Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
Кафедра «Комп'ютерні інформаційні технології»

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: « Розробка застосунку для швидкого конфігурування мережевих пристроїв »
за освітньою програмою: «12 Інженерія програмного забезпечення»
зі спеціальності: «121 Інженерія програмного забезпечення»
Виконав: студент групи «ПЗ20130»

Студент	(підпис)	/Іван НЕСТЕРОВ/ (Ім'я ПРІЗВИЩЕ)
Керівник:	(підпис)	/Іван КЛИМЕНКО/ (Ім'я ПРІЗВИЩЕ)
Нормоконтрол	(підпис)	/Світлана ВОЛКОВА/ (Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____ (підпис)

Дніпро - 2023

Ministry of Education and Science of Ukraine

Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»

Department «Computer information technology»

Explanatory Note

to Bachelor's Thesis

on the topic: « Development of an application for quick configuration of network devices »
according to educational curriculum «12 Software engineering»
in the Speciality: «121 Software engineering»

Done by the student of the group PZ20130: / / Ivan NESTEROV

Scientific Supervisor: / / Ivan KLIMENO

Normative controller: / / Svetalna VOLKOVA

Dnipro - 2023

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: «Комп'ютерні технології і системи»
Кафедра: «Комп'ютерні інформаційні технології»
Освітня програма: «Інженерія програмного забезпечення»
Спеціальність: «Інженерія програмного забезпечення»

«ДО ЗАХИСТУ»
Завідувач кафедри
_____ Вадим ГОРЯЧКІН
(підпис)
202_ р. _____ «_____»

ЗАВДАННЯ
до дипломної роботи на здобуття ОС «бакалавр»

студента групи _____ Нестеров Іван Євгенович
номер групи (ПІБ)

1. Тема роботи: « Розробка застосунку для швидкого конфігурування мережевих пристроїв »
затверджена наказом по університету від «___»_____202_ р. № ___ ст.

2. Термін подання студентом роботи «20» червня 2023 р.

3. Вихідні дані до дипломної роботи, приклади постановок завдань розробки застосунку для швидкого конфігурування, прототипи мережевих пристроїв сфера їх застосування, особливості завдань і спеціалізовані мережеві прострої для роботи з операційною системою, область компромісних рішень їх властивості, програмні засоби реалізації.

4. Зміст пояснювальної записки (перелік питань до розробки):
Вступ, аналіз існуючих рішень та загальні відомості взаємодії з мережевими приладами, опис алгоритму роботи програмного забезпечення для взаємодії та налаштування роутера, розробка програмного забезпечення, висновок.
5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень):
Презентація
Відео роботи програми

КАЛЕНДАРНИЙ ПЛАН

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, виріб та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.22 - 18.02.22
Робочий проект	Програмування та відлагодження програми.	19.02.22 - 20.05.22
	Тестування програми	20.05.22 - 27.05.22
	Розробка, узгодження і затвердження програмної документації.	27.05.22 - 12.06.22
	Подання кваліфікаційної роботи до кафедри	20.06.22
	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	28.06.22

Студент _____ /Іван НЕСТЕРОВ/
(підпис студента) (Ім'я ПРІЗВИЩЕ)

Керівник _____ /Іван КЛИМЕНКО/
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка складається з 55 сторінок, 3 таблиці, 27 рисунків, та 5 розділів, 14 використані джерела.

– Вступ – в даному розділі описується сутність розробки, її актуальність. Складається з 2 сторінок;

– Аналіз існуючих рішень та загальні відомості взаємодії з мереживими приладами – у цьому розділі ми розглядаємо та описуємо історію вивчення мережевих приладів, аналізуємо загальні поняття та складаємо постановку подальшої роботи. Складається з 19 сторінок;

– Опис алгоритму роботи програмного забезпечення для взаємодії та налаштування роутера – у цьому розділі описуються алгоритм налаштування маршрутизатора, його функціональні вимоги, вхідні та вихідні дані, описується вибір модифікації методу доступа, . Складається з 20 сторінок;

– Розробка програмного забезпечення – у цьому розділі, обирається мова програмування. на основі прототипу виконується вибір набору даних, розробляється динаміка системи, екрани додатку, інтерфейс додатку. Складається з 6 сторінок;

– Загальні висновки – підсумки всієї роботи. Складається з 1 сторінки;

– Список використаних джерел – включає в себе бібліографічний список використаної літератури. Складає 2 сторінки;

– Додатки – містить технічне завдання та код робочого проекту.

ЗМІСТ

ВСТУП

У всьому світі стрімко зростає потреба у бездротових з'єднаннях, особливо у сфері бізнесу. Користувачі з бездротовим доступом до інформації завжди і скрізь можуть працювати набагато продуктивніше і ефективно, ніж їхні колеги, прив'язані до дротових телефонних і комп'ютерних мереж.

Зазвичай бездротові мережеві технології групуються в три типи, що різняться за масштабом дії їх радіосистем, але вони успішно застосовуються у бізнесі.

PAN (персональні мережі) - короткодіючі, радіусом до 10 м мережі, які пов'язують ПК та інші пристрої - КПК, мобільні телефони, принтери тощо. За допомогою таких мереж реалізується проста синхронізація даних, усуваються проблеми з великою кількістю кабелів в офісах, реалізується простий обмін інформацією невеликих робочих групах. Найбільш перспективний стандарт для PAN - це Bluetooth .

Бездротові локальні мережі (WLAN) — радіус дії до 100 м. З їх допомогою реалізується бездротовий доступ до групових ресурсів у будівлі, університетському блоці тощо. Зазвичай такі мережі використовуються для продовження провідних корпоративних локальних мереж. У невеликих компаніях WLAN можуть повністю замінити дротові з'єднання. Основний стандарт для WLAN - 802.11.

WWAN (бездротові мережі широкої дії) — бездротовий зв'язок, який забезпечує мобільним користувачам доступ до їх корпоративних мереж та Інтернету. Поки що тут немає домінуючого стандарту, але найактивніше впроваджується технологія GPRS — найшвидше в Європі та з деяким відставанням у США.

На сучасному етапі розвитку мережевих технологій, технологія бездротових мереж Wi-Fi є найбільш зручною в умовах, що потребують мобільність, простоту встановлення та використання. Wi-Fi (від англ. Wireless fidelity - бездротовий зв'язок) - стандарт широкосмугового бездротового зв'язку сімейства 802.11 розроблений у 1997р. Як правило, технологія Wi-Fi

використовується для організації бездротових локальних комп'ютерних мереж, а також створення гарячих точок високошвидкісного доступу в Інтернет.

Між собою обчислювальні мережі об'єднуються різними пристроями, такими як мости, комутатори, шлюзи та маршрутизатори. З усіх перелічених пристроїв маршрутизатори мають найбільш повний набір функцій для забезпечення ефективної міжмережевої взаємодії (збір інформації про топологію міжмережевих з'єднань, ізоляція трафіку окремих частин мережі один від одного, вибір найбільш раціонального маршруту з декількох можливих, здатність зв'язувати в єдину мережу підмережі, побудовані з використанням різних мережевих технологій, наприклад Ethernet та X.25 та ін.). Такі потенційні можливості, які несуть у собі маршрутизатори і той новий потенційний підйом, який при цьому відчуває інформаційний комплекс, а також значне прискорення виробничого процесу не дають нам права не приймати це до розробки і дослідження та не застосовувати їх на практиці.

Тому є серйозна необхідність у вивченні дослідження моделей різних елементів та пристроїв мереж та комутаційного обладнання, принципів їх роботи та логічної побудови.

Дані дослідження допомагають у подальшому виробляти рекомендації як щодо створення обладнання на сьогоднішній день, так і з урахуванням прогресу технологій його розвитку надалі. Цей дипломний проект виконує саме таке завдання, як опис принципів організації та функціонування маршрутизаторів.

Об'єкт дослідження – процес навчання темі «Установка та налаштування маршрутизатора».

Предмет дослідження – методика навчання темі «Установка та налаштування маршрутизатора».

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ЗАГАЛЬНІ ВІДОМОСТІ ВЗАЄМОДІЇ З МЕРЕЖИВИМИ ПРИЛАДАМИ

1.1 Історія вивчення

Розвиток бездротових технологій передачі почався з передачі першого радіосигналу та появою у 20-х роках XX століття перших радіоприймачів з амплітудною модуляцією. У 1930-ті роки з'явилося радіо з частотною модуляцією та телебачення. У 1970-ті роки було створено перші бездротові телефонні системи. Спочатку це були аналогові мережі, на початку 1980-х з'явився стандарт GSM, який ознаменував початок переходу на цифрові стандарти як такі, що забезпечують кращий розподіл спектру, кращу якість сигналу та більшу безпеку. З 90-х років XX ст. відбувається зміцнення позицій бездротових мереж. Бездротові технології міцно входять у наше життя. Розвиваючись із величезною швидкістю, вони стимулюють створення нових пристроїв та послуг.

Бездротові мережі розгортаються в аеропортах, університетах, готелях, ресторанах, підприємствах. Бездротові мережі особливо на підприємствах, де співробітники активно переміщаються територією під час робочого дня з метою обслуговування клієнтів або збору інформації.

Точкою відліку в галузі розробки стандартів бездротових мереж є освіта всесвітньою організацією IEEE (Інститут інженерів з електрики та електроніки) (англ. Institute of Electrical and Electronics Engineers) комітету 802.11 в 1990 році. "Бездротова якість" або "Бездротова точність") - це сучасна технологія з'єднання комп'ютерів в локальну мережу і підключення їх до Internet. Датою створення Wi-FI вважається 1991 рік. Він був створений у Нідерландах Віком Хейз, який входив до складу команди, яка займалася розробкою таких стандартів як IEEE 802.11b IEEE 802.11a та IEEE 802.11g. Сам стандарт Wi-FI був затверджений у 2009 році і дозволив передавати дані на швидкості аж до 54 Мбітс/с. Це було значним кроком уперед, але виробники обладнання намагалися самостійно підвищити швидкодію та випускали пристрої з позначеннями 802.11b+ (22 Мбіт/с або 27,5 Мбіт/с), 802.11g+, Super G, Turbo (до 108 або 125 Мбіт/с) та іншими подібними доповненнями. Далі було розроблено новий стандарт 802.11n. Він передбачає підвищення швидкодії до 300 Мбіт/с. Стандарт 802.11n

працює на частотах 2.4 – 2.5 та 5 ГГц. Крім того, стандарт 802.11 n може працювати в трьох режимах:

- успадкованому (Legacy), в якому забезпечується підтримка пристроїв 802.11b/g та 802.11a;

- змішаному (Mixed), в якому підтримуються пристрої 802.11b/g, 802.11a та 802.11n;

- "чистому" режимі - 802.11n (саме в цьому режимі і можна скористатися перевагами підвищеної швидкості та збільшеною дальністю передачі даних, що забезпечуються стандартом 802.11n).

Сегмент Wi - Fi мережі може використовуватися як самостійна мережа, або у складі більш складної мережі, що містить бездротові, так і звичайні провідні сегменти.

Переваги та недоліки Wi - Fi :

- дозволяє розгорнути мережу без прокладання кабелю, може зменшити вартість розгортання та розширення мережі. Місця, де не можна прокласти кабель, наприклад, поза приміщеннями та в будинках, що мають історичну цінність, можуть обслуговуватися бездротовими мережами;

- Wi-Fi пристрої широко поширені над ринком. А устрою різних виробників можуть взаємодіяти на базовому рівні сервісів;

- Wi-Fi мережі підтримують роумінг, тому клієнтська станція може переміщатися в просторі, переходячи від однієї точки доступу до іншої;

- Wi-Fi – це набір глобальних стандартів. На відміну від мобільних телефонів, Wi-Fi обладнання може працювати в різних країнах по всьому світу.

- частотний діапазон та експлуатаційні обмеження у різних країнах неоднакові; у багатьох європейських країнах дозволено два додаткові канали, які заборонені в США; У Японії є ще один канал у верхній частині діапазону, інші країни, наприклад Іспанія, забороняють використання низькочастотних каналів. Більше того, деякі країни, наприклад Італія, вимагають реєстрації всіх Wi-Fi мереж, що працюють поза приміщеннями, або вимагають реєстрації Wi-Fi-оператора;

- досить високе в порівнянні з іншими стандартами споживання енергії, що зменшує час життя батарей та підвищує температуру пристрою;

- найпопулярніший стандарт шифрування, Wired Equivalent Privacy або WEP, може бути відносно легко зламаний навіть за правильної конфігурації (через слабку стійкість ключа). Незважаючи на те, що нові пристрої підтримують більш досконалий протокол Wi-Fi Protected Access (WPA), багато старих точок доступу не підтримують його та потребують заміни. Прийняття стандарту 802.11i (WPA2) у червні 2004 року зробив доступною більш безпечну схему, яка доступна в новому устаткуванні. Обидві схеми вимагають більш стійкого пароля, ніж ті, які зазвичай призначаються користувачами. Багато організацій використовують додаткове шифрування (наприклад, VPN) для захисту від вторгнення;
- Wi-Fi мають обмежений радіус дії. Типовий домашній Wi-Fi маршрутизатор стандарту 802.11b або 802.11g має радіус дії 45 м у приміщенні та 90 м зовні. Відстань залежить також від частоти. Wi-Fi у діапазоні 2.4 ГГц працює далі, ніж Wi-Fi у діапазоні 5 ГГц, і має радіус менше, ніж Wi-Fi (і пре-Wi-Fi) на частоті 900 МГц.
- накладання сигналів закритої або використовуючої шифрування точки доступу та відкритої точки доступу, що працюють на одному або сусідніх каналах, може перешкодити доступу до відкритої точки доступу. Ця проблема може виникнути при великій щільності точок доступу, наприклад у великих багатоквартирних будинках, де багато мешканців ставлять свої точки доступу Wi-Fi;
- неповна сумісність між пристроями різних виробників або неповна відповідність стандарту може призвести до обмеження можливостей з'єднання або зменшення швидкості.

1.2. Загальні поняття

Першою технологією захисту бездротових мереж прийнято вважати протокол безпеки WEP (Wired Equivalent Privacy – еквівалент провідної безпеки), який спочатку закладений у специфікаціях стандарту 802.11. WEP. Протокол шифрування використовує нестійкий алгоритм RC4 на статичному ключі. Існує 64-, 128-, 256- та 512-бітове шифрування. Чим більше біт використовується для зберігання ключа, тим більше можливих комбінацій ключів, а відповідно вища стійкість мережі до злому. Частина WEP-ключа є статичною (40 біт у разі 64-бітного шифрування), а інша частина (24 біта) – динамічною (вектор ініціалізації), вона змінюється у процесі роботи мережі. Основною вразливістю протоколу WEP є те, що вектори ініціалізації повторюються через деякий проміжок часу, і зломщику потрібно лише обробити ці повтори і обчислити статистичну частину ключа. Для підвищення рівня безпеки можна до WEP-шифрування використовувати стандарт 802.1x або VPN.

Особливості WEP – протоколу:

досить стійкий до атак, пов'язаних з простим перебором ключів шифрування, що забезпечується необхідною довжиною ключа та частотою зміни ключів та вектора, що ініціалізує;

самосинхронізація кожного повідомлення. Ця властивість є ключовою для протоколів рівня доступу до середовища передачі, де велика кількість спотворень та втрачених пакетів;

WEP легко реалізувати;

відкритість;

використання WEP -шифрування не є обов'язковим у мережах стандарту IEEE 802.11.

Таким чином, технологія WEP не забезпечує належного рівня безпеки корпоративної мережі підприємства, але її цілком достатньо для домашньої бездротової мережі, коли обсяг перехопленого мережного трафіку занадто малий для аналізу та розкриття ключа.

802.1X

IEEE 802.1X – стандарт, який виявився ключовим для розвитку індустрії бездротових мереж загалом. За основу взято виправлення недоліків технологій безпеки, що застосовуються у 802.11, зокрема, можливість злому WEP,

залежність від технологій виробника тощо. 802.1X дозволяє підключати до мережі навіть PDA-пристрої, що дозволяє вигідніше використовувати саму ідею бездротового зв'язку. З іншого боку, 802.1X та 802.11 є сумісними стандартами. У 802.1X застосовується той же алгоритм, що і в WEP, а саме - RC4, але з деякими відмінностями.

.1X базується на протоколі розширеної автентифікації Extensible Authentication Protocol (EAP), протоколі захисту транспортного рівня Transport Layer Security (TLS) та сервері доступу RADIUS (Remote Access Dial-in User Server). Плюс до цього варто додати нову організацію роботи клієнтів мережі. Після того, як користувач пройшов етап аутентифікації, йому надсилається секретний ключ у зашифрованому вигляді на певний незначний час - час сеансу, що діє на даний момент. По завершенні цього сеансу генерується новий ключ і знову надсилається користувачеві. Протокол захисту транспортного рівня TLS забезпечує взаємну автентифікацію та цілісність передачі даних. Усі ключі є 128-розрядними за промовчанням.

WPA

Протокол шифрування WPA. Більш стійкий протокол шифрування, ніж WEP, хоча використовується той самий алгоритм RC4. Технологія WPA призначена для використання з сервером автентифікації 802.1X, який розподіляє різні ключі кожному користувачеві. Однак її також можна використовувати в менш безпечному режимі Pre-Shared Key (PSK). Ключ PSK призначений для домашніх і невеликих мереж офісів, де для всіх користувачів використовується однаковий пароль. Протокол WPA-PSK також називається WPA-Personal. Протокол WPA-PSK дозволяє бездротовому пристрої Brother обмінюватися даними з точками доступу за допомогою способу шифрування TKIP або AES. Протокол WPA2-PSK дозволяє бездротовому пристрої Brother обмінюватися даними з точками доступу за допомогою способу шифрування AES. (Temporal Key Integrity Protocol) - протокол динамічних ключів мережі, які часто змінюються. TKIP відповідає за збільшення розміру ключа з 40 до 128 біт, а також за заміну одного статичного ключа WEP ключами, які автоматично генеруються та розсилаються сервером аутентифікації. Крім того, у TKIP використовується спеціальна ієрархія ключів та методологія управління

ключами, яка прибирає зайву передбачуваність, яка використовувалася для несанкціонованого зняття захисту WEP ключів.

