

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Технологія IPTV з використанням протоколів РІМ та IGMP»

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Великодний Д.В.

Студент гр. ІН–61

Мороз Е.В.

Суми 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІКТ

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2019 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи ІН-61 спеціальності “Інформатика”
денної форми навчання Мороза Едуарда Вадимовича.

Тема: “Технологія IPTV з використанням протоколів РІМ та IGMP”

Затверджена наказом по СумДУ

№ _____ від _____ 2019 р.

Зміст пояснювальної записки: 1) аналітичний огляд технології IPTV;
2) постановка задачі і формування завдань дослідження; 3) Вивчення GNS3 та
його можливостей моделювання IPTV 4) розробка програмного забезпечення
веб-орієнтованої системи; 5) аналіз результатів виконання програм.

Дата видачі завдання “ _____ ” _____ 2019 р.

Керівник випускної роботи _____ Великодний Д.В.

Завдання прийняв до виконання _____ Мороз Е.В.

РЕФЕРАТ

Записка: 60 стор., 30 рис., 15 джерел інформації.

Об'єкт дослідження – Технологія IPTV.

Предмет дослідження – особливості роботи та налаштування елементів мережі Multicast, як компонентів технології IPTV

Мета роботи — Створення та налаштування мережі Multicast у симуляторі GNS3 з можливістю трансляції через неї IPTV трафіку

Методи дослідження – моделювання технології IPTV у симуляторі GNS3

Результати — розроблено веб-орієнтовану інформаційну систему в яку можна ввести дані: ір-адреси та маски для всіх інтерфейсів. Через буфер обміну їх можна скопіювати та використовувати для налаштування схем.

IPTV, GNS3, Multicast, PIM, CISCO

Зміст

ВСТУП.....	5
1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	6
1.1 Динамічна маршрутизація. Основні алгоритми.....	6
1.2. Технологія IPTV	8
1.3 Протокол PIM	9
1.4 IGMP	11
1.5 Постановка задачі.....	16
2. ТЕХНОЛОГІЇ РОЗВ'ЯЗКУ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	17
2.1 Конфігурація мережі на базі роутерів Cisco.....	17
2.2 Конфігурація мережі в симуляторі GNS3 на базі обладнання Cisco та налаштування віртуальних машин для перевірки.	18
2.3 Розробка веб-орієнтованої використовуючи мову програмування JavaScript	24
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	27
3.1 Розробка графічного інтерфейсу налаштування протоколів динамічної маршрутизації	27
3.2 Тестування веб-орієнтованої інформаційної системи в симуляторі GNS3 та на реальному обладнанні Cisco	29
ВИСНОВКИ	36
СПИСОК ЛІТЕРАТУРИ	37
ДОДАТОК.....	38
Додаток А.....	38

ВСТУП

В сучасній сфері інтернету дуже важливо мати сервіс для побудови мережі IPTV, який би підходив всім користувачам інтернету. Адже це не тільки можливість перегляду інтернет телебачення у високій якості, але й можливість вибрати що дивитись, та коли.

Завданням наукової роботи було обрано сервіс для написання конфігурації мережі для побудови реальної мережі. На сьогодні не так багато аналогів даного сервісу, бо в основному провайдери займаються подібними налаштуваннями мереж.

Але в даному випадку сервіс можна буде приєднати до будь-якого сайту та отримати інструмент для налаштування мережі з інтернет телебаченням, це дасть змогу спростити процес налаштування для рядового користувача, та обійтись без допомоги майстра, адже мета створення даного сервісу, в першу чергу – спрощення проектування мережі.

Для розробки сервісу будемо використовувати засоби мови програмування JavaScript та GNS3 для моделювання реальної мережі. В процесі розробки ми спроектуємо реальну мережу з конфігурацією, щоб пізніше перенести реальну конфігурацію в веб-сервіс.

Веб-орієнтована інформаційна система, завдяки якій конфігурація роутерів для динамічної маршрутизації в мережах Ethernet стане легше, це і є рішення даної проблеми.

В рамках всієї роботи необхідно провести аналіз предметної області, проаналізувати уже існуючі технології, вивчити їх слабкі і сильні сторони, щоб створити власний проект.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Динамічна маршрутизація. Основні алгоритми.

Маршрутизація - є однією з ключових функцій мережного рівня EMBVC (базова еталонна модель взаємодії відкритих систем, англ. Open Systems Interconnection Basic Reference Model) [1]. Таким чином, під маршрутизацією мається на увазі, перш за все, процес визначення в телекомунікаційній мережі одним або набором способів (маршрутів), оптимальних в межах обраних критеріїв, між заданою парою або набором мережевих вузлів. Таким чином, шлях - це послідовність мережевих вузлів та шляхів передачі, які з'єднують задану пару мережевих вузлів.

Основними цілями маршрутизації є мінімізація (максимізація) значень вибраних показників якості обслуговування (швидкість передачі, середня затримка, втрата пакету тощо), а також забезпечення збалансованого навантаження мережі, її каналу та буфера ресурси. Тому основними завданнями, пов'язаними з галуззю маршрутизації, є: контроль та збір інформації про стан мережі (її топологію, завантаження мережевих ресурсів тощо), розрахунок необхідних шляхів (маршрутів) та реалізація рішень маршрутизації.

Завдання маршрутизації зважається на основі аналізу таблиць маршрутизації, розміщених у всіх маршрутизаторах та кінцевих вузлах мережі. Основна робота зі створення таблиць маршрутизації виконується автоматично, але також надається можливість вручну коригувати або доповнювати таблицю.

Використовуючи протоколи маршрутизації, маршрутизатори відображають мережеві з'єднання з різним ступенем деталізації. Виходячи з цієї інформації, для кожного номера мережі приймається рішення про те, якому наступному маршрутизатору слід надсилати пакети, надіслані на цю адресу, щоб маршрут був раціональним. Результати таких рішень заносяться до таблиці маршрутів. Коли ви змінюєте конфігурацію мережі, деякі записи в

таблиці стають недійсними. У таких випадках пакети, надіслані помилковими маршрутами, можуть стати петельними та втраченими. Якість всієї мережі залежить від того, наскільки швидко протокол маршрутизації вирівнює вміст таблиці з реальним станом мережі.

Маршрутизатори зберігають та оновлюють у таблицях маршрутизації наступну важливу інформацію:

- тип протоколу, тобто інформація про протокол маршрутизації, який зробив запис у таблиці маршрутизації;
- повідомлення одержувача / наступного вузла, яке сповіщає маршрутизатор, що приймаючий вузол або безпосередньо підключений, або до нього можна дістатися через інший маршрутизатор, наступний стрибок на шляху до пункту призначення. Маршрутизатор аналізує адресу одержувача у вхідних пакетах і порівнює його, щоб відповідати записам у таблиці маршрутизації;
- метрики маршрутизації;

вихідний інтерфейс - інтерфейс, через який потрібно надсилати дані для досягнення пункту призначення.

Протокол маршрутизації - це протокол, який підтримує мережеві протоколи та забезпечує механізми обміну інформацією про маршрутизацію.

Критерієм вибору шляху між заданою парою мережевих вузлів є мінімальна або максимальна його "довжина" (вага, вартість), задана як сума "довжин" шляхів передачі, які формують цей шлях. "Довжина" шляху передачі з точки зору протоколів маршрутизації називається його метрикою (метрикою маршрутизації). Існуючі протоколи маршрутизації, залежно від особливостей вирішеної проблеми маршруту, використовують досить широкий перелік метрик, що характеризують різні властивості шляху передачі, а саме його фізичну довжину, надійність, пропускну здатність, навантаження, вартість.

1.2. Технологія IPTV

IPTV розшифровується як "Інтернет-протокол телебачення". "IP" в IPTV такий же, як у вашій IP-адресі або VoIP (голос через IP). Все, що означає, що телевізійне програмування передається за допомогою Інтернет-протоколу.[2]

Щоб зрозуміти, що це означає, потрібно трохи знати про те, як працює не-IPTV. За допомогою кабельного або супутникового телебачення мовники розсилають сигнали, а глядачі отримують їх - ви можете лише дивитися, що транслюється. Якщо у вас є якийсь пристрій запису, ви не можете диктувати, що робити коли. Ви просто налаштуєте, коли зможете переглядати наявні.

IPTV відрізняється. Замість передачі вмісту за допомогою світлових імпульсів волоконно-оптичного кабелю або радіохвиль із супутника IPTV надсилає шоу та фільми через стандартне підключення до Інтернету. (Можливо, ви використовуєте кабельне або супутникове підключення до Інтернету, але вони не залежать від тих, які зазвичай передають ваші телевізійні сигнали.)

Замість того, щоб транслювати низку шоу за певним розкладом, більшість IPTV використовує відео на вимогу (VOD) або зміщену в часі медіа - ми розглянемо ці, а також третій формат, лише за мить.

За цим процесом стоїть якась складна мережева архітектура, яка включає багато перекодування від традиційних сигналів до сприятливих для IP-адрес. Але важливо, що вам не потрібно дивитися, що транслюється. Ви можете сказати своєму постачальнику, що ви хочете переглянути, і він негайно надішле вам його.

Якщо ви використовували потоковий сервіс, як Netflix або Hulu, це та сама ідея, але з телебаченням замість фільмів чи синдікованих шоу.

1.3 Протокол PIM

Protocol Independent Multicast направляє багатоадресні пакети до багатоадресних груп і призначений для ефективного встановлення розподілів дерев через глобальну мережу (WANs). PIM називається "незалежним протоколом", оскільки він може використовувати інформацію про маршрут, яку будь-який протокол маршрутизації доставляє в адресну базу даних багатоадресної маршрутизації (RIB), або, як відомо, в термінології Windows, в режимі багатоадресного перегляду. Приклади цих протоколів маршрутизації включають одноадресні протоколи, такі як протокол маршрутизації інформації (RIP) і перший відкритий найкоротшим шляхом (OSPF), але багатоадресні протоколи, які заповнюють таблиці маршрутизації, такі, як протокол дистанції вектора багатоадресної маршрутизації (DVMRP), також можна використовувати. Розріджений режим означає, що протокол розроблено для ситуацій, коли багатоадресні групи є тонко розповсюдженими через великий регіон. Розріджений режим Протоколи розрідженого режиму можуть виконуватися у середовищах LAN, але вони найбільш ефективні в глобальних мережах.[3]

Розріджену групу можна визначити як "ту, в якій:

- кількість мереж або доменів, в яких присутні члени групи, значно менша, ніж кількість мереж / доменів в Інтернеті;
- члени групи охоплюють занадто велику / широку область, щоб покладатися на обмеження кількості стрибків або на якусь іншу форму обмеження сфери розповсюдження багатоадресних пакетів;
- робота в Інтернеті не є достатньо багатою на ресурси, щоб ігнорувати накладні витрати на поточний щільний режим.

На відміну від цього, протоколи у щільному режимі, такі як DVMRP та Multicast OSPF (MOSPF), розроблені для ситуацій, коли групи багатоадресної передачі широко представлені і пропускна здатність є достатньою. За допомогою цих схем, пакети даних та / або інформаційний звіт про членство можуть надмірно надсилатися на інтерфейси, які не призводять до джерел

багатоадресної передачі або зацікавлених приймачів; крім того, маршрутизатори зберігають пов'язаний стан для цих незацікавлених вузлів, що також непотрібно. Ці накладні витрати є прийнятним, коли більшість хостів зацікавлені в даних і є достатня пропускна здатність для підтримки потоку контрольних повідомлень, але в іншому випадку є неефективною. PIM-SM припускає, що жоден хост не хоче даних, якщо це прямо не запитується. Однак PIM має аналог у щільному режимі (PIM-DM), сумісний із розрідженим режимом.

