

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ____ ” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація системи забезпечення
обміну інформацією у корпоративній мережі на базі технології
DMVPN”**

Виконав здобувач вищої освіти
II курсу, групи КІ-20М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»

Панасюк Б.В.

« ____ » _____ 2021 р.

Керівник проекту

кандидат технічних наук, доцент

Анна КОВАЛЕНКО

« ____ » _____ 2021 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ
« 6 » вересня 2021 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Панасюку Богдану Володимировичу

(прізвище, ім’я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN

2. Керівник роботи Коваленко Анна Степанівна, канд. техн. наук, доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

<u>1. Призначення та область використання.</u>	<u>7. Економічна ефективність розробленої програми.</u>
<u>2. Перегляд аналогічних існуючих систем.</u>	<u>8. Заходи з охорони праці та техніки безпеки</u>
<u>3. Опис і обґрунтування проектних рішень.</u>	<u>9. Висновки.</u>
<u>4. Етапи програмування системи.</u>	
<u>5. Впровадження системи в промислову експлуатацію</u>	
<u>6. Наукова новизна</u>	
<u>6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)</u>	
<u>Наукова новизна</u>	<u>1 аркуш</u>
<u>Структурна схема системи</u>	<u>1 аркуш</u>
<u>Функціональна схема системи</u>	<u>1 аркуш</u>
<u>Діаграма процесів</u>	<u>1 аркуш</u>
<u>Блок-схема алгоритму роботи додатку</u>	<u>2 аркуша</u>
<u>Показники економічної ефективності</u>	<u>1 аркуш</u>

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Панасюк Б.В. Дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Метою розробки є дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Об'єктом дослідження є процес забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Предметом дослідження є методи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Методи дослідження базуються на методах теорії інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi 10.3.2 Rio.

Ключові слова: комп'ютерна інженерія, корпоративна мережа, DMVPN

ABSTRACT

Panasiuk B.V. Research and software implementation of information exchange system in the corporate network based on DMVPN technology. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualifying work for the second (master's) level of higher education, software has been developed that is designed for the system of information exchange in the corporate network based on DMVPN technology.

The purpose of development is research and software implementation of information exchange system in the corporate network based on DMVPN technology.

The object of research is the process of ensuring the exchange of information in the corporate network based on DMVPN technology.

The subject of the research is the methods of information exchange in the corporate network based on DMVPN technology.

Research methods are based on methods of information theory, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the system of information exchange in the corporate network based on DMVPN technology.

In the process of working on the software model, the analysis of existing hardware and software was performed. All components of the developed software are fully described.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment Delphi 10.3.2 Rio.

Keywords: computer engineering, corporate network, DMVPN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	26
2.3 Розгорнута постановка завдання	31
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	33
3.1 Опис функціонування системи.....	33
3.2 Розробка структурної схеми	40
3.3 Розробка функціональної схеми.....	58
3.4 Розробка діаграми процесів	65
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	67
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	67
4.2 Захист розробленого програмного забезпечення	81
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	85
6 НАУКОВА НОВИЗНА	93

ВКРМ-123.21.0014.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>		Панасюк Б.В.			Дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN	Лім.	Аркуш	Аркушів
<i>Перев.</i>		Коваленко А.С.				М	1	134
<i>Н.контр.</i>		Гермак В.С.			ЦНТУ КІ-20М-1,4			
<i>Затв.</i>		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	94
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	94
7.2 Розрахунок трудомісткості розробки програмної продукції	96
7.3 Визначення чисельності виконавців і планового фонду зарплати	98
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	103
7.5 Визначення собівартості розробки та ціни програмної продукції.	107
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	110
7.7 Визначення експлуатаційних витрат.....	110
7.8 Визначення економічної ефективності програмної продукції.....	112
7.9 Висновок.	114
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	115
8.1 Пожежна безпека для приміщень з ПЕОМ.....	115
8.2 Профілактичні заходи щодо запобігання пожеж.....	117
8.3 Інструкція з пожежної і вибухової техніки безпеки.....	119
8.4 Розрахункова частина	120
8.5 Висновки до розділу.....	123
9 ОСНОВНІ ВИСНОВКИ.....	124
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	126

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AH	–	автентифікуючий заголовок
CA	–	сертифікаційне співтовариство
DES	–	Data Encrption Standard
DoS	–	атака "Відмова в обслуговуванні"
DOI	–	область інтерпретації
ESP	–	Інкапсуляція зашифрованих даних
HTTPS	–	зашифрований http
IAB	–	координаційна рада мережі Internet
IDS	–	система, яка автоматизує процес перегляду подій
IETF	–	проблемна група проектування Internet
IKE	–	протокол обміну ключами за замовчуванням для ISAKMP
IKMP	–	протоколу керування ключами прикладного рівня
IPsec	–	комплект протоколів захисту інформації по IP
ISAKMP	–	механізми узгодження атрибутів використовуваних протоколів
ISP	–	постачальник послуг Internet
MAC	–	коди на перевірку цілісності
MD5	–	дайджест повідомлення
Oakley	–	сесійні ключі на комп'ютери мережі Інтернет
PFS	–	ідеальна пряма безпека
PRF	–	псевдовипадкова функція
SA	–	Security Association
SKIP	–	команда підготовки наступної команди
SPI	–	індекс параметрів безпеки
SPD	–	база даних політики безпеки
SSL	–	протокол захищених сокетів
TCP	–	транспортний протокол
VPN	–	віртуальні приватні мережі

ВСТУП

Актуальність теми. DMVPN (Dynamic Multipoint Virtual Private Network) – динамічна багатоточкова віртуальна приватна мережа. Ця технологія є подальшим розвитком VPN компанією Cisco Systems. Якщо коротко, то вона дозволяє створити віртуальну приватну мережу з можливістю динамічного створення тунелів між вузлами.

Припустимо, що є якесь підприємство із центральним офісом і декількома філіями, територіально рознесеними по декількох площадках (одна з них в іншому місті або навіть державі), які потрібно об'єднати в одну мережу.

Підприємство не дуже багате й не може дозволити собі використовувати для зв'язку між офісами орендовані канали. Зате на кожній площадці є вихід в Інтернет. Це може бути мережа одного конкретного провайдера або різних, неважливо.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.
- Дослідження системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.
- Програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Об'єктом дослідження є процес забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Предметом дослідження є методи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Методи дослідження базуються на методах теорії інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

– Розроблено вітчизняний продукт забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

DMVPN (Dynamic Multipoint Virtual Private Network – динамічна багатоточкова віртуальна приватна мережа, відмінно підходить для рішення цього завдання. У чому ж перевага використання цієї технології, скажемо, перед класичним VPN? Ця технологія є подальшим розвитком VPN компанією Cisco Systems – технологія об'єднання приватних мереж через публічні мережі.

Для об'єднання приватних мереж, як правило, використовується побудова тунелів через суспільні мережі. У звичайному варіанті необхідно набудувувати тунель для кожного з'єднання, тобто якщо ви хочете одержати повнозв'язну мережу з N вузлів потрібно налаштувати $N(N-1)/2$ тунелів. У певний момент таких зв'язків стає занадто багато й доводиться використовувати топологію типу «зірка». У цьому випадку весь трафік проходить через центральний вузол, що не додає йому продуктивності. Уже не говорячи про те, що у випадку його відмови вся мережа розсипається.

От у такій ситуації й приходить на допомогу DMVPN. Суть технології полягає в тому, що на кожному вузлі мережі настроюється тільки один тунельний інтерфейс, що і організує потім необхідну кількість тунелів автоматично, перетворюючи топологію мережі з «зірки» у повнозв'язну. Незважаючи на те, що в даній технології все-таки передбачене виділення центрального вузла, кінцеві вузли також організують між собою тунелі. В остаточному підсумку, після сходження такої мережі, навіть побудованої за технологією «зірка», центральний вузол уже не несе такої важливої ролі, і у випадку його відмови кінцеві вузли можуть продовжувати між собою взаємодіяти по вже встановлених тунелях (якщо, звичайно ж, мережа провайдеру дозволяє).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Технологія DMVPN дозволяє вирішити завдання об'єднання локальних мереж територіально віддалених офісів в одну корпоративну мережу більше масштабованим методом, чим створення зв'язків точка-точка між всіма офісами:

– При додаванні нових маршрутизаторів філій в існуючу мережу DMVPN, необхідно налаштувати тільки новий маршрутизатор, змін на вже існуючих маршрутизаторах не потрібно.

– DMVPN дозволяє використовувати динамічно призначені IP-адреси на віддалених маршрутизаторах (за центральним маршрутизатором закріплена статична публічна IP-адреса) .

– Якщо двом віддаленим маршрутизаторам необхідно встановити тунель прямо, то він устанавлюється динамічно.

1.2 Область застосування

Областю застосування є корпоративна мережа. Корпоративна мережа – це мережа, головним призначенням якої є підтримка роботи конкретного підприємства, що володіє даною мережею. Користувачами корпоративної мережі є тільки співробітники даного підприємства.

На відміну від мереж операторів зв'язку, корпоративні мережі, у загальному випадку, не роблять послуг стороннім організаціям або користувачам.

Хоча формально корпоративною мережею є мережа підприємства будь-якого масштабу, звичайно так називають мережа великого підприємства, що має відділення в різних містах і, можливо, різних країнах. Тому корпоративна мережа є складеною мережею, що включає як локальні, так і глобальні мережі.

Мережі відділів

Мережі відділів – це мережі, які використовуються порівняно невеликою групою співробітників, що працюють в одному відділі підприємства. Ці співробітники вирішують деякі загальні завдання, наприклад ведуть бухгалтерський облік або займаються маркетингом. Уважається, що відділ може

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

нараховувати до 100-150 співробітників. Мережа відділу – це локальна мережа, що охоплює всі приміщення, що належать відділу. Це можуть бути кілька кімнат або поверх будинку.

Головним призначенням мережі відділу є поділ локальних ресурсів, таких як застосунки, дані, лазерні принтери й модеми. Звичайно мережі відділів не ділять на підмережі й у їхній состав входять один або два файлових сервери й не більше тридцяти користувачів. У цих мережах локалізується більша частина трафіку підприємства. Мережі відділів звичайно створюються на основі якої-небудь однієї мережної технології – Ethernet (або кілька технологій із сімейства Ethernet – Ethernet, Fast Ethernet, рідше Gigabit Ethernet), Token Ring або FDDI. Для такої мережі характерний один або максимум два типи операційних систем.

Завдання мережного адміністрування на рівні відділу відносно прості: додавання нових користувачів, усунення простих відмов, установка нових вузлів і нових версій програмного забезпечення. Такою мережею може управляти співробітник, що присвячує обов'язкам адміністратора тільки частину свого часу. Найчастіше адміністратор мережі відділу не має спеціальної підготовки, але є тією людиною у відділі, що краще всіх розбирається в комп'ютерах, і саме собою виходить так, що він займається адмініструванням мережі.

Існує й інший тип мереж, близький до мереж відділів, – мережі робочих груп. До таких мереж відносять зовсім невеликі мережі, які включають до 10 -20 комп'ютерів. Характеристики мереж робочих груп практично не відрізняються від описаних характеристик мереж відділів. Такі властивості, як простота мережі й однорідність, тут проявляються найбільшою мірою, у те час як мережі відділів можуть наближатися в деяких випадках до наступного по масштабі типу мереж – мережам будинків і кампусів.

У мережах робочих груп ще часто використовуються технології локальних мереж на поділюваних середовищах. У міру просування по ієрархії нагору – до мереж відділів, будинків і кампусів, поділювані середовища зустрічаються усе рідше й рідше, поступаючись місцем мережам, що комутируються.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Мережа відділу може входити до складу мережі будинку (кампусу) або ж являти собою мережу віддаленого офісу підприємства. У першому випадку мережа відділу підключається до мережі будинку або кампусу за допомогою технології локальної мережі, яка сьогодні, швидке всього, буде одна із представниць сімейства Ethernet. У другому випадку мережа віддаленого офісу підключається безпосередньо до магістралі мережі за допомогою якої-небудь технології WAN, наприклад Frame Relay.

Мережі будинків і кампусів

Мережа будинку поєднує мережі різних відділів одного підприємства в межах окремого будинку, а мережа кампусу – однієї території (кампусу), що покриває площа в кілька квадратних кілометрів. Для побудови мереж будинків (кампусів) використовуються технології локальних мереж, можливостей яких досить, щоб забезпечити необхідну зону покриття.

Звичайно мережа будинку (кампусу) будується по ієрархічному принципі із власною магістраллю, побудованої на базі технології Gigabit Ethernet, до якої приєднуються мережі відділів, що використовують технологію Fast Ethernet або Ethernet. Магістраль Gigabit Ethernet практично комутується завжди, хоча ця технологія й має варіант на поділюваному середовищі.

Послуги такої мережі включають взаємодія між мережами відділів, доступ до загальних баз дані підприємства, доступ до загального факс-серверам, високошвидкісним модемам і високошвидкісним принтерам. У результаті співробітники кожного відділу підприємства одержують доступ до деяких файлів і ресурсів мереж інших відділів. Важливою послугою, надаваної мережами кампусів, є доступ до корпоративних баз даних незалежно від того, на яких типах комп'ютерів ці бази розташовуються.

Саме на рівні мережі кампусу виникають проблеми інтеграції неоднорідного апаратного й програмного забезпечення. Типи комп'ютерів, мережних операційних систем, мережного апаратного забезпечення можуть відрізнитися в кожному відділі. Звідси випливають складності керування

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

мережами кампусів. А оскільки мережі відділів, що входять у мережі кампусу, досить незалежні й часто побудовані на базі різних технологій, що поєднує технологією звичайно є IP.

Мережі масштабу підприємства

Мережі масштабу підприємства, або корпоративні мережі, відрізняються тим, що в них на перший план виходять інформаційні послуги. Якщо мережі операторів зв'язку можуть і не надавати інформаційних послуг, так як комп'ютери користувачів перебувають за межами зони їхньої відповідальності, то корпоративні мережі не можуть собі цього дозволити.

Настільні комп'ютери користувачів і сервери є невід'ємною частиною будь-якої корпоративної мережі, тому й розроблювачі, і фахівці з обслуговування корпоративних мереж повинні це враховувати. Можна сказати, що корпоративна мережа являє собою приклад інфокомунікаційної мережі, де дотримується паритет між двома типами послуг. Корпоративну мережу можна представити, у вигляді «острівців» локальних мереж, «плаваючих» у телекомунікаційному середовищі.

Іншою особливістю корпоративної мережі є її масштабність. Мережу рівня відділу або будинку рідко називають корпоративною, хоча формально це так. Звичайна назва «корпоративна» застосовують тільки для мережі, що включає велику кількість мереж масштабу відділу й будинку, розташованих у різних містах і об'єднаних глобальними зв'язками.

Число користувачів і комп'ютерів у корпоративній мережі може вимірятися тисячами, а число серверів – сотнями; відстані між мережами окремих територій можуть виявитися такими, що використання глобальних зв'язків стає необхідним. Для з'єднання віддалених локальних мереж і окремих комп'ютерів у корпоративній мережі застосовуються різноманітні телекомунікаційні засоби, у тому числі канали первинних мереж, радіоканали, супутниковий зв'язок.

Неодмінним атрибутом настільки складної й великомасштабної мережі є високий ступінь неоднорідності (гетерогенності) – не можна задовольнити

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

потреби тисяч користувачів за допомогою однотипних програмних і апаратних засобів. У корпоративній мережі обов'язково задіють різні типи комп'ютерів – від мейнфреймів до персональних комп'ютерів, кілька типів операційних систем і безліч різних застосунків. Неоднорідні частини корпоративної мережі повинні працювати як єдине ціле, надаючи користувачам за можливістю зручний і простий доступ до всіх необхідних ресурсів.

Поява корпоративних мереж – це гарна ілюстрація відомого філософського постулату про перехід кількості в якість. При об'єднанні в єдину мережу окремих мереж великого підприємства, що має філії в різних містах і навіть країнах, багато кількісних характеристик об'єднаної мережі перевершують деякий критичний поріг, за яким починається нова якість. У цих умовах існуючі методи й підходи до рішення традиційних завдань мереж менших масштабів для корпоративних мереж й виявилися непридатними. На перший план вийшли такі завдання й проблеми, які в мережах робочих груп, відділів і навіть кампусів або мали другорядне значення, або взагалі не проявлялися. Прикладом може служити найпростіша (для невеликих мереж) завдання – ведення облікових даних про користувачів мережі.

Найбільш простий спосіб її рішення – приміщення облікових даних всіх користувачів у локальну базу облікових даних кожного комп'ютера, до ресурсів якого ці користувачі повинні мати доступ. При спробі доступу дані витягають із локальної облікової бази й на їхній основі доступ надається або не надається. Для невеликої мережі, що складає з 5-10 комп'ютерів, цей підхід працює дуже добре. Але якщо в мережі налічується кілька тисяч користувачів, кожному з яких потрібний доступ до декількох десятків серверів, мабуть, що це рішення стає вкрай неефективним. Адміністратор повинен повторити кілька десятків разів (по числу серверів) операцію занесення облікових даних кожного користувача. Сам користувач також змушений повторювати процедуру логічного входу щораз, коли йому потрібний доступ до ресурсів нового сервера. Гарне рішення цієї проблеми для великої мережі – використання централізованої довідкової системи,

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

у базі даної яке зберігаються облікові записи всіх користувачів мережі. Адміністратор один раз виконує операцію занесення даних користувача в базу, а користувач один раз виконує процедуру логічного входу, причому не в окремий сервер, а в мережу цілком.

При переході від більше простого типу мереж до більше складного – від мереж відділу до корпоративної мережі – географічні відстані збільшуються, підтримка зв'язку комп'ютерів стає усе більше складним і дорогим. У міру збільшення масштабів мережі підвищуються вимоги до її надійності, продуктивності й функціональним можливостям. По мережі циркулюють всі зростаючі обсяги даних, і мережа повинна забезпечувати їхню безпека й захищеність поряд з доступністю. Все це приводить до того, що корпоративні мережі будуються на основі найбільш потужного й різноманітного встаткування й програмного забезпечення.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Знайти VPN-сервіс, що захистить вас від всіх онлайн-погроз і забезпечить необхідний рівень приватності, здається практично неможливим завданням. У цей час на ринку представлені сотні VPN-сервісів, і кожний з них заявляє про себе як про кращий.

Виявлено, що ряд невеликих VPN-сервісів, які не мають у своєму розпорядженні рекламні бюджети великих брендів, працюють краще, якісніше й надійніше багатьох більше відомих VPN.

Після строгих і навіть причепливих тестів з'ясувалося також, що деякі популярні VPN-сервіси насправді марні, а інші й зовсім небезпечні.

Провівши всі ці перевірки, у даній роботі можемо із чистою совістю представити вам рейтинг 10 кращих VPN-сервісів, які захистять вас і ваші дані, а також нададуть блискавично швидкі підключення.

NordVPN

VPN-сервіс №1 2020 року

- Виділені сервери для завантаження торрентів: так, тут є сервери, настроєні спеціально для роботи з P2P-трафіком.
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, HBO GO, Amazon Prime Video, Showtime, Sling TV.
- Є застосунки для: Windows, macOS, Linux, Android, iOS, BlackBerry, роутерів, Raspberry Pi, Android TV, Chrome, Firefox.
- Відношення до логів: строга безлогова політика.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Безпечний перегляд потокових трансляцій (стрімів) і роздача торрентів – це завдання, які в Nord VPN вирішуються в 1 клік. Завдяки DNS -налаштуванням спеціального режиму роботи SmartPlay, застосунок NordVPN автоматично надає надійний доступ і відмінну швидкість, коли ви відкриваєте Netflix і інші стрімінгові сервіси.

Режим CyberSec – унікальна функція NordVPN, що забезпечує захист від шкідливого ПЗ й що дозволяє блокувати рекламу. Просунуті налаштування обфускації дозволять вам установити VPN-підключення навіть із території країни, що активно обмежує вільний доступ в Інтернет (наприклад, у Китаї). Також вам не прийдеться турбуватися про VPN-витоки, так як NordVPN надає дві незалежні функції екстреного відключення від Мережі (Kill switch) і дає можливість задати власні налаштування DNS для профілактики витоків даних.

Якщо вам коли-небудь знадобиться допомога, ви можете сміло покластися на службу підтримки користувачів, доступну в онлайн-чаті 24/7. Більше того, в NordVPN є просто величезна база знань, звідки ви зможете почерпнути чимало корисного про сервери й установку застосунків.

Цей сервіс не підійде вам, якщо:

– Ви збираєтеся оплатити його послуги через PayPal.

Що можна сказати в підсумку? У даній роботі вважаємо, що NordVPN – це кращий VPN-сервіс для всіх і кожного, однак відсутність можливості оплати через PayPal все-таки розчаровує. Ви можете перевірити на ділі його зручні інтерфейси й видатні функції на 6 пристроях відразу, скориставшись 30-денною гарантією повернення засобів.

До слова – в NordVPN зараз іде просто відмінна акція! Скористайтесь 70% знижкою при оформленні передплати на NordVPN за допомогою цих промо-кодів.

ExpressVPN

VPN-сервіс №2 у загальному й беззастережному лідері в плані швидкості.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Виділені сервери для завантаження торрентів: немає (всі сервери підтримують P2P-трафік).
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, DAZN, HBO GO, Amazon Prime Video, Showtime, Sling TV.
- Є застосунки для: Windows, macOS, Linux, Android, iOS, BlackBerry, роутери, Raspberry Pi, Chrome, Firefox, Safari, PlayStation, Xbox, Apple TV, Nintendo Switch, Amazon Fire TV, Samsung Smart TV, Roku, Chromecast, Nvidia Shield.
- Відношення до логів: строга безлогова політика.
- У даній роботі тестували десятки VPN з різних локацій на предмет швидкості підключення, і ExpressVPN був беззастережним лідером.

В ExpressVPN є 3 000+ високошвидкісних серверів в 160 країнах. Це найшвидший VPN у нашій рейтингу, а тому він ідеально підходить для торрентів і стрімів.

Функція роздільного тунелювання дозволить вам налаштувати роботу ExpressVPN під себе й вибрати, трафік яких застосунків і сервісів необхідно шифрувати. Це дуже корисно, якщо ви хочете мати доступ до локального й іноземному контенту на тому самому пристрої.

Більше того: ExpressVPN допоможе вам одержати доступ до улюбленого контенту навіть на тих платформах, які не підтримують VPN -підключення. Функція MediaStreamer дозволить вам забути про обмеження на доступ до контенту на телевізорах Smart TV, Amazon Fire TV, Apple TV і навіть ігрових приставках.

ExpressVPN підтримує P2P-трафік на всіх своїх серверах, так що вам не прийде нічого набудовувати для використання торрентів. Функція Smart Location, доступна в додатках, автоматично вибере для вас самий підходящий сервер.

ExpressVPN пропонує 256-бітне шифрування й підтримку OpenVPN на всіх платформах, що забезпечує бездоганний захист. Також ви будете захищені від витоків DNS за допомогою функції Network Lock – екстреного відключення

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

підключення. TrustedServer – це інноваційна технологія ExpressVPN, що забезпечує кожному серверу в мережі сервісу однаковий рівень захисту й своєчасне відновлення ПЗ. Ця новинка дозволяє істотно знизити ймовірність зіткнення з різними кіберзагрозами. Технологія TrustedServer також дозволяє серверам ExpressVPN працювати, використовуючи тільки оперативну пам'ять. Так як серверам нічого не доводиться записувати на жорсткі диски, те після перезавантаження на пристроях не залишиться взагалі ніяких даних – у принципі. Це дозволяє домогтися безпрецедентного рівня приватності й захисту.

ExpressVPN стабільно обходить блокування Великого китайського файрволу, так що це кращий варіант для роботи з китайським сегментом Мережі.

З онлайн-чатом, готовим прийти на допомогу 24/7 і докладними інструкціями з установки для безлічі підтримуваних платформ, служба підтримки користувачів ExpressVPN також показує гідний рівень.

Цей сервіс не підійде вам, якщо:

– Ви шукаєте дешевий VPN. Послуги ExpressVPN коштують майже у два рази більше, ніж послуги деяких інших VPN-сервісів у цьому рейтингу.

ExpressVPN – це відмінне сполучення швидкості, бездоганної надійності в роботі й конфіденційності, що виразно коштує своїх грошей. Ви зможете перевірити ExpressVPN на ділі без якого-небудь ризику, скориставшись 30-денною гарантією повернення грошей. Але вам навряд чи захочеться повертати гроші: цей VPN може з легкістю задовольнити запити навіть самих вимогливих користувачів.

CyberGhost

Кращий VPN для початківців.

- Виділені сервери для завантаження торрентів: є.
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, DAZN, HBO GO, Amazon Prime Video, Sky Go.
- Є застосунки для: Windows, macOS, Linux, Android, iOS, роутери, Raspberry Pi, Chrome.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Відношення до логів: строга безлогова політика.
- Якщо ви шукаєте сервіс із інтуїтивно зрозумілим інтерфейсом і зручним дизайном, то CyberGhost може стати для вас ідеальним VPN.

Цей румунський сервіс дасть вам все необхідне: просунуте шифрування, повна відсутність логів і доступ до значної серверної мережі (5 600 серверів в 89 країнах). З застосунками, захищеними від витоків даних, і додатковою функцією екстреного відключення kill-switch цей VPN можна сміло назвати стабільним і безпечним.

Команда CyberGhost затверджує, що встановити й почати працювати з їхнім сервісом можна в один клік. Крім автоматичного підключення до найшвидшого сервера, у додатку можна знайти сервери, настроєні спеціально для стрімінгу й завантаження торрентів, і за допомогою яких ви зможете одержати доступ до американського каталогу контенту Netflix, Hulu і іншим популярним сервісам. Мабуть, це й справді простий VPN для щоденного використання.

Крім того, працювати з базовими й просунутими налаштуваннями застосунки дуже просто. Ви зможете задати власні правила роботи й налаштувати опції підключення.

Хоча він і уступає у швидкості підключення NordVPN і ExpressVPN, CyberGhost все-таки забезпечує досить швидкі підключення для того, щоб без проблем дивитися потокові трансляції й завантажувати торренти. Якщо під час роботи із сервісом ви зштовхнетеся з якими-небудь технічними проблемами, за допомогою можна звернутися в онлайн-чат.

Цей сервіс не підійде вам, якщо:

- Вам потрібний VPN, що працює в Китаї. На даний момент CyberGhost не може обійти Великий китайський файрвол.

У даній роботі вважаємо, що CyberGhost – це простий VPN, що відмінно справляється зі своїми завданнями. Як наслідок, це ще один відмінний VPN-варіант, по праву що займає 3-є місце в нашому рейтингу.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Surfshark

Кращий VPN для великої родини й малого бізнесу.

- Виділені сервери для завантаження торрентів: немає (P2P-трафік підтримується всіма серверами).
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, HBO GO, Amazon Prime Video, ESPN.
- Є застосунки для: Windows, macOS, Linux, Android, iOS, роутери, Chrome, Firefox, Amazon Fire TV.
- Відношення до логів: строга безлогова політика.

Surfshark – це відносно новий VPN-сервіс, що вже встиг стати популярним.

Незважаючи на те, що Surfshark – новачок на ринку VPN, він пропонує дуже рідкий бонус – можливість підключити необмежену кількість пристроїв. Якщо ви хочете захистити гаджети всієї вашої родини (або ж ви самі використовуєте безліч пристроїв з можливістю виходу в Мережу), то варіанта краще вам просто не знайти.

Звичайно, у даній роботі б не стали розглядати Surfshark всерйоз, не надавай цей сервіс послуги дійсно високого рівня. Тут ви знайдете роздільне тунелювання, подвійне шифрування, екстрене відключення від Мережі й спеціальний режим роботи CleanWeb, що блокує рекламу й шкідливе ПЗ. Додаткові налаштування приховання трафіку доступні в режимі NoBorders, а режим Camouflage не дозволяє відслідковувати, що ви використовуєте VPN, а не пряме підключення.

Зростаюча серверна мережа Surfshark містить у собі понад 1041 серверів більш ніж в 61 країнах. Це значить, що у вас завжди буде швидкий і надійний доступ до потрібного контенту.

Цей сервіс не підійде вам, якщо:

- Вам потрібно встановити VPN на роутер – в Surfshark немає окремого застосунки для роутерів.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

У загальному говорячи, Surfshark – це вражаючий VPN-сервіс із преміум-функціями й простими, але, у той же час, стильними застосунками. Крім того, ви можете знайти по-справжньому вигідні пропозиції, скориставшись відмінними знижками на довгострокові тарифні плани й 30-денною гарантією повернення грошей.

PrivateVPN

Кращий бюджетний VPN.

- Виділені сервери для завантаження торрентів: ні, але P2P-трафік підтримується серверами в Британії.
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, HBO GO, Amazon Prime Video.
- Є застосунки для: Windows, macOS, Linux, Android, iOS, роутери.
- Відношення до логів: строга безлогова політика.

PrivateVPN – це доступний сервіс із відмінною репутацією. Він надає доступ до компактної мережі з 150+ серверів, що належать самої компанії й установлених в 59 країнах і більш ніж 80 різних локаціях.

Цей шведський сервіс прикладає всі необхідні зусилля для захисту конфіденційності користувальницьких даних. Незважаючи на те, що місце реєстрації сервісу не саме вдале (Швеція входить у розвідальнянс «14 очей»), PrivateVPN у принципі не зможе передати комусь ваші дані, так як принципово не веде логи. Його прості й зручні застосунки дають доступ до шифрування AES-256 і підтримують різні протоколи, у тому числі визнаний галузевий стандарт – OpenVPN.

Крім екстреного відключення, PrivateVPN пропонує захист від витоків даних IPv6 і DNS. Також у додатку можна активувати спеціальний «режим невидимості», розроблений для обходу блокувань підключення через VPN і глибокої перевірки пакетів даних, за рахунок чого ви одержите необмежений доступ до Всесвітньої павутини.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Звичайно, цей сервіс не може похвастати тисячами серверів і дуже просунутими функціями, однак це не заважає PrivateVPN залишатися відмінним варіантом для всіх, хто просто хоче обійти блокування доступу до свого улюбленого контенту. Він підтримує P2P-трафік на своїх британських серверах і навіть пропонує спеціальні сервери для доступу до Netflix і іншим популярним стрімінговим сервісам. Вражаючі швидкості підключень від PrivateVPN – гарантія того, що ви не зштовхнетеся з лагами й буферизацією: просто виберіть потрібний режим і приступайте до перегляду!

Цей сервіс не підійде вам, якщо:

- Вам потрібний VPN з більшою кількістю серверів.

Якщо ви шукаєте бюджетний сервіс із доступними цінами як на короткострокові, так і на довгострокові тарифи, PrivateVPN може стати для вас відмінним варіантом. 30-денна гарантія повернення грошей забезпечить вам досить часу, щоб випробувати цей VPN на ділі.

IPVanish

Відмінні застосунки й стабільні швидкості.

- Виділені сервери для завантаження торрентів: немає.

- Відкриває доступ до: Netflix, BBC iPlayer.

- Є застосунки для: Windows, macOS, Linux, Android, iOS, Windows Phone, роутери, Chromebook, Amazon Fire TV.

- Відношення до логів: строга безлогова політика.

Баланс між безпекою й швидкістю доступу робить сервіс IPVanish цілком гідним VPN, що заслуговує вашої уваги. Більше 1 200 серверів, відмінний потенціал обходу блокувань і надійний захист трафіку – цей сервіс може багато чого вам запропонувати.

256-бітне шифрування й безліч VPN-протоколів, у тому числі OpenVPN, роблять IPVanish безпечним, а функція екстреного відключення й захист від витоків DNS/IPv6 гарантує вам і вашому трафіку постійний захист.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Швидкість підключень тут не тільки висока, але й стабільна, так що вам не прийде нервувати через дратівні обриви підключення або плаваючої швидкості. Як приємний бонус, IPVanish не дуже давно вдосконалив свої сервери для доступу до Netflix, оптимізувавши й поліпшивши їх.