Алгоритм AES (Advanced Encryption Standard) симетричний алгоритм блокового шифрування (розмір блоку 128 біт, ключ 128/192/256 біт), прийнятий як стандарт шифрування. AES є одним із найпоширеніших алгоритмів симетричного шифрування.

Якщо у точці доступу використовується протокол безпеки бездротового доступу WPA-PSK, то на комп'ютері (адаптері) потрібно використовувати протоколи WPA та TKIP, а при використанні протоколу WPA2-PSK на комп'ютері потрібно використовувати протокол WPA2 та стандарт AES. В іншому випадку користувачі не зможуть підключитися до бездротової точки доступу (неможливо підключитися до бездротової точки доступу при використанні WPA2 з протоколом TKIP). Це протокол безпеки даних, що відповідає стандарту IEEE 802.11i для бездротових локальних мереж починаючи з 2004 року. Більш надійний, ніж у проміжку, використаний протокол WPA, т.к. для шифрування використовує замість TKIP набагато безпечніший алгоритм CCMP на основі AES. Починаючи з 2006 року всі пристрої з логотипом Wi-Fi повинні підтримувати WPA2. На відміну від WPA, WPA2 також хоче порівняно нове обладнання, а це означає, що апаратне забезпечення, куплене до 2006 року, доведеться змінити. Існує дві версії WPA2. Є WPA2 для підприємств та для дому. Домашній варіант використовує цей ключ і перевіряє його правильність негайно. А версія для підприємств робить це через сервер.

VPN

Технологія віртуальних приватних мереж Virtual Private Network (VPN) була запропонована компанією Intel для забезпечення безпечного з'єднання клієнтських систем із серверами загальнодоступних інтернет-каналів. VPN дуже добре себе зарекомендували з погляду шифрування та надійності автентифікації. Плюс технології полягає і в тому, що протягом більш ніж трьох років практичного використання в індустрії цей протокол не отримав жодних нарікань з боку користувачів. Інформації про його злами не було.

Технологій шифрування VPN застосовується кілька, найбільш популярні з них описані протоколами PPTP, L2TP і IPSec з алгоритмами шифрування DES,

Triple DES, AES і MD5. IP Security (IPSec) використовується приблизно 65-70% випадків. За його допомогою забезпечується практично максимальна безпека лінії зв'язку.

І хоча технологія VPN не призначалася спочатку саме для Wi-Fi, вона може використовуватися для будь-якого типу мереж, і ідея захистити з її допомогою бездротові варіанти одна з кращих на сьогодні.

Для реалізації VPN-захисту в рамках мережі необхідно встановити спеціальний VPN-шлюз (програмний або апаратний), в якому створюються тунелі, по одному на кожного користувача. Наприклад, для бездротової мережі слід встановити шлюз безпосередньо перед точкою доступу. А користувачам мережі необхідно встановити спеціальні клієнтські програми, які також працюють за рамками бездротової мережі і розшифровка виноситься за її межі.

Хоча все це досить громіздко, але дуже надійно, головна вада такого рішення - необхідність адміністрування. Другий суттєвий мінус – зменшення пропускної спроможності каналу на 30-40%.

Пошук у мережі Internet пропонує величезну кількість інформації на запит «Налаштування маршрутизатора. З усього хаотичного набору посилань можливий відбір: відеокурси налаштування, посібники користувача на сайті або в електронному варіанті, а також статті, опубліковані на форумах або спеціалізованих сайтах. Найбільш ефективна інформація на тему це теоретичні статті на сайтах підкріплені візуальними матеріалами.

Сайт http://www.dlink.ru [1] містить опис продуктів компанії та їх вартість, рішення щодо настроювання пристроїв та розташування офісів компанії в потрібному регіоні. Також можна завантажити програмне забезпечення, посібники користувача, драйвера, прошивки для всіх пристроїв розроблених та випущених компанією D - Link .

Сайт http://support.akado-ural.ru/ [2] містить інформаційно-довідкові матеріали з налаштування та діагностики мережі, операційних систем, різного обладнання та програмного забезпечення. Всі інструкції наведені на сайті з теоретичними та візуальними матеріалами. Інформація на сайті добре структурована. Усі розділи пов'язані між собою. p align="justify"> Особливу

увагу слід приділити логічності переходу від статті до статті, наявність у кожній з них посилань на попередню інформацію з розглянутого питання.

Сайт http :// www . ixbit . com / [3] російськомовне інтернет-видання про комп'ютерну техніку, інформаційні технології та програмні продукти. На сайті публікуються новини IT -індустрії, статті з оглядами та тестами комп'ютерних комплектуючих та програмного забезпечення. Одним з найбільших форумів з питань комп'ютерної тематики в Росії та СНД є «Конференція iXBT » http :// forum . ixbt . com /.

http://www.wikipedia.org/ [5] головною особливістю інтернет-енциклопедії Вікіпедія є те, що створювати та редагувати її статті може, в принципі, до кожного користувача мережі інтернет, причому в абсолютній більшості випадків навіть без реєстрації на сайті енциклопедії Усі внесені такими добровольцями до будь-якої статті цієї енциклопедії зміни негайно стають видними всім відвідувачам сайту. Вікіпедія зараз є найбільшим і найпопулярнішим довідником в Інтернеті. За обсягом відомостей і тематичним охопленням Вікіпедія вважається найповнішою енциклопедією з коли-небудь створюваних за всю історію людства. Однією з основних переваг Вікіпедії як універсальної енциклопедії є можливість подання інформації рідною мовою користувача, зберігаючи таким чином цінність цієї інформації в аспекті культурної приналежності.

Надійність та точність Вікіпедії викликають питання. Інша критика вказує на схильність до Вікіпедії вандалізму, а також додавання неправдивої або неперевіреної інформації. Проте наукові дослідження свідчать, що у Вікіпедії сліди актів вандалізму

1.2 Архітектура мережевих пристроїв

Топологія «шина» своєю структурою передбачає ідентичність мережного устаткування комп'ютерів, і навіть рівноправність всіх абонентів. Приклад такої топології наведено на рисуноку 1 .1

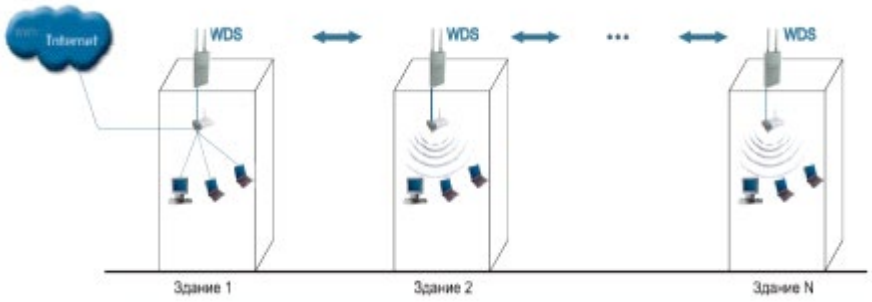

Рисунок 1.1 – Топологія типу «шина»

Тут немає центрального абонента, через якого передається вся інформація, що збільшує її надійність. Додати нових абонентів до шини досить просто. Потрібно ввести параметри нової точки доступу, що призведе до перезавантаження останньої точки.

Шині не страшні відмови окремих точок, тому що всі інші комп'ютери мережі можу нормально продовжувати обмін даними між собою, але при цьому частина комп'ютерів, що залишилися, не зможе отримати доступ в інтернет.

Топологія типу «кільце»

У цій топології кожна точка доступу з'єднується лише з двома іншими. Приклад такої топології наведено рисунку 1.2

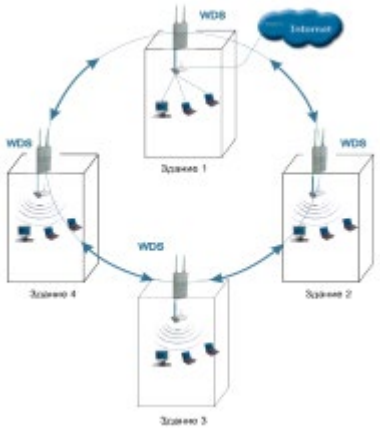

Рисунок 1.2 – Топологія типу «кільце»

Підключення нових абонентів у «кільце» здійснити дуже просто, хоча це і вимагає обов'язкової зупинки двох крайніх точок від нової точки доступу.

Основна перевага кільця полягає в тому, що ретрансляція сигналів кожним абонентам дозволяє суттєво збільшити розмір всієї мережі в цілому. Кільце у цьому відношенні значно перевищує будь-які інші топології.

Топологія типу «Зірка»

Ця топологія має яскраво виділений центр, до якого підключається решта абонентів. Весь обмін інформації йде виключно через центральну точку доступу, на яку в результаті лягає велике навантаження. Приклад такої топології наведено на рисунок 1.3 .

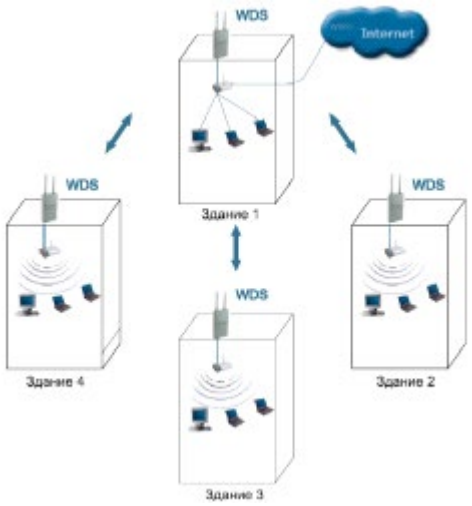

Рисунок 1.3 – Топологія типу «Зірка»

Якщо говорити про стійкість зірки до відмов точок, то вихід з ладу звичайної точки доступу ніяк не відбивається на функціонуванні частини мережі, що залишилася, але будь-яка відмова центральної струму робить мережу повністю непрацездатною.

Головний недолік цієї топології полягає у жорсткому обмеження кількості абонентів. Так як всі точки працюють на одному каналі, зазвичай центральний абонент може обслуговувати не більше 10 периферійних абонентів через велике падіння швидкості. [1]

Використовувані частоти та канали в діапазоні 2,4 ГГц

Для бездротового Wi-Fi зв'язку використовується певний діапазон частот, причому залежно від країни цей діапазон може бути різним. Весь діапазон частот розбито на кілька каналів, на яких може працювати обладнання. Стандарти 802.11b, 802.11g та 802.11n визначають наступні канали таблиця 1:

Таблиця 1 - Використовувані частоти та канали в діапазоні 2,4 ГГц

Канал	Центральна частота	Країни
1	2.412	США, Європа, Японія
2	2.417	США, Європа, Японія
3	2.422	США, Європа, Японія
4	2.427	США, Європа, Японія
5	2.432	США, Європа, Японія
6	2.437	США, Європа, Японія
7	2.442	США, Європа, Японія
8	2.447	США, Європа, Японія
9	2.452	США, Європа, Японія
10	2.457	США, Європа, Японія
11	2.462	Європа, Японія
12	2.467	Європа, Японія
13	2.472	Японія

З таблиці 1 видно, що крок каналів діапазоні 2.4 ГГц становить 5 МГц, а ширина каналу, як описано вище, становить 20МГц. Таким чином, спектр робочих частот обладнання перекривається і незалежних каналів, робота на яких можлива без взаємних перешкод, лише три - наприклад 1 (2,412 ГГц), 6 (2,437 ГГц) та 11 (2,462 ГГц), частоти яких відрізняються більш ніж на 20 МГц . Також можна використовувати як незалежні канали 2, 7, 12 або 3, 8, 13.

.3.2 Використовувані частоти та канали в діапазоні 5 ГГц

Частотні смуги та канали для 5 ГГц:

UNII -1:5150 - 5250 МГц (доступно 4 частотні канали);

UNII -2: 5250 - 5350 МГц (доступно 4 частотні канали);

UNII -2 Extended : 5470 - 5725 МГц (доступно 11 частотних каналів);

UNII -3: 5725 - 5825 МГц (доступно 4 частотні канали).

Сітка робочих каналів 5 ГГц наведена у таблиці 1.2.2.

Таблиця 2 - Сітка робочих каналів 5 ГГц

Смуга	UNII -1				UNII -3			
Канал	36	40	44	48	149	153	157	161
Центр. частота, МГц	5180	5200	5220	5240	5745	5765	5785	5805
Смуга	UNII - 2							
Канал	2	6	0	4	100	104	108	112	116	120	124	128	132	136	140
Центр. частота, МГц	260	280	300	320	500	520	540	560	580	600	620	640	680	680	700

Формування каналів Wi - Fi у діапазоні 5 ГГц рисунок 1.4.

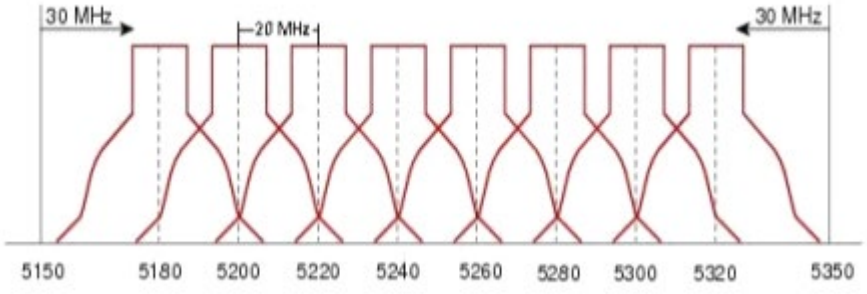

Рисунок 1.4 – Формування каналів у діапазоні 5 ГГц

У цьому дистанція від граничних діапазонів становить 30 МГц, а міжканальне рознесення становить 20 МГц.

1.4 Відомості програмного забезпечення мережевих пристроїв

Якщо вам просто потрібен хороший інструмент для аналізу Wi-Fi для особистого використання, це може бути інструмент. Інструмент аналізатора Wi-Fi підходить для вас. Ця програма аналізуватиме як 2,4 ГГц, так і 5 ГГц, використовуючи графічні спектри, які показують доступні мережі та те, що вони змішують з іншими. Wifi Analyzer надає таку інформацію, як потужність сигналу, IP-дані та дані про безпеку для кожної мережі. Ви також можете отримати доступ до тимчасової шкали, щоб побачити, коли швидкість вашої мережі була повільною або високою. Він включає в себе вибір відтінка або світлого відтінку, звукового тону для потужності сигналу і можливість збереження спектра зображення.

Він доступний за невелику плату за 4,95 долара, але якщо ви вчасно зареєструєтеся в Microsoft Store, він може бути проданий за 1,69 долара або навіть безкоштовно.

Інструмент inSSIDer Це потужний інструмент, в основному розроблений для компаній з великими наборами, якому цікаві всі доступні додаткові можливості. Оскільки розробники розробили їх для бізнесу, ви не можете аналізувати мережі 2.4 ГГц без оновлення до більш дорогого пакета.

Надає такі функції, як можливість контролювати насиченість каналів і визначати джерело поміху. Він виконує рутинний аналіз, щоб переконатися, що ви знаходитесь у кращій доступній мережі. Ви також можете запустити власну мгновену перевірку, щоб бути впевненим, що вибираєте найкращий канал для свого пристрою.

Онлайн-програми та інструменти для компіляції та об'єднання кількох зображень в одно.

inSSIDer розроблений для точної та надійної роботи з великими сетями, тому не призначений для домашнього використання. Ціна теж підтримує цю ідею. Самий дешевий пакет для inSSIDer Office коштує 149 доларів. Відсюда вартість пакетів становить 499 доларів за пакет inSSIDer Essential і 999 доларів за пакет Chanalyzer Essential.

Основні моменти Командир Wi-Fi відразу із-за красивої 3D-графіки, використаної в його репортажах. Ці звіти роблять процес аналізу Wi-Fi і мережі

простим і привабливим. Додаток сканує ваше середовище в пошуках кращої мережі в режимі реального часу, що допоможе вам швидко знайти кращий сет.

Він постачається з безліччю функцій, таких як фільтри, і можливістю сортування сетей за вашим вибором різних атрибутів. Є екран, який показує назви використовуваних каналів і підтримує кілька адаптерів Wi-Fi. У Wifi Commander немає реклами, але він коштує дорого. Зазвичай він продається за ціною 34.99 доларів. Однак, як я згадував раніше, ви обов'язково хочете перевірити це, що у них може бути скидка. В даний час вартість додатка становить всього 4,99 долара США.

1.5 Опис методів взаємодії з мережевим обладнанням

При описі будь-якої взаємодії можна виділяти різні рівні. Наприклад, уявіть собі, що двом людям, які проживають у різних населених пунктах, необхідно обмінюватися будь-якою інформацією, і вони використовують для цього традиційний спосіб надсилання листів. Вже у взаємодії такого роду можна назвати кілька рівнів:

рівень користувачів, які обмінюються листами, та використовують для цієї мети поштову службу;

рівень поштової служби, яка здійснює пересилання кореспонденції між поштовими відділеннями населених пунктів та використовує для роботи послуги транспортної мережі;

рівень транспортної мережі, що забезпечує доставку вантажів шляхами сполучення між населеними пунктами;

рівень шляхів сполучення, що забезпечує можливість фізичної доставки вантажів між населеними пунктами.

У випадку, якщо не існує прямих шляхів сполучення між населеними пунктами, до цієї схеми між рівнями поштової служби та транспортної мережі додається ще один рівень – рівень відділень перевезення пошти, що забезпечують правильне перевантаження поштових відправлень на транспортних вузлах, а також вибір альтернативних шляхів пересилання в у разі виходу з ладу транспортних ліній.

