ВСТУП

В умовах роботи ринку за ці роки можна визначити стрімке зростання великої кількості ІТ підприємств різного рівня. Є читки чинники для цього, зокрема такі як автоматизація безлічі різних процесів та швидка і зручна робота з великими обсягами даних.

Наразі, особливо гостро стоїть питання розробки автоматизованих систем для тих підприємств, яким потрібно реалізовувати свої задачі в умовах цифровізації різних галузей.

За допомогою комп'ютерних систем підприємства отримали можливість автоматизувати більшість процесів. Тому використання персональних комп'ютерів, що об’єднані в мережу, в наш час дуже зручно і необхідно. Комп'ютер – незамінний атрибут на підприємстві або в офісі, без якого неможлива автоматизація робочого місця.

Завдяки комп’ютерним системам, автоматизація більшості бізнес-потреб стала можливою для великої кількості підприємств. А це, в свою чергу, означає що використання комп’ютерних систем та мереж в наш час – це дуже зручно і необхідно. Комп’ютер – це головний елемент підприємства або офісу, без котрого, наразі, неможливо автоматизація і керування безліччю процесів.

Приватне підприємство «Ювентус» це передова ІТ компанія, яка являє собою вид аутсорсингово підприємство, що спеціалізується на визначеному виді діяльності й обслуговує велику кількість фірм, що дозволяє їй досконало розбиратися у всіх поточних питаннях і використовувати напрацьований досвід: надійність та стабільність.

Приватне підприємство «Ювентус», насамперед, це аутсорсингова ІТ компанія, яка в свою чергу базується на визначеному виді діяльності та займається розробкою, підтримкою та впровадженням програмного

забезпечення для своїх клієнтів. Великий об’єм замовлень та їх різноманітність дозволила цьому підприємству заробити цінний досвід, який в свою чергу переріс у залізну стабільність і безвідмовну надійність.

Щоб вести бізнес ефективно, безвідмовно та якісно, радше необхідно створення сучасної комп’ютерної системи та мережі, яка в свою чергу має бути новітньою, легко підтриманою та її треба постійно оптимізуватись та вдосконалюватись, для ще кращої продуктивності, безпеки, надійності та стабільності. Авжеж така система має дозволяти безперебійно та легко розробляти програмне забезпечення, здійснювати безпечне зберігання та обмін даних та дозволяти швидкий зв’язок між відділами.

Тому потрібно дослідити та проаналізувати предметну область; зробити проект схеми мережі підприємства; обрати мережеве обладнання; розробити застосунок для моніторингу версій third-party компонентів; впровадити постійний моніторинг систем та мережі.

Мета роботи і завдання дослідження

Удосконалення комп’ютерної системи та мережі підприємства.

Визначити вузли мережі, на які найсильніше впливає інформаційне перевантаження мережі.

Визначити умови, та параметри властивостей технічних пристроїв мережі за яких може виникнути втрата продуктивності.

Розробити рекомендації що до модернізації мережі для підвищенні її стійкості до перевантажень.

Розробити застосунок для відслідковування версій third-party компонентів на вузлах.

Впровадити використання моніторингу комп’ютерної системи та мережі.

Об’єкт дослідження

Комп’ютерна система та мережа ІТ компанії «Ювентус», що лежить

в основі забезпечення роботи усіх підрозділів цього підприємства.

Предмет і методи дослідження

Структура комп'ютерної системи та мережі, її інформаційні властивості, а також технічний та продуктивний потенціал апаратних засобів стане предметом дослідження цієї робити.

Для виконання завдань дослідження використовуються методи теорії масового обслуговування, розроблена математична модель комп'ютерної мережі як мережі масового обслуговування. Проведено дослідження властивостей мережі з різними параметрами інформаційного середовища та апаратних засобів.

Ідея роботи

Провести дослідження, результатом якого стане виявлення найбільш вразливих до просадки пропускної здатності вузлів у комп'ютерній системі та мережі, які у свою чергу, можуть призводити до виникнення затримок та відмову працездатності мережі та розробити рекомендаційні тези для вирішення знайдених проблем.

Розробити додаток для відслідковування версій third-party компонентів на вузлах.

Практичні результати – заходження шляхів удосконалення, визначення ймовірних недоліків комп'ютерної системи та мережі, у першу чергу завдяки застосуванню наукового підхожу до знаходження рішень поставлених задач. У наданих прикладах використання наукових розробок з моделювання комп'ютерних систем та мереж, добре видно їх потенціал та достовірність.

1 СТАН ПИТАННЯ І ЗАВДАННЯ ДОСЛІДЖЕННЯ

1.1 Загальна характеристика об’єкта дослідження

За останній час цифровізація та автоматизація усіх процесів життєдіяльності займають першу позицію в розвитку країн світу, а особливо в Україні, тому ринок IT компаній та IT галузь в цілому дуже поширилась.

Автоматизація процесів торкнулась майже усіх відомих світових галузей, починаючи від звичайних заводів, які вироблять металеві вироби, закінчуючи складними генетичними дослідженнями та провідними лабораторіями. Так само як і автоматизація, цифровізація певних процесів також відіграє важливу роль,так як наприклад цифровий додаток української розробки «Дія», який дивує своєю здатністю спрощувати бюрократичні процеси в рази та безпечно зберігати дані користувачів і відавати їм потрібні документи у будь-якій точці світу.

Все це не може бути можливим без IT фахівців і відділів у цілому. IT сфера проникла вже у кожну компанію і може бути представлена, як декількома фахівцями на підприємстві, маленьким відділом або суцільна, велика IT компанія, яка впроваджує і розробляє свої рішення для інших підприємств від малого до великого бізнесу.

Підприємство «Ювентус» це яскравий приклад середньої IT компанії, яка розробляю проекти під ключ і виконує замовлення для своїх клієнтів по усьому світу. Такий тип компанії називається аутсорсингом, коли компанія бере різні проекти від різних клієнтів і займається їх розробкою. В такому випадку, формуються команди розробників під певні проекти зі знаннями і використанням певних технологій. Окрім компанії «Ювентус», всесвітньо відомі представники компаній цього типу є EPAM та SoftServe.

1.2 Комп’ютерна система компанії «Ювентус»

Компанія «Ювентус» належить до аутсорсингово типу, а отже має різні проекти, різної компоновки та складності, з різними напрямками діяльності та рівнями безпеки, і авжеж з різними рівнями потреб, що до потужностей.

У компанії підключена та працює система контролю якості програмного коду, що означає присутність умов та правил, що до написання коду, безпечного доступу до даних та їх зберігання, позиціювання усіх вузлів мережі тощо.

Для розробки та подальшого вдосконалення плану мережі для підприємства, необхідно провести певні дослідження і аналіз підрозділів, що будуть поєднані мережею.

Площа підприємства «Ювентус» складає 831 м^2 та є трьох поверховою будівлею.

Кількість ПК складає 102 шт. Кількість серверів – 3 шт.

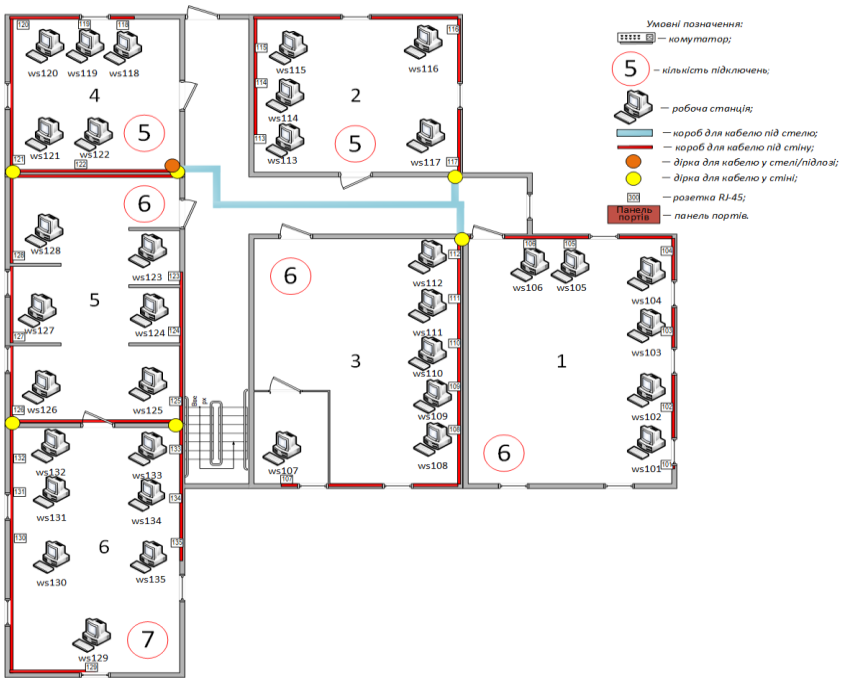


Рисунок 1.1 – Схема першого поверху IT компанії «Ювентус»

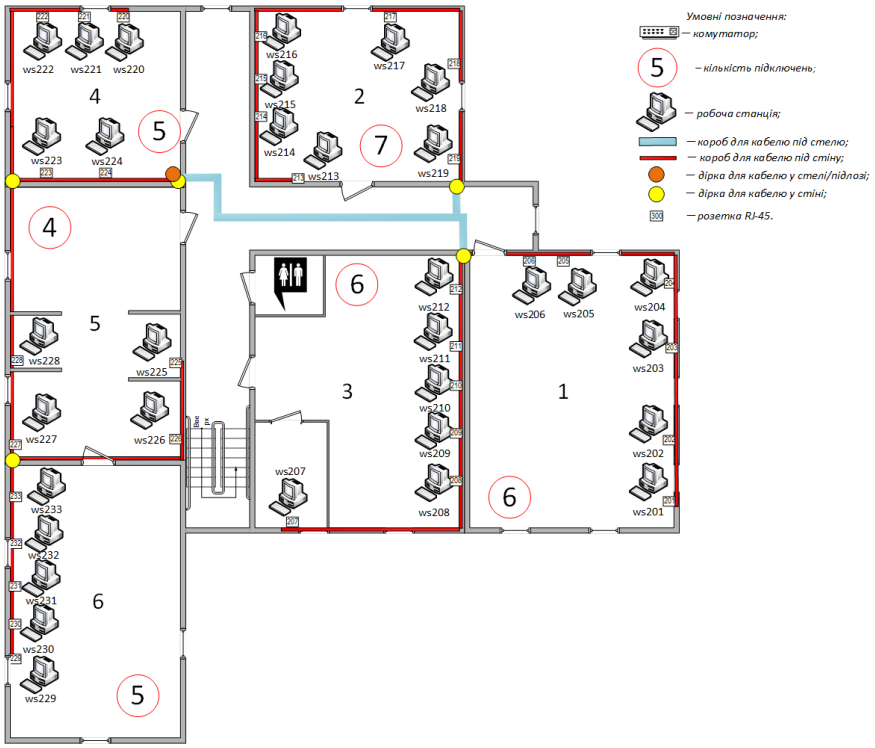


Рисунок 1.2 – Схема другого поверху IT компанії «Ювентус»

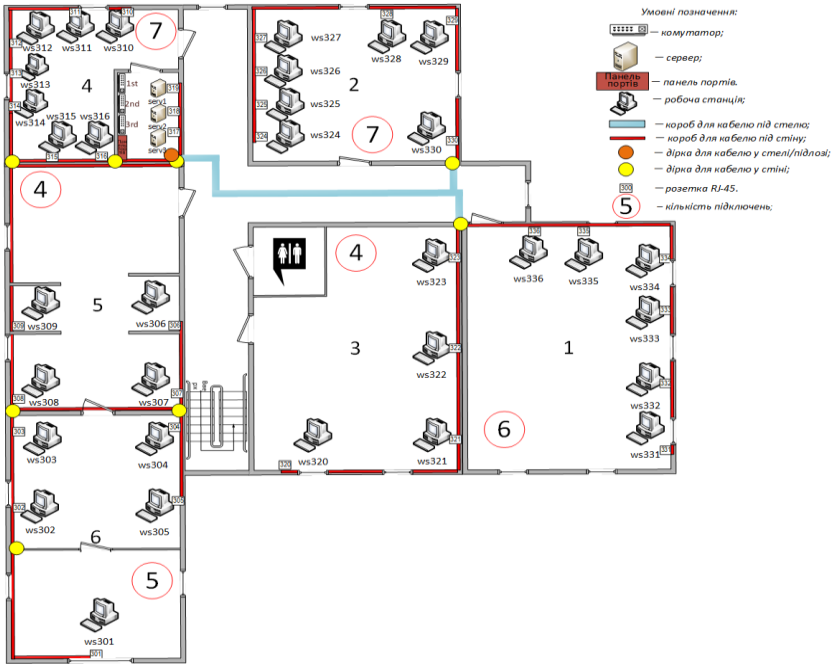


Рисунок 1.3 – Схема третього поверху IT компанії «Ювентус»

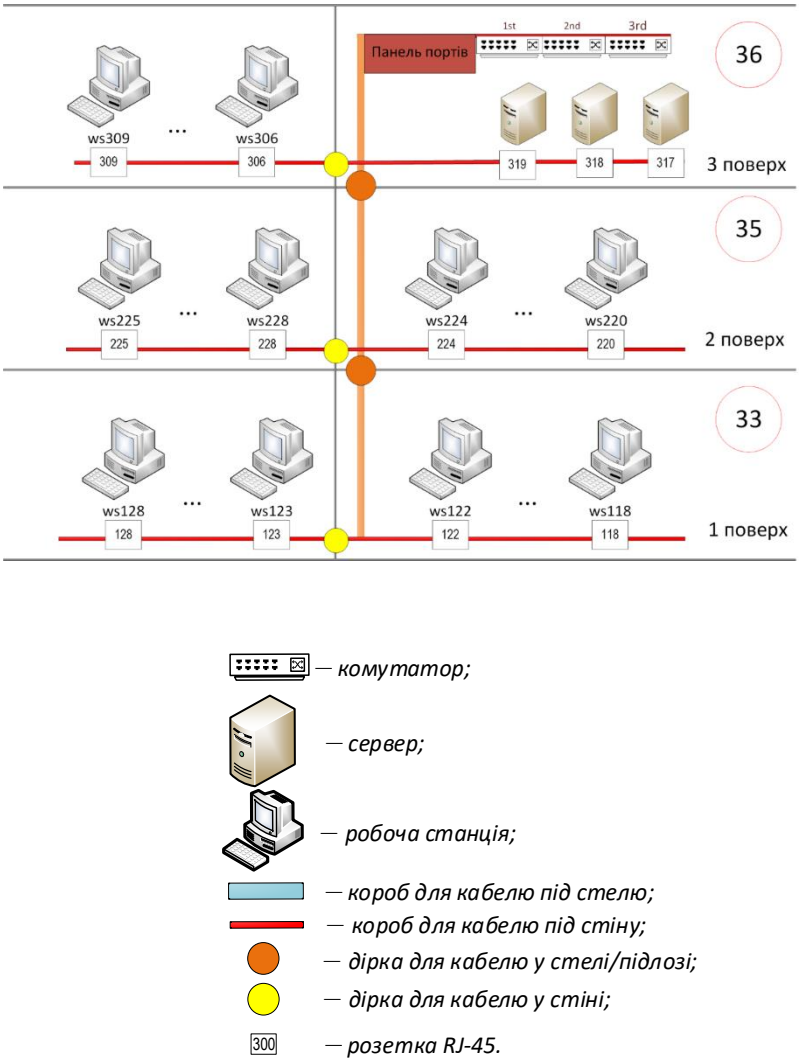


Рисунок 1.4 – Схема вертикального розміщення кабельної системи

1.2.1 Аналіз сучасних методик та технологічних рішень комп'ютерної системи

Підбір технологій для кожного технічного рішення відбувається в кілька етапів.

Етап 1: Визначення технічних вимог до майбутнього проекту. Цьому етапу приділяється достатня кількість часу і сил. Адже саме на цьому етапі криється більшість помилок. Неправильний збір вимог тягне за собою неправильне технічне рішення.

Етап 2: Декомпозиція проекту на логічні/бізнес компоненти. Наприклад, зареєструватися, завантажити файли.

Етап 3: Розробка бізнес-схеми проекту/продукту. Цей етап дає спільне розуміння між клієнтом і нашою командою, що ми розуміємо один одного, а також відповідає на низку запитань клієнта. Де закінчується мобільний додаток і починається Інтернет? Де ми будемо реєструватися та авторизуватися? Куди будуть завантажені файли.

Етап 4: Розробка технічного рішення з розбивкою на компоненти/технології. Цей етап проектування визначає вибір рішення і технології для компонентів бізнес-схеми.

Позиція полягає в тому, що немає універсальних рішень, є добре підібрані природні системи та технології, які органічно вписуються і вирішують завдання без зусиль. Щоб пришвидшити розробку проектів, створюються компоненти, які підходять під часто використовувану бізнес-логіку.

Проектування є першим етапом створення структурованої кабельної системи (СКС), що дозволяє підібрати компоненти кабельної системи, необхідні для реалізації конкретного варіанту структури мережі.

Конструкція структурованої кабельної системи повинна відповідати індивідуальним або комбінованим вимогам щодо передачі голосу, даних та/або відео, які містяться в специфікації замовника.

Існує три варіанти конструкції СКС - стандартна, розширена і комбінована.

Стандартний варіант конструкції SCS є найбільш економічним варіантом конструкції кабелю, який може підтримувати додатки голосу та даних:

- Один порт RJ45 на робочу зону (не відповідає стандартам IS 11801, EN 50173 або TIA-568A)

- один горизонтальний підвід мідної чотирипарної крученої пари в робочу зону.

Удосконалений дизайн SCS — це найбільш економічно ефективний план кабельної розводки, який забезпечує підвищену функціональність і можливості системи:

- два або більше портів RJ45 на робочу зону;

- Окремі чотирипарні кабелі до кожного порту RJ45.

Комбінований варіант конструкції передбачає використання оптоволоконних компонентів разом із кабельною системою, створеною за розширеним варіантом конструкції:

- використання оптичного кабелю в горизонтальній та/або вертикальній підсистемі;

- розетки гібридного (мідь + оптика) типу в робочих зонах.

При проектуванні структурованих кабельних систем дотримуються наступних основних етапів:

- визначення технічних вимог замовника;

- визначення фізичних характеристик будівлі або комплексу будівель, в яких буде встановлено кабельну систему;

- визначення структури телекомунікаційної мережі та типу середовища передачі даних:

- надання замовнику попереднього варіанту схеми проекту та кошторису проекту;

- після узгодження схеми та вартості реалізація остаточного варіанту проекту, включаючи:

1. креслення трас прокладки кабелів, кабельних каналів, розміщення робочих місць;

2. схеми крос-зв'язку та з'єднувального обладнання центральної диспетчерської та поверхових відсіків зв'язку;

3. схеми кросової та сполучної апаратури волоконно-оптичних ліній зв'язку;

4. деталі інформації про зрощення оптичних волокон;

5. узгодження схем прокладання кабелю з інженерними службами будинку.

Правильно спроектована структурована кабельна система повинна мати можливість:

- підтримувати зміни додатків без необхідності масштабних модифікацій;

- пропускати сигнали сучасних пристроїв передачі голосу, даних та відео.

Крім того, проект кабельної системи повинен бути адаптований до можливих унікальних вимог замовника.

Стандарти SCS в офісних будівлях визначають вимоги до кабельної системи, здатної взаємодіяти з активним обладнанням багатьох різних постачальників.

Фізичні сервери підприємства

Головна діяльність підприємства це розробка програмного забезпечення, яке в свою чергу складається з безлічі етапів, таких як проектування проекту, безпосередньо розробка, тестування та інші.

Оскільки бюджет компанії має в своєму розпорядженні певні ресурси, було прийнято рішення запровадити серверні рішення від компанії IBM**.**

IBM Power E1080 (IBM Power 10) сервер IBM Power 10 покоління доступний для замовлення. IBM Power E1080 (m/t: 9080-HEX) — це найкращий сервер десятого покоління середнього та початкового рівня з процесорами Power 10.

Нові процесори доступні за технологією 7 нм і мають на 33% менше енергоспоживання в порівнянні з серверами на базі процесорів IBM Power 9 (порівняно з Power 980).

Серверний вузол (системний вузол) IBM Power E1080 поставляється з 4 процесорами. Ви можете встановити:

- 10 ядерні процесори з частотою 3,65 - 3,90 ГГц;
- 12 ядерних трош9ітшориоооооорр з частотою 3,6 - 4,15 ГГц;
- 15 ядерні процесори з частотою 3,55 - 4,00 ГГц.

Всі процесори мають можливість працювати в режимі 8 апаратних потоків на ядро, що дає можливість отримати до 120 потоків на процесор або 480 потоків на серверний вузол.

Вузол сервера може мати до 16 ТБ оперативної пам’яті DDR4 у 64 слотах DIMM.

До 4 серверних вузлів IBM Power E1080 об'єднані в один сервер, що дає:

- до 240 фізичних ядер;
- до 1920 ниток;
- до 64 ТБ оперативної пам'яті.

У максимальній конфігурації один сервер підтримує:

- до 1000 віртуальних машин (LPAR);
- до 32 слотів PCIe 5 (у системному вузлі)
- до 192 слотів PCIe 3.0 (в окремих полицях PCIe);

Рисунок 1.5 – Сервер IBM Power 1080. Конфігурація з 4 вимірними вузлами

Сервер IBM Power 1080 має надлишковий апаратний гіпервізор IBM PowerVM. Операційні системи IBM AIX, IBM i, Linux (дистрибутиви для IBM Power) можуть бути встановлені:

- Мінімальна підтримувана версія AIX 7.1 TL 5S P5 (у режимі сумісності Power8);
- Мінімальна підтримувана версія IBM i становить 5.4;
- Мінімальна підтримувана версія Red Hat Enterprise Linux 8.4 (у рідному режимі Power10) або 8.2 у режимі сумісності з Power9;
- Мінімум підтримуваних SUSE 15 SP3 (у рідному режимі Power10) або 12 SP5 у режимі сумісності з Power9;

Серверний вузол поставляється з 4 процесорами IBM Power10, займає 5U в стійці. В межах одного фізичного сервера можна об’єднати до 4 серверних вузлів (можливі варіанти доставки з 1, 2, 3 або 4 серверними вузлами). Серверні вузли не можуть працювати незалежно, для роботи потрібен один системний блок керування (SCU).

