UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Architettura di rete dinamica: Ottimizzazione della configurazione fisica per affidabilità e sicurezza

Dynamic Network Architecture: Optimizing the Physical Configuration for Reliability and Security

Relatore

Prof. Ennio Gambi

Correlatore

Ing. Adelmo De Santis

Candidato

Nicolò Ianni

ANNO ACCADEMICO 2022-2023

INDICE

INTRODUZIONE

Nell'ambito delle tecnologie digitali odierne, le reti di comunicazione rivestono un ruolo imprescindibile nell'interconnettere dispositivi vari e nel facilitare un trasferimento dei dati efficace ed efficiente. Queste reti, che si estendono da configurazioni domestiche semplici fino ad arrivare a complesse infrastrutture di rete aziendali o di provider di servizi internet, costituiscono l'infrastruttura vitale che mette in comunicazione persone, imprese e organizzazioni in tutto il mondo. La progettazione di una rete, soprattutto quando si tratta di architetture complesse, richiede una pianificazione meticolosa e ben calibrata, che tenga conto di diversi aspetti fondamentali come la scalabilità, la resilienza, la sicurezza e le necessità specifiche degli utenti o delle entità aziendali coinvolte.

Professionisti del settore delle reti devono dunque armarsi di competenze tecniche avanzate e di una profonda comprensione dei vari protocolli di rete e delle architetture, oltre a dover interpretare accuratamente le esigenze degli utenti finali. Una volta che la rete è operativa, il monitoraggio costante e dinamico diventa un aspetto critico per assicurarne il funzionamento ottimale. Questo include la sorveglianza delle prestazioni della rete, la gestione della sicurezza e la continua ottimizzazione delle risorse, che insieme contribuiscono a mantenere l'infrastruttura reattiva ed efficiente di fronte a eventuali criticità o malfunzionamenti.

L'impiego di strumenti specializzati e di tecnologie avanzate per il monitoraggio permette di raccogliere dati in tempo reale sull'operatività della rete, analizzarli e, se necessario, intervenire con azioni correttive. In questo contesto di continua evoluzione tecnologica, con un incremento parallelo delle minacce alla sicurezza informatica, la progettazione e il monitoraggio delle reti si collocano come elementi chiave per fornire servizi di alta qualità e per sostenere le crescenti richieste di connettività della società contemporanea, rendendo il settore un terreno fertile per l'innovazione e lo sviluppo costante.

CAPITOLO 1

Analisi del progetto

All'interno del progetto in esame, la configurazione della rete è stata realizzata inizialmente tramite il collegamento fisico mediante cavi di rete di cinque router e due switch, tutti forniti dalla Huawei. È essenziale sottolineare che il semplice collegamento di questi dispositivi rappresenta soltanto l'avvio del processo di creazione di una rete operativa. Il vero fulcro del progetto si trova nella configurazione dettagliata di questi dispositivi, che andrà a formare la topologia fisica prevista. Questo capitolo approfondirà la progettazione di questa topologia complessa, mettendo in luce le sfide tecniche incontrate e le decisioni critiche adottate durante l'implementazione. Si esplorerà anche il software di gestione dinamica adoperato per il monitoraggio e l'ottimizzazione della rete.

1.1 Obiettivo della tesi

L'obiettivo principale di questa tesi è esplorare il processo di creazione di una topologia di rete principalmente dal punto di vista fisico, concentrandosi sulla progettazione, configurazione e monitoraggio per realizzare un'infrastruttura che possa rispondere efficacemente alle variazioni di traffico. Si punterà a mettere in luce le sfide tecniche e le scelte strategiche intraprese, valutando come la configurazione fisica contribuisca a raggiungere gli obiettivi di prestazione, affidabilità e sicurezza nella comunicazione.

1.1.1 Software di gestione dinamica

Per ampliare l'efficienza della rete, è stato progettato un avanzato software di gestione in Python, specificamente orientato a fornire un monitoraggio dettagliato delle interfacce dei router Huawei. Questo strumento, attraverso l’interazione con i dispositivi, consente una regolazione precisa e tempestiva del traffico di rete, assicurando così un flusso dati ottimale. La presente tesi dedicherà un'attenzione particolare alla componente fisica della rete, approfondendo la configurazione specifica dei dispositivi e l'architettura delle loro connessioni. Verranno esaminati con cura i meccanismi tecnologici che facilitano la comunicazione tra il software gestionale e l'infrastruttura hardware, mettendo in evidenza come l'integrazione tra

questi elementi sia fondamentale per garantire la resilienza e la dinamicità della rete.

1.2 Progetto

Questa sezione introduce il progetto di implementazione della topologia di rete e del software di gestione dinamica, discutendo gli obiettivi, le fasi di sviluppo e le tecnologie impiegate.

1.2.1 Fasi di sviluppo

Il progetto è stato suddiviso in diverse fasi, tra cui:

1. Progettazione della topologia di rete fisica, inclusa la scelta dei dispositivi Huawei.
2. Configurazione iniziale dei dispositivi e interconnessione fisica.
3. Implementazione delle regole di routing e delle politiche di sicurezza.
4. Sviluppo del software di gestione dinamica.
5. Test e validazione delle funzionalità di monitoraggio e ottimizzazione.
6. Analisi dei dati raccolti e valutazione delle prestazioni.

Queste fasi di sviluppo rappresentano effettivamente anche quali sono gli obiettivi del progetto.

1.2.2 Configurazione fisica

Nella fase iniziale del progetto, è stata implementata la configurazione fisica della rete utilizzando il simulatore di dispositivi Huawei eNSP. Nella figura 3.5 è mostrata la topologia fisica dei dispositivi di rete:

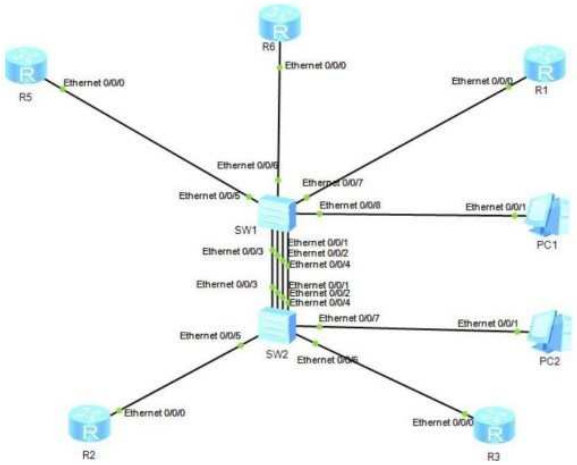


Figura 1.1: Configurazione fisica della rete

Come illustrato nella figura 1.1, i router R1, R5 e R6 sono connessi allo switch SW1, mentre i router R2 e R3 sono connessi allo switch SW2. Questa configurazione fisica costituisce la base sulla quale è stata implementata la topologia logica desiderata.

1.2.3 Configurazione logica

La topologia logica in figura1.2 è stata ottenuta attraverso la configurazione di apparati di rete, utilizzando una serie di tecniche, proprie di un approccio enterprise.

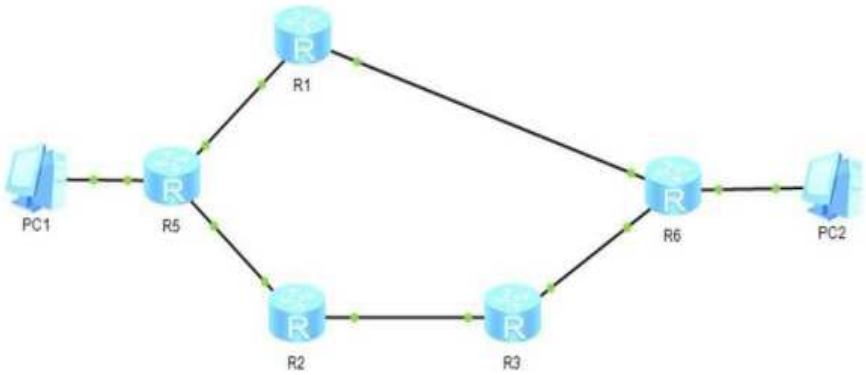


Figura 1.2: Configurazione logica desiderata della rete

1.2.4 Implementazione Software

In questa sezione, esploreremo l'implementazione del software che è stato sviluppato per ottimizzare il traffico dei router R5 e R6 e monitorare gli altri apparati quali R1, R2 ed R3. Il software è dotato di un'interfaccia grafica intuitiva che consente agli utenti di selezionare quale router desiderano monitorare e gestire. Di seguito, forniamo un'introduzione alle principali caratteristiche e funzionalità del software. La connessione tra il software in Python e i dispositivi di rete avviene tramite il protocollo SSH (Secure Shell). Tutti i router sono configurati come server SSH, consentendo al software di stabilire una connessione sicura con essi. Il computer dell'utente funge da client SSH e comunica con i router in modalità wireless, eliminando la necessità di cavi di rete fisici.