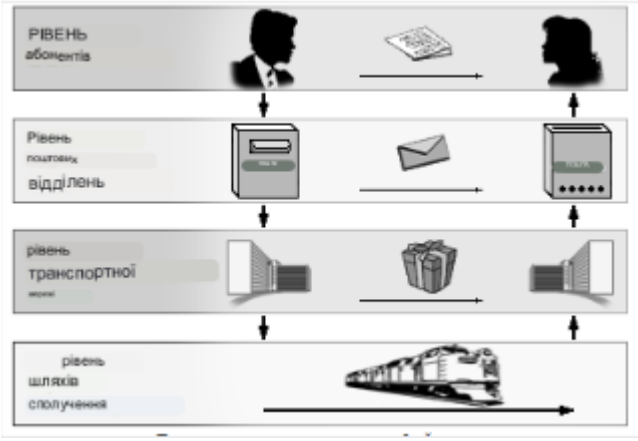

Рисунок 1.5 – Приклад рівневої взаємодії

Поділ процесу взаємодії на рівні дозволяє функціонально ізолювати різні засоби, що беруть участь у цьому процесі за принципом - "кожен займається своєю справою". Це дозволяє забезпечити достатню гнучкість у разі розширення функціональності цих коштів. Так, наприклад, виділення рівня транспортної мережі дозволяє за необхідності забезпечити транспортування між населеними пунктами не тільки поштових вантажів, а й пасажирів, не вимагаючи для цього перебудови шляхів сполучення. Виділення поштової служби забезпечує можливість пересилання не тільки листів, а й посилок, перекладів тощо, використовуючи стандартні засоби транспортної мережі та опосередковано існуючі шляхи сполучення.

Взаємодія з комп'ютерними мережами також можна описувати за допомогою рівнів. В даний час для цього широко використовується так звана модель взаємодії відкритих систем (Open Systems Interconnection, OSI).

У 1984 році Міжнародною Організацією зі Стандартизації (International Standard Organization, ISO) було розроблено модель взаємодії відкритих систем (Open Systems Interconnection, OSI). Модель є міжнародним стандартом для проектування мережевих комунікацій і передбачає рівневий підхід до побудови мереж. Кожен рівень моделі обслуговує різні етапи процесу взаємодії. За допомогою поділу на рівні мережева модель OSI полегшує спільну роботу обладнання та програмного забезпечення. Модель OSI поділяє мережеві функції на сім рівнів: прикладний, рівень вистави, сесійний, транспортний, мережевий, канальний та фізичний.

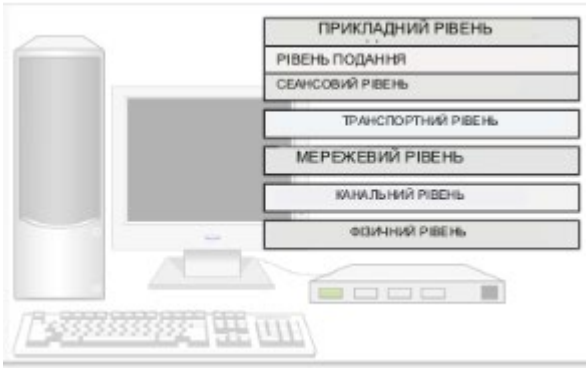

Рисунок 1.6 – Рівні моделі OSI

Фізичний рівень (Physical layer) визначає спосіб фізичного з'єднання комп'ютерів у мережі. Функціями засобів, що належать до даного рівня, є побітове перетворення цифрових даних на сигнали, що передаються по фізичному середовищу (наприклад, по кабелю), а також передача сигналів.

Канальний рівень (Data Link layer) відповідає за організацію передачі даних між абонентами через фізичний рівень, тому на даному рівні передбачені засоби адресації, що дозволяють однозначно ідентифікувати відправника та одержувача в усьому безлічі абонентів, підключених до загальної лінії зв'язку. До функцій даного рівня також входить упорядкування передачі з метою паралельного використання однієї лінії зв'язку кількома парами абонентів. Крім того, засоби канального рівня забезпечують перевірку помилок, які можуть виникати під час передачі даних фізичним рівнем.

Мережевий рівень (Network layer) забезпечує доставку даних між комп'ютерами мережі, що є об'єднання різних фізичних мереж. Цей рівень передбачає наявність засобів логічної адресації, що дозволяють однозначно ідентифікувати комп'ютер об'єднаної мережі. Однією з основних функцій, виконуваних засобами цього рівня, є цілеспрямована передача даних конкретного одержувача.

Транспортний рівень (Transport layer) реалізує передачу даних між двома програмами, що функціонують на різних комп'ютерах, забезпечуючи відсутність втрат і дублювання інформації, які можуть виникати в результаті помилок передачі нижніх рівнів. Якщо дані, що передаються через транспортний рівень, піддаються фрагментації, то кошти даного рівня гарантують складання фрагментів у правильному порядку.

Сесійний (або сеансовий) рівень (Session layer) дозволяє двом програмам підтримувати тривалу взаємодію мережі, зване сесією (session) чи сеансом. Цей рівень керує встановленням сеансу, обміном інформацією та завершенням сеансу. Він також відповідає за ідентифікацію, дозволяючи цим лише певним абонентам брати участь у сеансі, та забезпечує роботу служб безпеки з метою упорядкування доступу до інформації сесії.

Рівень подання (Presentation layer) здійснює проміжне перетворення даних вихідного повідомлення на загальний формат, передбачений засобами нижніх рівнів, і навіть зворотне перетворення вхідних даних із загального формату на формат, зрозумілий одержує програмі.

Прикладний рівень (Application layer) надає високорівневі функції мережевої взаємодії, такі як передача файлів, надсилання повідомлень електронною поштою тощо.

При рівневій організації процесу взаємодії повинні дотримуватися такі вимоги:

- компоненти одного рівня однієї системи можуть взаємодіяти з компонентами лише одного рівня іншої системи;

- у межах однієї системи компоненти будь-якого рівня можуть взаємодіяти лише з компонентами суміжних (вищележачого і нижчележачого) рівнів.

Рисунок 1.7 – Порядок рівневої взаємодії

Набір правил, визначальних порядок взаємодії коштів, які стосуються одному й тому рівні і що у різних системах, називається протоколом (protocol). Правила взаємодії між собою засобів, що належать до суміжних рівнів та функціонують в одній системі, називають інтерфейсом (interface).

На практиці протоколи та інтерфейси регламентують технічні вимоги до програмних та апаратних засобів. Програмні (апаратні) модулі, призначені для забезпечення практичної взаємодії, що визначається тим чи іншим протоколом (або інтерфейсом), зазвичай називають реалізацією протоколу (або інтерфейсу).

Хоча різні компоненти, що стосуються різних рівнів мережевої моделі формально повинні бути функціонально незалежними один від одного, при практичній розробці протоколів така незалежність не завжди витримується. Це тим, що спроба домогтися точного відповідності еталонної моделі може призвести до неефективності роботи програмно-апаратного забезпечення, реалізує протокол. В даний час спостерігається два типи відхилень, що виникають при реалізації рівневої взаємодії:

функції деяких рівнів можуть об'єднуватися одним протоколом і навпаки, функції одного рівня можуть ділитися між різними протоколами;

- функціонування протоколу будь-якого рівня мають на увазі використання лише певних протоколів нижчого рівня.

Тому розробка практичних методів мережевої взаємодії, як правило, передбачає розробку окремих протоколів, а цілих наборів протоколів. Такі набори зазвичай включають протоколи, що відносяться до кількох суміжних рівнів еталонної моделі OSI, і називаються стеками (або сімействами, наборами) протоколів (protocol stack, protocol suite). Найбільш відомим стеком протоколів, що забезпечує взаємодію в мережі Інтернет, є стек протоколів TCP/IP

Оскільки за реалізації протоколів допускаються відхилення від еталонної моделі, стеки протоколів можуть передбачати власну схему поділу рівні. Зокрема, стек протоколів TCP/IP поділяє весь процес мережевої взаємодії чотирма рівня. На цьому малюнку показано відповідність рівнів моделі OSI і рівнів стека TCP/IP.

2 ОПИС АЛГОРИТМУ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЗАЄМОДІЇ ТА НАЛАШТУВАННЯ РОУТЕРА

2.1. Метод доступа першого рівня

У наш час безлімітним інтернетом нікого не здивуєш, тому що й велика швидкість його роботи. Люди масово купують планшети, ноутбуки та смартфони, які мають вбудований модуль Wi-Fi , який дозволяє користуватися бездротовим інтернетом по всьому будинку. А дехто просто хоче позбутися проводів по всій квартирі . В обох випадках на допомогу прийшли Wi-Fi роутери, які дозволяють легко роздавати інтернет по всьому будинку. Про це й поговоримо далі.

Рисунок 2.1 – Роутер TP- Link TL-WR841N

Підключення TP- Link TL-WR841N

1. Принесли Ви роутер додому, або в офіс, це не важливо, відкриваємо коробочку і знаходимо там багато папірців, диск на якому інструкція та майстер з налаштування роутера. Так само в комплекті йде звичайно сам роутер, якщо ні, то Вас обдурили :), мережевий кабель для підключення його до комп'ютера і блок живлення, начебто все.

Рисунок 2.2 – Склад TP- Link TL-WR841N

2. Підключаємо роутер до комп'ютера. Тут усе дуже просто. Ставимо роутер неподалік комп'ютера, просто кабель в комплекті йде не дуже довгий, якщо потрібно, то можна обтиснути більше кабелю (це навіть можна зробити самому, тут http://f1comp.ru/sovety/kak-sdelat-obzhat-krossover/детальніше). Зробити це можуть практично в будь-якому комп'ютерному магазині.

Підключаємо до роутера живлення та вмикаємо його в розетку. Потім підключаємо в синє гніздо WAN інтернет-кабель. У роутері TP- Link TL-WR841N є 4 LAN порти, це означає, що можна підключити 4 комп'ютери по мережному кабелю.

З'єднуємо комп'ютер із роутером за допомогою кабелю, що йде в комплекті. Ось картинки:

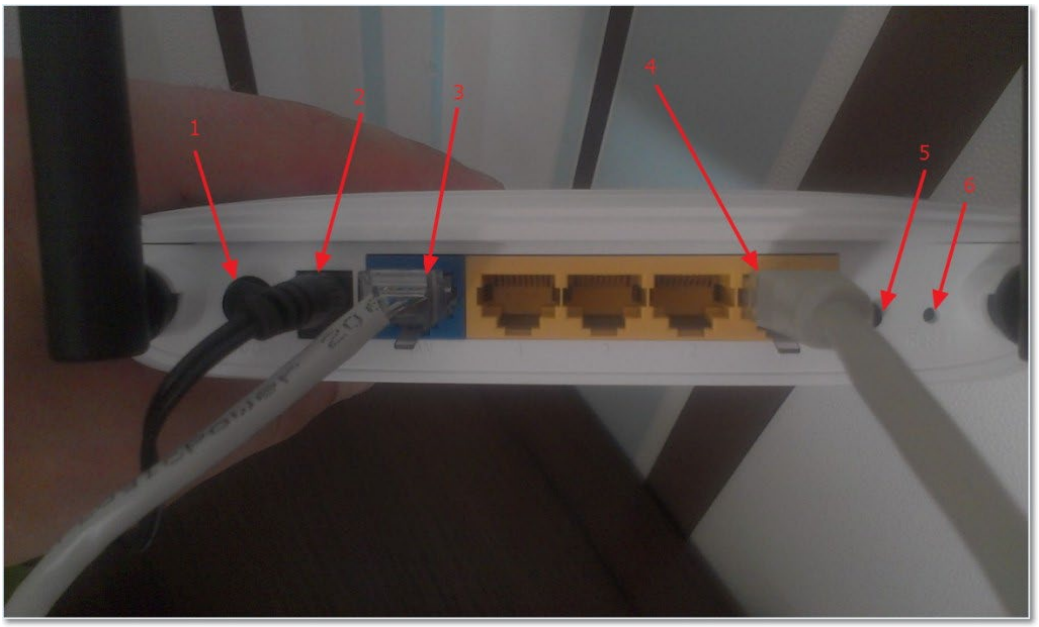

Рисунок 2.3 – Підключення TP- Link TL-WR841N

Давайте швиденько пройдемося по кнопках та роз'ємах:

1. Кнопка увімкнення/вимкнення.
2. Кабель електромережі.
3. Роз'єм WAN, для підключення інтернету.
4. Роз'єм LAN для підключення роутера до комп'ютера через мережний кабель.
5. Увімкнення функції QSS .
6. Кнопка для скидання налаштувань роутера.

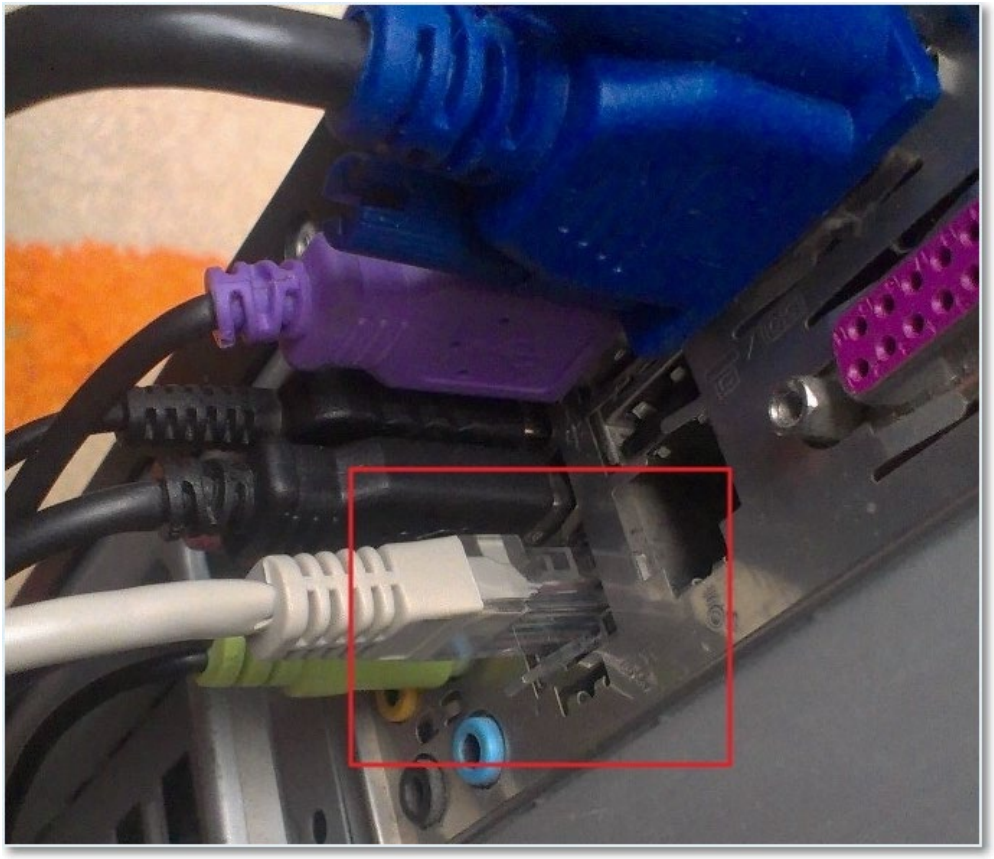

Рисунок 2.4 – Підключення до ПК

Налаштування роутера TP- Link TL-WR841N

Перед початком налаштування я раджу зробити скидання налаштувань .

Для налаштування роутера відкриваємо будь-який браузер, і в адресному рядку пишемо 192.168.0.1 , зазвичай проходить 192.168.1.1 , але мені вдалося отримати доступ до налаштувань тільки через 192.168.0.1. Вже тільки після налаштування оновлення оновлення доступу до налаштувань я отримую по 192.168.1.1.

З'явиться вікно, в якому потрібно ввести логін та пароль для доступу до налаштувань роутера. За замовчуванням логін - admin і пароль - admin .

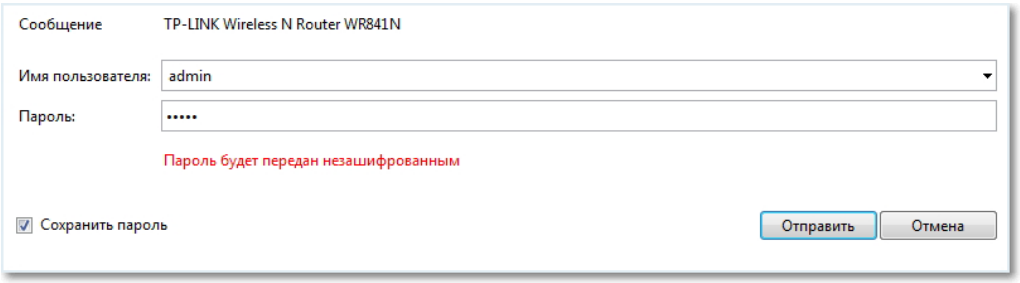

Рисунок 2.5 – Вхід в адмінку

Потрапляємо на сторінку налаштування.

Давайте для початку оновимо прошивку на нашому TP- Link TL-WR841N. Для цього її спочатку потрібно завантажити із сайту tp-linkru.com . Знаходимо для нашої моделі та завантажуємо останню версію. Розархівуємо файл прошивки на комп'ютер і повертаємось до налаштування.

Заходимо в меню « System Tools » і вибираємо « Firmware Upgrade » . Потім натискаємо «Огляд» , вибираємо завантажений нами файл і натискаємо « Upgrade » . Чекаємо, поки роутер оновить прошивку і перевантажиться.

