UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in

Ingegneria Informatica e dell'Automazione

Progetto e configurazione di una rete enterprise

Project and configuration of an enterprise network

Relatore: Chiar.mo/a

Prof. Ennio Gambi

Tesi di Laurea di:

Enrico Gregorini

Correlatore:

Dr. Adelmo De Santis

A.A. 2020 / 2021

INDICE

1. Introduzione

Questa tesi ha come obiettivo quello di analizzare il progetto e la configurazione di una rete enterprise di telecomunicazioni; si andranno a simulare i dispositivi che compongono la maggior parte delle infrastrutture di rete per poterne apprezzare il loro impiego e funzionamento.

Le competenze, conseguite durante il corso HCIA Routing & Switching, sono state fondamentali al fine di comprendere il ruolo delle varie componenti della rete e di far in modo che esse collaborassero per rendere perfettamente funzionante il tutto. Infatti, l'obiettivo di questo elaborato doveva proprio essere quello di creare una topologia, che simuli una rete reale, attraverso l'utilizzo di hardware e software connessi tra loro mediante canali di comunicazione. Lo sviluppo di questo progetto è stato effettuato con l'utilizzo del software Huawei eNSP che, grazie a una macchina virtuale, permette di simulare il comportamento dei dispositivi di rete (switch, router, PC, client, server, ecc.).

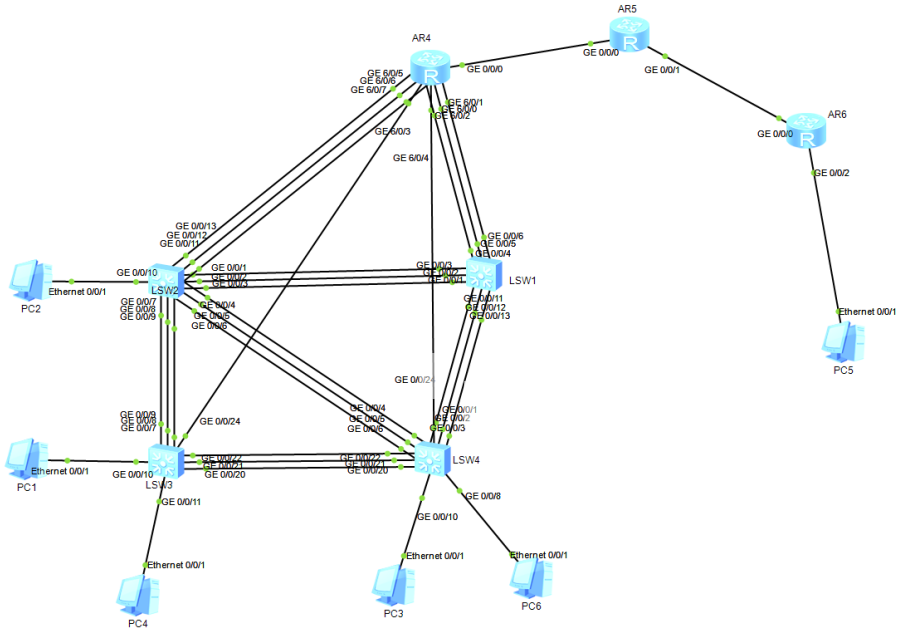


Figura 1: topologia di riferimento (screenshot da eNSP)

La seguente figura mostra la topologia di riferimento per il progetto. La rete è stata completamente configurata dal candidato e tutti i comandi e tecnologie che verranno mostrati nel corso dell'elaborato sono stati assegnati da quest'ultimo.

I dispositivi che compongono la rete sono:

- PC che fungono da terminali delle sottoreti che compongono la topologia;
- Switch 5700, dotato di 24 interfacce GigabitEthernet per collegare fisicamente più dispositivi tra loro;
- Router AR2200 costituito da 3 interfacce GigabitEthernet. Nel router 4 (R4) per poter gestire il gran numero di interfacce collegate agli switch 1 e 2 è stata aggiunta una card da 24 porte GE di livello 2.

Nel proseguo dell'elaborato verranno ripercorsi i passaggi principali che hanno portato al risultato finale così da mostrare le strategie, le motivazioni che hanno portato a prendere determinate decisioni, e il corretto funzionamento della rete. I capitoli verranno suddivisi in base al livello, dello stack protocollare TCP/IP, a cui ognuno di essi farà riferimento. Di seguito è riportata una figura che mostra esplicitamente i vari livelli che compongono lo stack protocollare TCP/IP, messo a confronto con quello ISO/OSI, e a cui si farà riferimento nel corso del testo. In particolare, verranno approfonditi alcuni protocolli e aspetti dei seguenti layer:

- Data Link, livello 2;
- Network, livello 3;
- Application, livello 5-7.