Il software è dotato di un'interfaccia grafica utente (GUI) che semplifica l'interazione con gli utenti. Utilizzando questa GUI, gli utenti possono scegliere se gestire il

traffico del router R5 o del router R6. Di seguito, mostriamo un'immagine dell'interfaccia grafica del software (Figura 1.3):

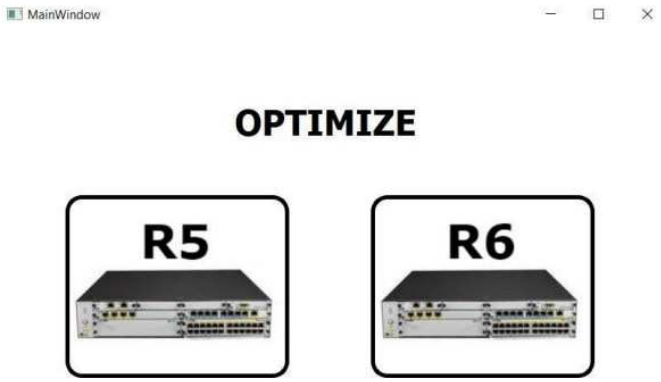


Figura 1.3: Interfaccia grafica del software

Gestione del Traffico

Una delle funzionalità chiave del software è la capacità di monitorare e gestire il traffico sulle interfacce dei router. Il software effettua un controllo periodico del traffico sulle interfacce 0/0/0 e 0/0/1 dei router R5 ed R6, ciò avviene ad intervalli predefiniti.

Se la percentuale di utilizzo di una delle due porte supera una certa soglia, il software avvierà automaticamente lo script che permette di instradare nuovo traffico secondo un percorso logico differente attraverso rotte statiche, ottimizzando così la distribuzione del traffico di rete.

Monitoraggio dello Stato delle Interfacce

Il software fornisce anche la possibilità di monitorare lo stato delle interfacce di tutti i router utilizzati nella nostra rete. Gli utenti possono verificare in tempo reale lo stato delle interfacce, rilevando eventuali anomalie o problemi di connettività. In conclusione, il software rappresenta una soluzione potente per la gestione e l'ottimizzazione del traffico di rete nei router R5 e R6 della nostra configurazione. La connessione wireless e la GUI user-friendly rendono l'uso del software semplice e accessibile agli utenti.

CAPITOLO 2

Tecnologie Utilizzate

In questo capitolo, saranno illustrate le tecnologie utilizzate per soddisfare gli obiettivi del progetto. La sezione sarà suddivisa in due parti principali: la prima parte esaminerà le risorse tecnologiche utilizzate per lo sviluppo della topologia di rete e il loro funzionamento, mentre la seconda parte si concentrerà sulle tecnologie impiegate per lo sviluppo del software di gestione.

2.1 Topologia di Rete

2.1.1 eNSP (Emulated Network Simulation Platform)

Una delle tecnologie chiave utilizzate per la progettazione e l'implementazione della topologia di rete è stata eNSP, acronimo di "Emulated Network Simulation Platform". eNSP è un simulatore di dispositivi Huawei che ha consentito di creare e testare la topologia di rete in un ambiente virtuale prima di implementarla fisicamente. Con eNSP, sono stato in grado di:

- Creare una rappresentazione virtuale dei dispositivi Huawei, inclusi router e switch, che sarebbero stati utilizzati nella rete.
- Simulare l'interconnessione dei dispositivi tramite cavi di rete virtuali, consentendo di pianificare e configurare la topologia fisica. eNSP è stato uno strumento prezioso nel processo di progettazione, consentendo di esplorare diverse configurazioni e valutare il funzionamento della topologia prima della sua implementazione fisica. La sua flessibilità e la possibilità di simulare scenari realistici hanno permesso di affinare la progettazione e garantire che la topologia soddisfacesse gli obiettivi di prestazioni e affidabilità.

2.1.2 OSPF (Open Shortest Path First)

OSPF è il protocollo di routing utilizzato nella topologia di rete. Questo protocollo è stato scelto per la sua capacità di gestire dinamicamente il contenuto delle tabelle di routing, ottimizzando il percorso dei pacchetti dati attraverso la sua metrica. Con

OSPF, i router sono stati configurati per condividere informazioni sulle rotte di rete e calcolare il percorso più efficiente per la trasmissione dei dati.

2.1.3 VLAN (Virtual LAN)

Le VLAN, acronimo di "Virtual LAN", sono state implementate per isolare il traffico di rete in segmenti logici separati. Questa tecnologia ha permesso di suddividere la rete fisica in gruppi logici, consentendo di realizzare una topologia logica diversa da quella fisica.

2.1.4 Sub-Interface

Le sub-interface, o sottointerfacce, sono un elemento fondamentale nell'ambito della configurazione di reti complesse, in particolare nelle reti che devono supportare la suddivisione del traffico in diverse VLAN (Virtual LAN). Queste consentono di dividere le interfacce dei router in sottointerfacce virtuali, ciascuna delle quali può avere la propria configurazione di rete, anche in termini di indirizzo IP e netmask. L'uso delle sub-interface è cruciale per trasformare una configurazione di rete fisica in quella logica desiderata e per gestire in modo efficace il traffico VLAN. Nel contesto della tesi, esploreremo come l'implementazione di sub-interface abbia consentito di gestire in modo efficiente il traffico di più VLAN, assegnando a ciascuna sub-interface un ruolo specifico per garantire la connettività e la separazione virtuale tra le VLAN.

2.1.5 DHCP (Dynamic Host Configuration Protocol)

Il protocollo DHCP è stato implementato per semplificare la gestione degli indirizzi IP all'interno della rete. Il protocollo consente, ad un dispositivo identificato come "dhcp server" di assegnare automaticamente un indirizzo IP ai client che ne fanno richiesta. In questo modo la risorsa degli indirizzi IP può essere usata con maggiore efficienza, evitando conflitti.

2.2 Software di ottimizzazione del traffico

2.2.1 Python

Python è un linguaggio di programmazione molto popolare ed estremamente versatile che è diventato una scelta comune per lo sviluppo di software in molte

aree, inclusa la gestione del traffico di rete. La sua crescente adozione nelle applicazioni di networking è dovuta a diversi fattori chiave:

1. ***Semplicità e chiarezza del codice***: Python è noto per la sua sintassi chiara e leggibile, che rende il codice più comprensibile rispetto ad altri linguaggi. Questa caratteristica è particolarmente vantaggiosa quando si sviluppano soluzioni per la gestione delle reti, dove la chiarezza del codice è fondamentale per evitare errori costosi.

2. ***Ampia comunità e supporto***: Python ha una vasta comunità di sviluppatori e un ecosistema di librerie in continua crescita. Questo offre accesso a una ricca raccolta di librerie e framework specifici per le reti, semplificando lo sviluppo di applicazioni di gestione del traffico.

3. ***Portabilità:*** Python è multipiattaforma, il che significa che le applicazioni sviluppate in Python possono essere eseguite su diverse piattaforme senza la necessità di modifiche significative. Questo è cruciale nella gestione di reti eterogenee con dispositivi e sistemi operativi diversi.

4. ***Rapidità nello sviluppo*****:** Python è noto per la sua capacità di sviluppo rapido. Gli sviluppatori possono scrivere, testare e iterare rapidamente sul codice, consentendo di implementare nuove funzionalità o risolvere problemi più velocemente.

5. ***Ampia adozione nell'automazione di rete***: Python è ampiamente utilizzato per l'automazione di rete grazie a librerie come Paramiko per la connessione SSH, Netmiko per l'automazione di dispositivi di rete e Scapy per la manipolazione dei pacchetti di rete. Questi strumenti semplificano la gestione dinamica del traffico e la configurazione dei dispositivi di rete.

6. ***Librerie specifiche per il networking***: Python offre numerose librerie specifiche per il networking, come Scapy, NetworkX e Twisted, che consentono di manipolare e analizzare il traffico di rete in modo efficace.

Nella configurazione di rete descritta, l'uso di Python come linguaggio di programmazione principale per lo sviluppo del software di ottimizzazione del

traffico ha permesso di sfruttare appieno queste caratteristiche. La versatilità di Python e l'accesso alle librerie di rete specializzate hanno reso possibile la realizzazione delle funzionalità necessarie per il monitoraggio e l'ottimizzazione dinamica del traffico.

2.2.2 Netmiko

Netmiko è una libreria Python specializzata nella gestione di dispositivi di rete tramite connessioni SSH. Nel contesto del progetto, è stata sfruttata appieno questa potente libreria per stabilire connessioni sicure con i router R5 e R6. Questa scelta ha permesso di interagire direttamente con i dispositivi di rete, fornendo al software un mezzo affidabile e sicuro per monitorare e ottimizzare il traffico di rete. Netmiko semplifica notevolmente la comunicazione con i dispositivi di rete, offrendo una serie di funzionalità essenziali. Queste includono l'autenticazione, la gestione delle sessioni SSH, la configurazione di dispositivi e la raccolta di dati di rete in modo efficiente. Inoltre, Netmiko supporta una vasta gamma di dispositivi di rete, tra cui i router Huawei che fanno parte dell'infrastruttura del progetto. Utilizzando Netmiko, si è in grado di automatizzare il processo di monitoraggio e ottimizzazione del traffico di rete. Il software può inviare comandi e richieste direttamente ai router, acquisendo dati in tempo reale sulle prestazioni della rete. Questi dati sono stati fondamentali per valutare l'efficacia delle politiche di routing e sicurezza, nonché per identificare potenziali aree di miglioramento. In sintesi, Netmiko è stata una componente chiave del progetto, consentendoci di gestire in modo efficiente e sicuro i dispositivi di rete e di raggiungere gli obiettivi di monitoraggio e ottimizzazione del traffico di rete.