PIM був розроблений для підтримки наступних цілей:

Підтримка традиційної IP - багатоадресної моделі обслуговування приймача, ініційованої членством багатоадресної групи. У даній моделі джерела просто поставляють пакети на перший-хоп Ethernet, без будь-яких сигналів. Приймачі подають сигнал маршрутизаторам, щоб приєднатися до групи багатоадресної передачі, яка отримує дані.

Залишити модель хоста незмінною. PIM-SM - це протокол типу маршрутизатор - маршрутизатор, що означає, що хости не потребують постійного оновлення, але маршрутизатори з підтримкою PIM-SM повинні бути розгорнуті в мережі.

Підтримка як спільних, так і вихідних дерев розповсюдження. Для спільних дерев PIM-SM використовує центральний маршрутизатор, який називається Точка рандеву (RP), як корінь спільного дерева. Всі хости-джерела надсилають свій трафік багатоадресної передачі в RP, яка, в свою чергу, пересилає пакети через загальне дерево всім членам групи. Джерельні дерева безпосередньо підключають джерела до приймачів. Для кожного джерела є окреме дерево. Дерев-джерела вважаються деревами, що мають найкоротший шлях, з точки зору таблиць маршрутів одноадресної передачі. PIM-SM може використовувати будь-який тип дерева або обидва одночасно.

Підтримувати незалежність від будь-якого конкретного протоколу маршрутизації одноадресної передачі.

Використовуйте механізми м'якого стану для адаптації до мінливих умов мережі та групової багатоадресної динаміки. М'який стан означає, що, якщо вона не оновлено, то конфігурація стану маршрутизатора є короткостроковою та закінчується через певний час.

1.4 IGMP

Протокол управління групами Інтернету IGMP (Internet Protocol Management Protocol) - протокол управління груповою (багатоадресною) передачею даних у мережах, заснованих на протоколі IP. IGMP використовує маршрутизатори та IP-узлами для організацій мережевих пристроїв у групі. [4]

Протокол IGMP використовується виключно при взаємодії безпосередньо пов'язаних один з одним маршрутизатора і хоста, коли останній виступає (або бажає виступати) в ролі одержувача трафіку групового мовлення.

До основних функцій протоколу IGMP відносяться оповіщення маршрутизатора про бажання хоста бути включеним в групу і опитування членів групи.

Щоб стати одержувачем групових даних, вузол повинен «висловити» свою зацікавленість маршрутизатора, до якого безпосередньо приєднана його мережу. Для цього хост повинен встановити взаємодію з маршрутизатором по протоколу IGMP. Версія IGMP для хоста прямо залежить від типу операційної системи, встановленої на хості. Так, ранні версії Windows (Windows 95) підтримували тільки версію IGMPv1, більш пізні (Windows 2000) - версію IGMPv2, а починаючи з Windows XP, підтримується версія IGMPv3. Протоколи IGMPv2 і IGMPv3 підтримуються в багатьох версіях Mac OS, Linux, Unix-подібних операційних системах. Опитування членів групи: для виконання цієї функції один з маршрутизаторів локальної мережі вибирається домінуючим. Домінуючий маршрутизатор засобами протоколу IGMP періодично опитує всі системи (груповий адресу 224.0.0.1) в безпосередньо

приєднаних до нього підмережах, перевіряючи, чи активні члени всіх відомих йому груп. Решта (не вибрані) маршрутизатори прослуховують мережу, і якщо виявляють відсутність повідомлень-запитів протягом деякого періоду (зазвичай 250 секунд), то повторюють процедуру вибору нового домінуючого маршрутизатора.

У IGMPv2 визначено три типи повідомлень:

Запит про членство (*membership query*). За допомогою цього повідомлення маршрутизатор намагається дізнатися, до якої групи складаються хости в локальній мережі, приєднаної до якого-небудь його інтерфейсу. Запит про членство існує в двох варіантах: в одному з них маршрутизатор робить загальний запит про всі групи, в іншому його цікавить інформація тільки про деяку конкретній групі, адреса якої вказується в запиті.

Звіт про членство (*membership report*). Цим повідомленням хости відповідають маршрутизатора, який послав у мережу запит про членство. У повідомленні міститься інформація про адресу групи, до якої вони належать. Маршрутизатор, будучи членом усіх груп, отримує повідомлення, спрямовані на будь-яку групову адресу. Для маршрутизатора, який отримує відповідні повідомлення, важливий тільки факт наявності членів тієї чи іншої групи (груп), а не приналежність конкретних хостів конкретних груп. Цей факт буде використаний іншими маршрутизаторами мережі для просування пакетів групового мовлення в ту частину мережі, за яку «відповідає» даний маршрутизатор. Звіт про членство хост може послати не тільки у відповідь на запит маршрутизатора, але і з власної ініціативи, коли він намагається приєднатися до певної групи. Після такого повідомлення хост може розраховувати на те, що трафік для цієї групи дійсно буде доставлятися в мережу, до якої цей хост належить.[4]

Покинути групу (*leave group*). Це повідомлення хост може використовувати, щоб сигналізувати «власного» маршрутизатора про бажання покинути деяку групу, в якій він до цього перебував. Отримавши це повідомлення, маршрутизатор посилає специфічний запит про членство

членам тільки цієї конкретної групи, і якщо не отримує на нього відповідь (тобто це був останній хост в групі), то перестає передавати трафік групового мовлення для цієї групи. Слово «може» означає в даному випадку, що хост може бути виключений з групи, просто не відповідаючи маршрутизатора на запит про членство (такий підхід реалізований в протоколі IGMPv1). Тоді маршрутизатор буде продовжувати передавати небажаний трафік групового мовлення до тих пір, поки не закінчиться деякий період часу з моменту надходження останнього звіту про членство. Такий підхід може значно подовжити період прихованого знаходження хоста в стані виходу з групи, що знижує ефективність роботи мережі.

Повідомлення з запитами про членство надсилаються маршрутизатором регулярно з певною частотою. На кожному з інтерфейсів до встановлених засобами IGMP маршрутизаторами підтримуються кеш-таблиці груп. Кеш-таблиця містить список усіх груп, в складі яких є хоча б один член. Для кожного рядка таблиці встановлений тайм-аут. Маршрутизатор регулярно посилає запити (за замовчуванням - кожні 125 секунд), щоб перевірити, що в кожній групі ще є члени. Якщо для деякої групи відповідь не надходить протягом встановленого для неї тайм-ауту, то відповідний рядок видаляється з кеш-таблиці, і маршрутизатор вважає, що членів цієї групи в мережі більше немає.

Локальна мережа може мати декілька хостів, зацікавлених в отриманні трафіку однієї і тієї ж групи, але маршрутизатору досить підтвердження тільки від одного хоста для того, щоб продовжити передавати трафік в мережу для цієї групи. При використанні протоколу IGMPv1 або IGMPv2 для обмеження числа відповідей хостів на запит маршрутизатора будь-який хост, що знаходиться в групі, замість того щоб негайно відповісти на запит, спочатку чекає протягом деякого інтервалу часу, чи не з'явиться в мережі відповідь якогось іншого хоста. Якщо після закінчення цього часу він так і не зміг дочекатися появи в мережі відповіді іншого хоста, то він посилає маршрутизатору власний звіт про членство. Якщо ж використовується

протокол IGMPv3, то ніяких пауз не встановлюється, і хости відразу генерують повідомлення про членство.

Грунтуючись на інформації, отриманої за допомогою IGMP, маршрутизатори можуть визначати, в які підключені до них мережі необхідно передавати груповий трафік.

Всі типи IGMP-повідомлень мають довжину 8 байт і складаються з чотирьох полів. Залежно від версії протоколу IGMP призначення полів може дещо змінюватися. На рис. 1 показана структура повідомлення для версії IGMPv2.



Рис. 1.1 - Структура IGMP-повідомлення [4].

Поле максимального часу відповіді використовується хостами для обчислення часу затримки відповіді. Час затримки вибирається випадковим чином з інтервалу від нуля до значення, заданого в цьому полі.

Зауважимо, що поле адреси групового мовлення в IGMP-повідомленні не містить адресу призначення, воно несе в собі інформацію, по-різному використовується в різних типах повідомлень. Наприклад, маршрутизатор, посилаючи запит про членство, поміщає в цьому полі нулі, а хост в повідомленнях «Звіт про членство» і «Покинути групу» поміщає в це поле адресу групи, в яку він хоче вступити або яку він хоче покинути відповідно.

Щоб хост зміг отримувати трафік групового мовлення, недостатньо встановити на ньому протокол IGMP, за допомогою якого хост може відправити повідомлення свого маршрутизатора про бажання приєднатися до групи. Крім цього, треба конфігурувати мережевий інтерфейс хоста так, щоб він став захоплювати з локальної мережі кадри, що несуть в собі пакети групового мовлення для тієї групи, до якої приєднався хост. Для цього необхідно налаштувати інтерфейс на прослуховування певного групової адреси канального рівня, відповідного групового IP-адресу. На жаль, адресний простір групових IP-адрес в 32 рази об'ємніше простору групових MAC-адрес. Тобто відображення цих двох адресних просторів виявляється далеко неоднозначним - на один і той же груповий MAC-адреса відображається цілий блок з 32 різних групових IP-адрес. Отже, коли мережевий адаптер захоплює кадр, що містить пакет групового мовлення, існує значна ймовірність того, що цей пакет був спрямований зовсім іншій групі. Однак ця помилка швидко виявляється. Коли кадр передається вгору по стеку, протокол IP перевіряє, чи збігається груповий IP-адрес в полі адреси призначення інкапсульованого пакета з груповим IP-адресом даного інтерфейсу.

1.5 Постановка задачі

Проаналізувавши зібрані дані, мету роботи можна сформулювати наступним чином: необхідно створити веб-орієнтовану інформаційну систему, графічний інтерфейс якої буде дозволяти налаштувати мережеві інтерфейси роутерів та IPTV з протоколами PIM та IGMP без потреби від користувача знань побудови подібних мереж. Має гарантувати зручне копіювання та перенесення отриманого набору команд налаштувань в реальне обладнання Cisco або симулятор GNS3.

Програмне забезпечення не повинно вимагати спеціальних знань в командах налаштування обладнання Cisco, при цьому бути направленим на допомогу початківцям, а саме полегшене налаштування мережі для подальшого використання їх. Графічний інтерфейс не повинен бути перевантажений складними елементами, мати інтуїтивно-зрозумілий дизайн.

Графічний інтерфейс має бути реалізований у вигляді веб-сторінки для легшої інтеграції її у майбутньому, на сторінці буде можливість ввести необхідні вхідні дані і в результаті отримати згенерований набір налаштувань під конкретну мережу, яку задає користувач, також інтерфейс повинен передбачати можливість копіювання та перенесення набору команд в симулятор GNS3. У результаті роботи з цим програмним забезпеченням користувач отримує повністю працюючу мережу.

