено прототипи цих двох функцій.

### Лістинг 3.3 – Структуру REQUEST

```
typedef struct auth_req {
#ifdef NDEBBUG
    uint32_t          magic; /* for debugging only */
#endif
    RADIUS_PACKET    *packet;
    RADIUS_PACKET    *proxy;
    RADIUS_PACKET    *reply;
    RADIUS_PACKET    *proxy_reply;
    VALUE_PAIR       *config_items;
    VALUE_PAIR       *username;
    VALUE_PAIR       *password;
    request_data_t   *data;
    char              secret[32];
    child_pid_t      child_pid;
    time_t           timestamp;
    int               number; /* internal server number */

    /*
     * We could almost keep a const char here instead of a
     * _copy_ of the secret... but what if the RADCLIENT
     * structure is freed because it was taken out of the
     * config file and SIGHUPed?
     */
    char              proxysecret[32];
    int               proxy_try_count;
    int               proxy_outstanding;
    time_t            proxy_next_try;
    int               simul_max;
    int               simul_count;
    int               simul_mpp; /* WEIRD: 1 is false, 2 is true */
    int               finished;
    int               options; /* miscellaneous options */
}
```

```

const char      *component;
const char      *module;
void            *container;
} REQUEST;

```

Лістинг 3.4 – Прототипи функцій `rlm_ism_authorize`, `rlm_ism_authenticate`, `rlm_ism_accounting`

```

static int rlm_ism_authorize(void *instance, REQUEST * request);
static int rlm_ism_authenticate(void *instance, REQUEST * request);
static int rlm_ism_accounting(void *instance, REQUEST * request);

```

Для роботи з налаштуваннями підключення PPPoE у файлі модуля `rlm_ism.c` створено масив `module_config` який містить масив структур `CONF_PARSER`. Модуль одержує параметри налаштування з конфігураційного файлу. Фрагмент масиву структур `CONF_PARSER` модуля `rlm_ism` наведено в додатку Б.

Налаштування облікового запису абонента зберігаються в структурі `isp_user`. У ліст. 3.5 наведено фрагмент структури `isp_user`, у додатку В наведено повний текст коду оголошення структури. Структура містить поля, що зберігають облікові дані абонента (ПІБ, пароль, дані рахунку, рух коштів, дані споживання трафіку, службову інформацію тощо).

Лістинг 3.5 – Структура `isp_user`

```

typedef struct isp_user {
    char      name[65];
    char      prefix[5];
    char      passwd[255];
    ...
    ULONGINT  current_in_bytes;
    ULONGINT  current_out_bytes;
    ULONGINT  sub_total_time;
    ULONGINT  sub_total_traffic;
    double    sub_total_money;
    double    sub_money;
    char      allowed_servers[MAX_SERVERS_PATH];
    uint32_t  nas_ip;
    char      nas_address[16];
} ISP_USER;

```

Загальні налаштування модуля `rlm_ism` збережено в конфігураційному файлі **`isp.conf`**. У додатку А наведено повний зміст цього файлу. Файл містить службову інформацію щодо БД (тип, дані для з'єднання, конфігурацію, запити (перегляд, додавання, оновлення, видалення записів) тощо), щодо поточної сесії абонента тощо.

### 3.4 Створення процедури перевірки прийняття Угоди

Як уже зазначалося у п. 2.2 особливістю роботи інтернет-оператора, для якого призначена ця система АСР, є надання послуги за публічним договором приєднання. Угода за цим договором виконується в електронному вигляді на спеціально створеній вебсторінці (рис. 3.5), що відображено на діаграмі (рис. 2.2). Тобто абонент може отримати послугу доступу до мережі Інтернет тільки прийняв угоду з Договором приєднання. Функцію перевірки прийняття угоди покладено на систему АСР.

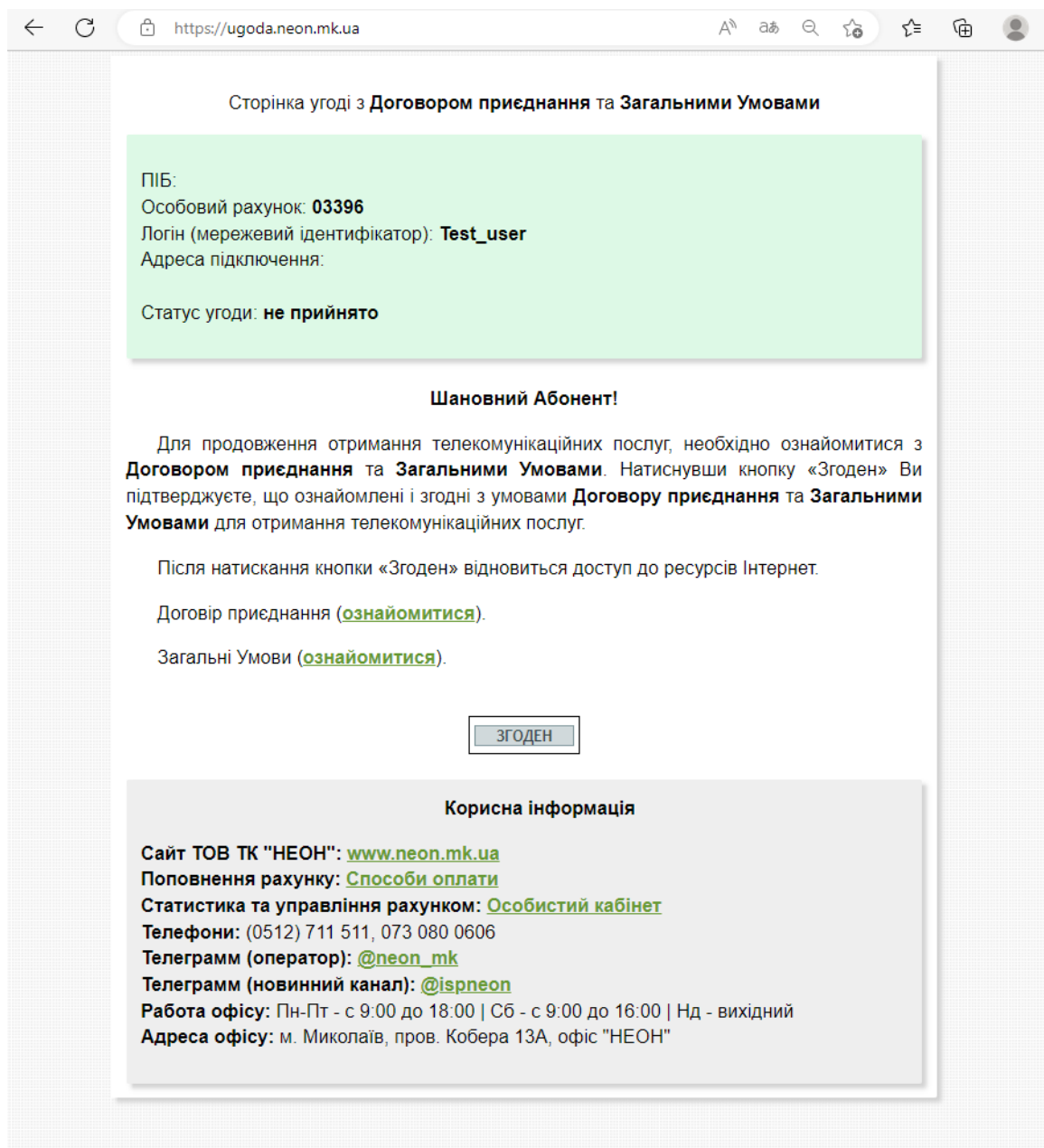


Рисунок 3.5 – Сторінка Угоди ТОВ ТК «НЕОН»

Для реалізації зазначеної перевірки було розроблено скрипт **check\_agreement.py**. Запуск цього скрипту прописано у файлі `ip-up.local`. Скрипти файлу `ip-up.local` виконуються автоматично на NAS-сервері відразу після створення PPPoE-з'єднання. У файл `ip-up.local` було створено процедуру запуску скрипта `check_agreement.py` (ліст. 3.6).

### Лістинг 3.6 – Команди запуску скрипта `check_agreement.py`

```
iptables-save | grep -v "${PEERNAME}" | iptables-restore
/root/bin/check_agreement.py --add $PEERNAME $PPP_REMOTE $PPP_IFACE
```

Алгоритм роботи скрипту наведено на рис. 3.6. Тобто після успішної авторизації користувача на NAS-сервері та створення PPPoE-з'єднання запускається скрипт `check_agreement.py` з файлу `ip-up.local`. У процесі запуску скрипта йому в якості параметрів передаються такі аргументи:

- `$PEERNAME` – ім'я користувача;
- `$PPP_REMOTE` – IP-адрес, призначений користувачеві;
- `$PPP_IFACE` – ім'я інтерфейсу.

Ці параметри необхідні для організації процедури перевірки прийняття Угоди абонентом. Скрипт `check_agreement.py` з отриманими значеннями параметрів виконує такі дії:

1. Перевіряє в локальній БД, чи прийнято Угоду.
2. Якщо Угоду прийнято, то робота скрипта завершується.
3. Якщо ні, то перевіряє на сервері (в головній БД), чи прийнята Угода.
4. Якщо ні, то робота скрипта завершується.
5. Якщо в головній БД Угоду прийнято, то додає в локальну БД відповідний запис і робота скрипта завершується.
6. Якщо й у головній БД Угоду не прийнято, то створює обмежувальні правила Firewall такі, щоб трафік користувача перенаправлявся на вебсторінку прийняття Угоди (рис. 3.5), де користувач може її або прийняти, або відхилити, робота скрипта завершується.

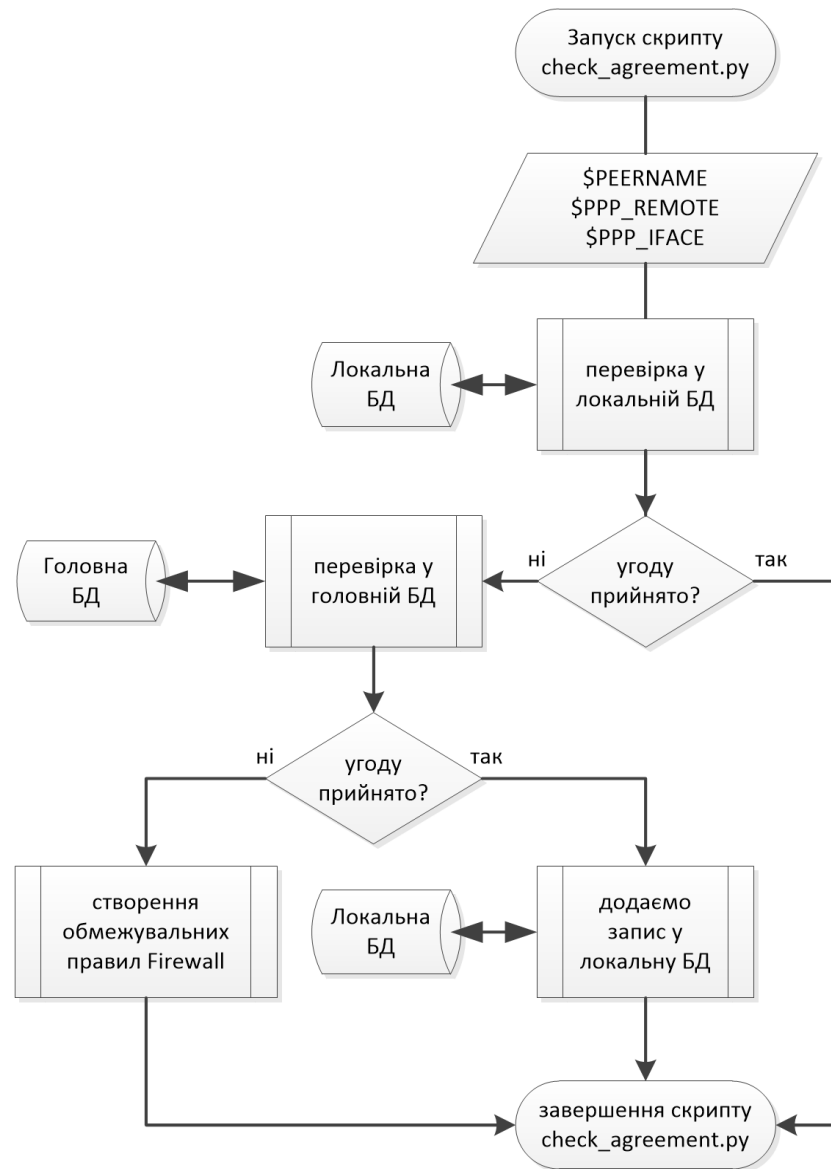


Рисунок 3.6 – Блок-схема алгоритму роботи скрипту check\_agreement.py

Локальну БД розміщено на NAS-сервері та призначено для кешування головної БД, розміщеної на сервері БД. Локальна БД містить дублюючі дані про прийняття Угоди користувачами (ці дані також збережено в головній БД). З огляду на те, що здебільшого абоненти Угоду прийняли, то майже всі запити процесу перевірки прийняття Угоди виконуються на одній фізичній машині, що не потребує мережевої взаємодії. Така організація запитів пришвидшує весь процес перевірки та дає змогу розподілити робоче навантаження запиту із головної бази на інші.