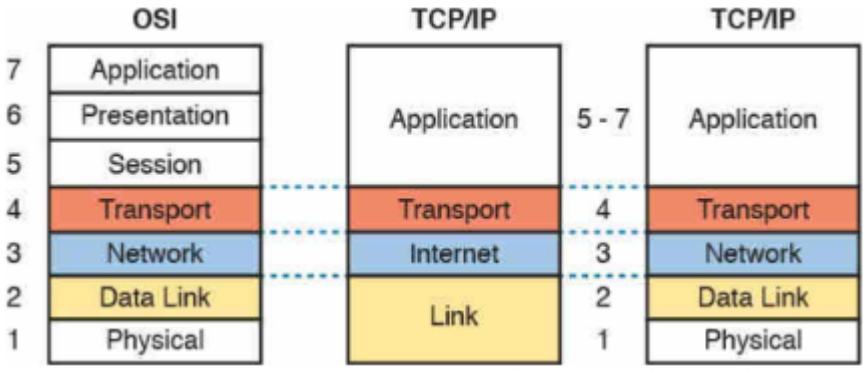


Figura 2: esempi stack protocollari (ISO/OSI e TCP/IP). Immagine tratta da: https://www.quora.com/How-many-layers-are-available-in-TCP-IP-protocols-layers

Infine, l'ultimo capitolo illustra alcuni meccanismi e protocolli adottati all'interno delle reti enterprise per garantire un certo livello di sicurezza sul traffico che circola tra i dispositivi.

2. Software utilizzati

I software utilizzati per lo sviluppo di questo progetto sono stati principalmente due:

- **Huawei eNSP**, simulatore di rete;
- **Wireshark,** software per il packet sniffer e cattura di pacchetti di rete**.**

2.1. eNSP

Huawei eNSP (enterprise Network Simulator Platform) è un programma in cui vengono simulati i comportamenti di una rete osservando le interazioni tra le varie entità (router, switch, host, link ecc.); in questo modo, in ambiente di laboratorio, è possibile analizzare i vari servizi offerti dalla rete ed è possibile cambiare o settare parametri per controllare il modo in cui quest'ultima risponde. Le simulazioni sono particolarmente utili in quanto permettono agli amministratori di replicare i modelli che si possono aspettare di vedere nel mondo reale, e quindi analizzare i risultati e utilizzarli durante il processo di sviluppo. Le caratteristiche principali del simulatore eNSP sono le seguenti *(1)*:

- **ottenimento di conoscenze.** Permette di acquisire familiarità con la famiglia di prodotti Huawei di rete aziendale. Consente di comprendere il funzionamento, la configurazione e l'ottimizzazione dei dispositivi in un ambiente virtuale senza influire in alcun modo su alcuna rete fisica;
- **ottimizzazione del comportamento di rete**. Rispecchiando accuratamente il comportamento della rete fisica, eNSP consente di identificare rapidamente i problemi e mettere a punto i componenti per migliorare l'efficienza e le prestazioni di quest'ultima;
- **valutazione della cyber-security.** Rende possibile capire come il sistema si possa comportare in caso di attacchi informatici. eNSP fornisce alcune funzionalità e supporta protocolli per la sicurezza della rete;
- **testing per servizi e applicazioni.** Fornisce un ambiente di test per valutare le prestazioni delle applicazioni e l'interoperabilità con altri software per monitorare la rete.

Di seguito è mostrata una figura (*figura 3)* che rappresenta l'interfaccia utente del programma.