IPVanish підтримує P2P-трафік і SOCKS 5-проксі, що дозволяє швидко завантажувати торренти. Так, сервіс зареєстрований у США (і, відповідно, повинен підкорятися вимогам DMCA), але це не заважає йому дотримуватися строгої безлогової політики, так що навіть американська «прописка» не повинна відлякувати аматорів торрентів.

IPVanish також надає доступ до додаткових налаштувань обфускації й IP-адрес(наприклад, ви можете з певною періодичністю одержувати нова IP-адресу). Все це, а також доступна 24/7 служба підтримки й 10 одночасних підключень у рамках однієї підписки дозволяють сервісу виділитися із числа інших VPN. Хотілося б, звичайно, щоб термін дії гарантії повернення грошей був побільше, але навіть 7 днів – це набагато краще, ніж нічого.

Цей сервіс не підійде вам, якщо:

- Ви ставите вартість VPN у главу кута.

IPVanish – не найдешевший сервіс, однак його ціна повністю виправдана відмінним функціоналом.

Hotspot Shield

Надійний захист і вільний доступ до потокових трансляціям.

- Виділені сервери для завантаження торрентів: немає.
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer, Amazon Prime Video.
- Є застосунки для: Windows, macOS, Android, iOS, Chrome.
- Ведення логів: так (протягом VPN-сесій ведуться логи підключень, також збираються певні дані в маркетингових цілях).

Платний VPN-сервіс від Hotspot Shield, він же Hotspot Shield Premium, надасть вам доступ до більш ніж 2 500 серверам по усьому світі, а також до стильних застосунків і швидких підключень. Але чи коштує воно того?

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Мабуть, що так! Цей VPN стане гідним вибором для звичайних користувачів, адже тут використовується відмінне шифрування (128/ 256-бітне) і власна розробка – протокол Catapult Hydra (заснований на OpenSSL).

Hotspot Shield стискає потоковий трафік, що поліпшує доступ до Netflix і дозволяє забути про буферизацію.

Що ж стосується функцій, то тут є екстрене відключення від Мережі й...нічого більше. Немає ні можливості задіяти користувальницькі DNS, ні додаткових налаштувань для торрентів. І нехай навіть P2P-трафік тут підтримується, вам краще вибрати інший VPN-сервіс із цього списку, якщо торренти для вас – основний пріоритет.

Незважаючи на відмінну якість роботи в цілому, Hotspot Shield не можна включити в число безлогових VPN -сервісів. Додайте сюди реєстрацію в США, щоб зробити однозначно сумний вивід: ви будете легкою мішенню для копірайтних тролів і запитів DMCA. Для жителів США це актуально подвійно.

Цей сервіс не підійде вам, якщо:

– Ви шукаєте VPN для торрентів.

Нехай навіть в Hotspot Shield є свої недоліки, він був, є й залишається популярним, надійним і доступним VPN-сервісом. У даній роботі радимо перевірити його на практиці, скориставшись 45-денною гарантією повернення грошей.

UltraVPN

№1 у плані користувальницької підтримки.

– Спеціальний режим для роботи з торрентами: немає.

– Відкриває доступ до: Netflix, Hulu, HBO GO.

– Підтримується на: Windows, macOS, Android, iOS, Fire TV.

– Ведення логів: так (сервіс збирає IP-адреси, дані про місце розташування й відомості про пристрій, однак не веде логи активності).

UltraVPN – це бюджетний сервіс із відмінною користувальницькою підтримкою. Скажемо так – ви зможете зв'язатися з нею навіть по телефоні!

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Сервіс управляє мережею з 100+ серверів в 55+ локаціях по усьому світі. Проводячи тести, у даній роботі з'ясували, що цей сервіс відкриває доступ до Netflix, Hulu і HBO GO. Його сервери забезпечують швидкі підключення, так що з буферизацією у даній роботі практично не зіштовхувалися.

UltraVPN забезпечує надійний захист користувальницьких даних, застосовуючи протокол OpenVPN і 256-бітне шифрування. На жаль, у нього та ж проблема, що й в Hotspot Shield – ім'я їй «логі». Знову ж, це американська компанія, тобто вона зареєстрована на території розвідальянса «14 очей». Результат? Цей VPN не варто розглядати як основний варіант, якщо ви хочете домогтися повної анонімності.

Перевагою UltraVPN можна назвати просту установку й зручність роботи як з настільними, так і з мобільними застосунками.

Не використовуйте цей сервіс, якщо:

– Ви шукаєте VPN, здатний відкрити доступ до BBC iPlayer за межами УК.

Що ж можна сказати в підсумку? Якщо вам потрібний швидкий VPN-сервіс для перегляду стрімів, обов'язково зверніть увагу на UltraVPN. Ви зможете протестувати його, нічим не ризикуючи, скориставшись 30-денною гарантією повернення грошей від сервісу.

VyprVPN

Вражаючий і недешевий VPN-сервіс для настільних пристроїв.

- Виділені сервери для завантаження торрентів: немає.
- Відкриває доступ до: Netflix, Hulu, BBC iPlayer.
- Є застосунки для: Windows, macOS, Android, iOS, роутери, Android TV, QNAP, Blackphone, Anonabox.
- Відношення до логів: строга безлогова політика.

VyprVPN пропонує своїм користувачам два різних тарифи, але якщо у вас є пристрій під керуванням ОС Windows або Mac, те вам потрібний тариф Premium.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Крім доступу до 700+ серверів і строгої безлогової політики, VuprVPN Premium дозволить вам скористатися своєю добре відомою функцією – протоколом Chameleon власної розробки.

Будучи поліпшеною версією OpenVPN, Chameleon надає додатковий захист і допомагає ще ефективніше обходити блокування доступу по географічній ознаці. Як наслідок, VuprVPN – це відмінний варіант для тих, хто хоче обійти геообмеження Netflix і інших популярних стрімінгових сервісів.

На жаль, протокол Chameleon доступний тільки в додатках для настільних пристроїв. Втім, базовий застосунок VuprVPN також забезпечує достатні приватність і якість роботи й для мобільних пристроїв. Застосунки захищені від витоків даних, мають стильний інтерфейс і, що важливіше всього, зручні й прості.

VuprVPN складно назвати бюджетним сервісом, однак відмінними знижками при оформленні передплати на довгостроковий тариф доступні гарні знижки. VuprVPN надає 30-денну гарантію повернення грошей, так що ви зможете перевірити сервіс на ділі, нічим не ризикуючи.

Цей сервіс не підійде вам, якщо:

- Ви хочете оплатити VPN-послуги біткоїнами або іншою криптовалютою.
- Ви шукаєте VPN із цілодобово доступною техпідтримкою.

VuprVPN довів, що це надійний сервіс, що цінує користувальницьку конфіденційність.

ZenMate

Простий VPN-сервіс, що справляється зі своїми завданнями.

- Виділені сервери для завантаження торрентів: немає (але P2P-трафік дозволений).
- Відкриває доступ до: Netflix.
- Є застосунки для: Windows, macOS, Android, iOS, Chrome, Firefox, Opera.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

– Відношення до логів: строга безлогова політика.

І наш рейтинг замикає сервіс, що сміло можна назвати простим, якщо не базовим. ZenMate мало чим виділяється на тлі інших сервісів у цьому списку, однак у даній роботі поставили його на почесне 10-е місце не без причини.

За досить помірну ціну ZenMate захистить вас за допомогою протоколу OpenVPN і строгої безлогової політики, а також відкриє доступ до більш ніж 290 серверів в 30 країнах. P2P-трафік підтримується, нехай і без окремого режиму роботи для завантаження торрентів, а швидкість підключень досить висока щоб дивитися стріми у форматі HD без яких-небудь проблем.

Є тут і функція екстреного відключення від Мережі, так що ваші конфіденційні дані не залишаться без захисту у випадку обриву підключення. Треба сказати, що в минулому користувачі ZenMate часто страждали від обривів зв'язку, але після ряду поліпшень і відновлень ZenMate став як ніколи стабільним.

Якщо для вас важливо мати доступ до техпідтримки через онлайн-чат, то ZenMate VPN вас, швидше за все, розчарує – у сервісу є тільки система тікетів. Втім, компетентні співробітники техпідтримки відповідають швидко й докладно.

Нам подобається, що VPN-сервіси стають усе краще й краще. ZenMate – саме один з них: після істотного поліпшення якості роботи цей VPN-сервіс став досить гарний, щоб одержати місце в числі кращих провайдерів 2020 року.

Цей сервіс не підійде вам, якщо:

– Ви шукаєте VPN для стрімів. Zenmate надає доступ тільки до Netflix.

Перевірити ZenMate на ділі можна, скориставшись безкоштовним 7-денним пробним періодом і 30-денною гарантією повернення грошей.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero RAD Studio Delphi 10.3.2 Rio Architect – це найшвидший спосіб створювати й оновлювати інтенсивно працюючі з даними, сильно

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

взаємодіючі застосунки з візуально насиченим користувацьким інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою Object Pascal і C++. Широкий вибір функцій підтримки Windows 10, у тому числі нові компоненти VCL для Windows 10, стилі для VCL і FMX, а також служби UWP (універсальної платформи Windows), наприклад повідомлення, дозволяють легко й швидко перенести застосунки в Windows 10, зберігши користувачів. Нова платформа дозволяє підтримувати великі проекти на більшому числі платформ із подвоєним обсягом пам'яті в середовищі розробки й удвічі більшим розміром підтримуваних проектів. Крім того, підтримка декількох моніторів і десятки нових функцій середовища розробки, призначених для прискорення створення коду, зроблять роботу як ніколи ефективною. За допомогою RAD Studio 10 розроблювачі зможуть створювати застосунки в 5 разів швидше в порівнянні з іншими інструментами, а розробка застосунків для декількох настільних, мобільних, хмарних платформ і платформ баз даних, включаючи 32 і 64 -розрядні версії Windows 10, Mac OS X, iOS і Android, стане ще швидше.

Зміни у версії 10.3 Rio:

– Створюйте міжплатформні застосунки. 80% інтернет користувачів мають смартфони й застосунки доступу, а також дані з мобільного пристрою й ноутбука / настільного комп'ютера, саме тому так важливо в цей час, щоб застосунки працювали в будь-якому пристрої.

– У всіх версіях Professional, Enterprise і Architect RAD Studio 10.3 надається підтримка процесу розробки застосунків для мобільних пристроїв. Розроблювачі RAD Studio кодують лише один раз, компілюють споконвічно для кожної платформи, що скорочує час і трудозатрати на вивчення декількох мов і дозволяє паралельно управляти циклами розробки.

– Підтримка Android API26, відповідність вимогам Google Play Store відносно нових застосунків із серпня 2018 року й відновлення застосунків з листопада 2018 року.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

– Чудові застосунки Windows з VCL. Бібліотека візуальних компонентів (Visual Component Library, VCL) пропонує просту й візуальну розробку користувацького інтерфейсу застосунки, у версії 10.3 представлені нові відновлення, які дозволять вашим додаткам виглядати сучасними й свіжими.

– Розширена підтримка HighDPI. Завдяки новому елементу керування VCL High DPI ImageList у версії 10.3 розроблювачі, що створюють нові застосунки VCL для Windows або оновлюючи існуючі застосунки для High DPI дисплеїв, можуть повністю підтримувати зроблені до рівня пікселів зображення зі змінною розв'язною здатністю на всіх елементах керування, а також будь-яке користувацьке креслення, що вимагає масштабованих зображень для моніторів з різною розв'язною здатністю.

– Підтримка Per Monitor V2. Переконаєтеся, що ваш застосунок масштабується правильно для всіх типів масштабування в Windows, реагуючи на зміни масштабування DPI на різних екранах під час виконання.

– Розширена підтримка Windows 10 і WinRT API. Сюди відноситься ряд ключових API-інтерфейсів WinRT і останні API-інтерфейси Windows 10, включаючи готові до використання компоненти для убудованих у застосунки покупок і випробувань у магазині Windows 10 Store.

– Розгортання застосунків на основі служб за допомогою RAD Server. Продуктивність RAD Server була значно поліпшена завдяки десятикратному збільшенню потужності відносно простих операцій.

– Нові компоненти обробки JSON допоміжного засобу.

– Розширена підтримка RAD Server для клієнта Ext JS. Об'єднайте зовнішній інтерфейс javascript і веб-службу, підтримувану REST Server REST. (У версії Architect тепер включена ліцензія ExtJS Professional).

– Версії Enterprise включають ліцензію для одиничного розгортання RAD Server.

– Версії Architect включають ліцензію для розподіленого розгортання RAD Server.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Що нового в C++? Підтримка C++17 Win32 збільшує продуктивність, поліпшує роботу компілятора й прискорює процес кодування. Були оновлені RTL і STL.
- Нова версія STL/Dinkumware 2018 для Win32 і Win64.
- Поліпшене автодоповнення коду Автодоповнення коду для даного компілятора тепер асинхронне, швидше й із кращими результатами, чим у попередньому автодоповненні коду C++. Уведення тексту не буде припинятися, поки виконується розрахунок.
- Тепер є підтримка налагодження для оптимізації компонувань.
- 2X швидкість математичної продуктивності для Win64.
- Нові додаткові лабораторії C++ в GetIt.
- Нові й поліпшені можливості роботи з базами даних. InterBase 2017 / IBToGo 2017 в RAD Studio. Версії Professional включають ліцензію розроблювача InterBase 2017, у той час як версії Enterprise і Architect містять у собі ліцензії InterBase ToGo. InterBase ToGo доповнена можливістю шифрування, функціями зміни подань, призначених для простої синхронізації даних застосунку по підписці без обмежень за розміром файлу бази даних.
- Поліпшена й оновлена підтримка для популярних баз даних, включаючи MySQL v8.0, MariaDB 10.3, SQL Server 2017, PostgreSQL v10, Firebird v3.0, MongoDB, InterBase, SQLite 3.23.1, SQL Anywhere і багатьох інших.
- Удосконалення DataSnap.
- Поліпшення REST. Підтримка додаткових родинних REST методів, типів і властивостей.
- Повністю оновлений модуль живлення версії Architect. Одержіть більше від версії Architect, включаючи ці ліцензії сімейства Idera.
- Ліцензія Sencha ExtJS Professional: Створіть свій ідеальний мережний вхідний інтерфейс за допомогою javascript і ExtJS.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Ліцензія на розгортання InterBase ToGo. Додайте сховище даних у свої застосунки за допомогою цієї гнучкої, зашифрованої бази даних, що вбудовується.

– Ліцензія для розподіленого розгортання RAD Server. Ідеально підходить для серверного застосунку архітектури мікросервісів.

– Ліцензія AquaData Studio. Вражаючий аналіз бази даних.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

У цей час існують сотні VPN -сервісів, і всі заявляють про себе як про лідерів. У таких умовах стає складно знайти по-справжньому кращий VPN. Наші рекомендації опираються на найважливіші критерії, що дозволить вам найбільш зваженим образом вибрати кращий VPN для роботи в Україні.

Безпека

Двома найважливішими аспектами будь-якого гарного VPN-сервісу є конфіденційність і безпека. Вам необхідно знайти сервіс, що пропонує все це не тільки на папері, але й у реальності.

VPN-сервіс із підтверженою безлоговою політикою зможе надійно сховати всі ваші цифрові сліди від сторонніх. Безлогова політика означає, що сервіс не збирає, не зберігає й не передає дані про те, що ви робили в Мережі. Такий сервіс захистить ваші дані від сторонніх і забезпечить вам можливість працювати в Інтернеті конфіденційно. Як наслідок, наявність безлогової політики – це один з головних критеріїв, які лягли в основу VPN-рейтингів.

Швидкість

Але не можна назвати кращим VPN-сервіс без гідної швидкості підключення. Використання технології VPN означає, що ваш трафік буде перенаправлятися через додаткові сервери, що за замовчуванням сповільнює обмін даними. Будь-який VPN буде в тому або іншому ступені знижувати швидкість підключення; так що правильно буде порушити питання «наскільки?». Найшвидші VPN-сервіси впливають на якість підключення мінімальним образом. З ними ви зможете переглядати сайти, завантажувати торренти й дивитися потокові трансляції без лагів і буферизації.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

По-справжньому швидкий VPN навіть допоможе вам обійти обмеження швидкості підключення, установлені інтернет-провайдером. Цілком можливо, що так ваша швидкість виявиться навіть вище, ніж була!

Потоковий контент

До слова про стріми: VPN-сервіси – це популярне рішення для тих, хто хоче мати вільний доступ до всьому контенту таких платформ, як Netflix, Hulu, BBC iPlayer і DAZN. Ці сервіси обмежують доступність контенту залежно від регіону, звідки приходить користувач. На щастя, підключення до підходящого VPN-серверу дозволить вам одержати доступ до улюблених фільмів і серіалів.

Торренти

VPN-сервіси припадуться й для аматорів торрентів, особливо в країнах, де такий формат обміну даними заборонений. Навіть якщо ви дотримуєте всі правила (а саме це у даній роботі й рекомендуємо), завантаження торрентів без додаткових заходів захисту можуть поставити вас під удар з боку хакерів і жадібних копірайтних тролів.

P2P-підключення не захищені самі по собі, але виправити цей недолік можна за допомогою гарного VPN для торрентів. Сховавши свій вихідна IP-адресу й сховавши трафік за шаром шифрування, ви відгородите себе від всіх погроз, пов'язаних із завантаженням торрентів, зберігаючи їхньої переваги. Становлячи цей VPN-рейтинг, у даній роботі проаналізували всі важливі особливості сервісів у плані торрентів: P2P-режими роботи, SOCKS 5-проксі й легке налаштування торрент-клієнтів.

Цензура

Нарешті, давайте поговоримо про цензуру. Ви зрозумієте, що знайшли надійний сервіс, коли обраний вами VPN буде працювати в країні, незважаючи на активні спроби її влади заблокувати такого роду ресурси. Проводячи наше дослідження, у даній роботі звертали особливу увагу на спеціальні функції й налаштування, що забезпечують додаткову обфускацію (приховання) трафіку. Кращі VPN-сервіси в цьому рейтингу довели, що можуть працювати навіть у

						ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			34

таких складних країнах, як Китай, Іран й т.і. Є й інші фактори, що визначають місця в нашому рейтингу: розмір серверної мережі, шифрування, функції безпеки, простота роботи й, звичайно ж, ціна. У даній роботі не поспішали, уважно вивчаючи й перевіряючи всі аспекти роботи VPN-сервісів для складання наших оглядів.

Є думка, що конфіденційність у Мережі – це базове право людини й платити за це не можна. У теорії у даній роботі дотримуємося тих же поглядів, однак не можна забувати про те, як обстають справи на практиці.

VPN-сервіси пропонують своїм послуги в обмін на щось. І якщо мова йде не про гроші, варто насторожитися! Навіть сир у мишоловці не буває безкоштовним: якщо ви не платите за товар, те самі стаєте товаром.

На жаль, але більшість безкоштовних VPN-сервісів надають свої послуги з величезними обмеженнями, а іноді й прямо загрожують безпеці й конфіденційності користувальницьких даних. Низька швидкість, обмежена пропускна здатність каналу, що дратує реклама й навіть продаж користувальницьких даних – це далеко не повний список того, із чим можна зіштовхнутися, вибравши безкоштовний VPN.

Відмінним прикладом служить Hola. Цей безкоштовний VPN-сервіс по суті являє собою проксі-сервіс, заснований на протоколі P2P і перенаправляючий трафік одних користувачів через пристрої інших. Таким чином, сервіс не тільки використовує інтернет-канали своїх користувачів, але й збирає й записує практично всієї їхньої дії. В «політику конфіденційності» прямо зазначено, що Hola буде надавати ці дані третім сторонам, партнерам і рекламодавцям.

Звичайно, не всі безкоштовні VPN погані. У даній роботі ретельно протестували десятки таких сервісів і знайшли декілька безкоштовних і безпечних VPN. Але чи буде їх досить?

Вивчивши все, що можуть запропонувати вам платні VPN, ви навряд чи захочете користуватися безкоштовними сервісами. Все просто – якщо ви шукаєте спосіб надійно захистити свою конфіденційність і одержати стабільний і

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

швидкий доступ до необхідних ресурсів і серверів, краще заплатити пари сотень рублів, чим знову й знову розчаровуватися в роботі безкоштовних VPN.

Помнете, що оформити передплату на VPN-сервіс можна на куди більше вигідних умовах, якщо скористатися знижками й купонами. Ви зможете одержати послуги сервісу преміум-класу за відносно невеликі гроші. Вигода отут незаперечна!

Кращі VPN-сервіси, упевнені як своя пропозиція, пропонують потенційним клієнтам скористатися безкоштовними пробними періодами на свої послуги. Також більшість сервісів дають гарантію повернення грошей, так що ви можете детально протестувати сервіс і не боятися викинути гроші на вітер.

VPN-сервіси часто використовують у незаконних цілях, тому багато хто вважають, що й самі ці сервіси незаконні. Це далеко від істини.

Використання VPN зовсім законно в 95,9% країн. Поза законом (або в так званій «сірій зоні») VPN-сервіси в наступних країнах:

- Китай.
- Ірак.
- Іран.
- Росія.
- Білорусія.
- Туреччина.
- Оман.
- ОАЕ.

Робота з VPN – це зовсім повсякденна справа. Багато компаній використовують VPN, щоб захистити свої дані, що представляють комерційну цінність, від хакерів, а також для проведення маркетингових досліджень у різних країнах миру.

Ховатися за VPN заради здійснення злочинів – от що незаконно. Але погодитися, це не проблема VPN, особливо якщо врахувати, що мільйони користувачів вибирають ці сервіси заради захисту.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Спокійно використовувати VPN можна в більшості країн. Втім, дотримувати правил і не вплутуватися в сумнівні авантюри – це вже ваш обов'язок.

Крім того, ви можете зіткнутися з певним ризиком у наступних ситуаціях:

– Ви працюєте з VPN, що не має урядової ліцензії, на території країни, де дозволені тільки мають цю ліцензію сервіси;

– Ви зачуєте торрент-файли з добутками, захищеними авторськими правами, через VPN-сервіс, ведучий логи або не вміє негайно обривати з'єднання.

У наші дні для роботи з VPN досить скачати застосунок із сайту сервісу й установити його, що займе від сили пари мінут.

Багато які VPN-сервіси можна назвати дуже зручними у використанні: у них інтуїтивно зрозумілі інтерфейси й прості інструкції. Вам потрібно буде тільки вибрати безпечний сервер зі списку або дозволити VPN-Додатку автоматично підключати вас до найбільш підходящої локації. Для цього в ньому повинна підтримуватися функція швидкого підключення.

Якщо ви ніколи не працювали з VPN, не переживайте! Всі набагато простіше, ніж здається, особливо якщо ви прочитаєте наше докладне керівництво для починаючих користувачів.

На щастя, VPN-сервіси преміум класу коштують зовсім не космічних грошей. Якщо купувати місячну підписку, ціни вигляд переконливо. А от якщо заплатити відразу за рік-два, то в перерахуванні вийде близько 3-7\$ на місяць.

По факту, ви можете оформити передплату на високоякісний VPN-сервіс менше ніж за 3 долари на місяць!

Також варто відзначити, що в багатьох VPN-сервісів є безкоштовні пробні періоди, що допоможе вам перевірити їх на практиці й зрозуміти, чи відповідають їхні послуги вашим вимогам у плані швидкості, безпеки й так далі.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Багато сервісів також дають гарантію повернення грошей: у вас буде від 7 до 45 днів (залежить від самого сервісу), щоб встигнути скасувати підписку й одержати свої гроші назад, якщо щось вам не сподобається.

Також ви зможете заощадити на вартості підписки, скориставшись нашими акціями й знижковими купонами.

Tor, проксі й VPN – це три зовсім різні типи сервісів. Втім, часом вони використовуються заради однієї й тієї ж мети. Давайте уважніше розглянемо цей момент.

Tor (скорочення The Onion Router) – це анонімна мережа, де ваше підключення перенаправляється через ланцюжок серверів, підтримуваних або надаваних волонтерами. Як наслідок, вас і ваші дані стає набагато складніше відстежити. При цьому швидкість з'єднання падає. Це досить складна тема, так що почитайте наш повний посібник з мережі Tor, якщо ви хочете краще в ній розібратися.

Що й говорити, на тлі VPN в Tor є чимало недоліків. Так, це безкоштовна мережа, однак шифрування тут використовується менш просунуте, а підключення виходять більше повільними. Найкраще буде використовувати Tor і VPN разом – так ви зможете скористатися перевагами обох технологій. Стабільне VPN-підключення зведе до 0 уразливості Tor, а Tor зробить ваше перебування в Мережі ще більш конфіденційним.

От тільки не будь -який VPN можна використовувати з Tor. Щоб фокус удався, вам потрібний VPN типу NordVPN, у якого є спеціальні сервери для підключення Tor-через-VPN.

Проксі-сервіси мають єдину перевагу перед VPN: вони забезпечують за замовчуванням високу швидкість підключення й приховують користувальницьку IP-адресу. От тільки ніяких захисних функцій у них ні, трафік вони не шифрують, а в багатьох і якогось окремого інтерфейсу немає.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Все це значить, що проксі-сервіси й не безпечні, і не зручні. Втім, якщо конфіденційність підключення для вас не пріоритетна, то можна скористатися й ними.

Є сотні й тисячі проксі-сервісів, тому важливо не помилитися з вибором. Але краще буде оформити недорогу передплату на швидкий VPN і насолоджуватися відмінною швидкістю підключень, бездоганним захистом і повною конфіденційністю.

Примітка. В vpnMentor є власний безкоштовний проксі! Скористайтеся ним, якщо поспішаєте й не можете дозволити собі вибирати.

Той самий VPN може надавати своїм клієнтам застосунки для самих різних платформ. Втім, отут не всі так однозначно! Наприклад, навіть якщо ваш VPN бездоганно працює на Windows, його мобільний застосунок може працювати нестабільно й ненадійно.

По правді сказати, немає такого VPN-сервісу, що був би однозначно кращим для всіх пристроїв, версій ПЗ й ОС. Щоб зробити правильний вибір, вам необхідно провести невелике дослідження, вибираючи краще рішення для ваших потреб із числа перевірених VPN.

Нижче ви знайдете наші регулярно оновлювані рейтинги кращих VPN для всіх основних платформ:

- Windows.
- Mac.
- iOS, iPhone, iPad.
- Android.
- Chrome.
- Firefox.
- Safari.
- Apple TV.
- Smart TVs.
- Kodi.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- Fire TV Stick.
- PS4.
- Роутери.
- Roku.

3.2 Розробка структурної схеми

Покажемо приклад налаштування DMVPN – Dynamic Multipoint VPN, що є VPN рішенням компанії Cisco. Дане рішення використовується, коли потрібна висока масштабованість і легкість налаштування при підключенні філій до головного офісу.

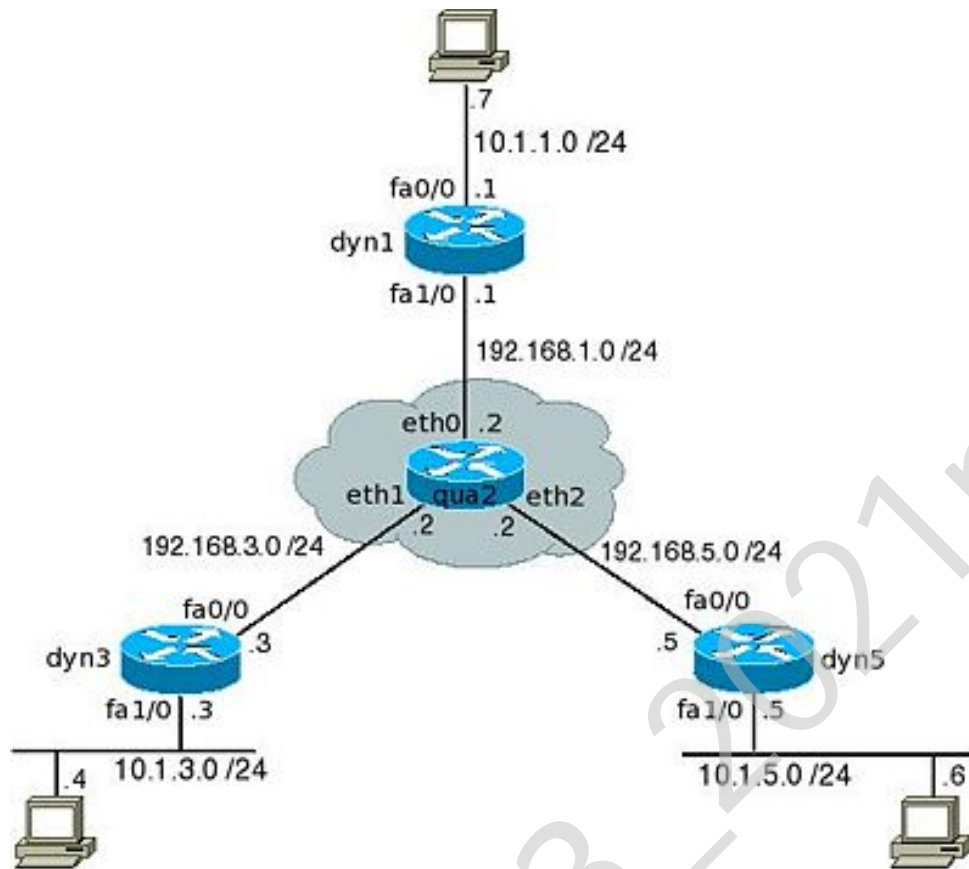
DMPVN одне із самих масштабованих і ефективних рішень VPN підтримуваних компанією Cisco. В основному воно використовується при топології Hub-and-Spoke, де ви хотіли б бачити прямі VPN тунелі Spoke -to-Spoke на застосунок до звичайних Spoke-to-Hub тунелям. Це означає, що філії зможуть спілкуватися з один одним прямо, без необхідності проходження трафіку через HQ. Як уже згадували, ця технологія є пропрієтарною технологією Cisco.

Якщо вам необхідно підключити більше десяти сайтів до головного офісу, то DMPVN буде ідеальним вибором. Крім того, DMPVN підтримує не тільки Hub-and-Spoke, але й Full-Mesh топологію, так як всі сайти мають між собою зв'язність без необхідності налаштування статичних VPN тунелів між сайтами.

Для початку перелічимо важливі характеристики даного способу організації Site-to-Site VPN для кращого розуміння:

- Центральний маршрутизатор (HUB) – даний роутер працює як DMVPN сервер, і Spoke маршрутизатори працюють як DMVPN клієнти.
- У даного маршрутизатора є публічна статична IP-адреса на WAN інтерфейсі.
- В Spoke маршрутизаторів на WAN інтерфейсах може як статична, так і динамічна публічна IP-адреса.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40



Початок налаштування системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN



Рисунок 3.1 – Структурна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0014.00.00.ПЗ

Арк.

42

На сторінці dmvpn/config представлені схема й конфігураційні файли для віртуальної мережі Xentaур, що розглядається на цій сторінці.

Центральний офіс (що перебуває за dyn1) з'єднаний з декількома віддаленими (що перебувають за dyn3 і dyn5). Необхідно забезпечити зв'язок віддалених офісів із центральним і, при необхідності, установлювати тунель між віддаленими офісами.

Маршрутизація між мережами офісів буде виконуватися за допомогою протоколів динамічної маршрутизації.

Зображена топологія називається – зірка (hub-and-spoke), де:

- dyn1 – центр зірки, hub-маршрутизатор;
- dyn3 і dyn5 – вершини зірки, spoke-маршрутизатори.

Рішення

Якщо в даній схемі використовувати звичайну віртуальну приватну мережу, що з'єднує мережі (site -to-site VPN), то на маршрутизаторах, які перебувають у віддалених офісах, необхідно буде налаштувати тунель для зв'язку із центральним офісом і тунелі для зв'язку з іншими віддаленими офісами. Для того щоб всі віддалені офіси могли працювати між собою прямо, необхідно буде налаштувати тунелі з усіма офісами, що приведе до створення повнозв'язної топології (full mesh). Як наслідок, кількість налаштувань на маршрутизаторах, як у центральному офісі, так і у віддалені істотно збільшується.