Кожен процесор Power10 має вісім ліній PowerAXON для з’єднання між процесорами. Кожен процесор в одному вузлі безпосередньо

підключений до кожного іншого процесора зі швидкістю 128 ГБ/с. Сумарна пропускна здатність шини між процесорами становить 128 Гбіт / с, що на 33% вище в порівнянні з Power E980.

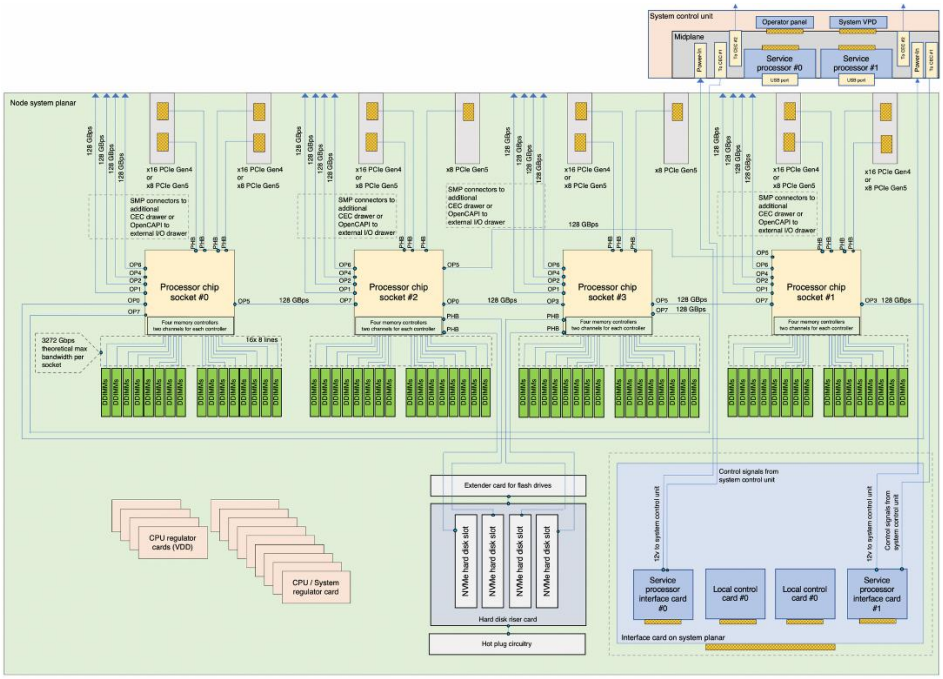


Рисунок 1.6 – IBM Power 1080 system node логістична схема

Ресурсу та можливостей такого сервера буде достатньо на певний час для виконання поставлених підприємством задач. За допомогою HMC/vHMC дуже зручно ділити сервер IBM на віртуальні машини для розробки, тестування, веб-серверу, серверу баз даних та інших.

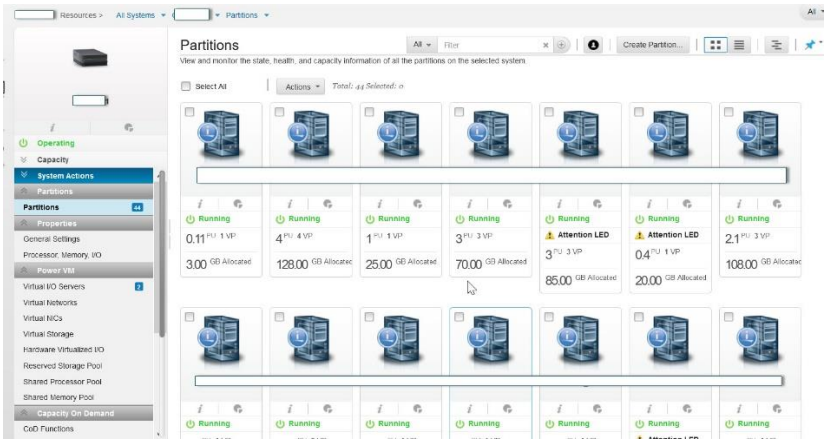


Рисунок 1.7 – інтерфейс Hardware management console (HMC)

Веб-сервер

Термін «веб-сервер» може стосуватися як апаратного, так і програмного забезпечення. Або навіть обидві частини працюють разом.

З точки зору апаратного забезпечення, «веб-сервер» — це комп’ютер, який зберігає файли сайту (документи HTML, стилі CSS, файли JavaScript, зображення тощо) і доставляє їх на пристрій кінцевого користувача (веб-браузер тощо). . .d.). Він підключений до Інтернету та доступний через доменне ім’я, наприклад uventus.com.

З точки зору програмного забезпечення, веб-сервер містить кілька компонентів, які контролюють доступ веб-користувачів до файлів, розміщених на сервері, принаймні на сервері HTTP. HTTP-сервер — це частина програмного забезпечення, яка розуміє URL-адреси (веб-адреси) і HTTP (протокол, який використовує ваш браузер для перегляду веб-сторінок).

На найпростішому рівні, коли браузеру потрібен файл, розміщений на веб-сервері, він запитує його через протокол HTTP. Коли запит досягає потрібного веб-сервера (апаратне забезпечення), HTTP-сервер (програмне забезпечення) приймає запит, знаходить запитуваний документ (якщо ні, повідомляє про помилку 404) і надсилає його назад, також через HTTP.

У якості веб-серверу для потреб компанії було обрано рішення від Apache - The Apache HTTP Server Project. Це рішення є з відкритою ліцензію і має весь набір сучасних функцій, тому це найкращій вибір.

Проект HTTP-сервера Apache — це спільна розробка програмного забезпечення, спрямована на створення надійної реалізації комерційного рівня з функціональними можливостями та вільно доступного вихідного коду HTTP (веб-сервера). Проектом спільно керує група волонтерів, розташованих по всьому світу, які використовують Інтернет і Інтернет для спілкування, планування та розробки сервера та відповідної документації. Цей проект є частиною Apache Software Foundation. Крім того, сотні користувачів додали ідеї, код і документацію до проекту. Цей файл призначений для короткого опису історії HTTP-сервера Apache і визначення багатьох учасників.

Сервер баз даних

Кожна організація має інформацію, яку їй необхідно зберігати та керувати нею для задоволення її потреб. Наприклад, компанії необхідно збирати та підтримувати кадрову документацію для своїх співробітників. Ця інформація має бути доступною для тих, хто її потребує.

Інформаційна система — це формалізована система зберігання та обробки інформації. Інформаційною системою також може бути серія картонних коробок, що містять папки та правила зберігання та пошуку папок. Проте сьогодні більшість компаній використовують бази даних для автоматизації своїх інформаційних систем. База даних — це організований набір даних, який розглядається як єдине ціле. База даних призначена для збору, зберігання та отримання відповідної інформації для використання в програмах баз даних.

Сервери баз даних є ключовими для управління інформацією. Вони зазвичай надійно керують великими обсягами даних у багатокористувацькому середовищі, щоб багато користувачів могли

отримати доступ до тих самих даних одночасно. Він захищає від несанкціонованого доступу, одночасно забезпечуючи ефективне рішення для аварійного відновлення.

Для вирішення цих задач, було обрано SQL сервер від компанії Oracle.

Сервер бази даних Oracle складається з однієї бази даних і одного або кількох екземплярів бази даних (зазвичай їх називають просто екземплярами).

Екземпляри та бази даних настільки тісно пов’язані, що термін база даних Oracle іноді використовується для позначення як екземплярів, так і баз даних. Ось точне значення цих термінів:

База даних — це набір файлів на диску, в яких зберігаються дані користувача. Ці файли даних можуть існувати окремо від примірника бази даних. Починаючи з Oracle Database 20c, база даних конкретно стосується мультитенантної контейнерної бази даних (CDB), підключаємої бази даних (PDB) або файлів даних контейнера програми.

Екземпляр — це набір іменованих структур пам’яті, які керують файлами бази даних. Екземпляр бази даних складається із спільної області пам’яті, яка називається системною глобальною областю (SGA), і набору фонових процесів. Примірники можуть існувати окремо від файлів бази даних.

Корпоративний VPN-сервер

Корпоративний VPN-сервер **–** **ц**е безпечна зашифрована система зв’язку, яка забезпечує безпечну передачу даних для всіх пристроїв корпоративної мережі. І якщо локальна мережа компанії працює в межах однієї будівлі або комплексу будівель, то VPN дає можливість підключатися до корпоративної мережі з будь-якої точки світу. За звичайних обставин ні хакери, ні державні організації, ні конкуренти не

зможуть перехопити інформацію, яка передається по корпоративному каналу.

Головне, що дає VPN компанії та її співробітникам це безпека. Навіть безпечні мережеві з’єднання, які перевіряються брандмауерами та антивірусами, не є повністю безпечними. Зловмисники використовують складні методи для викрадення цінної інформації.

VPN значно підвищує рівень захисту інформації користувача або компанії за рахунок шифрування даних. Чим потужніший алгоритм шифрування використовується, тим вищий ступінь захисту внутрішніх корпоративних документів, інформації користувачів, листування співробітників, комерційної таємниці тощо. Кожній компанії є що захищати. Корпоративні VPN пропонують надійні протоколи шифрування, такі як OpenVPN або IKEv2/IPSec. Важливий нюанс - VPN краще використовувати як додатковий засіб захисту, а не єдиний.

Віддалене підключення та можливість обміну інформацією. Підрозділи однієї компанії можуть бути розкидані по всій країні, якщо не по всьому світу. Як взаємодіяти з далекими один від одного співробітниками цієї компанії? Найкраще підключитися до однієї приватної мережі або хмари, що полегшить доступ до важливої інформації.

VPN-клієнт забезпечує безпечне з'єднання: наскрізне шифрування, передана інформація надійно захищена. Це є обов’язковим у світі, де більшість бездротових точок доступу вразливі до злому.

Зараз налічується близько 400 мільйонів точок доступу, вони працюють в літаках, ресторанах, центральних площах міст, торгових центрах. Кожен раз, коли користувач підключається до Інтернету за допомогою такої точки, він наражає свої дані на небезпеку. Зломщики знають про найпопулярніші точки підключення і намагаються їх контролювати, що дозволяє їм перехоплювати інформацію інших

користувачів, підключених до такої точки. Перебуваючи в зоні дії публічних точок доступу, зловмисники за допомогою спеціальних програм перехоплюють дані користувачів цих точок WiFi. Це можуть бути доступ до соціальних мереж, месенджерів, корпоративних ресурсів, банківських рахунків і т. д. VPN дозволяє уникнути цієї небезпеки, віддалено підключаючись до робочої «хмари» для роботи з файлами, які потрібні в конкретний момент.

Можливість скоротити витрати. Корпоративна мережа VPN надає можливість використовувати IP-телефонію, CRM, інтранет корпоративні ресурси без необхідності розгортання власної мережевої інфраструктури компанії між головним офісом, регіональними офісами та співробітниками, які працюють поза офісом.

Якщо компанія велика, варто виділити відповідний персонал для підтримки інфраструктури VPN. Але це можуть бути звичайні системні адміністратори, і штатне завантаження у випадку з VPN не потрібно, на налаштування і обслуговування віртуальної приватної мережі не потрібно багато часу.

Загалом, VPN — це недорогий і надійний бізнес-інструмент для керування мережевими підключеннями та передачі даних. Безпека, низька вартість і масштабованість – три основні характеристики сучасної корпоративної VPN.

Поштовий сервер

Для спілкування між підрозділами підприємства, а також для офіційного та безпечного зв’язку між іншими підприємствами, клієнтами та структурами було створено поштовий сервер на базі рішення від компанії Google.

Google дає можливість робити персоналізовані електронні адреси в домені компанії, як-от tamara@uventus.com чи petro@uventus.com, створюють у клієнтів відчуття надійності та довіри. Також поштовий

сервер від Google дає можливість робити розсилки від групу на зразок admins@uventus.com.

Рішення від гугл, в свою чергу, дає можливість інтеграції з іншими рішеннями для спілкування безпосередньо у компанії, такі як Google Meet або Google Chat, надсилати запрошення з Календаря, додавати завдання в список, а також використовувати інші можливості. Крім того, завдяки доповненням Google Workspace ви зможете підключити улюблені сторонні додатки на бічній панелі.

Якщо говорити про безпеку, то інженери Google докладають усіх зусиль, щоб захистити користувачів. Їх моделі машинного навчання блокують понад 99,9% спаму, фішингових атак і зловмисного програмного забезпечення.

Хмарні рішення

Хмарні рішення – модель споживання IT-ресурсів, коли користувач отримує до них доступ і платить за фактично спожиту або забронювану кількість – без великих капіталовкладень.

Хмарні рішення поділяються на різні види та типи.

Приватна хмара – побудова власної IT-інфраструктури з подальшим наданням в якості послуги користувачам всередині компанії. Ця модель зазвичай використовується великими підприємствами, для яких критично важливий повний контроль над усією IT-інфраструктурою.

Публічна хмара – оренда віртуальних сервісів, інфраструктури, послуг, ресурсів у провайдера, який їх надає, в тому числі багатьом іншим клієнтам. Приблизно однаково поширена модель споживання для малого, середнього та великого бізнесу.

Гібридна хмара – це рішення, яке об’єднує локальні ресурси та публічну хмару в одну мережу. Залежно від завдань і вимог клієнта ця модель часто буває дуже ефективною.

SaaS – програмне забезпечення як послуга. Програмне забезпечення для реалізації основних бізнес-процесів;

PaaS – платформа як послуга. Віртуальна IT-інфраструктура, що відповідає вимогам інформаційної безпеки;

IaaS – інфраструктура як послуга. Базові елементи для розгортання багатофункціональної віртуальної IT-інфраструктури для бізнес-задач;

DaaS – віртуальний робочий стіл. Створення робочих місць для користувача зі знайомим інтерфейсом і ліцензійним офісним програмним забезпеченням.

Для компанії було обрано хмарне рішення AWS EC2 від компанії Amazon.

Amazon Elastic Compute Cloud (Amazon EC2) забезпечує масштабовану обчислювальну потужність у Amazon Web Services (AWS) Cloud. Використання Amazon EC2 позбавляє вас необхідності інвестувати в апаратне забезпечення, тож ви можете швидше розробляти та розгортати програми. Ви можете використовувати Amazon EC2, щоб запускати стільки чи менше віртуальних серверів, скільки вам потрібно, налаштовувати безпеку та мережу, а також керувати сховищем..

Amazon EC2 надає такі функції:

- Попередньо налаштовані шаблони для екземплярів, відомі як Amazon Machine Images (AMI), які упаковують біти, необхідні для вашого сервера (включаючи операційну систему та додаткове програмне забезпечення).
- Різні конфігурації ЦП, пам’яті, пам’яті та мережевої ємності для ваших екземплярів, відомі як типи екземплярів
- Захист інформації для входу до екземплярів за допомогою пар ключів (AWS зберігає відкритий ключ, а закритий ключ зберігається у безпечному місці)

- Об’єми зберігання для тимчасових даних, які видаляються, коли зупиняєм, переводим у сплячий режим або припиняєте свій екземпляр, відомі як томи сховища екземплярів
- Постійні томи зберігання даних за допомогою Amazon Elastic Block Store (Amazon EBS), відомі як томи Amazon EBS
- Кілька фізичних місць для ресурсів, таких як екземпляри та томи Amazon EBS, відомі як регіони та зони доступності
- Брандмауер, який дає змогу вказувати протоколи, порти та діапазони вихідних IP-адрес, які можуть досягати екземплярів за допомогою груп безпеки
- Статичні адреси IPv4 для динамічних хмарних обчислень, відомі як еластичні IP-адреси
- Метадані, відомі як теги, які можна створити та призначити своїм ресурсам Amazon EC2

Віртуальні мережі, які ви можете створити, які логічна ізольовані від решти хмари AWS і які ви можете підключити до власної мережі, відомі як віртуальні приватні хмари (VPC).

За його допомогою можна отримати потрібно кількість допоміжних та резервних потужностей, якими зручно керувати та доповнювати.

1.2.2 Задачі дослідження

Наразі, маючи передові технічні засоби та рішення можна сформувати технічне завдання.

Продіагностувати та провести дослідження існуючої мережі.

Вдосконалити та підвищити надійність на базі отриманих діагностичних та дослідницьких даних.

Інтегрувати новий сервер IBM у комп’ютерну систему і зробити базові налагодження.

Створити віртуальні машини для потрібних для бізнесу серверів.

Запровадити та налаштувати роботу веб-сервера Apache для робити веб-додатку.

Запровадити та налаштувати роботу сервера бази даних Oracle.

Запровадити та налаштувати роботу VPN-сервера.

Запровадити та налаштувати роботу поштовий сервер Google.

Інтегрувати хмарні рішення AWS до комп’ютерної системи та мережі підприємства.

Провести певні тести у системі моделювання системи та мережі Cisco Paket Tracer або аналогічній до неї.

Packet Tracer — це симулятор маршрутизатора Cisco, який можна використовувати під час навчання, а також для простого моделювання комп’ютерної мережі. Інструмент, створений компанією Cisco Systems, надається для безкоштовного розповсюдження серед викладачів, студентів і випускників, які відвідували або брали участь у Мережевій академії Cisco.

Graphic Network Simulator. Якщо перекладати дослівно - графічний симулятор мережі. Це дозволяє створювати різні топології мережі прямо на вашому комп'ютері. Найчастіше GNS використовується як лабораторний стенд, де можна перевірити ту чи іншу технологію чи схему.

Насправді GNS3 не симулятор, а емулятор. Варто зрозуміти різницю між цими поняттями.

Симулятор імітує поведінку системи та її інтерфейс. Яскравим прикладом є Cisco Packet Tracer. Програмісти цього програмного забезпечення просто створили пристрої зі схожим інтерфейсом і схожими командами.

Dynamips — це комп’ютерна емуляторна програма, написана для емуляції маршрутизаторів Cisco. Він був створений Крістофом Філлом у серпні 2005 року. Dynamips працює на Linux, Mac OS X або Windows і може емулювати апаратне забезпечення платформи маршрутизації серії

Cisco, безпосередньо завантажуючи програмне забезпечення Cisco IOS в емулятор. Dynamips емулює платформи Cisco 1700, 2600, 2691, 3600, 3725, 3745 і 7200.

1.3 Функціональні особливості комп'ютерної системи

Передача та зберігання даних це одні з функцій, які забезпечують роботу будь-якого підприємство, а тим паче націленого на розробку ПЗ.

Надійне зберігання даних – завдання, яке має вирішити кожна організація. Проблеми виникають, коли збільшується обсяг інформації та зростають вимоги до її захисту. Сучасні системи зберігання даних являють собою складні програмно-технічні комплекси, кожна з яких спеціально розроблена під потреби конкретного замовника.

Система зберігання даних (DSS) — це конгломерат програмного забезпечення та спеціалізованого апаратного забезпечення, призначеного для зберігання та передачі великих обсягів інформації. Особливістю системи зберігання є оптимальний розподіл ресурсів при зберіганні інформації на дискових майданчиках.

Надійне зберігання даних і швидкісний доступ до них вимагають організації засобів зберігання як окремої підсистеми комп'ютерних систем. Ця підсистема має бути належним чином розроблена та реалізована, щоб забезпечити можливість відновлення втрачених даних.

Сховище має бути масштабованим, тобто гнучким, стійким до збоїв і аварій. Необхідно забезпечити його відповідність стандартам і вимогам інформаційної та фізичної безпеки.

У тих випадках, коли потрібно зберігати великі обсяги даних, важливо не тільки створити систему зберігання, але й зробити її оптимальною для вирішення конкретних завдань компанії.

Говорячи про технології зберігання даних, неможливо обійти увагою термін RAID. Резервний масив незалежних дисків — це технологія

віртуалізації даних, яка об’єднує кілька дисків у логічний елемент для підвищення продуктивності. Залежно від обраного типу RAID технології зберігання поділяються на два класи:

Використання апаратного RAID є більш дороге і не завжди виправдане рішення, пов'язане з придбанням додаткової комп'ютерної техніки з власною пам'яттю і виділеним процесором. Апаратний RAID необхідний, якщо система має принаймні чотири або більше дисків.

Використання програмного RAID – технологія використовує контролери на материнській платі, які не мають власної пам’яті та виділеного процесора. Вони використовують від 2-5% ресурсів процесора сервера. Не менш надійні, ніж апаратні рішення, що використовуються в невеликих системах.

Запам'ятовуючі пристрої:

- DAS. Диски розміщуються безпосередньо на сервері для додаткового місця з відносно швидким доступом. Найпростіший і дешевий варіант.
- NAS. Сховище, підключене до мережі. Він гнучкий і централізовано керований, але швидкість доступу обмежена швидкістю мережі.
- S.A.N. Сховище, підключене через оптоволоконний кабель. Поєднує в собі всі переваги NAS з високою швидкістю доступу.

Вартість зберігання залежить від масштабу, логічної моделі та обладнання. Створення систем зберігання може тривати від одного місяця до півроку. Важливим фактором, який слід враховувати, є потреба в сервісній підтримці обладнання. Його можна замовити безпосередньо в представництві світового виробника або в місцевій IT-компанії. У другому випадку вартість володіння системами зберігання помітно знизиться.

Якщо ж ми говоримо про передачу даних, що також є невід’ємною функцією підприємства, то тут на допомогу виділені файлові репозиторії та S/FTP-сервери.

Файловий сервер, як правило, є центральним сервером у комп’ютерній мережі, який з’єднує користувачів із мережевою системою зберігання (SAN).

Цей термін може стосуватися як апаратного, так і програмного забезпечення, необхідного для виконання функцій файлового сервера.

Користувачі, отримавши необхідні права доступу до певних файлів в системі мережевого зберігання, можуть відкривати і редагувати їх, а також видаляти файли і папки, як якщо б вони працювали на власному комп'ютері.