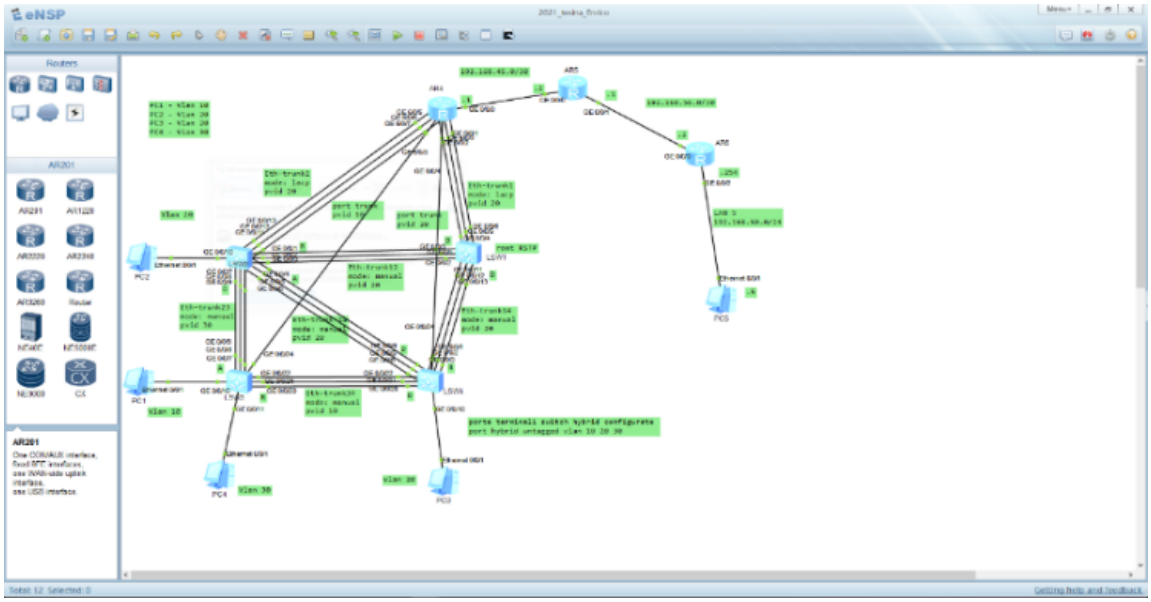


Figura 3: esempio interfaccia di eNSP

Sulla sinistra vengono mostrati i vari dispositivi, che rende disponibili il simulatore, divisi per categoria tra le quali vi sono: routers, switches, dispositivi per reti WAN (access point, ecc.), FireWall, End Devices (PC, server, ecc.) e Connections (cavi seriali, in rame, console, ecc.). Nella parte centrale poi viene mostrata la topologia che l'utente può costruire e infine nella parte in alto si trova una barra delle funzioni per, tra l'altro, avviare, controllare e terminare il funzionamento dei dispositivi.

2.2. WireShark

Wireshark è uno strumento utile per analizzare in maniera molto puntuale i pacchetti e i dati che vengono scambiati all'interno della rete. Viene definito un packet sniffer, ed è utilizzato, per lo più, per il troubleshooting, cioè l'attività di risoluzione dei problemi che riguardano la rete. Permette, grazie a un'interfaccia grafica, di catturare tutto il traffico di una o più interfacce di rete specificate. Per lo sviluppo della topologia del progetto è stato fondamentale perché associato al simulatore eNSP è stato possibile catturare il traffico di rete per controllare le informazioni sui protocolli utilizzati.