2.2.3 Qt Designer

Qt Designer è un'applicazione grafica per la creazione e la progettazione di interfacce utente (UI) per software. È un componente del framework Qt, un popolare framework di sviluppo cross-platform utilizzato per la creazione di applicazioni desktop, mobile e embedded. Qt Designer offre un ambiente visuale che consente agli sviluppatori di progettare l'aspetto e il layout delle interfacce utente delle loro applicazioni in modo intuitivo e senza la necessità di scrivere manualmente il codice sorgente per la UI. Questo strumento semplifica notevolmente il processo di creazione di UI, consentendo agli sviluppatori di trascinare e rilasciare elementi grafici (come pulsanti, caselle di testo, menu a discesa, ecc.) sulla finestra di

progettazione e configurarli con facilità. Sono state sfruttate le potenzialità di Qt Designer per progettare e creare l'interfaccia grafica del software. L'utilizzo di un'interfaccia utente ben progettata è cruciale per garantire che il software sia accessibile e facile da utilizzare per gli utenti finali. Qt Designer ha permesso di creare un'interfaccia utente intuitiva e user-friendly, consentendo agli utenti di interagire con il software in modo efficiente e senza la necessità di conoscenze tecniche approfondite. L'interfaccia grafica, creata con l'ausilio di Qt Designer, è stata progettata per consentire agli utenti di interagire con il software in modo intuitivo. Gli utenti possono facilmente avviare operazioni di monitoraggio e ottimizzazione del traffico di rete in modo semplice ed efficiente.

2.2.4 Iperf

Iperf è uno strumento di test delle prestazioni di rete è stato utilizzato per lo sviluppo del software. Consente di generare traffico di rete per valutare l'efficacia delle ottimizzazioni del traffico. Iperf opera come un'applicazione client/server, consentendo di generare traffico di rete tra due dispositivi. È una potente utility da riga di comando che offre diversi vantaggi:

- ***Flessibilità del protocollo***: Iperf consente agli utenti di scegliere il protocollo di funzionamento, inclusi TCP e UDP, in base alle esigenze del test delle prestazioni. Questa flessibilità è fondamentale per simulare scenari specifici di traffico di rete.
- ***Misurazioni accurate***: Iperf fornisce misurazioni accurate delle prestazioni di rete, inclusa la larghezza di banda, la latenza e la perdita di pacchetti. Queste misurazioni sono essenziali per valutare l'efficacia delle ottimizzazioni del traffico.
- ***Facilità di utilizzo***: Nonostante sia un'applicazione da riga di comando, Iperf è relativamente semplice da utilizzare e offre una serie di opzioni per configurare il test delle prestazioni in base alle esigenze specifiche.

Nel contesto del progetto, Iperf è stato impiegato per generare traffico all'interno della rete, consentendo di eseguire test delle prestazioni e valutare il comportamento della rete in situazioni diverse. Questi test sono stati fondamentali per valutare l'efficacia delle ottimizzazioni del traffico e per identificare eventuali problemi di prestazioni. Inoltre, l'opzione di scelta del protocollo ha consentito di simulare scenari realistici di traffico di rete, contribuendo a migliorare la comprensione del comportamento della rete in condizioni variabili.

CAPITOLO 3

Sviluppo del Progetto

In questa sezione, affronteremo i dettagli del processo di sviluppo del progetto, concentrandoci sulle diverse fasi e attività coinvolte nell'implementazione della topologia di rete.

3.1 Sviluppo Topologia

Lo sviluppo della rete di comunicazione ha richiesto un approccio alla progettazione complesso, iniziato con l'analisi della topologia fisica.

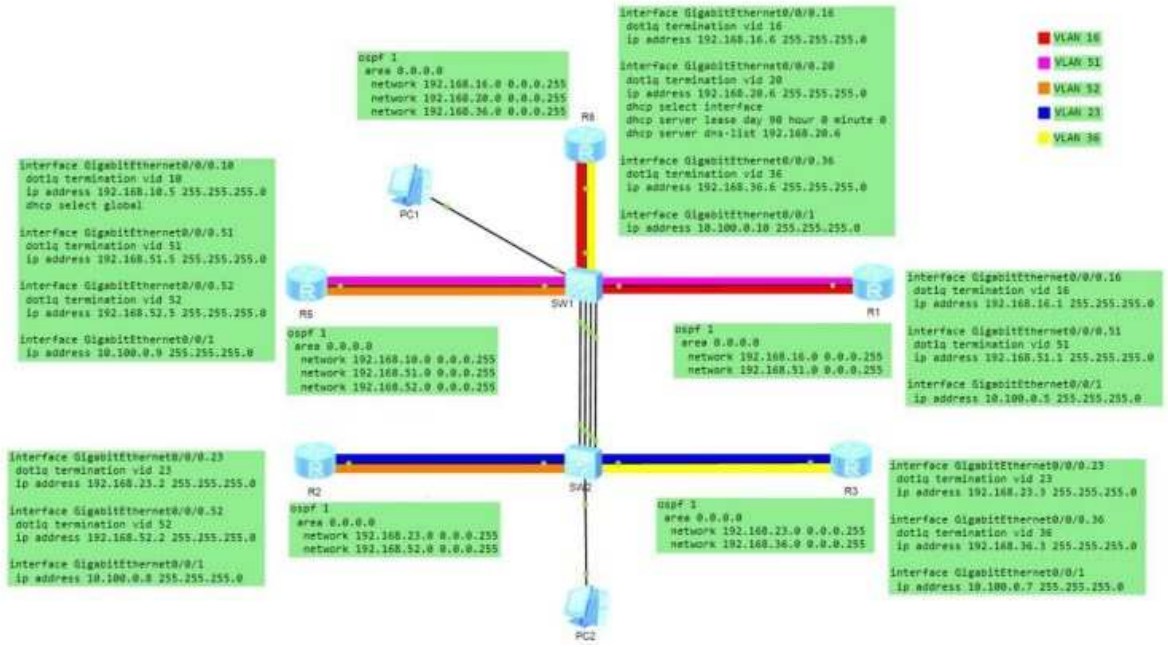


Figura 3.1: Configurazione fisica della rete ottenuta con eNSP.

All'inizio del progetto, è stata sviluppata un'idea generale su come far funzionare la rete e su quali strumenti e tecnologie potessero essere utilizzati per raggiungere gli obiettivi prefissati. Questa fase iniziale ha comportato una valutazione delle esigenze di rete e la progettazione di una topologia che potesse soddisfarle.

Successivamente, ho trasformato il concetto iniziale in una solida implementazione pratica. Questo processo ha coinvolto l'effettiva configurazione dei dispositivi di rete. In sostanza, ho tradotto la teoria in pratica, passando dalla fase di progettazione concettuale alla configurazione reale dei dispositivi e

all’implementazione delle tecnologie. Questo approccio ha richiesto una comprensione dettagliata delle tecnologie di rete e delle loro applicazioni, nonché un’attenta pianificazione e configurazione per garantire che la topologia di rete funzionasse come desiderato.

3.2 Hardware utilizzato

3.2.1 Cavi di rete

Il cavo Ethernet, noto anche come cavo straight-through, è uno dei tipi di cavi più comunemente utilizzati nelle reti cablate per collegare diversi dispositivi di rete, come computer, switch, router e hub. Questo tipo di cavo è così chiamato perché i conduttori all'interno del cavo corrono dritti attraverso i connettori RJ-45 ad entrambe le estremità, senza alcun incrocio.

Struttura e Specifiche:

- ***Connettori***: Ai due estremi del cavo Ethernet sono presenti connettori RJ-45, che sono standardizzati e ampiamente utilizzati per connessioni di rete.

Figura 3.2.1 Connettore RJ-45

- ***Coppie intrecciate***: Il cavo è composto da quattro coppie di cavi intrecciati. L'intreccio riduce la diafonia e le interferenze elettromagnetiche provenienti da fonti esterne e da altri fili all'interno dello stesso cavo.

- ***Categoria***: I cavi Ethernet sono disponibili in varie categorie (Cat 5, Cat 5e, Cat 6, Cat 6a, Cat 7), che indicano le prestazioni del cavo, in particolare in termini di larghezza di banda e riduzione delle interferenze. Cat 5e e Cat 6 sono tra le più comuni nelle reti domestiche e aziendali, offrendo un buon equilibrio tra costo e prestazioni.

- ***Colore dei conduttori***: Sebbene la colorazione dei fili possa variare, esiste uno standard di disposizione dei conduttori, che deve essere seguito. Questo aiuta a garantire la coerenza e la corretta terminazione dei cavi durante l'installazione.

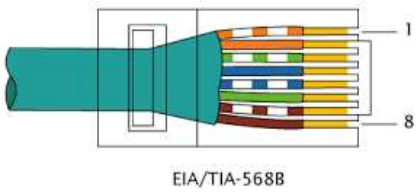


Figura 3.2.2 Standard EIA/TIA-5688

Vantaggi:

- ***Versatilità***: Può essere utilizzato per la maggior parte delle connessioni di rete nelle configurazioni domestiche e aziendali.