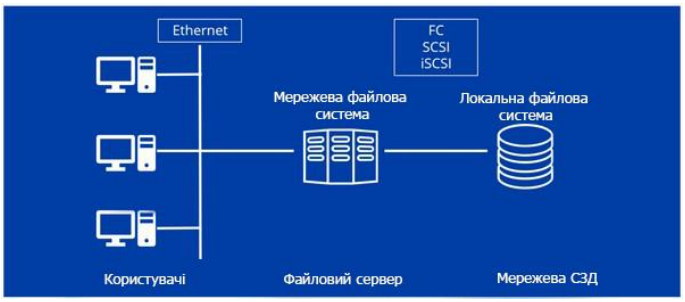


Рисунок 1.8 – Розташування файлового сервера у комп’ютерній мережі

На файловому сервері кожному авторизованому користувачеві надається певний простір для зберігання робочих файлів. Інші користувачі також можуть відкривати, читати та редагувати їх відповідно до своїх прав доступу. Ці права встановлює адміністратор файлового сервера. Він визначає, хто може відкривати та переглядати які файли та в яких папках, а також (якщо дозволено) редагувати, видаляти або додавати нові файли.

Файловий сервер, як правило, є центральним сервером у комп’ютерній мережі, який з’єднує користувачів із мережевою системою зберігання (SAN).

Цей термін може стосуватися як апаратного, так і програмного забезпечення, необхідного для виконання функцій файлового сервера.

Користувачі, отримавши необхідні права доступу до певних файлів в системі мережевого зберігання, можуть відкривати і редагувати їх, а також видаляти файли і папки, як якщо б вони працювали на власному комп'ютері.

На файловому сервері кожному авторизованому користувачеві надається певний простір для зберігання робочих файлів. Інші користувачі також можуть відкривати, читати та редагувати їх відповідно до своїх прав доступу. Ці права встановлює адміністратор файлового сервера. Він визначає, хто може відкривати та переглядати які файли та в яких папках, а також (якщо дозволено) редагувати, видаляти або додавати нові файли.

Будь-яка сучасна операційна система Windows, Linux або macOS може працювати з файловим сервером, але потрібно мати на увазі, що мережеві пристрої повинні бути з ними сумісні.

Слід також враховувати, що файлові сервери часто використовуються не тільки для зберігання та обробки файлів, але і як сховище програм, які доступні користувачам корпоративної мережі, а також резервний сервер.

FTP (протокол передачі файлів) – це протокол для передачі файлів по мережі. Це один із основних протоколів Ethernet. З'явилася в 1971 році і спочатку працювала в мережах DARPA. Зараз, як і HTTP, передача файлів базується на моделі, що складається з набору протоколів TCP/IP (протокол керування передачею/протокол Інтернету). Визначено в RFC 959.

Протокол визначає наступне:

- Як перевірити наявність помилок

- Метод кондиціонування даних (якщо використовується кондиціонування)
- Як пристрій-відправник вказує, що він закінчив повідомлення
- Як приймаючий пристрій вказує, що він отримав повідомлення

Передача даних може здійснюватися в будь-якому з трьох режимів:

- Потокова передача – дані надсилаються безперервним потоком, звільняючи FTP від будь-якої обробки. Натомість уся обробка виконується TCP. Індикатор кінця файлу не потрібен, за винятком розділення даних на записи.
- Блоковий режим – FTP розбиває дані на кілька блоків (блок заголовка, кількість байтів, поле даних), а потім передає їх в TCP.
- Режим стиснення – дані стискаються за єдиним алгоритмом (зазвичай кодування по довжині).

FTP-сервер – це сервер, який забезпечує можливість використання протоколу передачі файлів. Він має певні особливості, які відрізняють його від звичайних веб-серверів:

- Необхідна автентифікація користувача
- Усі операції виконуються в поточному сеансі
- Можливість виконання різних дій з файловою системою
- Для кожного підключення використовується окремий канал

FTP-клієнт — це програма, яка дозволяє підключатися до віддаленого сервера по FTP, а також виконувати на ньому необхідні дії з елементами файлової системи. Клієнт цілком може бути браузером, в адресному рядку якого слід ввести адресу, яка є шляхом до певного каталогу або файлу на віддаленому сервері, відповідно до загальної блок-схеми URL, наприклад ftp://user:pass@address:port/directory/file.

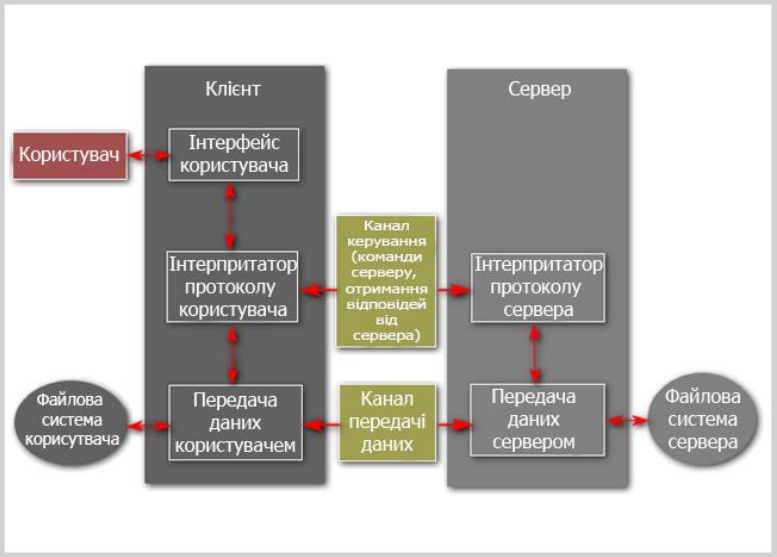


Рисунок 1.9 – Взаємодія «клієнт-сервер» за FTP-з’єднанням

SFTP (протокол безпечної передачі файлів) — це протокол передачі файлів прикладного рівня, який працює поверх безпечного каналу. Не плутати з (Simple File Transfer Protocol), який має таку саму абревіатуру. Якщо FTPS є лише розширенням FTP, то SFTP є окремим і не пов’язаним протоколом, який використовує SSH (Secure Shell) як основу.

SSH – це мережевий протокол, який дозволяє віддалено керувати операційною системою та тунелювати TCP-з’єднання (наприклад, для передачі файлів). Подібний за функціями до протоколів Telnet і rlogin, але на відміну від них шифрує весь трафік, включаючи передані паролі. SSH дозволяє вибрати різні алгоритми шифрування. Клієнти SSH і сервери SSH доступні для більшості мережевих операційних систем.

SSH дозволяє безпечно передавати майже будь-який інший мережевий протокол у незахищеному середовищі. Таким чином, ви можете не тільки віддалено працювати на комп'ютері через командну оболонку, але і передавати аудіопотік або відео по зашифрованому каналу (наприклад, з веб-камери). SSH також може використовувати стиснення

переданих даних для подальшого шифрування, що зручно, наприклад, для віддаленого запуску клієнтів X WindowSystem.

SSH вимагає SSH-сервер і SSH-клієнт. Сервер прослуховує підключення від клієнтських машин і, коли з'єднання встановлено, виконує аутентифікацію, після чого починає обслуговувати клієнта. Клієнт використовується для входу на віддалену машину та виконання команд.

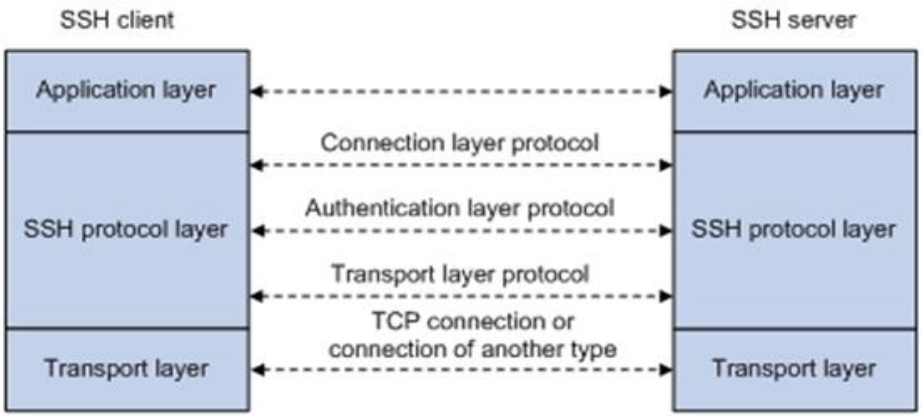


Рисунок 1.10 – Зв’язок між ssh-client та ssh-server

1.4 Сучасні світові стандарти забезпечення бізнес-процесів

Розробка програмного коду складається з великої кількості різних етапів, як інколи тісно, а інколи не дуже пов’язані між собою. Але те що їх точно усіх об’єднує – це потреба в автоматизації, яка зменшить кількість людей, яким потрібно бути задіяним до певних інженерів, які розробляють та підтримують автоматизацію.

Розробка, контроль версій, збірка коду у версії, зберігання коду пакетами у репозиторіях, доставка цього коду у потрібні місця на внутрішні середовища компанії або середовища клієнтів, розгортка, тестування тощо – це і багато інших процесів задіяно у сучасних IT компаніях, які займають створюванням програм. Для кожного з цих

процесів, а буває що і для декількох з них є певна утіліта, яка автоматизує процес, або ж інженер пиши певні сценарії, скрипти, за допомогою певної мови програмування, що б не робити однакові дії багато разів.

DevOps – це скорочення від Development Operations, і насправді це не назва професії. Це культура, методика, якщо завгодно. DevOps-рух виник у 2008 році і був покликаний вирішити проблеми, що накопичилися. Дуже багато компаній бачили проблему у взаємодіях команд розробки та експлуатації.

Розробники вважали, що якщо код запустився у них локально, то немає проблем – можна запускати у продакшен. Якщо все ж таки проблеми виникали, то з боку команди експлуатації звучало: «Та це проблеми з кодом, нехай розробники розбираються». Через такий підхід релізи продуктів постійно затягувалися і часто страждала якість кінцевого продукту. Сильно накладало відбиток ще й те, що за один реліз викочувалося дуже багато змін і було дуже важко розібратися, що породило проблеми на продакшені.

DevOps був покликаний вирішити ці проблеми. Він мав стати сполучною ланкою між командою розробки та командою експлуатації. У DevOps культурі можна виділити кілька ролей, які дуже добре співвідносяться з професіями:

Build Engineer - людина, яка відповідає за складання коду. Підтягування залежностей, аналіз конфліктів у коді — це все про нього.

Release Engineer відповідає за доставку коду від розробки в продакшн. Яка гілка піде у тестування, який білд потрапить на продакшн, реліз-інженер займається саме цим.

Automation Engineer - інженер з автоматизації. Автоматизує все, що рухається. А що не рухається, рухає і також автоматизує. Автоматичне складання при пуші в гіт, прогін тестів, деплой на staging, деплой у продакшн — це його завдання. Ключова роль у підході DevOps.

Continuous Integration (CI) і Continuous Delivery (CD) — це культура, набір принципів і практик, які дозволяють розробникам частіше та надійніше розгортати зміни програмного забезпечення.

CI/CD — одна з практик DevOps. Це також стосується гнучких методів: автоматизація розгортання дозволяє розробникам зосередитися на дотриманні бізнес-вимог, якості коду та безпеці.

Continuous Integration — це методологія розробки та набір практик, у яких невеликі зміни вносяться до коду з частими комітами. А оскільки більшість сучасних додатків розробляються з використанням різних платформ і інструментів, виникає потреба в механізмі інтеграції та тестуванні внесених змін.

Jenkins — це інструмент автоматизації процесів CI/CD, призначений для моніторингу виконання повторюваних завдань. Це проект з відкритим кодом, який може працювати на різних платформах. Він має кілька вбудованих плагінів для створення конвеєрів CI/CD. Використовуючи сервер Jenkins CI/CD, ви можете автоматизувати різні етапи роботи над додатком.

Сервер Azure DevOps також є інструментом автоматизації CI/CD, чиї можливості охоплюють увесь життєвий цикл програми. Цей проект розроблено корпорацією Майкрософт і робить його доступним для розробників за допомогою системи керування версіями Team Foundation (TFVC) або Git. Він підтримує систему звітності, вимоги та системи управління проектами, інструменти для автоматичного збирання, тестування та випуску проектів.

Існують платформи розробки додатків, які включають механізми DevOps, які можуть підтримувати роботу над додатком протягом усього його життєвого циклу. Однією з таких платформ є OutSystems.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Оцінка пропускної здатності корпоративної мережі

2.1.1 Характеристика пропускної здатності пристроїв мережі

З’єднання «точка-точка» можна придбати або взяти в оренду з різною швидкістю передачі даних за секунду або пропускною здатністю. У цьому розділі розповідається про різні значення пропускної здатності для каналів WAN.

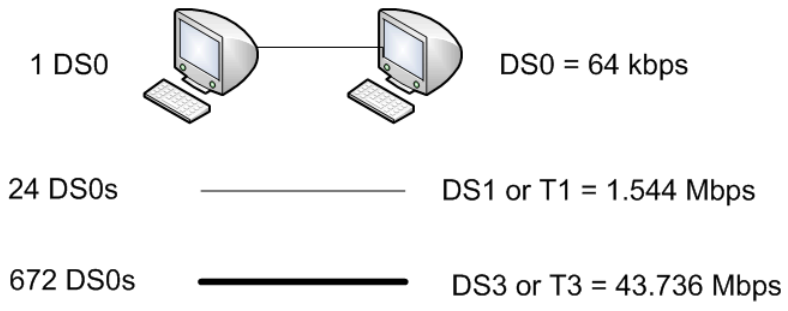


Рисунок 2.1 – Пропускна здатність каналів зв’язку

Пропускна здатність означає швидкість, з якою дані передаються по каналу зв’язку. Технологія, що лежить в основі передачі даних у розподіленій мережі, залежить від доступної смуги пропускання. Що стосується пропускної здатності, існують специфікації для північноамериканських (24-канальна система PCM типу T - T-несуча) і європейських (система PCM типу E - E-несуча) систем зв'язку. Обидві системи засновані на плезіохронній цифровій ієрархії (PDH), яка підтримується в їхніх мережах. Оптичні мережі використовують ієрархію пропускної здатності, яка знову ж таки відрізняється між Північною Америкою та Європою. У Сполучених Штатах оптичний носій (Optical Carrier - OS) визначає значення пропускної здатності, а в Європі значення пропускної здатності визначається відповідно до Синхронної цифрової ієрархії (SDH).

У Північній Америці пропускна здатність зазвичай виражається числом рівня цифрового сигналу (DS) - DSO, DS1 і так далі, що в інженерії зв'язку відноситься до швидкості та формату сигналу. Базова швидкість в ISDN становить 64 кілобіт в секунду (kilobits per second - kbps), що відповідає DSO і означає необхідну пропускну здатність, необхідну для нестиснутого голосового потоку (цифрової телефонної розмови).

Багатопроцесорна архітектура на основі комутаторів

Щоб усунути проблему перевантаження, спричинену архітектурою шини маршрутизаторів другого покоління, наступне покоління пристроїв було розроблено на основі комутаційних систем (комутаційних структур). Це забезпечило достатню пропускну здатність для передачі пакетів між інтерфейсними картами та дозволило збільшити продуктивність на кілька порядків.

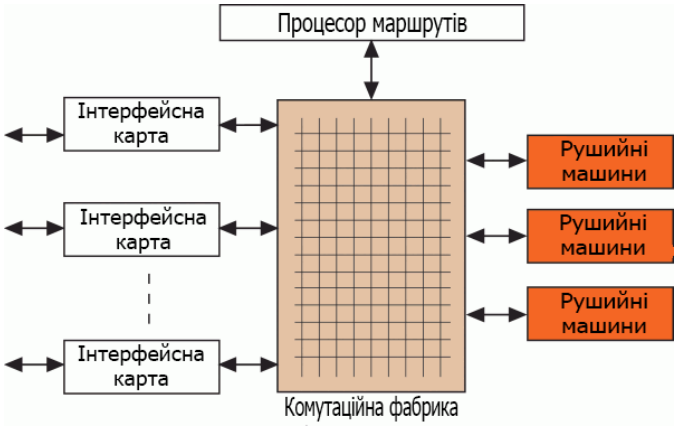


Рисунок 2.2 – Архітектура комутації з кількома машинами пересилання

Приклад маршрутизатора з мультигігабітною архітектурою комутатора показано на рисунку 2.2. Його конструкція передбачає виділені машини для пересилання IP-пакетів із кеш-пам’яттю в кожному з них. Мультигігабітний маршрутизатор (МММ) складається з кількох

інтерфейсних карт (кожна підтримує один або більше мережевих інтерфейсів) і машин пересилання, які підключені через матричний комутатор. Інтерфейсні карти та автомати випередження розміщені на окремих платах. Коли пакет надходить на вхідний інтерфейс, його заголовок видаляється, і він проходить через комутатор до машини пересилання. Решта пакету зберігається у вхідному інтерфейсі. Механізм пересилання аналізує заголовок і визначає, до якого вихідного інтерфейсу потрібно надіслати пакет, змінює заголовок і пересилає його до вхідного інтерфейсу.

Заголовки пакетів, що надходять до механізму пересилання, розміщуються в черзі FIFO (першим увійшов, першим вийшов) для обробки процесором пересилання.

Архітектура кешування маршруту має низку обмежень, пов’язаних із так званими промахами кешу. При випадковому трафіку час на пошук маршрутів у головній таблиці та зміну записів кешу стає неприйнятно довгим, що впливає на продуктивність маршрутизаторів.

Комутаційна архітектура з розподіленими процесорами

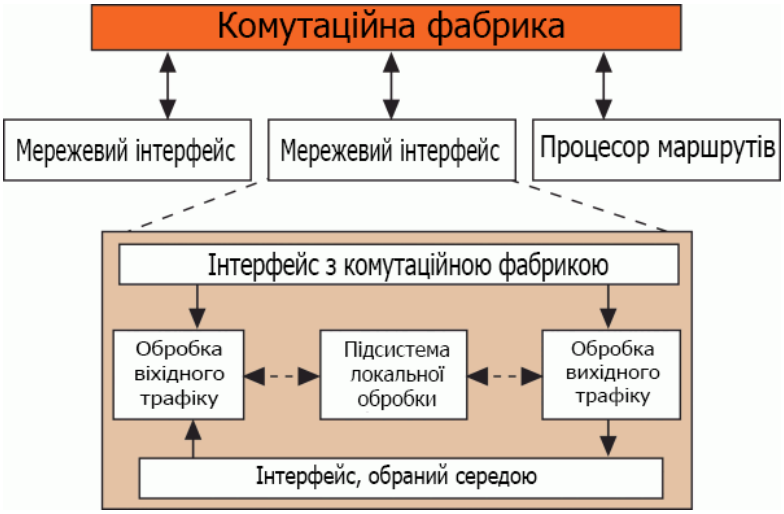


Рисунок 2.3 – Базова архітектура розподіленої комутації (функціональна схема)

Як бачите, є три основні фактори, що перешкоджають підвищенню продуктивності: недостатня обчислювальна потужність, пропускна здатність підсистеми пам’яті та внутрішні шини. Щоб подолати ці проблеми, була запропонована архітектура, заснована на розподілених комутаторах з відповідним чином розробленими мережевими інтерфейсами. Ідея полягала в тому, щоб перекласти на них частину роботи з просування пакетів, таким чином розвантаживши центральний процесор і шину пам’яті. Нижче описана архітектура, у якій кожен мережевий інтерфейс оснащений обчислювальним блоком і буферною пам’яттю. Принципова схема маршрутизатора з архітектурою розподіленої комутації наведена на рисунку 2.3.

Функціональні компоненти (вхід, вихід і локальні блоки) обробляють вхідний і вихідний трафік. Вони реалізують усі функції протоколу (плюс QoS). Щоб забезпечити гарантований рівень QoS, порту може знадобитися класифікувати пакети за попередньо визначеними класами обслуговування.

Media Specific Interface (MCI) виконує всі функції фізичного рівня та, у випадку протоколу IEEE 802, підрівня Media Access Control (MAC). Інтерфейс комутаторної мережі відповідає за підготовку пакетів для передачі через поле комутатора. Ви можете додати до пакета мітку внутрішньої маршрутизації з адресою вихідного порту, пріоритетом QoS і скиданням.