Технологія DMVPN дозволяє вирішити це завдання більше масштабованим методом, чим створення зв'язків точка-точка між всіма офісами й об'єднання їх у повнозв'язну топологію:

- При додаванні нових маршрутизаторів в існуючу мережу DMVPN, необхідно налаштувати тільки новий маршрутизатор, змін на вже існуючих маршрутизаторах не потрібно.
- DMVPN дозволяє використовувати динамічно призначені IP-адреси на spoke-маршрутизаторах.

– Якщо двом spoke-маршрутизаторам необхідно встановити тунель прямо, то він установлюється динамічно.

В основі DMVPN лежать кілька технологій:

- mGRE-тунелі;
- Протокол NHRP (Next Hop Resolution Protocol);
- Протоколи динамічної маршрутизації;

Так як DMVPN використовує кілька технологій, то пошук несправностей у його налаштуваннях може займати досить багато часу. Для того щоб уникнути помилок, бажано поетапно перевіряти виконані налаштування, роботу відповідних протоколів і доступність мереж.

Нижче опишемо послідовність дій при налаштуванні системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Базові налаштування мережі

Налаштування мережі DMVPN будемо виконувати на умовному прикладі, що відбиває ключові особливості мереж, у яких доцільно використовувати цю технологію.

За кожним маршрутизатором перебуває мережа, що імітує локальну мережу офісів:

- dyn1 – 10.1.1.0/24;
- dyn3 – 10.1.3.0/24;
- dyn5 – 10.1.5.0/24.

Для зовнішніх інтерфейсів обрані мережі:

- dyn1 – 192.168.1.0/24;
- dyn3 – 192.168.3.0/24;
- dyn5 – 192.168.5.0/24.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Налаштування маршрутизації між зовнішніми інтерфейсами маршрутизаторів

У даній тестовій мережі на кожному маршрутизаторі прописані статичні маршрути до мереж зовнішніх інтерфейсів інших маршрутизаторів.

Статичні маршрути на hub-маршрутизаторі (dyn1).

Але можна зробити простіше – прописати default GW.

mGRE-тунелі

У топології зірка (hub-n-spoke) використання GRE-тунелів точка-точка приведе до великої кількості налаштувань, так як IP-адреси всіх spoke-маршрутизаторів повинні бути відомі й настроєні на центральному маршрутизаторі (hub).

Альтернативою GRE-тунелів точка-точка є multipoint GRE (mGRE) тунель, що дозволяє термінувати на собі декілька GRE -тунелів. mGRE-тунель дозволяє одному GRE-інтерфейсу підтримувати декілька IPsec-тунелів і спрощує кількість і складність налаштувань, у порівнянні з GRE-тунелями точка-точка.

Крім того, mGRE-інтерфейс дозволяє використовувати динамічно призначені IP-адреси на spoke-маршрутизаторах.

Налаштування mGRE-тунелів

Для адресації тунелів виділена мережа 10.10.10.0/24, всі тунелі перебувають в одній мережі. На hub-маршрутизаторі й spoke-маршрутизаторах mGRE-тунелі настроюються аналогічно (далі приклад для hub-маршрутизатора).

Створення тунельного інтерфейсу.

Завдання IP-адреси на інтерфейсі.

Так як GRE додає додаткові заголовки до IP -пакета, необхідно змінити значення MTU на інтерфейсі.

Як адреса відправника в пакеті вихідному з mGRE-інтерфейсу буде використовуватися IP-адреса фізичного інтерфейсу, а адреса одержувача буде виучений динамічно за допомогою протоколу NHRP.

Налаштування відповідності між тунельним інтерфейсом і фізичним.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Включення mGRE-тунелю.

(Опціонально) Завдання ключа, що ідентифікує тунель.

Якщо буде використовуватися топологія із двома hub-маршрутизаторами й, відповідно, двома мережами DMVPN, то ключ указує на приналежність інтерфейсу однієї з мереж DMVPN.

Next Hop Resolution Protocol (NHRP)

Next Hop Resolution Protocol (NHRP) – клієнт-серверний протокол перетворення адрес, що дозволяє всім хостам, які перебувають в NBMA(Non Broadcast Multiple Access)-мережі, динамічно вивчити NBMA-адреси (фізичні адреси) один одного звертаючись до next-hop-сервера (NHS). Після цього хости можуть обмінюватися інформацією один з одним прямо.

У мережі DMVPN:

– Hub-маршрутизатор буде працювати як NHS, а spoke-маршрутизатори будуть клієнтами.

– Hub-маршрутизатор зберігає й обслуговує базу даних NHRP, у якій зберігаються відповідності між фізичними адресами й адресами mGRE-тунелів spoke-маршрутизаторів.

– На кожному spoke-маршрутизаторі hub-маршрутизатор статично зазначений як NHS і задан відповідність між фізичною адресою й адресою mGRE-тунелю hub-маршрутизатора.

– При включенні кожний spoke-маршрутизатор реєструється на NHS і, при необхідності, запитує в сервера інформацію про адреси інших spoke-маршрутизаторів для побудови spoke-to-spoke тунелів.

– Network ID – характеристика локальна для кожного маршрутизатора (аналогія – OSPF process ID). Різні інтерфейси можуть брати участь у різних NHRP сесіях; це просто розмежувач того, що сесія NHRP на одному інтерфейсі не та що на іншому. Відповідно, зовсім не обов'язково, щоб Network ID збігалися на різних маршрутизаторах.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Налаштування NHRP

Налаштування NHRP на hub-маршрутизаторі

Включення NHRP на інтерфейсі.

Hub-маршрутизатор буде автоматично додавати відповідності між адресами spoke-маршрутизаторів.

(Опціонально) налаштування автентифікації.

Налаштування NHRP на spoke-маршрутизаторах

Включення NHRP на інтерфейсі.

Адреса тунельного інтерфейсу hub-маршрутизатора вказується як next-hop-сервер.

Статична відповідність між адресою mGRE-тунелю й фізичною адресою hub-маршрутизатора (перша адреса – адреса тунельного інтерфейсу, друга – адреса зовнішнього фізичного інтерфейсу).

Адреса зовнішнього фізичного інтерфейсу hub-маршрутизатора вказується як одержувач multicast-пакетів від локального маршрутизатора.

(Опціонально) Налаштування автентифікації.

(Опціонально) Налаштування прапора неунікальності ір-адреси тунелю в базі nhrp на hub-маршрутизаторі.

Якщо зміниться зовнішня адреса spoke-маршрутизатора й цієї команди не буде, то hub-маршрутизатор не оновить свою базу nhrp через помилку: unique address registered already

Перевірка роботи mGRE-тунелів і протоколу NHRP

Перевірка тунельних інтерфейсів

Після того, як на маршрутизаторах настроєні mGRE-тунелі й настроєний протокол NHRP, тунелі перебувають у стані up і всі IP-адреси тунелів доступні.

На dyn5.

Перегляд інформації про тунельний інтерфейс.

З dyn5 пінгується й dyn1 і dyn3.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Перевірка NHRP на spoke-маршрутизаторі

Інформація про NHS.

Сумарна інформація NHRP (уже встановлена динамічний тунель із dyn3).

Докладна інформація про інші маршрутизатори отримана по NHRP.

Одержувач multicast-трафіку.

Сумарна інформація про записи NHRP.

Статистика по пакетах NHRP.

Перевірка NHRP на hub-маршрутизаторі

Сумарна інформація NHRP.

Докладна інформація про інші маршрутизатори отримана по NHRP.

Одержувачі multicast-трафіку.

Сумарна інформація про записи NHRP.

Статистика по пакетах NHRP.

Налаштування маршрутизації

Розглядається використання протоколів EIGRP і OSPF для налаштування маршрутизації між мережами офісів. Для реального використання досить вибрати один із цих протоколів, і виконати відповідні йому налаштування.

Налаштування EIGRP

Якщо як протокол маршрутизації буде використовуватися EIGRP, то необхідно.

- Включити EIGRP для мереж mGRE-інтерфейсів і локальних мереж;
- Відключити автоматичне підсумовування мереж по класовій ознаці;
- Налаштувати EIGRP на mGRE-інтерфейсі.

Включення EIGRP для мереж mGRE-інтерфейсів і локальних мереж.

Відключення автоматичного підсумовування мереж.

Налаштування EIGRP на mGRE-інтерфейсах

Для роботи в мережі DMVPN необхідно додатково налаштувати EIGRP.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

На hub-маршрутизаторі необхідно відключити правило розщеплення обрію (split horizon), інакше EIGRP не буде анонсувати маршрути, виучені через mGRE-інтерфейс назад у цей же інтерфейс.

За замовчуванням EIGRP буде підставляти IP-адреса hub-маршрутизатора в якості next-hop для маршрутів які він анонсує, навіть коли анонсує маршрути назад через той же інтерфейс, на якому вони були виучені. Для мережі DMVPN необхідно щоб EIGRP використовував у якості next-hop адреси spoke-маршрутизаторів. Тому на hub-маршрутизаторі необхідно відключити це правило:

При використанні DMVPN у більших мережах час збіжності мережі може збільшуватися. Для того щоб уникнути можливих проблем з маршрутизацією, на маршрутизаторах необхідно змінити hold time (за замовчуванням 15 секунд).

Якщо на одному з маршрутизаторів цей параметр змінений, то сусіди цього маршрутизатора будуть використовувати цей таймер. Для того щоб маршрутизатор сам використовував певне значення таймера, необхідно змінити таймер на відповідному інтерфейсі сусіда.

Перевірка роботи EIGRP

Сусіди EIGRP на dun1.

Сусіди EIGRP на dun5.

Таблиця маршрутизації dun1.

Таблиця маршрутизації dun3 (мережа 10.1.5.0 доступна через dun5, а не через dun1).

Налаштування OSPF

Для OSPF обрана топологія в якій mGRE-інтерфейси перебувають у зоні 0, а локальні мережі – кожна у своїй зоні. Мережа розбита на зони, так як це дозволяє підсумувати мережі усередині зони й не передавати в інші офіси докладну інформацію про мережі (для даного приклада це не актуально, але в реальній мережі може істотно зменшити таблицю маршрутизації й кількість обчислень найкоротшого шляху).

						ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			49

На маршрутизаторах необхідно.

- Включити OSPF для мереж mGRE-інтерфейсів у зоні 0;
- Включити OSPF для локальних мереж у відповідній зоні;
- Налаштувати OSPF на mGRE-інтерфейсі.

Включення OSPF для мереж mGRE-інтерфейсів у зоні 0.

Включення OSPF для локальних мереж у відповідній зоні (офісу за dyn1 відповідає зона 1).

Налаштування OSPF на mGRE-інтерфейсі

Так як mGRE-інтерфейси утворюють NBMA-мережа, то на цих інтерфейсах необхідно змінити налаштування OSPF. Докладніше про роботу OSPF в NBMA-мережах на сторінці OSPF в Cisco.

На hub-маршрутизаторі й spoke-маршрутизаторах відрізняються налаштування пріоритетів, інші налаштування збігаються.

Пріоритет інтерфейсу впливає на те, який маршрутизатор буде обраний виділеним маршрутизатором (DR) для мережі. У топології "зірка" роль DR повинен взяти на себе центральний маршрутизатор.

На hub-маршрутизаторі пріоритет встановлюється 10.

На spoke-маршрутизаторах пріоритет встановлюється 0 для того щоб маршрутизатори не могли брати участь у виборах DR.

Вказівка типу мережі OSPF на mGRE-інтерфейсі.

Зміна інтервалу між відправленням hello-пакетів.

Перевірка роботи OSPF

Сусіди OSPF на dyn1.

Сусіди OSPF на dyn3.

Таблиця маршрутизації на dyn1.

Таблиця маршрутизації на dyn3.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Фази DMVPN

В DMVPN розрізняють три фази або версії: першу, другу й третю.

Розглянемо докладніше кожен з них. Як протокол динамічної маршрутизації для наочності будемо використовувати EIGRP.

Перша фаза

У першій фазі допускається динамічне підключення spoke-маршрутизаторів до hub, при цьому вся взаємодія між мережами, розташованими за spoke, ведеться через центр – через hub-маршрутизатор. Тобто в першій фазі неможливо пряма взаємодія між spoke-маршрутизаторами. Всі spoke-маршрутизатори в даній фазі використовують тільки point-to-point тунелі.

До переваг першої фази можна віднести можливість значного скорочення кількості маршрутів, що перебувають у таблицях маршрутизації spoke-пристроїв за рахунок агрегації або фільтрації маршрутів, анонсуємих hub-маршрутизатором. Виродженням прикладом є анонсування лише дефолтного маршруту з боку hub.

У цьому випадку на spoke-пристроях по EIGRP буде відомий тільки один маршрут.

Перевіримо, що трафік між spoke-маршрутизаторами передається через hub.

До недоліків першої фази ставиться неоптимальний маршрут проходження трафіку між spoke-маршрутизаторами.

Друга фаза

Друга фаза містить у собі оптимізацію шляху проходження трафіку, переданого між spoke-пристроями, за рахунок динамічної побудови тунелів між кінцевими маршрутизаторами, що стає можливим, якщо spoke-маршрутизатори мають повну інформацію про всі префікси в мережі.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Налаштування hub-маршрутизатора повинна задовольняти трьом основним правилам:

- Агрегація мереж або не повинна вироблятися зовсім, або не повинна приховувати реального розташування префіксів;
- На hub-пристрої повинне бути відключене розщеплення обрію;
- Hub-маршрутизатор не повинен підмінювати адресу next-hop, отриманий від spoke-пристрою.

Ніяке додаткове налаштування, що включає підтримку другої фази, не потрібно.

Таблиця маршрутизації spoke-маршрутизатора представлена нижче й містить у собі всі префікси в мережі.

Незважаючи на те, що таблиця маршрутизації містить у собі всі префікси, в CEF не втримується повної інформації про кожну підмережу до моменту передачі першого пакета. Перший пакет убік нової підмережі обробляється процесором і відправляється убік hub-маршрутизатора, після чого завершується формування запису в CEF. До приходу першого пакета запис в CEF позначається як incomplete.

Для завершення формування запису в CEF повинен бути зроблений пошук L2 даних по існуючій L3 інформації. Такий пошук виробляється з використанням протоколу NHRP.

Після завершення роботи протоколу NHRP завершується формування запису в CEF.

Трафік між мережами spoke-маршрутизаторів передається прямо між ними (не через hub).

Перевагом використання другої фази є оптимізація шляхів передачі трафіку між spoke-пристроями. До недоліків можна віднести ріст таблиці маршрутизації на з.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Третя фаза

Третя фаза позбавлена недоліків перших двох фаз, але при цьому володіє їхніми перевагами: трафік передається оптимальним шляхом, при цьому таблиця маршрутизації не зобов'язана містити в собі всі можливі префікси мережі. Домогтися цього вдалося за рахунок приміщення в таблицю маршрутизації тільки тих префіксів, які реально використовуються в цей момент часу. Запис у таблиці маршрутизації з'являється тільки після того, як з'являється трафік, призначений для відповідних одержувачів. Все чарівництво третьої фази стає можливим за рахунок використання опцій `redirects` і `shortcuts`. При використанні третьої фази немає необхідності відмовлятися від підсумовування маршрутів (як це було в першій фазі) і не виникає проблеми з адресами `next-hop` маршрутизаторів (як у другій фазі). По суті `hub`-маршрутизатор у бік `spoke`-пристроїв може повідомляти єдиний маршрут (наприклад, на мережу `0.0.0.0/0`), що відразу ж міститься в `CEF`. У такий спосіб у третій фазі `DMVPN` на маршрутизаторах немає частково заповнених записів в `CEF`, що дозволяє всі пакети (включаючи перший) маршрутизувати із використанням `CEF`, тобто без використання процесора.

Розглянемо процес передачі даних між двома `spoke`-маршрутизаторами. Перший пакет передається у бік `hub`-пристрою відповідно до таблиці маршрутизації, після чого `hub`-маршрутизатор через той же тунельний інтерфейс відправляє пакет у бік маршрутизатора одержувача. Така подія (відправлення пакета через той же інтерфейс, через який він був отриманий) змушує `hub`-маршрутизатор відправити повідомлення `redirect`, що означає, що трафік передається не самим оптимальним шляхом. Включення можливості відправлення `hub`-маршрутизатором повідомлень `NHRP redirect` включається за допомогою команди `ip nhrp redirect`, що вводиться в режимі налаштування тунельного інтерфейсу.

Одержавши повідомлення `redirect` `spoke`-маршрутизатор відправляє повідомлення `NHRP request`, за допомогою якого намагається з'ясувати, `NBMA` адреса пристрою, за яким розташований одержувач. Повідомлення `NHRP request`

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

передається через hub у бік другого spoke-пристрій, що і відповідає на запит. Одержавши відповідь на відповідний запит, spoke-маршрутизатор додає новий запис у таблицю маршрутизації й використовує неї для пересилання трафіку. Для включення опції redirect на тунельних інтерфейсах spoke-пристроїв повинна бути уведена команда `ip nhrp shortcut`.

Трафік між spoke-пристроями передається прямо, тобто минаючи hub-маршрутизатор.

У виводі debug-повідомлень на spoke-маршрутизаторі одержання пакета `nhrp redirect` називається «NHRP: Receive Traffic Indication», тоді як відправлення повідомлення `nhrp request` можна розпізнати по фразі «NHRP: Sending NHRP Resolution Request for dest».

Налаштування IPsec

Перша фаза (налаштування IKE)

Для тестової мережі в політику `isakmp` використовуються налаштування за замовчуванням.

- Алгоритм шифрування: DES – Data Encryption Standard (56 bit keys).
- Алгоритм хешування: Secure Hash Standard.
- Метод автентифікації: Rivest-Shamir-Adleman Signature.
- Група Diffie-Hellman: #1 (768 bit).
- Час життя SA: 86400 seconds, no volume limit.

Автентифікація по pre-shared key

У політику з автентифікацією по pre-shared key, метод автентифікації змінений, так як за замовчуванням використовується автентифікація по сертифікатах. Інші параметри використовують значення за замовчуванням.

Створити політику `isakmp`.

Так як адреса peer заздалегідь не відома й може бути отриманий динамічно, те при налаштуванні pre-shared key необхідно вказувати шаблонуву адресу (wildcard address).

Налаштувати `isakmp pre-shared key`.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Автентифікація по сертифікатах

Автентифікацію по pre-shared key не рекомендується використовувати для DMVPN, так як pre-shared key повинен бути зазначений із шаблоновою адресою, і немає прив'язки до адреси маршрутизатора з яким встановлюється тунель.

У тестовій мережі hub-маршрутизатор буде виконувати роль центра сертифікатів. Всім маршрутизаторам видані сертифікати.

Більш докладно про налаштування центра сертифікатів на маршрутизаторі Cisco і процедурі видачі сертифікатів маршрутизаторам на сторінці Центр сертифікатів на маршрутизаторі Cisco.

Створити політику isakmp (так як за замовчуванням використовується автентифікація по сертифікатах, то в тестових цілях можна залишити всі налаштування політики за замовчуванням).

Друга фаза (налаштування IPsec-профілю)

Так як hub-маршрутизатор не знає заздалегідь IP-адреси spoke-маршрутизаторів, то для другої фази IPsec потрібна динамічна crypto map (dynamic crypto map), але для тунельних інтерфейсів еквівалентом dynamic crypto map є IPsec-profile.

- застосовується на тунельному інтерфейсі;
- після застосування будь-який трафік вихідний з тунельного інтерфейсу ініціює створення IPsec-тунелю (немає необхідності використовувати ACL, як у звичайної crypto map);
- source і destination адреси тунельного інтерфейсу використовуються для створення IPsec-тунелю. Адреси можуть бути прописані в налаштуваннях інтерфейсу або отримані динамічно за допомогою NHRP (не задається адреса peer, як у звичайної crypto map);

Створюємо transform set.

Так як mGRE-інтерфейс забезпечує створення тунелю, то IPsec можна перевести в транспортний режим (за замовчуванням використовується тунельний режим).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Створити IPSec-профіль.

Застосування IPsec-профілю

Застосування IPsec-профілю на mGRE-інтерфейсі.

Перевірка роботи IPsec

Перевірка роботи IKE з автентифікацією по pre-shared key

Перевірка isakmp SA на dyn1.

Перевірка isakmp SA на dyn3 до установки тунелю spoke-to-spoke.

Перевірка isakmp SA на dyn3 після установки тунелю spoke-to-spoke. Для цього досить просто пропінгувати тунельний інтерфейс сусіднього spoke-маршрутизатора (але пакетів істр бажано послати штук 10).

Більше докладна інформація про isakmp SA (автентифікація по pre-shared key).

Перевірка роботи IKE з автентифікацією по сертифікатах

Перевірка isakmp SA на dyn1.

Перевірка isakmp SA на dyn3 до установки тунелю spoke-to-spoke.

Перевірка isakmp SA на dyn3 після установки тунелю spoke-to-spoke. Для цього досить просто пропінгувати тунельний інтерфейс сусіднього spoke-маршрутизатора (але пакетів істр бажано послати штук 10).

Більше докладна інформація про isakmp SA (автентифікація по сертифікатах).

Перевірка роботи IPsec

Перегляд інформації про crypto map на dyn1.

Перегляд інформації про crypto map на dyn3 до створення динамічного тунелю з dyn5.

Перевірка IPsec SA на dyn1.

Перевірка IPsec SA на dyn3 після установки тунелю spoke-to-spoke.

Перегляд інформації про DMVPN мережі

Так як на маршрутизаторі dyn3 запису різних типів – динамічна й статична, те вся інформація перевіряється на ньому.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

відбивають їхній зміст: Next Header указує на наступний заголовок, Payload Len представляє довжину пакета, SPI є покажчиком на контекст безпеки й Sequence Number Field містить послідовний номер пакета.

Послідовний номер пакета був введений в АН в 1997 році в ході процесу перегляду специфікації IPsec. Значення цього поля формується відправником і служить для захисту від атак, пов'язаних з повторним використанням даних процесу автентифікації. Оскільки мережа Інтернет не гарантує порядок доставки пакетів, одержувач повинен зберігати інформацію про максимальний послідовний номер пакета, що пройшов успішну автентифікацію, і про одержання деякого числа пакетів, що містять попередні послідовні номери (звичайно це число дорівнює 64).

На відміну від алгоритмів обчислення контрольної суми, застосовуваних у протоколах передачі інформації з лініям зв'язку, що комутуються або по каналах локальних мереж і орієнтованих на виправлення випадкових помилок середовища передачі, механізми забезпечення цілісності даних у відкритих телекомунікаційних мережах повинні мати засоби захисту від внесення цілеспрямованих змін. Одним з таких механізмів є спеціальне застосування алгоритму MD5: у процесі формування АН послідовно обчислюється хеш-функція від об'єднання самого пакета й деякого попередньо погодженого ключа, а потім від об'єднання отриманого результату й перетвореного ключа. Даний механізм застосовується за замовчуванням з метою забезпечення всіх реалізацій IPv6, принаймні, одним загальним алгоритмом, не підданим експортним обмеженням.

Заголовок ESP – інкапсуляція зашифрованих даних

У випадку використання інкапсуляції зашифрованих даних заголовок ESP є останнім у ряді опціональних заголовків, "видимих" у пакеті. Оскільки основною метою ESP є забезпечення конфіденційності даних, різні види інформації можуть вимагати застосування істотно різних алгоритмів шифрування. Отже, формат ESP може перетерплювати значні зміни залежно від

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

використовуваних криптографічних алгоритмів. Проте, можна виділити наступні обов'язкові поля: SPI (SPI – Security Parameter Index – індекс параметра безпеки), що вказує на контекст безпеки, поле порядкового номера, що містить послідовний номер пакета, і контрольна сума, призначена для захисту від атак на цілісність зашифрованих даних. Крім цього, як правило, у тілі ESP присутні параметри (наприклад, режим використання) і дані (наприклад, вектор ініціалізації) застосовуваного алгоритму шифрування. Частина ESP заголовка може бути зашифрована на відкритому ключі одержувача або на спільному ключі пари відправник-одержувач. Одержувач пакета ESP розшифровує ESP заголовок і використовує параметри й дані застосовуваного алгоритму шифрування для декодування інформації транспортного рівня.

Розрізняють два режими застосування ESP – транспортний і тунельний.

Транспортний режим – використовується для шифрування поля даних IP пакета, що містить протоколи транспортного рівня (TCP, UDP, ICMP), який, у свою чергу, містить інформацію прикладних служб. Прикладом застосування транспортного режиму є передача електронної пошти. Всі проміжні вузли на маршруті пакета від відправника до одержувача використовують тільки відкриту інформацію мережного рівня й, можливо, деякі опціональні заголовки пакета (в IPv6). Недоліком транспортного режиму є відсутність механізмів приховання конкретних відправника й одержувача пакета, а також можливість проведення аналізу трафіку. Результатом такого аналізу може стати інформація про об'єми й напрямки передачі інформації, області інтересів абонентів, розташування керівників.

Тунельний режим – припускає шифрування всього пакета, включаючи заголовки мережного рівня. Тунельний режим застосовується якщо буде потреба приховання інформаційного обміну організації із зовнішнім миром. При цьому, адресні поля заголовка мережного рівня пакета, що використовує тунельний режим, заповнюються міжмережним екраном організації й не містять інформації про конкретного відправника пакета. При передачі інформації із зовнішнього

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

коли кіцеві системи не належать одній й тій же системі адміністративного керування. Наприклад, метод DH можна використовувати в електронній комерції при встановленні з'єднання для передачі транзакцій між двома компаніями.

Хоча цей метод і вимагає більших обчислювальних ресурсів, при його застосуванні можливий компроміс між криптостійкістю алгоритму (при використанні менш довгих відкритих ключів) і необхідним об'ємом обчислень. Обмін ключами у швидкому режимі не вимагає великого об'єму обчислень, так як тут використовується набір простих математичних операцій. Існує обмеження припустимого числа швидких фаз, перевищення якого веде до того, що ключі, згенеровані в основній фазі, а потім використовувані у швидких фазах, виявляться під погрозою розкриття. На сьогоднішній день немає твердого правила, що визначає число швидких фаз на одну основну фазу; криптографи діють, керуючись загальними міркуваннями й з огляду на оперативну обстановку.

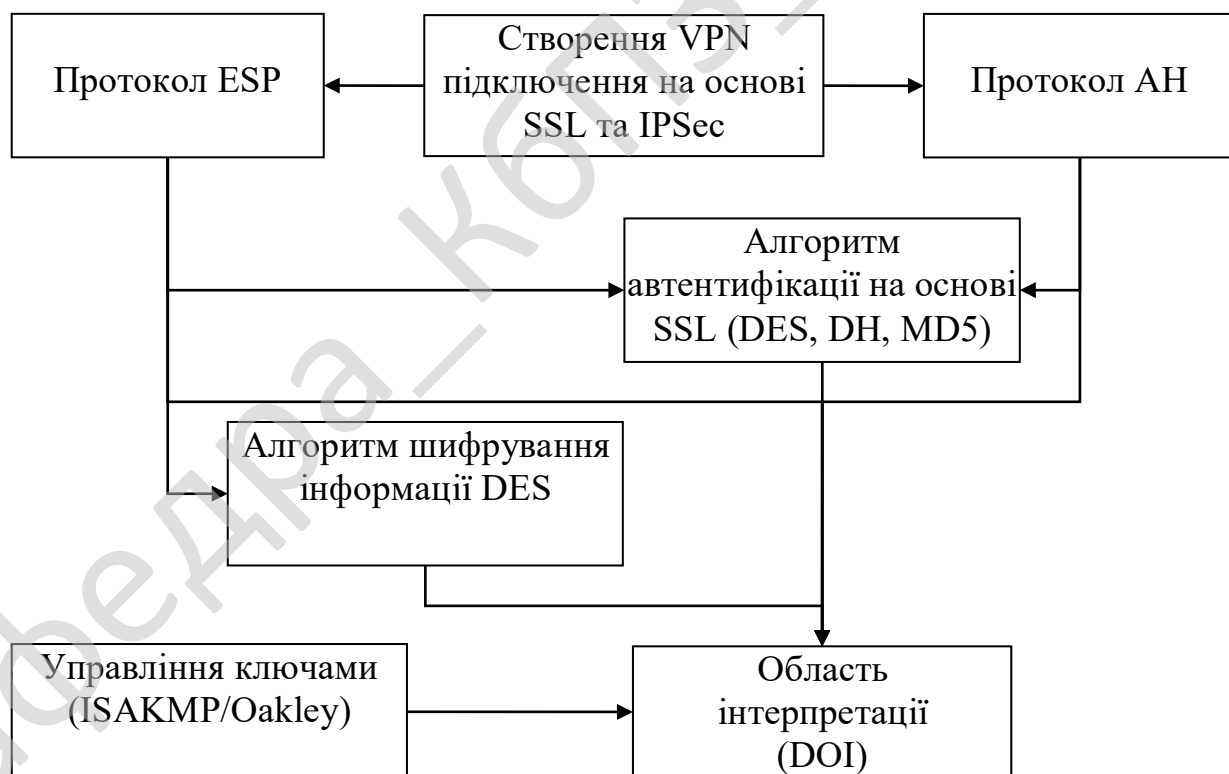


Рисунок 3.2 – Функціональна схема системи

В основному режимі обоє учасника обміну встановлюють SA-угоди для безпечного спілкування один з одним по протоколу Oakley. У швидкому режимі SA-угоди встановлюються вже "від імені" протоколу IPsec або будь-якої іншої служби, який необхідні дані для генерації ключів або узгодження параметрів. Протокол Oakley розроблений таким чином, що він ніяк не пов'язаний з IPsec. Наприклад, для підвищення безпеки процесу встановлення сеансів його цілком можна використовувати разом із протоколом SSL (Secure Sockets Layer) версії 4.0 замість механізму обміну ключами SSL 3.0.

DOI – область інтерпретації

Протокол ISAKMP/Oakley не був спеціально розроблений для спільного використання із протоколом IPsec, тому виникає необхідність у так званій області інтерпретації (Domain Of Interpretation – DOI), що забезпечила б спільну роботу протоколів IPsec і ISAKMP/Oakley. Щоб інші протоколи також могли використовувати ISAKMP/Oakley, вони повинні мати власні DOI-області. У даний момент таких областей для інших протоколів не існує, але ситуація може змінитися на черговій конференції групи IETF або в тому випадку, якщо приватний розроблювач, наприклад фірма Netscape, вирішить використовувати цей механізм. Більш докладно про це можна прочитати в документі "The Internet Key Exchange (IKE)", розробленому робочою групою IP Security Protocol Working Group (<ftp://ftp.ietf.org/internet-draft/draft-ietf-ipsec-isakmp-oakley-06.txt>).

В основному режимі між сторонами погоджуються методи шифрування, хешування, автентифікації й так звана група DH (їх усього чотири), що визначає криптографічну стійкість алгоритму відкритого розподілу ключів. Перша група DH характеризується високою стійкістю й дозволяє використовувати стандарт DES, у той час як для другої й третьої груп варто застосовувати Triple DES.

Оскільки в основному режимі іноді потрібно передавати до шести пакетів, то, наприклад, при використанні космічного сегмента з великою тимчасовою затримкою, DES краще застосовувати з більш сильною групою DH. Тоді перед виконанням чергового основного режиму, сполученого з інтенсивними

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

обчисленнями й обміном пакетами, вам вдасться виконати більше обмінів у швидкому режимі.

Коли SA-угода для обміну по протоколу Oakley устанавлюється в основному режимі, створюється ланцюжок випадкових біт, що використовують для генерації ключів. Також визначається тривалість (за часом або кількістю переданих даних) "життя" SA-угоди Oakley і дані для генерації ключів до того, як буде потрібно наступний обмін в основному режимі.

Швидкий режим простіше основного, і узгодження SA для IPsec здійснюється за допомогою трьох пакетів. IPsec-ключі створюються за допомогою простих операцій піднесення в ступінь переданих в основному режимі даних. У швидкому режимі погодяться також алгоритми шифрування й строки існування SA для IPsec-сеансів.