- ***Affidabilità***: Fornisce connessioni stabili e affidabili con velocità di trasmissione dati elevate.

- ***Facilità di utilizzo***: Facile da installare e collegare, senza la necessità di configurazioni complesse.

3.2.2 Dispositivi

In questa sezione, esploreremo la struttura e le caratteristiche tecniche dei dispositivi di rete impiegati per costruire la topologia di rete. Utilizzando router della serie Huawei AR G3 e switch della serie Huawei S5700, analizzeremo come questi dispositivi contribuiscano all'affidabilità, scalabilità e sicurezza della rete, essenziali per le esigenze di applicazioni aziendali e di campus.

- **Router**

  Appartengono alla serie Huawei AR G3.

  I router Huawei AR G3, sono noti per la loro versatilità, offrendo funzionalità integrate di routing, switching, sicurezza, ideali per ambienti aziendali di piccole e medie dimensioni.

Figura 3.2.3 Router Huawei AR G3

- **Switch**

  Gli switch della serie Huawei S5700, sono progettati per reti aziendali e di campus, fornendo servizi di rete avanzati, sicurezza e gestione energetica efficiente. Sono in grado di gestire un alto numero di connessioni con elevate prestazioni, garantendo al contempo la sicurezza e la stabilità della rete.

Figura 3.2.4 Switch Huawei S5700

3.3 Configurazione dispositivi

La comunicazione tra i router è un aspetto cruciale della nostra configurazione di rete. Essa avviene in due scenari principali, ognuno con le proprie implicazioni e dinamiche:

• ***Collegamento Diretto***: In situazioni in cui due router sono collegati direttamente tra di loro, la preferenza di routing è impostata a 0, poiché il collegamento è ritenuto il percorso più affidabile. In questo caso, la comunicazione tra i due router è immediata e avviene senza l'intervento di protocolli di routing.

• ***Routing OSPF***: Quando due o più router non sono collegati direttamente, la comunicazione è gestita tramite il protocollo di routing dinamico OSPF (Open Shortest Path First). Questo protocollo permette ai router di scambiare informazioni sullo stato delle reti e calcolare il percorso ottimale per inoltrare il traffico.

Le tabelle di routing dei router contengono le rotte calcolate tramite OSPF, permettendo a ciascun router di determinare la migliore via di comunicazione per raggiungere le destinazioni finali.

3.3.1 Router

- **R1**

  Nome del sistema: R1

  Interfacce:

  1. GigabitEthernet0/0/0.16: Configurata con l'indirizzo IP 192.168.16.1/24.
  2. GigabitEthernet0/0/0.51: Configurata con l'indirizzo IP 192.168.51.1/24.
  3. GigabitEthernet0/0/1: Configurata con l'indirizzo IP 10.100.0.5/24.

  OSPF: Router ID impostato su 1.1.1.1, con reti OSPF annunciate nelle aree 192.168.16.0/24 e 192.168.51.0/24.

Figura 3.3.1 Router R1

- **R3**

Nome del sistema: R3

Interfacce:

1. GigabitEthernet0/0/0.23: Configurata con l'indirizzo IP 192.168.23.3/24.
2. GigabitEthernet0/0/0.36: Configurata con l'indirizzo IP 192.168.36.3/24.
3. GigabitEthernet0/0/1: Configurata con l'indirizzo IP 10.100.0.7/24.

OSPF: Router ID impostato su 3.3.3.3, con reti OSPF annunciate nelle aree 192.168.23.0/24 e 192.168.36.0/24.

Figura 3.3.2 Router R3

- **R4**

Nome del sistema: R4

Interfacce:

1. GigabitEthernet0/0/0.23: Configurata con l'indirizzo IP 192.168.23.2/24.
2. GigabitEthernet0/0/0.52: Configurata con l'indirizzo IP 192.168.52.2/24.
3. GigabitEthernet0/0/1: Configurata con l'indirizzo IP 10.100.0.8/24.

OSPF: Router ID impostato su 4.4.4.4, con reti OSPF annunciate nelle aree 192.168.23.0/24 e 192.168.52.0/24.

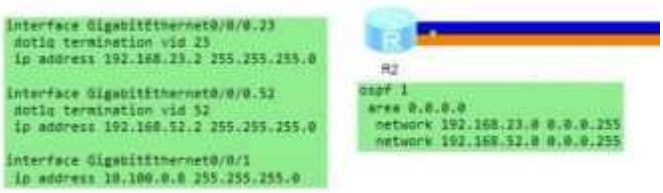


Figura 3.3.3 Router R4

- **R5**

  Nome del sistema: R5

  Interfacce:

  1. GigabitEthernet0/0/0: Configurata con l'indirizzo IP 192.168.51.5/24.
  2. GigabitEthernet0/0/1: Configurata con l'indirizzo IP 192.168.52.5/24.
  3. GigabitEthernet0/0/2: Configurata con l'indirizzo IP 10.100.0.9/24.

  OSPF: Router ID impostato su 5.5.5.5, con reti OSPF annunciate nelle aree 192.168.51.0/24 e 192.168.52.0/24.

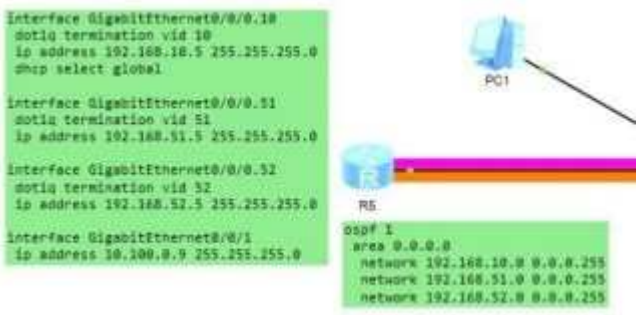


Figura 3.3.4 Router R5

- **R6**

  Nome del sistema: R6

  Interfacce:

  1. GigabitEthernet0/0/0: Configurata con l'indirizzo IP 192.168.16.6/24.
  2. GigabitEthernet0/0/1: Configurata con l'indirizzo IP 192.168.36.6/24.

3. GigabitEthernet0/0/2: Configurata con l'indirizzo IP 10.100.0.10/24.

OSPF: Router ID impostato su 6.6.6.6, con reti OSPF annunciate nelle aree 192.168.16.0/24 e 192.168.36.0/24.

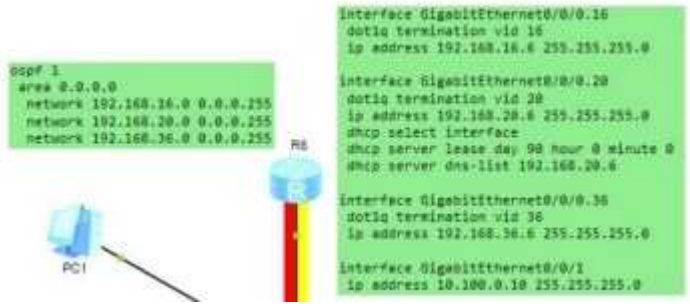


Figura 3.3.5 Router R6

3.3.2 Switch

La configurazione degli switch è un elemento critico della topologia. Questi switch collegano i router e le VLAN e devono essere configurati adeguatamente. Sono state utilizzate due modalità principali:

• ***Port Access***: Questa modalità consente ai dispositivi connessi di appartenere a una singola VLAN. Ad esempio, le porte dei router comunicanti con i PC sono configurate come access ports in modo che i pacchetti inviati e ricevuti su queste porte abbiano un tag VLAN specifico.

• ***Port Trunk***: Questa modalità consente la trasmissione di dati su più VLAN. Ad esempio, le interfacce tra gli switch e i router sono configurate come trunk ports, consentendo il passaggio di frames appartenenti a più VLAN.

- **SW1**

  Nome del sistema: SW1

  VLAN: Batch configurato per le VLAN 10, 16, 20, 23, 36, 51, 52.

  Interfacce:

  1. GigabitEthernet0/0/5: Configurata come **trunk** con permesso VLAN 51.
  2. GigabitEthernet0/0/6: Configurata come **trunk** con permesso VLAN 16.
  3. GigabitEthernet0/0/7 e 0/0/28: Configurate come **trunk** con permesso VLAN 16, 51.
  4. GigabitEthernet0/0/8: Configurata come **trunk** con permesso VLAN 10.
  5. GigabitEthernet0/0/15: Configurata come **trunk** con permesso VLAN 52.
  6. GigabitEthernet0/0/16: Configurata come **trunk** con permesso VLAN 36.

- **SW2**

  Nome del sistema: SW2

  VLAN: Batch configurato per le VLAN 10, 16, 20, 23, 36, 51, 52.

  Interfacce:

  1. GigabitEthernet0/0/5: Configurata come **trunk** con permesso VLAN 23, 52.
  2. GigabitEthernet0/0/6: Configurata come **trunk** con permesso VLAN 23, 36.
  3. GigabitEthernet0/0/7: Configurata come **access** con VLAN 20.