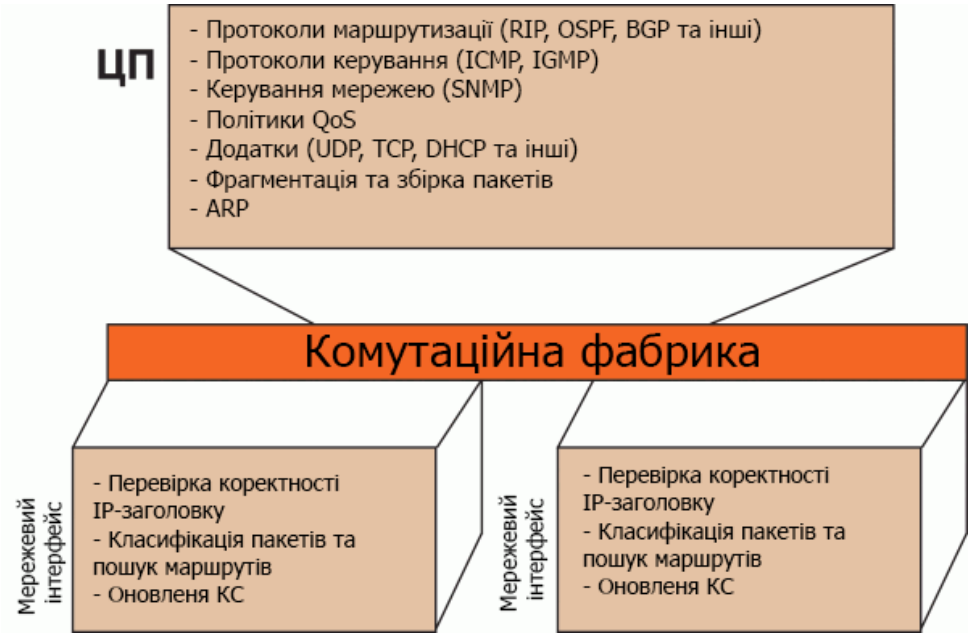


Рисунок 2.4 – Приклад функціонального поділу в розподіленій архітектурі

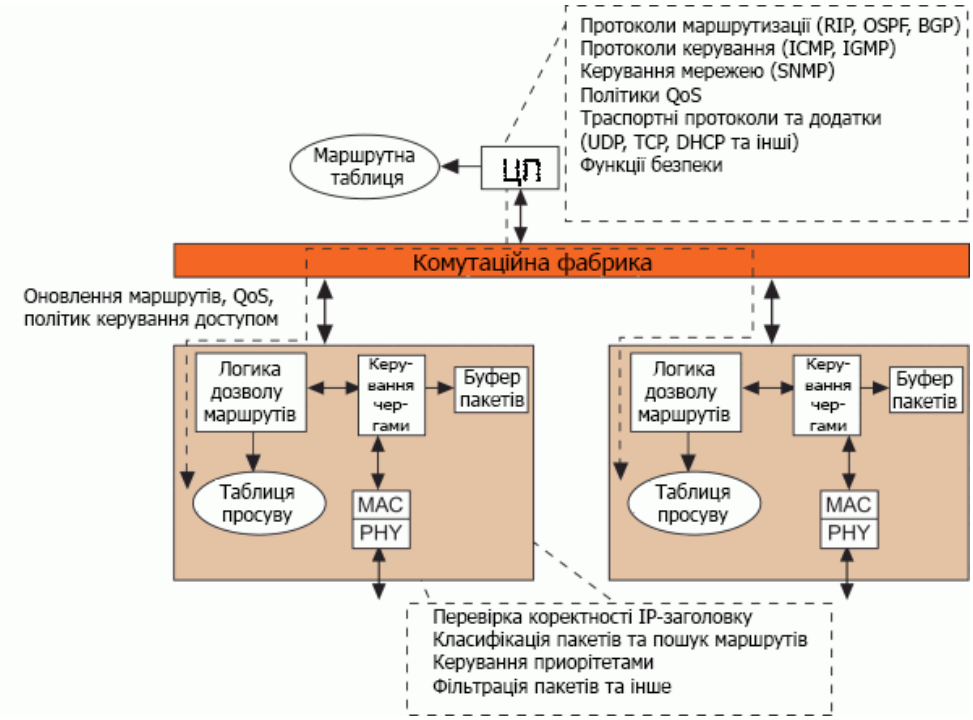


Рисунок 2.5 - Функціональна схема розподіленої архітектури

Обробка IP-протоколу є найбільш ресурсномісткою в маршрутизації пакетів і зазвичай визначає продуктивність. Розподілена архітектура дозволяє розкласти протокол для більшої паралельної обробки, а також відокремити завдання, пов’язані з визначенням маршрутів у мережі, від критичних завдань обробки IP-заголовків. Приклад функціонального поділу в розподіленій архітектурі показано на рисунку 2.4.

Діаграма високого рівня маршрутизатора з розподіленою архітектурою показана на рисунку 2.5. Інтерфейсної карти може містити процесор загального призначення або спеціальну інтегральну схему (ASIC) для виконання обчислень.

Так чи інакше, але одним з основних компонентів, що визначають продуктивність роутера, є фабрика комутації, яку ми зараз і розглянемо.

Типові фабрики з виробництва маршрутизаторів

Архітектура заводської комутації вивчена досить добре. У маршрутизаторах структура комутатора відповідає за передачу пакетів між функціональними блоками, зокрема, направляє пакети користувача від входу до відповідних вихідних модулів. Конструкція комутаційних структур ускладнюється додатковими вимогами, такими як відмовостійкість, можливість багатоадресної передачі та рішення про пріоритетність, коли пакети відкидаються або затримуються.

Насправді конструкція всіх IP-маршрутизаторів базується на поєднанні наступних основних підходів: спільна пам'ять і медіа, буферизований розподілений вихід і просторове розділення (перехресна коса риска). Для заданого вхідного трафіку конструкція комутаційної мережі повинна бути спрямована на максимізацію пропускної здатності та мінімізацію затримок і втрати пакетів.

Спільне середовище

У маршрутизаторі пакети можуть передаватися через спільне середовище, таке як шина, кільце або подвійна шина. Найпростішою

конструкцією комутаційної мережі є спільна шина. Це панель підключення, через яку здійснюються всі інформаційні зв'язки між компонентами. Дані передаються по шині за допомогою мультиплексування з тимчасовим поділом. Однак його продуктивність обмежена як пропускною здатністю, так і витратами на арбітраж для спільного використання цього важливого ресурсу.

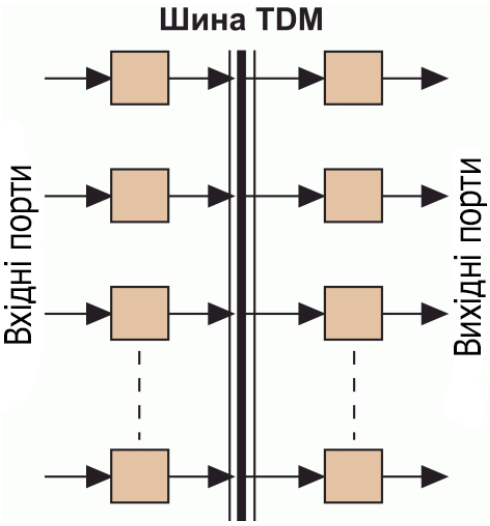


Рисунок 2.6. – Спільна комутаційна структура на основі шини TDM

Цей підхід дозволяє природну реалізацію широкомовних і багатоадресних передач. Однак адресні фільтри та вихідні буфери повинні працювати зі швидкістю спільного носія, яка може бути в N разів вищою за швидкість портів. А оскільки існують фізичні обмеження на швидкість шини, адресні фільтри та вихідні буфери, ця архітектура погано масштабується з точки зору кількості портів і швидкості передачі. Можна збільшити як кількість портів, так і швидкість їх роботи, але так, щоб добуток N•S залишався постійним.

Спільна пам'ять

Типова архітектура спільної пам'яті показана на рисунку 2.7. Вхідні пакети перетворюються з послідовного порту на паралельний, а потім записуються в двопортову пам’ять. Заголовки пакетів із внутрішніми мітками маршрутизації надсилаються до контролера пам’яті, який визначає порядок, у якому вони зчитуються з пам’яті. Потім пакети надсилаються на відповідні вихідні порти, де вони перетворюються назад у послідовний формат.

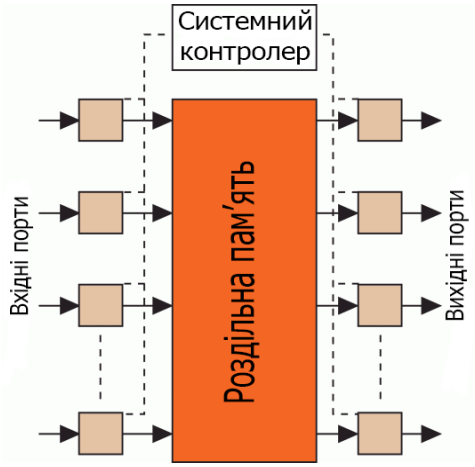


Рисунок 2.7 – Заводські зміни на основі спільної пам'яті

При такому підході вихідні буфери включені в загальний пул, що мінімізує кількість буферів, необхідних для забезпечення бажаної швидкості втрат пакетів. Основна ідея полягає в тому, що загальний буфер є більш вигідним у разі статистичного розділення. Якщо трафік на одному з портів високий, він може зайняти більше буферного простору, якщо спільний пул буферів не повністю зайнятий.

На жаль, ця архітектура має ряд недоліків. Оскільки пакети записуються в пам’ять і зчитуються з пам’яті одночасно, ви повинні мати повну пропускну здатність, тобто записувати та читати пакети кожні 1/N•S

секунди, або в N разів швидше, ніж швидкість портів. Це обмежує масштабованість таких маршрутизаторів. Крім того, центральний контролер пам'яті повинен обробляти пакети з тією ж швидкістю, що і пам'ять. Таке завдання може бути важко виконати у випадку керування кількома класами пріоритетів і складними операціями планування. Маршрутизатори зі спільною пам’яттю мають єдину точку відмови, оскільки додавання іншої комутаційної мережі є надто складним і дорогим.

Розподілені буфери виведення

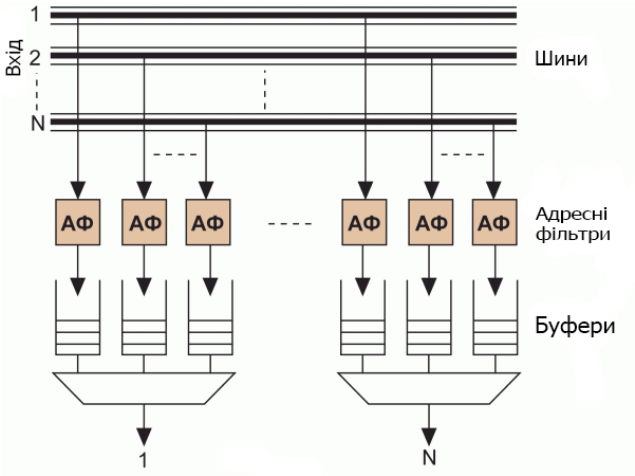


Рисунок 2.8 – Розподілений буферизований вихід

В архітектурі, показаній на рисунку 2.8, існують незалежні шляхи між усіма можливими парами вхідних і вихідних портів (N2). Вхідні широкомовні пакети надсилаються по окремих шинах на всі виходи. Фільтри адреси направляють відповідні пакети до буферів черги на вихідних портах. У класичному варіанті кількість буферів N2.

Пропонований підхід має багато привабливих рис. По-перше, між незалежними шляхами пакетів N2 немає конфліктів, і тому черги виникають лише на виході. Як і в архітектурі спільного середовища,

широкомовні та багатоадресні повідомлення відбуваються тут природно. Фільтри адрес і буфери виводу прості в застосуванні, і, на відміну від проектів спільних носіїв, вони повинні працювати лише на швидкості порту.

Просторове розділення

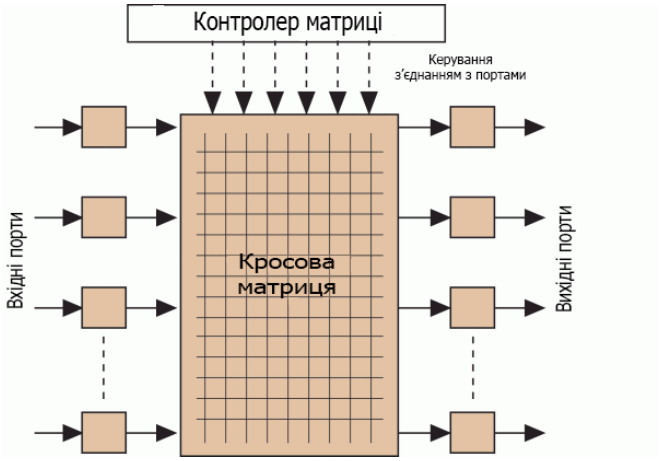


Рисунок 2.9 – Фабрика перемикання на основі крос-матриці

З просторовим розділенням між парами вхідних і вихідних портів може бути один або кілька шляхів. Найпростішим прикладом такої архітектури є перехресна матриця, що показана на рисунку 2.9. У ньому на кожному циклі генератора синхронізації контролер аналізує адресну інформацію вхідних пакетів і встановлює з'єднання між портами по виділеному каналу. Як правило, ці маршрутизатори буферизують вхідні порти, тому швидкість пам’яті не може перевищувати швидкість окремого порту. Буферизація вхідних даних допомагає вирішувати конфлікти, коли кілька пакетів з різних вхідних портів потрібно одночасно направляти на один вихідний порт.

Ця архітектура добре підходить для дуже високої продуктивності і водночас створює низку технічних проблем. Одним з них є блокування

голови хвоста (HOQ), яке відбувається, коли пакети повинні бути спрямовані на різні порти. При цьому, якщо порт першого пакета зайнятий, другий не буде відправлений, навіть якщо його порт вільний. Загальновідомо, що якщо вхідний трафік має рівномірний розподіл, продуктивність досягає лише 58,6%. Крім того, показано, що максимальна пропускна здатність міжкомпонентного маршрутизатора монотонно зменшується зі збільшенням довжини пакета.

Мета розробників маршрутизаторів із цією архітектурою — знайти компроміс між мінімальною необхідною пропускною здатністю та системами CIOB і VOQ.

Щоб до виходу в інтернет в корпоративній мережі є момент з організації цього процесу та створення безпечних заходів навколо його використання.

Існує два основних способи надання користувачам корпоративної мережі доступу до веб-сервісів і ftp-служб: через маршрутизацію (трансляцію) або через проксі-сервер.

У першому випадку доступ забезпечується за IP-адресою комп’ютера, на якому працює співробітник. Така схема може бути повністю реалізована на базі програмного рішення - шлюзу ОС FreeBSD і брандмауера IPFW. Крім того, існують складні спеціалізовані апаратно-програмні шлюзи. Для термінальних робочих станцій організація доступу за IP-адресами технічно неможлива, оскільки всі вони використовують одну IP-адресу термінального сервера.

У другому випадку користувач авторизується за ім'ям доступу (логіном) і паролем, присвоєним співробітнику. Ця опція, зокрема, може бути реалізована за допомогою проксі-сервера SQUID і системи аутентифікації ncsa_auth. Розглянемо типову схему, де SQUID встановлюється на мережевий шлюз: сервер «дивиться» через один інтерфейс в локальну мережу, а інший підключається до Інтернет-каналу.

При такому налаштуванні SQUID для роботи в Інтернеті (через HTTP, FTP і DNS) на машинах в локальній мережі не потрібні NATD і маршрутизація, так як SQUID відправляє всі запити на інтернет-ресурси «самостійно» - з IP адреса зовнішнього інтерфейсу шлюзу. Службу DNS на клієнтських комп'ютерах можна вимкнути, оскільки SQUID сам отримує доступ до DNS.

Як правило, корпоративна мережа використовує електронну пошту, і все одно потребує маршрутизації та NATD на шлюзі для роботи, але для веб-пошти, що працює через HTTP, достатньо проксі SQUID.

Крім самого доступу системний адміністратор повинен також вирішувати проблеми авторизації доступу, обліку трафіку і часу користувача в Інтернеті, забезпечення безпеки локальної мережі підприємства. Також необхідно визначити правила розподілу пропускної здатності Інтернет-каналу між користувачами мережі та правила доступу до Інтернет-ресурсів; можливо, вам знадобиться встановити інші обмеження для користувачів.

Всі ці процедури, в залежності від типу прийнятого доступу (за IP-адресою або через проксі-сервер), мають свої особливості.

Підключити фізичний інтернет-канал безпосередньо до корпоративної мережі - це все одно, що перенести робочі місця вашого підприємства на багатолюдну площу. Як правило, інформація, що циркулює в локальній мережі, є критичною для роботи підприємства, і шкідливий вплив вірусів (наприклад, поштових), атака ззовні або витік даних зсередини можуть повністю порушити її роботу.

Шкідливий вплив вірусів важко недооцінити, але вирішення цієї проблеми на 90% залежить від обізнаності користувачів - чи запустить хтось вірус, прикріплений до листа. Вірусні атаки можуть бути заблоковані та відбиті антивірусними програмами на поштових серверах і

на комп’ютерах користувачів. Головне при цьому – своєчасне оновлення антивірусних баз.

Атаки ззовні, в залежності від того, як організовано з'єднання, блокуються правильною конфігурацією шлюзу, використанням проксі-сервера на шлюзі без NAT і маршрутизації, а також переміщенням проксі-сервера, поштового і веб-сервера в «демілітаризовану зону». » (DMZ, підмережа корпоративної мережі, доступна з Інтернету).

Витоки корпоративних даних мають переважно організаційний характер і є найскладнішою проблемою для служби безпеки підприємства. Існують технічні рішення, які мінімізують можливість цього: зокрема, закриття всіх портів TCP / UDP на інтерфейсі шлюзу, який «дивиться» в локальну мережу (залишається тільки порт проксі-сервера). Слід вимкнути маршрутизацію та трансляцію адрес (NAT) між Інтернетом і внутрішніми («сірими») IP-адресами корпоративної мережі.

Перелічені заходи застосовуються, коли проксі-сервер встановлено на шлюзі, але система з проксі-сервером, розташованим у DMZ, вважається більш безпечною.

Найбільш повний захист забезпечується фізичним розділенням локальної корпоративної мережі та Інтернету. У цьому випадку підприємство організовує комп'ютерну мережу для роботи в Інтернеті, не пов'язану з локальною мережею каналами передачі інформації. Дані, які потрібно надіслати електронною поштою, передаються на знімні носії (наприклад, компакт-диски), які перевіряються системою безпеки та шифруються, наприклад, за допомогою PGP, безкоштовної програми для шифрування електронної пошти та файлів.

2.1.2 Імітаційна модель мережі в Cisco Paket Tracer

Для проектування топології мережі було виділено наступну IP адресу – 172.16.36.0/22. Метод VLSM було успішно використано для

розбиття данної адреси. VLSM має під собою на увазі розбивку, так щоб підмережі були розбиті на рівну кільки хостів у кожній з них.

Для того, щоб зменшити непродуктивні витрати, вводиться ряд нових правил розподілу адрес при використанні метода VLSM.

Якщо використовувати цей метод, то не буде потрібно видаляти підмережі з номерами, які складаються тільки з нулів та одиниць. Ці підмережі тепер можуть використовуватись для розміщення в них хостів. Проте все одно залишиться необхідність у видаленні першої та останньої IP- адреси з кожної підмережі.

Додатково, з’являється можливість застосовувати різні маски до різних частин мережі. Це допоможе розділити мережу на менші частини, якщо з’явиться така необхідність. При цьому обов’язковою вимогою є те, щоб діапазони адрес у підмережах не перекривали один одного.

Було побудовано модель мережі за допомогою програмного пакету Cisco Packet Tracer відповідно до організаційної структури компанії «Ювентус».

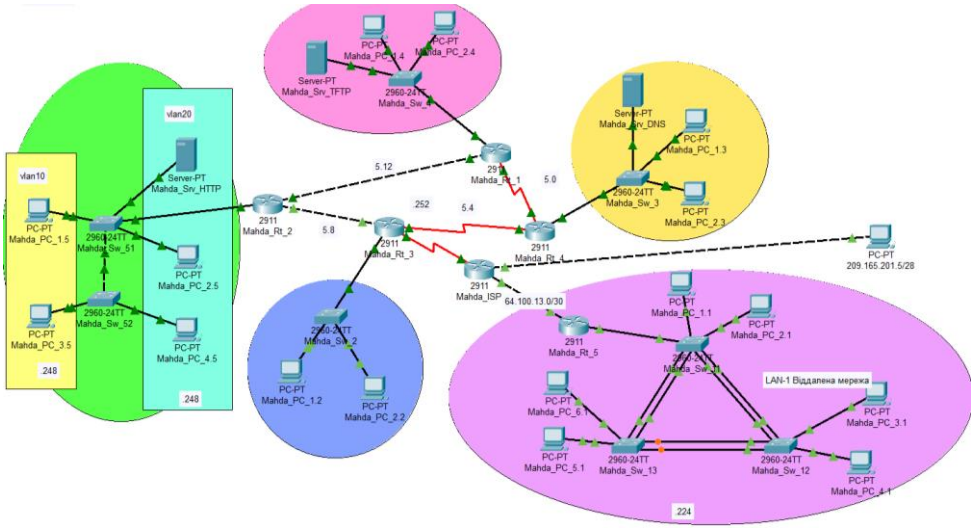


Рисунок 2.10 – Модель мережі компанії «Ювентус»

Таблиця 2.1 – Розрахунок IP-адрес методом VLSM

Назва мережі	Кількість вузлів	Номер мережі	Маска мережі	Діапазонадрес вузлів у підмережі
Lan1	30	172.16.36.0	255.255.255.224	172.16.36.1-30
Lan2	30	172.16.36.32	255.255.255.224	172.16.36.33-62
Lan3	30	172.16.36.64	255.255.255.224	172.16.36.65-94
Lan4	30	172.16.36.96	255.255.255.224	172.16.36.97-126
Lan5	30	172.16.36.128	255.255.255.224	172.16.36.129-158
Wan1	30	10.10.5.0	255.255.255.252	10.10.5.1-2
Wan2	30	10.10.5.4	255.255.255.252	10.10.5.5-6
Wan3	30	10.10.5.8	255.255.255.252	10.10.5.9-10
Wan4	30	10.10.5.12	255.255.255.252	10.10.5.13-14

Таблиця 2.2 – Схема адресації пристроїв

Пристрій	Інтерфейс	IP-адреса	Маска	VLAN	Інтерфейс
Mahda_Rt_5	G0/1	172.16.36.1	255.255.255.224	-	G0/1
Mahda_Rt_3	G0/0	172.16.36.33	255.255.255.224	-	G0/1
Mahda_Rt_4	G0/0	172.16.36.65	255.255.255.224	-	G0/1
Mahda_Rt_1	G0/0	172.16.36.93	255.255.255.224	-	G0/1
Mahda_Rt_2	G0/0.10	172.16.36.129	255.255.255.248	10	G0/1
Mahda_Rt_2	G0/0.20	172.16.36.137	255.255.255.248	20	G0/1
Mahda_Rt_2	G0/0.30	172.16.36.145	255.255.255.248	30	G0/1
Mahda_ISP	G0/0	64.100.13.1	255.255.255.252	-	G0/0
Mahda_ISP	G0/1	209.165.201.1	255.255.255.240	-	Fa0

Для побудови мережі було використано програмний пакет Cisco Packet Tracer. Це програма, за допомогою якої можна змоделювати комп’ютерні мережі. Вона допомагає системним адміністраторам у

проведенні експериментів з поведінкою мережі та оцінюванні можливих сценаріїв розвитку подій. Програмний пакет Cisco Packet Tracer доповнює фізичне обладнання і ця комбінація допомагає створювати мережі з майже необмеженою кількістю пристроїв, що у свою чергу дозволяє отримати практичні навички пошуку, конфігурації та усунення проблем.