Постановка задачі:

1. Конфігурація мережі в симуляторі GNS3 на базі обладнання Cisco та налаштування віртуальних машин для перевірки.
2. Розробка графічного інтерфейсу налаштування IPTV з протоколами PIM та IGMP
3. Тестування розробленої веб-орієнтованої інформаційної системи в симуляторі GNS3 та на реальному обладнанні Cisco.

2. ТЕХНОЛОГІЇ РОЗВ'ЯЗКУ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Конфігурація мережі на базі роутерів Cisco

Cisco Systems, Inc. - відома американська транснаціональна компанія, безпосередньою діяльністю якої є розробка і продаж мережевого спеціального обладнання.[5]

Компанія прагне представити своїм покупцям якомога ширший спектр сучасного оригінального устаткування, щоб покупець здійснив покупку всього необхідного саме у одній єдиній компанії одноразово - у Cisco Systems.

Cisco виробляє велику кількість різноманітних пристроїв, наприклад:

- Маршрутизатор
- Ethernet-комутатори
- Платформи оптичної комутації
- DSL-обладнання
- Великі системи відеоконференцій TelePresence (на основі обладнання поглиненої компанії Tandberg).
- Сервери (UCS, поставляються, зокрема, в складі комплексів FlexPod, що випускаються Cisco спільно з NetApp)
- Програмне забезпечення управління мережею
- Системи відеонагляду
- Комутатори мереж зберігання даних
- АТМ-комутатори
- Точки доступу Wi-Fi
- Програмне та апаратне забезпечення для IP- телефонії
- Пристрої мережевої безпеки (міжмережеві екрани, VPN, IDS та ін.)
- кабельні модеми
- Універсальні шлюзи і шлюзи віддаленого доступу

Ресурси Cisco - це найбільший в галузі збірник апаратних засобів, що використовуються для побудови інформаційних мереж і надання доступу до

них, плюс операційна система Cisco IOS (призначена для підтримки мережевих послуг і додатків), розгортання мереж і досвід проектування, а також система технічної підтримки та професійних послуг по обслуговуванню і оптимізації мережевої роботи. Унікальність Cisco в тому, що вона здатна - самостійно або разом з партнерами - надати всі перераховані вище ресурси.

Cisco обслуговує замовників на трьох цільових ринках: Великі підприємства, тобто великі організації зі складними мережевими потребами. Вони мають безліч відділень, де встановлені різноманітні комп'ютерні системи. У цю групу замовників входять корпорації, державні установи, комунальні служби і навчальні заклади. Сервіс-провайдери - компанії, які надають інформаційні послуги, в тому числі оператори зв'язку, провайдери доступу в Інтернет, кабельні компанії і оператори бездротового зв'язку. Малі та середні підприємства, які потребують власних мережах для передачі даних і в засобах доступу в Інтернет.

2.2 Конфігурація мережі в симуляторі GNS3 на базі обладнання Cisco та налаштування віртуальних машин для перевірки.

Проблема комп'ютерного моделювання телекомунікаційних систем сьогодні має безліч рішень різного роду. Деякі з популярних продуктів - OPNET, OMNET ++, NS2, NS3, які є потужним інструментом моделювання завдяки об'єктно-орієнтованим мовам програмування як вбудованої мови для опису моделей телекомунікаційних систем. Існують також вузькоспеціалізовані тренажери, призначені лише для моделювання певного обладнання. Таке програмне забезпечення виробляють виробники телекомунікаційного обладнання.

CISCO Systems, виробник мережевого обладнання, пропонує програмне забезпечення мережевого моделювання, яке дозволяє студентам експериментувати з різними топологіями та їх поведінкою всередині: Packet Tracer, Dynamips, GNS3-тренажерами, що є графічним інтерфейсом для симулятора Dynamips.

Середовище моделювання GNS3 дозволяє будувати складні мережеві проекти, перевіряти ефективність топології, визначати кількість мережевого обладнання, перевіряти роботу віртуальної комп'ютерної мережі. Забезпечити наявність необхідної кількості пристроїв та різних типів обладнання майже неможливо, оскільки це вимагає значних матеріальних витрат. Вирішення цієї проблеми може бути через використання програмного забезпечення для моделювання та моделювання, що дозволяє працювати з відповідними пристроями, налаштовувати їх, тестувати, організовувати їх роботу.

Що стосується налаштування комп'ютерної мережі, то потрібно заздалегідь знати, як вона буде працювати, адже обладнання, яке потрібно придбати, не з дешевих. Для того, щоб мати змогу імітувати взаємодію між різними елементами мережі, ви можете використовувати програми для емуляції її. GNS3 (Графічний мережевий симулятор) дозволяє в режимі реального часу моделювати віртуальну мережу будь-якої топології без необхідності реального мережевого обладнання.

Особливість GNS3 полягає в тому, що він є кросплатформним симулятором і може використовуватися на декількох платформах, включаючи підтримувані Windows, Linux, Mac. GNS3 має графічний інтерфейс, що дозволяє вибрати кожен з елементів, що входять до мережі, та встановити його параметри. Це значно полегшує створення віртуальних лабораторій, оскільки заздалегідь можна визначити кількість мережевого обладнання, його продуктивність у певних умовах, конфігурацію пристроїв. Важливою перевагою GNS3 є простота та зручність створення проектів для лабораторних робіт. Залежно від апаратної платформи, на якій буде використовуватися GNS3, ви можете створювати складні проекти, що складаються з маршрутизаторів Cisco, Cisco ASA, Juniper, а також серверів, що працюють під управлінням мережевих операційних систем. GNS3 працює з зображеннями IOS, які використовуються як програмне забезпечення на пристроях Cisco. Ви також можете отримати доступ до реальної мережі та використовувати віртуальні машини VirtualBox.

Для розробки графічного інтерфейсу налаштування IPTV з початку змодельюємо реальну мережу з реальною конфігурацією в GNS3.

Створимо мережу яка складається з серверу (Multicast Server, 172.16.0.5/24) і клієнта (Multicast Client, 192.168.4.2), який буде транслювати сигнал в мультикастового адреса 224.1.1.1. А між ними буде знаходитися зв'язуючий маршрутизатор Cisco 3640.

Належність того чи іншого клієнта до певної групи «мовлення» відстежується протоколом IGMP (Internet Group Management Protocol). Передача multicast працює по протоколу PIM (Protocol Independent Multicast). Цей протокол складає свою таблицю маршрутизації multicast на основі звичайної таблиці маршрутизації.

Є 3 режими роботи даного протоколу:

- dense mode - коли маршрут прокладається безпосередньо від джерела до одержувача (пропозиція мовлення потоків, зареєстрованих на роутері, йде всім, а потім відсікаються ті, кому вона не потрібна);
- sparse mode - маршрут прокладається тільки до так званої точки рандеву RP (Rendezvous point). Трансляція потоку починається, тільки якщо є запит від клієнта;
- sparse-dense mode - змішаний режим.

Як джерело потоку будемо програвати .mp4 файл в VLC media player. Приймати потік ми будемо за допомогою нього ж.

В рамках цієї роботи налаштуємо sparse mode.

Створюємо нашу мережу як показано на рисунку 2, де XP-1 це клієнт, XP-8 сервер з якого ми будемо проводити трансляцію, та R1 маршрутизатор який зв'язує їх

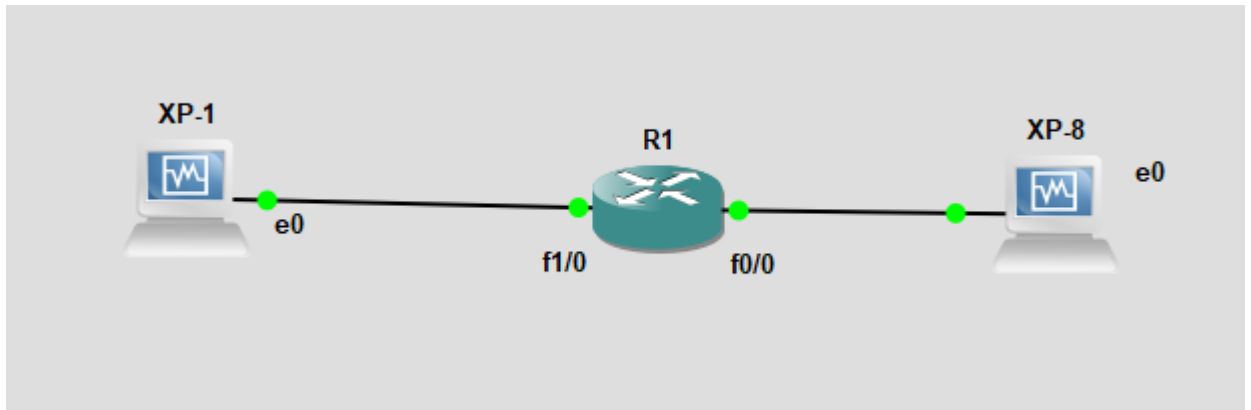


Рисунок 2.1 - Приклад мережі

Розпочнемо з конфігурації маршрутизатора R1, заходимо до його налаштувань через консоль telnet та прописуємо йому такі команди:

```

ip multicast-routing (глобально вмикаємо маршрутизацію multicast)
interface FastEthernet0/0
ip address 172.16.0.1 255.255.255.0
ip pim sparse-mode (включаємо потрібний режим на вибраному
інтерфейсі)
no sh
exit
interface FastEthernet1/0
ip address 192.168.4.1 255.255.255.0
ip pim sparse-mode
no sh
exit
ip pim rp-address 172.16.0.1

```

потім для наглядного прикладу пропишемо IP віртуальним машинам, як показано на рисунку 2.2 для машини клієнта

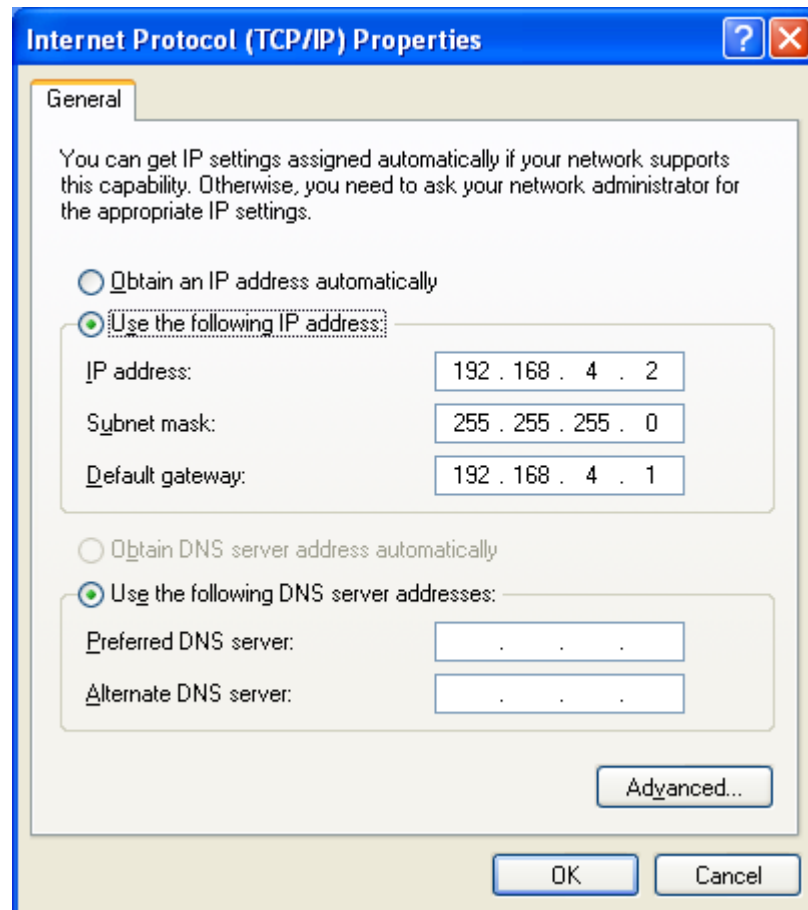


Рисунок 2.2 - Налаштування IP для клієнта

На цьому приготування для трансляції трафіку закінчені, перейдемо до перевірки нашої конфігурації.