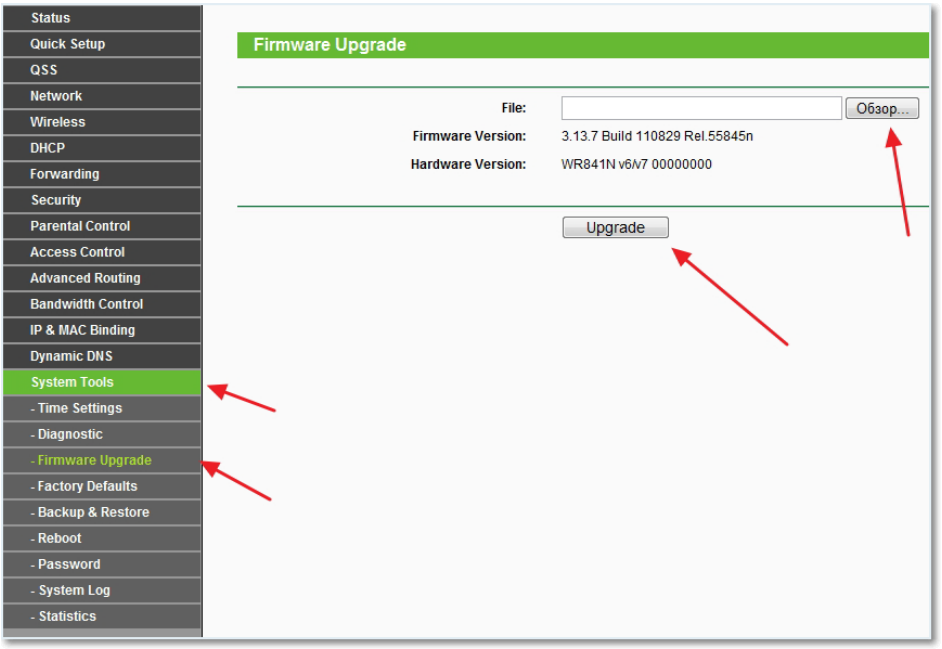

Рисунок 2.6 – Потрапляємо на сторінку налаштування

Продовжуємо налаштування. Давайте поміняємо логін та пароль для входу в налаштування роутера. Заходимо на вкладку « System Tools » , а потім « Password » . заповнюємо всі поля та натискаємо « Save » .

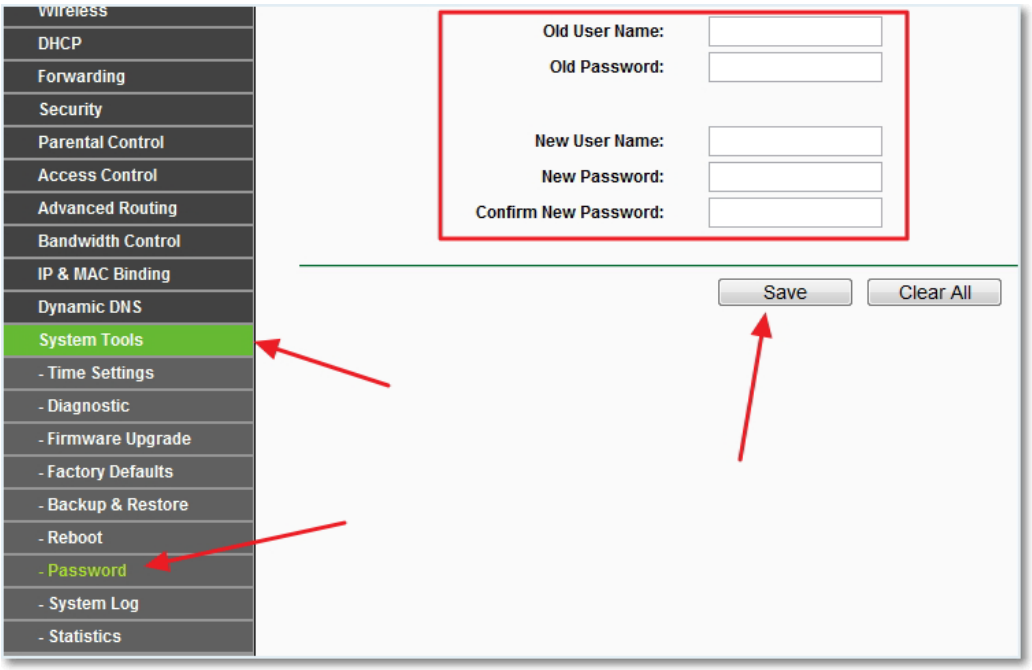

Рисунок 2.7 – Налаштування інтернету на TP- Link TL-WR841N

Налаштування інтернету на TP- Link TL-WR841N

Заходимо в " Network " і "WAN" . Тут потрібно вибрати тип мережі. Якщо не знаєте, що поставити, то зателефонуйте і запитайте у свого провайдера.

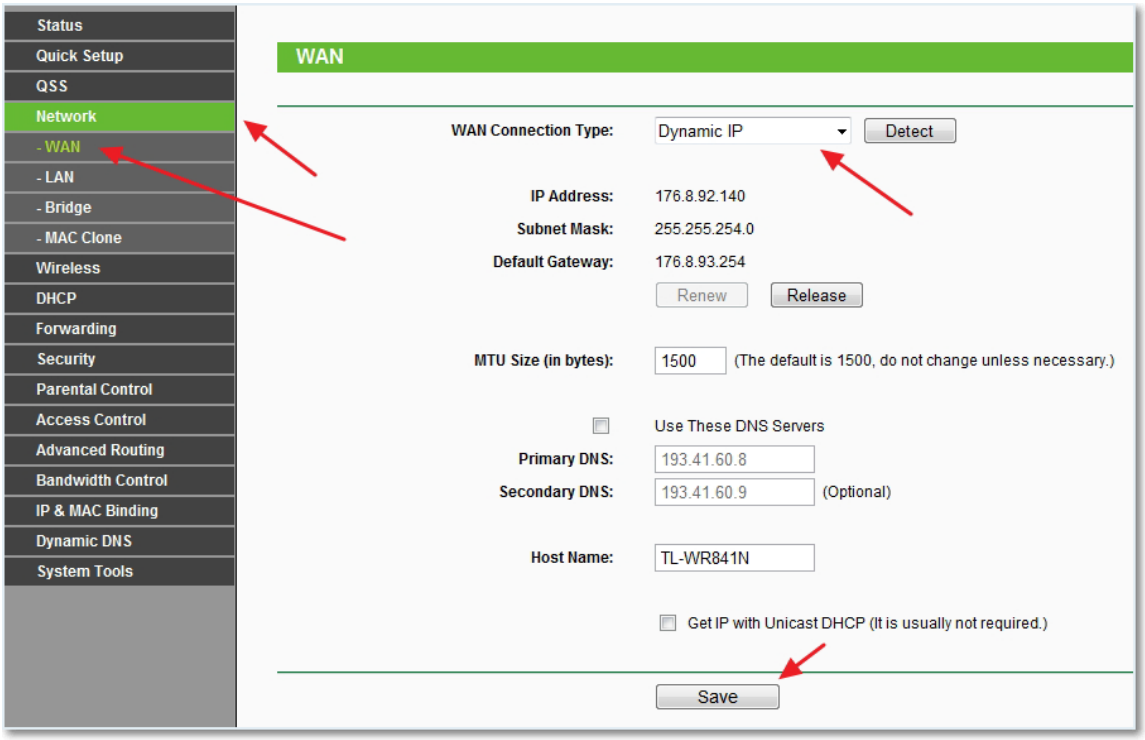

Рисунок 2.8 – Налаштування інтернету на TP- Link TL-WR841N

Натискаємо Save рухаємося далі . Тут же переходимо на вкладку MAC Clone натискаємо кнопочку Clone MAC Address і Save .

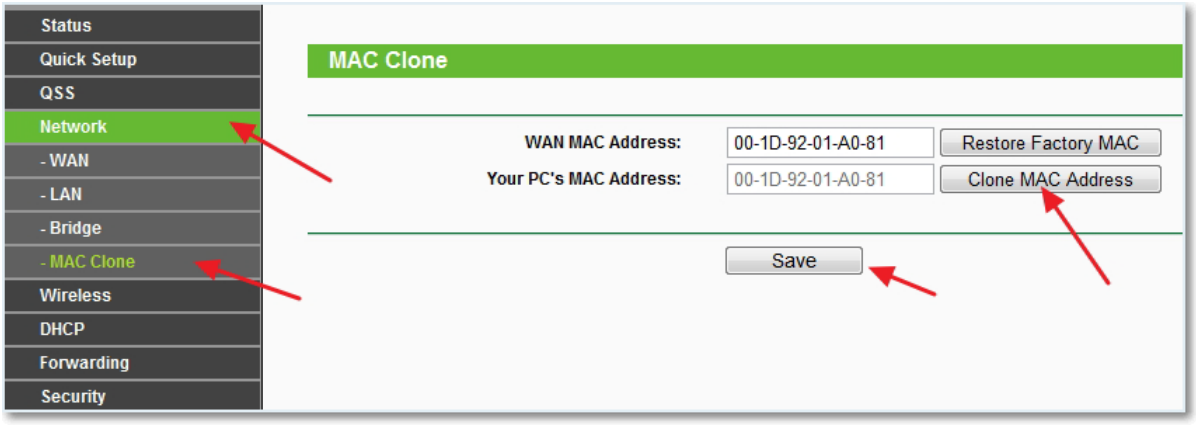

Рисунок 2.9– Налаштування MAC на TP- Link TL-WR841N

Налаштування Wi-Fi мережі на TP- Link TL-WR841N

Йдемо на вкладку Wireless і налаштовуємо наступні параметри. У полі « Wireless Network Name » пропишете назву Wi-Fi мережі. Трохи нижче можна вибрати регіон , де ви живете.

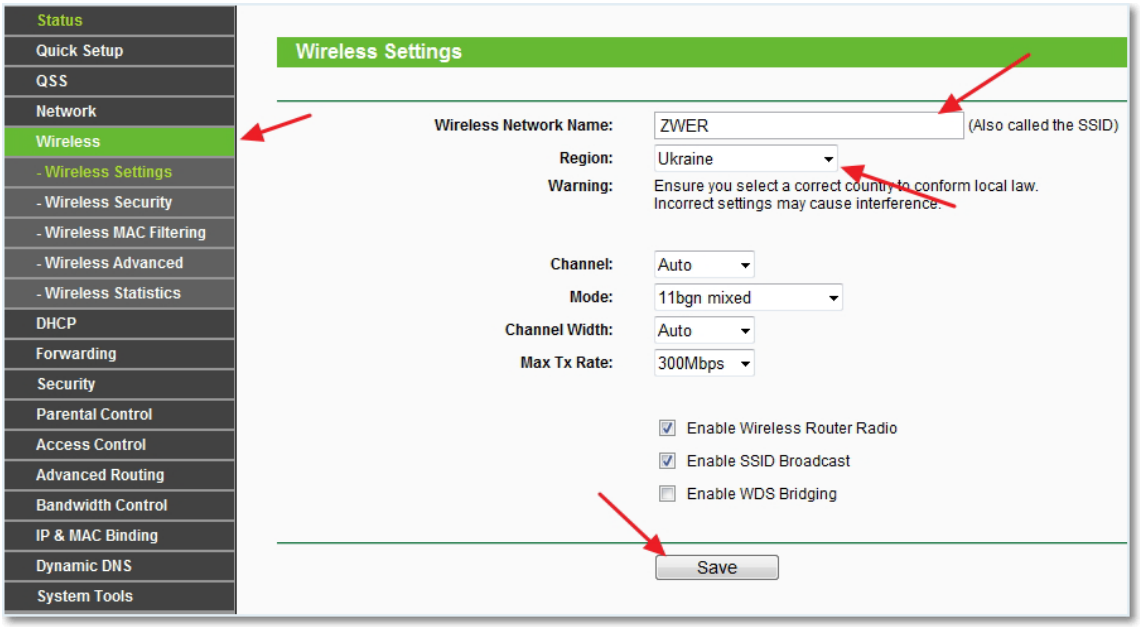

Рисунок 2.10– Налаштування інтернету на TP- Link TL-WR841N

Не забуваємо натиснути Save і переходимо на вкладку Wireless Security » . Це найголовніша сторінка, на ній ми налаштуємо параметри безпеки нашої мережі Wi-Fi .

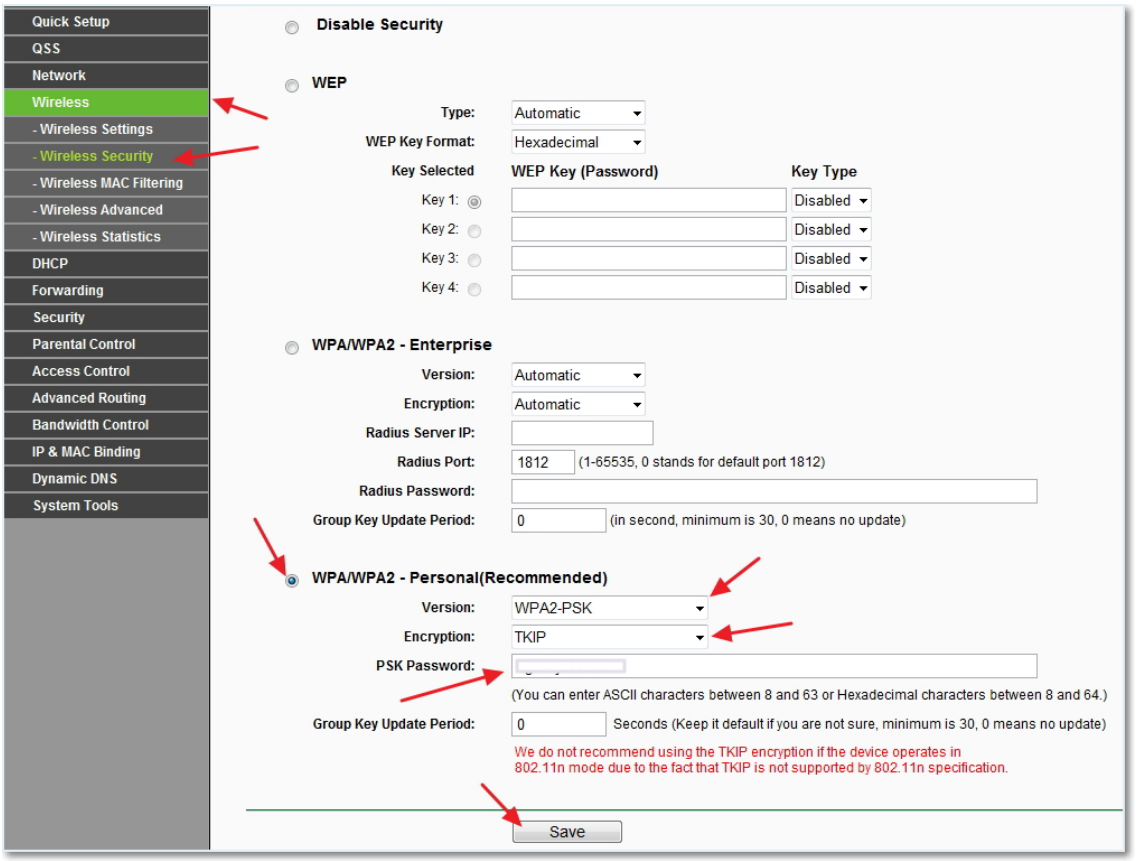

Рисунок 2.11– Налаштування інтернету на TP- Link TL-WR841N

Виставляємо все, як на скріншоті вище. У полі PSK Password вигадуємо та вписуємо пароль, який використовуватиметься для підключення до вашої Wi-Fi мережі.

Зберігаємо наші налаштування кнопкою Save . Налаштування закінчено, тепер перезавантажимо наш роутер. Для цього переходимо на вкладку « Tools » , а потім « Reboot » . Натискаємо на кнопку " Reboot " і підтверджуємо перезавантаження.

2.2 Модифікації методу доступа

Методика – сукупність дій задля досягнення мети. В даному випадку, це сукупність дій, що дозволяє здійснити перехоплення мережевого трафіку, а значить для досягнення мети потрібно виконати ряд пунктів [1-5]:

1. Визначити мету перехоплення;
2. Визначити застосовувані канали зв'язку;
3. Визначити, чи потрібна фільтрація трафіку;
4. Вибрати обладнання для віддаленого сніфера;

5. Вибрати спосіб перехоплення трафіку

Друга методика – адміністрування у мережі. на малюнку 3 наведено приклад збору трафіку в корпоративній мережі. Для реалізації методу необхідний окремий сервер, який збирає дані, до якого звертається адміністратор для аналізу отриманих результатів та складання точної картини стану мережі.

Адміністрування комп'ютерних мереж передбачає перехід від управління роботою окремих пристроїв до аналізу трафіку різних мережевих ділянках, управлінню логічної конфігурацією мережі, її робочими параметрами. Таким чином, завдання адміністрування можна розбити на дві основні групи:

• контроль за функціонуванням мережевого оборудования;

• керування роботою мережі загалом.

Ключовою метою стає досягнення та підтримання параметрів роботи інформаційної системи, що найбільш точно відповідають потребам користувача, для цього потрібно оцінити її роботу за характеристиками трафіку, використовуваними протоколами, швидкістю відгуку сервера на запити та особливостями сценаріїв.

Канали передачі даних комп'ютерних мереж - комплекс пристроїв для обміну потоками інформації в обох напрямках. Він складається з обладнання та ліній. Канали з'єднують прилади, що уловлюють сигнали з інформаційними вузлами.

Залежно від фізичного середовища передачі даних лінії зв'язку можна поділити на:

– дротові лінії зв'язку без ізолюючих та екрануючих обплетень;

– кабельні, де для передачі сигналів використовуються такі лінії зв'язку як кабелі "кручена пара", коаксіальні кабелі або оптоволоконні кабелі;

– бездротові (радіоканали наземного та супутникового зв'язку), що використовують для передачі сигналів електромагнітні хвилі, які поширюються по ефіру.

Звичні кабелі активно витісняють оптико-волоконні мережі. Його структура дозволяє передавати сигнали в рази швидше та точніше. Це відбувається за рахунок наявності елементів намагніченого кремнію, які обрамлені

матеріалом, що заломлює світло. Ними проходять світлові коливання, у яких трансформовані електромагнітні хвилі. Оптиковолоконне спорядження захищено вказівками, що санкціонують, тому така лінія надійна.

Зв'язок бездротовими каналами забезпечується декількома різними способами:

– електромагнітні канали (Wi-Fi хвилі);

– супутниковий зв'язок, забезпечений системою антен, що уловлюють сигнали та транслюють їх до наземних об'єктів прийому; – канали радіомовлення з дальністю сигналу не більше 50 км;

– стільникові радіоканали, сформовані системою стаціонарних приладів та мобільної апаратури. Дальність не обмежена певною відстанню, оскільки сигнали обробляються скрізь, де є апаратура;

– мультиканали з радіусом 60 км;

– Bluetooth — Надсилання даних на короткій відстані безкоштовно.