Було побудовано модель заданої комп'ютерної системи згідно з вимогами вихідної топології за допомогою програмного пакету Cisco Packet Tracer. Програма допомагає виконати валідацію роботи системи після виконання базових налаштувань пристроїв всередині мережі.

Відповідно до завдання було виконано призначення IP-адреси ПК за протоколом DHCP.

DHCP – це протокол, що відповідає за динамічний розподіл IP-адрес між пристроями всередині мережі. Цей протокол позбавляє системного адміністратора від зайвих клопотів – фахівцю більше не буде потрібно призначати IP-адреси новим комп'ютерам вручну.

Спираючись на організаційну структуру підприємства було реалізовано 3 сервери TFTP, DNS та HTTP.

TFTP (Trivial Transfer Protocol – простий протокол передачі файлів) – протокол, який в основному використовується для первинного завантаження бездискових робочих станцій. На відміну від FTP, цей протокол не підтримує методи аутентифікації (проте можлива фільтрація за IP-адресою) і в його основі лежить протокол транспортного рівня UDP.

HTTP – широко поширений протокол передачі даних, спочатку призначений для передачі гіпертекстових документів (тобто документів, які можуть містити посилання, що дозволяють організувати перехід до інших документів).

HTTP – це протокол передачі даних, який є дуже поширеним. Спочатку він був призначений для передачі гіпертекстових документів

(документів, які в собі містять посилання на інші документи, що дозволяє організувати перехід від одного документу до іншого).

Служба DNS – це глобальний розподілений сервіс, який перетворює доменні імена (наприклад, example.com.ua) в числові IP-адреси (наприклад, 192.145.76.2), які використовуються при взаємодії між серверами. Система DNS нагадує телефонну книгу, за допомогою якої можна знайти телефонний номер абоненту за його ім'ям. Сервери DNS перетворюють запити по доменних іменах на IP-адреси, що забезпечуватиме коректне з'єднання користувача з певним сервером, який знаходиться саме по тій IP-адресі, яка відповідає доменному імені. Такі запити називають DNS-запитами.

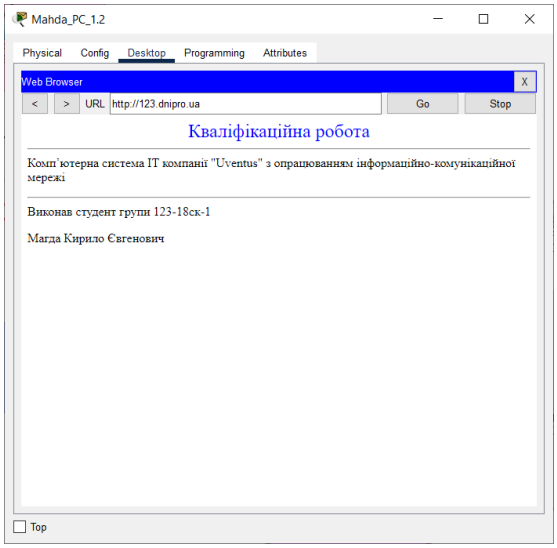
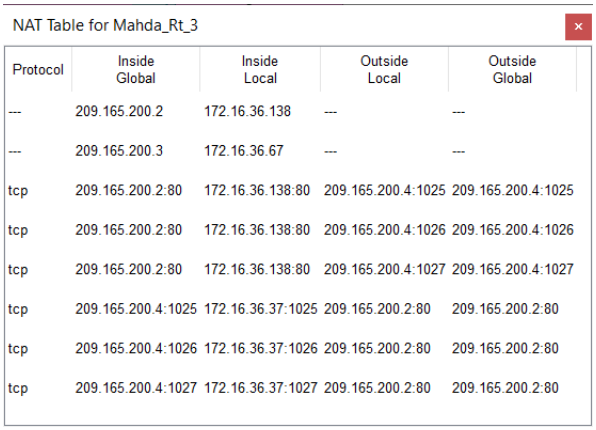


Рисунок 2.11 – Результат роботи HTTP серверу

Згідно з завданням було реалізовано протокол NAT. В процесі проектування мереж зазвичай використовують приватні IP-адреси. Це робиться для того, щоб була підтримка локальної взаємодії між пристроями всередині організації або компанії, а не маршрутизація у глобальну мережу. Для того, щоб пристрій, якому призначена адреса IPv4,

міг звернутись до інших серверів в глобальній мережі інтернет, його приватну адресу потрібно перетворити в загальнодоступну.



NAT Table for Mahda_Rt_3

Protocol	Inside Global	Inside Local	Outside Local	Outside Global
---	209.165.200.2	172.16.36.138	---	---
---	209.165.200.3	172.16.36.67	---	---
tcp	209.165.200.2:80	172.16.36.138:80	209.165.200.4:1025	209.165.200.4:1025
tcp	209.165.200.2:80	172.16.36.138:80	209.165.200.4:1026	209.165.200.4:1026
tcp	209.165.200.2:80	172.16.36.138:80	209.165.200.4:1027	209.165.200.4:1027
tcp	209.165.200.4:1025	172.16.36.37:1025	209.165.200.2:80	209.165.200.2:80
tcp	209.165.200.4:1026	172.16.36.37:1026	209.165.200.2:80	209.165.200.2:80
tcp	209.165.200.4:1027	172.16.36.37:1027	209.165.200.2:80	209.165.200.2:80

Рисунок 2.12 – Результат роботи налаштованого протоколу NAT

Після оцінювання мережі та прийняття рішення щодо вибору маршрутизації, було вирішено використовувати протокол маршрутизації RIPv2. RIP є одним з найпростіших протоколів маршрутизації. Він застосовується в невеликих комп'ютерних мережах та дозволяє маршрутизаторам динамічно оновлювати маршрутну інформацію.

2.2 Моделі комп'ютерних мереж

2.2.1 Сучасні пакети для моделювання комп'ютерних мереж

Перш за все, необхідно згадати офіційну віртуальну лабораторію від Cisco. Це Cisco VIRL (Virtual Internet Routing Lab). Поточна версія 1.6. Офіційний сайт http://virl.cisco.com (кумедно, що у 2020 році сайт, створений одним із найбільших виробників рішень для мережевої безпеки, не має версії TLS).

Продукт розповсюджується як віртуальна машина або пакет для встановлення на «голому металі». Вартість - нелюдські 199 доларів за 365 днів і не більше 20 вузлів віртуальної мережі (термінова підписка на

локальний софт - все дуже модно і сучасно). Packet.net має хмарну версію VIRL.

VIRL уже містить навчальні версії образів IOSv, IOSvL2, IOS XRv, NX-OSv, CSR1000v, ASAv. До неї також можна додати сторонні віртуальні машини інших виробників мережі.

VIRL використовує власний клієнт GUI VM Maestro.

GNS3 – це одне з найпопулярніших програм для емуляції мережі, яке дозволяє спостерігати за взаємодією мережевих пристроїв у різних топологіях мережі. Це програмне забезпечення з відкритою ліцензією, яке є інтегрованим сегментом міжнародної мережі навчання сертифікації. Цього факту достатньо, щоб показати, наскільки сучасним і комплексним є цей програмний інструмент, коли справа доходить до успішного моделювання мережі. Його легко встановити та застосувати, що робить його популярним вибором як для любителів, так і для професіоналів.

Як уже згадувалося, GNS3 — це програмне забезпечення з відкритим вихідним кодом, яке можна завантажити та використовувати безкоштовно. Вихідний код доступний на GitHub, якщо вам цікаво поглянути на код. Сподіваємось, ви знайдете це корисним і корисним, але якщо вам щось не подобається або ви хочете щось додати, чому б вам не взяти участь, зробивши внесок? Приєднайтеся до спільноти або зробіть добровільну роботу, щоб перевірити код або додати рекомендації щодо коду. Маючи понад 800 000 членів спільноти, ми всі можемо вчитися один у одного.

Однак існують інші доступні варіанти, якими ви можете скористатися. Деякі з них безкоштовні, деякі платні. Використовуйте те, що вам найкраще підходить. Використовуйте кілька варіантів, якщо хочете. Ми щасливі, що сьогодні доступна велика кількість варіантів, які допомагають усім нам вдосконалюватися та дізнаватися більше про мережеві зв’язки.

Переваги:

- Безкоштовне програмне забезпечення
- Програмне забезпечення з відкритим кодом
- Ні щомісячної, ні річної ліцензійної плати
- Немає обмежень щодо кількості підтримуваних пристроїв (єдиним обмеженням є ваше обладнання: процесор і пам’ять)
- Підтримує кілька варіантів перемикання (модуль Etherswitch NM-ESW16, зображення рівня 2 IOU/IOL, VIRL IOSvL2)
- Підтримує всі образи VIRL (IOSv, IOSvL2, IOS-XRv, CSR1000v, NX-OSv, ASAv)
- Підтримує середовища багатьох постачальників
- Можна запускати з гіпервізорами або без них
- Підтримує як безкоштовні, так і платні гіпервізори (Virtualbox, VMware workstation, VMware player, ESXi, Fusion)
- Доступні безкоштовні, попередньо налаштовані та оптимізовані пристрої, які можна завантажити, щоб спростити розгортання
- Вбудована підтримка Linux без додаткового програмного забезпечення віртуалізації
- Програмне забезпечення від багатьох постачальників у вільному доступі
- Велика та активна спільнота (800 000+ учасників)

Недоліки:

- Зображення Cisco має надати користувач (завантажити з Cisco.com або придбати ліцензію VIRL або скопіювати з фізичного пристрою).
- Не є самостійним пакетом, але вимагає локальної інсталяції програмного забезпечення (GUI).

- На GNS3 можуть впливати налаштування та обмеження вашого комп’ютера через локальне встановлення (налаштування брандмауера та безпеки, політика корпоративного ноутбука тощо).

EVE-NG: Emulated Virtual Environment Next Generation або EVEN-NG – це єдина в своєму роді багатокористувацька онлайн-симуляція, розроблена для малого бізнесу та окремих осіб. Впровадження цього інструменту моделювання віртуальної мережі є як платним, так і безкоштовним. Безкоштовна версія має обмеження в 63 вузли на лабораторію. Немає необхідності завантажувати та встановлювати додаткову програму на додаток до сервера для віртуалізації, з’єднання та налаштування мережевих пристроїв. Все проектування, підключення та керування мережевими топологіями можна легко виконати за допомогою інтегрованого клієнта HTML5.

Важливим фактором, який робить EVE-NG одним із найкращих інструментів моделювання мереж, є те, що програма економить час, дозволяючи вносити зміни в топологію мереж, коли вони працюють одночасно. Крім того, він підходить як для Ethernet, так і для послідовного інтерфейсу.

Ключові риси:

- KVM HW прискорення
- Дизайнер топології «натисніть і грайте»
- Конфігурація імпорту/експорту
- Формат файлу Labs xml
- Імпорт зображень і карт «натисніть і грайте»
- Спеціальна підтримка ядра для протоколів L2
- Оптимізація пам'яті (УКСМ)
- CPU Watchdog
- Повний інтерфейс користувача HTML5
- Можливість використання без додаткових інструментів

- Багатокористувацький
- Повністю підтримується взаємодія з реальною мережею
- Одночасний лабораторний екземпляр

Думка спільноти:

- Спільне використання дизайну (можливість обміну лабораторіями, конфігураціями онлайн з друзями чи іншими)
- Загальні вдосконалення інтерфейсу користувача (щоб мати можливість робити 99% речей з інтерфейсу користувача, CLI, звичайно, залишиться для досвідчених користувачів)
- Без клієнта – telnet, rdp, vnc через html5
- Перехоплення локального клієнта Wireshark
- Імпорт/експорт конфігурацій
- І багато, багато іншого

Професійна думка:

- Динамічне перенесення консолі, без обмежень, виправлення проблем для консолі кількох користувачів, перенесення Telnet вибирається випадково
- Гарячі з’єднання, запущені вузли з’єднання, миттєва реакція портів, закриття без закриття, лише Ethernet
- Підтримка 1024 вузлів на лабораторію
- Підтримка Docker контейнерів
- Консоль робочого столу HTML для керування EVE, безклієнтське керування EVE
- Функція закриття запущеної лабораторії поміщається в папку запущених, опція запуску кількох лабораторій одночасно

2.2.2 Особливості моделювання комп’ютерних мереж статистичними методами

Закон Меткалфа сприяв розвитку та зростанню глобальних мереж. Вчений стверджував, що ефективність Ethernet пропорційна квадрату кількості користувачів. Відкинемо математичне формулювання, навівши висновок: група людей ефективніша за одну людину. Звичайно, можна робити винятки, згадуючи окремих геніїв, зокрема, Ніколу Тесла, який передбачив появу глобальних систем зв'язку, але факт залишається фактом. Команда розробників виконує роботу швидше. Очевидність підтримується професійними виробниками програмного забезпечення, включаючи Microsoft Office. Зростаючі потужності потребують подальшої оптимізації та контролю.

Моделювання мережі

Окремою лінійкою красується моделювання пакетів. Сучасний дизайнер заздалегідь оцінює продуктивність, долаючи етап проектування. У комп'ютерних мережах приблизна продуктивність відома, техніка стала паличкою-виручалочкою стільникових операторів. Інженери вибирають топологію, мережеві протоколи, склад обладнання. Вирішити проблему допомагають генератори пакетів, попутно збирається статистика придуманої мережі. Тут активно допомагає знання елементів теорії ймовірностей. Необхідно правильно оцінити розподіл трафіку шляхом вибору моделей модельних законів (Пуассона, Ерланга, Парето).

Моделювання комп'ютерних пакетів

Моделюванню піддаються мережі будь-якого типу. Часто розробники стандартів допомагають дизайнерам, даючи рекомендації. Стільниковим зв'язком займається комітет 3GPP2. Організація рекомендує кілька типів генераторів трафіку. Сучасна мобільна мережа мало чим поступається комп'ютерній мережі, активно впроваджуючи типові

протоколи, в тому числі IP. Відмінність обмежується яскравим поділом напрямку руху інформації:

- висхідна гілка;
- низхідна гілка.

Наприклад, генератор протоколу FTP імітує завантаження файлів. Паралельно віртуальні «користувачі» відкривають сторінки Всесвітньої павутини, окремі особи здійснюють дзвінки. Існують методи моделювання:

- Природні. Будується справжня жива мережа, підключене обладнання дає інформацію. Недоліком є висока вартість, хоча фактор адекватності отриманих вимірювань часто переважує.
- Імітація. Майбутня мережа детально моделюється спеціальними пакетами програм (наприклад, CISCO Packet Tracer). Імітуються умови перевантаження, втрати кадрів.
- Аналітичний. Метод повністю обмежений віртуальним простором, точність відносно низька. Інженер закладає формули, математична статистика. Набагато легше, ніж підхід моделювання.
- Комбінований.

Імітація

Програмне забезпечення імітаційної моделі підтримує низку основних пристроїв: концентратори, комутатори, робочі станції, ядро стільникового зв’язку. Програміст задає маски підмережі, адреси. Конструкція ретельно підбирається відповідно до представленого списку пристроїв. При необхідності відсутні елементи замінюються близькими аналогами. Іноді допускається спрощення фізичної структури відповідно до цілей експерименту. Отримане середовище дозволить повністю імітувати процеси, включаючи простий пінг персональних комп'ютерів, абонентів.

Аналіз комп’ютерної мережі – це процес обробки зібраних даних щодо функціонування системи. Адміністратору важливо знати слабкі місця, вчасно виявляти несправності. Іноді елементи керування стоять окремо, як окремий пакет програм. В інших випадках зайві грошові витрати недоречні. Тому набір методів аналізу дуже різноманітний. Процедура контролю складається з двох етапів:

- Моніторинг – це збір інформації.
- Аналіз - це обробка інформації.

Етап моніторингу вимагає збору інформації, здебільшого він відбувається автоматично. Проблема вирішується за допомогою датчиків, програмного забезпечення. Аналіз проводить людина, використовуючи досвід, узагальнений експертами. Розроблені рішення спрощують процес аналізу. Наприклад, експерти пропонують кілька критеріїв збору статистики:

- Кількість помилок.
- Рівень зіткнення.
- Короткі (менше 64 байтів)/довгі (більше 1518 байт) кадри.
- Помилки контрольної суми. Розбіжність є результатом неякісних контактів, перешкод, несправних портів, зламаного обладнання.
- «Привиди», утворені пікапами.

Обробка даних

Очевидно, природа помилок тісно пов'язана з топологією, протоколами. Вони супроводжують мережі Ethernet. Співвідношення груп дефектів іноді допомагає виявити характер несправності. Типова частота помилок кадрів становить менше 0,01%. Наявність зламаних пакетів призводить до зниження пропускної здатності. Місцезнаходження проблем визначається оцінкою кількості, типу колізій:

Локальний — це результат одночасної передачі пакетів кількома мережевими картами сегмента. Високий показник часто супроводжує пошкоджений кабель.

Пульт проникає ззовні в контрольований сегмент. У мережах Ethernet, утворених декількома повторювачами, 100% колізій мають цей тип.

Запізнення (термінологія Ethernet) виявляється після передачі 64 байтів. Частіше це дозволяє локалізувати несправність, непрацездатність мережевого адаптера. Іноді причиною є перевищення довжини кабельної системи, кількість проміжних повторювачів.

Кількість зіткнень зазвичай менше 5%. Крім зазначених статистичних цифр, обладнання часто видає:

- Співвідношення протоколів мережевого рівня. Наявність надлишку ICMP супроводжує сигналізацію про помилку маршрутизаторів.
- Основні відправники/одержувачі.
- Адреси генерації широкомовного трафіку.

Структура методів контролю

Часто засоби аналізу та моніторингу перетинаються. До складу методів входять:

- Системи управління (HP Open View, IBMNetView, SunNetManager) складають ядро. Адміністратор дистанційно зчитує конфігурацію, налаштовує обладнання. У той же час він аналізує продуктивність. Правильно налаштована система надішле сповіщення адміністратору на електронну пошту: були збої, збій. Оповіщення доступні на пейджері, месенджері. Відповідно до визначення ISO класифікація включає:
- SunNetManager
- Обробка помилок.
- Конфігурація та іменування мережі.

- Продуктивність.

Безпека (захист від несанкціонованого доступу): привілеї, аутентифікація, шифрування.

Реєстрація параметрів мережі.

Засоби управління системою (Microsoft System Management Server, Intel LANDeskManager) контролюють програмні та апаратні ресурси. Виробляти облік, збір інформації про локальні комп'ютери. Адміністратор чітко бачить список машин, їх стан і потребу в оновленнях. Установка ПЗ централізована. Деяким людям подобається спостерігати за діяльністю користувача. Розглянута група інструментів дозволить шпигувати. Деякі з наведених вище функцій включають HP Open View, IBMNetView, SunNetManager.

Експертні системи (Spectrum Cabletron) закріплюють досвід попередніх поколінь. Контроль розвивався паралельно з ростом мереж, поступово ускладнюючись. Сформовані бази даних, автоматизовані довідкові системи значно спрощують життя фахівця. Network Monitor від Microsoft спочатку був позбавлений експертної системи.

Агенти системи керування — це програмні модулі, які забезпечують аналітичний центр інформацією. Зазвичай використовуються протоколи SNMP, CMIP, RMON. Стандарти баз даних - MIB-I, MIB-II, RMONMIB. Специфікації цих сховищ визначають величезну кількість об'єктів (журнали протоколів, інформацію про пристрої, таблиці трансляції адрес). Розширені агенти протоколу RMON (LANalyzeNovell) більш інтелектуальні, ноутбуки, робочі станції оснащені програмними модулями. Агент системи управління

Вбудовані системи можуть бути частиною операційної системи або мережевої інфраструктури. Запуск моніторингу, автоматичне відновлення окремого об'єкта. Часто входить до складу системного агента.

Аналізатори протоколів оцінюють трафік. Реалізується програмно-технічними засобами. Аналіз пакетів дозволяє оптимізувати структуру.

Окремо діагностується кабельна промисловість, частина обладнання портативна. Види методів:

- Мережеві монітори показують параметри трафіку. Іноді вони охоплюють кілька рівнів ієрархічної системи OSI (фізичний, канальний, мережевий).
- Кабельні сканери оцінюють стан провідників лінії.
- Пристрої сертифікації дозволяють здати установку замовнику, оцінити клас мережі.
- Тестери дозволяють контролювати обриви ланцюга.

Деякі зразки обладнання багатофункціональні, поєднують в собі можливості тестерів, аналізаторів протоколів:

- Графічний інтерфейс користувача.
- Рядкова розгортка.
- Розпіновка жила, картографування.
- Електричні параметри.
- Швидкість поширення хвилі.
- Тестування ділянок ланцюга з метою локалізації несправності.
- Оцініть справність адаптера.
- Збірка статистики.
- Генерація додаткового трафіку для перевірки пропускної здатності лінії. Згенеровані пакети дозволяють оцінити доступність вузла.

Системи керування мережею

Програмне забезпечення для діагностики мережі має такі функції:

- відкритість;
- масштабованість;
- розподіл.

Спочатку, в історичному плані, ключі брав адміністратор. Royal PC містив повнофункціональну автономну версію програмного забезпечення. Клієнти-раби перебували під пильним наглядом. Наявність великомасштабної мережі порушує принцип. Малопотужний ПК адміністратора вже не впорається. Іноді допомагає найняти команду мережарів. Система управління стає розподіленою. Сумісність обладнання провідних компаній залишає бажати кращого. Виникають помилки відображення, додаткові корисні функції обладнання ігноруються.

Виробники повинні враховувати повний перелік стандартів баз даних: MIB-I, MIB-II, RMONMIB. Сюди ж додаються «фірмові» розробки. Враховуються оригінальні різнорідні варіанти, розширюючи трудомісткість постачальників обладнання та програмного забезпечення. Наприклад, Spectrum підтримує близько 1000 модифікацій. Рекомендується встановлювати оболонки компанії, чиє обладнання становить переважну більшість інфраструктури, знижуючи ризик помилок. У пошуках консенсусу виробники почали використовувати уніфіковане сховище: Oracle, Informix, Ingres.