Згідно із цими строками визначається, як незабаром, залежно від часу або об'єму переданих даних, буде потрібно нове узгодження у швидкому режимі. Помітьте, є два різних строки існування SA -угоди. Основний режим задає його для протоколу Oakley, а швидкий – для обміну по протоколу IPsec. Як приклад пропонуємо значення цих параметрів для шифрування IPsec-сеансів за допомогою алгоритму DES: 15 хв або 10 Мбайт для швидкого режиму, і 60 хв або 40 Мбайт для основного. Ці числа варто збільшити для Triple DES і зменшити для ARCFour (в ARCFour застосовується 40-бітний, а в TripleDES – 112-бітний ключ). Такий підхід дозволяє збалансувати криптографічну стійкість сервісів IPsec і вартість накладних витрат на передачу пакетів ISAKMP/Oakley.

При генерації ключів в основному режимі сеанс можна примусово перервати на підставі відкликання сертифіката. Сертифікати кінцевих вузлів використовуються тільки під час основного режиму. Таким чином, при анулюванні одного із сертифікатів обмін перерветься тільки в основному режимі. Тимчасові обмеження, погоджені в основному й швидкому режимах, значно відрізняються друг від друга й залежать від типу даних і транзакцій, що використовують IPsec-з'єднання. Для правильного визначення цих обмежень із

обліком, з одного боку, об'єму обчислень і навантаження на мережу, а з іншого боку – імовірності порушення захисту даних, потрібно деякий аналіз.

Сполучення різних IPsec-механізмів забезпечує цілком безпечні з'єднання як між мережами, так і між кінцевими станціями. Оскільки практично всі постачальники підтримують ці стандарти, рано або пізно це приведе до виникнення середовища для реалізації безпечних з'єднань через Інтернет. Таким чином, протокол IPsec стане основним для безпечної е-комерції в Інтернет.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

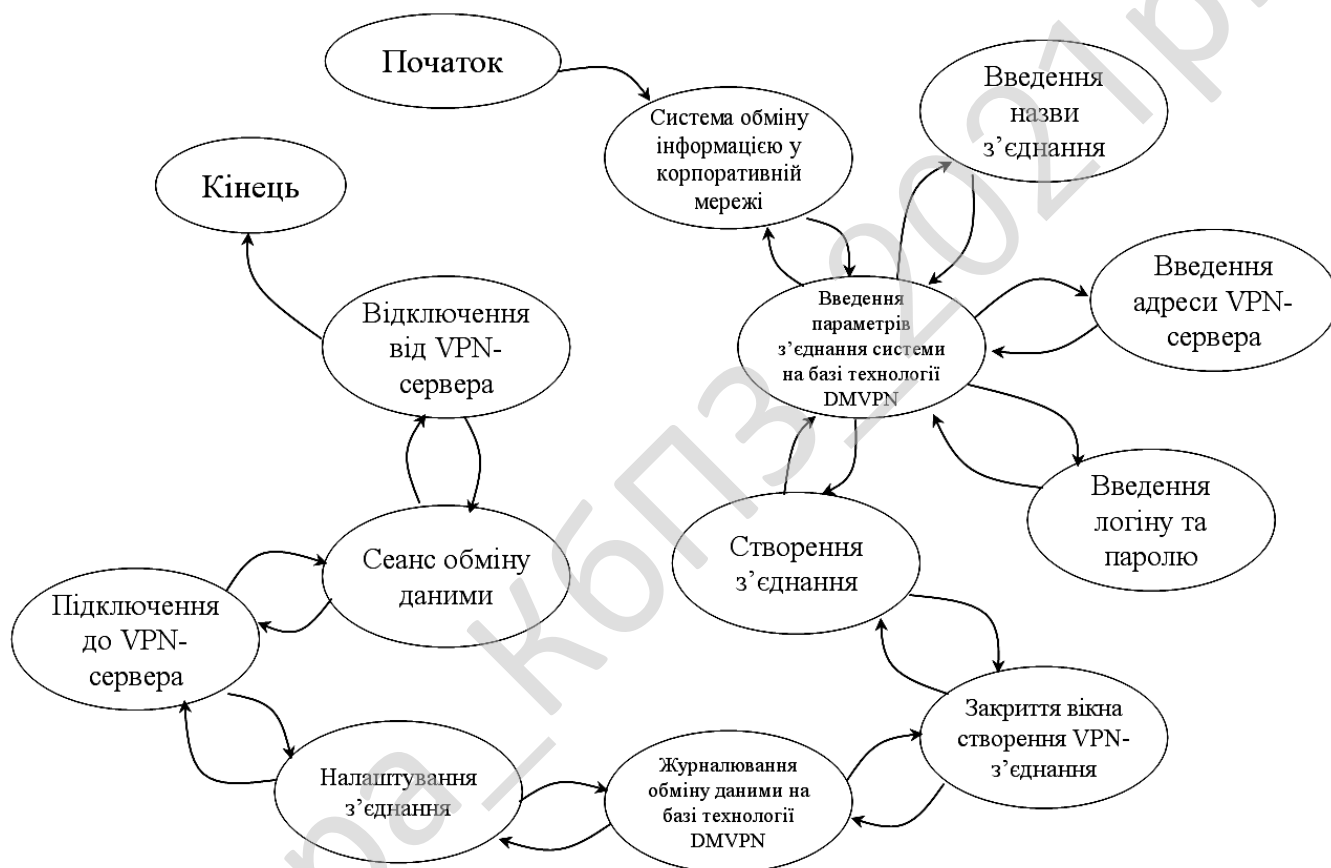


Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію магістерської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень. Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

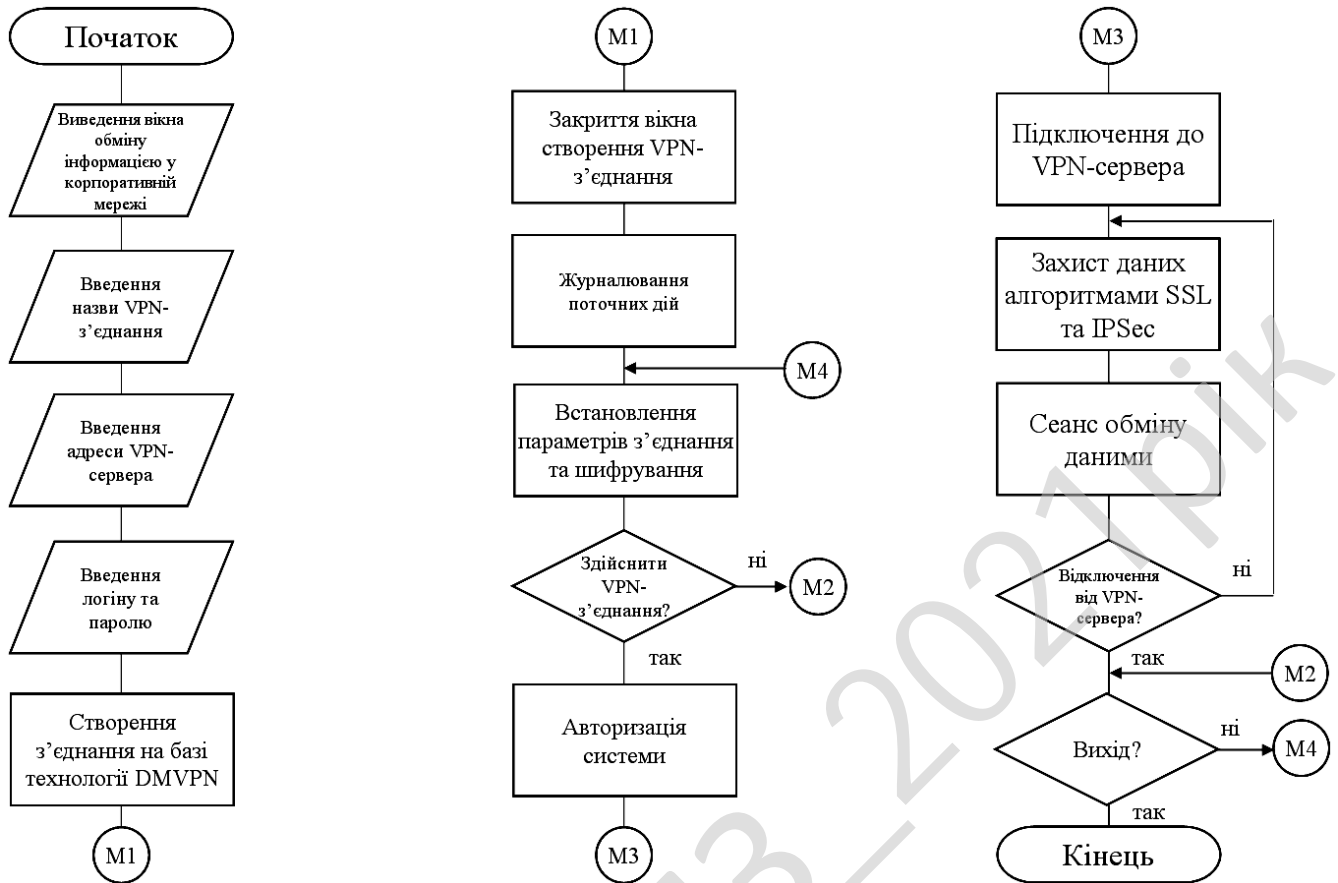


Рисунок 4.1 – Блок-схема основної програми

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

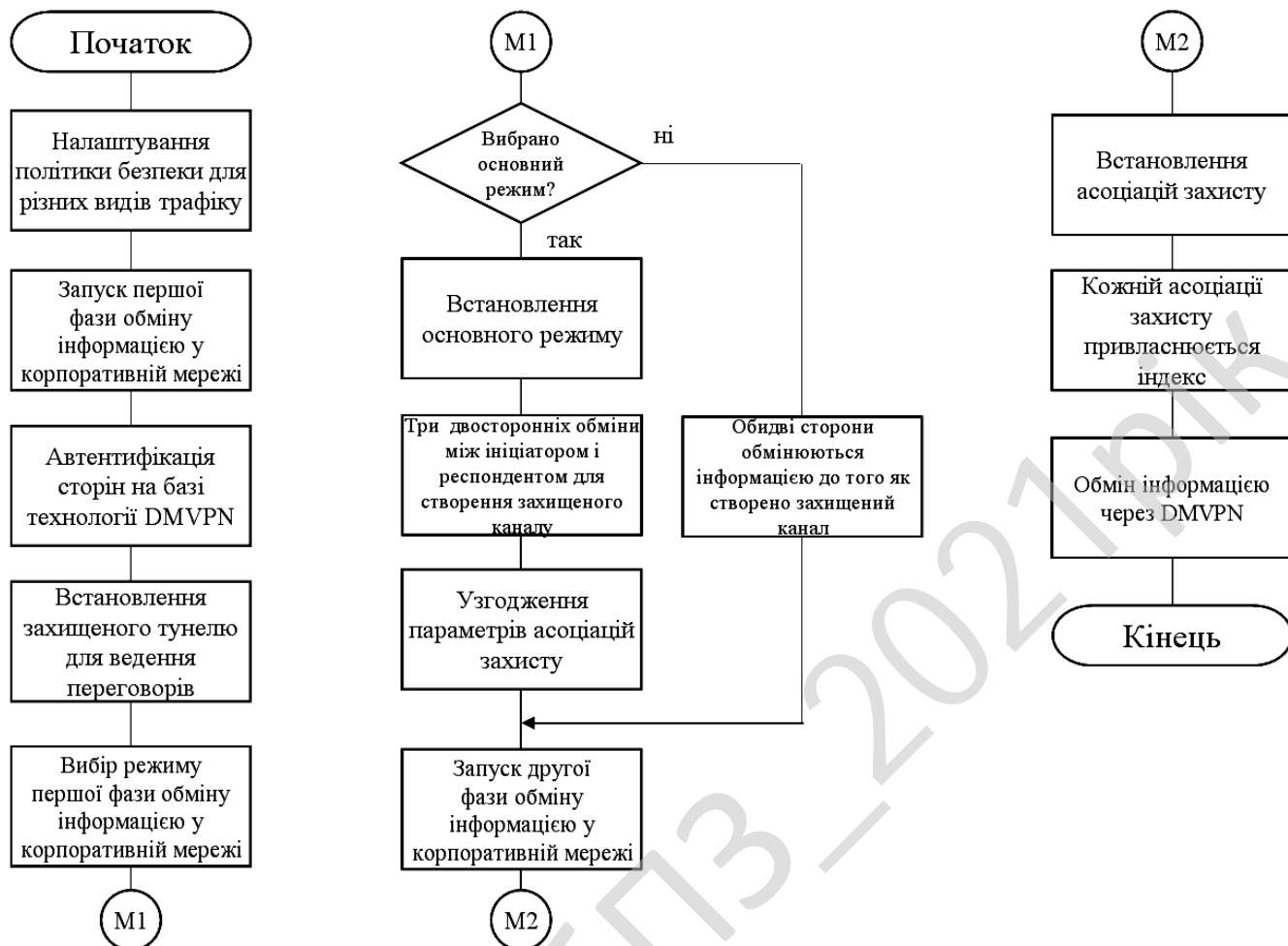


Рисунок 4.2 – Блок-схема роботи підпрограми

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інстальювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Розглянемо протокол SNMP (Simple Network Management Protocol, простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP.

На основі концепції TMN в 1980–1990 р. різними органами стандартизації був вироблений ряд протоколів керування мережами передачі даних з різним спектром реалізації функцій TMN. До одного з типів таких протоколів керування належить Simple Network Management Protocol (SNMP).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й програмами в мережі зв'язку шляхом обміну керуючою інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як сукупність мережних керуючих станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керуючими станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC).

Зазвичай при використанні SNMP присутні керовані та керуючі системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керуючій системі. По суті SNMP агенти передають управлінську інформацію на керуючі системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість працюючих процесів» тощо).

Керуюча система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM. Управляючі системи можуть також відправляти конфігураційні оновлення або контролюючі запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керуючої Інформації (Management Information Bases (MIBs)).

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керуючої Інформації (MIB – Management Information Base).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Бази Керуючої Інформації описують структуру керуючої інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (object identifier (OID)).

Грубо кажучи, кожен унікальний ідентифікатор об'єкта ідентифікує змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1.

Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями. Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт включає в себе один або більше екземплярів об'єкта (що ідентифікуються за OID), які насправді є змінними.

Існує два типи керованих об'єктів: Скалярні об'єкти визначають єдиний екземпляр об'єкта; Табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB.

Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора.

Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB.

В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, пе редакці і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

зв'язаними змінними для змінної, яка є наступною в базі MIB в лексикографічному порядку. Обхід всієї бази MIB агента може бути проведений ітераційним використанням GetNextRequest, починаючи з OID 0. Рядки таблиці можуть бути прочитані, якщо вказати в запиті OID-и колонок в пов'язаних змінних.

4. GetBulkRequest. Покращена версія GetNextRequest. Запит від менеджера до об'єкту для численних ітерацій GetNextRequest. Менеджеру буде повернений Response з декількома пов'язаними змінними, обійденими починаючи зі пов'язаної змінної (змінних) в запиті. Специфічні для PDU поля non-repeaters і max-repetitions використовуються для контролю за поведінкою відповіді. GetBulkRequest був введений в SNMPv2.

5. Response. Повертає зв'язані змінні і значення від агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest і InformRequest. Повідомлення про помилки забезпечуються полями статусу помилки і індексу помилки.

Ця одиниця використовується як відповідь і на Get-, і на Set-запити, в SNMPv1 називається GetResponse.

6. Trap. Асинхронне повідомлення від агента – менеджеру. Включає в себе поточне значення sysUpTime, OID, що визначає тип trap (пастки), і необов'язкові зв'язані змінні. Адресація одержувача для пасток визначається за допомогою змінних trap-конфігурації в базі MIB. Формат trap -повідомлення був змінений в SNMPv2 і PDU перейменували в SNMPv2-Trap.

7. InformRequest. Асинхронне повідомлення від менеджера менеджеру або від агента менеджеру. Повідомлення від менеджера менеджеру були можливі вже в SNMPv1 (за допомогою Trap), але SNMP зазвичай працює на протоколі UDP, в якому доставка повідомлень не гарантована, і не повідомляється про втрачені пакетах. InformRequest виправляє це зворотним відправленням підтвердження про отримання. Одержувач відповідає Response-му, що повторює всю інформацію з InformRequest. Цей PDU був введений в SNMPv2.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

NetBIOS (Network Basic Input/Output System) – протокол для роботи в локальних мережах на персональних ЕОМ типу IBM/PC, розроблений у вигляді інтерфейсу, який не залежить від фірми–виробника. Був розроблений фірмою Sytek Corporation за замовленням IBM в 1983 році. Він включає в себе інтерфейс сеансового рівня (англ. NetBIOS interface), в якості транспортних протоколів використовує TCP і UDP.

Особливістю NetBIOS є можливість його роботи поверх різних протоколів, найпоширенішими/відомими з яких є NetBEUI, IPX і стек протоколів TCP/IP; причому якщо старі версії Windows орієнтувалися на більш легкі в реалізації і менш ресурсомісткі NetBEUI і IPX, то сучасні Windows орієнтуються на TCP/IP.

При використанні NetBEUI і IPX NetBIOS сам забезпечує надійність доставки даних (функціональність SPX не використовувати), а при використанні TCP/IP надійність доставки забезпечує TCP, за що удостоївся окремого імені «NBT».

Інтерфейс NetBIOS являє собою типовий інтерфейс взаємодії програм (API) для забезпечення мережових операцій введення–виведення і управління транспортним протоколом.

Програми, що використовують NetBIOS API інтерфейс, можуть працювати тільки при наявності протоколу, що допускає використання такого інтерфейсу.

NetBIOS також визначає протокол, що функціонує на сеансовому/транспортному рівнях моделі OSI. Цей протокол використовується протоколами нижчих рівнів, такими як NBFP (NetBEUI) і NetBT для виконання мережових запитів вводу–виводу і операцій, описаних в стандартному інтерфейсному наборі команд NetBIOS.

Тобто, NetBIOS сам не підтримує виконання файлових операцій. Ця функція покладається на протоколи нижчих рівнів, а сам NetBIOS забезпечує тільки зв'язок з цими протоколами і NetBIOS API–інтерфейс.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Пакет NETBIOS створений для використання групою EOM. Підтримує як режим сесій (робота через з'єднання), так і режим дейтаграм (без встановлення з'єднання). 16-символьні імена об'єктів в NETBIOS розподіляються динамічно.

NETBIOS має власну DNS, яка може взаємодіяти з інтернетівськй. Ім'я об'єкта при роботі з NETBIOS не може починатися з символу *.

Додатки можуть знайти через NETBIOS потрібні їм ресурси, встановити зв'язок і послати або отримати інформацію. NETBIOS використовує для служби імен порт 137, для служби дейтаграм – порт 138, а для сесій – порт 139. Будь-яка сесія починається з NETBIOS-запиту, завдання IP-адреси і визначення TCP-порту віддаленого об'єкта, далі йде обмін NETBIOS-повідомленнями, після чого сесія закривається.

Сесія здійснює обмін інформацією між двома NETBIOS – додатками. Довжина повідомлення лежить в межах від 0 до 131 071 байт. Припустимо одночасне встановлення декількох сесій між двома об'єктами.

При організації IP-транспорту через NETBIOS IP-дейтаграмма вкладається в NETBIOS-пакет. Інформаційний обмін відбувається в цьому випадку без встановлення зв'язку між об'єктами. Імена NETBIOS повинні містити в собі IP-адреси. Так, частина NETBIOS-адреси може мати вигляд IP.XX.XX. XX.XX, де IP вказує на тип операції (IP через Netbios), а XX. XX. XX.XX – IP- адреса.

Система NETBIOS має власну систему команд (call, listen, hang up, send, receive, session status, reset, cancel, adapter status, unlink, remote program load) і примітивів для роботи з дейтаграммами (send datagram, send broadcast datagram, receive datagram, receive broadcast datagram).

Всі кінцеві вузли NETBIOS діляться на три типи:

- широкомовні («b») вузли;
- вузли точка-точка («p»);
- вузли змішаного типу («m»).

IP-адреса може асоціюватися з одним із зазначених типів. В-вузли встановлюють зв'язок зі своїм партнером за допомогою широкомовних запитів. Р-

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

і M-вузли для цієї мети використовують netbios сервер і мен (NBNS) і сервер розподілу дейтаграм (NBDD).

NetBIOS забезпечує:

- реєстрацію і перевірку мережевих імен;
- встановлення і розрив з'єднань;
- зв'язок з підтвердженням доставки інформації;
- зв'язок без підтвердження доставки інформації;
- підтримку управління і моніторингу драйвера і мережевої карти.

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму Md5. Він отримує на вході повідомлення довільної довжини і створює на виході дайджест повідомлення довжиною 128 біт. Алгоритм складається з наступних кроків:

1. Додавання недостаючих біт. Повідомлення доповнюється так, щоб його довжина стала рівна 448 по модулю 512 (довжина $448 \bmod 512$). Це означає, що довжина доданого повідомлення на 64 біта менше, ніж число, кратне 512. Додавання проводиться завжди, навіть якщо повідомлення має потрібну довжину. Наприклад, якщо довжина повідомлення 448 біт, воно доповнюється 512 бітами до 960 біт. Таким чином, число біт, що додаються, знаходиться в діапазоні від 1 до 512.

Додавання складається з одиниці, за якою слідує необхідна кількість нулів.

2. Додавання довжини. 64-бітове представлення довжини початкового (до додавання) повідомлення в бітах приєднується до результату першого кроку. Якщо первинна довжина більша, ніж 264, то використовуються тільки останні 64 біта. Таким чином, поле містить довжину початкового повідомлення по модулю 264.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

В результаті перших двох кроків створюється повідомлення, довжина якого кратна 512 бітам. Це розширене повідомлення представляється як послідовність 512-бітових блоків Y_0, Y_1, \dots, Y_{l-1} , при цьому загальна довжина розширеного повідомлення рівна $L \cdot 512$ бітам. Таким чином, довжина отриманого розширеного повідомлення кратна шістнадцяти 32-бітовим словам.

3. Ініціалізація MD-буфера. У алгоритмі Md5 використовується 128-бітовий буфер для зберігання проміжних і остаточних результатів хеш-функції. Буфер може бути представлений як чотири 32-бітові регістри (A, B, C, D). Ці регістри ініціалізувалися наступними шістнадцятковими числами:

$$A = 01234567$$

$$B = 89abcdef$$

$$C = Fedcba98$$

$$D = 76543210$$

4. Обробка послідовності 512-бітових (16-словних) блоків. Основою алгоритму Md5 є модуль, що складається з чотирьох циклічних обробок, позначений як Hmd5. Чотири цикли мають схожу структуру, але кожен цикл використовує свою елементарну логічну функцію, ff , що позначається, fg , fh і fi відповідно.

Кожен цикл приймає на вхід поточний 512-бітовий блок Y_q , що обробляється в даний момент, і 128-бітове значення буфера ABCD, яке є проміжним значенням дайджесту, і змінює вміст цього буфера. Кожен цикл також використовує четверту частину 64-елементної таблиці $T[1 \dots 64]$, побудованої на основі функції \sin . i -ий елемент T , $T[i]$, що позначається, має значення, рівне цілій частині від $2^{32} \cdot \text{abs}(\sin(i))$, і задане в радіанах. Оскільки $\text{abs}(\sin(i))$ є числом між 0 і 1, кожен елемент T є цілим, яке може бути представлене 32 бітами. Таблиця забезпечує “випадковий” набір 32-бітових значень, які повинні ліквідувати будь-яку регулярність у вхідних даних.

Для отримання $Mdq+1$ вихід чотирьох циклів складається по модулю 2^{32} з Mdq . Складання виконується незалежно для кожного з чотирьох слів в буфері.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

кожному циклі, і кожен елемент таблиці T , що складається з 64 32-бітових слів, використовується тільки один раз. Після кожного кроку циклу відбувається циклічне зрушення вліво чотирьох слів A, B, C і D . На кожному кроці змінюється тільки одне з чотирьох слів буфера $ABCD$. Отже, кожне слово буфера змінюється 16 разів, і потім 17-й раз в кінці для отримання остаточного виходу даного блоку.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1-5.5 зображено розроблене у магістерській дипломній роботі система забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN. Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма. На рисунку 5.1 зображено скриншот вікна створення VPN-з'єднання. На рисунку 5.2 зображено скриншот вікна створення VPN-з'єднання (введення параметрів з'єднання). На рисунку 5.3 зображено скриншот вікна створення VPN-з'єднання (завершення роботи). На рисунку 5.4 зображено скриншот ярлика створеного VPN-з'єднання на робочому столі. На рисунку 5.5 зображено скриншот іконки створеного VPN-з'єднання у папці «Мережні з'єднання».

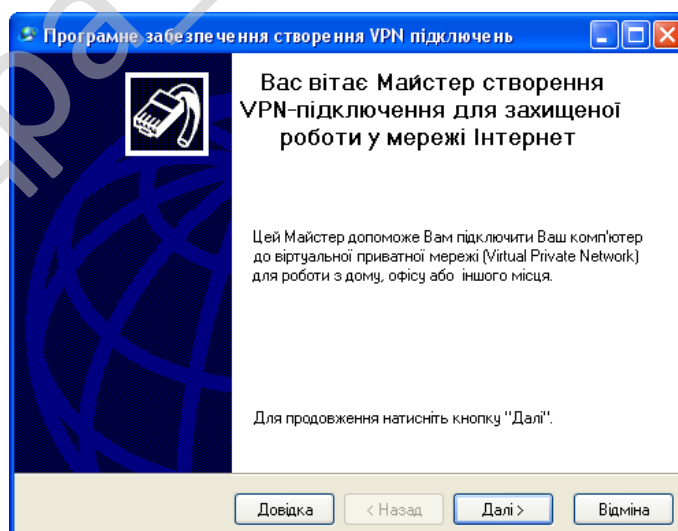


Рисунок 5.1 – Вікно створення VPN-з'єднання

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частиною життєвого циклу програмного забезпечення.

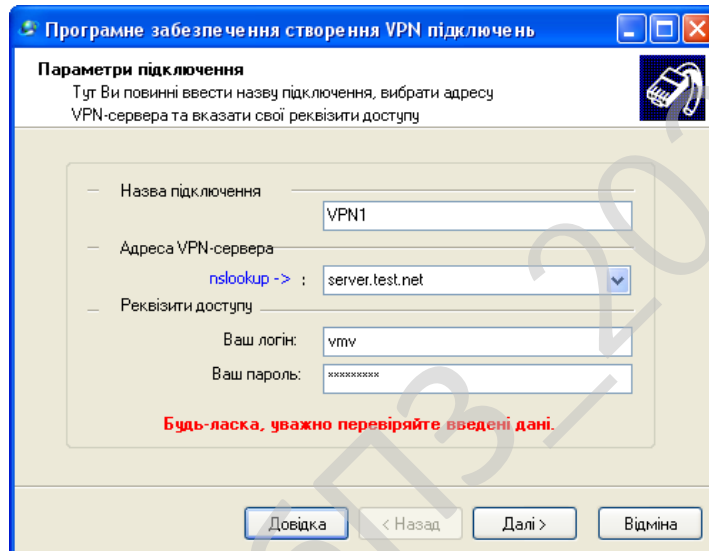


Рисунок 5.2 – Вікно створення VPN-з'єднання (введення параметрів з'єднання)

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

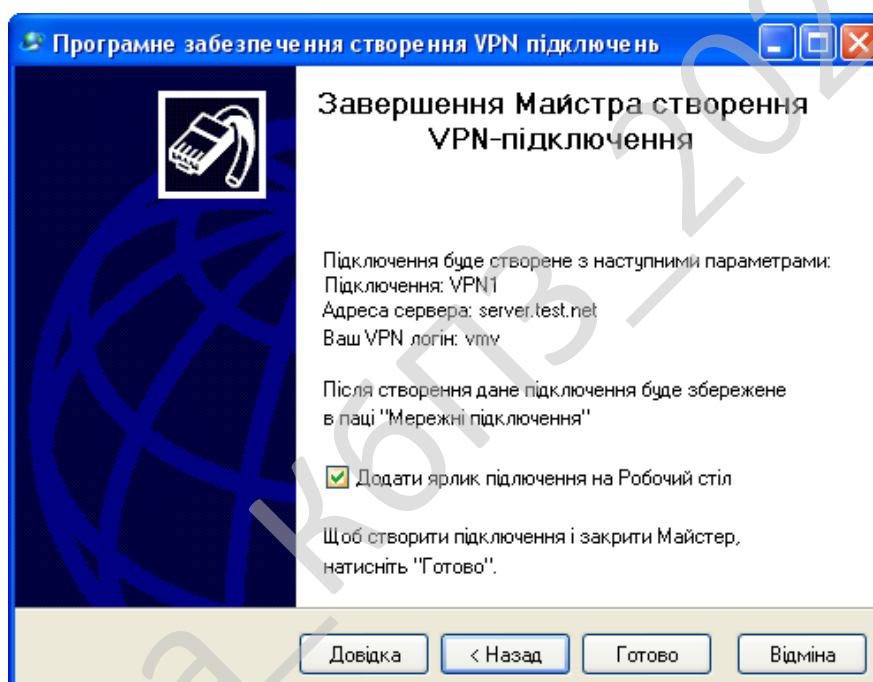


Рисунок 5.3 – Вікно створення VPN-з'єднання (завершення роботи)

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

- Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво

важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

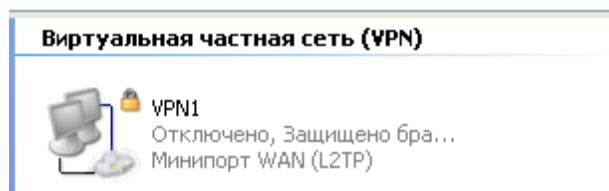


Рисунок 5.4 – Іконка створеного VPN-з'єднання у папці «Мережні з'єднання»

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ).
Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

Кафедра _ КБПЗ _ 2021 рік

					VKPM-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Метою розробки є дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Об'єктом дослідження є процес забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Предметом дослідження є методи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Методи дослідження базуються на методах теорії інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

– Розроблено вітчизняний продукт забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці). В магістерській роботі було проведено дослідження та виконана програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	№	140 (2 ост. цифри № зал*10 ¹)
3. Запланований термін розробки, днів	Frq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0014.00.00.ПЗ

Арк.

95

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	15000 (2 ост. цифри № зал*10 ³)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	37
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	60
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

де: A – коефіцієнт Боєма, $A = 2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1,1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 3,23 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 86 = 168 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1- 7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{nz} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м. п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2012 R2, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0014.00.00.ПЗ

Арк.

100

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0014.00.00.ПЗ

Арк.