### 3.5 Схеми бази даних

Для збереження інформації було створено БД, яка використовує реляційну модель даних та СКБД MySQL. У процесі аналізу предметної галузі було виокремлено три сутності: користувачі (абоненти), пакети (тарифні плани), сеанси зв'язку (підключення до Інтернету, користуванням ресурсами Інтернету). Цим сутностям відповідають таблиці: users, packets, actions:

- users – містяться дані про абонентів: персональні інформація (ПІБ, телефон, адрес, пароль тощо); платіжна інформація (стан рахунку, метод оплати тощо); інформація щодо обраного тарифного плану (пакету); службова інформація тощо;
- packets – зберігається інформація про тарифні плани (пакети) оператора телекомунікацій;
- actions – зберігається інформація о сесіях користувачів, така як час підключення, час відключення, тривалість сесії, причина відключення, скільки отримано й передано даних, IP-адрес і порт сервера доступу. У цю ж таблицю записуються дані про поповнення і зняття грошей з особистого рахунку абонента.

Для виокремлених сутностей було складено набір їхніх властивостей, що використовуються в бізнес-процесах оператора. На основі цього побудовано ER-діаграму БД (рис. 3.7). На діаграмі первинні ключі мають позначку жовтого ключа. Прозорим ромбом позначено поля не обов'язкові для заповнення, блакитним ромбом – поля обов'язкові для заповнення, червоним ромбом – поля, що є зовнішніми ключами. На рис. 3.8-3.9 наведено типи даних, розмір та інші властивості полів окремо для кожної таблиці.

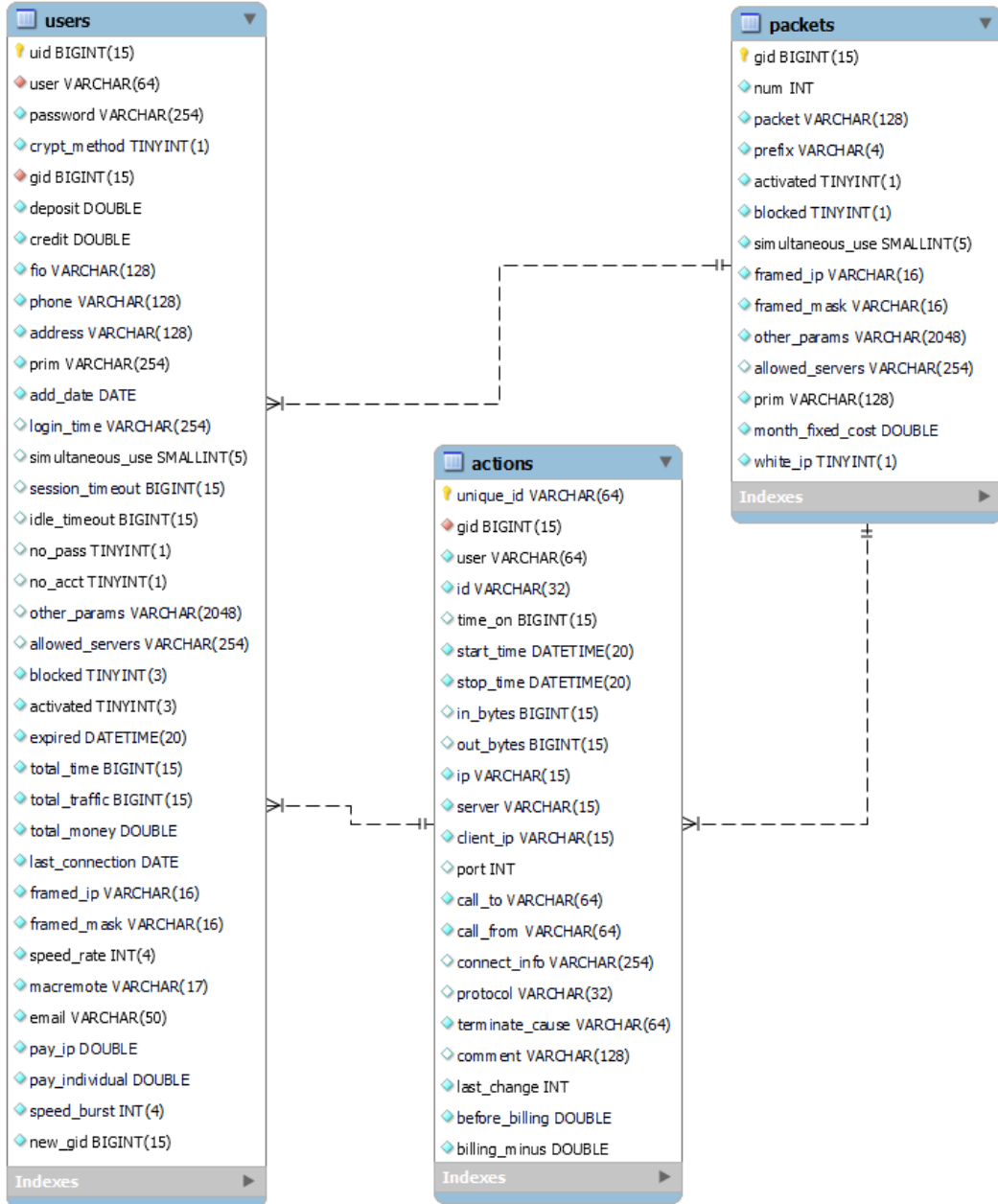


Рисунок 3.7 – ER діаграма бази даних

Table Name: packets Schema: mydb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
gid	BIGINT(15)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
num	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
packet	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
prefix	VARCHAR(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
activated	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
blocked	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
simultaneous_use	SMALLINT(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
framed_ip	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
framed_mask	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
other_params	VARCHAR(2048)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'Acct-Interim-Interval=60,S...
allowed_servers	VARCHAR(254)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
prim	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
month_fixed_cost	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
white_ip	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Рисунок 3.8 – Властивості полів таблиці packets



Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
uid	BIGINT(15)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
user	VARCHAR(64)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
password	VARCHAR(254)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
crypt_method	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
gid	BIGINT(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
deposit	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
credit	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
fio	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
phone	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
address	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
prim	VARCHAR(254)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
add_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0000-00-00
login_time	VARCHAR(254)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
simultaneous_use	SMALLINT(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
session_timeout	BIGINT(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
idle_timeout	BIGINT(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
no_pass	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
no_acct	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
other_params	VARCHAR(2048)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
allowed_servers	VARCHAR(254)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
blocked	TINYINT(3)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
activated	TINYINT(3)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
expired	DATETIME(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0000-00-00 00:00:00
total_time	BIGINT(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
total_traffic	BIGINT(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
total_money	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
last_connection	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0000-00-00
framed_ip	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
framed_mask	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
speed_rate	INT(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
macremote	VARCHAR(17)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
email	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
pay_ip	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
pay_individual	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
speed_burst	INT(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
new_gid	BIGINT(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Рисунок 3.9 – Властивості полів таблиці users

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
unique_id	VARCHAR(64)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
gid	BIGINT(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
user	VARCHAR(64)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
time_on	BIGINT(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
start_time	DATETIME(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2003-01-01 00:00:00
stop_time	DATETIME(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0000-00-00 00:00:00
in_bytes	BIGINT(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
out_bytes	BIGINT(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ip	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
server	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
client_ip	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
port	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
call_to	VARCHAR(64)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
call_from	VARCHAR(64)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
connect_info	VARCHAR(254)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
protocol	VARCHAR(32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
terminate_cause	VARCHAR(64)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
comment	VARCHAR(128)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
last_change	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
before_billing	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000
billing_minus	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.000000

Рисунок 3.10 – Властивості полів таблиці actions

Між таблицями встановлено такі зв'язки:

- таблиця users зв'язана з packets за ключем gid;
- таблиця actions зв'язана з packets за ключем gid;

– таблиця actions зв'язана з users за ключем user.

Тобто абонент (users) обирає тарифний план, за яким він буде отримувати послугу доступу до Інтернету. Обраний тарифний план зберігається в таблиці users. У таблиці action зберігаються дані за сеансом зв'язку та тарифікацією за обраним тарифним планом абонента. Надмірний зв'язок між таблицями actions та packets необхідний для збереження історії сеансів зв'язку абонента, оскільки в таблиці users зберігається тільки поточний тарифний план абонента.

### **Висновки до розділу 3**

У процесі реалізації системи АСР оператора телекомунікацій було проаналізовано середовище функціонування системи та інструментарій її розроблення. Отже, з огляду на функціональні можливості найбільш популярних серверних ОС та вимоги замовника, було обрано дистрибутив Linux – Ubuntu.

Для реалізації логіки взаємодії клієнта із системою АСР було обрано у якості RADIUS-сервера – FreeRADIUS. Розроблено модуль розширення RADIUS-сервера, що виконує логіку процесу AAA (п. 2.2) та скрипт, що реалізує процедуру перевірки прийняття Угоди. Модуль розширення RADIUS-сервера написано на мові С, скрипт перевірки прийняття Угоди – на мові Python.

Для збереження інформації, що використовується в системі АСР було створено БД, яка використовує реляційну модель даних та СКБД MySQL. БД відповідає предметній галузі та бізнес-процесам оператора. Для процесу перевірки прийняття Угоди організовано кешування головної БД із використанням локальної БД розміщеної на NAS-сервері Така організація запитів дає змогу пришвидшити весь процес перевірки та дає змогу розподілити робоче навантаження запиту із головної бази на інші.

## 4 НАЛАГОДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

### 4.1 Налagodження системи

Для налагодження роботи, пошуку і виправлення помилок сервер FreeRADIUS необхідно запустити з ключами `-Xx`. За довідковою інформацією FreeRADIUS Ключі мають таке призначення:

- `-x` – режим налагодження. У цьому режимі сервер друкує детальну інформацію про кожен запит на виході `stderr`. Найбільш корисний у поєднанні з ключем `-s`. Ви можете вказати цей параметр 2 рази (`-x -x` або `-xx`), щоб отримати більше результатів налагодження;
- `-X` – розширений режим налагодження, еквівалентно `-sfx`.

Проте, перед тим у процесі написання коду, якщо інформація потрібна для процесу налагодження, її необхідно позначити для додавання в логфайл. Для цього призначено інструкцію `DEBUG` і обробник виключень. Так, наприклад, у такому фрагменті коду:

```
if (ret) {
    DEBUG("rlm_isc (sql_xlat): SQL query did not succeed");
    (inst->module->als_finish_select_query)(sqlsocket, inst->config);
    als_release_socket(inst, sqlsocket);
...