Перейдемо до програми VLC на віртуальній машині сервера.

Заходимо до налаштувань трансляції та вибираємо файл який ми хочем транслювати. (Рисунок 2.3) та переходимо на наступних налаштувань де в напрямку транслювання вибираємо RTP/MPEG Transport Stream з включеним перекодуванням Video – H.264 + MP3 (MP4). Для RTP вводимо адресу багатоадресної передачі 224.2.2.4 з портом 5004 та в ході налаштування виставляємо TTL в залежності від мережі, в нашому випадку стандартного 1 досить. Потім заходимо з машини клієнта до цієї ж програми, але в цей раз знаходимо пункт відкрити URL і водимо IP адресу яку ми зазначили раніше `rtp://@224.2.2.4:5004`

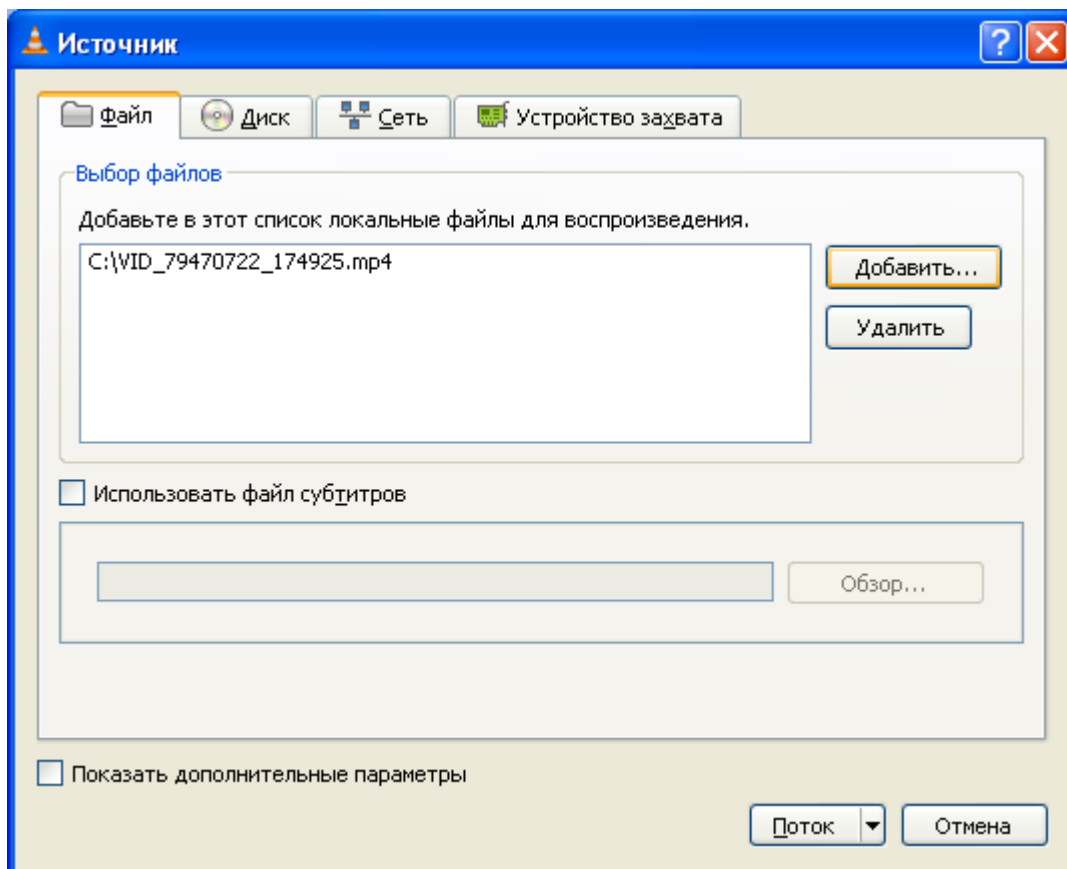


Рисунок 2.3 - Вікно вибору файлу для трансляції.

Після приєднання ми можемо спостерігати що відео успішно транлюється до клієнта разом зі звуком рисунок 2.4

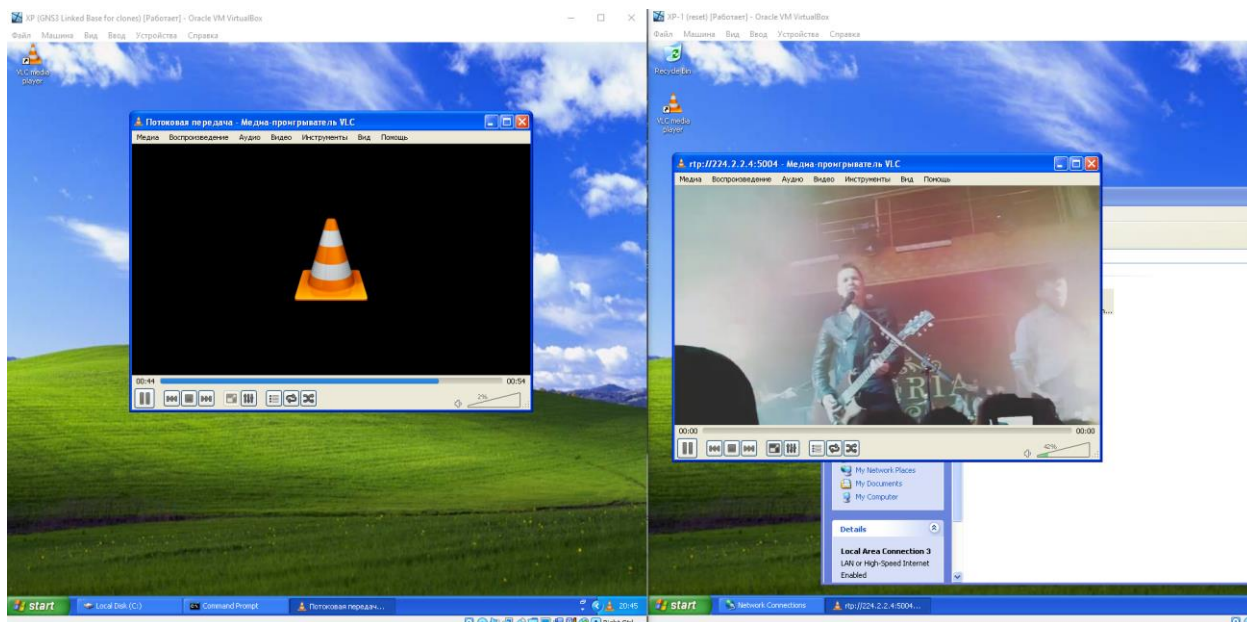


Рисунок 2.4 - Проверка трансляції трафіку

GNS3 є дуже важливим і необхідним інструментом для мережевих адміністраторів, оскільки він дозволяє відтворити конфігурацію реального пристрою за допомогою емульованого, що полегшує роботу проектування мереж та допомагає уникати помилок в реальній мережі.

2.3 Розробка веб-орієнтованої використовуючи мову програмування JavaScript

Програма була розроблена з використанням мови JavaScript.

JavaScript спочатку був створений, щоб "оживити веб-сторінки".

Програми цією мовою називаються скриптами. Їх можна записати прямо в HTML веб-сторінки та запустити автоматично під час завантаження сторінки. [6]

Сценарії надаються та виконуються як звичайний текст. Для запуску їм не потрібна спеціальна підготовка чи компіляція.

Сьогодні JavaScript може виконуватись не тільки в браузері, але і на сервері або фактично на будь-якому пристрої, який має спеціальну програму під назвою the JavaScript engine.

У браузері є вбудований двигун, який іноді називають "віртуальною машиною JavaScript".

У різних двигунах є різні "кодові назви". Наприклад:

V8 - в Chrome і Opera.

SpiderMonkey - у Firefox.

... Існують і інші кодові назви, такі як «Trident» та «Chakra» для різних версій IE, «ChakraCore» для Microsoft Edge, «Nitro» та «SquirrelFish» для Safari тощо.

Двигуни складні. Але основи легкі.

Двигун (вбудований, якщо це браузер) читає ("розбирає") сценарій.

Потім він перетворює ("компілює") скрипт на машинну мову.

І тоді машинний код працює досить швидко.

Двигун застосовує оптимізації на кожному етапі процесу. Він навіть переглядає скомпільований сценарій під час його запуску, аналізує дані, що протікають через нього, та додатково оптимізує машинний код на основі цих знань.

Сучасний JavaScript - це "безпечна" мова програмування. Він не надає низькорівневий доступ до пам'яті або процесора, оскільки він був створений для браузерів, які цього не потребують.

Можливості JavaScript значною мірою залежать від середовища, в якому він працює. Наприклад, Node.js підтримує функції, які дозволяють JavaScript читати / записувати довільні файли, виконувати мережеві запити тощо.

В браузері JavaScript можна робити все, що стосується маніпуляції веб-сторінками, взаємодії з користувачем та веб-сервером.

Наприклад, JavaScript у браузері здатний:

Додати новий HTML на сторінку, змінити існуючий вміст, змінити стилі.

Реагувати на дії користувача, виконувати натискання миші, рухи вказівника, натискання клавіш.

Надсилати запити по мережі на віддалені сервери, завантажувати та вивантажувати файли (так звані технології AJAX та COMET).

Отримувати та встановлювати файли cookie, задавати питання відвідувачеві, показувати повідомлення.

Запам'ятовувати дані на стороні клієнта («локальне зберігання»).

JavaScript дуже схожий на мову C++ що стосується синтаксису, але в порівнянні з мовою C++ має такі корінні відмінності:

- JavaScript - мова вищого рівня, ніж C++, що вважається середнім рівнем. Це означає, що JavaScript ближче до людської мови, ніж C++, що ближче до одиниць і нулів. Це означає, що JavaScript може виконувати більше рядків, ніж C++, а код JavaScript, набагато коротший, ніж C++
- У C++ є те, чого немає у JavaScript: налаштування. Якщо ви кодуєте в C++, ви кодуєте на середньому рівні, це означає, що менше дороги

прокладено для вас. Якщо ви хочете зробити щось для інших платформ, окрім Windows, у вас буде більше роботи.

- Хоча для C ++ існує досить багато попереднього коду, не завжди обіцяється, що ви знайдете інтеграцію, яку шукаєте, і вам не доведеться кодувати щось на замовлення, щоб відповідати вашій програмі. Навколо JavaScript є багато готових рішень, майже у всіх є браузер, а всі сучасні браузери працюють з JavaScript.