– Конкретний тип каналу корпоративного зв'язку створюється з урахуванням параметрів мережі та вимог замовника.

У корпоративній мережі найчастіше використовується кабельні та бездротові лінії зв'язку. Особливо уразливі зараз саме бездротові зв'язки.

При адмініструванні необхідність фільтрації трафіку вирішується за допомогою поставленої мети. Наприклад, фільтрація трафіку може здійснюватися за протоколами різних рівнів TCP/IP, що передаються. Так, використовуючи два способи передачі даних – дротовий та бездротовий, дані можна фільтрувати за протоколами Ethernet та 802.11.

Можна фільтрувати трафік по джерелу, його IPv4 або IPv6, або визначати фільтрацію відразу по VLAN.

LAN — у розшифровці «Локальна Обчислювальна Мережа» (Local Area Network) позначає з'єднання за допомогою проводового або бездротового зв'язку обчислювальних пристроїв з розміщенням в обмеженому (тобто локальному) територіальному просторі.

Локальна мережа (LAN) потрібна для спільного використання ресурсів (принтер, файлове сховище, обмін даними, загальний доступ до Інтернету та інші). Система масштабується і налаштовується як для двох-трьох

користувачів у домашніх умовах, так і для кількох тисяч робочих столів у великих організаціях, коли з'єднання виконується не тільки в офісі, а й між кількома будинками (наприклад, у наукових лабораторіях чи штаб-квартирах корпорацій).

Мінімальний склад LAN:

– комп'ютери,

– сервери - комп'ютер, виділений із групи персональних комп'ютерів (чи робочих станцій) до виконання будь-якої сервісної завдання без безпосередньої участі людини. Сервер і робоча станція може мати однакову апаратну конфігурацію, оскільки різняться лише з участі у роботі людини за консоллю.

– маршрутизатори – це пристрій, що з'єднує мережі різного типу, але використовують одну операційну систему. Це, по суті, той самий міст, який має свою мережеву адресу. Використовуючи можливості адресації маршрутизаторів, вузли мережі можуть надсилати маршрутизатору повідомлення, призначені для іншої мережі. Для пошуку найкращого маршруту до будь-якого адресата в мережі використовуються таблиці маршрутизації. Ці таблиці можуть бути статичними та динамічними.

– Комутатори - пристрій, призначений для підключення кількох вузлів комп'ютерної мережі в межах одного або кількох сегментів мережі. Комутатор працює на канальному (другому) рівні мережевої моделі OSI. Комутатори були розроблені з використанням мостових технологій та часто розглядаються як багатопортові мости. Для з'єднання кількох мереж на основі мережного рівня служать маршрутизатори (3 рівень OSI).

– додаткове обладнання (принтери, системи резервного копіювання тощо)

Для побудови локальної обчислювальної мережі будинку, в офісі або на великому підприємстві, потрібні комутаційні пристрої з десятком типів призначень від посилення сигналу до безпеки даних всередині ЛОМ.

Cisco IOS — це програмне забезпечення, яке використовується у маршрутизаторах та мережних комутаторах Cisco. Cisco IOS є багатозадачною операційною системою, що виконує функції мережевої організації, маршрутизації, комутації та передачі даних.

Junos – операційна система, яка використовується в устаткуванні компанії Juniper Networks. Інноваційно розроблена для простоти, є єдиною операційною системою, яка підтримує широкий портфель продуктів Juniper для фізичних і віртуальних мереж та забезпечення безпеки. Створений для забезпечення надійності, безпеки та гнучкості, він підтримує одні із найскладніших мережевих розгортань у світі, що дає операторам конкурентну перевагу перед тими, хто використовує інші мережеві операційні системи.

Універсальна платформа маршрутизації (Versatile Routing Platform VRP) — це мережна операційна система, що використовується в мережевих пристроях Huawei, таких як маршрутизатори та комутатори. Він надає користувачам цих мережевих пристроїв узгоджену та потужну платформу конфігурації за рахунок стандартизації мережевих, користувальницьких та керуючих інтерфейсів.

Windows — група сімейств комерційних операційних систем корпорації Microsoft, орієнтованих управління з допомогою графічного інтерфейсу.

Unix - сімейство переносних, багатозадачних і розрахованих на багато користувачів операційних систем, які засновані на ідеях оригінального проекту AT&T Unix, розробленого в 1970-х роках.

Як очевидно з опису устаткування, для методики адміністрування мережі, підходять відразу кілька способів перехоплення трафіку. Насамперед, при побудові мережі з вищезгаданих пристроїв, таких як комутатор і маршрутизатор, варто звернути знімання на здатність конкретного обладнання можливості дзеркалізації трафіку. ОС кожної системи, що розглядаються, побудовані на основі BDS UNIX. BSD (Berkeley Software Distribution) - система поширення програмного забезпечення у вихідних кодах. Особливістю пакетів ПЗ BSD була спеціальна ліцензія BSD, яку коротко можна охарактеризувати так: весь вихідний код - власність BSD, всі правки - власність їх авторів.

Тому, якщо пристрій не може віддзеркалювати трафік, на допомогу йде бібліотека libpcap спільно з демоном RPCAPD. Варто врахувати, що для цього, можливо, доведеться змінити код під ядро ОС, що використовується на мережному пристрої.

У разі використання перехоплення трафіку на робочих станціях відмінно підходить використання способу перехоплення демоном RPCAPD. Також, якщо адміністратор захоче сканувати радіоефір бездротової мережі, він може виділити для цього окремий пристрій для сканування, що передає інформацію на окремий сервер, що обробляє та аналізує трафік.

Таким чином, виходячи з вищесказаного, отримаємо, що для реалізації методу необхідний окремий сервер, який збирає дані, до якого звертається адміністратор для аналізу отриманих результатів та складання точної картини стану мережі. Приклад ЛОМ, що складається з Робочих станцій, комутаторів і сервера для збору та аналізу даних представлений на малюнку 53. На ньому також зображено потік трафіку, що перехоплюється, а саме, з пристрою 1, 2 і 3 перехоплюваний трафік збирається на сервері. Після цього результат оброблених даних відправляється Адміністратору мережі, щоб виявити нестандартну активність.

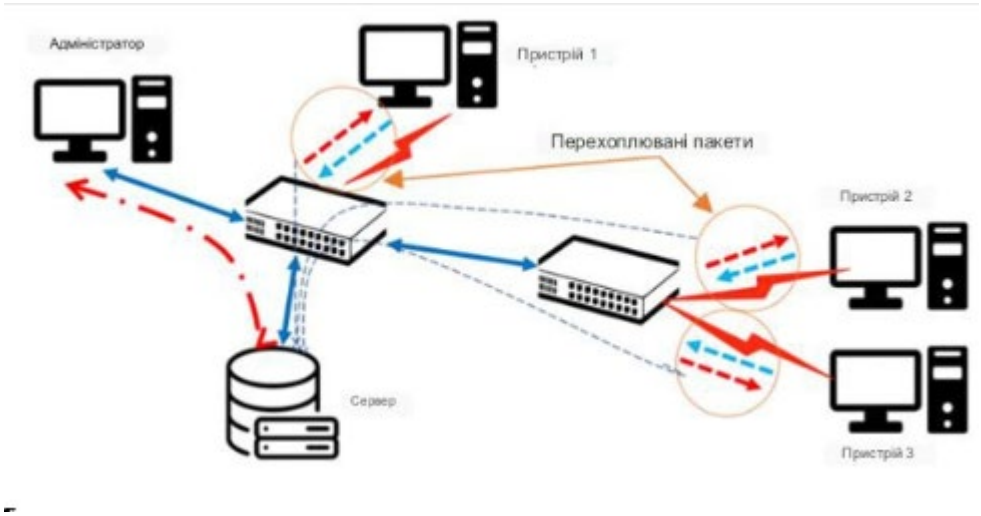

Рисунок 2.12 – Схема прикладу перехоплення трафіку при адмініструванні мережі

Для реалізації перехоплення трафіку адміністратору необхідно виконати такі кроки:

1. Вивчити схему мережі;

2. Визначитися з методами віддаленого перехоплення та пристроями, з яких здійснюватиметься перехоплення;

3. Визначитися з місцем збору даних, що перехоплюються;

4. Налаштувати мережу, згідно з обраною методикою;

5. Почати захоплення трафіку;

6. Вивчити отриманий результат та зробити висновки про стан мережі.

Докладно розбираючи методику, пов'язану з організацією віддаленої навчальної лабораторної, розглянемо варіант її побудови, коли студенти мають віддалений доступ до локальної мережі по якому-небудь тунелю.

Вивчення способів перехоплення трафіку і даних, що містяться в інформації, що передається. Студент може навчитися визначати сигнатури атак, описувати їх та застосовувати отримані знання на практиці.

У рамках цієї методики розглядається підготовка лабораторних робіт до віддаленого формату, включаючи специфічні роботи, пов'язані з бездротовими мережами. Тому основними каналами передачі даних є бездротові мережі та тунелювання з використанням будь-яких видів каналів зв'язку, наприклад провідних.

Бездротова мережа необхідна для вивчення радіоефіру з метою навчання. Тунелювання, у свою чергу, допоможе забезпечити доступ до лабораторної установки університету для студента, що знаходиться поза стінами.вузу.

Фільтрація трафіку з метою може бути необхідна для дослідження мережі. Наприклад, можна забезпечити фільтрацію за конкретними пристроями, цим проводити аналіз не кожного пристрою, а лише певного.

Для відтворення лабораторної установки може знадобитися ПК, з встановленою на ньому Linux або з віртуальною машиною з тією ж ОС. Завдяки ОС Linux у студента з'явиться можливість просканувати в режимі «Monitoring» радіоефір та отримати заголовки стандарту 802.11.

Також для вивчення бездротової мережі буде потрібний Wi-Fi адаптер з можливістю переходу в вищезгаданий режим.

Крім цього, необхідно забезпечити доступ установки до зовнішньої мережі для створення VPN-тунелю.

Для вибору способу перехоплення у разі можна використовувати таблицю, описану раніше. Ця методика може бути реалізована з використанням RPCAPD-

демону. Крім цього, метод використання RPCAPD зручніший, тому що підтримується різними операційними системами і не прив'язаний до виробників.

На Рисунок 2.11 а представлений варіант віддаленого підключення до лабораторної установки з використанням тунелю.

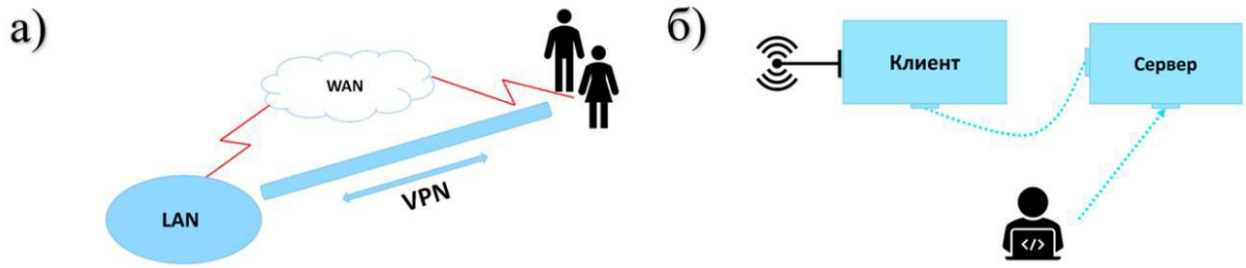

Рисунок 2.13. Сценарій реалізації віддаленого перехоплення трафіку під час виконання лабораторних робіт

На рисунку 2.13 наведено приклад віддаленої лабораторної установки. Підключаючись до локальної мережі, студенти мають доступ до сервера. Сервер відправляє запит на клієнта, де стоїть інтерфейс, з якого збирається трафік.

Таким чином, студенту необхідно виконати такі дії:

1. На своїй стороні (серверній) забезпечити з'єднання VPN з лабораторною установкою (клієнтом);

2. Встановити на своєму ПК будь-який сніффер, наприклад, Wireshark, Tcpdump, Tshark і т.д.

3. Знайти за допомогою додаткових утиліт, наприклад, за допомогою ifconfig на linux, додатково підключені мережні пристрої, такі як бездротовий адаптер;

4. Встановити на стороні клієнта демон RPCAPD;

5. Налаштувати наявний бездротовий адаптер, перевівши його в режим "Monitoring";

6. Почати віддалене захоплення трафіку, що йде в радіоефірі з бездротового інтерфейсу клієнта, на стороні сервера.

7. Вивчити отриманий трафік та зробити висновки.

2 3. Тестування модифікованого

Особливості

Створіть AP (точку доступу) на будь-якому каналі.

Виберіть одне з таких шифрувань: WPA, WPA2, WPA/WPA2, Open (без шифрування).

Приховайте свій SSID.

Вимкнути зв'язок між клієнтами (ізоляція клієнта).

Підтримка IEEE 802.11n і 802.11ac

Методи спільного доступу до Інтернету: NATed, Bridged або None (без доступу до Інтернету).

Виберіть IP-адресу шлюзу точки доступу (лише для методів спільного доступу до Інтернету «NATed» і «None»).

Ви можете створити точку доступу з тим самим інтерфейсом, що й підключення до Інтернету.

Ви можете передати свій SSID і пароль через канал або через аргументи (див. приклади).

Залежності

Загальний

bash (для запуску цього сценарію)

util-linux (для getopt)

procps або procps-ng

hostapd

iproute2

iw

iwconfig (це потрібно лише якщо 'iw' не може розпізнати ваш адаптер)

haveged (необов'язково)

Для методу спільного використання Інтернету «NATed» або «None».

dnsmasq

iptables

монтаж

загальний

git клон https://github.com/lakinduakash/linux-wifi-hotspot

cd linux-wifi-hotspot/src/scripts

зробити інсталяцію

ArchLinux

pacman -S create_ap

Gentoo

виникнути мирянин

мирянин -f -a jorgicio

emerge net-wireless/create_ap

Приклади

Без парольної фрази (відкрита мережа):

create_ap wlan0 eth0 MyAccessPoint

Парольна фраза WPA + WPA2:

create_ap wlan0 eth0 MyAccessPoint MyPassPhrase

AP без доступу до Інтернету:

create_ap -n wlan0 MyAccessPoint MyPassPhrase

Спільне використання Інтернету по мосту:

create_ap -m bridge wlan0 eth0 MyAccessPoint MyPassPhrase

Мостовий спільний доступ до Інтернету (попередньо налаштований інтерфейс мосту):

create_ap -m bridge wlan0 br0 MyAccessPoint MyPassPhrase

Спільне використання Інтернету з одного інтерфейсу WiFi:

create_ap wlan0 wlan0 MyAccessPoint MyPassPhrase

Виберіть інший драйвер адаптера WiFi

create_ap --driver rtl871xdrv wlan0 eth0 MyAccessPoint MyPassPhrase

Немає парольної фрази (відкрита мережа) за допомогою каналу:

echo -e "MyAccessPoint" | create_ap wlan0 eth0

Парольна фраза WPA + WPA2 з використанням каналу:

echo -e "MyAccessPoint\nMyPassPhrase" | create_ap wlan0 eth0

Увімкніть IEEE 802.11n

create_ap --ieee80211n --ht_capab '[HT40+]' wlan0 eth0 MyAccessPoint MyPassPhrase

Ізоляція клієнта:

create_ap --isolate-clients wlan0 eth0 MyAccessPoint MyPassPhrase

Служба Systemd

Використання постійної служби systemd

Негайно розпочати обслуговування:

systemctl запустити create_ap

Почати під час завантаження:

systemctl увімкнути create_ap

Загальний

util-linux (для getopt)

procps або procps-ng

hostapd

iproute2

iw

iwconfig (це потрібно лише якщо 'iw' не може розпізнати ваш адаптер)

haveged (необов'язково)

Переконайтеся, що у вас є ці залежності, ввівши їх у терміналі. Якщо будь-яка із залежностей не вдається встановити її за допомогою менеджера пакунків дистрибутива

Для методу спільного використання Інтернету «NATed» або «None».

dnsmasq

iptables

Для створення з джерела

зробити

gcc і g++

побудувати необхідний

pkg-config

gtk

libgtk-3-dev

libqrencode-dev (для генерації qr-коду)

libpng-dev (для генерації qr-коду)

На Ubuntu або debian інсталюйте залежності за допомогою,

sudo apt install -y libgtk-3-dev build-essential gcc g++ pkg-config make hostapd libqrencode-dev libpng-dev

На Fedora/CentOS/Red Hat Enterprise Linux/Rocky Linux/Oracle Linux

sudo dnf install -y gtk3-devel gcc gcc-c++ kernel-devel pkg-config make hostapd qrencode-devel libpng-devel

монтаж

git клон https://github.com/lakinduakash/linux-wifi-hotspot

cd linux-wifi-hotspot

#побудувати двійкові файли

зробити

#встановити

sudo make install

Видалення

sudo make uninstall

Біг

Ви можете запустити GUI, знайшовши «Wifi Hotspot» у меню програм або використовуючи термінал за допомогою:

wihotspot

Запустити під час запуску

Графічний інтерфейс wihotspot використовує create_ap для створення точок доступу та керування ними. Цю службу та основну логіку спочатку створив @oblique, і тепер вони зберігаються в цьому репозиторії.

Запустіть службу точки доступу під час запуску (використовуючи збережену конфігурацію) за допомогою:

systemctl увімкнути create_ap.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Компоненти програми

У розробці програми для даного проекту була обрана мова програмування Pascal. Цей вибір має кілька переваг, які впливають на якість та продуктивність застосунку:

По перше, Lazarus використовує мову програмування Pascal/Object Pascal, яка підтримує об'єктно-орієнтований підхід до розробки програм. Це дозволяє створювати структуровані та модульні компоненти, що полегшує розробку, тестування та підтримку застосунку.