```

інструкція `DEBUG` виведе в логфайл повідомлення `"rlm_isc (sql_xlat): SQL query did not succeed"` якщо змінна `ret` не 0.

Отже, у консолі виконуємо команду:

```
radiusd -Xx
```

У результаті ми отримуємо в консолі інформацію про налагодження, яку зручно зберігати у файл:

```
Sat Jan 14 14:33:51 2023 : Info: Starting - reading configuration files ...
Sat Jan 14 14:33:51 2023 : Debug: reread_config: reading radiusd.conf
Sat Jan 14 14:33:51 2023 : Debug: Config: including file:
/usr/local/freeradius/etc/raddb/proxy.conf
Sat Jan 14 14:33:51 2023 : Debug: Config: including file:
/usr/local/freeradius/etc/raddb/clients.conf
...
Sat Jan 14 14:33:51 2023 : Debug: Module: Instantiated sql (sql)
Sat Jan 14 14:33:51 2023 : Debug: Listening on authentication *:1812
Sat Jan 14 14:33:51 2023 : Debug: Listening on accounting *:1813
Sat Jan 14 14:33:51 2023 : Info: Ready to process requests.
```

Зокрема, останній рядок інформує о готовності системи до роботи, тобто система очікує запити.

Логфайл загалом містить майже 3500 рядків. З них приблизно 600 рядків містять інформацію щодо запуску модуля rlm\_isp. Нижче наведено фрагмент логфайлу:

```
Sat Jan 14 14:33:51 2023 : Debug: Config: including file:
/usr/local/freeradius/etc/raddb/isp.conf
Sat Jan 14 14:33:51 2023 : Debug: Module: Loaded ISP
Sat Jan 14 14:33:51 2023 : Debug: isp: driver = "rlm_isp_mysql"
Sat Jan 14 14:33:51 2023 : Debug: isp: server = "localhost"
Sat Jan 14 14:33:51 2023 : Debug: isp: port = "3306"
Sat Jan 14 14:33:51 2023 : Debug: isp: login = "*****"
Sat Jan 14 14:33:51 2023 : Debug: isp: password = "*****"
Sat Jan 14 14:33:51 2023 : Debug: isp: isp_db = "ispdb"
Sat Jan 14 14:33:51 2023 : Debug: isp: auth_table = "users"
Sat Jan 14 14:33:51 2023 : Debug: isp: packets_table = "packets"
...
Sat Jan 14 14:33:51 2023 : Debug: isp: sqltrace = no
Sat Jan 14 14:33:51 2023 : Debug: isp: sqltracefile =
"/usr/local/freeradius/var/log/radius/isp_mysqltrace.sql"
Sat Jan 14 14:33:51 2023 : Debug: isp: num_socks = 256
Sat Jan 14 14:33:51 2023 : Debug: isp: connect_failure_retry = 100
Sat Jan 14 14:33:51 2023 : Debug: isp: connect_failure_retry_delay = 3
Sat Jan 14 14:33:51 2023 : Debug: isp: sql_user_name = "%{User-Name}"
Sat Jan 14 14:33:51 2023 : Debug: isp: group_select_query = "SELECT gid, prefix,
tos, do_with_tos, direction, fixed, fixed_cost, activation_time, total_time_limit,
month_time_limit, week_time_limit, day_time_limit, total_traffic_limit,
month_traffic_limit, week_traffic_limit, day_traffic_limit, total_money_limit,
month_money_limit, week_money_limit, day_money_limit, session_timeout,
idle_timeout, simultaneous_use, port_limit, login_time, other_params,
huntgroup_name, framed_ip, framed_mask, no_acct, no_pass, no_pay_day FROM packets"
Sat Jan 14 14:33:51 2023 : Debug: isp: group_select_query_prof = "SELECT
allowed_servers FROM packets WHERE gid='%d'"
...
Sat Jan 14 14:33:51 2023 : Info: rlm_isp (rlm_isp_instantiate): Driver
rlm_isp_mysql (module rlm_isp_mysql) loaded and linked
Sat Jan 14 14:33:51 2023 : Info: rlm_isp (rlm_isp_instantiate): Attempting to
connect to *****@localhost:3306/ispdb
Sat Jan 14 14:33:51 2023 : Debug: sql_als->connect_single_socket: Attempting to
connect rlm_isp_mysql #0
Sat Jan 14 14:33:51 2023 : Info: rlm_isp_mysql: Starting connect to MySQL server
for #0
Sat Jan 14 14:33:51 2023 : Debug: sql_als->connect_single_socket: Attempting to
connect rlm_isp_mysql #1
Sat Jan 14 14:33:51 2023 : Info: rlm_isp_mysql: Starting connect to MySQL server
for #1
Sat Jan 14 14:33:51 2023 : Debug: sql_als->connect_single_socket: Attempting to
connect rlm_isp_mysql #2
Sat Jan 14 14:33:51 2023 : Info: rlm_isp_mysql: Starting connect to MySQL server
for #2
Sat Jan 14 14:33:51 2023 : Debug: sql_als->connect_single_socket: Attempting to
connect rlm_isp_mysql #3
Sat Jan 14 14:33:51 2023 : Info: rlm_isp_mysql: Starting connect to MySQL server
for #3
...
```

```
Sat Jan 14 14:33:51 2023 : Info: rlm_ismysql: Starting connect to MySQL server
for #255
Sat Jan 14 14:33:51 2023 : Info: rlm_ism (ism_init): Initializing main structures
`ism'
...
Sat Jan 14 14:33:51 2023 : Debug: Module: Instantiated ism (ism)
Sat Jan 14 14:33:51 2023 : Debug:  sql: login = "*****"
Sat Jan 14 14:33:51 2023 : Debug:  sql: radius_db = "ismdb"
Sat Jan 14 14:33:51 2023 : Info: rlm_sql (sql): Attempting to connect to
*****@localhost:/ismdb
```

У першому рядку міститься інформація про файл конфігурації `ism.conf` модуля `rlm_ism`. Зміст файлу `ism.conf` наведено в додат. А. Із логів можна побачити, що конфігурацію одержано правильно. Модуль успішно підключено до БД. Структуру модуля `rlm_ism` успішно ініціалізовано. Помилки не виявлено.

## 4.2 Тестування системи

Тестування програмного забезпечення – це процес технічного дослідження, призначений для виявлення інформації про якість продукту щодо контексту, у якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових із метою оцінювання. У процесі тестування може оцінюватись:

- відповідність вимогам, якими керувалися проєктувальники та розробники;
- правильна відповідь для всіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність із програмним забезпеченням та ОС;
- відповідність завданням замовника.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, тому стратегія тестування полягає в тому, щоби провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат програмне забезпечення тестується стандартним виконанням програми щоби виявити «баги» (помилки або інші дефекти).

Тестування може проводитись, як тільки створено виконуваний код (навіть частково завершено). Методологія розробки програмного забезпечення зазвичай передбачає, коли та як відбуватиметься тестування. Наприклад, у разі поетапного процесу здебільшого тестування відбувається після визначення системних вимог і тоді воно реалізується в тестових програмах. На противагу цьому, відповідно до вимог гнучкої розробки ПЗ, програмування і тестування часто відбувається одночасно.

Тестування програмного забезпечення – це техніка контролю якості, що перевіряє відповідність між реальною і очікуваною поведінкою програми завдяки кінцевому набору тестів, які обираються певним чином. Водночас треба розрізняти тестування програмного забезпечення і забезпечення якості програмного забезпечення, до якого належать усі складові цього бізнес-процесу, а не тільки тестування. Зазвичай, поняття якості обмежується такими поняттями як коректність, надійність, практичність, безпечність, але може містити більше технічних вимог, котрі описані в стандарті ISO 9126. Склад та зміст супутньої документації процесу тестування визначається стандартом IEEE 829-1998 Standard for Software Test Documentation. Є багато підходів до тестування програмного забезпечення, але ефективне тестування складних продуктів – це за суттю дослідницький та творчий процес, а не тільки створення та виконання рутинної процедури.

Отже, ґрунтуючись на специфікації вимог та діаграмі варіантів використання система АСР для перевірки правильності її функціонування було створено тестового користувача Test\_user та тестовий тарифний план із такими налаштуваннями:

```
INSERT INTO users (user, password, crypt_method, deposit, gid, add_date) VALUES ('Test_user', '*****', 0, 0.1, 48, NOW());
```

```
mysql> INSERT INTO packets (num, packet, gid, other_params, prim, month_fixed_cost, white_ip) VALUES (0, 'Тестовый', 1, 'Acct-Interim-Interval=60, Service-Type=Framed-User, Framed-Protocol=PPP, Framed-Compression=Van-Jacobson-TCP-IP, PPPD-Downstream-Speed-Limit=10240, PPPD-Upstream-Speed-Limit=10240', 'Пакет для тестов', 50.000000, 0);
```

В останньому запиті:

- поле `other_params` містить пари: атрибут та його значення розділені комами. Ці параметри надалі будуть передаватися серверу доступу модулем RADIUS-сервера. У такий спосіб ми, наприклад, будемо встановлювати `Acct-Interim-Interval` – періодичність у секундах, із якою сервер доступу буде надсилати RADIUS-серверу інформацію про сесії користувача, скільки було передано або прийнято даних, тривалість сесії тощо;
- з допомогою атрибутів `PPPD-Downstream-Speed-Limit` і `PPPD-Upstream-Speed-Limit` ми передаємо значення максимальної швидкості в кілобайтах для цієї сесії.

Коли надходить запит на підключення (`Access-Request`) від клієнта в логфайлі можна побачити його коректне оброблення модулем `rlm_isc` (рис. 3.11-3.12).

```
Sat Jan 14 15:12:30 2023 : Debug: Listening on authentication *:1812
Sat Jan 14 15:12:30 2023 : Debug: Listening on accounting *:1813
Sat Jan 14 15:12:30 2023 : Info: Ready to process requests.
rad recv: Access-Request packet from host 93.171.170.240:34689, id=1, length=153
User-Name = "Test user"
NAS-Identifier = "accel-ppp"
NAS-IP-Address = 93.171.170.240
NAS-Port = 0
NAS-Port-Id = "ppp0"
NAS-Port-Type = Virtual
Service-Type = Framed-User
Framed-Protocol = PPP
Calling-Station-Id = "52:54:00:30:d4:a2"
Called-Station-Id = "52:54:00:c9:85:8c"
CHAP-Challenge = 0xc69858ad5e47923c55371b4c01508fc2
CHAP-Password = 0x01d4a62371c6ddal6a8cbd53ce75aacde4
Sat Jan 14 15:12:57 2023 : Debug: Processing the authorize section of radiusd.conf
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: entering group authorize for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: calling preprocess (rlm_preprocess) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: returned from preprocess (rlm_preprocess) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: module "preprocess" returns ok for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: calling chap (rlm_chap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: rlm_chap: Setting 'Auth-Type := CHAP'
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: returned from chap (rlm_chap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: module "chap" returns ok for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: calling mschap (rlm_mschap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: returned from mschap (rlm_mschap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: module "mschap" returns noop for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: calling files (rlm_files) for request 0
Sat Jan 14 15:12:57 2023 : Debug: users: Matched entry DEFAULT at line 215
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: returned from files (rlm_files) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: module "files" returns ok for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: calling isp (rlm_isc) for request 0
Sat Jan 14 15:12:57 2023 : Debug: radius_xlat: 'Test user'
Sat Jan 14 15:12:57 2023 : Debug: rlm_isc (sql_set_user): sql_set_user escaped user --> 'Test user'
Sat Jan 14 15:12:57 2023 : Debug: rlm_isc (isp_fill_user): begin for user 'Test user' -----
```