- C ++ також є статичною мовою. Це означає, що ви повинні оголошувати кожен тип змінної під час кодування. JavaScript, з іншого боку, динамічна мова, що означає, що змінні можуть змінювати тип під час виконання. Вам не потрібно так сильно турбуватися про те, чи є щось у рядку чи цілим числом у JavaScript, і тому ви можете ввести його трохи швидше, ніж ви на C ++.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка графічного інтерфейсу налаштування протоколів динамічної маршрутизації

Мережа Ethernet була сконфігурована в емуляторі GNS3, де були зроблені налаштування роутерів та multicast передача за допомогою протоколу RIPv2 та визначена приналежність клієнта до певної групи трансляції використовуючи протокол IGMP. Ці налаштування були зроблені на реальному обладнанні Cisco. За отриманими результатами проведеної роботи було виявлено, що значним недоліком симулятора GNS3 є те, що в ньому відсутній графічний інтерфейс який би спрощував конфігурацію мультисервісних задач. Також рутинне налаштування багатьох маршрутизаторів власноруч займають багато часу. Саме тому на сьогодні є актуальною задача розробки веб-орієнтованого графічного інтерфейсу для автоматичної конфігурації multicast передачі трафіку.

Проект був розроблений за допомогою мови програмування JavaScript та мови розмітки гіпертексту HTML, а також з використанням каскадних таблиць стилів CSS.

Інтерфейс веб-сторінки вийшов інтуїтивно зрозумілим

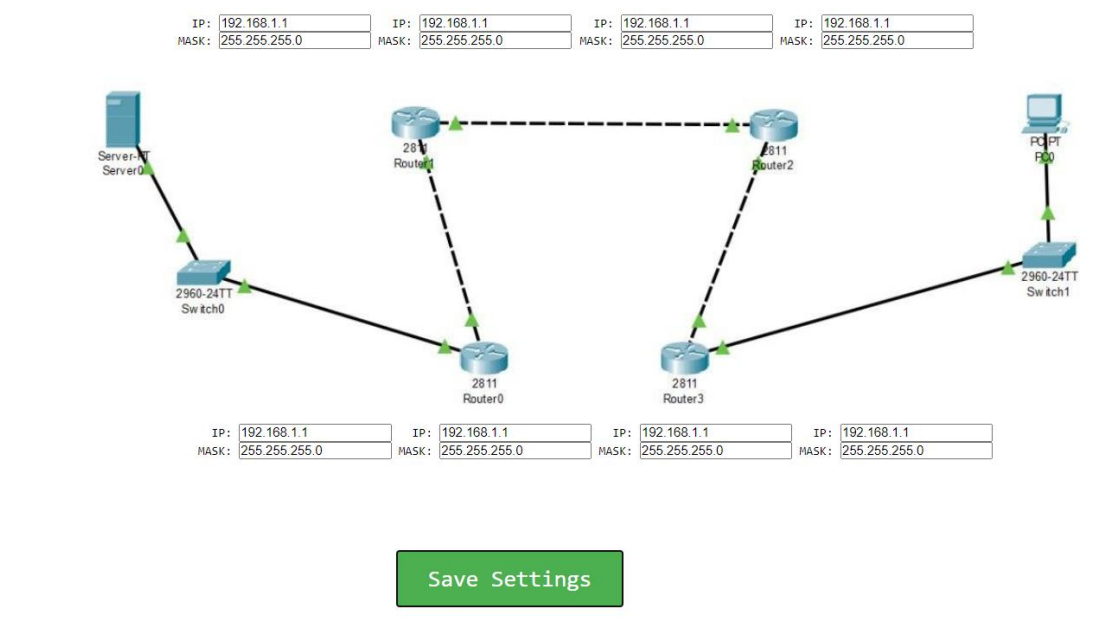


Рисунок 3.1 - Інтерфейс веб-орієнтованого графічного додатку

Відкривши веб-сторінку, спостерігаємо мережу Ethernet, кнопку “Save Settings”, а нижче ми можемо бачити форми з початковими налаштуваннями маршрутизаторів.

Адміністратор повинен заповнити відповідні поля кожного маршрутизаторів де розташування полів відповідає інтерфейсу для якого відбувається заповнення інформації. Після того як всі будуть заповнені, натискаємо на клавішу “Save Settings”. На рисунку 2.4. отримуємо бажані налаштування для кожного з маршрутизаторів у вигляді команд налаштування, також для зручності передбачена кнопка “Copy Settings”, яка передбачає копіювання коду до буфера обміну системи.

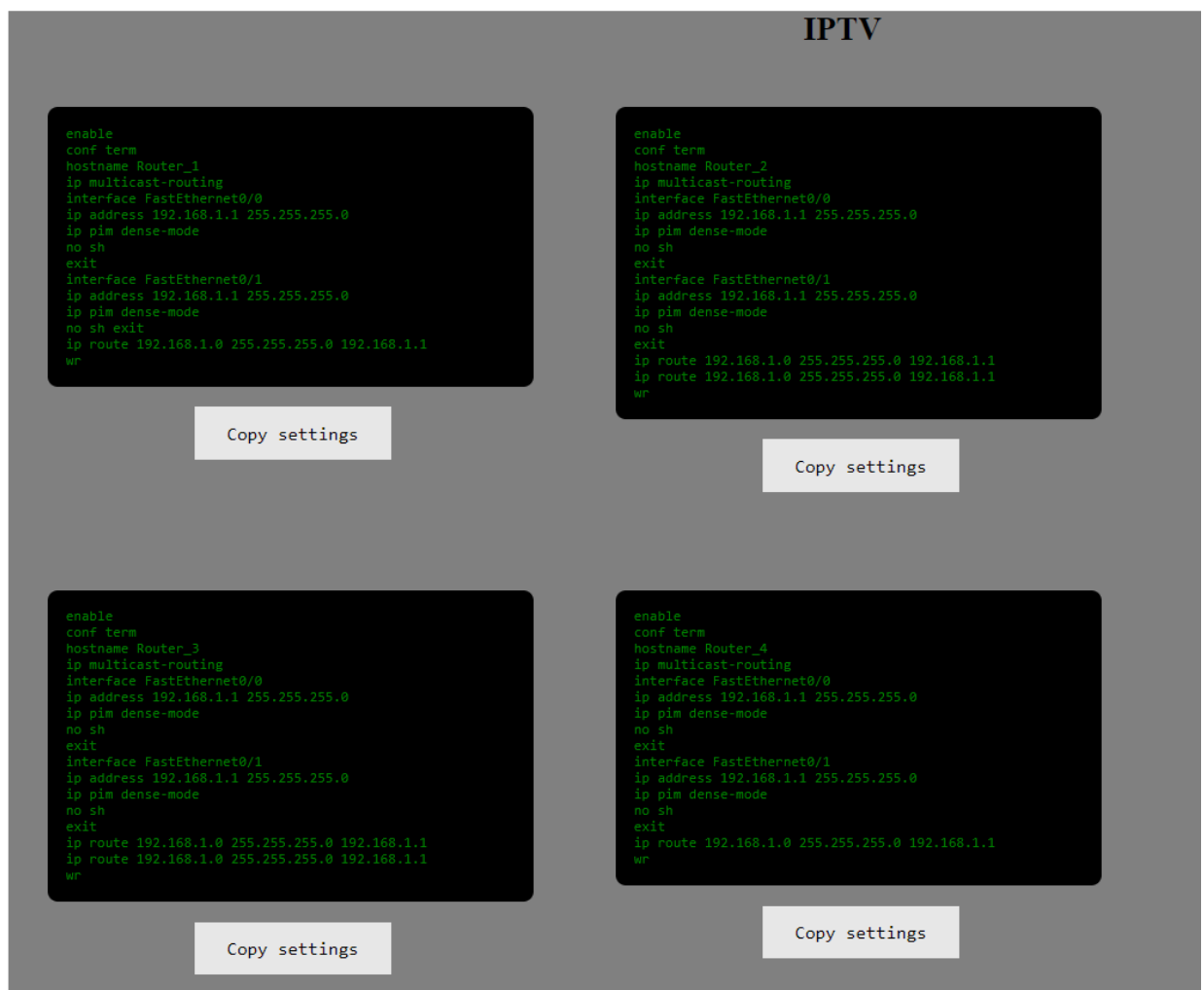


Рисунок 3.2 - Згенеровані налаштування Multicast

Після отримання відповідного набору команд Адміністратор має можливість внести ці зміни до вікна налаштувань маршрутизатора.

3.2 Тестування веб-орієнтованої інформаційної системи в симуляторі GNS3 та на реальному обладнанні Cisco

Для того щоб перевірити працездатність розробленої веб-орієнтованої інформаційної системи потрібно її протестувати в симуляторі GNS3 та на реальному обладнанні Cisco.

Перевіримо працездатність системи на можливість транлювати Multicast трафік. За допомогою розробленого графічного інтерфейсу заповнюємо IP та MASK поля (рис. 3.3).

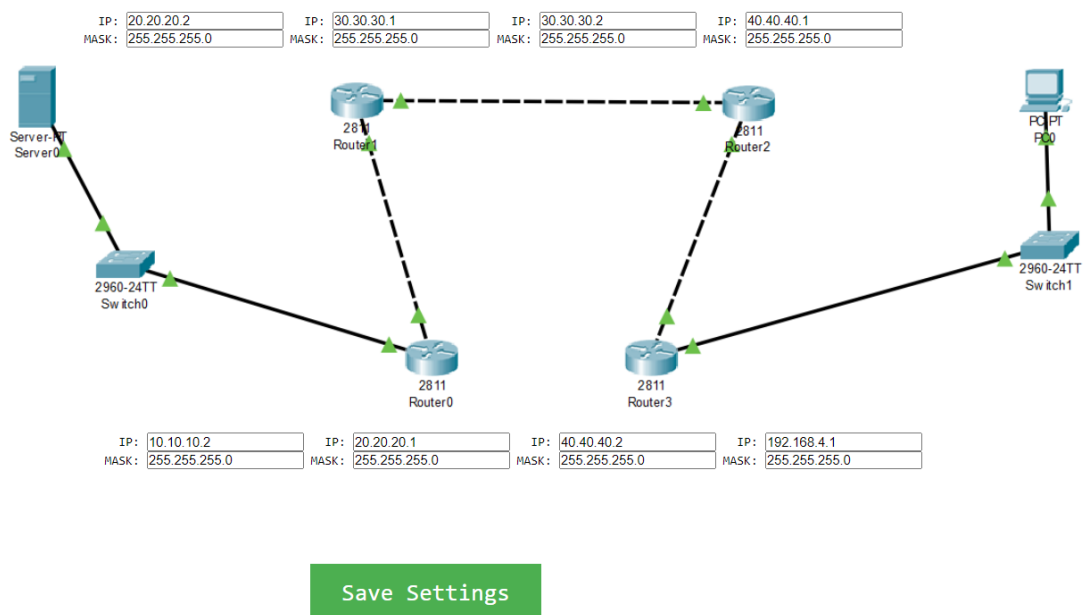


Рисунок 3.3 - Заповнені поля для роутерів у графічному інтерфейсі

Натискаємо кнопку “Save Settings” та отримуємо необхідні користувачу налаштування для кожного з маршрутизаторів зображених на нашій схемі.