101

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	4500	13500
Продакт-менеджер	0,25	4200	3150
Інженер-програміст	3,8	4215	48051
Інженер-електронщик	1,2	4200	15120
Інженер-системотехнік	0,25	4200	3150
Адміністратор мережі	0,5	4236,5	6354,8
Системний програміст	0,25	4200	3150
Дизайнер WEB	0,25	4200	3150
Інженер-верстальник	0,25	4239	3179,25
Бухгалтер-економіст	0,5	4500	6750
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 105555,05$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{105555}{8,25 \cdot 60} = 213 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 200...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 28 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.6.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет-магазину Розетка за 29.10.19 – джерело <http://rozetka.com.ua/>.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10747
Системний блок		7347
Процесор	AMD Phenom II X6 1045T BOX AM3+/AM3/AM2+	1750
Системна плата	MB Gigabyte GA-MA770T-ES3 (770/SB710) AM3, 5200MT/s, DDR3 1666(OC), 1xPCI-E 16x v2.0, 6xSATA2, 1xIDE, FDD, RAID 0,1,0+1, Lan 1000 Mb/s, SB 8Ch)	1200
Відеокарта	PCIeX: ATI HD5650 SAPPHIRE 512MB/128bit/DDR2/TV/DualDVI	750
Жорсткий диск	HDD: 640 Gb 7200 Serial ATA WD 32MB	1200
Оперативна пам'ять	DIMM 2048Mb DDR3 PC3-10600 CL Transcend JetRam 1333Mhz -2 модуля	900
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	416
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA 489, PSU 350W(FSP Brand: ATX-350PNE 12cm), black, (front bezel – black+light silver body material – 0.6mm), 80mm fan (rear 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	911

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 170/160, D-SUB, Wide)	3400
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10747	8597,6	94573,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	113125,1

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	113125	-	-
Всього по групі	113125	40	45250
Нематеріальні активи			
4. Нематеріальні активи	15000	10	1500
Група 5, 6			
5. Вимірювальні пристрої	5190	25	1297,5
6. Транспортні засоби	0	20	0,0
7. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
Разом	$K_p = 1569315$		$A_p = 125447,5$

Примітка: вартість транспортного засобу приймаємо рівним нулю.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 213 \cdot 209 / 150 = 297 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 297 \cdot 10 \cdot 0,01 = 29,7 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 37\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 37(297 + 29,7) = 121 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 297 \cdot 15 \cdot 0,01 = 44,5 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджей, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Згідно виданих викладачем норм приймаємо одну пачку паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 70$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 70 \cdot 3 = 210 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуваних пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 4 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 6 грн./шт.

$$Z_{M2} = 15 \cdot 4 + 6 = 66 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (210 + 66 + 1702) / 150 = 13 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 297 \cdot 15 \cdot 0,01 = 44,5 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 150$ прим.):

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 125448 \cdot 3 / (15 \cdot 12) = 209 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	297
2. Додаткова зарплата виконавців	Z_d	29,7
3. Відрахування на соціальні потреби	C_{oc}	121
4. Загальногосподарські витрати	G_{ocn}	44,5
5. Витрати на матеріали	Z_M	13
6. Освоєння нових операційних систем, мов програмування	O_n	44,5
7. Амортизація основних фондів	A_m	209
8. Повна собівартість програмного забезпечення	C_n	758,7
9. Плановий прибуток	P_p	455,2
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1213,9
11. Податок на додану вартість $ПДВ = 0.01 \cdot N_{об} \cdot C_n$	$ПДВ$	242,8
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1456,7

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 297 + 29,7 + 121 + 44,5 + 13 + 44,5 + 209 = 758,7 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 60%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 60 \cdot 758,7 = 455,2 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Вартість програмного рішення взятого для порівняння складе 15000 грн. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	1500	1457
Всього капітальних витрат	1500	1457

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	3617	2713
2. Витрати на електроенергію	$Z_{ел}$	0	0
3. Витрати на амортизацію	$Z_{ам}$	375	375
Всього витрат за рік	I	3992	3088

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування сервера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин для обслуговування системи зменшилось з 400 годин на рік до 300 годин на рік, тому витрати складуть:

$$Z_{p \text{ баз}} = 400 \cdot 6 \cdot 1,1 \cdot 1,37 = 3617 \text{ грн.}$$

до:

$$Z_{p \text{ нов}} = 300 \cdot 6 \cdot 1,1 \cdot 1,37 = 2713 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

Визначити різницю споживання електроенергії при впровадженні систем не має можливості, тому витрати на електроенергію в розрахунку приймаємо рівними нулю.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	1500	1457	375	364,25
Всього відрахувань	-	1500	1457	375	364,25

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum E_p K_p, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.

$$E_e = (1213,9 - 758,7) \cdot 150 - (0,05 \cdot 1408000 + 0,4 \cdot 113125 + 0,25 \cdot 33190 + 0,1 \cdot 15000) \frac{3}{12} = 36918 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{161315}{(1213,9 - 758,7) \cdot 150 \cdot 12 / 3} = 0,7 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	150
2. Повна собівартість розробленої програми	Грн.	758,7
3. Ціна розробленої програми	Грн.	1213,9
4. Плановий прибуток від реалізації розробленої програми	Грн.	455,2
5. Рентабельність програмної продукції	%	60
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1569315
7. Загальний прибуток від реалізації програмної продукції	Грн.	68280
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	36918
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1457
11. Величина економічного ефекту у користувача програмної продукції	Грн.	915
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\text{б}} + E_n K_{\text{б}}) - (I_{\text{н}} + E_n K_{\text{н}}), \quad (7.27)$$

де: $I_{\text{б}}$, $I_{\text{н}}$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно; $K_{\text{б}}$, $K_{\text{н}}$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (3992 + 0,25 \cdot 1500) - (3088 + 0,25 \cdot 1457) = 915 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{\Delta K}{E_{cn}}, \quad (7.28)$$

$$T_{cn} = \frac{43}{915} < 0,1 \text{ року.}$$

Як бачимо з розрахунків запропонований варіант є більш економічно доцільним ніж варіант вибраний для порівняння, який на даний момент є лідером продаж на ринку.

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Пожежна безпека для приміщень з ПЕОМ

Пожеж безпечних не буває. Якщо вони і не створюють прямої загрози життю та здоров'ю людини, то завдають збитків довкіллю, призводять до значних матеріальних втрат. Коли людина перебуває в зоні впливу пожежі, то вона може потрапити під дію наступних небезпечних та шкідливих факторів: токсичні продукти згорання; вогонь; підвищена температура середовища; дим; недостатність кисню; руйнування будівельних конструкцій; вибухи, витікання небезпечних речовин, що відбуваються внаслідок пожежі; паніка.

При роботі з комп'ютерною технікою одним із небезпечним фактором є електробезпека та пожежна безпека, тому забезпечення пожежної безпеки для приміщень з ЕОМ є одним із основних вимог щодо безпечності працівників. Тема є актуальною, розглянемо її в нашому розділі.

Забезпечення пожежної безпеки є складовою частиною виробничої або іншої діяльності посадових осіб, працівників підприємств та підприємців. Це повинно бути відображено у трудових договорах (контрактах) та статутах підприємств.

Власники підприємств та орендарі або уповноважені ними особи зобов'язані забезпечувати дотримання користувачами персональних ЕОМ, що встановлені на їхніх підприємствах, вимог пожежної безпеки, зазначених у технічній документації заводів-виробників персональних ЕОМ.

Ступінь вогнестійкості будинків приймається в залежності від їхнього призначення, категорії по вибухопожежній і пожежній небезпеці, по поверховості, площі поверху в межах пожежного відсіку.

Будинок, у якому знаходиться дане приміщення по пожежній небезпеці будівельних конструкцій відноситься до категорії Д, оскільки тут присутні

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

- електрообладнанням та освітлювальними приладами;
- періодичний контроль цілісності і надійності електроізоляції;
- наявність інструкцій з пожежної безпеки;
- навчання, атестація і переатестація персоналу з пожежної безпеки;
- наявність системи захисту від атмосферної електрики;
- періодичне зняття зарядів статичної електрики;
- заборона куріння в приміщенні.
- застосування будівельних конструкцій із ступенем вогнестійкості не нижче II, а також використання важкогорючих або негорючих матеріалів в інтер'єрі виробничого приміщення;
- наявність схеми евакуації;
- наявність пристроїв автоматичного вимкнення ПЕОМ та іншого електроустаткування на випадок пожежі;
- наявність первинних засобів пожежогасіння.

8.2 Профілактичні заходи щодо запобігання пожеж

Пожежна профілактика являє собою комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобіганні пожежі, обмеження його поширення, а також створення умов для успішного гасіння пожежі. Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів, засобів захисту та попередження пожежі.

Одне з умов забезпечення пожежної безпеки – ліквідація можливих джерел запалення.

У даному приміщенні джерелами запалення можуть бути:

- несправне електроустаткування, несправності в електропроводці, електричних розетках і вимикачах. Для виключення виникнення пожежі з цих

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

причин необхідно вчасно виявляти й усувати несправності, проводити плановий огляд і вчасно усувати всі несправності;

– несправні електроприлади. Необхідні міри для виключення пожежі містять у собі своєчасний ремонт електроприладів, якісне виправлення поломок, не використання несправних електроприладів;

– обігрівання приміщення електронагрівальними приладами з відкритими нагрівальними елементами. Відкриті нагрівальні поверхні можуть спричинити пожежу, тому що в приміщенні знаходяться паперові документи і довідкова література у виді книг, посібників, а папір – легкозаймистий предмет. З метою профілактики пожежі пропоную не використовувати відкриті обігрівальні прилади;

– коротке замикання в електропроводці. З метою зменшення імовірності виникнення пожежі внаслідок короткого замикання необхідно, щоб електропроводка була схованою.

– влучення в будинок блискавки. У літній період під час грози можливе влучення блискавки внаслідок чого можливий пожежа. Щоб уникнути цього я рекомендую установити на даху будинку блискавковідвід;

– недотримання мір пожежної безпеки і паління в приміщенні також може спричинити пожежу. Для усунення загоряння в результаті паління в приміщенні пропоную категорично заборонити паління, а дозволити тільки в строго відведеному для цього місці.

З метою запобігання пожежі пропоную проводити з працівниками, що працюють з ПЕОМ, протипожежний інструктаж, на якому ознайомити їх із правилами протипожежної безпеки, а також навчити використанню первинних засобів пожежогасіння.

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефоні пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації, приступити до ліквідації пожежі вогнегасниками. При наявності невеликого вогнища полум'я, можна скористатися підручними засобами з метою припинення доступу повітря до об'єкта загоряння.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

8.3 Інструкція з пожежної і вибухової техніки безпеки

Згідно з ГОСТ 12.1.004-91 та ГОСТ 12.1.010-76 до роботи з системою контролю і управлінням доступу допускаються особи інженерно-технічного складу, що вивчили пристрій, інструкцію з експлуатації, дану інструкцію і ті, хто склав залік з техніки безпеки і пожежної безпеки.

1. Перед початком роботи на комп'ютері необхідно:

1.1. Ознайомитися з посібником по експлуатації і розташуванням рубильників.

1.2. Для вмикання ПЕОМ необхідні кабелі електроживлення комп'ютера і зовнішні пристрої підключити до мережі з напругою 220 В і частотою 50 Гц.

1.3. Щоб уникнути ушкодження окремих вузлів ПЕОМ забороняється підключати кабелі при включених у мережу зовнішніх пристроїв.

2. У процесі експлуатації комп'ютера:

2.1. Забороняється установка додаткових модулів розширення при включеному блоці електроживлення, а також використання саморобних кабелів, блоків розширення і блоків електроживлення, необхідно зайву довжину кабелів необхідно звернути у так звану гармошку.

2.2. Встановлюються такі внутрішні змінні режими праці та відпочинку при роботі з ЕОМ:

– при 8-годинній робочій зміні для операторів із застосуванням ЕОМ слід призначити регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за візуальним дисплейним терміналом (ВДТ);

– при 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4 годин роботи через кожну годину тривалістю 15 хв.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години.

3. При закінченні роботи на ПЕОМ необхідно виключити електроживлення системного блоку і всіх зовнішніх пристроїв, що входять до складу комп'ютерної системи.

4. Обов'язки працюючих при аварійних ситуаціях:

4.1. У випадку пожежі необхідно знеструмити все устаткування, перекрити канали вентиляції і кондиціонування, ввімкнути пожежну сигналізацію і вжити заходів щодо гасіння пожежі (вогнегасник вуглекислий ОУ-5).

4.2. У випадку аварійної ситуації необхідно знеструмити все устаткування, перекрити канали вентиляції і кондиціонування, доповісти керівнику робіт про подію та, у разі потреби, приступити до ліквідації аварійної ситуації.

8.4 Розрахункова частина

Розрахувати і побудувати блискавкозахист димової труби, яка має висоту $H_{\text{тр}} = 25$ м, зовнішній діаметр її верхнього отвору $D_{\text{в}} = 1$ м, зовнішній діаметр нижнього отвору $D_{\text{н}} = 2$ м, розташованої в місцевості з середньорічною тривалістю гроз від 40 до 60 год. Питомий опір ґрунту $\rho = 0,8 \cdot 10^4$ Ом·см.

1. Визначають очікувану на рік кількість уражень блискавкою будівель, не обладнаних блискавкозахистом за формулою:

$$N = \left[(A + 6H_M)(B + 6H_M) - 7,7H_M^2 \right] \cdot n \cdot 10^{-6}, \quad (8.1)$$

де

B і A – відповідно ширина і довжина будівлі, що має в площі прямокутну форму, м;

H_M – найбільша висота будівлі (споруди), м;

n – середньорічна кількість ударів блискавки на 1 км^2 земної поверхні в

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120

місці розташування будівлі, залежить від інтенсивності грозової діяльності К (n=4), тоді

$$N = \frac{[(1 + 6 \cdot 25)(1 + 6 \cdot 25) - 7,7 \cdot 25^2] \cdot 4}{10^6} = 0,093 \text{ рази на рік,}$$

оскільки $A=B=Дв$.

2. Встановлюємо категорію блискавкозахисту і тип зони захисту.

Димова труба висотою D більше 15 м у місцевості з середньорічною тривалістю гроз понад 20 год/рік, відноситься до III категорії блискавкозахисту. Тип зони захисту – зона Б.

3. Обираємо одиночний стрижневий блискавковідвід і визначаємо його висоту за формулою :

$$H = \frac{R_x + 1,63H_x}{1,5} \quad (8.2)$$

де H_x – висоту будинку, R_x – радіус кола горизонтального перетину зони захисту на висоті H_x

$$H = \frac{1 + 1,63 \cdot 25}{1,5} = 27,8 \text{ м.},$$

оскільки $R_x = D_n/2 = 1\text{м}$, то висота блискавкоприймача H_m дорівнює:

$$H_m = H - H_{тр} = 27,8 - 25 = 2,8 \text{ м.}$$

Перетин сталевого блискавкоприймача і перетин струмовідводу приймаємо рівним 100 мм^2 .

4. Побудову зони блискавкозахисту виконуємо за такою послідовністю.

Через блискавкоприймач проводимо осьову лінію, на якій відкладаємо відрізок $OA = 0,92 H = 25,6\text{м}$. Від основи труби (точка O) відкладаємо відрізки OB і OB_1 рівні $1,5H = 39,5 \text{ м}$. Точку A з'єднуємо з точками B і B_1 . Отримуємо зону захисту димової труби у вигляді конусу з основою BB_1 і утворюючими AB і AB_1 . (Рис 8.1)

5. Заземлюючий пристрій обираємо керуючись наступним:

Для III категорії блискавкозахисту і $\rho = 0,8 \cdot 10^4 \text{ Ом} \cdot \text{см}$, припустимий

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		121

імпульсний опір R_i не більше 20 Ом, знаходимо величину опору розтікання струму промислової частоти заземлювача R_{50} для ґрунту з питомим опором $\rho = 0,8 \cdot 10^4$ Ом · см. Він дорівнює 14 Ом.

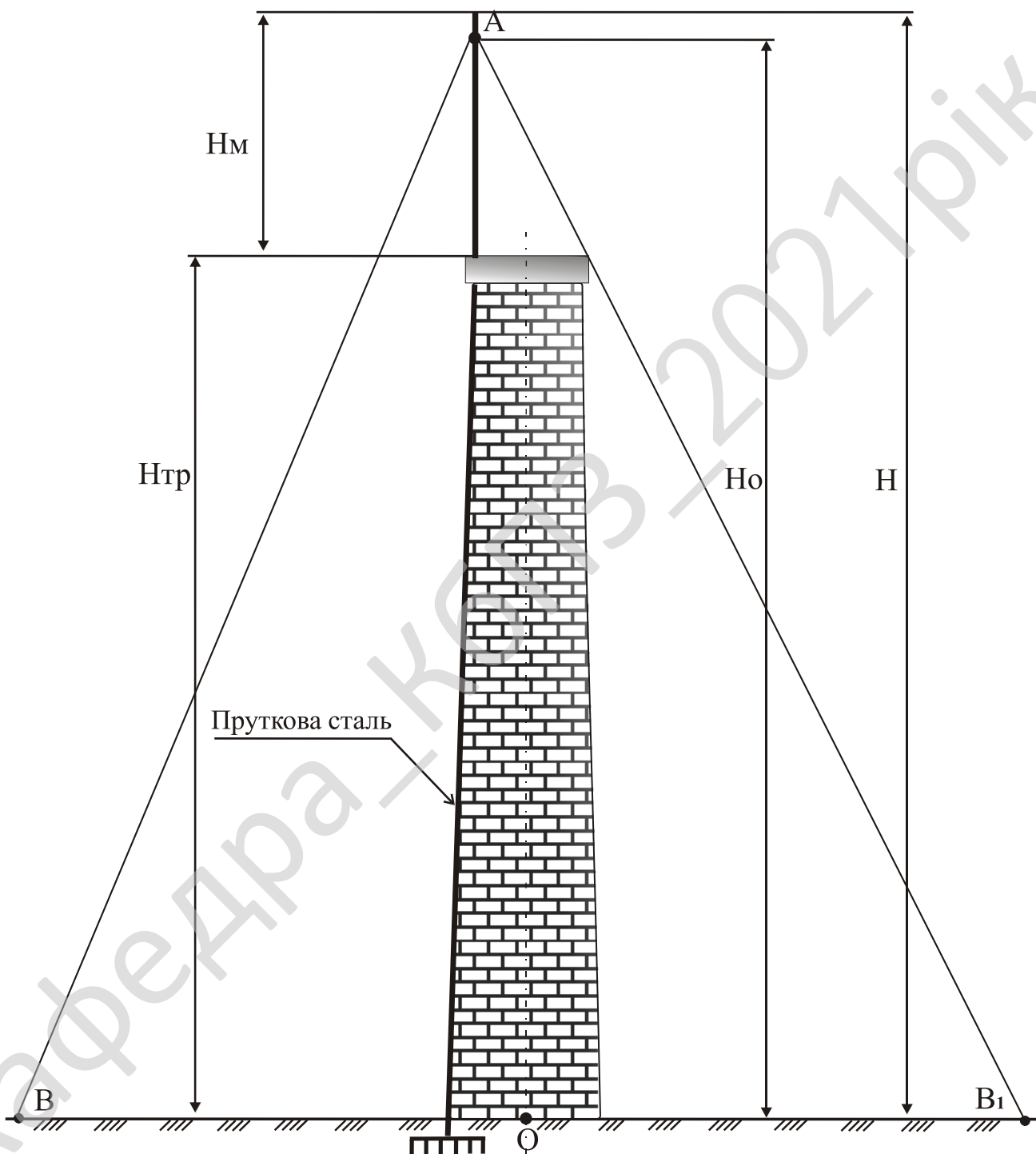


Рисунок 8.1 – Побудова зони захисту блискавковідводу

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0014.00.00.ПЗ

Арк.

122

Потім за допомогою таблиць обираємо форму заземлювача, який має опір, близький до шуканого. Таким опором володіє заземлювач з порядковим номером 5, а саме: заземлювач двохстрижневий із труби діаметром 50 мм, заглиблений на 0,8 м. Висота електродів $L = 2,5\text{м}$, відстань між ними $C = 3\text{ м}$.

8.5 Висновки до розділу

В розділі розглянуто загальні вимоги пожежної безпеки та вимоги пожежної безпеки для приміщень з ЕОМ . Запропоновані заходи щодо зменшення вірогідності виникнення пожежі, а також наведено приклад розрахунку блискавкозахисту виробничого приміщення. Розглянуті організаційні заходи щодо забезпечення пожежної безпеки – є складовою частиною виробничої або іншої діяльності посадових осіб, працівників підприємств та підприємців, що визначено законодавством про пожежну безпеку.

Всі ці питання залишаються актуальними , тому що пожежа – це явище страшне. Воно не тільки наносить величезні матеріальні збитки, а й забирає життя людей.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		123

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.
- Досліджена система забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.
- На основі отриманих результатів досліджень створена програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		124

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.3.2 Rio. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Md5.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 915 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		125

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Панасюк Б.В. Дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.
2. Назаров А.Н. Модели и методы расчета структурно-сетевых параметров АТМ сетей / Алексей Николаевич Назаров. – М.: Горячая линия – Телеком, 2002. – 256 с.
3. Одом Ш. Коммутаторы CISCO / Ш. Одом, Х. Ноттингем – М.: "Кудиц-Образ", 2003. – 528 с.
4. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 2-е изд. – СПб.: Питер, 2007. – 958 с.
5. Партыка С.А. Метод ускоренной коррекции SPT с использованием динамических алгоритмов [Электронный ресурс]. – Режим доступа до ресурсу: http://openarchive.nure.ua/bitstream/123456789/936/1/ASU_158_2012%20%2842-47%29.pdf
6. Руководство по технологиям объединенных сетей. 4-е изд. / пер.с англ. и ред. А.Н. Крикуна – М.: Изд. дом «Вильямс», 2005. – 1040 с.
7. Семенов С.Г. Математическая модель мультисервисного канала связи на основе экспоненциальной GERT-сети / С.Г. Семенов, Є.В. Мелешко, Я.В. Ілюшко // Системи озброєння і військова техніка. – Х.:ХУ ПС. – 2011. –Вип. 3(27). – С. 64-67.
8. Семенов С.Г. Математична модель системи криптографічного захисту електронних повідомлень на основі GERT-мережі / С.Г. Семенов,

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		126

О.О. Сур // Системы управління, навігації та зв'язку. – К.: ЦНДІ навігації і управління. – 2012. – Том 1. Вип. 1(21). – С. 131-137

9. Семенов С.Г. Исследования вероятностно-временных характеристик мультисервисного канала связи с использованием математического аппарата GERT-сети / С.Г. Семенов, В.В. Босько, І.А. Березюк // Системы обробки інформації. – Х.: ХУ ПС, 2012. – Т. 1., Вип. 3(101). – С. 139-142.

10. Семенов С.Г. Моделирование защищенного канала связи с использованием экспоненциальной GERT-сети / С.Г. Семенов, А.А. Можаяев // Информатика, математическое моделирование, экономика. – Смоленськ.: Смоленский филиал АНО ВПО ЦС РФ "Российский университет кооперации". – 2012. – Том.1. – С. 152-160.

11. Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов // Вісник Національного технічного університету «Харківський політехнічний інститут». – Х.:НТУ «ХПІ». – 2012. –№62 (968). – С 173-181.

12. Семенов С.Г. Защита данных в компьютеризированных управляющих системах / С.Г. Семенов, В.В. Давыдов, С.Ю. Гавриленко. – LAP Lambert Academic Publishing GmbH & Co. KG (Саарбрюккен, Германия), 2014. – 236 с.

13. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системы обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

14. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

15. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		127

Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

16. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

17. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

18. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

19. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

20. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		128

21. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

22. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. – Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

23. Смирнов С. А. Комплекс GERT-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

24. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

25. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

26. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

27. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		129

28. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

29. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных вирусов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

30. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

31. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

32. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

33. Смирнов С. А. Технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		130

технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

34. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

35. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

36. Смирнов С. А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

37. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

38. Смирнов С. А. Разработка комплекса GERT-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії (IT & I): зб. тез II міжнар. наук. -практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		131

39. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

40. Смирнов С. А. GERT-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

41. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць II Мі жнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

42. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

43. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформатика та системні науки (ІСН-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

44. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы /

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		132

А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблемы кибербезопасности информационно-телекоммуникационных систем: сб. тез наук. -практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

45. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук. -практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

46. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня – 1 квітня 2016 р. – Х.: НТУ «ХП», 2016. – С. 14.

47. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). –Кіровоград: КНТУ, 2016. – С. 182-186.

48. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

49. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна та економічна безпека (INFECO-

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		133

2016): зб. тез III міжнар. наук.-практ. конф., м. Харків, 28-30 кві. 2016 р. – Х.: ХННІ ДВНЗ «УБС», 2016. – С. 178-182.

50. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Сборник тезисов XII международной конференции «Стратегия качества в промышленности и образовании» (г. Варна, Болгария, 30 мая – 02 июня 2016 г.). – Варна: ТУВ, 2016. – С. 581-585.

51. Смирнов С. А. Оценка эффективности метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. С. Коваленко // РадіоЕлектроніка та ІнфоКомунікації: зб. тез першої наук. – техн. конф., м. Київ, 11-16 вересня 2016 р. – К.: НТУУ «КПІ», 2016. – С. 17.

52. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

53. Столлингс В. Современные компьютерные сети / Вильям Столлингс. – СПб.: Питер, 2003. – 778 с.

54. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.

55. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.

56. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1103 с.

57. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.

					ВКРМ-123.21.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		134

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Економічні вимоги.....	5
8	Вимоги щодо охорони праці.....	5
9	Перелік документів, що розробляються.....	6
10	Етапи розробки.....	6
11	Порядок контролю та приймання.....	6

					ВКРМ-123.21.0014.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Панасюк Б.В.				Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.			М			
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М-1,4		
Затв.	Смірнов О.А.						
Дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.21.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи забезпечення обміну інформацією у корпоративній мережі на базі технології DMVPN;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.21.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.3.2 Rio.

					ВКРМ-123.21.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута пожежна безпека для приміщень з ПЕОМ.

					ВКРМ-123.21.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 134 аркуші.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2021 р.

					ВКРМ-123.21.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Коваленко А.С.

*Дослідження та програмна реалізація
системи забезпечення обміну інформацією у корпоративній мережі на базі
технології DMVPN*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 54

Літера: РП

Кропивницький – 2021 року

DMVPN_WinMain.pas - основна програма

```

unit DMVPN_WinMain;

interface

uses
  Windows, Messages, CommCtrl, DMVPN_CommCtrl, DMVPN_SysUtils, DMVPN_FileInfo,
  DMVPN_Resources,
  DMVPN_WelcWind, DMVPN_SetWind, DMVPN_FinsWind;

function WinMain(hInstance: HINST; hPrevInstance: HINST; lpCmdLine: LPSTR;
nCmdShow: Integer): Integer; stdcall;

implementation

function WinMain(hInstance: HINST; hPrevInstance: HINST; lpCmdLine: LPSTR;
nCmdShow: Integer): Integer; stdcall;
var
  hMutex : THandle;
  pszText: WideString;
  iccex  : TInitCommonControlsEx;
  psh    : TPropSheetHeader;
begin
  // витягаємо інформацію з ресурсу версії й заповнюємо їй підготовлену
  // структуру, що у подальшому будемо використовувати для читання/запису
  // налаштувань програми й виводу тексту в заголовку повідомлень.

  ZeroMemory(@exeInfo, SizeOf(TStringFileInfo));
  GetFileInfo(AnsiStringToWide(ParamStr(0), CP_ACP), exeInfo);

  // створюємо Mutex для перевірки запуску копій додатка.

  hMutex := CreateMutex(nil, FALSE, MAKEINTRESOURCEW(exeInfo.pszProductName));
  if (GetLastError = ERROR_ALREADY_EXISTS) then
  begin
    pszText := LoadStrInst(hInstance, RC_STRING_COPYRUN);
    MessageBox(
      GetActiveWindow,
      @pszText[1],
      MAKEINTRESOURCEW(exeInfo.pszProductName),
      MB_OK or MB_ICONEXCLAMATION or MB_SYSTEMMODAL
    );
    Halt;
  end;

  // ініціалізуємо бібліотеку стандартних органів управління.
  iccex.dwSize := SizeOf(TInitCommonControlsEx);
  iccex.dwICC := ICC_ANIMATE_CLASS or ICC_PROGRESS_CLASS or ICC_TAB_CLASSES or
  ICC_STANDARMDVPN_CLASSES or ICC_WIN95_CLASSES;
  InitCommonControlsEx(iccex);

  //

  pszText := Format(LoadStrInst(hInstance, RC_STRING_CWINDOW),
    [exeInfo.pszProductName, exeInfo.pszFileVersion]);

  // створюємо й відображаємо сторінки майстра.

  ZeroMemory(@psp, SizeOf(TPropSheetPage));

  psp.dwSize := SizeOf(TPropSheetPage);
  psp.dwFlags := PSP_USETITLE or PSP_HIDEHEADER;
  psp.pszTitle := @pszText[1];

```

```

    psp.pfnDlgProc      := @WelcDlgProc;
    psp.pszTemplate    := MAKEINTRESOURCEW(RC_DIALOG_WELCOME);
    ahpsp[0]           := CreatePropertySheetPage( psp );

    ZeroMemory(@psp, SizeOf(TPropSheetPage));

    psp.dwSize         := SizeOf(TPropSheetPage);
    psp.dwFlags        := PSP_USETITLE or PSP_USEHEADERTITLE or
PSP_USEHEADERSUBTITLE;
    psp.pszTitle       := @pszText[1];
    psp.pszHeaderTitle := MAKEINTRESOURCEW(LoadStrInst(hInstance,
RC_STRING_THEADER));
    psp.pszHeaderSubTitle := MAKEINTRESOURCEW(LoadStrInst(hInstance,
RC_STRING_SHEADER));
    psp.pszTemplate    := MAKEINTRESOURCEW(RC_DIALOG_SETTINGS);
    psp.pfnDlgProc     := @SettDlgProc;
    ahpsp[1]           := CreatePropertySheetPage( psp );

    ZeroMemory(@psp, SizeOf(TPropSheetPage));

    psp.dwSize         := SizeOf(TPropSheetPage);
    psp.dwFlags        := PSP_USETITLE or PSP_HIDEHEADER;
    psp.pszTitle       := @pszText[1];
    psp.pszTemplate    := MAKEINTRESOURCEW(RC_DIALOG_FINISH);
    psp.pfnDlgProc     := @FinsDlgProc;
    ahpsp[2]           := CreatePropertySheetPage( psp );

    ZeroMemory(@psh, SizeOf(TPropSheetHeader));

    psh.dwSize         := SizeOf(TPropSheetHeader);
    psh.hInstance      := hInstance;
    psh.hwndParent     := 0;
    psh.phpage         := @ahpsp[0];
    psh.nStartPage     := 0;
    psh.nPages         := Length(ahpsp);
    psh.pszbmWatermark := MAKEINTRESOURCEW(RC_BITMAP_WATERMARK);
    psh.pszbmHeader    := MAKEINTRESOURCEW(RC_BITMAP_HEADER);
    psh.dwFlags        := PSH_WIZARD97 or PSH_WATERMARK or PSH_HEADER or
PSH_USEICONID;
    psh.pszIcon        := MAKEINTRESOURCEW(RC_ICONEX_CAPTION);

    PropertySheet( psh );

    // видаляємо іменований об'єкт.

    if (hMutex <> 0) then
    begin
        ReleaseMutex(hMutex);
        CloseHandle(hMutex);
    end;

    //
    Result := 0;

end;

end.