Рисунок 3.11 – Запит на підключення `Access-Request` абонента `Test_user`



```

Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_fill_user): ----- prof mode begin for user 'Test_user' -----
Sat Jan 14 15:12:57 2023 : Debug: sql_als->sql_get_socket (isp): Reserving sql socket id: 226
Sat Jan 14 15:12:57 2023 : Debug: sql_als->sql_release_socket: Released sql socket id: 226
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_fill_user): ----- prof mode end for user 'Test_user' -----
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_fill_user): end for user 'Test_user' -----
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): begin for user 'Test_user' -----
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): add PW_PASSWORD
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): add PW_SIMULTANEOUS_USE
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): add PW_SESSION_TIMEOUT
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): add all other params
Sat Jan 14 15:12:57 2023 : Debug: rlm_osp (isp_add_attrs): end for user 'Test_user' -----
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authorize]: returned from isp (rlm_osp) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authorize]: module "isp" returns noop for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall: leaving group authorize (returns ok) for request 0
Sat Jan 14 15:12:57 2023 : Debug: rad_check_password: Found Auth-Type CHAP
Sat Jan 14 15:12:57 2023 : Debug: auth: type "CHAP"
Sat Jan 14 15:12:57 2023 : Debug: Processing the authenticate section of radiusd.conf
Sat Jan 14 15:12:57 2023 : Debug: modcall: entering group CHAP for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authenticate]: calling chap (rlm_chap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: rlm_chap: login attempt by "Test_user" with CHAP password
Sat Jan 14 15:12:57 2023 : Debug: rlm_chap: Using clear text password " " for user Test_user authentication.
Sat Jan 14 15:12:57 2023 : Debug: rlm_chap: chap user Test_user authenticated successfully
Sat Jan 14 15:12:57 2023 : Debug: modsingle[authenticate]: returned from chap (rlm_chap) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[authenticate]: module "chap" returns ok for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall: leaving group CHAP (returns ok) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall: Processing the session section of radiusd.conf
Sat Jan 14 15:12:57 2023 : Debug: modcall: entering group session for request 0
Sat Jan 14 15:12:57 2023 : Debug: modsingle[session]: calling sql (rlm_sql) for request 0
Sat Jan 14 15:12:57 2023 : Debug: radius_xlat: 'Test_user'
Sat Jan 14 15:12:57 2023 : Debug: rlm_sql (sql): sql_set_user escaped user --> 'Test_user'
Sat Jan 14 15:12:57 2023 : Debug: radius_xlat: 'SELECT COUNT(*) FROM actions WHERE user = 'Test_user' AND terminate_cause = 'Online''
Sat Jan 14 15:12:57 2023 : Debug: rlm_sql (sql): Reserving sql socket id: 255
Sat Jan 14 15:12:57 2023 : Debug: rlm_sql (sql): Released sql socket id: 255
Sat Jan 14 15:12:57 2023 : Debug: modsingle[session]: returned from sql (rlm_sql) for request 0
Sat Jan 14 15:12:57 2023 : Debug: modcall[session]: module "sql" returns ok for request 0

```

Рисунок 3.12 – Оброблення модулем rlm\_osp запиту Access-Request абонента Test\_user

Також у процесі тестування було помічено затримку під час формування вебсторінки моніторингу всіх активних сесій абонентів (на головній сторінці порталу адміністратора посилання Online (рис. 2.4)). Цей моніторинг був однією з головних вимог із боку замовника щодо функціоналу системи АСР. Аналіз показав, що проблема спричинена неоптимальним запитом до БД. Метод explain дає змогу отримати план виконання запиту та проаналізувати його (рис. 3.13). Характеристики запиту показують, що індекси не використовуються (key = NULL) і скануються всі записи таблиці actions (row = 488440).

```

mysql> EXPLAIN SELECT billing_minus, user, port, server, ip,
call_from, date_format(start_time, '%d %b, %H:%i:%s') as start_time, time_on,
connect_info, in_bytes, out_bytes FROM actions WHERE terminate_cause='Online'
order BY server, user;

```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	actions	ALL	NULL	NULL	NULL	NULL	488440	Using where; Using filesort

1 row in set (0,00 sec)

Рисунок 3.13 – План виконання запиту для формування списку всіх активних сесій абонентів



Для оптимізації запиту було створено простий індекс за полем `terminate_cause`:

```
mysql> ALTER TABLE actions ADD INDEX (terminate_cause);
```

```
Query OK, 488440 rows affected (5,49 sec)
Records: 488440 Duplicates: 0 Warnings: 0
```