Рисунок 3.4 - Згенеровані набори команд для кожного з роутерів

Після того як ми отримали набір команд для кожного маршрутизатора ми переходимо симулятора GNS3, де відтворюємо схему зображену на нашій веб-системі (рис.3.5)

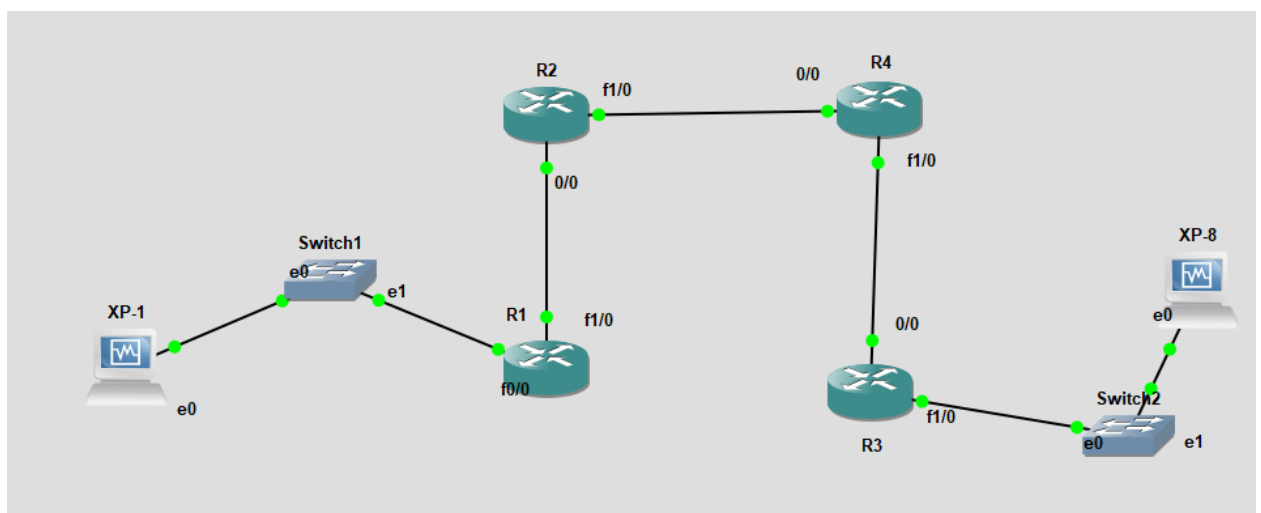


Рисунок 3.5 - Схема тестованої мережі у симуляторі GNS3

Після відтворення схеми переходимо до переносу налаштувань до роутерів.

Розпочнемо з роутера який позначений на схемі як “R1”

```

R1
e0/0, changed state to down
*Mar 1 00:00:03.858: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
e1/0, changed state to down
*Mar 1 00:00:03.858: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
e2/0, changed state to down
R1#enable
R1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#hostname Router_1
Router_1(config)#ip multicast-routing
Router_1(config)#interface FastEthernet0/0
Router_1(config-if)#ip address 10.10.10.2 255.255.255.0
Router_1(config-if)#ip pim dense-mode
Router_1(config-if)#no sh
Router_1(config-if)#exit
Router_1(config)#interface FastEthernet1/0
Router_1(config-if)#ip address 20.20.20.1 255.255.255.0
Router_1(config-if)#ip pim dense-mode
Router_1(config-if)#no sh
Router_1(config-if)#exit
Router_1(config)#ip route 192.168.10.0 255.255.255.0 20.20.20.2
Router_1(config)#wr
*Mar 1 00:00:38.991: %PIM-5-DRCHG: DR change from neighbor 0.0.0.0 to 10.10.10.2 on interface FastEthernet0/0
*Mar 1 00:00:39.107: %PIM-5-DRCHG: DR change from neighbor 0.0.0.0 to 20.20.20.1 on interface FastEthernet1/0
Router_1(config)#wr
*Mar 1 00:00:39.755: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:00:40.007: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Mar 1 00:00:40.755: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router_1(config)#wr
*Mar 1 00:00:41.007: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
Router_1(config)#

```

Рисунок 3.6 - Вікно налаштування роутера

Аналогічно налаштовуємо всі роутери за допомогою отриманих команд. Переконаємося у виконанні всіх команд використовуючи команду “show configuration”

```

R1
interface FastEthernet0/0
 ip address 10.10.10.2 255.255.255.0
 ip pim dense-mode
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 20.20.20.1 255.255.255.0
 ip pim dense-mode
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
ip route 192.168.4.0 255.255.255.0 20.20.20.2
ip route 192.168.10.0 255.255.255.0 20.20.20.2
!
!
!
no cdp log mismatch duplex
!
!
!
control-plane
--More--

```

Рисунок 3.7 - Результат команди “show configuration”

Після того як всі роутери були налаштовані і переконавшись, що налаштування бралися з нашої веб-сторінки, переходимо до перевірки нашої мережі, оскільки сервер (Multicast Server) та клієнт (Multicast Client)

знаходяться в різних мережах. Отже якщо конфігурація всіх роутерів коректна, то ми повинні мати змогу відправити команду “ping” від серверу до клієнта або навпаки.

Для цієї перевірки в нашій схемі ми використовуємо віртуальні машини, які працюють під управлінням операційної системи Windows XP.

Перша віртуальна машина буде виступати у ролі Multicast Server для транслявання відеофайлу формату .mp4 за допомогою плеєра VLC. Переглянемо налаштування серверу (рис. 3.8) і клієнта (рис. 3.9.), а також можемо бачити що команда “ping” до нашого клієнта пройшла успішно і без втрат пакетів.