По друге, мова програмування надає розширену бібліотеку компонентів для розробки графічного інтерфейсу користувача. Можливо швидко та легко створювати вікна, кнопки, текстові поля, таблиці та інші елементи інтерфейсу для програми.

По трете, Lazarus базується на фреймворку Free Pascal, що підтримує кросплатформену розробку. Можливо розробляти програму в Lazarus на одній платформі (наприклад, Debian) і компілювати її для різних операційних систем, таких як Windows, macOS, Linux і багатьох інших. Це забезпечує широку доступність застосунку для користувачів на різних платформах.

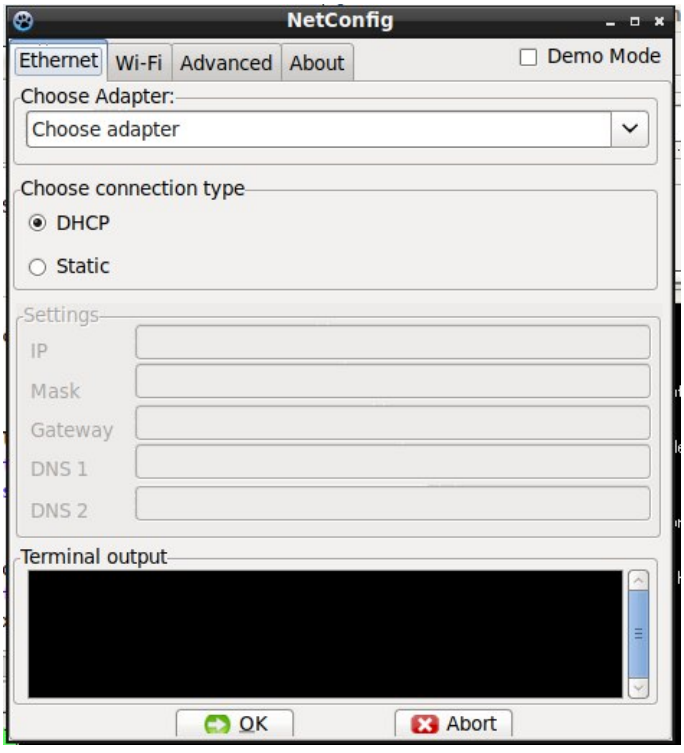

Рисунок 3.1. Головна сторінка програми

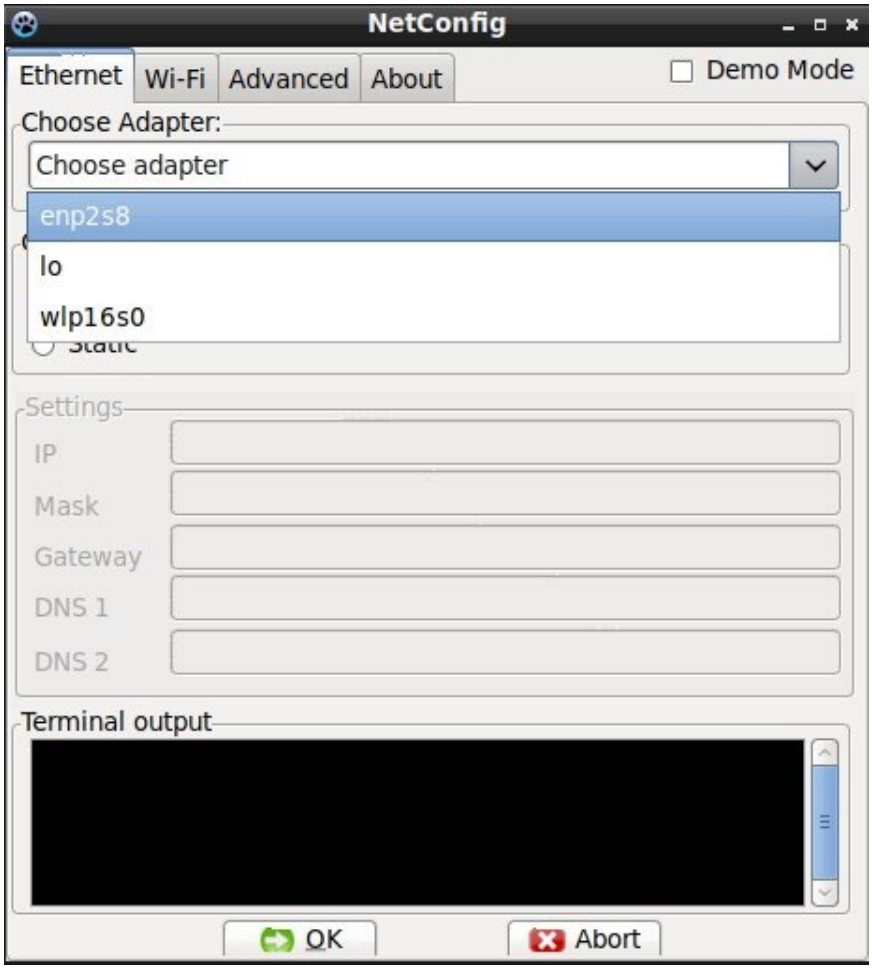

Рисунок 3.2. Головна сторінка програми з мережевими інтерфейсами

На рисунку 3.2 представлена головна сторіна програми де пропонуеться обрати адаптер котрий користувачеві потрібно налаштувати.

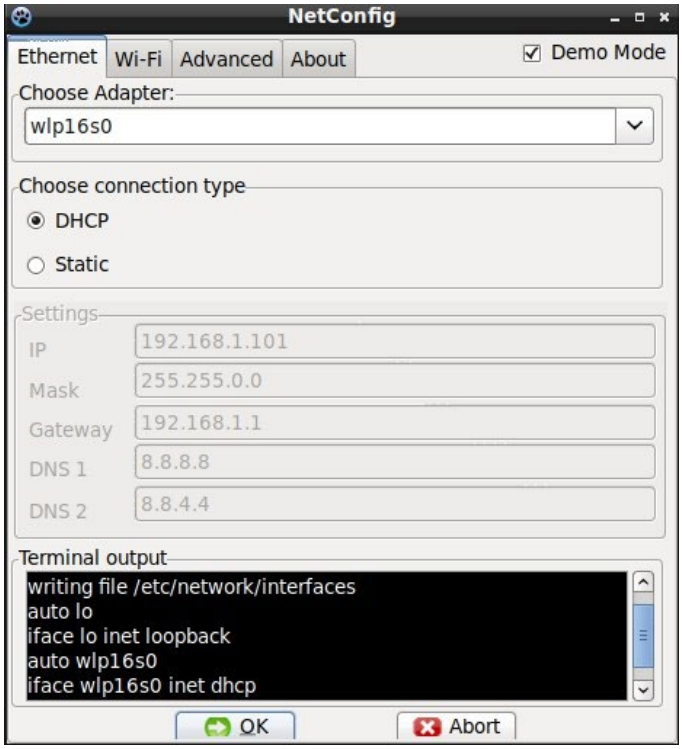

Рисунок 3.3. Повідомлення висновку терміналу

На рисунку 3.3 зображене успішне підключення нового мережевого з'єднання, обрано мережеве обладнання wlp16s0 тип підключення DHCP.

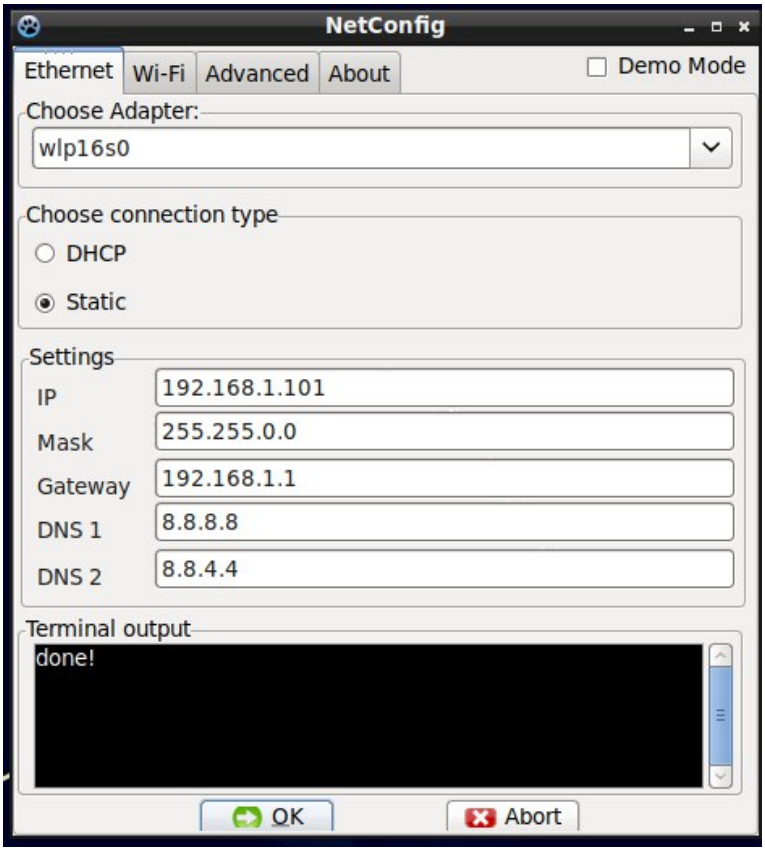

Рисунок 3.4. Повідомлення висновку терміналу

На рисунку 3.4 зображене ручне підключення Ethernet кабелю, обрано мережеве обладнання wlp16s0 статичний тип підключення.

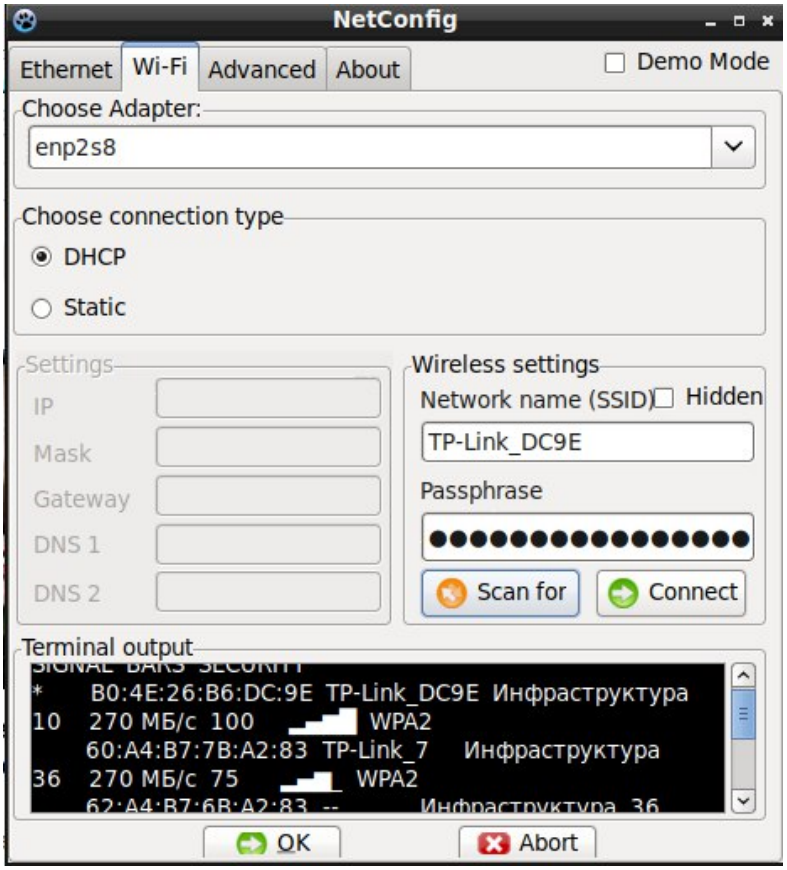

Рисунок 3.5. Підключення Wi-Fi

На рисунку 3.6 обрано новий мережевий адпаптер enp2s8, просканово усі досутпні Wi-Fi мережі які відображаються у термінали виводу. Зображено підключення до одної з Wi-Fi мереж.

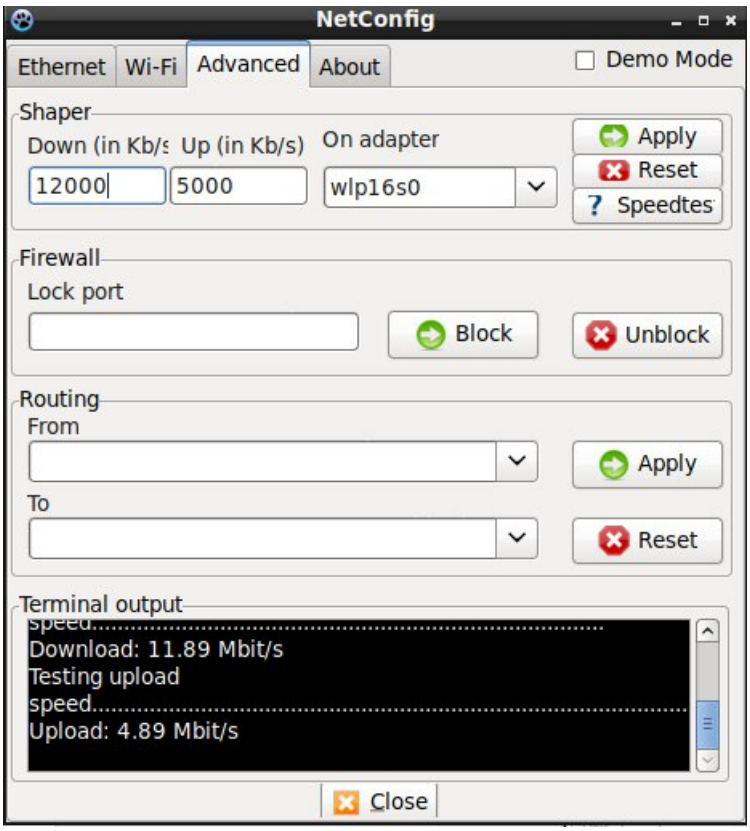

Рисунок 3.6. Вікно просунутих налаштувань

Функціонал зображений на рис. 3.6 відповідає за обмеження вхідної та вихідної швидкості. Присутній тест для перевірки швидкості з'єднання, та скидання налаштувань до первинних . Firewall який може блокувати порти на компьютері. Та функція роутера яка дозволяє компютеру роздавати інтернет якщо на копмьюетрі присутне відповідне мережеве обладнання.

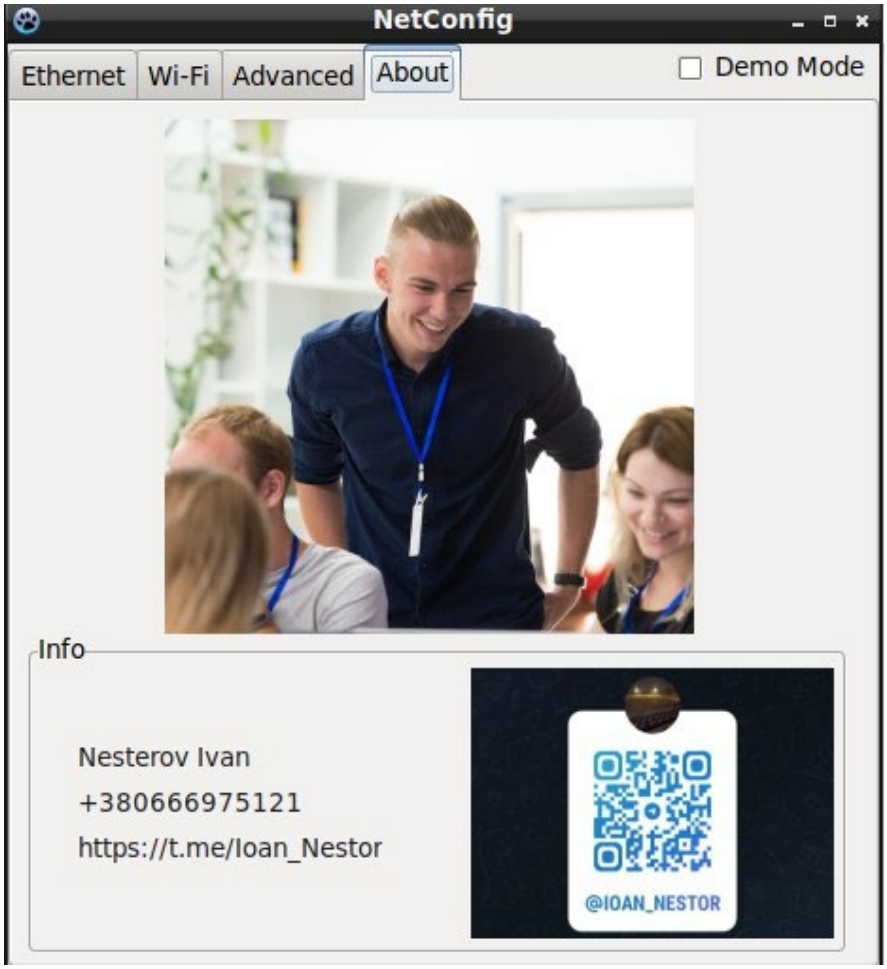

Рисунок 3.7. Інформація про розробника

3.3 Тестування розробленого програмного забезпечення

Команда	Результат	Помилки
Ввід даних	Успішно записано	Помилки було знайдено та виправлено
Зберігання даних	Успішно збережено	Відсутні
Зчитування даних	Успішно зчитано	Відсутні
Вивід даних	Успішно введено	Відсутні
Пошук даних	Пошук працює правильно	Відсутні
Редагування	Успішно відредаговано	Відсутні
Видалення	Успішно видалено	Помилки було знайдено та виправлено

Для використання програми потрібно всього 20 мб ОЗУ але радше використовувати з залишком в 100 мб, за для уникнення навантажень при завантаженні та зчитуванні великих файлів.

ВИСНОВКИ

У даній дипломній роботі була розроблена програма для швидкого мережевого з'єднання з використанням середовища розробки Lazarus та мови програмування Pascal/Object Pascal. Розроблений застосунок має декілька ключових функціональних можливостей, що дозволяють ефективно встановлювати та керувати мережевими з'єднаннями.