```

DMVPN_WinSvc.pas - cepицн

```

unit DMVPN_WinSvc;

interface

uses
  Windows;

type
  LPSERVICE_STATUS = ^SERVICE_STATUS;
  _SERVICE_STATUS = record
    dwServiceType: DWORD;
    dwCurrentState: DWORD;
    dwControlsAccepted: DWORD;
    dwWin32ExitCode: DWORD;
    dwServiceSpecificExitCode: DWORD;
    dwCheckPoint: DWORD;
    dwWaitHint: DWORD;
  end;
  SERVICE_STATUS = _SERVICE_STATUS;
  TServiceStatus = SERVICE_STATUS;
  PServiceStatus = LPSERVICE_STATUS;
  LPSERVICE_STATUS_PROCESS = ^SERVICE_STATUS_PROCESS;
  _SERVICE_STATUS_PROCESS = record
    dwServiceType: DWORD;
    dwCurrentState: DWORD;
    dwControlsAccepted: DWORD;
    dwWin32ExitCode: DWORD;
    dwServiceSpecificExitCode: DWORD;
    dwCheckPoint: DWORD;
    dwWaitHint: DWORD;
    dwProcessId: DWORD;
    dwServiceFlags: DWORD;
  end;
  SERVICE_STATUS_PROCESS = _SERVICE_STATUS_PROCESS;
  TServiceStatusProcess = SERVICE_STATUS_PROCESS;
  PServiceStatusProcess = LPSERVICE_STATUS_PROCESS;

  //
  // Структура перерахунку статусу послуги
  //
  LPENUM_SERVICE_STATUSA = ^ENUM_SERVICE_STATUSA;
  {$EXTERNALSYM LPENUM_SERVICE_STATUSA}
  _ENUM_SERVICE_STATUSA = record
    lpServiceName: LPSTR;
    lpDisplayName: LPSTR;
    ServiceStatus: SERVICE_STATUS;
  end;
  {$EXTERNALSYM _ENUM_SERVICE_STATUSA}
  ENUM_SERVICE_STATUSA = _ENUM_SERVICE_STATUSA;
  {$EXTERNALSYM ENUM_SERVICE_STATUSA}
  TEnumServiceStatus = ENUM_SERVICE_STATUSA;
  PEnumServiceStatus = LPENUM_SERVICE_STATUSA;
  LPENUM_SERVICE_STATUSW = ^ENUM_SERVICE_STATUSW;
  {$EXTERNALSYM LPENUM_SERVICE_STATUSW}
  _ENUM_SERVICE_STATUSW = record
    lpServiceName: LPWSTR;
    lpDisplayName: LPWSTR;
    ServiceStatus: SERVICE_STATUS;
  end;
  {$EXTERNALSYM _ENUM_SERVICE_STATUSW}
  ENUM_SERVICE_STATUSW = _ENUM_SERVICE_STATUSW;
  {$EXTERNALSYM ENUM_SERVICE_STATUSW}
  TEnumServiceStatus = ENUM_SERVICE_STATUSW;
  PEnumServiceStatus = LPENUM_SERVICE_STATUSW;

```

```

PEnumServiceStatus = PEnumServiceStatus;

_SC_STATUS_TYPE = (SC_STATUS_PROCESS_INFO);
SC_STATUS_TYPE = _SC_STATUS_TYPE;

//
// Типи заголовків
//

{$EXTERNALSYM SC_HANDLE}
SC_HANDLE = THandle;
{$EXTERNALSYM LPSC_HANDLE}
LPSC_HANDLE = ^SC_HANDLE;

const //
// Стан сервісу - для перерахуємих послуг (бітова маска)
//
{$EXTERNALSYM SERVICE_ACTIVE}
SERVICE_ACTIVE = $00000001;
{$EXTERNALSYM SERVICE_INACTIVE}
SERVICE_INACTIVE = $00000002;
{$EXTERNALSYM SERVICE_STATE_ALL}
SERVICE_STATE_ALL = (SERVICE_ACTIVE or
SERVICE_INACTIVE);

//
// Менеджер сервісу управління об'єкту типу специфічного доступу
//
{$EXTERNALSYM SC_MANAGER_CONNECT}
SC_MANAGER_CONNECT = $0001;
{$EXTERNALSYM SC_MANAGER_CREATE_SERVICE}
SC_MANAGER_CREATE_SERVICE = $0002;
{$EXTERNALSYM SC_MANAGER_ENUMERATE_SERVICE}
SC_MANAGER_ENUMERATE_SERVICE = $0004;
{$EXTERNALSYM SC_MANAGER_LOCK}
SC_MANAGER_LOCK = $0008;
{$EXTERNALSYM SC_MANAGER_QUERY_LOCK_STATUS}
SC_MANAGER_QUERY_LOCK_STATUS = $0010;
{$EXTERNALSYM SC_MANAGER_MODIFY_BOOT_CONFIG}
SC_MANAGER_MODIFY_BOOT_CONFIG = $0020;

{$EXTERNALSYM SC_MANAGER_ALL_ACCESS}
SC_MANAGER_ALL_ACCESS = (STANDARDMVPN_RIGHTS_REQUIRED or
SC_MANAGER_CONNECT or
SC_MANAGER_CREATE_SERVICE or
SC_MANAGER_ENUMERATE_SERVICE or
SC_MANAGER_LOCK or
SC_MANAGER_QUERY_LOCK_STATUS or
SC_MANAGER_MODIFY_BOOT_CONFIG);

//
// Сервіс об'єкту типу специфічного доступу
//
SERVICE_STOP = $0020;
SERVICE_QUERY_STATUS = $0004;
SERVICE_ENUMERATE_DEPENDENTS = $0008;
//
// Стан сервісу - для поточного стану
//
{$EXTERNALSYM SERVICE_STOPPED}
SERVICE_STOPPED = $00000001;
{$EXTERNALSYM SERVICE_START_PENDING}
SERVICE_START_PENDING = $00000002;
{$EXTERNALSYM SERVICE_STOP_PENDING}
SERVICE_STOP_PENDING = $00000003;
{$EXTERNALSYM SERVICE_RUNNING}
SERVICE_RUNNING = $00000004;
{$EXTERNALSYM SERVICE_CONTINUE_PENDING}
SERVICE_CONTINUE_PENDING = $00000005;

```

```

SERVICE_CONTINUE_PENDING      = $00000005;
{$EXTERNALSYM SERVICE_PAUSE_PENDING}
SERVICE_PAUSE_PENDING        = $00000006;
{$EXTERNALSYM SERVICE_PAUSED}
SERVICE_PAUSED                = $00000007;
//
// Управління
//
{$EXTERNALSYM SERVICE_CONTROL_STOP}
SERVICE_CONTROL_STOP         = $00000001;
{$EXTERNALSYM SERVICE_CONTROL_PAUSE}
SERVICE_CONTROL_PAUSE        = $00000002;
{$EXTERNALSYM SERVICE_CONTROL_CONTINUE}
SERVICE_CONTROL_CONTINUE      = $00000003;
{$EXTERNALSYM SERVICE_CONTROL_INTERROGATE}
SERVICE_CONTROL_INTERROGATE  = $00000004;
{$EXTERNALSYM SERVICE_CONTROL_SHUTDOWN}
SERVICE_CONTROL_SHUTDOWN     = $00000005;

function OpenSCManager(lpMachineName: LPCWSTR; lpDatabaseName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;
function OpenSCManager(lpMachineName: LPCWSTR; lpDatabaseName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;
function OpenSCManager(lpMachineName: LPCWSTR; lpDatabaseName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;

function OpenService(hSCManager: SC_HANDLE; lpServiceName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;
function OpenService(hSCManager: SC_HANDLE; lpServiceName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;
function OpenService(hSCManager: SC_HANDLE; lpServiceName: LPCWSTR;
dwDesiredAccess: DWORD): SC_HANDLE; stdcall;

function CloseServiceHandle(hSCObject: SC_HANDLE): BOOL; stdcall;

function QueryServiceStatusEx(hService: SC_HANDLE; InfoLevel: SC_STATUS_TYPE;
lpBuffer: PByte; cbBufSize: DWORD; var pcbBytesNeeded: DWORD): BOOL; stdcall;

function ControlService(hService: SC_HANDLE; dwControl: DWORD; var
lpServiceStatus: TServiceStatusProcess): BOOL; stdcall;

function EnumDependentServices(hService: SC_HANDLE; dwServiceState: DWORD;
lpServices: LPENUM_SERVICE_STATUSA; cbBufSize: DWORD; var pcbBytesNeeded,
lpServicesReturned: DWORD): BOOL; stdcall;
function EnumDependentServices(hService: SC_HANDLE; dwServiceState: DWORD;
lpServices: LPENUM_SERVICE_STATUSW; cbBufSize: DWORD; var pcbBytesNeeded,
lpServicesReturned: DWORD): BOOL; stdcall;
function EnumDependentServices(hService: SC_HANDLE; dwServiceState: DWORD;
lpServices: LPENUM_SERVICE_STATUSA; cbBufSize: DWORD; var pcbBytesNeeded,
lpServicesReturned: DWORD): BOOL; stdcall;

implementation
function OpenSCManager; external advapi32 name 'OpenSCManager';
function OpenSCManager; external advapi32 name 'OpenSCManager';
function OpenSCManager; external advapi32 name 'OpenSCManager';
function OpenService; external advapi32 name 'OpenService';
function OpenService; external advapi32 name 'OpenService';
function OpenService; external advapi32 name 'OpenService';
function CloseServiceHandle; external advapi32 name 'CloseServiceHandle';
function QueryServiceStatusEx; external advapi32 name 'QueryServiceStatusEx';
function ControlService; external advapi32 name 'ControlService';
function EnumDependentServices; external advapi32 name 'EnumDependentServices';
function EnumDependentServices; external advapi32 name 'EnumDependentServices';
function EnumDependentServices; external advapi32 name 'EnumDependentServices';
end.

```

DMVPN_SysUtils.pas - утилита

```

unit DMVPN_SysUtils;

interface

uses
  Windows, Messages;

function LoadStrInst(hInst: HMODULE; I: Integer): WideString;
function Format(szString: WideString; const Params: Array of const): WideString;
function WideStringToAnsi(pszText: WideString; CodePage: WORD): AnsiString;
function AnsiStringToWide(pszText: AnsiString; CodePage: WORD): WideString;
function ExtractFilePath(pszText: WideString): WideString;
function ExcludeTrailingPathDelimiter(szString: WideString): WideString;
function SetCenterDialogPos(hDialog, hParent: HWND; IsParent: Boolean): Boolean;
function GetWindowFontSize(hWnd: HWND; pSize: Integer): Integer;
function GetWindowBoldFont(hWnd: THandle; fntHeight: Integer): HFONT;

implementation

//

function LoadStrInst(hInst: HMODULE; I: Integer): WideString;
var
  lpBuffer: Array [0..MAX_PATH-1] of WideChar;
begin
  LoadString(hInst, I, lpBuffer, Length(lpBuffer));
  Result := lpBuffer;
end;

//

function Format(szString: WideString; const Params: Array of const): WideString;
var
  lpChar: Array [0..1023] of WideChar;
  lpWord: Array [0..15] of LongWord;
  nIndex: Integer;
begin
  for nIndex := High(Params) downto 0 do
    lpWord[nIndex] := Params[nIndex].VInteger;
  wvsprintf(@lpChar, @szString[1], @lpWord);
  Result := lpChar;
end;

//

function WideStringToAnsi(pszText: WideString; CodePage: WORD): AnsiString;
var
  dwBytes: Integer;
  dwFlags: DWORD;
begin
  if (pszText <> '') then
    begin
      dwFlags := WC_COMPOSITECHECK or WC_DISCARDNS or WC_SEPCHARS or
        WC_DEFAULTCHAR;
      dwBytes := WideCharToMultiByte(CodePage, dwFlags, @pszText[1], -1, nil, 0,
        nil, nil);
      SetLength(Result, dwBytes - 1);
      if (dwBytes > 1) then
        WideCharToMultiByte(CodePage, dwFlags, @pszText[1], -1, @Result[1],
          dwBytes - 1, nil, nil);
    end
  else
    Result := '';
  end;
end;

//

```

```

function AnsiStringToWide(pszText: AnsiString; CodePage: WORD): WideString;
var
  dwBytes: Integer;
begin
  if (pszText <> '') then
    begin
      dwBytes := MultiByteToWideChar(CodePage, MB_PRECOMPOSED, @pszText[1], -1,
        nil, 0);
      SetLength(Result, dwBytes - 1);
      if (dwBytes > 1) then
        MultiByteToWideChar(CodePage, MB_PRECOMPOSED, @pszText[1], -1,
@Result[1],
        dwBytes - 1);
      end
    else
      Result := '';
end;

//

function ExtractFilePath(pszText: WideString): WideString;
var
  L: Integer;
begin
  Result := '';
  L := Length(pszText);
  while (L > 0) do
    begin
      if (pszText[L] = ':') or (pszText[L] = '\\') then
        begin
          Result := Copy(pszText, 1, L);
          Break;
        end;
      Dec(L);
    end;
end;

//

function ExcludeTrailingPathDelimiter(szString: WideString): WideString;
var
  I: Integer;
begin
  Result := szString;
  I := Length(Result);
  while (I > 0) and (Result[I] = '\\') do
    Dec(I);
  SetLength(Result, I);
end;

//

function SetCenterDialogPos(hDialog, hParent: HWND; IsParent: Boolean): Boolean;
var
  wRect : TRect;
  pRect : TRect;
  wArea : TRect;
  xLeft : Integer;
  yTop : Integer;
  iWidth : Integer;
  iHeight: Integer;
  dwFlags: DWORD;
begin
  case IsParent of
    FALSE:
      begin
        GetWindowRect(hDialog, wRect);
        iWidth := wRect.Right - wRect.Left;

```

```

    iHeight := wRect.Bottom - wRect.Top;
    xLeft := (GetSystemMetrics(SM_CXSCREEN) - iWidth) div 2;
    yTop := (GetSystemMetrics(SM_CYSCREEN) - iHeight) div 2;
end;
TRUE:
begin
  GetWindowRect(hDialog, wRect);
  GetWindowRect(hParent, pRect);
  iWidth := wRect.Right - wRect.Left;
  iHeight := wRect.Bottom - wRect.Top;
  SystemParametersInfo(SPI_GETWORKAREA, 0, @wArea, 0);
  xLeft := pRect.Left + ((pRect.Right - pRect.Left - iWidth) div 2);
  if (xLeft < 0) then
    xLeft := 0
  else
    if ((xLeft + iWidth) > (wArea.Right - wArea.Left)) then
      xLeft := wArea.Right - wArea.Left - iWidth;
    yTop := pRect.Top + ((pRect.Bottom - pRect.Top - iHeight) div 2);
    if (yTop < 0) then
      yTop := 0
    else
      if ((yTop + iHeight) > (wArea.Bottom - wArea.Top)) then
        yTop := wArea.Bottom - wArea.Top - iHeight;
      end;
    end;
  dwFlags := SWP_NOACTIVATE or SWP_NOSIZE or SWP_NOZORDER;
  Result := SetWindowPos(hDialog, 0, xLeft, yTop, 0, 0, dwFlags);
end;

//

function GetWindowFontSize(hWnd: HWND; pSize: Integer): Integer;
var
  dc: HDC;
begin
  dc := GetDC(hWnd);
  Result := -MulDiv(pSize, GetDeviceCaps(dc, LOGPIXELSY), 72);
  ReleaseDC(hWnd, dc);
end;

//

function GetWindowBoldFont(hWnd: THandle; fntHeight: Integer): HFONT;
var
  lf : TLogFont;
  dwRes: Integer;
  hfnt : HFONT;
begin
  hfnt := HFONT(SendMessage(hWnd, WM_GETFONT, 0, 0));
  ZeroMemory(@lf, SizeOf(TLogFont));
  if (hfnt <> 0) then
    dwRes := GetObject(hfnt, SizeOf(TLogFont), @lf);
    if (dwRes <> 0) then
      begin
        lf.lfHeight := fntHeight;
        lf.lfWeight := FW_BOLD;
        hfnt := CreateFontIndirect(lf);
      end;
    Result := hfnt;
  end;
end;

end.

```

DMVPN_StatAnim.pas - анімація при підключенні DMVPN

```

unit DMVPN_StatAnim;

interface

uses
  Windows, Messages, CommCtrl, DMVPN_Windows;

procedure CreateAnimateStatic(hWnd: HWND);
procedure RemoveAnimateStatic(hWnd: HWND);

const
  SS_SETIMAGELIST      = WM_USER + 101;
  SS_SETELAPSEDTIME   = WM_USER + 102;
  SS_GETIMAGELIST     = WM_USER + 111;
  SS_GETELAPSEDTIME   = WM_USER + 112;

implementation

const
  IDC_ANIMATE_TIMER = 101;

type
  TStatWndProc = function(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT; stdcall;

  P_STAT_PRO = ^T_STAT_PRO;
  T_STAT_PRO = packed record
    StatProc   : TStatWndProc;
    rcClient   : TRect;
    //
    hdcMem     : HDC;
    hbmMem     : HBITMAP;
    hbmOld     : HBITMAP;
    //
    hIm1       : HIMAGELIST;
    //
    imgSize    : Integer;
    imgCount   : Integer;
    imgCurrent: Integer;
    //
    dwElapse  : Integer;
  end;

var
  psp: P_STAT_PRO;

//
function StatWndProc_OnWmSize(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT;
var
  hdcIn: HDC;
begin
  //

  GetClientRect(hWnd, psp.rcClient);

  //

  if (psp.hdcMem <> 0) then
  begin
    SelectObject(psp.hdcMem, psp.hbmOld);
    DeleteObject(psp.hbmMem);
    DeleteDC(psp.hdcMem);
  end;
end;

```

```

//

hdcIn := GetDC(hWnd);
psp.hdcMem := CreateCompatibleDC(hdcIn);
psp.hbmMem := CreateCompatibleBitmap(
    hdcIn,
    psp.rcClient.Right - psp.rcClient.Left,
    psp.rcClient.Bottom - psp.rcClient.Top
);
psp.hbmOld := SelectObject(psp.hdcMem, psp.hbmMem);
ReleaseDC(hWnd, hdcIn);

//

Result := CallWindowProc(@psp.StatProc, hWnd, uMsg, wParam, lParam);

//

RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);

end;

//

function StatWndProc_OnWmPaint(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT; stdcall;

var
    hdcIn: HDC;
    ps    : TPaintStruct;
begin
    //

    if (wParam = 0) then
        hdcIn := BeginPaint(hWnd, ps)
    else
        hdcIn := wParam;

    //

    CallWindowProc(@psp.StatProc, hWnd, WM_PRINTCLIENT, psp.hdcMem,
PRDMVPN_CLIENT);

    {
    CallWindowProc(@psp.StatProc, hWnd, WM_ERASEBKGD, psp.hdcMem, 0);
    }

    if (psp.himl <> 0) then
        ImageList_DrawEx(
            psp.himl,
            psp.imgCurrent - 1,
            psp.hdcMem,
            psp.rcClient.Left + ((psp.rcClient.Right - psp.rcClient.Left) div 2) -
(psp.imgSize div 2),
            psp.rcClient.Top + ((psp.rcClient.Bottom - psp.rcClient.Top) div 2) -
(psp.imgSize div 2),
            psp.imgSize,
            psp.imgSize,
            CLR_DEFAULT,
            CLR_DEFAULT,
            ILDMVPN_NORMAL or ILDMVPN_TRANSPARENT
        );

    BitBlt(
        hdcIn,
        0,
        0,

```

```

    psp.rcClient.Right - psp.rcClient.Left,
    psp.rcClient.Bottom - psp.rcClient.Top,
    psp.hdcMem,
    0,
    0,
    SRCCOPY
);

//

if (wParam = 0) then
    EndPaint(hWnd, ps);

//

Result := 0;

end;

//

function StatWndProc_OnWmTimer(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT;
begin
    //

    Inc(psp.imgCurrent);
    if (psp.imgCurrent > psp.imgCount) then
        psp.imgCurrent := 1;

    //

    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);

    //

    Result := 0;

end;

//

function StatWndProc_OnWmEraseBkgnd(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //

    Result := 1;

end;

//

function StatWndProc_OnSetImageList(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //

    KillTimer(hWnd, IDC_ANIMATETIMER);

    //

    psp.himl := HIMAGELIST(wParam);

    //

```

```

if (psp.himl <> 0) then
  begin
    ImageList_GetIconSize(psp.himl, psp.imgSize, psp.imgSize);
    psp.imgCount := ImageList_GetImageCount(psp.himl);
    SetTimer(hWnd, IDC_ANIMATETIMER, psp.dwElapse, nil);

    end;

  //

  Result := 0;

end;

//

function StatWndProc_OnSetElapsedTime(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  //

  KillTimer(hWnd, IDC_ANIMATETIMER);

  //

  psp.dwElapse := wParam;

  //

  SetTimer(hWnd, IDC_ANIMATETIMER, psp.dwElapse, nil);

  //

  Result := 0;

end;

//

function StatWndProc_OnGetImageList(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  //

  Result := LRESULT(psp.himl);

end;

//

function StatWndProc_OnGetElapsedTime(psp: P_STAT_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  //

  Result := LRESULT(psp.dwElapse);

end;

//

function StatWndProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
LRESULT; stdcall;

```

```

begin

    psp := P_STAT_PRO(GetWindowLong(hWnd, GWL_USERDATA));

    if (psp = nil) then
        begin
            Result := DefWindowProc(hWnd, uMsg, wParam, lParam);
            Exit;
        end;

    case uMsg of

        //

        WM_DESTROY:
            begin
                RemoveAnimateStatic(hWnd);
            end;

        //

        WM_SIZE:
            begin
                Result := StatWndProc_OnWmSize(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        WM_PRINTCLIENT,
        WM_PAINT,
        WM_UPDATEUISTATE: // перемальовування вікна без виклику WM_PAINT.
            begin
                Result := StatWndProc_OnWmPaint(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        WM_TIMER:
            begin
                Result := StatWndProc_OnWmTimer(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        WM_ERASEBKGD:
            begin
                Result := StatWndProc_OnWmEraseBkgnd(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        SS_SETIMAGELIST:
            begin
                Result := StatWndProc_OnSetImageList(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        SS_SETELAPSEDTIME:
            begin
                Result := StatWndProc_OnSetElapsedTime(psp, hWnd, uMsg, wParam, lParam);
            end;

        //

        SS_GETIMAGELIST:
            begin
                Result := StatWndProc_OnGetImageList(psp, hWnd, uMsg, wParam, lParam);
            end;
    end;
end;

```

```

//

SS_GETELAPSEDTIME:
begin
    Result := StatWndProc_OnGetElapsedTime (psp, hWnd, uMsg, wParam, lParam);
end;

else
    Result := CallWindowProc (@psp.StatProc, hWnd, uMsg, wParam, lParam);
end;

end;

//

procedure CreateAnimateStatic (hWnd: HWND);
begin

    RemoveAnimateStatic (hWnd);

    psp := P_STAT_PRO (HeapAlloc (GetProcessHeap, HEAP_ZERO_MEMORY,
    SizeOf (T_STAT_PRO)));
    ZeroMemory (psp, SizeOf (T_STAT_PRO));

    psp.StatProc := TStatWndProc (Pointer (GetWindowLong (hWnd, GWL_WNDPROC)));
    psp.himl := 0;
    psp.imgSize := 0;
    psp.imgCount := 0;
    psp.imgCurrent := 0;
    psp.dwElapse := 50;

    KillTimer (hWnd, IDC_ANIMATETIMER);

    SetWindowLong (hWnd, GWL_USERDATA, Longint (psp));

    SetWindowLong (hWnd, GWL_WNDPROC, Longint (@StatWndProc));

    SendMessage (hWnd, WM_SIZE, 0, 0);

end;

//

procedure RemoveAnimateStatic (hWnd: HWND);
begin

    psp := P_STAT_PRO (GetWindowLong (hWnd, GWL_USERDATA));
    if (psp <> nil) then
        begin
            //
            if (psp.hdcMem <> 0) then
                begin
                    SelectObject (psp.hdcMem, psp.hbmOld);
                    DeleteObject (psp.hbmMem);
                    DeleteDC (psp.hdcMem);
                end;

            //
            KillTimer (hWnd, IDC_ANIMATETIMER);
            //
            SetWindowLong (hWnd, GWL_WNDPROC, Longint (@psp.StatProc));
            RedrawWindow (hWnd, @psp.rcClient, 0, RDW_INVALIDATE or RDW_ERASE);
            SetWindowLong (hWnd, GWL_USERDATA, 0);
            HeapFree (GetProcessHeap, 0, psp);
        end;

end;

end.

```

DMVPN_Sh1Obj.pas - інтерпретатор команд операційної системи

```

unit DMVPN_Sh1Obj;

interface

uses
  Windows, DMVPN_Active;

{ Ідентифікація об'єкту у просторі імен(ItemID and IDList) }

const
  // Мережні та Dial-up підключення
  CSIDL_CONNECTIONS = $0031;

{ SHGetSpecialFolderLocation constants }

const
  // Десктоп
  CSIDL_DESKTOP = $0000;

{ Інтерфейс IDs }

const
  IIDMVPN_IShellFolder: TGUID = (D1: $000214E6; D2: $0000; D3: $0000; D4: ($C0,
  $00, $00, $00, $00, $00, $00, $46));
  CLSIDMVPN_ShellLink : TGUID = (D1: $00021401; D2: $0000; D3: $0000; D4: ($C0,
  $00, $00, $00, $00, $00, $00, $46));

{ IShellFolder.GetDisplayNameOf/SetNameOf uFlags }

const
  // по замовчуванню (виключно для дисплею)
  SHGDN_NORMAL = 0;
  // для перегляду по замовчуванню
  SHCONTDMVPN_NONFOLDERS = 64;

{ Рядкові константи для інтерфейсу IDs }

const
  SIDMVPN_IShellLink = '{000214 0000-0000-C 000-0000000000046}';
  SIDMVPN_IShellLink = '{000214F 0000-0000-C 000-0000000000046}';
  SIDMVPN_IShellFolder = '{000214E 0000-0000-C 000-0000000000046}';
  SIDMVPN_IEnumIDList = '{000214F 0000-0000-C 000-0000000000046}';

{ IShellLink інтерфейс }

type
  IShellLink = interface(IUnknown)
    [SIDMVPN_IShellLink]
    function GetPath(pszFile: PAnsiChar; cchMaxPath: Integer; var pfd:
    TWin32FindData; fFlags: DWORD): HRESULT; stdcall;
    function GetIDList(var ppidl: PItemIDList): HRESULT; stdcall;
    function SetIDList(pidl: PItemIDList): HRESULT; stdcall;
    function GetDescription(pszName: PAnsiChar; cchMaxName: Integer): HRESULT;
    stdcall;
    function SetDescription(pszName: PAnsiChar): HRESULT; stdcall;
    function GetWorkingDirectory(pszDir: PAnsiChar; cchMaxPath: Integer):
    HRESULT; stdcall;
    function SetWorkingDirectory(pszDir: PAnsiChar): HRESULT; stdcall;
    function GetArguments(pszArgs: PAnsiChar; cchMaxPath: Integer): HRESULT;
    stdcall;
    function SetArguments(pszArgs: PAnsiChar): HRESULT; stdcall;
    function GetHotkey(var pwHotkey: Word): HRESULT; stdcall;
    function SetHotkey(wHotkey: Word): HRESULT; stdcall;
    function GetShowCmd(out piShowCmd: Integer): HRESULT; stdcall;
    function SetShowCmd(iShowCmd: Integer): HRESULT; stdcall;
  end;

```

```

    function GetIconLocation(pszIconPath: PAnsiChar; cchIconPath: Integer; out
piIcon: Integer): HRESULT; stdcall;
    function SetIconLocation(pszIconPath: PAnsiChar; iIcon: Integer): HRESULT;
stdcall;
    function SetRelativePath(pszPathRel: PAnsiChar; dwReserved: DWORD): HRESULT;
stdcall;
    function Resolve(Wnd: HWND; fFlags: DWORD): HRESULT; stdcall;
    function SetPath(pszFile: PAnsiChar): HRESULT; stdcall;
end;

IShellLink = interface(IUnknown)
[SIDMVPN_IShellLink]
    function GetPath(pszFile: PWideChar; cchMaxPath: Integer; var pfd:
TWin32FindData; fFlags: DWORD): HRESULT; stdcall;
    function GetIDList(var ppidl: PItemIDList): HRESULT; stdcall;
    function SetIDList(pidl: PItemIDList): HRESULT; stdcall;
    function GetDescription(pszName: PWideChar; cchMaxName: Integer): HRESULT;
stdcall;
    function SetDescription(pszName: PWideChar): HRESULT; stdcall;
    function GetWorkingDirectory(pszDir: PWideChar; cchMaxPath: Integer):
HRESULT; stdcall;
    function SetWorkingDirectory(pszDir: PWideChar): HRESULT; stdcall;
    function GetArguments(pszArgs: PWideChar; cchMaxPath: Integer): HRESULT;
stdcall;
    function SetArguments(pszArgs: PWideChar): HRESULT; stdcall;
    function GetHotkey(var pwHotkey: Word): HRESULT; stdcall;
    function SetHotkey(wHotkey: Word): HRESULT; stdcall;
    function GetShowCmd(out piShowCmd: Integer): HRESULT; stdcall;
    function SetShowCmd(iShowCmd: Integer): HRESULT; stdcall;
    function GetIconLocation(pszIconPath: PWideChar; cchIconPath: Integer; out
piIcon: Integer): HRESULT; stdcall;
    function SetIconLocation(pszIconPath: PWideChar; iIcon: Integer): HRESULT;
stdcall;
    function SetRelativePath(pszPathRel: PWideChar; dwReserved: DWORD): HRESULT;
stdcall;
    function Resolve(Wnd: HWND; fFlags: DWORD): HRESULT; stdcall;
    function SetPath(pszFile: PWideChar): HRESULT; stdcall;
end;
IShellLink = IShellLink;

{ IEnumIDList інтерфейс }

type
    IEnumIDList = interface(IUnknown)
[SIDMVPN_IEnumIDList]
        function Next(celt: ULONG; out rgelt: PItemIDList; var pceltFetched: ULONG):
HRESULT; stdcall;
        function Skip(celt: ULONG): HRESULT; stdcall;
        function Reset: HRESULT; stdcall;
        function Clone(out ppenum: IEnumIDList): HRESULT; stdcall;
end;

{ record for returning strings from IShellFolder member functions }

type
    PSTRRet = ^TStrRet;
    _STRRET = record
        uType: UINT; { одне з значень STRRET_* }
        case Integer of
            0: (pOleStr: LPWSTR); { повинно бути вільно для
виклику GetDisplayNameOf }
            1: (pStr: LPSTR); { НЕ ВИКОРИСТОВУЄТЬСЯ }
            2: (uOffset: UINT); { Зсув у SHITEMID (ANSI) }
            3: (cStr: array[0..MAX_PATH-1] of Char); { Буфер для заповнювання }
        end;
    TStrRet = _STRRET;
    STRRET = _STRRET;

{ structure STRRET for returning strings from IShellFolder member functions }

```

```

const
    STRRET_WSTR = $0000;
    STRRET_CSTR = $0002;

{ IShellFolder интерфейс }

type
    IShellFolder = interface(IUnknown)
        [SIDMVPN_IShellFolder]
        function ParseDisplayName(hwndOwner: HWND; pbcReserved: Pointer;
lpszDisplayName: POLESTR; out pchEaten: ULONG; out ppidl: PItemIDList; var
dwAttributes: ULONG): HRESULT; stdcall;
        function EnumObjects(hwndOwner: HWND; grfFlags: DWORD; out EnumIDList:
IEnumIDList): HRESULT; stdcall;
        function BindToObject(pidl: PItemIDList; pbcReserved: Pointer; const riid:
TIID; out ppvOut): HRESULT; stdcall;
        function BindToStorage(pidl: PItemIDList; pbcReserved: Pointer; const riid:
TIID; out ppvObj): HRESULT; stdcall;
        function CompareIDs(lParam: LPARAM; pidl1, pidl2: PItemIDList): HRESULT;
stdcall;
        function CreateViewObject(hwndOwner: HWND; const riid: TIID; out ppvOut):
HRESULT; stdcall;
        function GetAttributesOf(cidl: UINT; var apidl: PItemIDList; var rgfInOut:
UINT): HRESULT; stdcall;
        function GetUIObjectOf(hwndOwner: HWND; cidl: UINT; var apidl: PItemIDList;
const riid: TIID; prgfInOut: Pointer; out ppvOut): HRESULT; stdcall;
        function GetDisplayNameOf(pidl: PItemIDList; uFlags: DWORD; var lpName:
STRRET): HRESULT; stdcall;
        function SetNameOf(hwndOwner: HWND; pidl: PItemIDList; lpszName: POLEStr;
uFlags: DWORD; var ppidlOut: PItemIDList): HRESULT; stdcall;
        end;

function SHGetMalloc(var ppMalloc: IMalloc): HRESULT; stdcall;
function SHGetFolderLocation(hwndOwner: HWND; csidl: Integer; hToken: THandle;
dwReserved: DWORD; var pidl: PItemIDList): HRESULT; stdcall;
function SHGetDesktopFolder(var ppshf: IShellFolder): HRESULT; stdcall;
function SHGetSpecialFolderLocation(hwndOwner: HWND; nFolder: Integer; var
ppidl: PItemIDList): HRESULT; stdcall;
function SHGetPathFromIDList(pidl: PItemIDList; pszPath: PChar): BOOL; stdcall;
function SHGetPathFromIDList(pidl: PItemIDList; pszPath: PAnsiChar): BOOL;
stdcall;
function SHGetPathFromIDList(pidl: PItemIDList; pszPath: PWideChar): BOOL;
stdcall;

implementation

const
    shell132 = 'shell32.dll';

function SHGetMalloc;                external shell132 name 'SHGetMalloc';
function SHGetFolderLocation;        external shell132 name
'SHGetFolderLocation';
function SHGetDesktopFolder;         external shell132 name 'SHGetDesktopFolder';
function SHGetSpecialFolderLocation; external shell132 name
'SHGetSpecialFolderLocation';
function SHGetPathFromIDList;        external shell132 name
'SHGetPathFromIDList';
function SHGetPathFromIDList;        external shell132 name 'SHGetPathFromIDList';
function SHGetPathFromIDList;        external shell132 name 'SHGetPathFromIDList';

end.