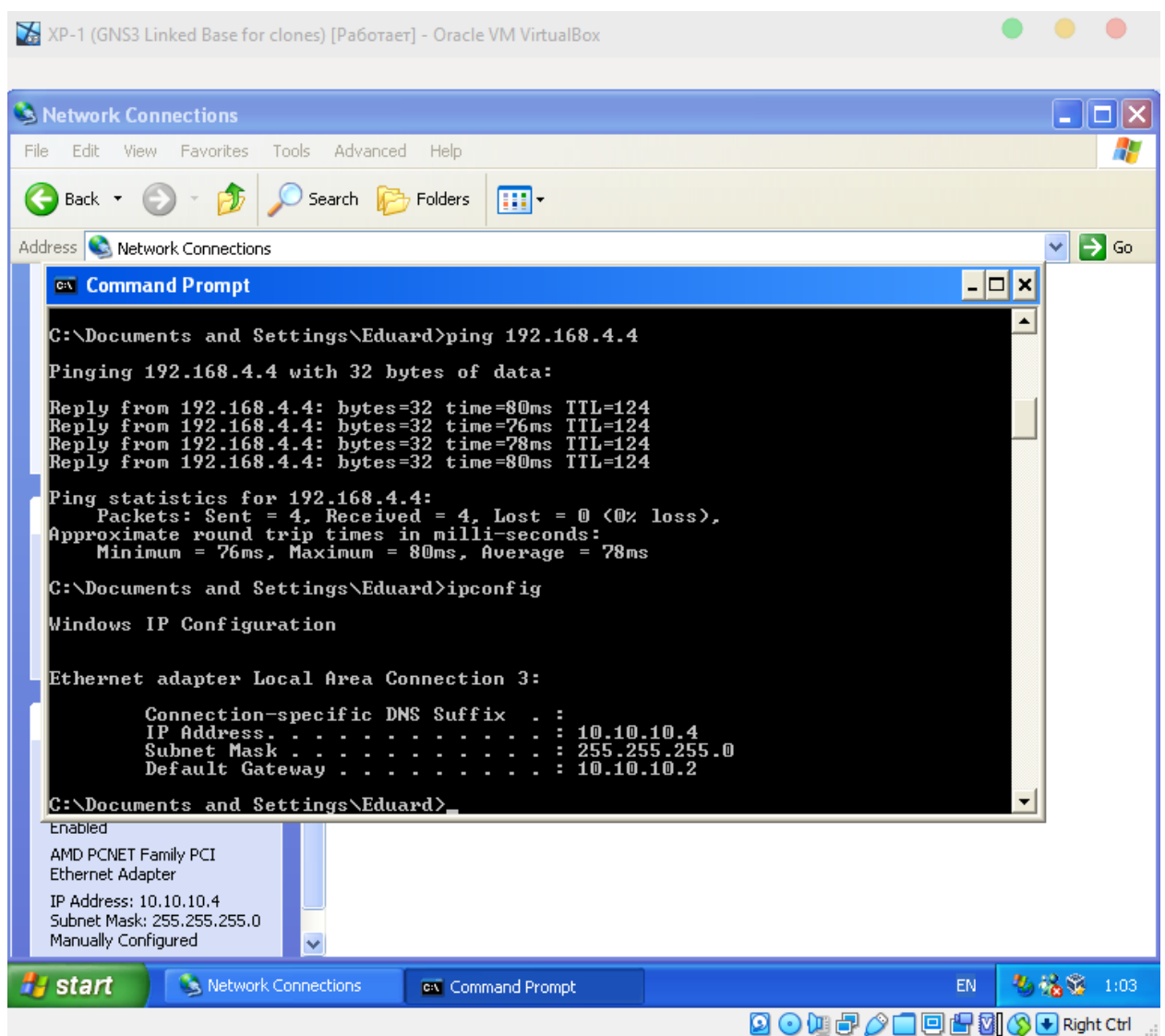


Рисунок 3.8 - Налаштування мережі для Multicast Server

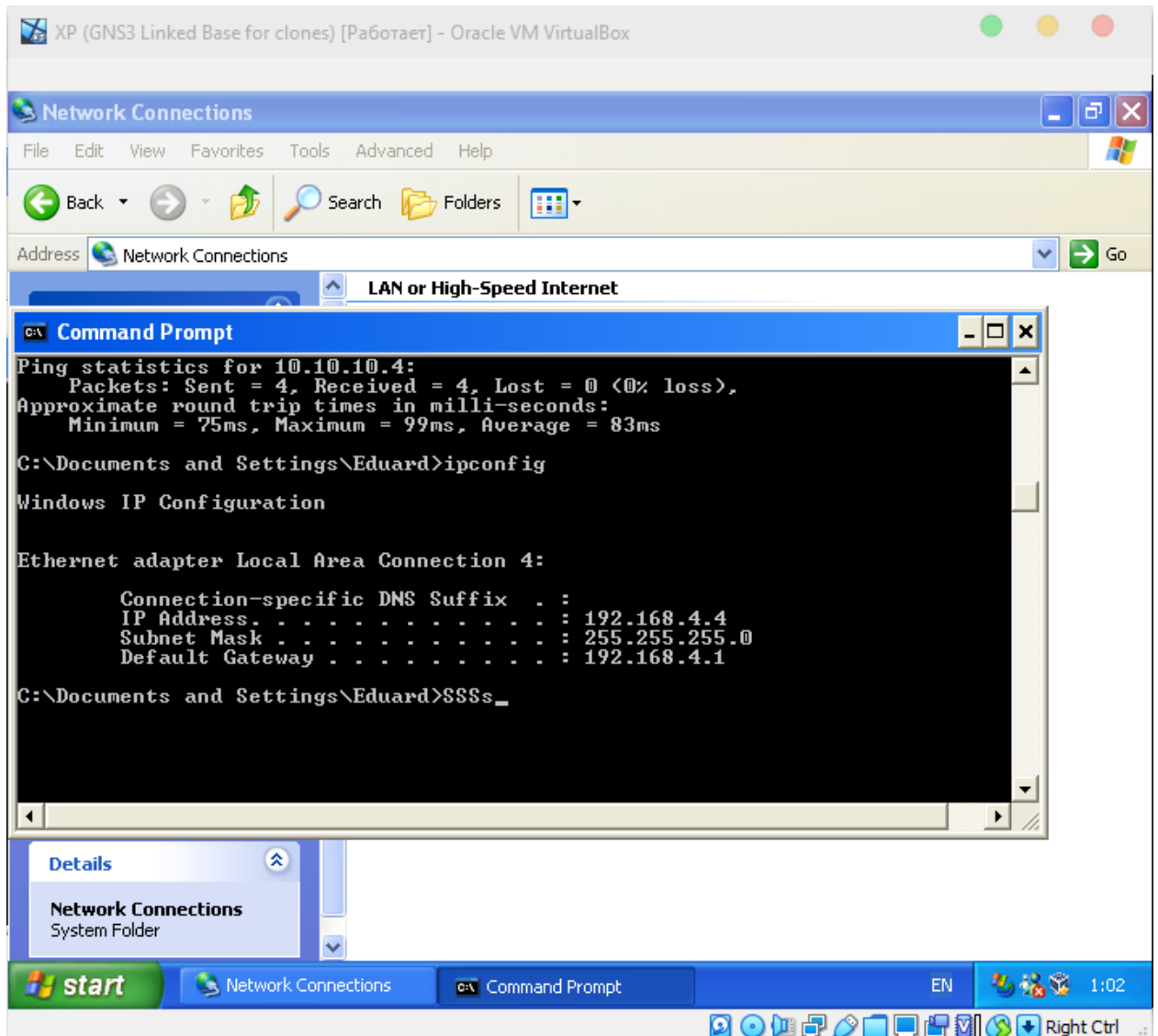


Рисунок 3.9 - Налаштування мережі для Multicast Client

Отже перша перевірка наших налаштувань командою “ping” пройшла успішно, тепер перейдемо до транслявання трафіку. Повторимо кроки з пункту 2.1 які відносяться до трансляції відео. Після успішного налаштування трансляції ми можемо наглядно побачити, що трафік передається без проблем (рис.3.10). Також можемо перевірити за допомогою команди “show ip mroute D” на першому роутері, як бачимо на рисунку 3.11 з адресу 10.10.10.4 транслюється потік 224.2.2.4.

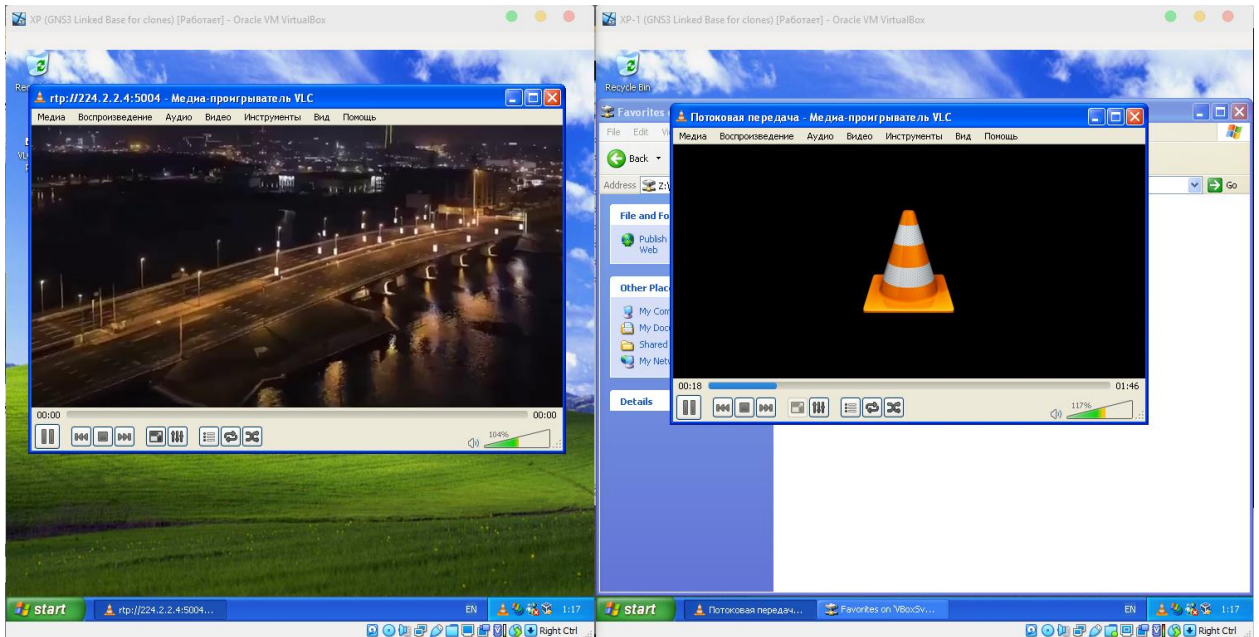


Рисунок 3.10 - Перевірка налаштувань схеми за допомогою трансляції відеофайлу

```

Router_1#show ip mroute D
(*, 239.255.255.250), 03:00:36/00:02:54, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet1/0, Forward/Dense, 02:57:24/00:00:00
    FastEthernet0/0, Forward/Dense, 03:00:36/00:00:00

(*, 224.2.2.4), 00:01:01/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet1/0, Forward/Dense, 00:01:01/00:00:00

(10.10.10.4, 224.2.2.4), 00:01:01/00:02:53, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet1/0, Forward/Dense, 00:00:55/00:00:00

(*, 224.0.1.40), 03:00:38/00:02:57, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet1/0, Forward/Dense, 02:57:24/00:00:00
    FastEthernet0/0, Forward/Dense, 03:00:38/00:00:00

Router_1#
  
```

Рисунок 3.11 - Перевірка налаштувань схеми за команди “show ip mroute D”

За допомогою роутерів Cisco 3640 та комутаторів було створено комп'ютерну мережу, яка працює на реальному обладнанні. Налаштування роутерів для Multicast передачі з використанням протоколів PIM та IGMP

здійснено за допомогою розробленої програми. Набори команд для роутерів можна легко скопіювати до буферу обміну та перенести їх до роутера.

На дане дослідження було витрачено 20 хвилин, що значно швидше ніж в перші спроби.

Протестувавши систему в ситуації наближеної до реальної, можна стверджувати що вона значно прискорює процес налаштування багатоадресної мережі.

ВИСНОВКИ

В ході роботи були розглянуті мережеві протоколи такі як PIM, IGMP, RIP, OSPF, технологію IPTV та як вона працює, базові принципи роботи з Multicast мережами та як їх налаштувати. Отримані знання та навички володіння в найпопулярніших мережевих симуляторів, таких як GNS3, Cisco Packet Tracer, UNetLab. На сьогодні мережеві симулятори становлять наглядний інструмент для отримання практичних навичок праці з основними типами мережевого обладнання такими як комутатори, маршрутизатори, хаби. Але до сих пір такі симулятори не мають “user-friendly” інтерфейсу для користувача, який би дозволяв на інтуїтивно-зрозумілому рівні освоювати програму та налаштувати різноманітні мережі, в тому числі і мережі Multicast.

На основі зібраної інформації було розроблено веб-орієнтована інформаційна система. Завдяки візуалізації інтерфейсу, користувач має змогу налаштувати конфігурацію мережевих інтерфейсів та групову передачу всередині мережі на реальному обладнанні CISCO, набори команд отриманих за допомогою цієї системи можна перенести на справжнє фізичне обладнання. Для роботи програми потрібно заповнити основні параметри роутерів (маску мережі та ip – адресу). За допомогою розробленого додатку у користувача наявна можливість зручно та без проблем перенести отриманий набір команд для роутерів безпосередньо до роутера.

Отримана програмна реалізація дає змогу недосвідченим користувачам налаштувати мережі Ethernet, без потреби знати базові команди для обладнання Cisco і прискорює процес налаштування та старт експлуатації мережі, відкидаючи необхідність ручного налаштування, яке займає більше часу та не виключає можливих помилок.

СПИСОК ЛІТЕРАТУРИ

1. Телекомунікаційні системи та мережі. Структура й основні функції. Том 1 [Електронний ресурс] - <http://www.znanius.com/3820.html>
2. IPTV [Електронний ресурс] - <https://www.explainthatstuff.com/how-iptv-works.html>
3. Протокол PIM-SM [Електронний ресурс] - <http://iptcp.net/protokol-pim-sm.html>
4. Протокол IGMP [Електронний ресурс] - <http://iptcp.net/protokol-igmp.html>
5. 2019 Annual Report Defining the Future of the Internet [Електронний ресурс] - https://www.cisco.com/c/dam/en_us/about/annual-report/cisco-annual-report-2019.pdf
6. JavaScript vs C++: Differences and Similarities [Електронний ресурс] - <https://careerkarma.com/blog/javascript-vs-cplusplus/>
7. RFC 7798, RTP Payload Format for High Efficiency Video Coding (HEVC)
8. M. Noormohammadpour; et al. (July 10, 2017). "DCCast: Efficient Point to Multipoint Transfers Across Datacenters". USENIX. Retrieved July 26, 2017.
9. IP Multicast Routing Configuration Guide, Cisco, p. 17-19, retrieved 2017-05-27
10. Internet Group Management Protocol, Version 2. doi:10.17487/RFC2236. RFC 2236.
11. Henry-Stocker, Sandra (2013-08-03). "Unix: Getting from here to there (routing basics)". Network World. Retrieved 2019-05-05.
12. "Globally Distributed Content Delivery, by J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Wehl, IEEE Internet Computing, Volume 6, Issue 5, November 2002" (PDF). Archived (PDF) from the original on 2017-08-09. Retrieved 2019-10-25.
13. Принципы работы протокола PIM [Електронний ресурс] - <https://habr.com/ru/post/450582/>

ДОДАТОК

Додаток А

```

<!DOCTYPE html>
<html>
<head>
  <title>Moroz Eduard</title>
  <meta charset="utf-8">

  <style type="text/css">
    * {margin: 0; padding: 0;} /* обнуляем отступы */
    body {
      text-align: center; /* выравниваем все содержимое body по центру */
      background: #fff; /* цвет фона для наглядности */
    }
    .center {
      width: 1110px; /* ширина основного блока */

      margin: 0 auto; /* задаем отступ слева и справа auto чтобы сработало
выравнивание по центру */
    }
    p {
      white-space: nowrap;
      font-family: monospace
    }
    .btn-clipboard3 {
      background-color: #4CAF50; /* Green */
      border: none;
      color: white;
      padding: 15px 32px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 16px;
    }

    .btn-clipboard, .btn-clipboard2 {
      background-color: #e7e7e7; /* red */
      border: none;
      color: black;
      padding: 15px 32px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 16px;
    }

    .tabl {
      background: #000 none repeat scroll 0 0;
      border-radius: 10px;
      color: green;
      float: left;
      font-family: monospace;
      font-size: 14px;
      margin-bottom: 20px;
      margin-top: 20px;
      padding: 18px;
      text-align: left;
      width: 440px;
    }
    .column {