- **SW3**

  Nome del sistema: SW3

  VLAN: Batch configurato per le VLAN 10, 20.

  Interfacce:

  1. GigabitEthernet0/0/5: Configurata come **trunk** con permesso VLAN 10.
  2. GigabitEthernet0/0/6: Configurata come **trunk** con permesso VLAN 20.
  3. GigabitEthernet0/0/8: Configurata come **access** con VLAN 10.

4. GigabitEthernet0/0/9: Configurata come **access** con VLAN 20.
5. GigabitEthernet0/0/13: Configurata come **access** con VLAN 10.
6. GigabitEthernet0/0/14: Configurata come **access** con VLAN 20.

```
interface GigabitEthernet0/0/1
 eth-trunk 1
#
interface GigabitEthernet0/0/2
 eth-trunk 1
#
interface GigabitEthernet0/0/3
 eth-trunk 1
#
interface GigabitEthernet0/0/4
 eth-trunk 1
#
```

Figura 3.3.6 Configurazione link-aggregation

Possiamo poi vedere come le prime quattro porte di SW1 e SW2 siano implementate come porte eth-trunk. Ciò significa che è stata utilizzata la tecnologia del link-aggregation. Questa tecnologia ci permette di ottimizzare le prestazioni della rete combinando più collegamenti fisici in un unico collegamento logico. L'aggregazione di link consente di aumentare la larghezza di banda disponibile e migliorare la ridondanza della rete. Nel progetto sono stati utilizzati solo tre dei quattro collegamenti tra switch 1 e 2, poiché uno rimarrà come scorta in caso di malfunzionamento di uno dei primi tre. Questo approccio garantisce una maggiore affidabilità della connessione tra i due switch, consentendo alla rete di continuare a funzionare in caso di guasto su uno dei collegamenti principali.

```
#
interface Eth-Trunk1
 port link-type trunk
 port trunk allow-pass vlan 2 to 4094
 load-balance dst-ip
#
```

Figura 3.3.7 Configurazione porte eth-trunk

Inoltre, è importante notare che è stato configurato il meccanismo di load balancing sullo switch Huawei per dividere la comunicazione tra i due switch su più porte.

Questa configurazione ottimizza il carico di traffico sulla rete, distribuendo in modo equo il traffico tra i collegamenti disponibili, contribuendo così a migliorare le prestazioni complessive della rete e a evitare congestioni in caso di traffico intenso

3.3.3 Configurazione delle VLAN e delle Sub-Interface

Nella configurazione di rete descritta, sono state stabilite cinque VLAN distinte (VLAN 16, 36, 52, 51 e 23) per facilitare la comunicazione diretta tra specifici router. Ogni router è dotato di sub-interfacce sulla sua interfaccia GigabitEthernet0/0/0, assegnate a determinate VLAN come segue:

Router	Sub-Interface 1	Sub-Interface 2
R6	0/0/0.16	0/0/0.36
R5	0/0/0.51	0/0/0.52
R1	0/0/0.16	0/0/0.51
R2	0/0/0.23	0/0/0.52
R3	0/0/0.23	0/0/0.36

Figura 3.3.8 Assegnazione Sub-Interface ai router

- **R6** comunica attraverso le VLAN 16 e 36 tramite le sub-interfacce 0/0/0.16 e 0/0/0.36.
- **R5** è collegato alle VLAN 51 e 52 attraverso le sub-interfacce 0/0/0.51 e 0/0/0.52.
- **R1** utilizza le sub-interfacce 0/0/0.16 e 0/0/0.51 per connettersi alle VLAN 16 e 51.
- **R2** è configurato con le sub-interfacce 0/0/0.23 e 0/0/0.52 per le VLAN 23 e 52.
- **R3** si connette alle VLAN 23 e 36 attraverso le sub-interfacce 0/0/0.23 e 0/0/0.36.

Questa disposizione consente a ciascun router di stabilire comunicazioni selettive con altri router specifici, tramite il trasporto di frame etichettati (tagged) attraverso le VLAN assegnate.

In aggiunta a queste VLAN, sono state introdotte altre due VLAN, la 10 e la 20, utilizzate esclusivamente per simulare connessioni dirette tra personal computer (PC) e router. Queste connessioni simulano l'accesso dei PC alla rete da posizioni esterne. Nella configurazione logica, il PC1 è connesso al router 5, mentre il PC2 è connesso al router 6. Questo schema permette ai due PC di comunicare seguendo due percorsi distinti:

- Il percorso superiore per il PC1 include solo il router 1. I frame inviati dal PC1 sono inizialmente incapsulati e etichettati per la VLAN 10, grazie alla terminazione dot1q configurata sulla sub-interfaccia del router 5. Successivamente, questi frame vengono instradati dallo switch al router 1, che poi li dirige verso il router 6 per il recapito finale al PC2.

- Nel percorso inferiore, i pacchetti dal PC1 sono prima diretti al router 2, da cui passano al router 3 e, infine, al router 6, che si occupa di instradarli al PC2.

Questo approccio garantisce che i pacchetti possano essere trasmessi efficacemente tra i PC1 e PC2, indipendentemente dalla direzione del flusso di dati. (Figura 3.1)

3.3.4 Configurazione DHCP

La configurazione del protocollo DHCP nei router R5 e R6 è stata progettata per facilitare l'amministrazione degli indirizzi IP all'interno dell'ambiente di rete, adottando due approcci distinti per l'implementazione di questo servizio.

Configurazione DHCP su R5

Per il router R5, la configurazione DHCP è centralizzata attraverso l'utilizzo di un pool DHCP globale denominato "poolR5". Questo pool è configurato con i seguenti parametri:

- ***Gateway predefinito***: Impostato su 192.168.10.5, che funge da via d'uscita predefinita per i dispositivi client all'interno della rete.

- ***Rete***: Definita come 192.168.10.0 con una subnet mask di 255.255.255.0, specificando l'intervallo di indirizzi IP che possono essere assegnati ai dispositivi client.

- ***Durata del lease***: Configurata per una durata di 90 giorni, dopo i quali i dispositivi client devono rinnovare il loro indirizzo IP.

- ***Server DNS***: Impostato su 192.168.10.5, fornendo ai dispositivi client l'indirizzo del server DNS per la risoluzione dei nomi di dominio.

Inoltre, la sub-interfaccia GigabitEthernet0/0/0.10 su R5 è configurata per la VLAN 10 con un indirizzo IP di 192.168.10.5. La direttiva `dhcp select global` abilita il router a fornire indirizzi IP da questo pool globale a tutti i dispositivi client che ne fanno richiesta attraverso la rete, indipendentemente dalla specifica interfaccia attraverso cui si connettono.

Configurazione DHCP su R6

Per il router R6, l'approccio adottato per la configurazione DHCP è basato su interfaccia. La sub-interface GigabitEthernet0/0/0.20, destinata alla VLAN 20, è configurata con un indirizzo IP di 192.168.20.6. La direttiva `dhcp select interface` specifica che il servizio DHCP opererà in modo indipendente su questa interfaccia, assegnando indirizzi IP solo ai dispositivi connessi direttamente ad essa. La configurazione include anche:

- ***Durata del lease***: Impostata su 90 giorni, definendo il periodo dopo il quale un dispositivo client deve rinnovare il suo indirizzo IP.

- ***Server DNS***: Fornito agli host client con l'indirizzo 192.168.20.6 per la risoluzione dei nomi di dominio.

Considerazioni sulla Scelta della Configurazione DHCP

La decisione di utilizzare un pool DHCP globale su R5 o una configurazione basata su interfaccia su R6 dipende dalle specifiche necessità della rete e dalla strategia di gestione degli indirizzi IP. La configurazione centralizzata su R5 semplifica la gestione degli indirizzi IP offrendo un punto unico di configurazione e distribuzione, adatta per scenari in cui i dispositivi client possono connettersi attraverso molteplici interfacce. Al contrario, l'approccio adottato su R6 permette una maggiore flessibilità e controllo, assegnando indirizzi IP in modo specifico ai dispositivi client basati sulla loro interfaccia di connessione, ideale per ambienti in cui è necessaria una segmentazione più definita della rete.

3.4 Configurazione logica

Nella struttura della rete delineata, le tecnologie messe in atto garantiscono un flusso di dati coerente, seguendo principi ben stabiliti. In particolare, è stato sottolineato come il traffico dati tra il Router 5 e il Router 6 possa percorrere due vie distinte, ognuna con le proprie caratteristiche e implicazioni. La via preferenziale è quella superiore, caratterizzata da un minor costo in quanto implica il passaggio attraverso un solo router, minimizzando così la latenza e migliorando l'efficienza complessiva del percorso. Al contrario, la via inferiore comporta un attraversamento attraverso due router, incrementando il costo e la latenza associata.