```

DMVPN_Resources.pas - ресурси DMVPN

```
unit DMVPN_Resources;

interface

uses
  Windows, CommCtrl, DMVPN_FileInfo;

const

  { ресурси id діалогу}

  RC_DIALOG_WELCOME      = 101;
  RC_DIALOG_SETTINGS    = 102;
  RC_DIALOG_FINISH      = 103;
  RC_DIALOG_UPDATE      = 104;

  { ресурси id вікна}

  RC_ICONEX_CAPTION      = 101;

  { ресурси id бітової площини}

  RC_BITMAP_WATERMARK   = 101;
  RC_BITMAP_HEADER      = 102;
  RC_BITMAP_WAITING     = 103;

  { елементи управління діалога#101 }

  IDC_STATIC_WELCOME    = 10101;

  { елементи управління діалога#102 }

  IDC_STATIC_ENTRY      = 10201;
  IDC_STATIC_SERVER     = 10202;
  IDC_COMBO_SERVER      = 10203;
  IDC_STATIC_USER       = 10204;
  IDC_STATIC_PASSW      = 10205;
  IDC_STATIC_WARN       = 10206;

  { елементи управління діалога#103 }

  IDC_STATIC_FINISH     = 10301;
  IDC_STATIC_DMVPNINFO  = 10302;
  IDC_CHECK_SHORTCUT    = 10303;

  { елементи управління діалога#104 }

  IDC_STATIC_ANIMATE    = 10401;
  IDC_STATIC_ADDRESS    = 10402;

  { Ресурси рядків таблиць }

  RC_STRING_CWINDOW     = 1600;
  RC_STRING_COPYRUN     = 1601;
  RC_STRING_QCANCEL     = 1602;
  RC_STRING_RESMAN      = 1603;

  RC_STRING_THEADER     = 1616;
  RC_STRING_SHEADER     = 1617;

  RC_STRING_DMVPNINFO   = 1632;

  RC_STRING_IPSERVER    = 1648;
  RC_STRING_IPHOST      = 1649;
```

```
var
  psp      : TPropSheetPage;
  ahpsp    : Array [0..2] of HPropSheetPage;
  hApp     : Array [0..3] of HWND;
  exeInfo  : TStringFileInfo;
  pszServ  : WideString = 'server.avtograd.ru';
  hThread  : DWORD;
```

```
implementation
```

```
end.
```

Кафедра КБПЗ – 2021 рік

DMVPN_RasApi.pas - з'єднання DMVPN

```

unit DMVPN_RasApi;

interface

uses
  Windows;

// RASIPADDR структура

type
  PRASIPADDR = ^RASIPADDR;
  RASIPADDR = record
    a: Byte;
    b: Byte;
    c: Byte;
    d: Byte;
  end;

const
  RAS_MaxAreaCode      = 10;
  RAS_MaxPhoneNumber   = 128;
  RAS_MaxDeviceType    = 16;
  RAS_MaxDeviceName    = 128;
  RAS_MaxPadType       = 32;
  RAS_Max25Address     = 200;
  RAS_MaxFacilities    = 200;
  RAS_MaxUserData      = 200;
  RAS_MaxDnsSuffix     = 255;
  RAS_MaxEntryName     = 256;
  RAS_MaxCallbackNumber = RAS_MaxPhoneNumber;
  UNLEN                = 256; // Максимальна довжина імені користувача
  PWLEN                = 256; // Максимальна довжина паролю
  CNLEN                = 15;  // Максимальна довжина імені комп'ютера
  DNLEN                = CNLEN; // Максимальна довжина імені домену

// структура RASCREENTIALS

type
  RASCREENTIALSA = record
    dwSize      : DWORD;
    dwMask      : DWORD;
    szUserName: Array [0..UNLEN] of AnsiChar;
    szPassword: Array [0..PWLEN] of AnsiChar;
    szDomain   : Array [0..DNLEN] of AnsiChar;
  end;

  RASCREENTIALSW = record
    dwSize      : DWORD;
    dwMask      : DWORD;
    szUserName: Array [0..UNLEN] of WideChar;
    szPassword: Array [0..PWLEN] of WideChar;
    szDomain   : Array [0..DNLEN] of WideChar;
  end;

  LPRASCREENTIALSW = ^RASCREENTIALSW;
  LPRASCREENTIALSA = ^RASCREENTIALSA;
  LPRASCREENTIALS  = ^RASCREENTIALS;
  RASCREENTIALS    = RASCREENTIALSA;

const
  // значення RASCREENTIALS dwMask

  RASCM_UserName = $00000001;
  RASCM_Password = $00000002;

```

```
// структура RASDIALPARAMS
```

```
type
```

```
tagRASDIALPARAMSA = record
    dwSize          : DWORD;
    szEntryName     : Array [0..RAS_MaxEntryName] of AnsiChar;
    szPhoneNumber   : Array [0..RAS_MaxPhoneNumber] of AnsiChar;
    szCallbackNumber: Array [0..RAS_MaxCallbackNumber] of AnsiChar;
    szUserName      : Array [0..UNLEN] of AnsiChar;
    szPassword      : Array [0..PWLEN] of AnsiChar;
    szDomain        : Array [0..DNLEN] of AnsiChar;
    // {$IFDEF WINVER_0x401_OR_GREATER}
    dwSubEntry      : DWORD;
    dwCallbackId    : DWORD;
end;
```

```
tagRASDIALPARAMSW = record
    dwSize          : DWORD;
    szEntryName     : Array [0..RAS_MaxEntryName] of WideChar;
    szPhoneNumber   : Array [0..RAS_MaxPhoneNumber] of WideChar;
    szCallbackNumber: Array [0..RAS_MaxCallbackNumber] of WideChar;
    szUserName      : Array [0..UNLEN] of WideChar;
    szPassword      : Array [0..PWLEN] of WideChar;
    szDomain        : Array [0..DNLEN] of WideChar;
    // {$IFDEF WINVER_0x401_OR_GREATER}
    dwSubEntry      : DWORD;
    dwCallbackId    : DWORD;
end;
```

```
PRASDIALPARAMSA = ^RASDIALPARAMSA;
PRASDIALPARAMSW = ^RASDIALPARAMSW;
PRASDIALPARAMS = PRASDIALPARAMSA;
tagRASDIALPARAMS = tagRASDIALPARAMSA;
RASDIALPARAMSA = tagRASDIALPARAMSA;
RASDIALPARAMSW = tagRASDIALPARAMSW;
RASDIALPARAMS = RASDIALPARAMSA;
```

```
// структура RASENTRY
```

```
type
```

```
tagRASENTRYA = record
    dwSize          : DWORD;
    dwfOptions      : DWORD;

    // Настроювання мережевого номера
    dwCountryID     : DWORD;
    dwCountryCode   : DWORD;
    szAreaCode      : Array [0..RAS_MaxAreaCode] of AnsiChar;
    szLocalPhoneNumber : Array [0..RAS_MaxPhoneNumber] of AnsiChar;
    dwAlternateOffset : DWORD;

    // PPP(Протокол Point-to-point)/Ip
    ipaddr          : RASIPADDR;
    ipaddrDns       : RASIPADDR;
    ipaddrDnsAlt    : RASIPADDR;
    ipaddrWins      : RASIPADDR;
    ipaddrWinsAlt   : RASIPADDR;

    // Протокол
    dwFrameSize     : DWORD;
    dwfNetProtocols : DWORD;
    dwFramingProtocol : DWORD;

    // Сценарії
    szScript        : Array [0..MAX_PATH-1] of AnsiChar;

    // Автодозвон
    szAutodialDll   : Array [0..MAX_PATH-1] of AnsiChar;
    szAutodialFunc  : Array [0..MAX_PATH-1] of AnsiChar;
```

```

// Пристрій
szDeviceType           : Array [0..RAS_MaxDeviceType] of AnsiChar;
szDeviceName           : Array [0..RAS_MaxDeviceName] of AnsiChar;

// X.25
sz25PadType            : Array [0..RAS_MaxPadType] of AnsiChar;
sz25Address             : Array [0..RAS_Max25Address] of AnsiChar;
sz25Facilities          : Array [0..RAS_MaxFacilities] of AnsiChar;
sz25UserData           : Array [0..RAS_MaxUserData] of AnsiChar;
dwChannels              : DWORD;

// Зарезервовано
dwReserved1            : DWORD;
dwReserved2            : DWORD;
// {$IFDEF WINVER_0x401_OR_GREATER}

// Підключення з багатьох з'єднань
dwSubEntries           : DWORD;
dwDialMode              : DWORD;
dwDialExtraPercent     : DWORD;
dwDialExtraSampleSeconds : DWORD;
dwHangUpExtraPercent   : DWORD;
dwHangUpExtraSampleSeconds : DWORD;

// Час простою до роз'єднання
dwIdleDisconnectSeconds : DWORD;
// {$IFDEF WINVER_0x500_OR_GREATER}
dwType                  : DWORD;
dwEncryptionType        : DWORD;
dwCustomAuthKey         : DWORD;
guidId                  : TGUID;
szCustomDialDll         : Array [0..MAX_PATH-1] of AnsiChar;
dwDMVPNStrategy        : DWORD;
// {$IFDEF WINVER_0x501_OR_GREATER}
dwfOptions2             : DWORD;
dwfOptions3             : DWORD;
szDnsSuffix             : Array [0..RAS_MaxDnsSuffix] of AnsiChar;
dwTcpWindowSize        : DWORD;
szPrerequisitePbk       : Array [0..MAX_PATH-1] of AnsiChar;
szPrerequisiteEntry     : Array [0..RAS_MaxEntryName] of AnsiChar;
dwRedialCount           : DWORD;
dwRedialPause           : DWORD;
// {$IFDEF WINVER_0x600_OR_GREATER}
//   ipv6addrDns         : RASIPV6ADDR;
//   ipv6addrDnsAlt      : RASIPV6ADDR;
// {$ENDIF}
//   dwIPv4InterfaceMetric : DWORD;
//   dwIPv6InterfaceMetric : DWORD;
end;

tagRASENTRYW = record
  dwSize           : DWORD;
  dwfOptions       : DWORD;

  // Настроювання мережного номера
  dwCountryID      : DWORD;
  dwCountryCode    : DWORD;
  szAreaCode       : Array [0..RAS_MaxAreaCode] of WideChar;
  szLocalPhoneNumber : Array [0..RAS_MaxPhoneNumber] of WideChar;
  dwAlternateOffset : DWORD;

  // PPP (Протокол Point-to-point) / Ip
  ipaddr           : RASIPADDR;
  ipaddrDns        : RASIPADDR;
  ipaddrDnsAlt     : RASIPADDR;
  ipaddrWins       : RASIPADDR;
  ipaddrWinsAlt    : RASIPADDR;

```

```

// Протокол
dwFrameSize           : DWORD;
dwfNetProtocols       : DWORD;
dwFramingProtocol     : DWORD;

// Сценарій
szScript              : Array [0..MAX_PATH-1] of WideChar;

// Автодозвон
szAutodialDll         : Array [0..MAX_PATH-1] of WideChar;
szAutodialFunc        : Array [0..MAX_PATH-1] of WideChar;

// Пристрій
szDeviceType          : Array [0..RAS_MaxDeviceType] of WideChar;
szDeviceName          : Array [0..RAS_MaxDeviceName] of WideChar;

// X.25
sz25PadType           : Array [0..RAS_MaxPadType] of WideChar;
sz25Address            : Array [0..RAS_Max25Address] of WideChar;
sz25Facilities         : Array [0..RAS_MaxFacilities] of WideChar;
sz25UserData          : Array [0..RAS_MaxUserData] of WideChar;
dwChannels            : DWORD;

// Зарезервовано
dwReserved1           : DWORD;
dwReserved2           : DWORD;
// {$IFDEF WINVER_0x401_OR_GREATER}

// Підключення з багатьох з'єднань
dwSubEntries          : DWORD;
dwDialMode             : DWORD;
dwDialExtraPercent    : DWORD;
dwDialExtraSampleSeconds : DWORD;
dwHangUpExtraPercent  : DWORD;
dwHangUpExtraSampleSeconds : DWORD;

// Час простою до роз'єднання
dwIdleDisconnectSeconds : DWORD;
// {$IFDEF WINVER_0x500_OR_GREATER}
dwType                : DWORD;
dwEncryptionType      : DWORD;
dwCustomAuthKey       : DWORD;
guidId                : TGUID;
szCustomDialDll       : Array [0..MAX_PATH-1] of WideChar;
dwDMVPNStrategy       : DWORD;
// {$IFDEF WINVER_0x501_OR_GREATER}
dwfOptions2           : DWORD;
dwfOptions3           : DWORD;
szDnsSuffix           : Array [0..RAS_MaxDnsSuffix] of WideChar;
dwTcpWindowSize      : DWORD;
szPrerequisitePbk     : Array [0..MAX_PATH-1] of WideChar;
szPrerequisiteEntry   : Array [0..RAS_MaxEntryName] of WideChar;
dwRedialCount         : DWORD;
dwRedialPause         : DWORD;
// {$IFDEF WINVER_0x600_OR_GREATER}
//   ipv6addrDns       : RASIPV6ADDR;
//   ipv6addrDnsAlt    : RASIPV6ADDR;
// {$ENDIF}
//   dwIPv4InterfaceMetric : DWORD;
//   dwIPv6InterfaceMetric : DWORD;
end;

tagRASENTRY = tagRASENTRYA;
RASENTRYA = tagRASENTRYA;
RASENTRYW = tagRASENTRYW;
RASENTRY = RASENTRYA;

const
// RASENTRY dwfOptions bit flags

```

```

RASEO_RemoteDefaultGateway      = $00000010;
RASEO_ModemLights               = $00000100;
RASEO_RequireEncryptedPw       = $00000400;
RASEO_RequireMsEncryptedPw     = $00000800;
RASEO_RequireDataEncryption    = $00001000;
RASEO_PreviewUserPw            = $01000000;
RASEO_ShowDialingProgress      = $04000000;

// Біти прапорів RASENTRY dwfOptions

RASEO2_DontNegotiateMultilink   = $00000004;
RASEO2_ReconnectIfDropped      = $00000100;

// Біти прапорів RASENTRY dwProtocols

RASNP_Ip = $00000004;

// Біти прапорів RASENTRY dwFramingProtocols

RASFP_Ppp = $00000001;

// константи RASENTRY dwIdleDisconnectSeconds

RASIDS_Disabled = $FFFFFFFF;

// рядок по замовчуванню RASENTRY szDeviceType

RASDT_DMVPN = 'DMVPN';

// значення RASENTRY dwDialMode

RASEDM_DialAll = 1;

// Тип входу використаний, для визначення того, які властивості UI повині бути
// представлені споживачу

RASET_DMVPN = 2; // DMVPN
// Немає ніякої різниці між RASCTRYINFOA та RASCTRYINFOW.

ET_None      = 0; // Без шифрування
VS_Default   = 0; // по замовчуванню (PPTP)

function RasSetEntryProperties(lpszPhonebook, szEntry: PAnsiChar; lpbEntry:
Pointer; dwEntrySize: Longint; lpbDeviceInfo: Pointer; dwDeviceInfoSize:
Longint): Longint; stdcall;
function RasSetEntryProperties(lpszPhonebook, szEntry: PWideChar; lpbEntry:
Pointer; dwEntrySize: Longint; lpbDeviceInfo: Pointer; dwDeviceInfoSize:
Longint): Longint; stdcall;
function RasSetEntryProperties(lpszPhonebook, szEntry: PAnsiChar; lpbEntry:
Pointer; dwEntrySize: Longint; lpbDeviceInfo: Pointer; dwDeviceInfoSize:
Longint): Longint; stdcall;

function RasGetEntryProperties(lpszPhonebook, szEntry: PAnsiChar; lpbEntry:
Pointer; var lpdwEntrySize: Longint; lpbDeviceInfo: Pointer; var
lpdwDeviceInfoSize: Longint): Longint; stdcall;
function RasGetEntryProperties(lpszPhonebook, szEntry: PWideChar; lpbEntry:
Pointer; var lpdwEntrySize: Longint; lpbDeviceInfo: Pointer; var
lpdwDeviceInfoSize: Longint): Longint; stdcall;
function RasGetEntryProperties(lpszPhonebook, szEntry: PAnsiChar; lpbEntry:
Pointer; var lpdwEntrySize: Longint; lpbDeviceInfo: Pointer; var
lpdwDeviceInfoSize: Longint): Longint; stdcall;

function RasSetEntryDialParams(lpszPhonebook: PAnsiChar; lprasdialparams:
PRASDIALPARAMSA; fRemovePassword: BOOL): DWORD; stdcall;
function RasSetEntryDialParams(lpszPhonebook: PWideChar; lprasdialparams:
PRASDIALPARAMSW; fRemovePassword: BOOL): DWORD; stdcall;
function RasSetEntryDialParams(lpszPhonebook: PAnsiChar; lprasdialparams:
PRASDIALPARAMS; fRemovePassword: BOOL): DWORD; stdcall;

```

```
function RasSetCredentials(lpszPhoneBook, lpszEntry: PAnsiChar; var
lpCredentials: RASCREENTIALSA; fRemovePassword: LongBool): Longint; stdcall;
function RasSetCredentials(lpszPhoneBook, lpszEntry: PWideChar; var
lpCredentials: RASCREENTIALSW; fRemovePassword: LongBool): Longint; stdcall;
function RasSetCredentials(lpszPhoneBook, lpszEntry: PAnsiChar; var
lpCredentials: RASCREENTIALS; fRemovePassword: LongBool): Longint; stdcall;
```

implementation

const

raslib = 'rasapi32.dll';

```
function RasSetEntryProperties; external raslib name 'RasSetEntryProperties';
function RasSetEntryProperties; external raslib name 'RasSetEntryProperties';
function RasSetEntryProperties; external raslib name 'RasSetEntryProperties';
```

```
function RasGetEntryProperties; external raslib name 'RasGetEntryProperties';
function RasGetEntryProperties; external raslib name 'RasGetEntryProperties';
function RasGetEntryProperties; external raslib name 'RasGetEntryProperties';
```

```
function RasSetEntryDialParams; external raslib name 'RasSetEntryDialParams';
function RasSetEntryDialParams; external raslib name 'RasSetEntryDialParams';
function RasSetEntryDialParams; external raslib name 'RasSetEntryDialParams';
```

```
function RasSetCredentials; external raslib name 'RasSetCredentials';
function RasSetCredentials; external raslib name 'RasSetCredentials';
function RasSetCredentials; external raslib name 'RasSetCredentials';
end.
```

Кафедра — КБПЗ —

DMVPN_MyMsgBox.pas - повідомлення DMVPN

```
unit DMVPN_MyMsgBox;

interface

uses
  Windows, Messages, DMVPN_SysUtils;

function ExtMessageBox(hWnd: HWND; pszText, pszCaption: PWideChar; dwFlags:
DWORD): Integer;

implementation

var
  hhk: HHOOK;
  ico: HICON;

//

function SysMsgProc(nCode: UINT; wParam: WPARAM; lParam: LPARAM): Integer;
stdcall;
begin
  case nCode of
    HCBT_ACTIVATE:
      begin
        if (ico <> 0) then
          SendMessage(wParam, WM_SETICON, ICON_SMALL, ico);
          SetCenterDialogPos(wParam, GetParent(wParam), TRUE);
          UnhookWindowsHookEx(hhk);
          Result := 0;
        end;
      else
        Result := CallNextHookEx(hhk, nCode, wParam, lParam);
      end;
  end;
end;

//

function ExtMessageBox(hWnd: HWND; pszText, pszCaption: PWideChar; dwFlags:
DWORD): Integer;
begin
  ico := GetClassLong(hWnd, GCL_HICON);
  if (ico = 0) then
    ico := SendMessage(hWnd, WM_GETICON, ICON_SMALL, 0);
  hhk := SetWindowsHookEx(WH_CBT, @SysMsgProc, hInstance, 0);
  Result := MessageBox(hWnd, pszText, pszCaption, dwFlags);
end;
end.
```

DMVPN_LinkStat.pas - інтерфейс користувача

```

unit DMVPN_LinkStat;

interface

uses
  Windows, Messages, CommCtrl, DMVPN_Windows;

const
  //
  SCM_EX_SETHOVERCLR = WM_USER + 101; // установити колір для наведеного стану.
  SCM_EX_SETNORMALCLR = WM_USER + 102; // установити колір для звичайного стану.
  SCM_EX_SETPRESSCLR = WM_USER + 103; // установити колір для натиснутого
стану.
  SCM_EX_SETBCKGNDCLR = WM_USER + 104; // установити колір для тла тексту.
  SCM_EX_SETTIPTTEXT = WM_USER + 105; // установити текст спливаючої підказки.
  //
  SCM_EX_GETHOVERCLR = WM_USER + 111; // одержати колір для наведеного стану.
  SCM_EX_GETNORMALCLR = WM_USER + 112; // одержати колір для звичайного стану.
  SCM_EX_GETPRESSCLR = WM_USER + 113; // одержати колір для натиснутого стану.
  SCM_EX_GETBCKGNDCLR = WM_USER + 114; // одержати колір для тла тексту.
  SCM_EX_GETTIPTTEXT = WM_USER + 115; // одержати текст спливаючої підказки.

  // створення елемента управління Hyperlink.

procedure CreateStaticHyperlink(hWnd: HWND);

  // видалення елемента управління Hyperlink.

procedure RemoveStaticHyperlink(hWnd: HWND);

implementation

type
  TLinkWndProc = function(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT; stdcall;

  P_LINK_PRO = ^T_LINK_PRO;
  T_LINK_PRO = packed record
    LinkProc : TLinkWndProc;
    hCursor  : HCURSOR;
    hFont    : HFONT;
    rcClient : TRect;
    //
    clrHover : TColorRef;
    clrNormal : TColorRef;
    clrPress : TColorRef;
    clrBckgnd : TColorRef; // CLR_NONE
    pszText  : Array [0..MAX_PATH-1] of WideChar;
    //
    bIsHover : Boolean;
    bIsPress : Boolean;
    bIsEnabled: Boolean;
    //
    hToolTip : HWND;
    ti       : TToolInfo;
    pszToolTip: Array [0..MAX_PATH-1] of WideChar;
    //
    dtStyle  : DWORD;
    //
    hdcMem   : HDC;
    hbmMem   : HBITMAP;
    hbmOld   : HBITMAP;
  end;

var
  plp: P_LINK_PRO;

```

```

//

function LinkWndProc_OnSetHoverClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.clrHover := TColorRef(wParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //
    Result := 0;
end;

//

function LinkWndProc_OnGetHoverClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //
    Result := LRESULT(plp.clrHover);
end;

//

function LinkWndProc_OnSetNormalClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.clrNormal := TColorRef(wParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //
    Result := 0;
end;

//

function LinkWndProc_OnGetNormalClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //
    Result := LRESULT(plp.clrNormal);
end;

//

function LinkWndProc_OnSetPressClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.clrPress := TColorRef(wParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //
    Result := 0;
end;

//

function LinkWndProc_OnGetPressClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //
    Result := LRESULT(plp.clrPress);
end;

//

function LinkWndProc_OnSetBckgdClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.clrBckgd := TColorRef(wParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //

```

```

    Result := 0;
end;

//

function LinkWndProc_OnGetBckgdClr(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //
    Result := LRESULT(plp.clrBckgd);
end;

//

function LinkWndProc_OnSetTipText(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    lstrcpyn(plp.pszToolTip, PWideChar(wParam), wParam);
    //
    Result := 0;
end;

//

function LinkWndProc_OnGetTipText(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    //
    lstrcpyn(PWideChar(lParam), plp.pszToolTip, lstrlen(plp.pszToolTip) + 1);
    //
    Result := 0;
end;

//

function LinkWndProc_OnWmSetFont(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.hFont := HFONT(wParam);
    //
    Result := CallWindowProc(@plp.LinkProc, hWnd, uMsg, wParam, lParam);
    //
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
end;

//

function LinkWndProc_OnWmSetText(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
    ZeroMemory(@plp.pszText, SizeOf(plp.pszText));
    lstrcpyn(plp.pszText, PWideChar(lParam), lParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    Result := DefWindowProc(hWnd, uMsg, wParam, lParam);
end;

//

function LinkWndProc_OnWmEnable(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT;
begin
    plp.bIsEnabled := BOOL(wParam);
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //
    Result := 0;
end;

//

```

```

function LinkWndProc_OnWmMouseLeave(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
var
  pt: TPoint;
begin
  if IsWindow(plp.hToolTip) then
    SendMessage(plp.hToolTip, TTM_TRACKACTIVATE, Integer(FALSE), 0);
  //
  GetCursorPos(pt);
  ScreenToClient(hWnd, pt);
  //
  plp.bIsHover := FALSE;
  plp.bIsPress := FALSE;
  //
  RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
  //
  Result := 0;
end;

//

function LinkWndProc_OnWmMouseMove(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
var
  tme: Windows.TTrackMouseEvent;
  pt : TPoint;
begin
  //
  GetCursorPos(pt);
  ScreenToClient(hWnd, pt);
  //
  tme.cbSize      := SizeOf(Windows.TTrackMouseEvent);
  tme.dwFlags     := TME_LEAVE;
  tme.hwndTrack  := hWnd;
  tme.dwHoverTime := HOVER_DEFAULT;
  //
  plp.bIsHover := Windows.TrackMouseEvent(tme) and PtInRect(plp.rcClient, pt);
  plp.bIsPress := {(wParam = MK_LBUTTON) and} (GetCapture = hWnd) and
PtInRect(plp.rcClient, pt);
  //
  RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
  //
  Result := 0;
end;

//

function LinkWndProc_OnWmCaptureChanged(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  plp.bIsPress := FALSE;
  //
  Result := 0;
end;

//

function LinkWndProc_OnWmNcHitTest(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  //
  Result := HTCLIENT;
end;

//

function LinkWndProc_OnWmLButtonDown(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin

```

```

//
if IsWindow(plp.hToolTip) then
    SendMessage(plp.hToolTip, TTM_TRACKACTIVATE, Integer(FALSE), 0);
plp.bIsPress := TRUE;
SetFocus(hWnd);
SetCapture(hWnd);
//
RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
//
Result := 0;
end;

//

function LinkWndProc_OnWmlButtonUp(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
var
    pt: TPoint;
begin
    //
    GetCursorPos(pt);
    ScreenToClient(hWnd, pt);
    if (PtInRect(plp.rcClient, pt) and (GetCapture = hWnd)) then
        SendMessage(GetParent(hWnd), WM_COMMAND, MakeLong(GetDlgCtrlID(hWnd),
STN_CLICKED), 0);
        // plp.bIsPress := FALSE;
        ReleaseCapture;
        //
        RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
        //
        Result := 0;
end;

//

function LinkWndProc_OnWmSetCursor(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
var
    pt: TPoint;
begin
    //
    if IsWindow(plp.hToolTip) then
        begin
            SendMessage(plp.hToolTip, TTM_TRACKACTIVATE, Integer(TRUE),
Integer(@plp.ti));
            GetCursorPos(pt);
            SendMessage(plp.hToolTip, TTM_TRACKPOSITION, 0, MakeLong(pt.x, pt.y));
        end;
    //
    if (plp.hCursor <> 0) then
        SetCursor(plp.hCursor);
    //
    Result := 0;
end;

//

function LinkWndProc_OnWmSize(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT;
var
    hdcIn: HDC;
begin
    GetClientRect(hWnd, plp.rcClient);
    //
    if (plp.hdcMem <> 0) then
        begin
            SelectObject(plp.hdcMem, plp.hbmOld);
            DeleteObject(plp.hbmMem);
            DeleteDC(plp.hdcMem);
        end;
end;

```

```

    end;
    hdcIn := GetDC(hWnd);
    plp.hdcMem := CreateCompatibleDC(hdcIn);
    plp.hbmMem := CreateCompatibleBitmap(hdcIn, plp.rcClient.Right -
plp.rcClient.Left, plp.rcClient.Bottom - plp.rcClient.Top);
    plp.hbmOld := SelectObject(plp.hdcMem, plp.hbmMem);
    ReleaseDC(hWnd, hdcIn);
    //
    RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
    //
    Result := CallWindowProc(@plp.LinkProc, hWnd, uMsg, wParam, lParam);
end;

//

function LinkWndProc_OnWmPaint(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT; stdcall;
var
    hdcIn : HDC;
    ps    : TPaintStruct;
    hbrNew: HBRUSH;
begin
    if (wParam = 0) then
        hdcIn := BeginPaint(hWnd, ps)
    else
        hdcIn := wParam;

    if (plp.clrBckgnd = CLR_DEFAULT) then
        FillRect(plp.hdcMem, plp.rcClient, HBRUSH(COLOR_BTNFACE + 1))
    else
        begin
            hbrNew := CreateSolidBrush(plp.clrBckgnd);
            FillRect(plp.hdcMem, plp.rcClient, hbrNew);
            DeleteObject(hbrNew);
        end;

    if plp.bIsEnabled then
        begin
            if (plp.bIsHover and plp.bIsPress) then
                SetTextColor(plp.hdcMem, plp.clrPress)
            else
                if (plp.bIsHover and not plp.bIsPress) then
                    SetTextColor(plp.hdcMem, plp.clrHover)
                else
                    SetTextColor(plp.hdcMem, plp.clrNormal);
            end
        end
    else
        SetTextColor(plp.hdcMem, GetSysColor(COLOR_GRAYTEXT));

    SetBkMode(plp.hdcMem, TRANSPARENT);
    SetBkColor(plp.hdcMem, TRANSPARENT);

    SelectObject(plp.hdcMem, plp.hFont);

    DrawText(plp.hdcMem, plp.pszText, {strlen(plp.pszText)}-1, plp.rcClient,
plp.dtStyle);

    BitBlt(hdcIn, 0, 0, plp.rcClient.Right - plp.rcClient.Left,
plp.rcClient.Bottom - plp.rcClient.Top, plp.hdcMem, 0, 0, SRCCOPY);

    if (wParam = 0) then
        EndPaint(hWnd, ps);

    Result := 0;
end;

//

```

```

function LinkWndProc_OnWmEraseBkgnd(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  if (plp.clrBckgnd <> CLR_DEFAULT) then
    begin
      FillRect(HDC(wParam), plp.rcClient, HBRUSH(COLOR_BTNFACE + 1));
      //
      Result := 1;
    end
  else
    Result := DefWindowProc(hWnd, uMsg, wParam, lParam);
end;

//

function LinkWndProc_OnWmSysColorChange(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT;
wParam: WPARAM; lParam: LPARAM): LRESULT;
begin
  RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE or RDW_UPDATENOW or RDW_NOERASE);
  //
  Result := 0;
end;

//

function LinkWndProc_OnWmNotify(plp: P_LINK_PRO; hWnd: HWND; uMsg: UINT; wParam:
WPARAM; lParam: LPARAM): LRESULT;
var
  pnmh: PNMHDR;
  ptit: PToolTipText;
begin
  //
  pnmh := PNMHDR(lParam);
  case pnmh.code of
    TTN_NEEDTEXTW:
      begin
        ptit := PToolTipText(lParam);
        ptit.lpszText := plp.pszToolTip;
      end;
  end;
  //
  Result := 0;
end;

//

function LinkWndProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
LRESULT; stdcall;
begin
  plp := P_LINK_PRO(GetWindowLong(hWnd, GWL_USERDATA));

  if (plp = nil) then
    begin
      Result := DefWindowProc(hWnd, uMsg, wParam, lParam);
      Exit;
    end;

  case uMsg of

    //

    SCM_EX_SETHOVERCLR:
      begin
        Result := LinkWndProc_OnSetHoverClr(plp, hWnd, uMsg, wParam, lParam);
      end;

    //