```

```
        width: 480px;
        float: left;
margin: 20px;
        padding: 18px;
    }
.window1 {
font-family:monospace;
margin-left:211px;
margin-top: -57px;
position: absolute;
text-align: right;
    width: 220px;
height: 100px;
    }

.window2 {
font-family:monospace;
margin-left: 421px;
margin-top: -57px;
position: absolute;
    text-align: right;
    width: 220px;
height: 100px;
    }

.window3 {
font-family:monospace;
margin-left: 190px;
margin-top: -486px;
position: absolute;
    text-align: right;
    width: 220px;
height: 100px;
    }

    .window4 {
font-family:monospace;
margin-left: 400px;
margin-top: -486px;
position: absolute;
    text-align: right;
    width: 220px;
height: 100px;
    }

    .window5{
font-family:monospace;
margin-left: 611px;
margin-top: -486px;
position: absolute;
    text-align: right;
    width: 220px;
height: 100px;
    }

    .window6{
font-family:monospace;
margin-left: 821px;
margin-top: -486px;
position: absolute;
    text-align: right;
    width: 220px;
height: 100px;
```

```

    }
    .window7{
        font-family:monospace;
margin-left: 631px;
margin-top: -57px;
position: absolute;
        text-align: right;
        width: 220px;
height: 100px;
    }
    .window8{
        font-family:monospace;
margin-left: 841px;
margin-top: -57px;
position: absolute;
        text-align: right;
        width: 220px;
height: 100px;
    }
    .bg {

        margin-top: 20px;
        background: gray none repeat scroll 0 0;

position: absolute;

        }

    .column h1 {
        color: #fff;
font-family: monospace;
margin-left: 25px;
        }
</style>

<script src="http://code.jquery.com/jquery-1.11.0.min.js"
type="text/javascript"></script>

<script
src="https://cdn.rawgit.com/zenorocha/clipboard.js/master/dist/clipboard.min.
js"></script>

</head>
<body>
<script type="text/javascript">
function CopyToClipboard1(containerid) {
    if (document.selection) {

        var range = document.body.createTextRange();
range.moveToElementText (document.getElementById(containerid));
range.select().createTextRange();
document.execCommand("copy");

    } else if (window.getSelection) {

        var range = document.createRange();
range.selectNode (document.getElementById(containerid));
window.getSelection().addRange(range);
document.execCommand("copy");
alert("Text has been copied, now paste in the text-area")

    }
}

```



```

}
</script>

<div class="center">

<div style="margin-top: 50px">

</div>

<div class="window1">
IP:      <input      type="text"      id="myText"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText2"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

<div class="window2">
IP:      <input      type="text"      id="myText3"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText4"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/><br /><br />
</div>

<div class="window3">
IP:      <input      type="text"      id="myText5"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText6"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/><br /><br />
</div>

<div class="window4">
IP:      <input      type="text"      id="myText7"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText8"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

<div class="window5">
IP:      <input      type="text"      id="myText9"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText10"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

<div class="window6">
IP:      <input      type="text"      id="myText11"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText12"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

<div class="window7">
IP:      <input      type="text"      id="myText13"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText14"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

```

```

<div class="window8">
IP:      <input      type="text"      id="myText15"      title="Write      ip"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"><br
/>
MASK:    <input      type="text"      id="myText16"      title="Write      mask"
placeholder='xxx.xxx.xxx.xxx' pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}">
</div>

```

```

<br /><br /><br /><br />

```

```

<button  onclick="myFunction()"  class="btn-clipboard3"  style="margin:  0
auto;font-size:24px;font-family:monospace">Save Settings</button>
<br />

```

```

<div class="bg">
<h1>IPTV</h1>
<div class="column">

```

```

<div id="textarea-example" class="tabl">enable<br />
conf term<br />
hostname Router_1 <br />
ip multicast-routing <br />
interface FastEthernet0/0<br />
ip address <w id="demo1"></w>&nbsp;<w id="demo2"></w><br />
ip pim dense-mode <br />
  no sh <br />
  exit <br />
interface FastEthernet1/0<br />
ip address <w id="demo3"></w>&nbsp;<w id="demo4"></w><br />
ip pim dense-mode <br />
no sh <br />
exit<br />
ip route <w id="demo17"></w>&nbsp;<w id="demo6"></w>&nbsp;<w id="demo5"></w><br
/>
wr</div>
<button class="btn-clipboard2"  onclick="CopyToClipboard1('textarea-example')"
data-clipboard-target="#textarea-example"  style="margin:  0  auto;font-
size:18px;font-family:monospace">Copy settings</button>

```

```

</div>

```

```

<div class="column">
<div id="textarea-example2" class="tabl">enable<br />
conf term<br />
hostname Router_2 <br />
ip multicast-routing <br />
interface FastEthernet0/0<br />
ip address <w id="demo50"></w>&nbsp;<w id="demo60"></w><br />
ip pim dense-mode <br />
  no sh <br />
  exit <br />
interface FastEthernet1/0<br />
ip address <w id="demo7"></w>&nbsp;<w id="demo8"></w><br />
ip pim dense-mode <br />
no sh <br />
exit<br />
ip  route      <w      id="demo220"></w>&nbsp;<w      id="demo9"></w>&nbsp;<w
id="demo9"></w><br />
ip  route      <w      id="demo2100"></w>&nbsp;<w      id="demo40"></w>&nbsp;<w
id="demo30"></w><br />
wr</div>

```

```
<button class="btn-clipboard2" onclick="CopyToClipboard1('textarea-example2')"  
data-clipboard-target="#textarea-example2" style="margin: 0 auto;font-  
size:18px;font-family:monospace">Copy settings</button>  
</div>
```

```
<div class="column">  
<div id="textarea-example3" class="tab1">enable<br />  
conf term<br />  
hostname Router_3 <br />  
ip multicast-routing <br />  
interface FastEthernet0/0<br />  
ip address <w id="demo90"></w>&nbsp;<w id="demo100"></w><br />  
ip pim dense-mode <br />  
no sh <br />  
exit <br />  
interface FastEthernet1/0<br />  
ip address <w id="demo11"></w>&nbsp;<w id="demo12"></w><br />  
ip pim dense-mode <br />  
no sh <br />  
exit<br />  
ip route <w id="demo22"></w>&nbsp;<w id="demo80"></w>&nbsp;<w  
id="demo70"></w><br />  
ip route <w id="demo21"></w>&nbsp;<w id="demo14"></w>&nbsp;<w  
id="demo13"></w><br />  
wr</div>  
<button class="btn-clipboard2" onclick="CopyToClipboard1('textarea-  
example3')" data-clipboard-target="#textarea-example3" style="margin: 0  
auto;font-size:18px;font-family:monospace">Copy settings</button>  
</div>
```

```
<div class="column">  
<div id="textarea-example4" class="tab1">enable<br />  
conf term<br />  
hostname Router_4 <br />  
ip multicast-routing <br />  
interface FastEthernet0/0<br />  
ip address <w id="demo130"></w>&nbsp;<w id="demo140"></w><br />  
ip pim dense-mode <br />  
no sh <br />  
exit <br />  
interface FastEthernet1/0<br />  
ip address <w id="demo15"></w>&nbsp;<w id="demo16"></w><br />  
ip pim dense-mode <br />  
no sh <br />  
exit<br />  
ip route <w id="demo210"></w>&nbsp;<w id="demo120"></w>&nbsp;<w  
id="demo110"></w><br />  
wr</div>  
<button class="btn-clipboard2" onclick="CopyToClipboard1('textarea-example4')"  
data-clipboard-target="#textarea-example4" style="margin: 0 auto;font-  
size:18px;font-family:monospace">Copy settings</button>  
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript">
```

```
function myFunction() {
```

```
    var a = document.getElementById("myText").value;
```

```
    var a1=a;
```

```
    var b = document.getElementById("myText2").value;
```

```
var b1=b;
```

```

    var c = document.getElementById("myText3").value;
var c1=c;
    var d = document.getElementById("myText4").value;
var d1=d;
    var e = document.getElementById("myText5").value;
    var e1= e;
    var f = document.getElementById("myText6").value;
var f1=f;
    var g = document.getElementById("myText7").value;
var g1=g;
    var h = document.getElementById("myText8").value;
    var h1=h;
    var j = document.getElementById("myText9").value;
    var j1= j;
    var k = document.getElementById("myText10").value;
    var k1=k;
    var l = document.getElementById("myText11").value;
    var l1=l;
    var m = document.getElementById("myText12").value;
    var m1=m;
    var n = document.getElementById("myText13").value;
    var n1=n;
    var z = document.getElementById("myText14").value;
    var z1=z;
    var x = document.getElementById("myText15").value;
    var x1=x;
    var o = document.getElementById("myText16").value;
    var o1=o;
    var ipr1 = (x.trim()).replace(/\d{1,3}$/,"0");
    var ipr13 = (x1.trim()).replace(/\d{1,3}$/,"0");
    var ipr12 = ipr1;
    var ipr2 = (c.trim()).replace(/\d{1,3}$/,"0");
    var ipr3 = (j.trim()).replace(/\d{1,3}$/,"0");
    var ipr4 = (a.trim()).replace(/\d{1,3}$/,"0");
    var ipr42 = ipr4;
    var ipr43 = (a1.trim()).replace(/\d{1,3}$/,"0");
    document.getElementById("demo1").innerHTML = a;
    document.getElementById("demo2").innerHTML = b;
    document.getElementById("demo3").innerHTML = c;
    document.getElementById("demo4").innerHTML = d;
    document.getElementById("demo5").innerHTML = e;
    document.getElementById("demo6").innerHTML = f;
    document.getElementById("demo7").innerHTML = g;
    document.getElementById("demo8").innerHTML = h;
    document.getElementById("demo9").innerHTML = j;

    document.getElementById("demo11").innerHTML = l;
    document.getElementById("demo12").innerHTML = m;
    document.getElementById("demo13").innerHTML = n;
    document.getElementById("demo14").innerHTML = z;
    document.getElementById("demo15").innerHTML = x;
    document.getElementById("demo16").innerHTML = o;
    document.getElementById("demo17").innerHTML = ipr1;
    document.getElementById("demo21").innerHTML = ipr12;
    document.getElementById("demo22").innerHTML = ipr4;
    document.getElementById("demo210").innerHTML = ipr42 ;
    document.getElementById("demo2100").innerHTML = ipr43 ;
    document.getElementById("demo220").innerHTML = ipr13 ;
    document.getElementById("demo130").innerHTML = n1;
    document.getElementById("demo140").innerHTML = z1;
    document.getElementById("demo50").innerHTML = e1;
    document.getElementById("demo60").innerHTML = f1;
    document.getElementById("demo30").innerHTML =c1 ;

```

```
document.getElementById("demo40").innerHTML = d1;
document.getElementById("demo90").innerHTML = j1;
document.getElementById("demo100").innerHTML = k1;
document.getElementById("demo70").innerHTML = g1;
document.getElementById("demo80").innerHTML = h1;
document.getElementById("demo110").innerHTML = l1;
document.getElementById("demo120").innerHTML = m1;

}
</script>
</script>
</body>
</html>
```