І знов виконано відладку (рис. 3.14).

```
mysql> EXPLAIN SELECT billing_minus, user, port, server, ip,
call_from, date_format(start_time, '%d %b, %H:%i:%s') as start_time, time_on,
connect_info, in_bytes, out_bytes FROM actions WHERE terminate_cause='Online'
order BY server, user;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	actions	ref	terminate_cause	terminate_cause	66	const	1651	Using index condition; Using where; Using filesort

1 row in set (0,00 sec)

Рисунок 3.14 – План виконання запиту з використанням індексу для формування списку всіх активних сесій абонентів

Характеристики запиту показують, що зараз використовується індекс (`key = terminate_cause`) і сканується тільки частина записів таблиці `actions` (`row = 1651`). Кількість записів сканування зменшилася майже на 99%, що дало змогу значно пришвидшити відображення на вебсторінці моніторингу всі активні сесії абонентів.

Отже, у процесі тестування було перевірено правильність виконання процедур ААА, оптимізовано запит до БД. Треба зауважити, що якість не є абсолютною, це суб'єктивне поняття. Тому тестування, як процес своєчасного виявлення помилок та дефектів, не може повністю забезпечити коректність роботи програмної системи. Воно тільки порівнює стан і поведінку продукту зі специфікацією.

## **Висновки до розділу 4**

У процесі налагодження роботи, пошуку і виправлення помилок системи АСР були виконані такі заходи:

- створено тестового користувача Test\_user та тестовий тарифний план;
- виконано налагодження роботи серверу FreeRADIUS з використанням спеціального режиму налагодження ОС Ubuntu;
- у режимі налагодження було зібрано та проаналізовано системні логи виконання процесів старту системи АСР та авторизації тестового користувача;
- виконано аудит запитів, виявлено повільні запити та виконано їхню оптимізацію.

Аналіз показав, що в процесі старту система AAA отримує конфігурацію правильно, модуль успішно підключається до БД, структуру модуля rlm\_isp успішно ініціалізовано, помилки не виникають. У процесі авторизації тестового користувача створюється PPPoE-сесія з відповідними налаштуваннями. Помилки також не виникають.

## 5 ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ В СКЛАДНИХ УМОВАХ ЕКСПЛУАТАЦІЇ

Оскільки в процесі тестування та налагодження системи АСР неможливо виявити проблеми пов'язані зі складними умовами експлуатації, то було прийнято рішення провести відповідні дослідження. Нині в місті, де експлуатується система АСР є плановий графік відключення електроенергії. Це створює складні умови експлуатації. Велика частина абонентів одночасно відключається та припиняється надходження потоку службових даних від NAS-серверів. У разі відновлення постачання електроенергії в абонентів, відбувається масова одночасна авторизація їхнього обладнання, а отже зростає навантаження на CPU RADIUS-серверу. Такі умови потребують аналізу працездатності системи в критичні моменти (масове зникнення клієнтів та масова авторизація).

У процесі аналізу також необхідно урахувати апаратну конфігурацію, на якій розгорнута система АСР. Для збору та представлення даних для аналізу використовувалася система Cacti. На офіційному сайті зазначено, що Cacti – це система оперативного моніторингу й управління відмовами. Система є комплексним рішенням для побудови мережевих графіків із допомогою RRDTool [30]. Cacti збирає системні статистичні дані за певні часові інтервали і представляє їх у графічному вигляді.

Отже, як зазначено на діаграмі розгортання (п. 2.3), розгортання системи було здійснено на апаратних засобах: сервері віртуальних машин та NAS-серверах. Комутатором ядра мережі є CISCO NEXUS N9-c9396px (48x1/10G SFP+, 12X40G QSFP), який на поточний момент часу забезпечує необхідний функціонал, швидкість передачі даних і швидкість комутації пакетів.

Сервер віртуальних машин:

Сервер: Dell Precision Rack 7910

CPU: Intel E5-2660V3 2.60GHz 10 Core (2шт)

Пам'ять: 8GB PC4-17000 (DDR4-2133P) RDIMM ECC (8шт)

SSD-SATA: Dell 240GB M.2 SATA 6Gb/s Micron 5100 Pro (5шт)

SSD-PCIЕ: Intel P3500 Series 400GB PCIe 3.0 SSD (1шт)

LAN: HP Ethernet 10GB 2-Port 560SFP+ Server Adapter

Такої конфігурації достатньо для забезпечення роботи 8 віртуальних машин як в робочій конфігурації, так і для тестових конфігурацій і систем.

Під віртуальну машину сервера АСР виділено 8 ядер CPU, 24GB ОЗП, 400GB високопродуктивного сховища даних для БД. На рис. 5.1 показано графік завантаження процесорів (вибірково) і load average (середнє значення завантаження системи за певний період часу) за один місяць, на рис. 5.2. – за один день.

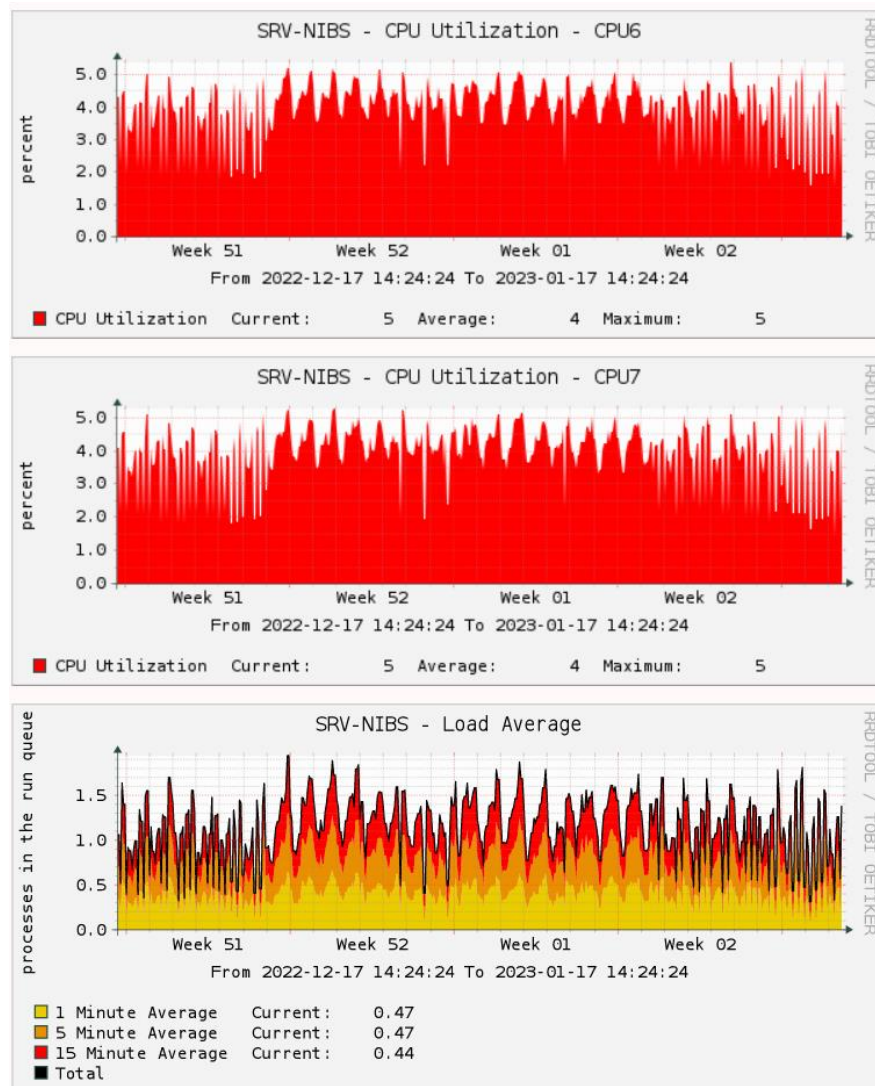


Рисунок 5.1 – Середні значення завантаження системи впродовж одного місяця

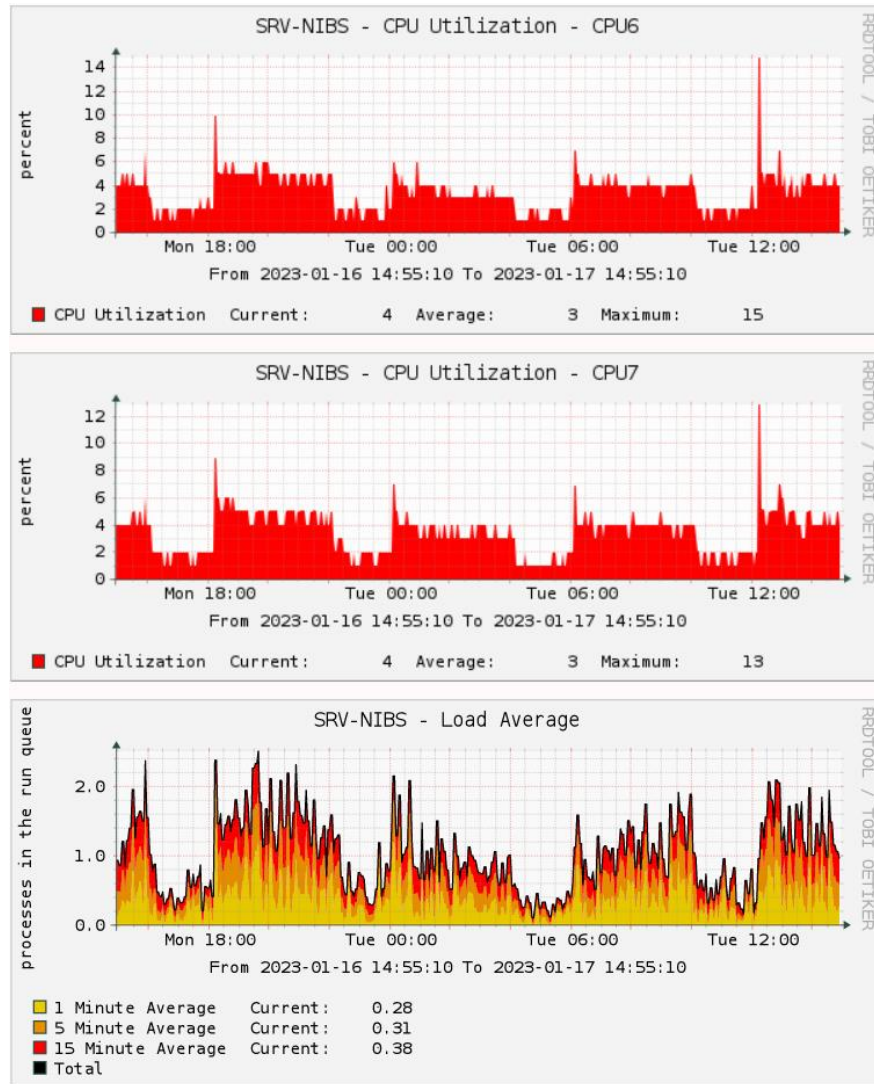


Рисунок 5.2 – Середні значення завантаження системи впродовж одного дня

Як видно з графіків, середньомісячне завантаження процесорів становить 5...10%. На денному графіку чітко видно просадки й різкий сплеск завантаження CPU RADIUS-серверу. Це зумовлено тим, що в цей час у місті діє плановий графік відключення електроенергії, і частина абонентів відключається. Отже, припиняється надходження потоку службових даних від NAS-серверів. У разі відновлення постачання електроенергії абонентам, відбувається масова авторизація їхнього обладнання і зростання навантаження на CPU RADIUS-серверу. З графіку видно, що цей сплеск не перевищує 15% і триває не більше за 5 хвилин. Це говорить про те, що RADIUS-сервер встигає авторизувати всіх клієнтів упродовж необхідного проміжку часу (до 10 хвилин за специфікацією вимог до АСР) від початку масової авторизації абонентів.

Завантаження CPU на 5...10% є також позитивним результатом щодо економії електроенергії. Оскільки впродовж відключень електропостачання обладнання живиться від джерела безперебійного живлення і зниження навантаження є вельми актуальним.

NAS-сервери використовуються в бездисковому виконанні з відключеним режимом Hyper Threading.

NAS1-сервер: DELL PowerEdge R330

CPU: Intel Xeon E3-1240 v5 3.5GHz 4 Core (1шт)

Пам'ять: 8GB DDR4-2400 UDIMM (2шт)

LAN: HP Ethernet 10GB 2-Port 560SFP+ Server Adapter

NAS2-сервер: Dell Precision 3430

CPU: Intel Xeon E-2146G 3.5GHz 6 Core (1шт)

Пам'ять: 8GB DDR4-2666 UDIMM (2шт)

LAN: HP Ethernet 10GB 2-Port 560SFP+ Server Adapter

На всіх серверах встановлені серверні мережеві карти з максимальною швидкістю передачі даних 10Gbps. Кожен NAS-сервер призначено для обслуговування приблизно 2000 PPPoE-з'єднань і в разі відмови одного із серверів має можливість резервування один одного.

На рис. 5.3-5.4 наведено добові графіки завантаження CPU, load average і швидкість мережевого потоку даних для обох NAS-серверів. Як видно із цих діаграм, для NAS2-серверу загалом CPU є менш завантаженим та load average показує середнє значення завантаження системи менше ніж для NAS1-серверу. Це зумовлено тим, що на NAS2-сервері використовується більш новий процесор із більш високою обчислювальною продуктивністю, ніж на NAS1-сервері.



Кафедра інженерії програмного забезпечення  
Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

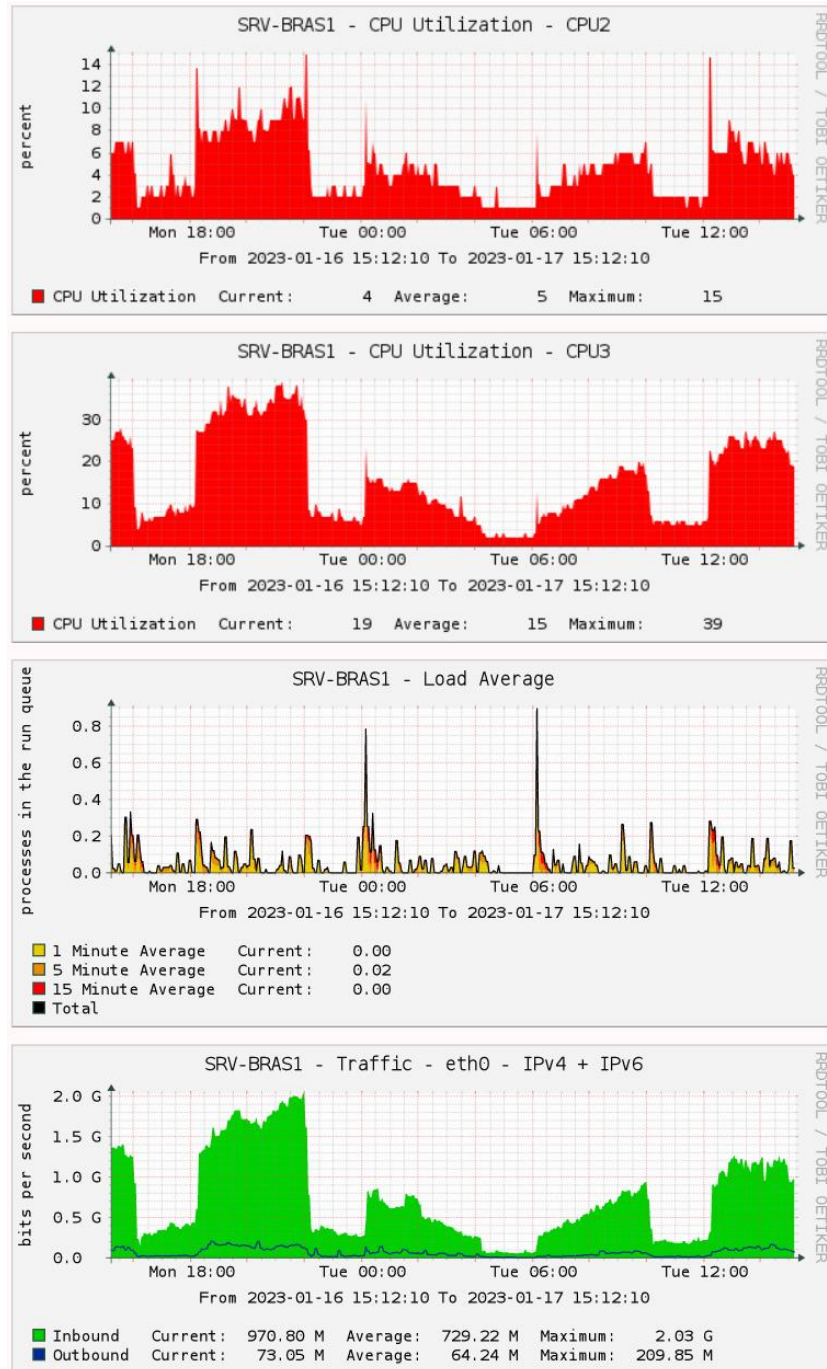


Рисунок 5.3 – Добовий графік завантаження CPU, load average і швидкість мережевого потоку даних NAS1-серверу

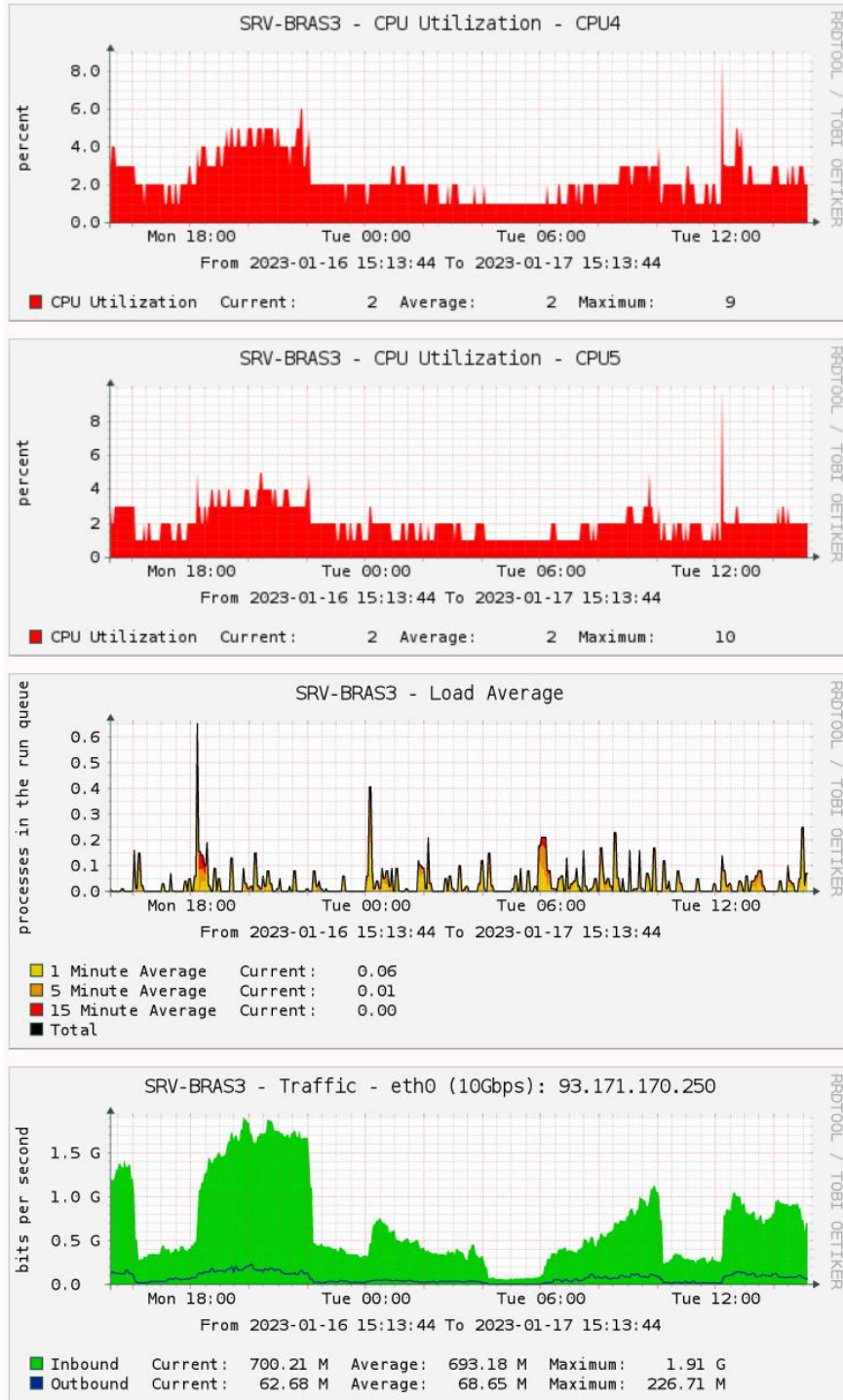


Рисунок 5.4 – Добовий графік завантаження CPU, load average і швидкість мережевого потоку даних NAS2-серверу

Також із графіків добре видно моменти відключення і включення електроживлення в місті, і як це позначається на завантаженні CPU NAS-серверів після відключення живлення (зниження навантаження на CPU), і момент включення електроенергії (короткий сплеск у роботі CPU).



На рис. 5.5. наведено денний графік онлайн-сесій абонентів. На вимогу замовника, кількість онлайн-сесій не наводиться.

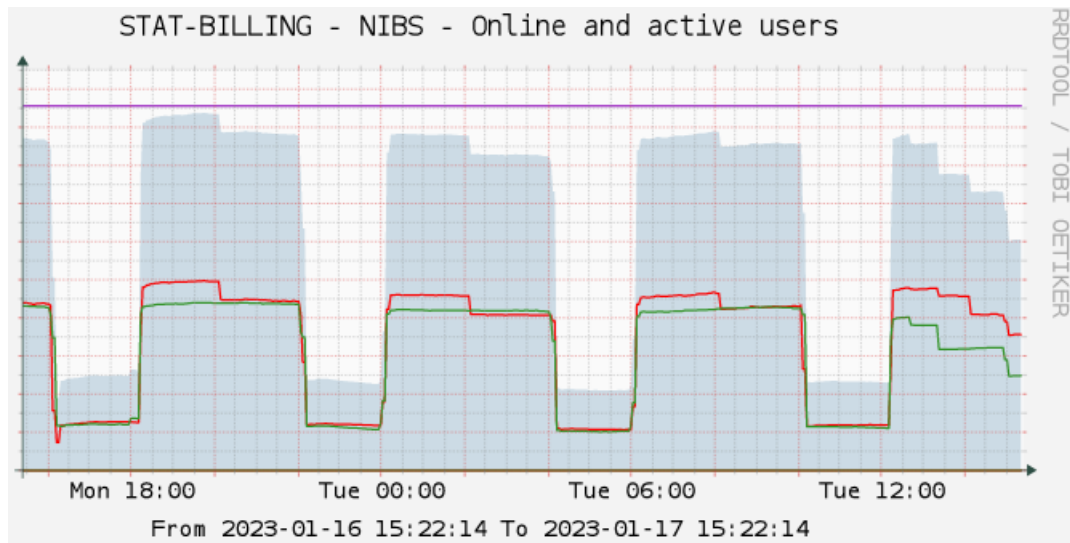


Рисунок 5.5 – Онлайн-сесії абонентів

На графіку чітко видно час відключення і підключення абонентів у разі припинення і відновлення електропостачання. Майже вертикальний характер зміни кількості PPPoE-сесій говорить про те, що всі сервери справляються з авторизацією всіх абонентів за відведений час (до 10 хвилин за специфікацією вимог до АСР).

Отже, можна сказати, що за наявною конфігурацією програмно-апаратного забезпечення система АСР у складних умовах експлуатації здатна впоратися з навантаженнями за відведений проміжок часу.

## Висновки до розділу 5

Проведене дослідження дало можливість перевірити працездатність системи АСР у складних умовах експлуатації. У процесі масового відключення та підключення абонентів, викликаного припиненням та відновленням електропостачання в місті, система здатна впоратися з навантаженнями за відведений проміжок часу.