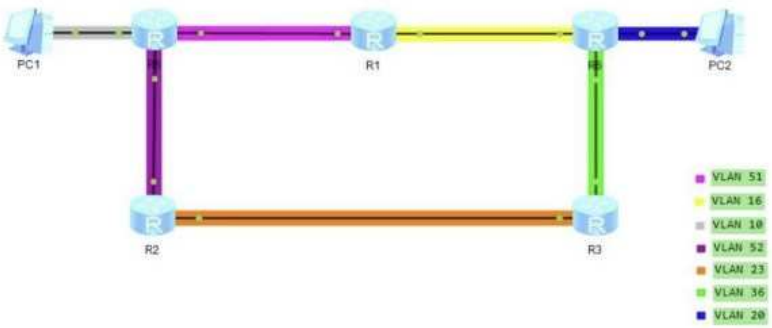


Figura 3.4 Configurazione Logica della rete

La visualizzazione grafica della disposizione logica della rete evidenzia chiaramente le scelte di instradamento effettuate dai router in base alle opzioni di percorso disponibili. La capacità di alternare tra il percorso superiore e quello inferiore tra il

Router 5 e il Router 6 sottolinea l'importanza delle decisioni di instradamento nel perfezionamento delle prestazioni della rete.

È importante riconoscere che questa disposizione logica riflette l'interazione complessa tra i protocolli di instradamento, i collegamenti diretti e la struttura fisica della rete, creando un'infrastruttura capace di supportare le esigenze comunicative con efficienza e affidabilità. La presenza di percorsi multipli aggiunge inoltre una dimensione di flessibilità, consentendo alla rete di adattarsi a cambiamenti o potenziali guasti, assicurando così la continuità del servizio.

Il principio di doppio percorso rappresenta un elemento chiave nel software di monitoraggio e gestione della rete che ho sviluppato. Questo strumento consentirà l'utilizzo di entrambe le vie disponibili per dirigere il traffico attraverso la configurazione di rete. Grazie a questo software, gli amministratori di rete avranno la possibilità di ottimizzare le comunicazioni, scegliendo dinamicamente le rotte in base alle necessità specifiche. Ad esempio, in presenza di congestione su un percorso, il software potrà reindirizzare il traffico su un'alternativa, mantenendo così un flusso di dati costante.

Questa capacità di adattamento fornisce un controllo avanzato sulle rotte OSPF, permettendo di regolare il flusso di traffico in funzione delle condizioni di rete e delle richieste specifiche, ottimizzando l'efficienza della comunicazione.

3.5 Problematiche riscontrate

Nel corso della configurazione della nostra rete, si è affrontata una sfida significativa legata al monitoraggio dell'efficienza delle sub-interfacce nei router, a causa della loro architettura complessa che prevedeva l'uso di sub-interfacce per gestire varie VLAN. La difficoltà principale risiedeva nella capacità dei router di fornire dati relativi all'utilizzo solo per le interfacce fisiche e non per quelle virtuali, ovvero le sub-interfacce.

Questo ostacolo ha imposto una riflessione critica sull'architettura di rete inizialmente pensata, portando alla necessità di rivedere e modificare l'intera topologia per ovviare a tale limitazione. La soluzione adottata ha riguardato, in particolare, i router R6 e R5: è stata presa la decisione di assegnare a ciascuna VLAN una specifica interfaccia fisica, anziché dividere una singola interfaccia fisica in

multiple sub-interfacce. Questo approccio ha richiesto una riconfigurazione accurata degli switch adiacenti per assicurare una corretta segregazione e instradamento del traffico nelle diverse VLAN e ha comportato un incremento nell'utilizzo delle porte disponibili sugli switch, richiedendo una gestione più attenta delle risorse di rete disponibili. Nonostante le sfide tecniche e logistiche incontrate, queste modifiche hanno permesso di ottenere una visione più chiara e dettagliata delle prestazioni di rete, migliorando significativamente la gestione delle VLAN e delle politiche di routing.

Grazie a questi interventi, la rete è stata ottimizzata per rispondere in modo più efficace e affidabile alle necessità del progetto, garantendo al contempo una maggiore facilità nel monitoraggio e nella manutenzione delle infrastrutture di rete.

CAPITOLO 4

Sviluppo Software

Questo strumento software è stato progettato per affrontare e risolvere complesse esigenze di configurazione, monitoraggio e ottimizzazione all'interno dell'infrastruttura di rete esaminata. L'obiettivo principale è stato quello di fornire agli amministratori di rete una soluzione efficace e intuitiva per il controllo delle operazioni di rete, con particolare enfasi sulla gestione dinamica delle rotte e sull'analisi delle prestazioni.

4.1 Strutturazione dei Requisiti

Requisiti Funzionali

Nel dettaglio, il software integra una serie di requisiti funzionali chiave, tra cui la selezione di specifici router per l'esame delle statistiche di utilizzo e la visualizzazione dello stato delle interfacce lungo determinati percorsi di instradamento. Questi requisiti sono stati definiti per garantire che gli utenti possano accedere facilmente a informazioni critiche e gestire in modo proattivo le configurazioni di rete.

Requisiti Non Funzionali

Dal punto di vista tecnico, il software sfrutta requisiti non funzionali come la connessione via protocollo SSH per l'acquisizione di dati di rete, e la capacità di adattare le rotte di traffico in base all'analisi delle condizioni di utilizzo delle interfacce.

4.2 Architettura del Software

L'architettura del software si basa sul modello Model-View-Controller (MVC), che facilita una chiara separazione delle responsabilità tra la gestione dei dati di rete (Model), la presentazione delle informazioni all'utente (View) e il controllo delle interazioni e delle operazioni (Controller). Attraverso l'utilizzo di classi specifiche come InterfaceObj per la rappresentazione delle interfacce di rete e ConnectionSSH per le connessioni sicure ai dispositivi di rete, il software assicura un'interazione efficiente con l'infrastruttura di rete, contribuendo significativamente alla gestione ottimale delle risorse di rete e alla risoluzione di eventuali criticità operative.

L'interfaccia utente del software di gestione della rete è stata progettata per offrire un'esperienza intuitiva e informativa, focalizzandosi sulla presentazione delle metriche essenziali e sul controllo della configurazione del router. Gli elementi principali dell'interfaccia includono:

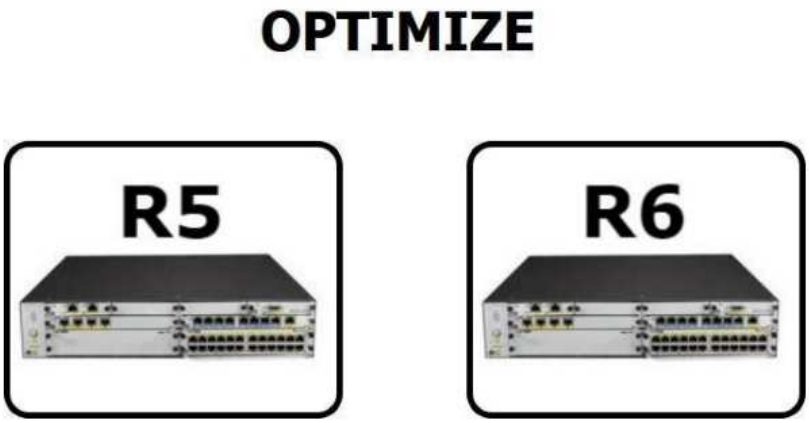


Figura 4.1 Router Management

- ***Etichetta Router Management***: Una QLabel che varia dinamicamente per riflettere il router selezionato dall'utente, visualizzando "R6 Management" o "R5 Management". Questo fornisce un feedback immediato sull'oggetto della gestione corrente.

- ***Immagine del Router***: Integrata in un QFrame, questa immagine statica arricchisce l'interfaccia utente, rendendo la schermata più accattivante e fornendo un contesto visuale.

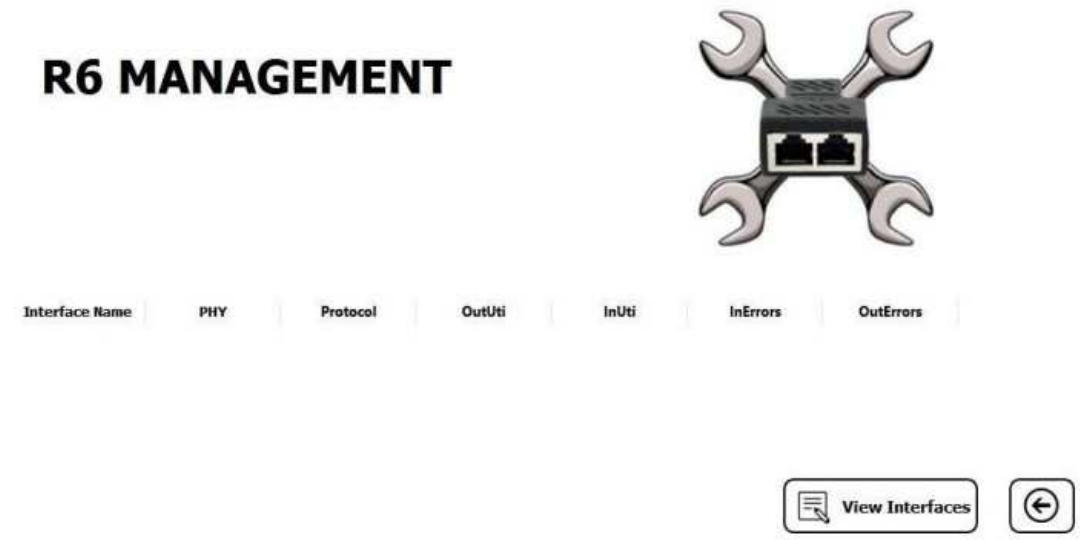


Figura 4.2 Tabella delle Statistiche

- ***Tabella delle Statistiche (QTableWidget)***: Questo componente centrale esibisce dettagliatamente le statistiche operative delle interfacce del router selezionato, attraverso colonne dedicate come:

- ***Interface Name***: Mostra l'identificativo dell'interfaccia, es. 0/0/0 o 0/0/1.
- ***PHY***: Indica lo stato fisico dell'interfaccia, che può essere "up" o "down".
- ***Protocol***: Riflette lo stato del protocollo a livello di collegamento dell'interfaccia.
- ***OutUti e InUti***: Presentano l'utilizzo medio della larghezza di banda in uscita e in entrata, rispettivamente.
- ***InErrors e OutErrors:*** Conteggiano i pacchetti di errore ricevuti e inviati dall'interfaccia.