Програмне забезпечення кожного виробника демонструє певні переваги. Інакше давно б окреслився переможець, який захопив би весь ринок.

Аналізатор трафіку (протоколу).

Трафік оцінюють сніфери. Інструменти (компонент WindowsNTPerformanceMonitor) перехоплюють будь-які пакети, включно з чужими. Оскільки Ethernet має широкомовне значення, адміністратор має повний контроль над локальним сегментом. Добре налаштований концентратор діє як захист від несанкціонованого прослуховування. Комутатори відсікають явно непотрібні зовнішні адреси вже на вході. Поділ мережі на сегменти значно знижує навантаження.

Спеціалізовані пристрої, ПК, ноутбук надають адміністратору можливість перехоплювати будь-які пакети. Необхідно відповідати мережевій карті, програмному забезпеченню, типу протоколу (Ethernet, Token Ring, FDDI). Адаптер аналізатора трафіку налаштований на виявлення будь-яких адрес. Це дозволяє зробити спеціальний режим безладним. Ядро забезпечує функціонування апаратного забезпечення шляхом декодування канального рівня протоколу, іноді вищих етапів (IP, TCP, Telnet, FTP, HTTP). Підтримка стеків TCP / IP, NetBIOS, Banyan VINES, Novell NetWare вважається стандартною. Паралельно сніффер виконує інші функції:

- Збір аналітичної інформації (статистики): кількість зламаних пакетів, крос-трафік вузлів, використання сегментів.
- Сніффер часто опитує агентів різних сегментів.

Графічний інтерфейс спрощує роботу людини-оператора. Використання тригерів дозволяє задавати умови початку і закінчення захоплення трафіку. Пакети фільтруються, відкидаючи явно непотрібні.

Деякі аналізатори підтримують шпигунство за кількома mac-адресами. Наприклад, мережевий монітор (Windows NT Server) версії 4 або новішої.

Фізичний рівень трохи порадує живого спостерігача. Розкидані кадри нечіткі.

Аналіз трафіку

Це включає (наприклад, систему Hewlett-Packard HP 4195A, 8510C) експертне обладнання для оцінки працездатності кабельної промисловості шляхом вимірювання електричних параметрів. Іноді вони фіксують ієрархію протоколів. Сигнали спеціальної форми дозволяють оцінити амплітудно-частотну характеристику лінії, загасання, перешкоди. Методика сильно нагадує принцип роботи обладнання для оцінки цілісності високовольтних кабелів. Хоча великих напруг не потрібно, тому

механізм більш мініатюрний (промислові займають фургон). Аналізатор мережі іноді може виконувати ряд функцій сніфера, але скоріше допоміжних. Базові перехоплювачі пакетів реалізуються окремо.

Перехресні перешкоди (NEXT) виникають між проводами крученої пари. Значення визначається частотою, категорією кабелю. Виражається в децибелах.

Імпеданс (сукупність активного, реактивного опорів). Реальна складова визначається матеріалом провідника, менше залежить від частоти. Уявна складова утворюється впливом реактивних складових (ємності, індуктивності). Хвильовий опір характеризує тип лінії. Зв'язок з опором слабкий.

Загасання обумовлено наявністю хвильового, активного опору. Скін-ефект провокує випромінювання частини потужності, знижуючи характеристики системи: вища частота → тонший шар → більший опір → збільшені втрати. П'ята категорія кабелів показує значення 23,6 дБ/100 м.

Кількість електромагнітного випромінювання. Екран значно зменшує випромінювання енергії назовні.

Сканери, тестери

Зазначений клас обладнання оцінює електричні параметри кабелю, довжину, значно поступається за точністю мережевим аналізаторам. Переносні варіанти стали незамінними супутниками ремонтників. Оцінено:

- хвильовий опір.
- ДАЛІ.
- Ослаблення сигналу.
- Монтажна схема.
- Електричні параметри кабелю

Принцип дії

Повторює методику високовольтних приладів. Видається імпульс - ловиться луна. Кількість і форма відповідних сигналів дозволяють оцінити фізичний стан кабелю. Принцип вимагатиме знання швидкості поширення сигналу в середовищі. Деякі моделі мають власну базу даних, що містить довідкову інформацію. Тестери набагато простіше. Урізана версія алгоритму просто відповідає на питання про цілісність кабелю, позбавлена деталей.

Сканер використовується для вимірювання довжини лінії. Деякі моделі коштують дуже дорого ($1-3 тис.). Виробник Datacom, Fluke, Microtest, Scope Communications.

Перемикачі

Досвідчений інженер розповість багато про поведінку мережевого обладнання. Перевантаження процесорів, портів безповоротно викликає втрату кадрів. Стандартні комутатори оснащені агентами, які повідомляють про проблеми. В іншому випадку завдання відстеження проблем значно ускладнюється. Працювати з хабами простіше: тестове обладнання вішається на вільний порт. Вимикачі вимагають послідовного з'єднання (розриву). Вільні порти отримуватимуть лише широкомовний трафік.

Справа ускладнюється наявністю кількох віртуальних мереж. Тоді обладнання отримує лише поточні пакети (визначені IP-адресою). Виробники комутаторів усвідомили брак пристроїв, моделі почали забезпечувати функцією дзеркального відображення. Апаратне забезпечення відображає трафік на тестовий сокет. Залишається важко побачити пакети двох портів одночасно або одного, який працює в повнодуплексному режимі.

Це пояснює перевагу адміністраторів щодо використання агентів. Інформація, що передається на контрольну точку, містить повний список

пакетів. RMON збирає трафік Ethernet, Token Ring, створює матриці перехресного трафіку. Проблема тут у вартості. Дорого відображати 9 об'єктів Ethernet, виробники скорочують список. Іноді скасовують роздільне тестування портів, об'єднуючи їх у групи.

2.2.3 Модель мережі як замкнутої системи масового обслуговування

Існує велика кількість методів аналізу комунікаційних мереж, серед яких основою є методи математичного аналізу та моделювання. Останнім часом великий інтерес викликає також тензорний метод мережевого аналізу , який можна виділити в окрему категорію, оскільки ідея методу принципово відрізняється від інших методів. Тензорний метод займає особливе місце, будучи несхожим на інші методи. Особливістю цього методу є те, що його автор намагався об'єднати всі процеси, що відбуваються в системі, і поглянути на систему з більш загальної точки зору. Отже, при аналізі будь-якої складної мережі необхідно спочатку отримати результати для одного елемента цієї мережі, а потім поширити ці результати на всю мережу. Більш того, введення таких понять, як «перетворення», «інваріантність» і «група» призводить до появи нової математичної сутності. Такою сутністю є геометричний об’єкт, який представлений не однією, а нескінченним набором матриць. Зміна системи координат (топології мережі) призводить до зміни компонентів геометричного об'єкта (матричних компонентів), а сам геометричний об'єкт залишається незмінним. Це дозволяє розглядати цілий клас мереж як один об'єкт, а перехід між ними здійснювати за допомогою спеціальної матриці переходів. Математичний аналіз комунікаційних мереж часто вимагає великих зусиль, оскільки суть методів полягає в пошуку аналітичних залежностей між бажаними значеннями. Виведення формул виявляється досить складним, особливо для мереж з великою кількістю

елементів, але отримані результати мають велике значення, оскільки дають можливість безпосередньо розв’язувати задачу та аналізувати залежність результату від зміни різні фактори. Аналітичні співвідношення отримують в основному для мереж з невеликою кількістю елементів. Моделювання, навпаки, практично не вимагає трудомісткого виведення математичних формул - все, що потрібно від дослідника, це знання засобів моделювання та параметрів системи, що моделюється.

Модель інтегральних мереж обслуговування в даній роботі являє собою два типи мереж масового обслуговування: контурну та ортогональну. При аналізі інтегрованих мереж обслуговування за допомогою імітаційного моделювання компонентами загальної моделі є моделі систем масового обслуговування з різними дисциплінами обслуговування. Система масового обслуговування (СМО) складається з сервера і черги.

Особливістю мережі є взаємопов'язаність систем масового обслуговування між собою: транзакції не видаляються після кожної системи масового обслуговування, а надходять до наступної СМО, в результаті чого повідомлення просуваються по мережі. Слід зазначити, що при обмеженому буфері в СМО частина повідомлень втрачається, а потік на виході системи масового обслуговування відрізняється від вихідного. Крім того, відбувається зміна структури самого потоку повідомлень. Розглянемо моделювання цілісної мережі обслуговування з СМО виду M/M/1/N. Результати моделювання порівняємо з розрахунком характеристик мережі тензорним методом. Розрахунок тензорним методом дозволяє аналізувати характеристики мереж, що складаються практично з будь-якої кількості систем масового обслуговування. Збільшення кількості СМО призводить лише до збільшення часу розрахунку без істотного ускладнення самого принципу розрахунку.

Залежно від складу і часу перебування в черзі перед початком обслуговування, а також від дисципліни вимог до обслуговування ОМК поділяються на різні групи. За складом СМО розрізняють одноканальні (з одним обслуговуючим пристроєм) і багатоканальні (з великою кількістю обслуговуючих пристроїв). Багатоканальні системи можуть складатися з обслуговуючих пристроїв як однакової, так і різної продуктивності. За часом перебування вимог у черзі до початку обслуговування системи поділяються на три групи: 1) з необмеженим часом очікування (з очікуванням), 2) з відмовами; 3) змішаний тип. У QS з необмеженим часом очікування наступний запит, виявивши, що всі пристрої зайняті, стає в чергу та чекає на обслуговування, доки один із пристроїв не звільниться. У системах зі збоями вхідний запит залишає систему після того, як всі пристрої будуть зайняті. Класичним прикладом системи зі збоями є робота АТС. У системах змішаного типу вхідна вимога, встигнувши все (пристрої зайняті, стоять у черзі і чекають обслуговування протягом обмеженого часу. Не дочекавшись обслуговування у встановлений час, вимога залишає систему. У системах з певним дисципліна обслуговування, вхідний запит, виявивши всі пристрої зайнятими, в залежності від його пріоритету, він або обслуговується позачергово, або ставиться в чергу. Основними елементами QS є: вхідний потік запитів, черга запитів , обслуговуючі пристрої (канали) і вихідний потік вимог. Дослідження QS починається з аналізу вхідного потоку вимог. це набір вимог, які надходять в систему і потребують обслуговування. Вхідний потік вимог є досліджено з метою встановлення закономірностей у цьому потоці та подальшого покращення якості обслуговування. У більшості випадків вхідний потік є неконтрольованим і залежить від ряду випадкових факторів. Кількість запитів, що надходять за одиницю часу, є випадковою величиною. Випадкова величина е також інтервал часу між сусідніми вхідними запитами. Однак передбачається, що задано середню кількість запитів,

отриманих за одиницю часу, і середній інтервал часу між сусідніми вхідними запитами. Середня кількість вимог, що надходять в систему обслуговування за одиницю часу, називається інтенсивністю надходження вимог. Для багатьох реальних процесів потік вимог досить добре описується законом розподілу Пуассона. Такий потік називають найпростішим. Найпростіший потік має такі важливі властивості: 1) Властивість стаціонарності, яка виражає незмінність імовірнісного режиму течії в часі. Це означає, що кількість клієнтів, які регулярно входять до системи, має бути в середньому постійною.

Наприклад, кількість вагонів, що прибувають під навантаження в середньому за добу, має бути однаковою для різних періодів часу, наприклад, на початку і в кінці декади. 2) Відсутність післядії, що визначає взаємну незалежність надходження тієї чи іншої кількості заявок на обслуговування в непересічні часові інтервали. Це означає, що кількість запитів, які надходять за певний проміжок часу, не залежить від кількості запитів, обслужених за попередній проміжок часу. Наприклад, кількість автомобілів, які прибули за матеріалами десятого числа місяця, не залежить від кількості автомобілів, обслужених четвертого або будь-якого іншого попереднього числа цього місяця. 3) Властивість звичайності, що виражає практичну неможливість одночасного надходження двох і більше вимог (ймовірність такої події незмірно мала по відношенню до розглянутого періоду часу, коли остання прагне до нуля). У найпростішому потоці вимог розподіл вимог, що надходять у систему, підкоряється закону розподілу Пуассона. На практиці умови найпростішого потоку не завжди строго виконуються. Часто спостерігається нестаціонарність процесу (в різні години доби і різні дні місяця потік вимог може змінюватися, інтенсивнішим він може бути вранці або в останні дні місяця). Існує також післядія, коли кількість заявок на відпуск товарів на кінець місяця залежить від їх задоволення на

початку місяця. Також спостерігається явище неоднорідності, коли на складі за матеріалами одночасно перебуває кілька клієнтів. Проте в цілому закон розподілу Пуассона з досить високим наближенням відображає багато процесів масового обслуговування. Чому таке припущення виявляється вірним у ряді важливих випадків, відповідає загальна теорема А.Я. Хінчина, що має виняткову теоретичну і практичну цінність. Ця теорема має місце у випадку, коли набігаючий потік можна представити як суму великої кількості незалежних потоків, жоден з яких не порівнянний за інтенсивністю з усім загальним потоком.

Модель Markov QS

Роботу СМО будемо розглядати як випадковий процес з неперервним часом і дискретністю багатьох держав. Тому в будь-який момент $t \in [0; +\infty)$ система перебуває в одному стані з заданої скінченної або зліченної множини.

Наприклад, у майстерні десять однотипних верстатів. Верстати обслуговується одним ремонтником. Таким чином, відповідна система може перебувати в одному з 11 станів: $S0$ - всі машини в хорошому стані, $S1$ - одна в ремонті і дев'ять в робочому стані, $S2$ - одна в ремонті, одна в черзі на ремонт і вісім роботи, . . . , $S10$ - одна в ремонті та дев'ять в черзі. Очевидно, момент відмови машини і час, необхідний для усунення несправностей, є випадковими величинами. В процесі функціонуючи, система іноді переходить з одного стану в інший. Причому, теоретично в будь-який час система може перебувати в будь-якому з перерахованих вище станів. Тому є сенс говорити лише про ймовірності відповідні стани: $P0(t)$, $P1(t)$, $P2(t)$. . . $P10(t)$.

Випадковий процес називається марковським, якщо такий є момент часу t умовні ймовірності всіх станів системи в майбутньому залежать тільки від стану системи в даний момент t і не залежать від того, коли і як воно прийшло в цей стан. Іншими словами, майбутнє залежить від

минулого лише через сьогодення. Марковський процес ще називають процесом без післядії. Процес функціонування СМО будемо вважати марковським. Хоча марківська модель QS не є єдино можливою, вона адекватно відображає широкий клас реальних систем.

Нехай система в будь-який момент часу перебуває в одному з n можливих станів S , де $i = 1, 2, \ldots n$. Зокрема, не виключений випадок $n = \infty$. Тобто багато результатів не що інше, як підрахунок. Іноді нам буде зручніше сказати не «стан Si», а «і-й стан». Ми часто нумеруємо штати почнемо не з одиниці, а з нуля. Зробимо припущення, що ймовірність переходу системи за час h від i − до j – стану задається рівністю $Pij\ (h) = \lambda ij\ h + \circ(h)$, де $i \neq j$. Тобто за короткий проміжок часу ймовірність переходу системи з і-го в j-й стан пропорційна довжині інтервалу. Рівності (18), очевидно, роблять процес марковським. Величина λij , де $i \neq j$, називається інтенсивністю переходу з і-го до j-го стану. Загалом, λij може залежати від часу, але тут ми обмежуємося випадком постійних інтенсивностей.

Рівності подібні до рівності, доведеної в першому розділі для найпростішого потоку $P1(h) = \lambda \cdot h + \circ(h)$. Більш того, він сам потік однорідних подій можна інтерпретувати як випадковий процес накопичення подій. Нехай $k(t)$ буде номером події, що відбулося до моменту t. Кожна реалізація такого випадковий процес є ступінчастою функцією, значення якого зростає на одиницю з появою наступної події.

Також можна пов'язати описаний випадковий процес із системою, маючи багато станів Si, де $i = 0, 1, 2, \ldots, n$. У цьому case і — кількість подій. Тоді інтенсивності переходів:

$$\lambda ij = \begin{cases} \lambda \text{ якщо } j = i + 1, \text{ де } i, j = 0, 1 \ldots \infty; \\ \\ 0 \text{ інакше.} \end{cases}$$

Тут під λ, як і в першому розділі, ми маємо на увазі інтенсивність вхідного потоку подій. В даному випадку на знімальному майданчику стани $S0$, $S1$, $S2$, . . . дозволені лише переходи зліва направо в порядку зростання чисел.

СМО з дискретною множиною станів часто буде схематично зображуватися у вигляді орієнтованого графа, вершинами якого є стани, а дугами – допустимі переходи з одного стану в інший.

До цих пір ми розглядали QS, в якому вхідний потік не залежить від того, скільки заявок зараз обслуговується або в черзі. Отже, у великому місті можна вважайте, що ваш дзвінок не вплине на загальну інтенсивність дзвінків у місті, і, хоча населення навіть дуже великого містах, звичайно, ви можете вважати кількість джерел додатків нескінченною в моделі. Інша справа, наприклад, коли в магазині лише десять машин і один майстер для їх ремонту. Тоді модель майстерні буде QS зі станами $\{Si\}$, де i = 0, 1, . . . , n : $S0$ – всі машини працюють, $S1$ - одна машина в ремонті, решта працюють, $S2$ - одна в ремонті, одна на черзі, решта працюють і, нарешті, Sn - один в ремонті, решта в черзі на ремонт!

Інтенсивність потоку заявок на ремонт від однієї машини, тобто інтенсивність потоку відмов - λ , інтенсивність обслуговування – 68 μ. Величина $1/\lambda$ в теорії надійності називається напрацюванням за відмову. Такі системи називають закритими, або системами Енгсет, що показано на рисунку 2.17.

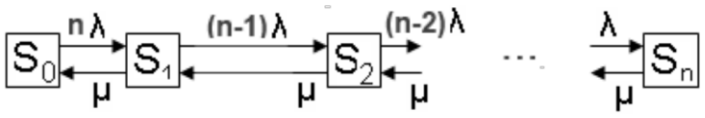


Рисунок 2.13 - Замкнена одноканальна СМО

Абсолютна пропускна здатність QS $A = P\text{zan}\ \mu$, де Pzan це ймовірність того, що система зайнята обслуговуванням запиту.

Нагадаємо, тут μ продуктивність системи за умови безперервного обслуговування додатків без простоїв.

У закритій СМО можна виділити деяку кількість активних (act) і пасивних (pass) джерел заявок. Наприклад, пасивними джерелами є машини в ремонті або в черзі на ремонт. Лише активне джерело може надсилати новий запит на послугу. Очевидно, N діяти + N проходити = n. Також для середніх: $\bar{N}+\bar{N} = n$. Середня інтенсивність вхідний потік $\Lambda = \bar{A} = (1-P0)\ \mu = (n-\bar{N})\ \lambda = \Rightarrow \bar{N} = n-1 - P0\ \rho$.

На жаль, обсяг посібника не дозволив розглянути багатоканальні системи з обмеженнями по довжині і без них, черги, багатоканальні системи з нетерплячими заявками і закриті багатоканальні системи. Теж не розглядається багатофазних систем, ряд теоретичних питань опущено існування розв'язків рівнянь Колмогорова, а також окремі випадки вхідного потоку та порядку обслуговування, випадки коли порушується звичайність потоку або коли заявки надходять за законами найпростішого потоку, а окремі заявки ні подаються. Наприклад, заняття на курсах англійської починаються після комплектування груп. У таких випадках не завжди класичні підручники дадуть відповідь на всі питання і звернутися до додаткової літератури. Проте студент, який оволодів методами моделювання СМО та основними характеристиками СМО на прикладах, розглянутих у книзі та озброєний необхідною літературою, зможе самостійно вивчити багато класів QS, які ми пропустили.

3 Синтез комп’ютерної СИСТЕМИ підприємства «Ювентус»

3.1 Структура комплексу технічних засобів комп’ютерної системи підприємства

Загальна структура компанії поділяється на такі відділи, як CEO компанії, адміністративних співробітників, відділи розробки, відділ тестування, відділ системних адміністраторів та системних інженерів, відділ бізнес аналітики, менеджменту.

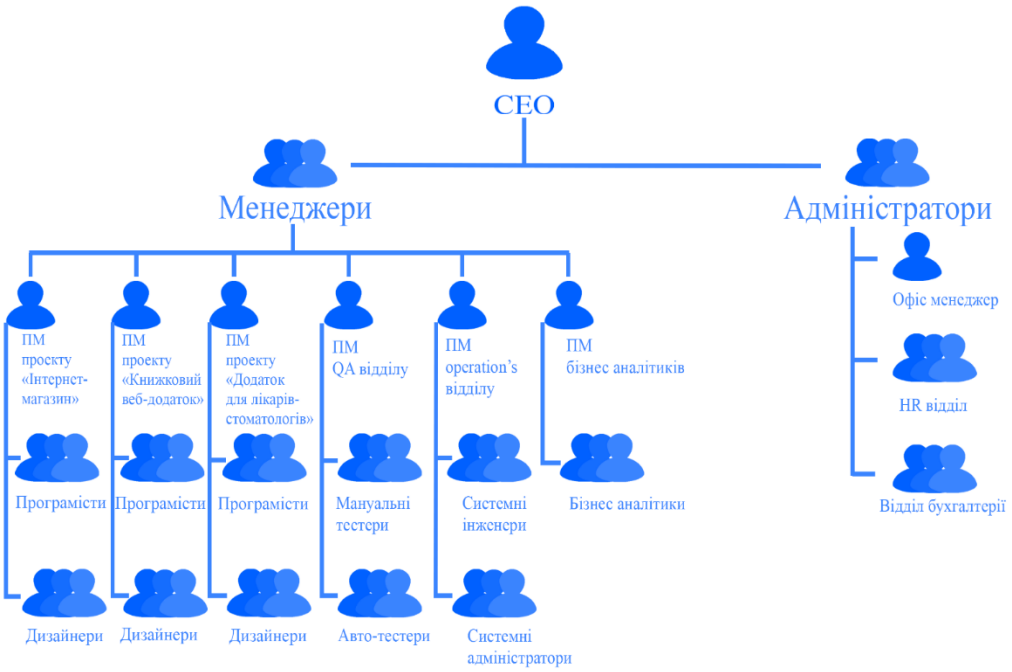


Рисунок 3.1 – Організаційна структура IT компанії «Ювентус»

Ієрархію підприємства наведено на рисунку 3.1. Вона поділяється на розробників, дизайнерів для різних напрямків компанії, тестувальників, аналітичний персонал, адміністрація офісу та керівничий відділ, який складається з CEO та проектних менеджерів.