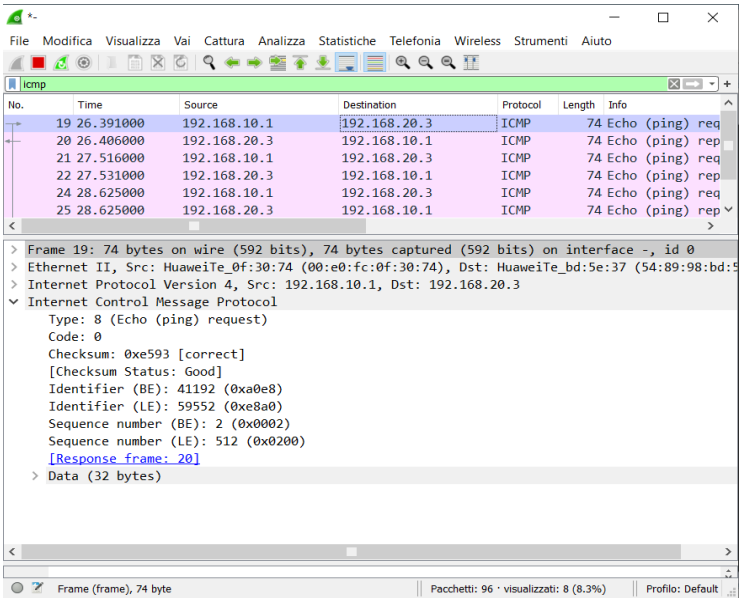


Figura 4: Esempio Cattura WireShark

Nella *figura 4* è riportata una cattura Wireshark dove si può osservare una serie di messaggi ICMP; con essi si possono analizzare le varie informazioni presenti, suddivise in base ai protocolli utilizzati nei vari livelli dello stack protocollare TCP/IP. Wireshark permette l'utilizzo di filtri in modo da evidenziare pacchetti di una certa tipologia rispetto che altri. I filtri si basano sui protocolli che vengono applicati per lo scambio di messaggi nella rete; ad esempio, nella figura si può osservare (dalla barra in verde in alto) come sia stato applicato il filtro ICMP che visualizza solo i pacchetti con l'omonimo protocollo a livello network. Per ogni pacchetto si possono analizzare i vari campi che lo compongono in base ai protocolli.

3. Livello Data-Link (L2)

Il livello Data-Link si presenta come uno strato di connessione tra il software dei processi che operano ai livelli superiori e l'hardware del livello fisico. La funzione principale del livello data-link è quella di permettere il trasferimento dei pacchetti provenienti dal livello di rete su reti di tipo diverso, che utilizzano differenti mezzi di comunicazione (cavi coassiali, doppini in rame, fibre ottiche) e differenti protocolli *(2)*.

In questa sezione vengono analizzati i protocolli e le tecnologie implementate, nella topologia, a questo livello dello stack TCP/IP, tra cui:

- Link Aggregation (LACP)
- Rapid Spanning Tree Protocol (RSTP)
- Virtual Local Area Network (VLAN)

3.1. Link Aggregation

Nella maggior parte delle reti enterprise è più che mai importante riuscire a migliorare la banda disponibile per il traffico all'interno di essa; per fare ciò è stata sviluppata una tecnologia, detta **link aggregation**. Per link aggregation si intende un meccanismo in cui si utilizzano più collegamenti fisici in parallelo tra due dispositivi, per creare una singola interfaccia logica. Questa nuova interfaccia logica è chiamata **Eth-Trunk** e ha una larghezza di banda uguale alla somma delle bande di tutte le interfacce fisiche membri. Oltre questo importante aspetto, link aggregation permette di implementare un processo di bilanciamento del carico e migliora l'affidabilità dei link. Questi due aspetti sono intrinsecamente presi in carico dall'eth-trunk in quanto:

- il bilanciamento del carico è facilmente ottenibile proprio dal fatto che più interfacce fisiche compongono il link e quindi il traffico può essere suddiviso su di esse, bilanciandolo;
- per quanto riguarda l'affidabilità dei link si può intuire che un link sarà inutilizzabile solo quando tutte le interfacce fisiche che lo compongono subiscono un guasto o sono fuori uso. Quindi le probabilità di guasto del trunk sono molto più basse rispetto ad un semplice collegamento.

Esistono due modalità per poter configurare link aggregation:

- manual mode: dove sia il trunk che tutte le interfacce fisiche del link sono manualmente aggiunte dall'amministratore e, vengono tutte utilizzate; in questa modalità non si ha la possibilità di settare alcuna interfaccia come backup, cioè che si attivi solo in caso di guasti.
- LACP mode: in questa modalità viene utilizzato il protocollo LACP (Link Aggregation Control Protocol) che effettua una negoziazione automatica dei parametri del link per decidere quali interfacce (manualmente aggiunte anche in questo caso) siano attive e quali siano inattive (le inattive sono quelle di backup). Viene anche detta modalità M:N in cui M rappresenta il numero di interfacce fisiche attive del trunk, mentre N quelle che forniscono ridondanza, quindi inattive.