4.3 Gestione degli Eventi e Interazione tra Interfacce

Nell'ambito dello sviluppo del software, un elemento cardine è rappresentato dalla capacità di intercettare e gestire le azioni intraprese dall'utente, come i click sui bottoni o altre forme di interazione con l'interfaccia grafica. Questo processo, noto come gestione degli eventi, è cruciale per rendere l'applicazione reattiva e interattiva.

4.3.1 Gestione degli Eventi

La rilevazione e la gestione delle azioni dell'utente si basa sull'impiego del framework PyQt, che facilita l'associazione tra le azioni dell'utente e le funzioni specifiche all'interno del codice. Questo collegamento è reso possibile mediante l'utilizzo di funzioni di connessione, che permettono di definire come l'applicazione debba reagire di fronte a determinati input dell'utente.

4.3.2 Collegamento delle Funzioni agli Eventi

Il processo di associazione di funzioni specifiche agli eventi generati dall'interazione dell'utente si realizza tramite l'uso della funzione `connect` di PyQt. Questa metodologia consente di legare eventi specifici, come il click di un bottone, a funzioni che definiscono le operazioni da eseguire in risposta. Per fare ciò, è essenziale importare le librerie necessarie, in questo caso PyQt5, che mette a disposizione una vasta gamma di strumenti per personalizzare l'interazione con gli elementi dell'interfaccia.

4.3.3 Interazione tra Interfacce

L'applicazione è progettata per consentire una navigazione fluida e intuitiva tra le varie schermate in risposta a specifiche azioni dell'utente. Ciò è reso possibile attraverso la definizione di funzioni di navigazione che modificano dinamicamente l'interfaccia visualizzata, facilitando il passaggio da una schermata all'altra.

Ad esempio, le funzioni `openViewInterface` e `openRouterChoice` permettono rispettivamente di aprire l'interfaccia di visualizzazione e quella di selezione del router, assicurando una transizione armoniosa tra le diverse parti dell'applicazione.

4.4 Corretta Visualizzazione dei Dati

Assicurare che i dati siano presentati correttamente e aggiornati in tempo reale nelle diverse interfacce è fondamentale per fornire all'utente informazioni accurate e aggiornate. Le funzioni sono state sviluppate con l'obiettivo di recuperare i dati necessari e aggiornare gli elementi dell'interfaccia di conseguenza.

L'integrazione di questi meccanismi ha contribuito a realizzare un'applicazione che non solo è interattiva e reattiva ma offre anche un'esperienza utente fluida e

soddisfacente, evidenziando l'importanza della gestione efficace degli eventi e delle interazioni tra le interfacce nello sviluppo di software.

4.5 Modulo ConnectionSSH.py

Comprende la classe Connection, che agisce come un controllore centralizzato per orchestrare le operazioni di business del software, in particolare per quanto concerne la connessione e l'interfaccia con i dispositivi di rete come i router. Questa classe è dotata di metodi dedicati al monitoraggio delle interfacce di rete e alla mappatura delle rotte di rete. Di seguito, verrà esplorato in dettaglio il metodo splitOutput, evidenziando la sua funzione all'interno del contesto del software.

4.5.1 Metodo splitOutput in Connection Class:

All'interno della classe Connection, il metodo splitOutput si distingue come una funzione statica incaricata di instaurare connessioni SSH con router specificati e di acquisire dati sulle interfacce di rete tramite l'esecuzione del comando 'display interface brief' sui dispositivi Huawei.

```
@staticmethod
def splitOutput(host, port, username, password, device_type):
    try:
        connection = ConnectHandler(host=host, port=port, username=username, password=password,
                                    device_type=device_type)
        output = connection.send_command('display interface brief')
        # Dividi l'output in righe
        linee = output.strip().split('\n')
        interfacce = []

        # Itera attraverso le righe per cercare le interfacce desiderate
        for linea in linee:
            colonne = linea.split()
            if len(colonne) >= 1 and colonne[0] in ["GigabitEthernet0/0/0","GigabitEthernet0/0/1"]:
                interfaccia = colonne[0]
                phy = colonne[1]
                protocol = colonne[2]
                inUti = colonne[3]
                outUti = colonne[4]
                inErrors = colonne[5]
                outErrors = colonne[6]
                # Crea un oggetto Interface
                interfaccia_obj = Interface(interfaccia, phy, protocol, inUti, outUti, inErrors, outErrors)
                interfacce.append(interfaccia_obj)

        connection.disconnect()
        return interfacce

    except Exception as e:
        print(f"Errore: {str(e)}")
```

Figura 4.5 Metodo slipOutput

Parametri richiesti dal metodo:
- ***host***: Si riferisce all'indirizzo IP del router su cui si desidera stabilire la connessione.
- ***port***: Designa la porta SSH attraverso cui avviene la connessione al router.
- ***username***: Identifica il nome utente necessario per l'autenticazione SSH.
- ***password***: Rappresenta la chiave di accesso per l'autenticazione.
- ***device_type***: Specifica il tipo di dispositivo, che per questo caso è indicato come 'huawei vrp'.

Funzionalità del Metodo:
Il metodo fa leva sulla libreria Python Netmiko, un toolkit per l'automazione di reti, per realizzare la connessione SSH con il dispositivo di rete. Una volta stabilita la connessione, il metodo procede all'invio del comando 'display interface brief' al dispositivo e si occupa di elaborare l'output per ricavare le informazioni necessarie sulle interfacce.

Dati estratti comprendono:
- ***Nome dell'interfaccia***: Identificativo unico dell'interfaccia di rete.
- ***Tipo fisico (phy)***: Indica la natura fisica dell'interfaccia.
- ***Protocollo (protocol)***: Specifica il protocollo di rete in uso.
- ***Utilizzo in ingresso (inUti)*** e ***in uscita (outUti):*** Mostrano la percentuale di utilizzo dell'interfaccia nelle direzioni di ingresso e uscita.
- ***Errori in ingresso (inErrors)*** e ***in uscita (outErrors)***: Registrano il numero di errori riscontrati nelle direzioni di ingresso e uscita.

Il metodo filtra le interfacce di interesse basandosi su criteri predefiniti, come per esempio i nomi 'GigabitEthernet0/0/0' e 'GigabitEthernet0/0/1'. Per ciascuna interfaccia selezionata, viene generato un oggetto Interface che incapsula tutte le informazioni ottenute. Gli oggetti delle interfacce vengono quindi raccolti in una lista e restituiti. In caso di eventuali errori durante il processo, questi vengono gestiti e stampati.

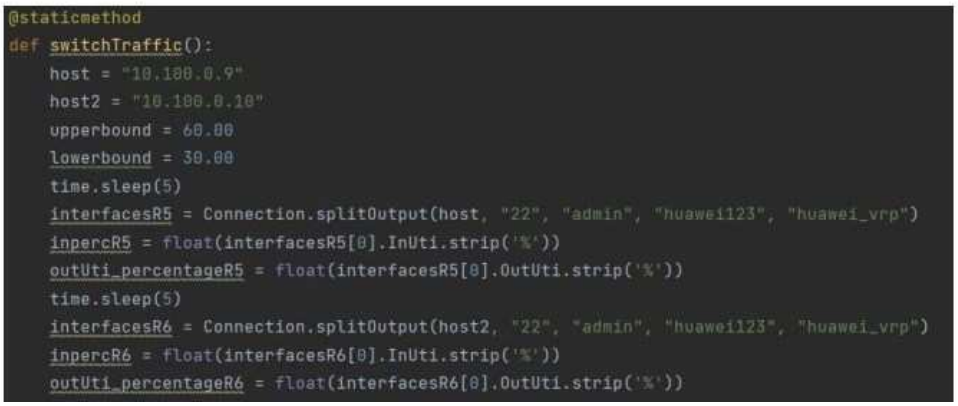
4.6 Metodo tracertRoute e switchTraffic

Il metodo `tracertRoute` all'interno della classe Connection del modulo ConnectionSSH è una funzione chiave progettata per eseguire l'analisi del percorso

di rete (tracert) sui router specificati, R5 o R6. Questa funzione verifica inizialmente quale dei due router è stato indicato tramite il parametro `sysname` e, a seconda del router scelto, stabilisce una connessione SSH utilizzando le credenziali appropriate. Una volta connesso, il metodo esegue il comando `tracert` verso un indirizzo IP di destinazione predeterminato, che è diverso per ciascun router (192.168.20.190 per R5 e 192.168.10.190 per R6), presumibilmente indirizzi di server cruciali per l'operatività del sistema. Dopo aver eseguito il comando, il metodo restituisce l'output del tracciamento e gestisce eventuali errori, facilitando così il debug.