```

```
SCM_EX_GETHOVERCLR:
begin
    Result := LinkWndProc_OnGetHoverClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_SETNORMALCLR:
begin
    Result := LinkWndProc_OnSetNormalClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_GETNORMALCLR:
begin
    Result := LinkWndProc_OnGetNormalClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_SETPRESSCLR:
begin
    Result := LinkWndProc_OnSetPressClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_GETPRESSCLR:
begin
    Result := LinkWndProc_OnGetPressClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_SETBCKGNDCLR:
begin
    Result := LinkWndProc_OnSetBckgdClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_GETBCKGNDCLR:
begin
    Result := LinkWndProc_OnGetBckgdClr(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_SETTIPTTEXT:
begin
    Result := LinkWndProc_OnSetTipText(plp, hWnd, uMsg, wParam, lParam);
end;

//

SCM_EX_GETTIPTTEXT:
begin
    Result := LinkWndProc_OnGetTipText(plp, hWnd, uMsg, wParam, lParam);
end;

//

WM_DESTROY:
begin
    RemoveStaticHyperlink(hWnd);
end;

//
```

```
WM_SETFONT:
  begin
    Result := LinkWndProc_OnWmSetFont(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_SETTEXT:
  begin
    Result := LinkWndProc_OnWmSetText(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_ENABLE:
  begin
    Result := LinkWndProc_OnWmEnable(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_MOUSELEAVE:
  begin
    Result := LinkWndProc_OnWmMouseLeave(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_MOUSEMOVE:
  begin
    Result := LinkWndProc_OnWmMouseMove(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_CAPTURECHANGED:
  begin
    Result := LinkWndProc_OnWmCaptureChanged(plp, hWnd, uMsg, wParam,
lParam);
  end;

//

WM_NCHITTEST:
  begin
    Result := LinkWndProc_OnWmNcHitTest(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_LBUTTONDOWN:
  begin
    Result := LinkWndProc_OnWmLButtonDown(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_LBUTTONUP:
  begin
    Result := LinkWndProc_OnWmLButtonUp(plp, hWnd, uMsg, wParam, lParam);
  end;

//

WM_SETCURSOR:
  begin
    Result := LinkWndProc_OnWmSetCursor(plp, hWnd, uMsg, wParam, lParam);
  end;
```

```

//

WM_SIZE:
begin
    Result := LinkWndProc_OnWmSize(plp, hWnd, uMsg, wParam, lParam);
end;

//

WM_PRINTCLIENT,
WM_PAINT,
WM_UPDATEUISTATE: // перемальовування вікна без виклику WM_PAINT.
begin
    Result := LinkWndProc_OnWmPaint(plp, hWnd, uMsg, wParam, lParam);
end;

//

WM_ERASEBKGD:
begin
    Result := LinkWndProc_OnWmEraseBkgnd(plp, hWnd, uMsg, wParam, lParam);
end;

//

WM_SYSCOLORCHANGE:
begin
    Result := LinkWndProc_OnWmSysColorChange(plp, hWnd, uMsg, wParam,
lParam);
end;

//

WM_NOTIFY:
begin
    Result := LinkWndProc_OnWmNotify(plp, hWnd, uMsg, wParam, lParam);
end;

else
    Result := CallWindowProc(@plp.LinkProc, hWnd, uMsg, wParam, lParam);
end;

end;

//

procedure CreateStaticHyperlink(hWnd: HWND);
var
    iccex : TInitCommonControlsEx;
    dtStyle: DWORD;
    dwLen : Integer;
begin
    InitCommonControls;
    iccex.dwSize := SizeOf(TInitCommonControlsEx);
    iccex.dwICC := ICC_BAR_CLASSES;
    InitCommonControlsEx(iccex);

    RemoveStaticHyperlink(hWnd);

    plp := P_LINK_PRO(HeapAlloc(GetProcessHeap, HEAP_ZERO_MEMORY,
SizeOf(T_LINK_PRO)));

    ZeroMemory(plp, SizeOf(plp));
    plp.LinkProc := TLinkWndProc(Pointer(GetWindowLong(hWnd, GWL_WNDPROC)));
    plp.hCursor := LoadImage(0, MAKEINTRESOURCEW(IDC_HAND), IMAGE_CURSOR, 0, 0,
LR_SHARED or LR_DEFAULTSIZE);

```

```

plp.hFont      := SendMessage(hWnd, WM_GETFONT, 0, 0);

GetClientRect(hWnd, plp.rcClient);

plp.clrHover   := RGB(255, 0, 0);
plp.clrNormal := RGB(0, 0, 255);
plp.clrPress  := RGB(0, 0, 128);
plp.clrBckgnd := CLR_DEFAULT;

dwLen := SendMessage(hWnd, WM_GETTEXTLENGTH, 0, 0);
if (dwLen > 0) then
begin
  ZeroMemory(@plp.pszText, SizeOf(plp.pszText));
  SendMessage(hWnd, WM_GETTEXT, SizeOf(plp.pszText), Integer(@plp.pszText));
end;

plp.bIsHover   := FALSE;
plp.bIsPress   := FALSE;
plp.bIsEnabled := IsWindowEnabled(hWnd);

plp.hToolTip   := CreateWindowEx(WS_EX_TOPMOST, TOOLTIPS_CLASS, nil, WS_POPUP
or TTS_NOPREFIX or TTS_ALWAYSSTIP, Integer(CW_USEDEFAULT),
Integer(CW_USEDEFAULT), Integer(CW_USEDEFAULT), Integer(CW_USEDEFAULT),
GetParent(hWnd), 0, hInstance, nil);
if IsWindow(plp.hToolTip) then
begin
  plp.ti.cbSize := SizeOf(TToolInfo);
  plp.ti.uFlags := TTDMVPN_SUBCLASS or TTDMVPN_IDISHWND;
  plp.ti.hwnd   := hWnd;
  plp.ti.uId    := hWnd;
  plp.ti.lpszText := LPSTR_TEXTCALLBACKW;
  SetRectEmpty(plp.ti.Rect);
  ZeroMemory(@plp.pszToolTip, SizeOf(plp.pszToolTip));
  SendMessage(plp.hToolTip, TTM_ADDTOOLW, 0, Integer(@plp.ti));
end;

dtStyle := GetWindowLong(hWnd, GWL_STYLE);

case (dtStyle and SS_TYPMASK) of
  SS_LEFT      : plp.dtStyle := DT_LEFT or DT_EXPANDTABS {or
DT_WORDBREAK};
  SS_CENTER    : plp.dtStyle := DT_CENTER or DT_EXPANDTABS {or
DT_WORDBREAK};
  SS_RIGHT     : plp.dtStyle := DT_RIGHT or DT_EXPANDTABS {or
DT_WORDBREAK};
  SS_SIMPLE    : plp.dtStyle := DT_LEFT or DT_SINGLELINE;
  SS_LEFTNOWORDWRAP: plp.dtStyle := DT_LEFT or DT_EXPANDTABS;
end;
if ((dtStyle and SS_CENTERIMAGE) = 0) then
  plp.dtStyle := plp.dtStyle or DT_VCENTER;
if ((dtStyle and SS_NOTIFY) = 0) then
  SetWindowLong(hWnd, GWL_STYLE, dtStyle or SS_NOTIFY);

SetWindowLong(hWnd, GWL_USERDATA, Longint(plp));
SetWindowLong(hWnd, GWL_WNDPROC, Longint(@LinkWndProc));

// так як ми створюємо hdcMem заново при зміні розмірів вікна елемента
// управління, то не будемо тут створювати споконвічно контексти, а просто
// повідомимо елемент управління повідомленням про зміну розмірів.

SendMessage(hWnd, WM_SIZE, 0, 0); // RedrawWindow(hWnd, nil, 0, RDW_INVALIDATE
or RDW_UPDATENOW or RDW_NOERASE);

end;

//

procedure RemoveStaticHyperlink(hWnd: HWND);

```

```
begin
    plp := P_LINK_PRO(GetWindowLong(hWnd, GWL_USERDATA));
    if (plp <> nil) then
        begin
            if (plp.hCursor <> 0) then
                DestroyCursor(plp.hCursor);

            plp.ti.hwnd := hWnd;
            plp.ti.uId := hWnd;
            if IsWindow(plp.hToolTip) then
                begin
                    SendMessage(plp.hToolTip, TTM_DELTOW, 0, Integer(@plp.ti));
                    DestroyWindow(plp.hToolTip);
                end;

            if (plp.hdcMem <> 0) then
                begin
                    SelectObject(plp.hdcMem, plp.hbmOld);
                    DeleteObject(plp.hbmMem);
                    DeleteDC(plp.hdcMem);
                end;

            //
            SetWindowLong(hWnd, GWL_WNDPROC, Longint(@plp.LinkProc));
            RedrawWindow(hWnd, @plp.rcClient, 0, RDW_INVALIDATE or RDW_ERASE);

            SetWindowLong(hWnd, GWL_USERDATA, 0);
            HeapFree(GetProcessHeap, 0, plp);
        end;
    end;
end.
```

DMVPN_SettWind.pas - параметри програми роботи з DMVPN

```

unit DMVPN_SettWind;

interface

uses
  Windows, Messages, CommCtrl, DMVPN_FileInfo, DMVPN_LinkStat, DMVPN_SysUtils,
  DMVPN_MyMsgBox,
  DMVPN_Controls, DMVPN_Resources, DMVPN_ScanProc;

function SettDlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
  BOOL; stdcall;

implementation

//

function SettDlgProc_OnWmInitDialog(hWnd: HWND; uMsg: UINT; wParam: WPARAM;
lParam: LPARAM): LRESULT;
var
  bldfnt: HFONT;
begin
  //

  hApp[1] := hWnd;

  //

  CreateStaticHyperlink(GetDlgItem(hApp[1], IDC_STATIC_SERVER));

  //

  SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_ADDSTRING, 0,
    Integer(@pszServ[1]));
  SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_SETCURSEL, 0, 0);

  //

  bldfnt := GetWindowBoldFont(hApp[1], GetWindowFontSize(hApp[1], 8));
  if (bldfnt <> 0) then
    SendMessage(GetDlgItem(hApp[1], IDC_STATIC_WARN), WM_SETFONT,
      Integer(bldfnt), Integer(TRUE));

  //

  Result := 0;
end;

//

function SettDlgProc_OnWmCommand(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT;
const
  dwRes: Array [Boolean] of DWORD = (PSWIZB_BACK, PSWIZB_BACK or PSWIZB_NEXT);
var
  dwEntry: DWORD;
  dwUser : DWORD;
  dwPass : DWORD;
begin
  //

  case HiWord(wParam) of

```

```

//
BN_CLICKED:
  case LoWord(wParam) of

    //

    IDC_STATIC_SERVER:
      begin

        DialogBox(hInstance, MAKEINTRESOURCEW(RC_DIALOG_UPDATE), hApp[1],
          @ScanDlgProc);

      end;

    end;

//

EN_UPDATE:
  case LoWord(wParam) of
    IDC_STATIC_ENTRY,
    IDC_STATIC_USER,
    IDC_STATIC_PASSW:
      begin

        dwEntry := SendMessage(GetDlgItem(hApp[1], IDC_STATIC_ENTRY),
          WM_GETTEXTLENGTH, 0, 0);
        dwUser := SendMessage(GetDlgItem(hApp[1], IDC_STATIC_USER),
          WM_GETTEXTLENGTH, 0, 0);
        dwPass := SendMessage(GetDlgItem(hApp[1], IDC_STATIC_PASSW),
          WM_GETTEXTLENGTH, 0, 0);

        SendMessage(
          GetParent(hApp[1]),
          PSM_SETWIZBUTTONS,
          0,
          dwRes[(dwEntry > 0) and (dwUser > 0) and (dwPass > 0)]
        );

      end;

    end;

end;

//

Result := 0;

end;

//

function SettDlgProc_OnWmCtlColorStatic(hWnd: HWND; uMsg: UINT; wParam: WPARAM;
lParam: LPARAM): LRESULT;
begin
  //

  case GetDlgCtrlId(lParam) of
    IDC_STATIC_WARN:
      begin

        SetBkMode(wParam, TRANSPARENT);
        SetTextColor(wParam, RGB(255, 0, 0));
        Result := GetStockObject(NULL_BRUSH);

      end;

    else

```

```

    Result := 0;

end;

end;

//

function SettdlgProc_OnWmNotify(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT;
var
    pnmh : PNMHDR;
    dwRes: DWORD;
begin
    //

    pnmh := PNMHDR(lParam);

    case pnmh.code of

        //

        PSN_WIZNEXT:
        begin

            SendMessage(GetParent(hWnd), PSM_SETWIZBUTTONS, 0,
                Integer(PSWIZB_NEXT));

        end;

        //

        PSN_SETACTIVE:
        begin

            SendMessage(hWnd, WM_COMMAND, MAKELPARAM(IDC_STATIC_ENTRY, EN_UPDATE),
                0);
            SendMessage(hWnd, WM_COMMAND, MAKELPARAM(IDC_STATIC_USER, EN_UPDATE),
                0);
            SendMessage(hWnd, WM_COMMAND, MAKELPARAM(IDC_STATIC_PASSW, EN_UPDATE),
                0);

        end;

        //

        PSN_QUERYCANCEL:
        begin

            dwRes := ExtMessageBox(
                GetParent(hWnd),
                MAKEINTRESOURCEW(LoadStrInst(hInstance, RC_STRING_QCANCEL)),
                MAKEINTRESOURCEW(exeInfo.pszProductName),
                MB_YESNO or MB_ICONASTERISK
            );

            SetWindowLong(hWnd, DWL_MSGRESULT, Integer(dwRes = IDNO));

        end;

        //

        PSN_WIZBACK:
        begin

            SendMessage(GetParent(hWnd), PSM_SETCURSEL, GetParent(hWnd),
                Integer(ahpsp[1]));

```

```

    end;

    end;

    //

    Result := 1;

end;

//

function SettdlgProc_OnWmDestroy(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT;
var
    bldfnt: HFONT;
begin
    //

    RemoveStaticHyperlink(GetDlgItem(hWnd, IDC_STATIC_SERVER));

    //

    bldfnt := HFONT(SendMessage(GetDlgItem(hWnd, IDC_STATIC_WARN),
        WM_GETFONT, 0, 0));
    if (bldfnt <> 0) then
        DeleteObject(bldfnt);

    //

    Result := 0;

end;

//

function SettdlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
BOOL; stdcall;
begin
    case uMsg of

        //

        WM_INITDIALOG:
            begin
                Result := BOOL(SettdlgProc_OnWmInitDialog(hWnd, uMsg, wParam, lParam));
            end;

        //

        WM_COMMAND:
            begin
                Result := BOOL(SettdlgProc_OnWmCommand(hWnd, uMsg, wParam, lParam));
            end;

        //

        WM_CTLCOLORSTATIC:
            begin

```

```
    Result := BOOL(SettDlgProc_OnWmCtlColorStatic(hWnd, uMsg, wParam,
lParam));

    end;

    //

    WM_NOTIFY:
    begin

        Result := BOOL(SettDlgProc_OnWmNotify(hWnd, uMsg, wParam, lParam));

    end;

    //

    WM_DESTROY:
    begin

        Result := BOOL(SettDlgProc_OnWmDestroy(hWnd, uMsg, wParam, lParam));

    end;

    else
        Result := FALSE;
    end;

end;

end.
```

Кафедра_КБПЗ_2021рік

DMVPN_ScanProc.pas - пошук підключень DMVPN

```

unit DMVPN_ScanProc;

interface

uses
  Windows, Messages, CommCtrl, WinSock, DMVPN_SysUtils, DMVPN_StatAnim,
  DMVPN_Resources;

function ScanDlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
  BOOL; stdcall;

implementation

//

function ThreadCallback(LpParameter: Pointer): DWORD; stdcall;
type
  TaPinAddr = Array [0..MAX_PATH-1] of PInAddr;
  PaPinAddr = ^TaPinAddr;
var
  pszText: WideString;
  pszUTF8: AnsiString;
  dwErr  : DWORD;
  phe    : PHostEnt;
  addr   : PaPinAddr;
  ws     : TWSAData;
  i      : Integer;
begin
  //

  Result := 0;

  //

  SetThreadPriority(hThread, THREADMVPN_PRIORITY_BELOW_NORMAL);

  //

  dwErr := WSASStartup(MAKEWORD(1, 0), ws);
  if (dwErr = NOERROR) then
  try
    pszUTF8 := WideStringToAnsi(pszServ, CP_ACP);
    phe := GetHostByName(@pszUTF8[1]);
    if (phe <> nil) then
    begin
      addr := PaPinAddr(phe^.h_addr_list);
      i := 0;
      SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_RESETCONTENT, 0, 0);
      SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_ADDSTRING, 0,
        Integer(@pszServ[1]));
      while (addr^[I] <> nil) do
      begin
        pszUTF8 := inet_ntoa(addr[I]^);
        pszText := AnsiStringToWide(pszUTF8, CP_ACP);
        SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_ADDSTRING, 0,
          Integer(@pszText[1]));
        pszText := Format(LoadStrInst(hInstance, RC_STRING_IPOST), [pszText]);
        SendMessage(GetDlgItem(hApp[3], IDC_STATIC_ADDRESS), WM_SETTEXT, 0,
          Integer(@pszText[1]));
        Inc(i);
        Sleep(35);
      end;
      SendMessage(GetDlgItem(hApp[1], IDC_COMBO_SERVER), CB_SETCURSEL, 0, 0);
    end;
  end;
end;

```

```

    finally
        WSACleanup;
    end;

    //

    SendMessage(hApp[3], WM_DESTROY, 0, 0);

end;

//

function ScanDlgProc_OnWmInitDialog(hWnd: HWND; uMsg: UINT; wParam: WPARAM;
lParam: LPARAM): LRESULT;
var
    pszText : WideString;
    ThreadID: LongWord;
    himl     : HIMAGELIST;
begin
    //

    hApp[3] := hWnd;

    //

    SetCenterDialogPos(hApp[3], hApp[1], TRUE);

    //

    pszText := Format(LoadStrInst(hInstance, RC_STRING_IPSERVER), [pszServ]);
    SendMessage(GetDlgItem(hApp[3], IDC_STATIC_ADDRESS), WM_SETTEXT, 0,
        Integer(@pszText[1]));

    //

    CreateAnimateStatic(GetDlgItem(hApp[3], IDC_STATIC_ANIMATE));
    himl := ImageList_LoadImage(hInstance, MAKEINTRESOURCEW(RC_BITMAP_WAITING),
        GetSystemMetrics(SM_CXSMICON), 0, CLR_DEFAULT, IMAGE_BITMAP, LR_DEFAULTCOLOR
        or LR_CREATEDIBSECTION);
    if (himl <> 0) then
        SendMessage(GetDlgItem(hApp[3], IDC_STATIC_ANIMATE), SS_SETIMAGELIST, himl,
            0);

    //

    hThread := CreateThread(nil, 0, @ThreadCallback, nil, 0, ThreadID);
    if (hThread <> 0) then
        begin
            CloseHandle(hThread);
            hThread := 0;
        end;

    //

    Result := 0;

end;

//

function ScanDlgProc_OnWmDestroy(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
LPARAM): LRESULT;
var
    himl: HIMAGELIST;
begin
    //

```

```
if (hThread <> 0) then
  begin
    CloseHandle(hThread);
    hThread := 0;
  end;

//

himl := SendMessage(GetDlgItem(hApp[3], IDC_STATIC_ANIMATE), SS_GETIMAGELIST,
  0, 0);
if (himl <> 0) then
  ImageList_Destroy(himl);
RemoveAnimateStatic(GetDlgItem(hApp[3], IDC_STATIC_ANIMATE));

//

EndDialog(hApp[3], wParam);

//

Result := 0;

end;

//

function ScanDlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
  BOOL; stdcall;
begin
  case uMsg of
    //
    WM_INITDIALOG:
      begin
        Result := BOOL(ScanDlgProc_OnWmInitDialog(hWnd, uMsg, wParam, lParam));
      end;
    //
    WM_DESTROY:
      begin
        Result := BOOL(ScanDlgProc_OnWmDestroy(hWnd, uMsg, wParam, lParam));
      end;
  else
    Result := FALSE;
  end;

end;

end.
```

DMVPN_FinsWind.pas - DMVPN-з'єднання

```

unit DMVPN_FinsWind;

interface

uses
  Windows, Messages, CommCtrl, DMVPN_Windows, DMVPN_Controls, DMVPN_FileInfo,
  DMVPN_SysUtils,
  DMVPN_MyMsgBox, DMVPN_Ole2, DMVPN_Active, DMVPN_Shlobj, DMVPN_RasApi,
  DMVPN_Resources;

function FinsDlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
  BOOL; stdcall;

implementation

//

function FinsDlgProc_OnWmInitDialog(hWnd: HWND; uMsg: UINT; wParam: WPARAM;
  lParam: LPARAM): LRESULT;
var
  bldfnt: HFONT;
begin
  //

  hApp[2] := hWnd;

  //

  bldfnt := HFONT(SendMessage(GetDlgItem(hApp[0], IDC_STATIC_WELCOME),
    WM_GETFONT, 0, 0));

  if (bldfnt <> 0) then
    SendMessage(GetDlgItem(hApp[2], IDC_STATIC_FINISH), WM_SETFONT,
      Integer(bldfnt), Integer(TRUE));

  //

  Result := 0;
end;

//

function FinsDlgProc_OnWmNotify(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam:
  LPARAM): LRESULT;
var
  pnmh : PNMHDR;
  dwRes : DWORD;
  pszText: WideString;
//

function GetNextItemID(pidl: PItemIDList): PItemIDList;
var
  cb: DWORD;
begin
  Result := nil;
  if (pidl = nil) then
    Exit;
  cb := pidl.mkid.cb;
  if (cb = 0) then
    Exit;
  pidl := PItemIDList(Cardinal(pidl) + cb);
  if (pidl.mkid.cb <> 0) then

```

```

    Result := pidl;
end;

//

function GetPIDSSize(pidl: PItemIDList): DWORD;
begin
    Result := 0;
    if (pidl <> nil) then
    begin
        Result := SizeOf(pidl.mkid.cb);
        while (pidl <> nil) do
        begin
            Inc(Result, pidl.mkid.cb);
            pidl := GetNextItemID(pidl);
        end;
    end;
end;

//

function IsDesktopFolder(pidl: PItemIDList): Boolean;
begin
    if Assigned(pidl) then
        Result := (pidl.mkid.cb = 0)
    else
        Result := FALSE;
end;

//

function ConcatPIDL(destpidl, srcpidl: PItemIDList): PItemIDList;
var
    cb1: DWORD;
    cb2: DWORD;
    pmc: IMalloc;
    hr : HRESULT;
begin
    Result := nil;
    hr := SHGetMalloc(pmc);
    if SUCCEEDED(hr) then
    begin
        cb1 := 0;
        cb2 := 0;
        if Assigned(destpidl) then
        begin
            if not IsDesktopFolder(destpidl) then
                cb1 := GetPIDSSize(destpidl) - SizeOf(destpidl^.mkid.cb);
        end;
        if Assigned(srcpidl) then
            cb2 := GetPIDSSize(srcpidl);
        Result := pmc.Alloc(cb1 + cb2);
        if Assigned(Result) then
        begin
            if Assigned(destpidl) then
                CopyMemory(Result, destpidl, cb1);
            if Assigned(srcpidl) then
                CopyMemory(Pointer(DWORD(Result) + cb1), srcpidl, cb2);
        end;
        pmc := nil;
    end;
end;

//

procedure CreateShellDMVPNLink(pszEntry: WideString);
var
    pMalloc      : IMalloc;
    Desktop      : IShellFolder;

```



```

        pMalloc.Free(pidlDesktop); // папка версій
    end;
finally
    Desktop := nil; // версії простору імен корню директорій
end;
finally
    pMalloc := nil; //
end;
finally
    CoUninitialize;
end;
end;

//

function CreateRasDMVPNConnection(szEntryName, szPhoneName, szUserName,
szPassword: WideString): HRESULT;
var
    osv1      : TOSVersionInfo;
    rEntry    : RASENTRYW;
    rDial     : RASDIALPARAMSW;
    lpCred    : RASCREDENTIALSW;
    dwSize    : Integer;
    EntrySize: Integer;
    InfoSize  : Integer;
    dwFlags   : DWORD;
    dwFlags2  : DWORD;
    dwRes     : DWORD;
begin
    // заповнюємо структуру RASENTRY і довідаємося потрібний розмір для
коректного виклику
    // функції RasSetEntryProperties
    dwSize := SizeOf(RASENTRYW);
    RasGetEntryProperties(nil, nil, nil, EntrySize, nil, InfoSize);
    if (EntrySize < dwSize) then
        dwSize := EntrySize;

    // Задаємо прапоріві параметри DMVPN з'єднання
    dwFlags :=

        // Вкладка 'Параметри', прапор 'Запитувати ім'я, пароль, сертифікат і
т.д.', вимк
        RASEO_PreviewUserPw or

        // Вкладка 'Загальні', прапор 'При підключенні вивести значок в області
повідомлень', вимк
        RASEO_ModemLights or

        // Вкладка 'Загальні', прапор 'Відобразити хід підключення', вимк
        RASEO_ShowDialingProgress or

        // Використовувати основний шлюз
        RASEO_RemoteDefaultGateway or

        // Зашифрований пароль буде використовуватися при перевірці дійсності із
сервером
        RASEO_RequireEncryptedPw or

        // Використовувати автоматично логін, пароль і домен з Windows
        RASEO_RequireDataEncryption or

        // Пароль буде зашифрований за схемою Microsoft
        RASEO_RequireMsEncryptedPw;
    dwFlags2 :=

        // Вкладка 'Параметри', прапор 'Погоджувати багатоканальне підключення для
одноканальних', вимк

```

```

RASEO2_DontNegotiateMultilink or
// Вкладка 'Параметри', прапор 'Передзвонити при розриві зв'язку', вмик
RASEO2_ReconnectIfDropped;

// Заповнюємо структуру RASENTRY
ZeroMemory(&rEntry, SizeOf(RASENTRYW));
rEntry.dwSize := dwSize;
rEntry.dwfOptions := dwFlags;

// Тип використовуваного протоколу = TCP/IP
rEntry.dwfNetProtocols := RASNP_Ip;

// Тип використовуваного протоколу сервера віддаленого доступу = Point-to-
Point Protocol (PPP)
rEntry.dwFramingProtocol := RASFP_Ppp;

// Тип створюваного підключення - Віртуальна приватна мережа (DMVPN)
rEntry.dwType := RASET_DMVPN;

// Значення списку, що випадає, 'Тип DMVPN' = 'Автоматично'
// Викликається спочатку тільки PPTP, якщо ж спроба закінчується невдачею,
то викликається L2TP
rEntry.dwDMVPNStrategy := VS_Default;
rEntry.dwfOptions2 := dwFlags2;

// Вкладка 'Безпека', прапор 'Потрібне шифрування даних', вмик
// Діалог 'Додаткові параметри безпеки', список 'Шифрування даних' =
'обов'язкове'
// Тип шифрування даних при підключенні = Шифрування не використовується
rEntry.dwEncryptionType := ET_None;

// Використовуємо з'єднання пристроїв з безліччю «підходів»
rEntry.dwDialMode := RASEDM_DialAll;

// Вкладка 'Параметри', 'Число повторень набору номера' = 3
rEntry.dwRedialCount := 3;

// Вкладка 'Параметри', 'Інтервал між повтореннями' = 60 секунд
rEntry.dwRedialPause := 60;
lstrcpy(rEntry.szLocalPhoneNumber, @szPhoneName[1]);
lstrcpy(rEntry.szDeviceType, RASDT_DMVPN);

// Створюємо нове підключення DMVPN з потрібними параметрами
dwRes := RasSetEntryProperties(nil, @szEntryName[1], &rEntry, dwSize, nil,
0);
case dwRes of
  ERROR_SUCCESS:
    Begin
      // виконуємо перевірку версії ОС. починаючи з ОС Win XP і старше,
логіні і
      // пароль (фактично це нам і потрібно) можна змінити функцією
      // RasSetCredentials, а в попередніх ОС можна за допомогою функції
      // RasSetEntryDialParams. в Win XP ще можна змінити пароль функцією
      // RasSetEntryDialParams, а от уже в Win Vista і старше не вийде.

      ZeroMemory(&osvi, SizeOf(TOSVersionInfo));
      osvi.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
      DMVPN_Windows.GetVersionEx(osvi);

      if ((osvi.dwPlatformId = VER_PLATFORM_WIN32_NT) and
          (osvi.dwMajorVersion >= 5) and (osvi.dwMinorVersion >= 1)) then
        Begin

          // Заповнюємо структуру RASCREDENTIALS
          ZeroMemory(&lpCred, SizeOf(RASCREDENTIALSW));
          lpCred.dwMask := RASCM_UserName or RASCM_Password;
          lpCred.dwSize := SizeOf(RASCREDENTIALSW);
          lstrcpy(lpCred.szUserName, @szUserName[1]);

```

```

        lstrcpy(lpCred.szPassword, @szPassword[1]);
        // Змінюємо логін і пароль створеного підключення
        dwRes := RasSetCredentials(nil, @szEntryName[1], lpCred, FALSE);
    end
else
    begin

        // Заповнюємо структуру RASDIALPARAMS
        ZeroMemory(@rDial, SizeOf(RASDIALPARAMSW));
        rDial.dwSize := SizeOf(RASDIALPARAMSW);
        lstrcpy(rDial.szEntryName, @szEntryName[1]);
        lstrcpy(rDial.szUserName, @szUserName[1]);
        lstrcpy(rDial.szPassword, @szPassword[1]);

        // Змінюємо логін і пароль створеного підключення
        dwRes := RasSetEntryDialParams(nil, @rDial, FALSE);
    end;
end;
Result := dwRes;
end;

begin
    //

    pnmh := PNMHdr(lParam);

    case pnmh.code of

        //

        PSN_SETACTIVE:
            begin

                pszText := Format(
                    LoadStrInst(hInstance, RC_STRING_DMVPNINFO),
                    [
                        Edit_GetText(GetDlgItem(hApp[1], IDC_STATIC_ENTRY)),
                        Edit_GetText(GetDlgItem(hApp[1], IDC_COMBO_SERVER)),
                        Edit_GetText(GetDlgItem(hApp[1], IDC_STATIC_USER))
                    ]
                );

                SendMessage(GetDlgItem(hApp[2], IDC_STATIC_DMVPNINFO), WM_SETTEXT, 0,
                    Integer(@pszText[1]));

                SendMessage(GetParent(hApp[2]), PSM_SETWIZBUTTONS, 0,
                    Integer(PSWIZB_BACK or PSWIZB_FINISH));
            end;

        //

        PSN_QUERYCANCEL:
            begin

                dwRes := ExtMessageBox(
                    GetParent(hApp[2]),
                    MAKEINTRESOURCEW(LoadStrInst(hInstance, RC_STRING_QCANCEL)),
                    MAKEINTRESOURCEW(exeInfo.pszProductName),
                    MB_YESNO or MB_ICONASTERISK
                );

                SetWindowLong(hApp[2], DWL_MSGRESULT, Integer(dwRes = IDNO));
            end;

        //
    end;
end;

```

```

PSN_WIZFINISH:
begin

    pszText := Edit_GetText(GetDlgItem(hApp[1], IDC_STATIC_ENTRY));

    dwRes := CreateRasDMVPNConnection(
        pszText,
        Edit_GetText(GetDlgItem(hApp[1], IDC_COMBO_SERVER)),
        Edit_GetText(GetDlgItem(hApp[1], IDC_STATIC_USER)),
        Edit_GetText(GetDlgItem(hApp[1], IDC_STATIC_PASSW))
    );

    if (dwRes = ERROR_SUCCESS) then
    begin
        dwRes := SendMessage(GetDlgItem(hApp[2], IDC_CHECK_SHORTCUT),
            BM_GETCHECK, 0, 0);
        if (dwRes = BST_CHECKED) then
            CreateShellDMVPNLink(pszText);
    end;

end;

end;

//

Result := 1;

end;

//

function FinsDlgProc(hWnd: HWND; uMsg: UINT; wParam: WPARAM; lParam: LPARAM):
    BOOL; stdcall;
begin
    case uMsg of
        //

        WM_INITDIALOG:
        begin
            Result := BOOL(FinsDlgProc_OnWmInitDialog(hWnd, uMsg, wParam, lParam));
        end;

        //

        WM_NOTIFY:
        begin
            Result := BOOL(FinsDlgProc_OnWmNotify(hWnd, uMsg, wParam, lParam));
        end;

        else
            Result := FALSE;
        end;

    end;

end.

```