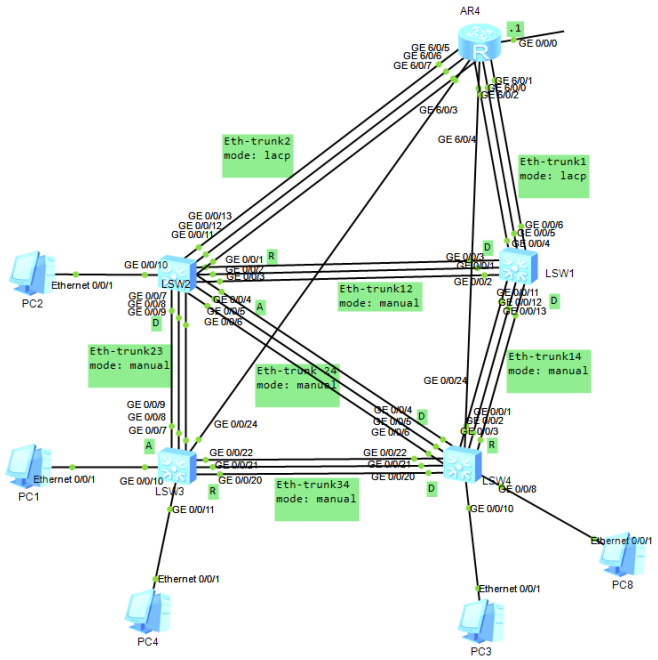


Figura 5: Visualizzazione interfacce Eth-Trunk all'interno della rete

All'interno della topologia di riferimento, vi è il "dominio di broadcast" *(3)* al di sotto del router 4, dove i collegamenti tra i dispositivi sono formati da più cavi disposti in parallelo. Così facendo, si garantisce una maggiore affidabilità in caso di guasti, in quanto un collegamento sarà completamente fuori uso solo quando tutti i cavi fisici che lo compongono saranno danneggiati. Questa è la situazione ideale per implementare link aggregation e configurare interfacce Eth-Trunk, che permettono l'utilizzo di più interfacce fisiche in un solo link contemporaneamente. Sono state

implementate entrambe le tipologie di trunk, ovvero attraverso manual mode e LACP mode, in ambiti diversi.

- I trunk tra gli switch (Eth-trunk 12, 14, 23, 24, 34) sono stati configurati mediante manual mode. Per fare ciò innanzitutto è stata creata l'interfaccia eth-trunk attraverso il comando int eth-trunk <num_int> che, di default, viene implementata con modalità manuale. Il comando per, eventualmente, configurare il trunk con modalità manuale è il seguente: mode manual load-balance. Dopodiché, all'interno dell'interfaccia appena creata, si devono aggiungere i collegamenti fisici al trunk con il comando trunkport <tipo_int> <num_int>. La *figura 6* mostra proprio un esempio di un'interfaccia Eth-Trunk (12) manuale; si può notare, tramite il comando display Eth-Trunk 12, che la voce 'WorkingMode' è settata a NORMAL e sta proprio a indicare la manual mode. Dall'output del comando si può notare dalla riga 'Operate status' se l'interfaccia trunk sia attiva (UP) o meno (DOWN). Oltre questo nella figura vengono mostrate anche le interfacce fisiche che compongono il link, e il loro stato nella colonna 'Status', della tabella centrale. Sotto la tabella poi sono specificate le interfacce attualmente attive che nell'esempio sono tre, ovvero tutte quelle che compongono il trunk.

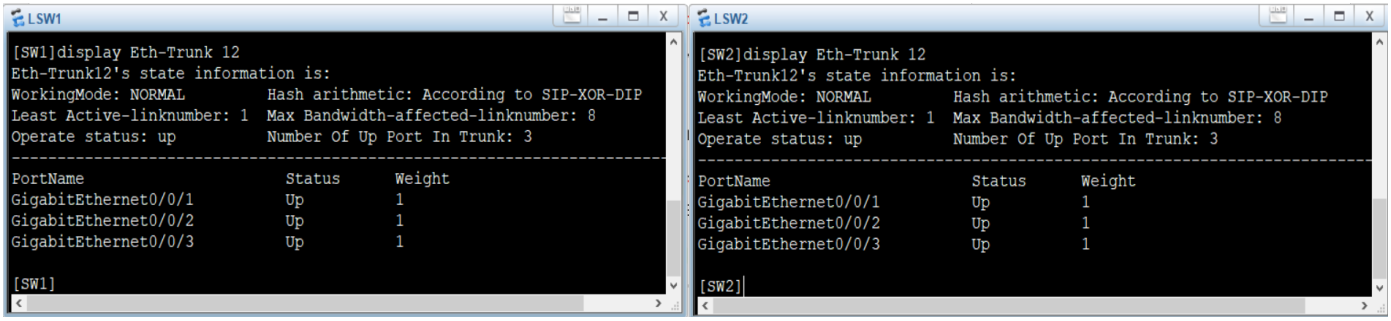


Figura 6: display interface Eth-Trunk <num_int>, configurate con modalità manual

- I due trunk collegati al router (Eth-trunk 1 e 2) sono implementati con LACP mode. Per fare ciò il primo passo è lo stesso dell'altra casistica (cioè si crea l'interfaccia Eth-trunk), poi si deve specificare che la modalità di utilizzo sia lacp tramite il comando mode lacp-static. In questa modalità si possono specificare opzioni aggiuntive come il numero minimo e massimo di interfacce attive rispettivamente con i comandi least/max active-linknumber <num_int_attive>. Dalla *figura 7* si può notare che si è specificato un numero massimo di 2 interfacce attive, così una risulta ridondante dato che il link è composto da 3