## **ВИСНОВКИ**

Темою КРМ було обрано дослідження методів контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій. Аналіз предметної галузі дав можливість зробити висновки щодо актуальності зазначеної теми та виявити певні невирішені завдання. Зокрема, це створення програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій, що буде відповідати сучасному розвитку засобів телекомунікацій та задовольняти технічним, користувацьким та законодавчим вимогам.

Для вирішення цього завдання було виконано низку заходів. Проведено аналіз предметної галузі та аналогічних програмних систем зі схожим функціоналом. Виокремлено технології організації доступу користувачів до мережі Інтернет (Ethernet, PPPoE, RADIUS), що дають змогу організувати доступ абонентів в Інтернет у контексті сучасних вимог. Також на основі аналізу предметної галузі, відповідних технологій та дослідження наявних АСР сформовано вимоги щодо програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій.

Обрано та досліджено технології для розроблення програмної системи контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій. Середовищем функціонування системи обрано дистрибутив Linux – Ubuntu. Для реалізації логіки взаємодії клієнта із системою АСР було обрано у якості RADIUS-сервера – FreeRADIUS, мови C та Python.

Програмну систему контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій реалізовано, налагоджено та протестовано. Розроблено модуль розширення RADIUS-сервера, що виконує логіку процесу AAA та скрипт, що реалізує процедуру перевірки прийняття Угоди. Для збереження інформації, що використовується в системі АСР було створено БД, яка використовує реляційну модель даних та СКБД MySQL.

Систему АСР розроблено та впроваджено (Акт № 128 від 23 січня 2023 р. (додат. Д)) на замовлення ТОВ ТК «НЕОН». У процесі розроблення та впровадження були враховані вимоги замовника.

Проведено дослідження роботи системи АСР у складних умовах експлуатації. Складними умовами є те, що в місті, де експлуатується система, є плановий графік відключення електроенергії, що спричиняє в певні проміжки часу її перевантаження. У процесі аналізу враховано апаратну конфігурацію, на якій розгорнута система АСР. Для збору та представлення даних для аналізу використовувалася система Састі.

Інновація системи АСР полягає в розробленні вдосконаленого алгоритму тарифікації пакетів абонентів оператора телекомунікацій, розробленні додаткових можливостей щодо керування доступом абонентами до мережі Інтернет та відмовостійкості системи під час пікових навантажень на сервери обробки даних. Результати цього були показані в процесі тестування та дослідження системи.

Отже, зміст роботи відповідає темі та поставленим задачам. Усі завдання КРМ були виконані, мету досягнуто. Практична складова КРМ – АСР оператора телекомунікацій повністю функціональна й може бути адаптована під різні потреби замовника.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. History of the Internet. *From Wikipedia, the free encyclopedia*: URL: [https://en.wikipedia.org/wiki/History\\_of\\_the\\_Internet](https://en.wikipedia.org/wiki/History_of_the_Internet) (дата звернення: 26.12.2022)
2. Couldry N. *Media, Society, World: Social Theory and Digital Media Practice*. London: Polity Press, 2012. P. 321.
3. Line Mode Browser 2013: вебсайт. URL: <https://line-mode.cern.ch> (дата звернення: 26.12.2022)
4. The World@: вебсайт. URL: <https://std.com> (дата звернення: 26.12.2022)
5. 42 U.S. Code § 1862 - Functions: вебсайт. URL: <https://www.law.cornell.edu/uscode/text/42/1862> (дата звернення 19.05.2022)
6. Hilbert M., López P. The World's Technological Capacity to Store, Communicate, and Compute Information. *Sciens*. 10 Feb 2011. Vol 332, Issue 6025. pp. 60-65. URL: <https://www.science.org/doi/10.1126/science.1200970> (дата звернення 19.05.2022)
7. Internet Systems Consortium. *From Wikipedia, the free encyclopedia*: URL: [https://wiki2.org/en/Internet\\_Systems\\_Consortium](https://wiki2.org/en/Internet_Systems_Consortium) (дата звернення: 26.12.2022)
8. Robertson A. The EU, US, and 32 other countries just announced a 'Declaration for the Future of the Internet'. *The Verge*. Apr 28, 2022. URL: <https://www.theverge.com/2022/4/28/23046652/declaration-for-the-future-of-the-internet-eu-us-regulation-commitment-announced> (дата звернення: 26.12.2022)
9. Про електронні комунікації: Закон України від 16 груд. 2020 р. № 1089-IX. URL: <https://zakon.rada.gov.ua/laws/show/1089-20#top>
10. Data link layer. *From Wikipedia, the free encyclopedia*: URL: [https://translated.turbopages.org/proxy\\_u/en-ru.ru.47df3a72-63a9b3c5-0f1eba64-74722d776562/https/en.wikipedia.org/wiki/Data\\_link\\_layer](https://translated.turbopages.org/proxy_u/en-ru.ru.47df3a72-63a9b3c5-0f1eba64-74722d776562/https/en.wikipedia.org/wiki/Data_link_layer) (дата звернення: 26.12.2022)
11. Мережева модель OSI. *Матеріал з Вікіпедії — вільної енциклопедії*: URL:

[https://uk.wikipedia.org/wiki/Мережева\\_модель\\_OSI#Канальний\\_рівень\\_\(Data\\_Link\\_layer\)](https://uk.wikipedia.org/wiki/Мережева_модель_OSI#Канальний_рівень_(Data_Link_layer)) (дата звернення: 26.12.2022)

12. Ethernet. *From Wikipedia, the free encyclopedia*: URL: <https://en.wikipedia.org/wiki/Ethernet> (дата звернення: 26.12.2022)

13. Point-to-Point Protocol. *From Wikipedia, the free encyclopedia*: URL: [https://en.wikipedia.org/wiki/Point-to-Point\\_Protocol](https://en.wikipedia.org/wiki/Point-to-Point_Protocol) (дата звернення: 26.12.2022)

14. Optical Carrier transmission rates. *From Wikipedia, the free encyclopedia*: URL: [https://translated.turbopages.org/proxy\\_u/en-ru.ru.19c26b17-63aac884-e0faf2ba-74722d776562/https/en.wikipedia.org/wiki/OC1](https://translated.turbopages.org/proxy_u/en-ru.ru.19c26b17-63aac884-e0faf2ba-74722d776562/https/en.wikipedia.org/wiki/OC1) (дата звернення: 26.12.2022)

15. Point-to-Point Protocol over Ethernet. *From Wikipedia, the free encyclopedia*: URL: [https://en.wikipedia.org/wiki/Point-to-Point\\_Protocol\\_over\\_Ethernet](https://en.wikipedia.org/wiki/Point-to-Point_Protocol_over_Ethernet) (дата звернення: 26.12.2022)

16. RFC 2516: A Method for Transmitting PPP Over Ethernet (PPPoE): вебсайт. URL: <https://www.rfc-editor.org/rfc/rfc2516> (дата звернення: 26.12.2022)

17. RADIUS. *From Wikipedia, the free encyclopedia*: URL: <https://en.wikipedia.org/wiki/RADIUS> (дата звернення: 26.12.2022)

18. RADIUS Servers for Noobs: Everything You Need to Know. Cloud RADIUS: вебсайт. URL: <https://www.cloudradius.com/a-complete-guide-to-radius-servers> (дата звернення: 26.12.2022)

19. Hassell J. RADIUS. O'Reilly Media, Inc. 2005. 89 p.

20. Автоматизована система розрахунків. *Матеріал з Вікіпедії – вільної енциклопедії*: URL: [https://uk.wikipedia.org/wiki/Автоматизована\\_система\\_розрахунків](https://uk.wikipedia.org/wiki/Автоматизована_система_розрахунків) (дата звернення: 26.12.2022)

21. Автоматизована система розрахунків. WikiZer: вебсайт. URL: [https://wikizero.com/uk/Автоматизована\\_система\\_розрахунків](https://wikizero.com/uk/Автоматизована_система_розрахунків) (дата звернення 19.05.2022)

22. Rumbaugh J., Jacobson I., Booch G. The unified modeling language reference manual. Addison Wesley Longman Inc. 1999. 537 p.

23. Bennett S., McRobb S., Farmer R. Object Oriented Systems Analysis and Design using UML. CovenGlossary of Key Terms: URL: [https://highered.mheducation.com/sites/0077110005/student\\_view0/glossary.htm](https://highered.mheducation.com/sites/0077110005/student_view0/glossary.htm) (дата звернення 19.05.2022)
24. Інтерфейс користувача. *Матеріал з Вікіпедії – вільної енциклопедії*: URL: [https://uk.wikipedia.org/wiki/Інтерфейс\\_користувача](https://uk.wikipedia.org/wiki/Інтерфейс_користувача) (дата звернення: 06.01.2023)
25. OS Market Share and Usage Trends. W3Cook: вебсайт. URL: <https://web.archive.org/web/20150806094138/http://www.w3cook.com/managed-dns/summary/> (дата звернення 09.01.20223)
26. Usage statistics of operating systems for websites. W<sup>3</sup>Techs: вебсайт. URL: [https://w3techs.com/technologies/overview/operating\\_system](https://w3techs.com/technologies/overview/operating_system) (дата звернення 09.01.20223)
27. C (мова програмування). *Матеріал з Вікіпедії – вільної енциклопедії*: URL: [https://uk.wikipedia.org/wiki/Інтерфейс\\_користувача](https://uk.wikipedia.org/wiki/Інтерфейс_користувача) (дата звернення: 09.01.2023)
28. Python: вебсайт. URL: <https://www.python.org> (дата звернення 09.01.20223)
29. radiusd.h File Reference. FreeRADIUS: вебсайт. URL: [https://doc.freeradius.org/radiusd\\_8h.html#a2cf86a287d2d29505c29aad4883533c9](https://doc.freeradius.org/radiusd_8h.html#a2cf86a287d2d29505c29aad4883533c9) (дата звернення 09.01.20223)
30. About Cacti. Cacti: вебсайт. URL: <https://cacti.net/> (дата звернення: 17.01.2023)

## ДОДАТОК А

### Конфігурація модуля rlm\_isp