CEO (Chief Executive Officer) – виконавчий директор. У англійському варіанті: Managing Director – посада у компанії, яка є найвищою.

PM (Project Manager) – посада, на якій людина керує проектами, планує та розподіляє завдання у команді, інколи спілкується з клієнтом, щоб розуміти можливі нюанси або вимоги.

Програміст – це інженер, який розробляє програмне забезпечення, за допомогою певних практик, методологій з використанням обраних технологій мови програмування.

Дизайнер – людина, яка займається графічним наповненням проекту. Він допомагає створити інтерфейс для користувача так, щоб він був максимально продуманим і гарним, авжеж з урахуванням нюансів від замовника.

QA (Quality Assurance) – людина, яка займається тестуванням готових компонентів або цілих програмних блоків на різному етапі створювання ПО. Тестування буває мануальне, тобто ручне, або автоматизоване, тобто тестування за допомогою написаного програмного коду.

Operation's department – відділ інженерів, які займаються мережевою та комп'ютерною частиною підприємства. Мета цього відділу доглядати за серверами, мережею, безпекою, контролем версій коду, базами даних, персональними машинами працівників, репозиторіями тощо. До складу такого відділу можуть входити SA System Administrators, SE System Engineer, DBA Data Base Administrators, Tech Support, CM Configuration Management, Security Team.

Бізнес-аналітик – людина, яка спілкується з замовником, для формування вимог, що до реалізації проекту. Згідно цих вимог програмісти пишуть програмний код, який згодом демонструють клієнту.

Адміністратори офісу це відділ який складається з HR та офіс менеджера. Основна його мета, це забезпечення комфортних та безперебійних умов для праці.

Бухгалтерія – відділ, який займається збором та обробкою даних про майно та зобов’язанням підприємства.

3.2 Характеристики використаних апаратних засобів

3.1.1 Робочі станції

На рисунку 3.2 зображенні ПК для були застосовані для підрозділів «CEO», «Менеджери», «Адміністратори», «Бізнес аналітики». Варіант був обраний спираючись на об’єм і тип задач та тип програмного забезпечення для цих підрозділів.

Рисунок 3.2 – ARTLINE Work H47 v16

Таблиця 3.1 – Технічні характеристики

Назва характеристики	Опис характеристики
Процессор	AMD Ryzen 3 3200G 3.6 GHz
Об’єм оперативної пам’яті	8 ГБ
Тип відео карти та об’єм відео пам’яті	Інтегрована AMD Vega 8
Об’єм HDD	1 ТБ
Потужність БП	550 Вт

Порти	На передней панели: 2 x USB 2.0; виход на наушники; вход для микрофона. На задней панели (материнская плата): 1 x PS/2 для миші; 1 x PS/2 для клавиатури; 2 x USB 2.0; 2 x USB 3.0; 1 x LAN (RJ45); 3 x Аудио разъема. На задней панели (видеокарта): 1 x DVI–D; 1 x HDMI;
Охолодження	BOX
Завантажене ПЗ	Без ОС
Вага	6 кг
Розміри	350 x 170 x 345 мм
Оптичний привід	DVD+/–RW
Тип памяті	DDR4–2133 МГц

58 таких станцій було придбано для працівників у даних підрозділах.

На рисунку 3.3 зображенні варіанти ПК для працівників підрозділів «Програмісти», «Тестувальники», «Дизайнери», «Системні інженери та адміністратори». Рішення було прийнято згідно потреб у продуктивності персональних машин.

Рисунок 3.3 - ARTLINE WorkStation W196 v56

Таблиця 3.3 – Технічні характеристики

Назва характеристики	Опис характеристики
Процессор	AMD Ryzen 5 7600x 4.7 Ghz
Тип видеокарти та об’єм видеопам’яті	Дискретная, nVidia GeForce GTX 1660 Super, 6 ГБ видеопамяти
Об’ем оперативної памяти	32 ГБ
Об’ем HDD	HDD 2 ТБ + SSD 250 ГБ
Потужність БП	600 Вт
Порти	На передней панели: 1 x USB 3.0; 3 x USB 2.0; вихід на навушників; вхід для мікрофона. На задней панели (материнская плата): 1 x PS/2 порт для клавиатуры/мыши 1 x USB 3.1 Type–A

	1 x USB 3.1 Type–C 4 x USB 3.0 1 x DisplayPort 1 x HDMI 1 x LAN (RJ–45) 1 x оптичний S/PDIF вихід 5 x Аудио раз'ємів На задній панели (видеокарта): 2 x DisplayPort 2 x HDMI
Охолодження	ЦП:be quiet! Dark Rock Pro 4 Охолодження корпуса: 1 x 120 мм
Завантажене ПЗ	Без ОС
Розміри	445 x 210 x 445 мм
Тип пам’яті	DDR4–3600 МГц
Модуль WiFi	Нет

44 таких станцій було придбано для працівників у даних підрозділах.

3.1.2 Сервери

Не враховуючи потужності з хмарних систем, для потреб компанії у обчислювальній потужності для віртуальних машин і серверів для різних потреб, в тому рахунку клієнтських, було вирішено придбати потужний сервер від компанії IBM, який зможе задовільнити усі бізнес процеси.

Рисунок 3.4 – Сервер IBM E9080

Таблиця 3.3 – Технічні характеристики

Назва характеристики	Опис характеристики
Виробник	IBM
Код виробника	9080-HEX
Тип	Сервер
Форм-фактор	Rack-dense
Серія	Power
Модель	E1080
Монтаж у стійку	5U
Для процесорів	IBM
Серія процесора	Power10
Встановлено процесорів	3
Процесор, кількість ядер	60
Максимально процесорів	4
Тип пам'яті	DDR4
Встановлено модулі пам'яті	24
ОЗП, об'єм одного модуля, ГБ	128
Максимальний обсяг пам'яті, ГБ	1024

Кількість слотів пам'яті	64
Тип HDD	NVMe
HDD, швидкість обертання шпинделя	4K
Встановлено HDD	8
Ємність встановленого HDD, ГБ	800
БП, тип	AC
Встановлено БП	1
Потужність одного встановленого БП, Вт	950
Максимальна кількість БП	16
Гаряча заміна БП	ТАК

На цей час, був придбан один такий сервер.

3.1.3 Активне обладнання мережі

Мережевий комутатор – пристрій, призначений для з'єднання декількох вузлів комп'ютерної мережі в межах одного або декількох сегментів мережі.

Рисунок 3.6 – Коммутатор L3 Gigabit Ethernet PoE Cisco SG350-28MP-K9-EU

Керовані комутатори серії 350 L3 є наступним поколінням (найпопулярнішими моделями) добре відомої на ринку серії 300. За своєю конструкцією вони використовуються як основа для надійної, високопродуктивної мережі та можуть підтримувати функції, необхідні для підвищення доступності критично важливих бізнес-додатків і захисту конфіденційної інформації. Ідеальне поєднання функцій для середнього та великого бізнесу та підтримка сучасних технологій роблять лінійку комутаторів 350 найкращою у своєму класі.

Основні характеристики та переваги:

Підтримка статичної маршрутизації між підмережами VLAN (до 512 статичних маршрутів і 128 IP-інтерфейсів) дозволяє сегментувати мережу на окремі робочі групи і налаштовувати взаємодію між ними без зниження продуктивності;

Перехід на елементну базу, засновану на техпроцесі 28 нм, покращує характеристики комутаторів і дозволяє значно підвищити їх продуктивність при оптимальному енергоспоживанні;

Розширені функції безпеки та підтримка живлення зовнішніх пристроїв за стандартом IEEE 802.3at (до 30 Вт на порт) і до 60 Вт (на виділених портах) роблять ці комутатори ідеальною платформою для реалізації відеоспостереження, мережі Wi-Fi.Fi, IP Телефонія та ін. ;

Окрім джерела живлення від мережі 220 В, 8-портові моделі комутаторів PoE можна запитувати від вищестоящого комутатора PoE (до 2x UPLINK, до 60 Вт кожен). При цьому вони можуть працювати в режимі пропуску PoE з бюджетом до 50 Вт на свічку;

Підтримувана функція «PoE powered device» також може бути використана для підтримки живлення комутатора: якщо є мережа 220 В і висхідна лінія PoE з потужністю 15,4 Вт або більше, комутатор працюватиме від мережі 220 В. У разі збою живлення змінного струму комутатор переключиться на живлення PoE;

Вбудований механізм захисту від «широкомовного шторму», зовнішніх атак, таких як «Відмова в обслуговуванні», покращує ефективність мережі;

Підтримка ACL (до 512 правил) дозволяє гнучко налаштовувати обмеження та доступ до об'єктів на основі таких параметрів, як: VLAN ID, MAC або IP-адреса, протокол, порт тощо;

Неблокуючий повнодуплексний комутатор забезпечує передачу даних зі швидкістю інтерфейсу;

Умови експлуатації: температура від 0°C до 40°C; відносна вологість від 10% до 90%

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Призначення і область застосування програми

Головна призначення застосунку – це демонстрація актуальних версій third-party компонентів на відповідних вузлах у локальній мережі.

Результатом розробки має бути веб-додаток, який максимально лаконічно, виводить потрібну інформацію о версіях third-party компонентів, як веб-сторінку у браузері.

Область застосування програми, може бути як системне адміністрування, так і комп’ютерна безпека.

4.2 Обґрунтування технічних характеристик програми

4.2.1 Постановка завдання на розробку програми

Розробити застосунок для моніторингу версій third-party компонентів, які встановленні на вузлах у підприємстві.

Програма повинна працювати так, щоб в режимі реального часу, відавати актуальну інформацію не використовуючи базу даних для цього.

Розроби зрозумілий, лаконічний інтерфейс веб-сторінки на якому буде видно назву хоста, назву компоненту та його версію, еталону версію, кольорові відмінності еталонної та фактичної версій.

На кожному вузлі, який буде моніторитись скриптом, повинена бути створена окрема файлова система з відповідним каталогом, для зберігання third-party компонентів у ньому.

Створити окремий віртуальний сервер, з потрібною кількістю ресурсів, для роботи додатку.

Налаштувати автоматичну доставку та інсталяцію скрипту для збору потрібної інформації.

Додати скрипт до systemd, як демона та зробити його автоматичний старт при завантажені системи.

Додати відповідну job у crontab для моніторингу робити скрипта та відправки листа у разі його не правильної робити, або якщо процесу скрипта немає.

4.2.2 Опис алгоритму функціонування програми

Для опису алгоритму роботи програми було створено блок-схему, яка зображена на рисунку 4.1.

Вона відображає весь цикл життя додатка починаючи від входу у програму до виводу обробленої інформації на екрані.

4.3 Опис і обґрунтування вибору методу організації вхідних та вихідних даних

Створений додаток працює як з вхідними, так і з вихідними даними. Обов’язковими вхідними даними для роботи програми є імена хостів, з яких треба отримати версії third-party компонентів, імена директорій з third-party компонентами, які будуть перевірені, а також список версій, які вважаються останіми. Ці вхідні дані повинні бути заповнені у відповідному config файлі на хості з програмою. Методом зберігання і запису вхідних даних було обрано текстовий файл. Такий тип буде найпростішим для ведення його інженерами, які будуть використовувати програму у своїй роботі та які не були задіяні у розробці додатку.

Структура config файлу вхідних даних наведено на рисунку 4.2.

Вихідні дані, які програма отримує для після перевірки скриптами на віддалених хостах, було вирішено формувати у JSON та відавати це файл на фронт для виводу інформації з нього.

Рисунок 4.1 – Блок-схема алгоритму роботи програми

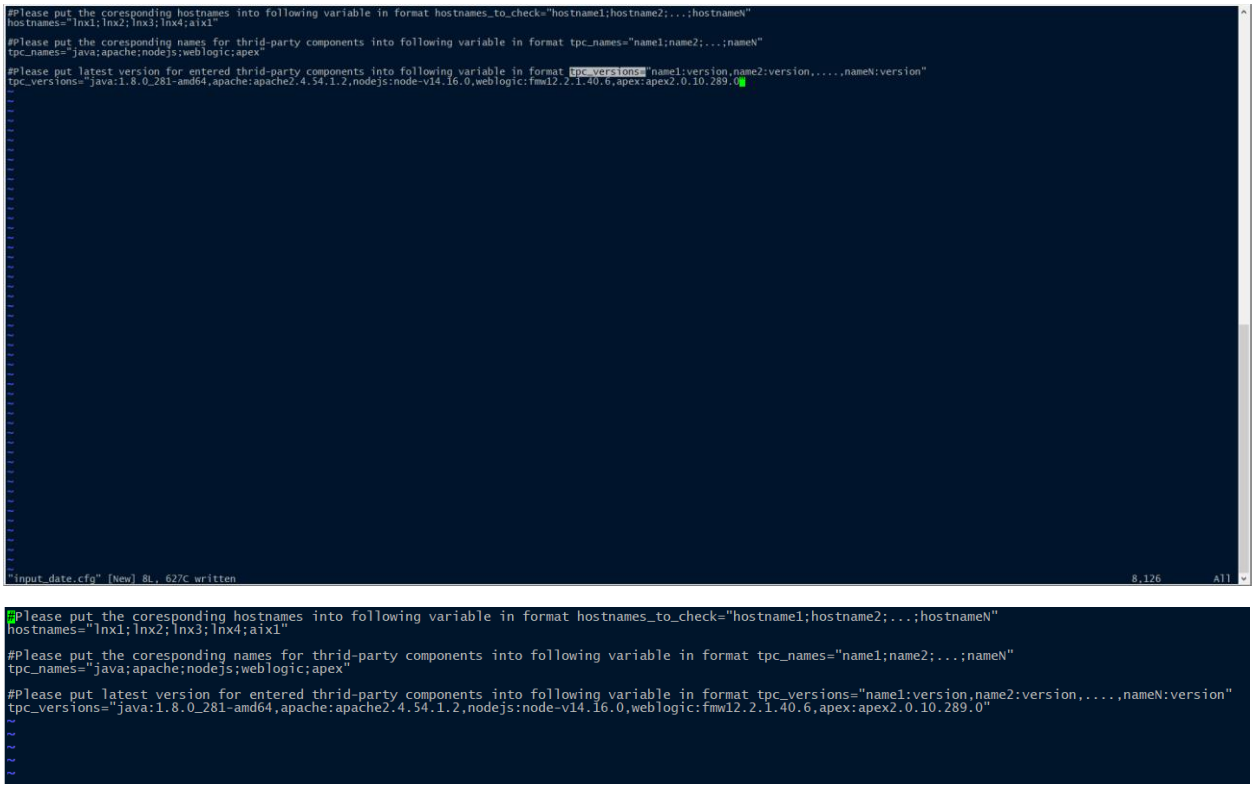


Рисунок 4.2 – Структура config файлу вхідних даних додатку

JSON (JavaScript Object Notation) — це текстовий формат обміну даними на основі JavaScript. Але при цьому формат не залежить від JS і може використовуватися в будь-якій мові програмування, також JSON використовується в REST API.

4.3.1 Опис і обґрунтування вибору та складу технічних та програмних засобів

У якості мови програмування для реалізації серверної частини додатку було обрано Python, останньої, на сьогоднішній день, версії 3.11.

Python – стрімко розвиваюча, скриптова мова, яка використовується для вирішення великої кількості найрізноманітніших проблем і завдань. Python корисний при створенні комп’ютерних і мобільних додатків, він використовується в роботі з великим обсягом інформації, при розробці

веб-сайтів і різних інших проектів, використовується в машинному навчанні. Цю мову програмування використовують великі відомі корпорації, такі як Spotify і Amazon (наприклад, для аналізу даних і створення алгоритму рекомендацій), YouTube і навіть Walt Disney. Таким чином, Python знайшов своє місце в різних сферах - з його допомогою можна вирішувати безліч завдань різної складності. Також для реалізації цього додатку було важливо, щоб у код можна було імпортувати bash та ksh скрипти, з чим Python дуже легко справляється.

Для реалізації клієнтської частини додатку було використано мову програмування JavaScript. Ця мова незамінна у реалізації front-end'у, бо це її пряме призначення.

JavaScript — це багатопарадигмальна мова програмування, яка зазвичай використовується як вбудований інструмент для програмного доступу до різних об'єктів програми. З точки зору веб-розробки, без знання цієї технології неможливо створювати сучасні інтерактивні сайти. JS – це те, що оживляє макет сторінки (HTML) і взаємодію з користувачем (CMS) сайтів. За допомогою цієї мови реалізується можливість реакції сторінки або окремих її елементів на дії відвідувача. Сьогодні JavaScript є основною мовою програмування для браузерів. Він повністю сумісний з операційними системами Windows, Linux, Mac OS, а також усіма популярними мобільними платформами.

4.4 Опис розробленої програми

4.4.1 Загальні відомості

Головне призначення додатку це демонстрація поточних версій third-party компонентів, для того щоб тримати їх up-to-date на усіх хостах.

4.4.2 Функціональне призначення

Програма призначена для збору інформації з віддалених хостів, та надсилання цієї інформації користувачу на екран у браузері.

4.4.3 Опис логічної структури програми

Програма поділяється на два модулі: серверний та клієнтський.

На серверному модулі виконується запити до інших, віддалених хостів та збір потрібної інформації у форматі JSON для передачі її на front для подальшої обробки.

На клієнтському модулі виконується отримання даних в відповіді запиту, після чого виконується парсинг цих даних, що прийшли у форматі JSON, і на їх основі будується таблиця з даними.

4.4.4 Використані технічні засоби

Для виконання даної програми необхідний окремо відділений сервер на базі Linux/Unix, або виділені ресурси на вже існуючому сервері, для опрацювання програми на ньому.

4.4.5 Виклик і завантаження програми

Після інсталяції програми на сервері, вона буде доступна у локальній мережі за посиланням: https://tcp/main.html. Далі треба розкласти скрипти на відповідних хостах, заповнити cofig файл для вхідних даних та завантажити сторінку.

Результат виконання програми можна побачити на рисунку 4.3.

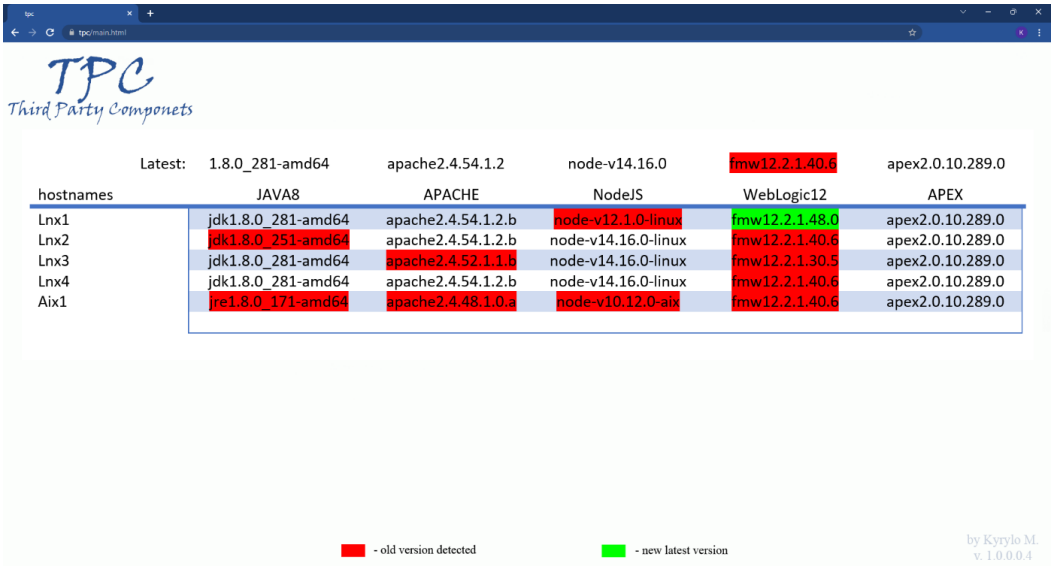


Рисунок 4.3 – Результат виконання програми у веб-браузері

4.5 Очікуванні техніко-економічні показники

Розроблений додаток дозволить моніторити та використовувати підприємству виключно up-to-date компоненти у своїх проектах, що в свою чергу гарантує наявність усіх останніх бібліотек і фрагментів, а також, що не мало важливо, мати встановлені останні патчі безпеки, які запровадять відтоку інформації та не потрібних проблем, які це може викликати. Наявність подібних утиліт для тримання усього софту, що використовує підприємство у статусі up-to-date, може бути гарантією для поточних та майбутніх клієнтів, а також першим кроком для отримання ліцензій стандартизації ISO та інших.

5 ЕКСПЕРЕМЕНТАЛЬНИЙ РОЗДІЛ

5.1 Розробка математичної моделі мережі як замкнутої системи масового обслуговування

На основі структурної схеми комп’ютерної мережі та її імітаційної моделі розроблено структуру математичної моделі комп’ютерної мережі як замкнутої системи масового обслуговування, яку можна побачити на рисунку 5.1.

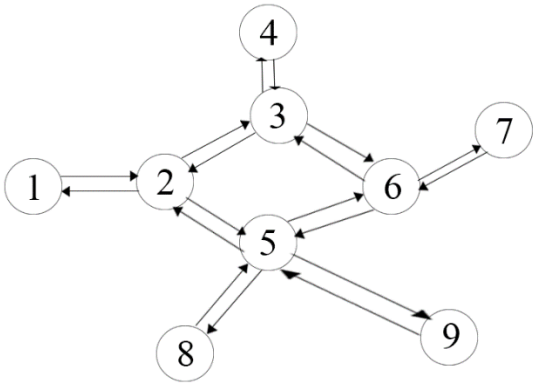


Рисунок 5.1 – Структура математичної моделі комп’ютерної мережі.