У процесі розробки була використана об'єктно-орієнтована архітектура, що дозволила організувати компоненти програми в логічні модулі з чіткою відповідальністю. Класи були створені для різних функцій, таких як встановлення мережевих параметрів, відправлення та отримання мережевих пакетів, а також для керування станом програми.

Для забезпечення графічного інтерфейсу користувача була використана бібліотека компонентів Lazarus, яка надала можливість швидко створити вікна, кнопки, текстові поля та інші елементи інтерфейсу. Це дозволило забезпечити зручний та інтуїтивно зрозумілий інтерфейс для користувачів програми.

Програма була розроблена з урахуванням кросплатформеності, що дозволило компілювати її для різних операційних систем, таких як Windows, macOS та Linux. Це забезпечило широку доступність програми для користувачів на різних платформах.

Застосунок був успішно протестований та виявив гарну продуктивність у встановленні та керуванні мережевими з'єднаннями. Він задовольняє поставлені вимоги щодо швидкості та надійності з'єднання.

СПИСОК ЛІТЕРАТУРИ

1. Ягудін І.Р., Волкогонов В.М., Аналіз мережевих атак: ARP-SPOOFING та DNS-SPOOFING. / Регіональна інформатика та інформаційна безпека - збірник наукових праць. / Санкт-Петербурзьке Товариство інформатики, обчислювальної техніки, систем зв'язку та управління. 2017 Сторінки: 329-332, УДК: 004.056

2. Нехань Є.Н, Гудков М.А., Дослідження проблеми прослуховування мережевого трафіку / Регіональна інформатика та інформаційна безпека – збірка наукових праць. / Санкт-Петербурзьке Товариство інформатики, обчислювальної техніки, систем зв'язку та управління. 2017. Сторінки: 140-142, УДК: 003.26 (075.8).

3. Мєшкова Є.В., Перехоплення та аналіз мережевого трафіку за допомогою «Wireshark», Пермський національний дослідницький політехнічний університет, Тип: стаття в журналі - наукова стаття Мова: російська Номер: 8 (49) Рік: 2016 Сторінки: 158-162 , УДК:

004.056.53

4. Ладигін П.С., Технологія перехоплення та аналізу трафіку в бездротовій wi-fi мережі, Алтайський державний університет, Тип: стаття в журналі - наукова стаття Мова: російська, Том: 2 Номер: 12 Рік: 2015 Сторінки: 102-105 , журнал: праці молодих вчених алтайського державного університету, ISSN: 2307-2628eISSN: 2686-8059

5. Кірєєв А.П., Колмиков Д.В., Михайлов С.Ю., Пепеляєв А.В., Аналіз мережевого трафіку корпоративної мережі через програмне забезпечення Wireshark, Омський державний технічний університет, Тип: стаття в журналі - наукова стаття Мова : російська, Номер: 3 Рік: 2019 Сторінки: 11-15 УДК: 004.7

6. Перехоплення даних через мережу [Електронний ресурс] - https://www.anti- malware.ru/threats/network-traffic-interception дата звернення: 03.02.2021р.

7. Сотников В.Д., Мельников В.М. Перехоплення та аналіз мережевого трафіку за допомогою бібліотеки PCAP. Воронезький державний університет,

Тип: стаття у збірнику праць конференції Мова: російська Рік видання: 2018, Сторінки: 203-208

8. Земляков П. Пасивний перехоплення трафіку. / СИСТЕМНИЙ АДМІНІСТРАТОР- Видавничий дім "Положевец и партнеры" (Москва) / ISSN: 1813-5579, Сторінки: 52-55, Рік: 2004.

9. Степанов П.П., Свалов А.А., Кобенко В.Ю., Гіль А.С., Методи перехоплення трафіку. Омський державний технічний університет, Омськ, Росія, Тип: стаття в журналі - наукова стаття Мова: російська, Том: 5, Номер: 1 Рік: 2018 Сторінки: 60-65, УДК:

004.7

10. Таргонський, А.І. Розробка захищеного веб-інтерфейсу для керування пристроями у мережі / А.І. Таргонський, А.Ю. Цвєтков // Актуальні проблеми

інфотелекомунікацій у науці та освіту. VIII Міжнародна науково-технічна та науково-методична конференція: зб. наук. ст. СПб.: СПбГУТ, 2019. С. 734-739.

11. Дослідження існуючих механізмів захисту операційних систем сімейства Linux/А.Ю. Цвєтков // Актуальні проблеми інфотелекомунікацій у науці та освіту. VII Міжнародна науково-технічна та науково-методична конференція: зб. наук. ст. у 4-х т. СПб.: СПбГУТ, 2018. С. 657-662.

12. Нурмахамет Д.Р. Аналіз VPN-протоколів: OPENVPN, PPTP, L2TP/IPSEC, IKEV2/IPSEC // Збірник статей за матеріалами LXXVI студентської міжнародної науково-практичної конференції. 2019

13. Темченко, В.І. Проектування моделі інформаційної безпеки в операційній системі/В.І. Темченко, О.Ю. Цвєтков // Актуальні проблеми інфотелекомунікацій у науці та освіту. VIII Міжнародна науково-технічна та науково-методична конференція: зб. наук. ст. СПб.: СПбГУТ, 2019. С. 740-745.

14. Багомедова А.Р., Ушаков І.А., Цвєтков А.Ю. Розробка методів перевірки відповідності серверів віртуалізації вимогам безпеки згідно зі стандартом ГОСТ Р 56938-2016 // Актуальні проблеми інфотелекомунікацій у науці та освіті (АПІНО 2018): збірка статей VII Міжнародної науково-технічної та науково-методичної конференції. 2018. С. 58-63.

1116130. 01320-01

Додатки

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Проректор Українського державного університету науки і технологій

Анатолій РАДКЕВИЧ

ПРОГРАМНИЙ ПРОДУКТ «Розробка застосунку для швидкого конфігурування мережевих пристроїв »

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ

1116130. 01320-01-ЛЗ

Представники

підприємства-розробника

Завідувач кафедри КІТ

Вадим ГОРЯЧКІН

Керівник розробки

Іван КЛИМЕНКО

Виконавець

Іван НЕСТЕРОВ

Норм-контролер

Світлана ВОЛКОВА

2023

ЗАТВЕРДЖЕНО

1116130. 01320-01-ЛЗ

ПРОГРАМНИЙ ПРОДУКТ « Розробка застосунку для швидкого конфігурування мережевих пристроїв »

Технічне завдання

1116130. 01320-01-ЛЗ

Листів 16

2023

ЗМІСТ

1 ВВЕДЕННЯ

Програмний продукт «Розробка застосунку для швидкого конфігурування мережевих пристроїв», що розробляється, має виконувати функції підключення і налаштування нових мережевих підключень.

Розробка застосунку для швидкого конфігурування мережевих пристроїв є сучасним програмним рішенням, призначеним для спрощення та автоматизації процесу налаштування мережевих пристроїв. Цей застосунок розробляється з метою надання користувачам зручного та ефективного інструменту для швидкого налаштування різноманітних мережевих пристроїв.

У сучасному мережевому середовищі все більше зростає складність і розмаїтість мережевих пристроїв, які потребують налаштування. Кожен пристрій може мати свої унікальні параметри, настройки та опції, і це може становити складність для адміністраторів мережі. Тому розробка застосунку для швидкого конфігурування мережевих пристроїв ставить за мету спростити цей процес та зменшити його часові затрати.

Основна мета застосунку полягає у наданні зручного інтерфейсу для налаштування мережевих пристроїв. Користувачі зможуть швидко та ефективно змінювати параметри, налаштування та опції пристроїв через інтуїтивно зрозумілий інтерфейс. Застосунок може включати різноманітні функції, такі як автоматичне виявлення пристроїв у мережі, перегляд та редагування конфігураційних файлів, встановлення безпекових політик та багато іншого.

У процесі розробки застосунку використовуються сучасні технології та інструменти для реалізації його функціональності. Мови програмування, такі як Lazarus, Object Pascal, можуть використовуватися для написання логіки програми. Крім того, використовуються мережеві протоколи та бібліотеки для взаємодії з мережевими пристроями.

Застосунок для швидкого конфігурування мережевих пристроїв призначений для адміністраторів мережі, інженерів з мережевого адміністрування та будь-яких фахівців, які відповідають за налаштування та управління мережевими пристроями. Він розробляється з метою полегшення та прискорення процесу конфігурування, забезпечуючи ефективну та надійну роботу мережі.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ від 08.12.21 №77ст ректора Українського державного університету науки і технологій “Про призначення наукових керівників та затвердження тем бакалаврських робіт” за спеціальністю 121 “Інженерія програмного забезпечення» факультету “Комп’ютерних технологій і систем” по кафедрі “Комп’ютерні інформаційні технології”.

Тема дипломної роботи – « Розробка застосунку для швидкого конфігурування мережевих пристроїв ». Керівник - Клименко І.В.

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення застосунку - полягає в створенні програмного рішення для швидкого конфігурування мережевих пристроїв. Цей програмний продукт має на меті полегшити процес налаштування та управління мережевими пристроями шляхом надання зручного та ефективного інтерфейсу.

Експлуатаційне призначення застосунку - забезпечення оперативного та ефективного конфігурування мережевих пристроїв. Він дозволяє адміністраторам мережі, інженерам з мережевого адміністрування та іншим фахівцям, відповідальним за мережеві налаштування, швидко налаштовувати та управляти мережевими пристроями з високою ефективністю. Застосунок забезпечує зручність використання та прискорює процес конфігурування, що дозволяє знизити час і зусилля, необхідні для роботи з мережевими пристроями.

4 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

4.1 Вимоги до функціональних характеристик

Функціональні вимоги системи:

- Система повинна дозволяти налаштовувати різні параметри мережевих пристроїв, такі як IP-адреси, мережеві протоколи, маршрутизація, VLAN.
- Застосунок повинен забезпечувати можливість переглядати поточний стан мережевих пристроїв, трафік, використання ресурсів.
- Застосунок повинен надавати можливість автоматизувати процес конфігурування мережевих пристроїв шляхом використання скриптів або шаблонів, що забезпечує швидкість та точність установки.
- Застосунок повинен забезпечувати захист конфігураційних даних мережевих пристроїв шляхом шифрування, аутентифікації та авторизації доступу до системи.
- Система повинна мати зрозумілий та інтуїтивно зрозумілий інтерфейс користувача, що дозволяє легко навігувати та виконувати необхідні функції конфігурування мережевих пристроїв.

Обмеження системи та додаткові можливості:

– забезпечення валідації введеної інформації та повідомлення про некоректність введених даних.

– обмежене виконання фільтрації та пошуку.

4.2 Вимоги до надійності:

- Застосунок повинен забезпечувати стабільне та надійне з'єднання з мережевими пристроями, зменшуючи можливість втрати зв'язку або збоїв під час конфігурування.
- Система повинна мати заходи безпеки, що запобігають випадковим змінам конфігурації мережевих пристроїв, забезпечуючи контроль доступу та авторизацію користувачів.
- Система повинна бути відмовостійкою і здатною відновлюватись після виникнення помилок або збоїв, забезпечуючи неперервну роботу та доступ до мережевих пристроїв.
- Система повинна мати можливість автоматично перевіряти доступність мережевих пристроїв.

4.3 Умови експлуатації:

Програмний продукт повинен використовуватись у приміщеннях які відповідають умовам роботи ЕОМ, а саме мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДНАОП 0.00-1.13-99 (див. табл. 1).

Таблиця 1. Кліматичні умови

Пора року	Категорія робіт згідно з ГОСТ 12.01-005-88	Температура повітря, град.С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		Оптимальна	Оптимальна	Оптимальна
Холодна	легка-1-а	22-24	40-60	0,1
	легка-1-б	21-23	40-60	0,1
Тепла	легка-1-а	23-25	40-60	0,1
	легка-1-б	22-24	40-60	0,2

Користувач, який працює з програмним продуктом, повинен мати навички роботи з персональним комп'ютером та бути ознайомленим з документацією до програми. Це допоможе забезпечити ефективне та безперебійне використання програмного продукту, оскільки користувач буде знати, як правильно взаємодіяти з програмою та виконувати необхідні дії.

3.4 Вимоги до складу та параметрів технічних засобів

Продукт, що розробляється повинен використовуватись на персональних комп’ютерах тощо, що мають наступні характеристики:

- роздільна здатність дисплею – HD (1280x720);
- оперативна пам’ять – 2 ГБ;
- операційна система – Linux, Debian.
- частота процесора – 1.6 ГГц;

4.5 Вимоги до інформаційної та програмної сумісності
Програмний продукт, що розробляється, має підтримку для наступних операційних систем:

- Ubuntu
- Debian

4.6 Вимоги до маркування і упаковки

Упаковка програмного продукту, включаючи документацію повинна бути захищена від пошкоджень різного роду (механічних, кліматичних).

На упаковці повинно бути вказана назва продукту, номер версії, мінімальні системні вимоги.

На зворотній стороні упаковки вказується розробник та його юридична адреса. На рисунку 3.1. представлено приклад маркування.

Програмний додаток «Застосунок для швидкого конфігурування мережевих пристроїв » Мінімальні системні вимоги: роздільна здатність дисплею – HD оперативна пам’ять – 2 ГБ; операційна система – Linux, Debian. частота процесора – 1.6 ГГц;	Розробник: Нестеров І. Є. Кафедра «КІТ», УДУНТ м. Дніпро, вул. Лазаряна 2 2023

Рисунок 3.1 Маркування

4.7 Вимоги до транспортування і зберігання

Транспортування програмного продукту повинно забезпечувати його безпеку, цілісність і захист від несанкціонованого доступу. Програмний виріб, який міститься на оптичному носії даних типу CD-R, потребує відповідної упаковки, щоб забезпечити захист від механічних пошкоджень та атмосферного впливу.

Оптимальним варіантом для захисту програмного продукту на CD-R може бути використання пластикового футляра або паперового конверту. Ці упаковки забезпечують фізичний захист від подряпин, ударів та інших механічних пошкоджень, які можуть виникнути під час транспортування.

Крім того, важливо враховувати атмосферний вплив на програмний продукт. Упаковка повинна бути стійкою до вологості, температурних змін та інших атмосферних факторів, які можуть негативно вплинути на якість CD-R та його вміст.

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Вся документація до програмного додатку повинна задовольняти вимоги до програмної документації.

До складу програмної документації має входити технічне завдання та робочий проект.

До складу робочого проекту мають входити:

– специфікація;

– текст програми;

– опис програми.

Вся документація до програми повинна задовольняти вимогам державного стандарту до оформлення програмних документів.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

В табл. 5.1 приведені стадії та етапи розробки програмного продукту.

Таблиця 5.1 – Стадії та етапи розробки

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, виріб та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.22 - 18.02.22
Робочий проект	Програмування та відлагодження програми.	19.02.22 - 20.05.22
	Тестування програми	20.05.22 - 27.05.22
	Розробка, узгодження і затвердження програмної документації.	27.05.22 - 12.06.22

7 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник розробки Клименко І.В.

Прийом здійснюється комісією у складі:

- Горячкін В. М. (керівник підрозділу);
- Клименко І.В.. (керівник розробки).

8 БІБЛІОГРАФІЧНИЙ СПИСОК

1. Івченко, Ю.М. Основи стандартизації програмних систем: методичні вказівки до дипломного проектування та лабораторних робіт/уклад.: Ю.М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. - 38 с

Додаток Б

Текст програми

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ComCtrls, StdCtrls,
  ExtCtrls, Buttons, Process;

type

  { TForm1 }

  TForm1 = class(TForm)
    BitBtn1: TBitBtn;
    BitBtn10: TBitBtn;
    BitBtn11: TBitBtn;
    BitBtn12: TBitBtn;
    BitBtn13: TBitBtn;
    BitBtn14: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    BitBtn5: TBitBtn;
    BitBtn6: TBitBtn;
    BitBtn7: TBitBtn;
```

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

ComboBox1: TComboBox;

ComboBox2: TComboBox;

ComboBox3: TComboBox;

ComboBox4: TComboBox;

ComboBox5: TComboBox;

Edit1: TEdit;

Edit10: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

GroupBox1: TGroupBox;

GroupBox10: TGroupBox;

GroupBox11: TGroupBox;

GroupBox12: TGroupBox;

GroupBox2: TGroupBox;

GroupBox3: TGroupBox;

GroupBox4: TGroupBox;

GroupBox5: TGroupBox;

GroupBox6: TGroupBox;

GroupBox7: TGroupBox;

GroupBox8: TGroupBox;

GroupBox9: TGroupBox;

Image1: TImage;

Image2: TImage;

Label1: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

LabeledEdit1: TLabeledEdit;

LabeledEdit2: TLabeledEdit;

LabeledEdit3: TLabeledEdit;

LabeledEdit4: TLabeledEdit;

LabeledEdit5: TLabeledEdit;

Memo1: TMemo;

Memo2: TMemo;

Memo3: TMemo;

PageControl1: TPageControl;

```
  RadioButton1: TRadioButton;
  RadioButton2: TRadioButton;
  RadioButton3: TRadioButton;
  RadioButton4: TRadioButton;
  RadioGroup1: TRadioGroup;
  RadioGroup2: TRadioGroup;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  TabSheet4: TTabSheet;
  Timer1: TTimer;
  procedure BitBtn10Click(Sender: TObject);
  procedure BitBtn13Click(Sender: TObject);
  procedure BitBtn14Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn5Click(Sender: TObject);
  procedure BitBtn6Click(Sender: TObject);
  procedure BitBtn7Click(Sender: TObject);
  procedure BitBtn8Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure RadioButton1Click(Sender: TObject);
  procedure RadioButton2Click(Sender: TObject);
  procedure RadioButton3Click(Sender: TObject);
  procedure RadioButton4Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
private
```

```
  public

  end;

var
  Form1: TForm1;

implementation
uses Unit2;       //додаємо unit2, саме тут, щоб не було кільця між unitами
{$R *.lfm}