```
isp {

    # Database type
    # Current supported are: rlm_isp_mysql, rlm_isp_postgresql,
    driver = "rlm_isp_mysql"

    # SQL Server configuration
    server = "localhost"
    port = "3306"
    login = "mabill"
    password = "xxxxxxxx"

    # Database configuration
    isp_db = "ispdb"

    # SQL tables configuration
    auth_table = "users"
    packets_table = "packets"
    acct_table = "actions"
    # number of sql connections to make to server
    num_socks = 16

    # number of seconds to delay retrying on a failed database
    # connection (per_socket)
    connect_failure_retry = 100
    connect_failure_retry_delay = 3

    sql_user_name = "%{User-Name}"

#####
# Prices select queries
#####
group_select_query = "SELECT gid, prefix, activation_time, session_timeout,
idle_timeout, simultaneous_use, login_time, other_params, huntgroup_name,
framed_ip, framed_mask, no_acct, no_pass FROM ${packets_table}"

#####
# Authenticate Query
#####
authenticate_query = "SELECT user, password, gid, crypt_method FROM ${auth_table}
WHERE user = '%{SQL-User-Name}'"

#####
# Authorization Queries
#####
authorize_query = "SELECT user, password, crypt_method, uid, gid, deposit, credit,
unix_timestamp(add_date), blocked, activated, unix_timestamp(expired), total_time,
total_traffic, total_money, unix_timestamp(last_connection), framed_ip,
framed_mask, callback_number FROM ${auth_table} WHERE user = '%{SQL-User-Name}'"

statistic_select_query = "SELECT time_on, in_bytes, out_bytes,
unix_timestamp(start_time), billing_minus, last_change FROM ${acct_table} WHERE id
= '%{Acct-Session-Id}' AND unique_id = '%{Acct-Unique-Session-Id}' AND server =
'%{NAS-IP-Address}' AND port = %{NAS-Port} AND user = '%{SQL-User-Name}' AND
terminate_cause='Online'"

```

## Кафедра інженерії програмного забезпечення

## Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

```
accounting_start_query = "INSERT INTO ${acct_table} (before_billing, gid, id,
unique_id, user, server, port, client_ip, protocol, ip, call_to, call_from,
terminate_cause, connect_info, start_time, last_change) VALUES ('%16.2f', '%d',
'${Acct-Session-Id}', '${Acct-Unique-Session-Id}', '${SQL-User-Name}', '${NAS-IP-
Address}', ${NAS-Port}, '${Client-IP-Address}', '${Framed-Protocol}', '${Framed-
IP-Address}', '${Called-Station-Id}', '${Calling-Station-Id}', 'Online',
'${Connect-Info}', 'S', unix_timestamp())"
```

```
accounting_start_query_alt = "INSERT INTO ${acct_table} (before_billing, gid,
id, unique_id, user, server, port, client_ip, protocol, ip, call_to, call_from,
terminate_cause, connect_info, start_time, last_change) VALUES ('%16.2f', '%d',
'${Acct-Session-Id}', '${Acct-Unique-Session-Id}', '${SQL-User-Name}', '${NAS-IP-
Address}', ${NAS-Port}, '${Client-IP-Address}', '${Framed-Protocol}', '${Framed-
IP-Address}', '${Called-Station-Id}', '${Calling-Station-Id}', 'Online',
'${Connect-Info}', unix_timestamp() - '%d', unix_timestamp())"
```

```
accounting_reopen = "UPDATE ${acct_table} SET terminate_cause = 'Online',
stop_time = 0, id = '${Acct-Session-Id}', unique_id = '${Acct-Unique-Session-Id}'
WHERE id LIKE '%${Acct-Session-Id}' AND unique_id LIKE '%${Acct-Unique-Session-
Id}' AND server = '${NAS-IP-Address}' AND port = ${NAS-Port} AND user = '${SQL-
User-Name}' AND terminate_cause != 'Online' AND (-10 <=
(unix_timestamp(start_time) - (unix_timestamp() - ${Acct-Session-Time}))) AND
((unix_timestamp(start_time) - (unix_timestamp() - ${Acct-Session-Time})) <= 10)"
```

```
accounting_alive_auth_query = "UPDATE ${auth_table} SET deposit = deposit -
%16.2f, total_traffic = total_traffic + %llu, total_time = total_time + %llu,
total_money = total_money + %16.2f, last_connection = now() WHERE user = '${SQL-
User-Name}'"
```

```
accounting_stop_auth_query = "UPDATE ${auth_table} SET deposit = deposit -
%16.2f, total_traffic = total_traffic + %llu, total_time = total_time + %llu,
total_money = total_money + %16.2f, last_connection = now() WHERE user = '${SQL-
User-Name}'"
```

```
accounting_update_query = "UPDATE ${acct_table} SET time_on = if(${Acct-
Session-Time},${Acct-Session-Time},time_on), in_bytes = %llu, out_bytes = %llu,
last_change = unix_timestamp(), billing_minus = billing_minus + %16.2f, ip =
'${Framed-IP-Address}' WHERE id='${Acct-Session-Id}' AND unique_id = '${Acct-
Unique-Session-Id}' AND server = '${NAS-IP-Address}' AND port = ${NAS-Port} AND
user = '${SQL-User-Name}' AND terminate_cause = 'Online'"
```

```
accounting_stop_query = "UPDATE ${acct_table} SET time_on = if(${Acct-Session-
Time},${Acct-Session-Time},time_on), stop_time = 'S', in_bytes = %llu, out_bytes
= %llu, billing_minus = billing_minus + %16.2f, terminate_cause='${Acct-
Terminate-Cause}', connect_info = '${Connect-Info}', id =
concat(unix_timestamp(start_time),'-${Acct-Session-Id}'), unique_id =
concat(unix_timestamp(start_time),'-${Acct-Unique-Session-Id}'), ip = '${Framed-
IP-Address}' WHERE id='${Acct-Session-Id}' AND unique_id = '${Acct-Unique-Session-
Id}' AND server = '${NAS-IP-Address}' AND port = ${NAS-Port} AND user = '${SQL-
User-Name}' AND terminate_cause = 'Online'"
```

```
# Comment simul_count_query to disable simultaneous use checking
simul_count_query = "SELECT id, user, server, port, ip, call_from FROM
${acct_table} WHERE user = '${SQL-User-Name}' AND terminate_cause = 'Online'"
simul_verify_query = "SELECT id, user, server, port, ip, protocol, last_change
FROM ${acct_table} WHERE terminate_cause = 'Online' ORDER BY user"
```

```
# Remove stale session if checkrad does not see a double login
deletestalesessions_by_checkrad = no
deletestalesessions_by_alive = no
alive_max_interval = 60
```

```
set_session_timeout = yes
use_ext_price_attr = yes
```



## Кафедра інженерії програмного забезпечення

## Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

```
accounting_on_alive = yes
ignore_localhost_pkt = yes
kill_by_alive = yes
round_to_minute = no
kill_program = "${sbindir}/pppkiller %{User-Name} %{NAS-IP-Address} %{Client-
IP-Address} %{NAS-Port} %{Framed-IP-Address}"
allowed_service = ""

bytes_per_megabyte = 1048576

}
```

## ДОДАТОК Б

### Масив структур CONF\_PARSER

```
static CONF_PARSER module_config[] = {
    {"driver",PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_driver), NULL,
    "mysql"},
    {"server",PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_server), NULL,
    "localhost"},
    {"port",PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_port), NULL, ""},
    {"login", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_login), NULL,
    "ispuser"},
    {"password", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_password), NULL,
    ""},
    {"isp_db", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_db), NULL, "ispdb"},
    {"auth_table", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_auth_table), NULL,
    "users"},
    {"packets_table", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_packets_table),
    NULL, "packets"},
    {"acct_table", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,isp_acct_table), NULL,
    "actions"},
    {"num_socks", PW_TYPE_INTEGER, offsetof(ISP_CONFIG,num_socks), NULL, "64"},
    {"connect_failure_retry", PW_TYPE_INTEGER,
    offsetof(ISP_CONFIG,connect_failure_retry), NULL, "10"},
    {"connect_failure_retry_delay", PW_TYPE_INTEGER,
    offsetof(ISP_CONFIG,connect_failure_retry_delay), NULL, "10"},
    {"sql_user_name", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,sql_user_name),
    NULL, ""},
    {"group_select_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,group_select_query), NULL, ""},
    {"authenticate_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,authenticate_query), NULL, ""},
    {"authorize_query", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,authorize_query),
    NULL, ""},
    {"statistic_select_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,statistic_select_query), NULL, ""},
    {"accounting_start_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_start_query), NULL, ""},
    {"accounting_start_query_alt", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_start_query_alt), NULL, ""},
    {"accounting_reopen", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_reopen), NULL, ""},
    {"accounting_alive_auth_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_alive_auth_query), NULL, ""},
    {"accounting_stop_auth_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_stop_auth_query), NULL, ""},
    {"accounting_update_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_update_query), NULL, ""},
    {"accounting_stop_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,accounting_stop_query), NULL, ""},
    {"simul_count_query", PW_TYPE_STRING_PTR,
    offsetof(ISP_CONFIG,simul_count_query), NULL, ""},
```

## Кафедра інженерії програмного забезпечення

## Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