Figura 4.6: Metodo tracertRoute.

Parallelamente, il metodo `switchTraffic` nella stessa classe Connection è incaricato dell'ottimizzazione del traffico di rete monitorando l'utilizzo delle interfacce sui router R5 e R6. Il metodo definisce degli indirizzi IP per questi router e stabilisce delle soglie di utilizzo (upperbound e lowerbound) per valutare l'efficienza del traffico. Attraverso l'utilizzo del metodo `splitOutput`, vengono raccolte le percentuali di utilizzo delle interfacce dei router. Qualora una di queste superi la soglia massima prefissata, il metodo procede alla configurazione di una rotta statica per indirizzi IP specifici al fine di redistribuire il traffico, stabilendo una connessione SSH per applicare le modifiche. Inversamente, se l'utilizzo di tutte le interfacce scende sotto la soglia minima, il metodo rimuove le rotte statiche precedentemente impostate per ritornare alla configurazione di rete originaria.



```
@staticmethod
def switchTraffic():
    host = "10.100.0.9"
    host2 = "10.100.0.10"
    upperbound = 60.00
    lowerbound = 30.00
    time.sleep(5)
    interfacesR5 = Connection.splitOutput(host, "22", "admin", "huawei123", "huawei_vrp")
    inpercR5 = float(interfacesR5[0].InUti.strip('%'))
    outUti_percentageR5 = float(interfacesR5[0].OutUti.strip('%'))
    time.sleep(5)
    interfacesR6 = Connection.splitOutput(host2, "22", "admin", "huawei123", "huawei_vrp")
    inpercR6 = float(interfacesR6[0].InUti.strip('%'))
    outUti_percentageR6 = float(interfacesR6[0].OutUti.strip('%'))
```

Figura 4.7: Metodo switchTraffic

Durante questo processo, eventuali errori sono catturati e registrati per il debug, e il metodo conclude l'operazione dopo un’attesa di 10 secondi.

Questi metodi illustrano un approccio dinamico alla gestione del traffico di rete, permettendo non solo di monitorare l'efficienza delle connessioni ma anche di intervenire attivamente per ottimizzare le prestazioni della rete in base alle condizioni di traffico in tempo reale.

CAPITOLO 5

Conclusioni e sviluppi futuri

La costruzione fisica della topologia di rete rappresenta il nucleo di questo studio, mirando a replicare le condizioni di una rete aziendale, con l'intento di valutare e perfezionare le strategie di connettività e le prestazioni delle interfacce di rete.

L'elaborazione della topologia è stata fondamentale per instaurare un contesto operativo simulato che riflette le sfide e le dinamiche delle reti moderne. Questo ambiente ha permesso di esaminare in dettaglio il comportamento delle interfacce di rete, l'efficienza della trasmissione dei dati e l'efficacia delle rotte di rete impostate, fornendo preziose intuizioni su come migliorare e ottimizzare le infrastrutture di rete esistenti.

Nel corso dello studio, l'accento è stato posto sull'ottimizzazione della rete, come dimostrato dall'analisi delle funzionalità quali il monitoraggio dinamico dell'utilizzo delle interfacce e l'adattamento delle rotte statiche per rispondere in modo flessibile alle variazioni del traffico. Questo approccio ha evidenziato l'importanza di un monitoraggio continuo e di una gestione adattiva delle risorse di rete per mantenere prestazioni ottimali e garantire la resilienza della rete.

Nonostante i risultati positivi, il progetto offre ampio spazio per miglioramenti e sviluppi futuri. Tra le potenziali aree di crescita vi sono l'approfondimento delle tecniche di ottimizzazione delle rotte di rete, l'estensione della topologia per includere una gamma più ampia di dispositivi e scenari di rete, e l'esplorazione di nuove strategie per rafforzare la sicurezza e l'affidabilità delle connessioni di rete.

Potrebbero inoltre essere esplorate strategie avanzate per il monitoraggio in tempo reale e la diagnostica delle prestazioni di rete, permettendo così di anticipare e risolvere proattivamente eventuali problemi, migliorando ulteriormente la stabilità e l'efficienza della rete.

In conclusione, il progetto ha fornito una solida base per la realizzazione di reti più efficienti e adattive, ponendo le premesse per future innovazioni nel campo della gestione delle infrastrutture di rete. Con un impegno continuo verso il miglioramento e l'innovazione, queste soluzioni potrebbero rivoluzionare il modo in

cui le reti vengono gestite, assicurando prestazioni elevate e connettività affidabile in ambienti di rete sempre più complessi e dinamici.

Bibliografia

ALADHAMI MAHMOOD MAZIN, M. K. A. R. M., RUHANI AB RAHMAN (2021), «Performance Analysis on Network Automation Interaction with Network Devices Using Python».

DONAHUE, G. A. (2007), «Network Warrior».

E ETHAN BANKS, R. W. (2017), «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks».

EDELMAN, J. (2018), «Network Programmability and Automation: Skills for theNext Generation Network Engineer». JIA, B. (2010), «Research of Physical Topology Discovery in Heterogeneous IP Networks with VLAN». KUMAR, B. K. (2019), «Fixing Network Security Vulnerabilities in Local Area Network».

KUROSE, J. (2022), «Reti di calcolatori e internet. Un approccio top-down». N., P. (2016), «Software-Defined Networking: Reconfigurable Network Systems in LAN Topology».

OLENA STARKOVA, K. N. O. Z. A. B. N. S., KOSTIANTYN HERASYMENKO (2022), «Virtualization and Programmability in Modern Networks in the Context of SDN Concept».

PAUL MIHĂILĂ, R. C. F. S., TITUS BĂLAN (2017), «Network Automation and Abstraction using Python Programming Methods».

STEVEN S. W. LEE, K.-Y. L. (2013), «Study of Dynamic Topology Change for Total Energy Consumption in Green IP Networks».

TANENBAUM, A. S. (2003), «Reti di calcolatori». 54 BIBLIOGRAFIA 55

Sitografia

• Huawei – https://e.huawei.com/it/solutions/enterprise-network
• Huawei ICS
• Paramiko – https://www.paramiko.org/
• Programmazione Python – https://docs.python.it/html/lib/module-threading. html
• Visure – https://visuresolutions.com/it/
• Visure – https://visuresolutions.com/it/blog/functional-requirements/
• Wikipedia – www.wikipedia.org
• Vega Training – https://www.vegatraining.eu/
• Huawei –https://support.huawei.com/enterprise/it/doc/EDOC1100008295/ a867611a/vlan-configuration
• Manage Engine – https://www.manageengine.com/it/network-monitoring/ what-is-snmp.html
• HTML.it – https://www.html.it/guide/guida-ad-ssh-il-protocollo-di-rete/
• Trovalost – https://trovalost.it/putty/
• HTML.it – https://www.html.it/disegnare-interfacce-con-qt-designer/
• Programmazione Python – https://www.python.it/wiki/show/qttutorial/
• IPERIUS – https://www.iperiusbackup.net/
• Iperf https://iperf.fr/
• Netmiko – https://pynet.twb-tech.com/blog/netmiko-python-library. html
• Librerie Python per l'automazione – https://datascience.eu/it/programmazione/ le-migliori-librerie-di-python-per-lapprendimento-automatico/

Ringraziamenti

Desidero esprimere i miei più sinceri ringraziamenti al Prof. Ennio Gambi, il mio relatore, per la sua preziosa guida, il sostegno costante e i consigli illuminanti che mi hanno accompagnato lungo tutto il percorso di questa tesi. La sua competenza e passione per la materia hanno notevolmente arricchito il mio percorso accademico e professionale.

Un ringraziamento speciale va anche all'ingegner Adelmo De Santis, il mio correlatore, per il suo contributo critico, l'incoraggiamento e il supporto. La sua disponibilità e il suo approccio costruttivo sono stati fondamentali nel superare le sfide incontrate durante lo sviluppo del lavoro.

Non posso dimenticare di ringraziare i miei amici, compagni di studi e di vita, che mi hanno sostenuto con la loro presenza, i loro incoraggiamenti e le loro inestimabili amicizie. La loro vicinanza e il loro supporto hanno reso questo viaggio meno arduo e decisamente più gioioso.

Infine, ma non per importanza, desidero dedicare un profondo ringraziamento alla mia famiglia. Ai miei genitori, per il loro amore incondizionato, la loro fiducia e il sostegno costante che mi hanno sempre offerto, permettendomi di perseguire i miei sogni. A mia sorella, per essere stata la mia forza e il mio rifugio nei momenti di dubbio. A tutti loro, va la mia eterna gratitudine per avermi sempre incoraggiato a dare il meglio di me.

Questo traguardo non sarebbe stato possibile senza il contributo di ciascuno di voi, e per questo vi sarò sempre grato.