{ TForm1 }

procedure TForm1.FormCreate(Sender: TObject);           //процедура створення форми в ОЗП
var
  TxtProc: TProcess;
begin
Form2:=TForm2.Create(nil);                              //ініціалізуємо форму 2
Form2.Show;
TxtProc:=TProcess.Create(nil);                          //запускаємо процес, що перевіряє адаптери в системі
TxtProc.Executable:='ls';                               //команда ls виводить список файлів в каталозі
TxtProc.Parameters.Add('/sys/class/net/');              //аргументом команди передаємо каталог, в якому зберігається інформація про адаптери
TxtProc.Options:=[poWaitOnExit, poUsePipes];            //опції потоку, чекати завершення виконання програми та використовувати "труби" (зв`язок між процесами)
```

```
TxtProc.Execute;                    //виконати процес
ComboBox1.Items.LoadFromStream(TxtProc.Output);    //перехоплення виводу з потоку у список ComboBox1
TxtProc.Free;                       //звільнення ОЗП
TxtProc:=TProcess.Create(nil);      //Той самий поток для кожного ComboBox
TxtProc.Executable:='ls';
TxtProc.Parameters.Add('/sys/class/net/');
TxtProc.Options:=[poWaitOnExit, poUsePipes];
TxtProc.Execute;
ComboBox2.Items.LoadFromStream(TxtProc.Output);
TxtProc.Free;
TxtProc:=TProcess.Create(nil);
TxtProc.Executable:='ls';
TxtProc.Parameters.Add('/sys/class/net/');
TxtProc.Options:=[poWaitOnExit, poUsePipes];
TxtProc.Execute;
ComboBox3.Items.LoadFromStream(TxtProc.Output);
TxtProc.Free;
TxtProc:=TProcess.Create(nil);
TxtProc.Executable:='ls';
TxtProc.Parameters.Add('/sys/class/net/');
TxtProc.Options:=[poWaitOnExit, poUsePipes];
TxtProc.Execute;
ComboBox4.Items.LoadFromStream(TxtProc.Output);
TxtProc.Free;
TxtProc:=TProcess.Create(nil);
TxtProc.Executable:='ls';
TxtProc.Parameters.Add('/sys/class/net/');
```

```
TxtProc.Options:=[poWaitOnExit, poUsePipes];
TxtProc.Execute;
ComboBox5.Items.LoadFromStream(TxtProc.Output);
TxtProc.Free;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
var
  FileModify: TStringList;                    //оголошення змінної типу StringList для збереження даних списку в ОЗП
begin
if not (ComboBox1.Text='Choose adapter') and not (ComboBox1.Text='lo') then begin    //перевірка чи вибраний один з адаптерів
  if (CheckBox2.Checked) then begin                    //що робить додаток, якщо вибрана демо-версія
  ComboBox1.Color:=clDefault;                    //зброс кольору ComboBox
  Memo1.Lines.Add('writing file /etc/network/interfaces');          //так як це емуляція (демо), просто виводимо текст у Memo
  Memo1.Lines.Add('auto lo');
  Memo1.Lines.Add('iface lo inet loopback');
  if (RadioButton1.Checked) then begin                    //перевірка вказаного типу з`єднання
  Memo1.Lines.Add('auto '+ComboBox1.Text);
  Memo1.Lines.Add('iface '+ComboBox1.Text+' inet dhcp');
  end else begin
  Memo1.Lines.Add('auto '+ComboBox1.Text);
  Memo1.Lines.Add('iface '+ComboBox1.Text+' inet static');
  Memo1.Lines.Add('address '+Edit1.Text);
  Memo1.Lines.Add('netmask '+Edit2.Text);
```

```
Memo1.Lines.Add('gateway '+Edit3.Text);
Memo1.Lines.Add('dns-nameservers '+Edit4.Text+','+Edit5.Text);
Memo1.Lines.Add('done!');
end;
end else begin                                  //що робить програма, якщо це робочий режим
ComboBox1.Color:=clDefault;
FileModify:=TStringList.Create;                 //створюємо в ОЗП список
with FileModify do begin                        //разом із цим списком
Add('auto lo');                                 //додаємо рядки локальної петлі
Add('iface lo inet loopback');
if (RadioButton1.Checked) then begin            //якщо вибраний DHCP
Add('auto '+ComboBox1.Text);                    //вписуємо протокол відповідно до вибраного адаптеру
Add('iface '+ComboBox1.Text+' inet dhcp');
SaveToFile('/etc/network/interfaces');          //зберігаємо у файл interfaces
Free;                                           //звільняємо список із ОЗП
end else begin
FileModify:=TStringList.Create;
with FileModify do begin
Add('auto '+ComboBox1.Text);                    //те саме для статичної IP
Add('iface '+ComboBox1.Text+' inet static');
Add('address '+Edit1.Text);
Add('netmask '+Edit2.Text);
Add('gateway '+Edit3.Text);
Add('dns-nameservers '+Edit4.Text+','+Edit5.Text);
```

```
  Memo1.Lines.Add('done!');
  SaveToFile('/etc/network/interfaces');
  Free;
  end;
  end;
  end;
end;
  end else ComboBox1.Color:=clRed;                    //ComboBox виділяється червоним, якщо не вибраний жоден адаптер
  end;

  procedure TForm1.BitBtn13Click(Sender: TObject);
  var
   WFCon: TProcess;
  begin
  if not (LabeledEdit1.Text='') then begin
  LabeledEdit1.Color:=clDefault;
  WFCon:=TProcess.Create(nil);
  WFCon.Executable:='nmcli';                          //запуск утиліти nmcli, що записує відомості про мережу Wi-FI (логін та пароль)
  WFCon.Parameters.Add('device');
  WFCon.Parameters.Add('wifi');
  WFCon.Parameters.Add('connect');
  WFCon.Parameters.Add(LabeledEdit1.Text);
  WFCon.Parameters.Add('password');
  WFCon.Parameters.Add(LabeledEdit2.Text);
  WFCon.Options:=[poWaitOnExit, poUsePipes];
  WFCon.Execute;
  WFCon.Free;
```

```
end else LabeledEdit1.Color:=clRed;
end;

procedure TForm1.BitBtn10Click(Sender: TObject);
var
  InProc, OutProc, MasqProc: TProcess;
begin
if not (ComboBox3.Text='') and not (ComboBox4.Text='') then begin     //для створення маршрутизації між адаптерами виконується команда
ComboBox3.Color:=clDefault;                                  //iptables із відповідними аргументами запуску
InProc:=TProcess.Create(nil);
InProc.Executable:='iptables';                               //обмін трафіку з адаптеру1 на адаптер2
InProc.Parameters.Add('-A');
InProc.Parameters.Add('FORWARD');
InProc.Parameters.Add('-i');
InProc.Parameters.Add(ComboBox3.Text);
InProc.Parameters.Add('-o');
InProc.Parameters.Add(ComboBox4.Text);
InProc.Parameters.Add('-j');
InProc.Parameters.Add('ACCEPT');
InProc.Parameters.Add(LabeledEdit5.Text);
InProc.Options:=[poWaitOnExit, poUsePipes];
InProc.Execute;
InProc.Free;
OutProc:=TProcess.Create(nil);
OutProc.Executable:='iptables';                              //зворотній рух з адаптеру2 на адаптер1
OutProc.Parameters.Add('-A');
```

```
    OutProc.Parameters.Add('FORWARD');
    OutProc.Parameters.Add('-i');
    OutProc.Parameters.Add(ComboBox4.Text);
    OutProc.Parameters.Add('-o');
    OutProc.Parameters.Add(ComboBox3.Text);
    OutProc.Parameters.Add('-j');
    OutProc.Parameters.Add('ACCEPT');
    OutProc.Parameters.Add(LabeledEdit5.Text);
    OutProc.Options:=[poWaitOnExit, poUsePipes];
    OutProc.Execute;
    OutProc.Free;
    MasqProc:=TProcess.Create(nil);
    MasqProc.Executable:='iptables';                    //переадресація
пакетів
    MasqProc.Parameters.Add('-t');
    MasqProc.Parameters.Add('nat');
    MasqProc.Parameters.Add('-A');
    MasqProc.Parameters.Add('POSTROUTING');
    MasqProc.Parameters.Add('-o');
    MasqProc.Parameters.Add(ComboBox4.Text);
    MasqProc.Parameters.Add('-j');
    MasqProc.Parameters.Add('MASQUERADE');
    MasqProc.Options:=[poWaitOnExit, poUsePipes];
    MasqProc.Execute;
    MasqProc.Free;
    Memo3.Lines.Add('FORWARDING MUST BE ENABLED IN KERNEL (IF
NOT):');     //попередження про необхідність ввімкнути маршрутизацію на
рівні ядра системи
    Memo3.Lines.Add('echo 1 > /proc/sys/net/ipv4/ip_forward');
    Memo3.Lines.Add('Rules addeded');
```

```
    end else ComboBox3.Color:=clRed;
    end;

    procedure TForm1.BitBtn14Click(Sender: TObject);
    var
      SPDuTest: TProcess;
    begin
    Form1.Cursor:=crHourGlass;
    SPDuTest:=TProcess.Create(nil);
    SPDuTest.Executable:='speedtest';                    //завантаження
процесу speedtest
    SPDuTest.Options:=[poWaitOnExit, poUsePipes];
    SPDuTest.Execute;
    Memo3.Lines.LoadFromStream(SPDuTest.Output);                  //та
перехоплення його виводу в Memo
    SPDuTest.Free;
    Form1.Cursor:=crDefault;
    end;

    procedure TForm1.BitBtn2Click(Sender: TObject);
    begin
      Application.Terminate;                          //вихід з програми
    end;

    procedure TForm1.BitBtn3Click(Sender: TObject);
    var
      WFScan: TProcess;
    begin
    WFScan:=TProcess.Create(nil);                          //команда
сканування мереж Wi-Fi
```

```
WFScan.Executable:='nmcli';
WFScan.Parameters.Add('device');
WFScan.Parameters.Add('wifi');
WFScan.Parameters.Add('list');
WFScan.Options:=[poWaitOnExit, poUsePipes];
WFScan.Execute;
Memo2.Lines.LoadFromStream(WFScan.Output);
WFScan.Free;
end;


procedure TForm1.BitBtn4Click(Sender: TObject);
var
  FileModify: TStringList;
begin
if not (ComboBox2.Text='Choose adapter') and not (ComboBox2.Text='lo') then begin
  ComboBox2.Color:=clDefault;
  if (CheckBox2.Checked) then begin
  Memo2.Lines.Add('writing file /etc/network/interfaces');          //така сама команда запису interfaces, тільки для Wi-Fi
  Memo2.Lines.Add('auto lo');
  Memo2.Lines.Add('iface lo inet loopback');
  if (RadioButton3.Checked) then begin
  Memo2.Lines.Add('auto '+ComboBox2.Text);
  Memo2.Lines.Add('iface '+ComboBox2.Text+' inet dhcp');
  end else begin
  Memo2.Lines.Add('auto '+ComboBox2.Text);
  Memo2.Lines.Add('iface '+ComboBox2.Text+' inet static');
  Memo2.Lines.Add('address '+Edit6.Text);
```

```
Memo2.Lines.Add('netmask '+Edit7.Text);
Memo2.Lines.Add('gateway '+Edit8.Text);
Memo2.Lines.Add('dns-nameservers '+Edit9.Text+','+Edit10.Text);
Memo2.Lines.Add('done!');
end;
end else begin
FileModify:=TStringList.Create;
with FileModify do begin
Add('auto lo');
Add('iface lo inet loopback');
if (RadioButton3.Checked) then begin
Add('auto '+ComboBox2.Text);
Add('iface '+ComboBox2.Text+' inet dhcp');
SaveToFile('/etc/network/interfaces');
Free;
end else begin
FileModify:=TStringList.Create;
with FileModify do begin
Add('auto '+ComboBox2.Text);
Add('iface '+ComboBox2.Text+' inet static');
Add('address '+Edit6.Text);
Add('netmask '+Edit7.Text);
Add('gateway '+Edit8.Text);
Add('dns-nameservers '+Edit9.Text+','+Edit10.Text);
Memo1.Lines.Add('done!');
SaveToFile('/etc/network/interfaces');
Free;
end;
end;
```

```
  end;
end;
end else ComboBox2.Color:=clRed;
end;

procedure TForm1.BitBtn5Click(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TForm1.BitBtn6Click(Sender: TObject);
var
  Shpr, SPDTest: TProcess;
begin
if not (LabeledEdit3.Text='') then begin
LabeledEdit3.Color:=clDefault;
Shpr:=TProcess.Create(nil);
Shpr.Executable:='tc';                          //команда запуску шейпера (обмеження трафіку)
Shpr.Parameters.Add('qdisc');
Shpr.Parameters.Add('add');
Shpr.Parameters.Add('dev');
Shpr.Parameters.Add(ComboBox5.Text);
Shpr.Parameters.Add('root');
Shpr.Parameters.Add('tbf');
Shpr.Parameters.Add('rate');
Shpr.Parameters.Add(LabeledEdit3.Text+'kbit');
Shpr.Parameters.Add('burst');
Shpr.Parameters.Add(LabeledEdit5.Text);
```

```
Shpr.Options:=[poWaitOnExit, poUsePipes];
Shpr.Execute;
Shpr.Free;
SPDTest:=TProcess.Create(nil);
SPDTest.Executable:='speedtest';
SPDTest.Options:=[poWaitOnExit, poUsePipes];
SPDTest.Execute;
Memo3.Lines.LoadFromStream(SPDTest.Output);
SPDTest.Free;
end else LabeledEdit3.Color:=clRed;
end;

procedure TForm1.BitBtn7Click(Sender: TObject);
var
  ShprRes: TProcess;
begin
ShprRes:=TProcess.Create(nil);
ShprRes.Executable:='tc';                    //зброс налаштувань
обмеження
ShprRes.Parameters.Add('qdisc');
ShprRes.Parameters.Add('del');
ShprRes.Parameters.Add('dev');
ShprRes.Parameters.Add(ComboBox3.Text);
ShprRes.Parameters.Add('root');
ShprRes.Options:=[poWaitOnExit, poUsePipes];
ShprRes.Execute;
ShprRes.Free;
end;
```

```
procedure TForm1.BitBtn8Click(Sender: TObject);
var
  PortLock: TProcess;
begin
if not (LabeledEdit4.Text='') then begin
LabeledEdit4.Color:=clDefault;                    //блокування портів за допомогою
PortLock:=TProcess.Create(nil);                   //UbuntuFireWall
PortLock.Executable:='ufw';
PortLock.Parameters.Add('deny');
PortLock.Parameters.Add(LabeledEdit4.Text);
PortLock.Options:=[poWaitOnExit, poUsePipes];
PortLock.Execute;
Memo3.Lines.LoadFromStream(PortLock.Output);
PortLock.Free;
end else LabeledEdit4.Color:=clRed;
end;

procedure TForm1.RadioButton1Click(Sender: TObject);
begin
  GroupBox2.Enabled:=false;                       //перемикачі між dhcp та static,
end;                                              //зміна доступності полів адрес

procedure TForm1.RadioButton2Click(Sender: TObject);
begin
  GroupBox2.Enabled:=true;;
end;
```

```
procedure TForm1.RadioButton3Click(Sender: TObject);
begin
  GroupBox5.Enabled:=false;
end;

procedure TForm1.RadioButton4Click(Sender: TObject);
begin
  GroupBox5.Enabled:=true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
Form1.AlphaBlendValue:=Form1.AlphaBlendValue+1;
//візуальний ефект прозорості форми
if Form1.AlphaBlendValue>=254 then begin
Form2.Free;
Form2:=nil;
Timer1.Enabled:=false;
end;
end;

end.
unit Unit2;

{$mode objfpc}{$H+}

interface

uses
```

```
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ComCtrls, StdCtrls,
  ExtCtrls, Process, Unit1;

type

  { TForm2 }

  TForm2 = class(TForm)
    Label1: TLabel;
    ProgressBar1: TProgressBar;
    procedure FormCreate(Sender: TObject);
  private

  public

  end;

var
  Form2: TForm2;

implementation

{$R *.lfm}

{ TForm2 }

procedure TForm2.FormCreate(Sender: TObject);
var
  DepUProc, DepGProc, DepIProc: TProcess;
```

```
begin
  Label1.Caption:='Starting';
  ProgressBar1.Position:=10;
  Label1.Caption:='Checking depencies';
  ProgressBar1.Position:=15;
  DepUProc:=TProcess.Create(nil);
  ProgressBar1.Position:=20;
  DepUProc.Executable:='apt';
  ProgressBar1.Position:=25;
  DepUProc.Parameters.Add('update');
  ProgressBar1.Position:=30;
  DepUProc.Parameters.Add('-y');
  DepUProc.Options:=[poWaitOnExit, poUsePipes];
  DepUProc.Execute;
  DepUProc.Free;
  ProgressBar1.Position:=35;
  DepGProc:=TProcess.Create(nil);
  DepGProc.Executable:='apt';
  ProgressBar1.Position:=40;
  DepGProc.Parameters.Add('upgrade');                    //оновлення
репозиторіїв
  ProgressBar1.Position:=45;
  DepGProc.Parameters.Add('-y');
  DepGProc.Options:=[poWaitOnExit, poUsePipes];
  DepGProc.Execute;
  DepGProc.Free;
  DepIProc:=TProcess.Create(nil);
  DepIProc.Executable:='apt';
  ProgressBar1.Position:=50;
```

```
    DepIProc.Parameters.Add('install');
    ProgressBar1.Position:=55;
    DepIProc.Parameters.Add('wireless-tools');          //встановлення пакетів,
що використані в додатку
    DepIProc.Parameters.Add('net-tools');
    DepIProc.Parameters.Add('iproute2');
    DepIProc.Parameters.Add('ufw');
    DepIProc.Parameters.Add('iptables');
    DepIProc.Parameters.Add('speedtest-cli');
    DepIProc.Parameters.Add('-y');
    DepIProc.Options:=[poWaitOnExit, poUsePipes];
    ProgressBar1.Position:=90;
    DepIProc.Execute;
    DepIProc.Free;
    ProgressBar1.Position:=100;
    Label1.Caption:='Running program';
    Form1.Timer1.Enabled:=true;
  end;

  end.
```