У структурі моделі комп'ютерної мережі вузли 1,4,7,8,9 є комутаторами, які обслуговують локальні мережі.

Вузли 2,3,5,6 є маршрутизаторами.

Взаємозв'язки між елементами структури є ймовірністю передачі пакетів від одного вузла до іншого. Кожен вузол є системою черги.

Ймовірність, що вузол спілкується сам із собою, дорівнює нулю.

Результатом є матриця маршрутизації.

$$Pr := \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.3 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.375 & 0 & 0.25 & 0 & 0.375 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.3 & 0 & 0.5 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Стовпець матриці

$$\tau = \begin{pmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

Визначає час обробки для одного повідомлення у відповідному вузлі.

5.2 Розрахунок параметрів мережі по її моделі

Потім методом Гауса обчислюємо матрицю-стовпець із коефіцієнтами передачі.

$$e := \begin{pmatrix} 1 \\ 5 \\ 4 \\ 1 \\ 10 \\ 5 \\ 1 \\ 2.5 \\ 2.5 \end{pmatrix}$$

Встановлюємо матрицю m, коефіцієнти якої означають кількість конвеєрів обробки пакетів для кожного з вузлів системи масового обслуговування.

$$m = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Для розрахунків ми припускаємо, що кожен пристрій має лише один конвеєр обробки пакетів.

Матриця B — це матриця, яка визначає, наскільки ймовірно відповідний вузол (рядки) чекатиме на обробку пакетів (номер стовпця).

$$B = \begin{pmatrix} 0.921 & 0.074 & 5.117 \times 10^{-3} & 2.869 \times 10^{-4} & 0 \\ 0.605 & 0.26 & 0.099 & 0.03 & 5.603 \times 10^{-3} \\ 0.684 & 0.23 & 0.068 & 0.016 & 2.295 \times 10^{-3} \\ 0.921 & 0.074 & 5.117 \times 10^{-3} & 2.779 \times 10^{-4} & 8.965 \times 10^{-6} \\ 0.21 & 0.25 & 0.253 & 0.197 & 0.09 \\ 0.605 & 0.26 & 0.099 & 0.03 & 5.603 \times 10^{-3} \\ 0.921 & 0.074 & 5.117 \times 10^{-3} & 2.779 \times 10^{-4} & 8.965 \times 10^{-6} \\ 0.802 & 0.164 & 0.029 & 4.132 \times 10^{-3} & 3.502 \times 10^{-4} \\ 0.802 & 0.198 & 0.034 & 4.482 \times 10^{-3} & 3.502 \times 10^{-4} \end{pmatrix}$$

За методом Бузена розраховуються середні значення для кожного з вузлів мережі.

Інтенсивність потоку вхідних пакетів на кожному вузлі.

$$\lambda = \begin{pmatrix} 0.013 \\ 0.066 \\ 0.052 \\ 0.013 \\ 0.131 \\ 0.066 \\ 0.013 \\ 0.033 \\ 0.033 \end{pmatrix}$$

Середня кількість пакетів, що очікують на обробку, на вузол

$$L = \begin{pmatrix} 0.085 \\ 0.572 \\ 0.423 \\ 0.085 \\ 1.707 \\ 0.572 \\ 0.085 \\ 0.236 \\ 0.28 \end{pmatrix}$$

Середній час обробки пакетів у вузлі

$$t = \begin{pmatrix} 6.465 \\ 8.726 \\ 8.075 \\ 6.466 \\ 13.029 \\ 8.726 \\ 6.466 \\ 7.21 \\ 8.546 \end{pmatrix}$$

5.2.1 Параметри роботи мережі

Нормальний режим роботи комп'ютерної мережі можна охарактеризувати наступними параметрами.

Кількість пакетів, що циркулюють у мережі, становить 5. Час обробки пакетів у всіх окремих вузлах мережі становить 5 годин (для досліджуваної мережі 1 година дорівнює 1 мілісекунді). Кількість етапів обробки пакетів у кожному вузлі мережі дорівнює 1.

З такими вихідними даними були отримані графіки, які відображають середні характеристики кожного з вузлів.

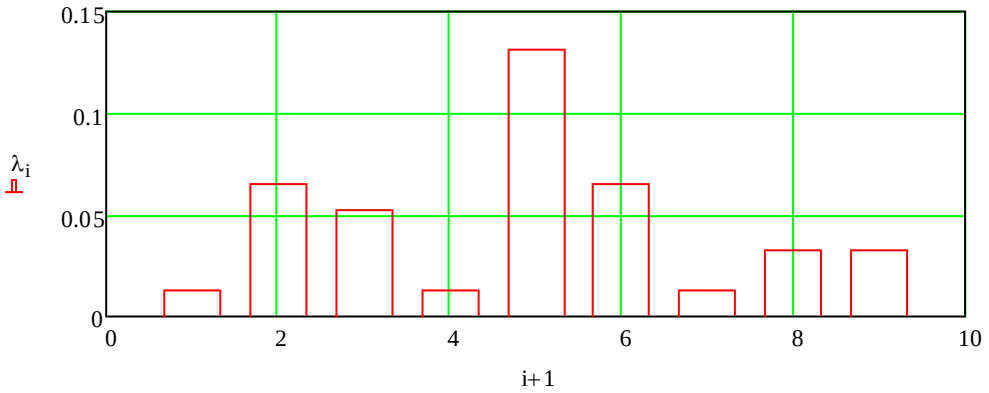


Рисунок 5.2 - Інтенсивність потоку, що надходить у вузол

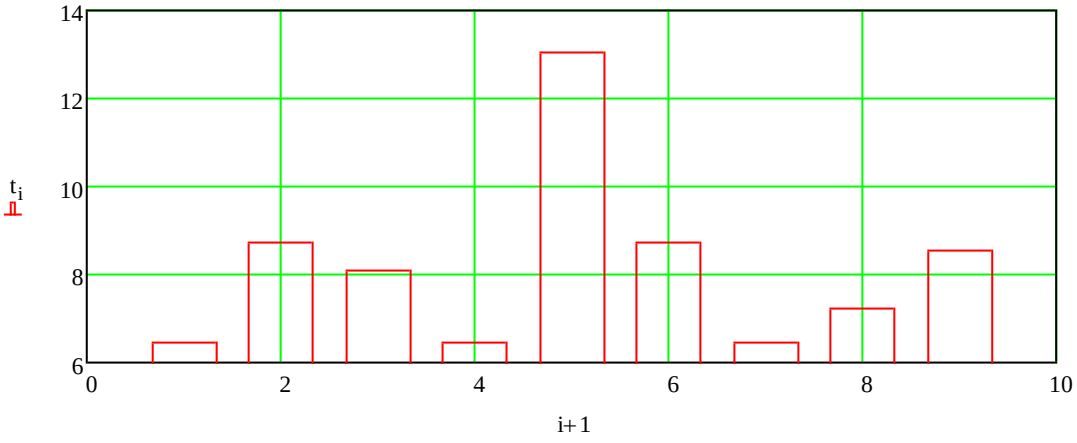


Рисунок 5.3 – Середній час перебування пакета у вузлі

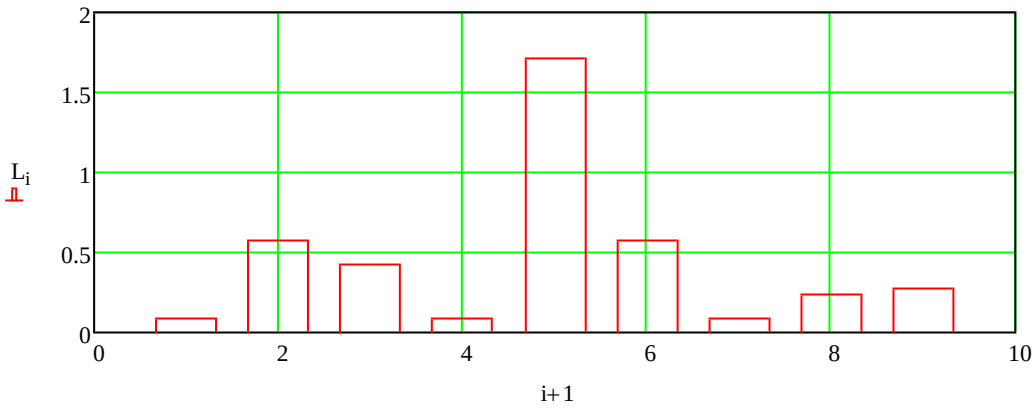
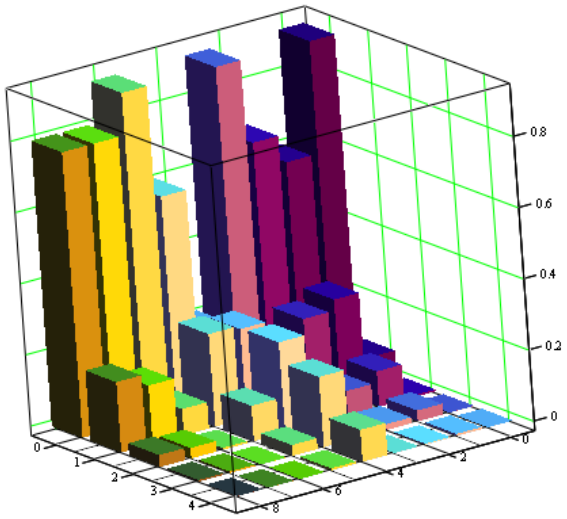


Рисунок 5.4 - Середня кількість пакетів у вузлі

Як бачимо, в цілому по всіх вузлах мережі, які є комутаторами, усереднені параметри показують, що всі повідомлення обробляються швидко і без черги. Виняток становлять вузли 2,3,5,6 що відповідають за маршрутизацію всередині мережі.

На малюнку 5.5 показано, з якою ймовірністю виникне черга у вузлах мережі.



в

Рисунок 5.5 – Вірогідність черги у вузлах мережі

Як можна зазначити, при заданих параметрах маршрутизатири є найбільш проблемним елементом мережі

5.3 Робота мережі зі скоригованими характеристиками проблемних вузлів

Корекція характеристик вузлів № 2,5,6 здійснюється шляхом збільшення швидкості обробки пакетів.

$$\tau = \begin{pmatrix} 5 \\ 3 \\ 5 \\ 5 \\ 2 \\ 3 \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

Відповідно до змін розраховуються усереднені та ймовірнісні характеристики.

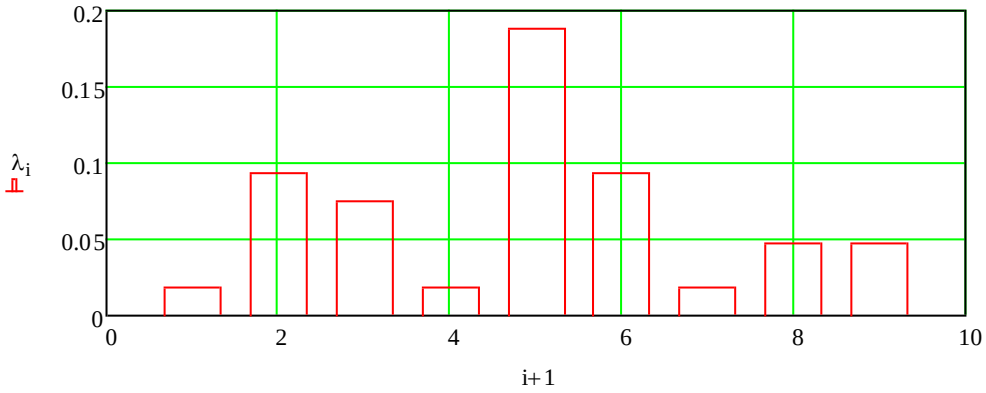


Рисунок 5.9 - Інтенсивність потоку, що надходить у вузол

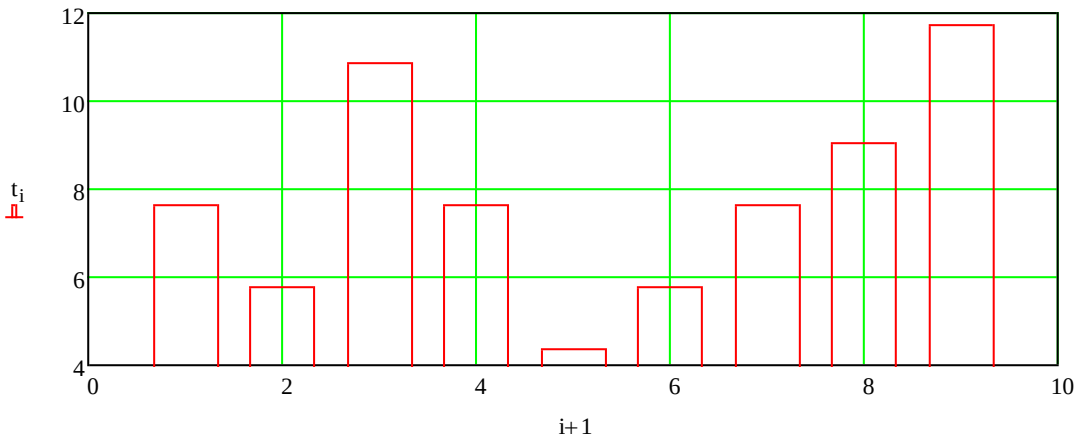


Рисунок 5.10 – Середній час перебування пакета у вузлі

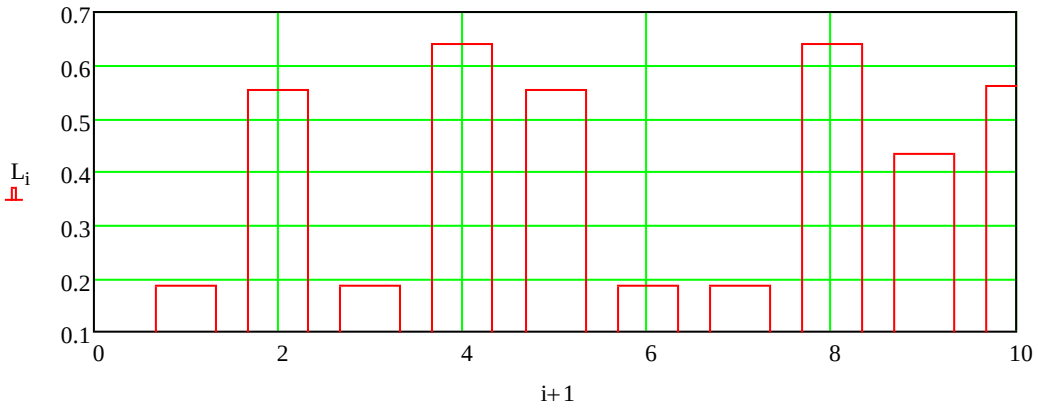
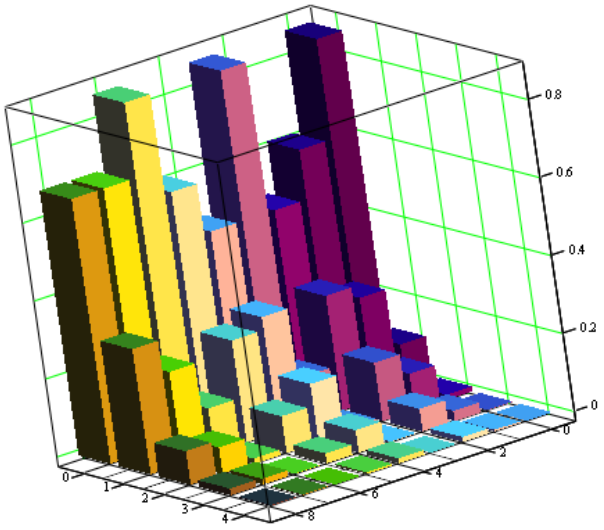


Рисунок 5.11 - Середня кількість пакетів у вузлі

Достовірність того, що у вузлах мережі може з'явитися черга, значно знизилася, як зазначено на рисунку 5.12.



B

Рисунок 5.12 – Ймовірність черги у вузлах, якщо в мережі циркулює 5 пакетів

Якщо в мережі з жорсткими характеристиками циркулює 5 пакетів, то для маршрутизаторів зменшується ймовірність відсутності черги на вузлах, що можна бачити на рисунку 5.13.

Збільшення швидкості обробки пакетів у вузлах, які мали найменшу стійкість до перевантаження, дозволило певним чином підвищити продуктивність мережі.

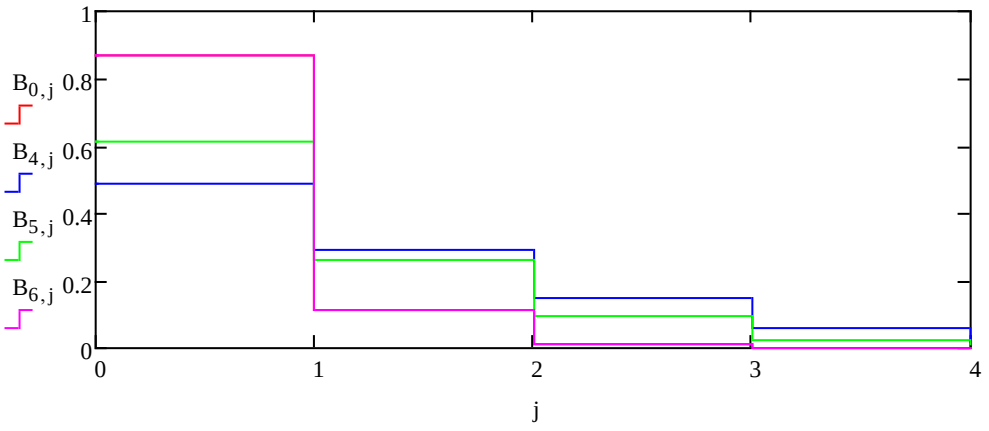


Рисунок 5.13 – Ймовірність черги у вузлах скорегованої мережі

Згідно з аналізом даних, отриманих у результаті дослідження стану мережі лише під впливом шкідливих програм, стохастичний характер матриці маршрутів, що описує мережу, може викликати нелінійне зростання основних характеристик у деяких вузлах мережі, хоча навантаження на вузли мережі зростатиме лінійно. Це явище може викликати збої в комп'ютерній мережі.

Аналіз характеристик розглянутої мережі під впливом тільки AFS показує, що через запуск антивірусної програми на вузлах мережі з досить високим середнім навантаженням каналів основні характеристики можуть істотно погіршуватися.

За результатами порівняння характеристик комп’ютерної мережі в усіх розглянутих станах було зроблено висновок, що найбільш негативно на характеристики мережі впливає атака шкідливого програмного забезпечення.

ВИСНОВКИ

Детально проаналізувавши комп’ютерну систему та мережу підприємства виконано наступні дії для її покращення:

1. Проаналізувавши комп’ютерну систему, було вирішено додати новий сервер, потужніший сервер для створення нових віртуальних локальних ресурсів та інтегрувати хмарні потужності для різних задач, зокрема резервних ресурсів.

2. Для автоматизації та полегшення розробки програмного забезпечення у підприємстві, було запровадження методологію DevOps з її практиками.

3. Маючі нові ресурси, було створено відокремленні віртуальні сервери для реалізації різних бізнес потреб: веб-сервер, сервер баз даних, було створено корпоративний VPN сервер, а також реалізовано корпоративний поштовий сервер.

4. Було створено та реалізовано веб-додаток для моніторингу версій third-party компонентів, завдяки якому потенційно можна підвищити безпеку підприємства та відкрити шлях для здобування сертифікацій стандартизації ISO та інших.

5. Проаналізувавши комп’ютерну мережу підприємства, було виявлено, що мережа є одним з ключових елементів комп’ютерної системи підприємства з точки зору продуктивної розробки проектів. Але як показало дослідження, що мережа не має можливості виконувати сповіщення про інформаційне перенавантаження.

ПЕРЕЛІК ПОСИЛАНЬ

1. Литвинов А. Л. Теорія систем масового обслуговування: навч. посібник / А. Л. Литвинов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків: ХНУМГ ім. О. М. Бекетова, 2018. – 141 с..

2. Документація Oracle Datebase [Електронний ресурс] – Режим доступу: URL : https://www.oracle.com/jp/database/technologies/oracle-database-documentation.html

3. Інструкція з інсталяції Oracle Datebase [Електронний ресурс] – Режим доступу: URL : https://docs.oracle.com/cd/F32587_01/cncpt/introduction-to-oracle-database.html#GUID-CF765A7D-9429-4901-BF33-36E0B0220293

4. Документація Mozila Foundation [Електронний ресурс] – Режим доступу:URL:https://developer.mozilla.org/ru/docs/Learn/Common_questions/What_is_a_web_server

5. Документація веб-серверу APACHE [Електронний ресурс] – Режим доступу: URL : https://httpd.apache.org/dev/devnotes.html

6. Документація IBM [Електронний ресурс] – Режим доступу: URL : https://www.ibm.com/docs/ru/power10?topic=9080-hex-power-e1080

7. Довідник з організації корпоративного поштового серверу Google [Електронний ресурс] – Режим доступу: URL : https://workspace.google.com/intl/uk/products/gmail

8. Документація пакету для моделювання мережі EVE-NG [Електронний ресурс] – Режим доступу: URL : https://www.eve-ng.net/index.php/features/

9. Документація пакету для моделювання мережі GNS3 [Електронний ресурс] – Режим доступу: URL : https://docs.gns3.com/docs/

10. Структурне моделювання комп’ютерних мереж [Електронний ресурс] – Режим доступу: URL : https://tekhnosfera.com/strukturnoe-

modelirovanie-kompyuternyh-setey-s-ispolzovaniem-raspredelennyh-informatsionno-vychislitelnyh-sistem

11. Брайн Керніган, Роб Пайк UNIX Програмна середа. – Wiley, 2022, – 478с.

12. Кристофер Негус Linux Bible. – Wiley, 2022, – 928 с.