```

{"simul_verify_query", PW_TYPE_STRING_PTR,
  offsetof(ISP_CONFIG,simul_verify_query), NULL, ""},
{"deletestalesessions_by_checkrad", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,deletestalesessions_by_checkrad), NULL, "no"},
{"alive_max_interval", PW_TYPE_INTEGER,
  offsetof(ISP_CONFIG,alive_max_interval), NULL, "60"},
{"set_session_timeout", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,set_session_timeout), NULL, "no"},
{"use_ext_price_attr", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,use_ext_price_attr), NULL, "no"},
{"accounting_on_alive", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,accounting_on_alive), NULL, "yes"},
{"ignore_localhost_pkt", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,ignore_localhost_acct), NULL, "yes"},
{"kill_by_alive", PW_TYPE_BOOLEAN, offsetof(ISP_CONFIG,kill_by_alive), NULL,
  "yes"},
{"kill_program", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,kill_program), NULL,
  "/bin/false"},
{"safe_characters", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,allowed_chars),
  NULL, "@abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.-_:/"},
{"allowed_service", PW_TYPE_STRING_PTR, offsetof(ISP_CONFIG,service_type),
  NULL, NULL},
{"bytes_per_megabyte", PW_TYPE_INTEGER,
  offsetof(ISP_CONFIG,bytes_per_megabyte), NULL, "1048576"},
{"round_to_minute", PW_TYPE_BOOLEAN, offsetof(ISP_CONFIG,round_to_minute),
  NULL, "no"},
{"set_max_octet_limit", PW_TYPE_BOOLEAN,
  offsetof(ISP_CONFIG,set_max_octet_limit), NULL, "no"},
{"max_octet_limit", PW_TYPE_INTEGER, offsetof(ISP_CONFIG,max_octet_limit),
  NULL, "0x7fffffff"},
{NULL, -1, 0, NULL, NULL}
};

```

## ДОДАТОК В

### Структура isp\_user

```

typedef struct isp_user {
    char        name[65];
    char        prefix[5];
    char        passwd[255];
    int         crypt_method;
    ULONGINT    gid;
    ULONGINT    uid;
    double      deposit;
    double      money;
    double      credit;
    int         add_date;
    int         blocked;
    int         activated;
    int         expired;
    ULONGINT    total_time;
    ULONGINT    total_traffic;
    ULONGINT    month_traffic;
    double      total_money;
    double      month_money;
    double      week_money;
    double      day_money;
    double      last_connection;
    char        framed_ip[17];
    char        framed_mask[17];
    int         session_timeout;
    char        other_params[255];
    char        huntgroup_name[65];
    int         idle_timeout;
    int         simultaneous_use;
    char        login_time[128];
    char        call_from[33];
    ULONGINT    time_on;
    ULONGINT    in_bytes;
    ULONGINT    out_bytes;
    Int         start_time;
    Int         stop_time;
    ULONGINT    hour_in_bytes;
    ULONGINT    hour_out_bytes;
    int         last_change;
    ULONGINT    current_hour_in_bytes;
    ULONGINT    current_hour_out_bytes;
    double      current_hour_traffic_money;
    ULONGINT    current_time_on;
    ULONGINT    current_in_bytes;
    ULONGINT    current_out_bytes;
    ULONGINT    sub_total_time;
    ULONGINT    sub_total_traffic;
    double      sub_total_money;
    double      sub_money;
    char        allowed_servers[MAX_SERVERS_PATH];
    uint32_t    nas_ip;
    char        nas_address[16];
} ISP_USER;
  
```

## ДОДАТОК Г

### Скрипт перевірки прийняття Угоди `check_agreement.py`

```
import subprocess
import sys

import MySQLdb

from nibs.dbconnector import DatabaseConnection
from nibs.dbhelper import is_user_online
from nibs.iptbllib import find_user_rules
from nibs.loghelper import logger
from nibs.sqlithelper import SqliteConnection

def check_agreement_user(user):
    query = "SELECT users.user,users.uid,agree_date,status FROM agreement,users
WHERE users.uid=agreement.uid AND status=1 AND users.user=%s"
    with DatabaseConnection() as conn:
        if conn:
            try:
                cur = conn.cursor()
                cur.execute(query, (user,))
                rows = cur.fetchall()
                for row in rows:
                    if row[0] > 0:
                        logger.debug(row[0])
                        add_to_local_db(user)
                        return 0
            except MySQLdb.Error as e:
                print "MySQL Error [%d]: %s" % (e.args[0], e.args[1])
                sys.exit(250)
            cur.close()
    return 255

def check_user_gid(user, gids):
    gids_str = ','.join(gids)
    row = None
    with DatabaseConnection() as conn:
        if conn:
            try:
                cur = conn.cursor()
                cur.execute("SELECT user FROM users WHERE gid in (" + gids_str +
") AND user=%s AND credit!=777", (user,))
                row = cur.fetchone()
            except MySQLdb.Error as e:
                logger.debug("MySQL Error [%d]: %s" % (e.args[0], e.args[1]))
                sys.exit(250)
            cur.close()
    return row

def add_to_local_db(user, dbname='/tmp/agreement.db'):
    with SqliteConnection(dbname) as connection:
        cursor = connection.cursor()
        try:
            cursor.execute('INSERT INTO agreement VALUES(?, ?)', (user, 1))
            connection.commit()
            logger.debug("add_to_local_db %s - OK" % user)
        except:
            logger.debug("Уже есть user %s", user)
```

## Кафедра інженерії програмного забезпечення

## Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

```

def create_agreement_table(dbname):
    with SqliteConnection(dbname) as connection:
        cursor = connection.cursor()
        cursor.execute('CREATE TABLE IF NOT EXISTS agreement(user text primary
key, status int)')
        # logger.debug("База %s создана" % dbname)

def check_local_db(user, dbname='/tmp/agreement.db'):
    with SqliteConnection(dbname) as connection:
        cursor = connection.cursor()
        cursor.execute('SELECT user FROM agreement WHERE user=?', (user,))

        row = cursor.fetchone()
        if row is None:
            logger.debug("Не нашел %s в %s" % (user, dbname))
            return False
    return True

def iptable_rule_add(ip, user, ppp_iface):
    if '192.168.2' in ip:
        return

        cmd = "iptables -t nat -I PREROUTING 1 -i %s -s %s -m comment --comment %s -j
AGREEMENT" % ((ppp_iface, ip, user))
        # print cmd
        res = subprocess.call(cmd, shell=True, stdout=open("/dev/null", 'w'),
stderr=subprocess.STDOUT)
        logger.debug("Выполнил (%s): %s" % (res, cmd))
        cmd = "iptables -I FORWARD 1 -i %s -s %s -m comment --comment %s -j AGREEMENT"
% ((ppp_iface, ip, user))
        # print cmd
        res = subprocess.call(cmd, shell=True, stdout=open("/dev/null", 'w'),
stderr=subprocess.STDOUT)
        logger.debug("Выполнил (%s): %s" % (res, cmd))

def iptable_rule_del_by_num(user):
    cmd = "iptables -t nat -L PREROUTING -n -v --line-num | grep ' %s ' | awk
'{print $1}'" % user
    p = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
    num, errors = p.communicate()
    if num:
        cmd = "iptables -t nat -D PREROUTING %d" % int(num)
        res = subprocess.call(cmd, shell=True, stdout=open("/dev/null", 'w'),
stderr=subprocess.STDOUT)
        logger.debug("Выполнил (%s): %s" % (res, cmd))

        cmd = "iptables -L FORWARD -n -v --line-num | grep ' %s ' | awk '{print $1}'"
% user
        p = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
        num, errors = p.communicate()
        if num:
            cmd = "iptables -D FORWARD %d" % int(num)
            # print cmd
            res = subprocess.call(cmd, shell=True, stdout=open("/dev/null", 'w'),
stderr=subprocess.STDOUT)
            logger.debug("Выполнил (%s): %s" % (res, cmd))

def iptable_rule_del(user):
    cmd = 'iptables-save | grep -v " %s " | iptables-restore' % user

```

## Кафедра інженерії програмного забезпечення

## Програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій

```

res = subprocess.call(cmd, shell=True, stdout=open("/dev/null", 'w'),
stderr=subprocess.STDOUT)
logger.debug("Виконав (%s): %s" % (res, cmd))

def clear_rules():
    users = find_user_rules()
    for user in users:
        if not is_user_online(user):
            iptable_rule_del(user[2])
            logger.debug("Удалил правила для %s", user[2])

def agreement_exclude_users(user, file='/root/bin/agreement_exclude.txt'):
    with open(file, 'r') as exclude_file:
        users = exclude_file.readlines()
        if user in users:
            return True
    return False

if __name__ == '__main__':
    create_agreement_table('/tmp/agreement.db')
    parser = argparse.ArgumentParser(description='User Agreement checker')
    group = parser.add_argument_group('group')
    group.add_argument('-a', '--add', nargs=3, help='Add rule by <name> <ip> <if_name>')
    group.add_argument('-g', '--gid', nargs='+', type=str, help='Add rule by <GID>')
    parser.add_argument('-r', '--rm', nargs=1, type=str, help='Del rule by <name>')
    parser.add_argument('--clear', required=False, action="store_true",
                        help='Clear all rules for users who is not Online')
    args = parser.parse_args()
    if args.add:
        gids = None
        (user, ip, ppp_iface) = args.add
        if check_local_db(user):
            logger.debug("check_local_db %s - OK" % user)
            sys.exit(0)
        if args.gid:
            gids = args.gid
            if not check_user_gid(user, gids):
                logger.debug('User %s does not in GID %s' % (user, gids))
                sys.exit(0)
        print user, ip, ppp_iface, gids
        res = check_agreement_user(user)
        if res == 255:
            iptable_rule_add(ip, user, ppp_iface)
    elif args.rm:
        print args.rm
        iptable_rule_del(args.rm[0])
    elif args.clear:
        clear_rules()

```

## ДОДАТОК Д

### Акт впровадження



Від 23 січня 2023 р.


№ 128

#### АКТ

**Про впровадження результатів наукового дослідження доцента кафедри інженерії програмного забезпечення Чорноморського національного університету ім. Петра Могили Кірей Катерини Олександрівни згідно договору №2/1 від 01.09.2020 р. о співпраці ТОВ ТК «НЕОН» і ЧНУ ім. Петра Могили**

Цей акт засвідчує те, що у ТОВ ТК «НЕОН» була впроваджена програмна система контролю, обліку та доступу в Інтернет абонентів оператора телекомунікацій:

- розроблено та впроваджено модуль розширення RADIUS-сервера та скрипт перевірки прийняття Угоди, що загалом надає додаткові можливості щодо керування доступом абонентами до мережі Інтернет;
- оптимізована система SQL запитів до корпоративного сховища даних;
- виконано дослідження ефективності роботи системи у пікові навантаження.

Директор  Конєв. П.О.



ТОВ «Телекомунікаційна компанія «НЕОН» ЄДРПОУ: 43011995  
Адреса: 54018, м. Миколаїв, пров. Кобера 13А  
Почтова адреса: 54018, м. Миколаїв, а/с 246  
п/р UA523266100000026003053242474  
Миколаївське РУ АТ КБ "ПРИВАТБАНК" МФО 326610  
тел. (0512) 711-511, 073-080-06-06  
web: www.neon.mk.ua, email: neon@neon.mk.ua