cavi GE; l'interfaccia attualmente unselected è la GE 0/0/6 per SW1, mentre la GE 6/0/2 per R4. Infine si devono aggiungere le interfacce fisiche al trunk nella medesima maniera della manual mode. Attraverso il comando display interface Eth-Trunk 1 viene evidenziata la modalità LACP poiché la sezione WorkingMode è configurata a STATIC, a differenza di NORMAL.

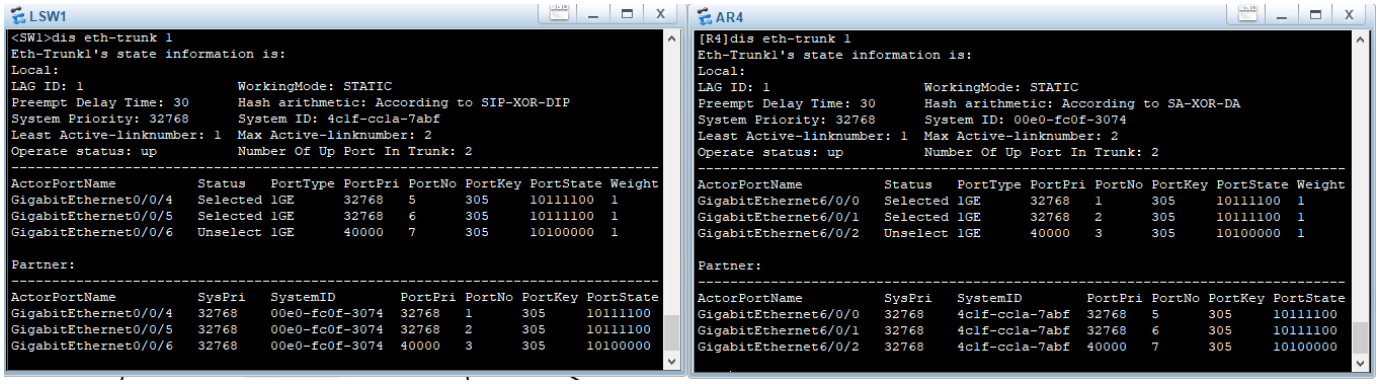


Figura 7: display interface Eth-Trunk <num_int>, configurate con modalità LACP

Nella *figura* 7, inoltre, si può notare che la configurazione LACP è sicuramente più dettagliata rispetto alla manual mode. Per ogni eth-trunk viene scelto un actor e un partner tra i due peer del link. Il primo è colui che "comanda" e quindi sceglie effettivamente quali interfacce siano attive e quali inattive, in base alla portPriority; vengono scelte, come attive, le interfacce con portPriority minore. Si nota infatti che la porta GigabitEthernet 6/0/2 è unselected proprio perché ha portPriority 40000, quindi maggiore rispetto alle altre due.

3.2. RSTP (Rapid Spanning Tree Protocol)

Link ridondanti sono estremamente comuni all'interno delle infrastrutture di rete per cercare di limitare al minimo la probabilità di avere collegamenti fuori uso, i quali, potrebbero bloccare il traffico. Il problema principale di questa ridondanza sono gli switching loop, ovvero percorsi chiusi che collegano i vari switch/bridge della rete; questi loop possono avere gravi conseguenze sull'efficienza del sistema. Tra i principali problemi che possono causare gli switching loop, vi sono:

- Broadcast storm. Situazione in cui frames broadcast sono continuamente inoltrati dalle interfacce degli switch, proprio per questi cicli; essi portano a un

enorme utilizzo di CPU da parte di questi dispositivi, causando, così, il blocco totale della rete;

- MAC instability. Gli indirizzi MAC memorizzati da una interfaccia di uno switch possono essere fasulli in quanto un frame, con stesso source MAC, può essere ricevuto da diversi collegamenti dati i loop esistenti nella rete.

Per risolvere questi problemi, all'interno della topologia si è scelto di implementare un protocollo che evita proprio il formarsi di questi loop, ovvero RSTP (Rapid Spanning Tree Protocol). Come dice il nome, questa è la versione "rapida" del suo antenato, STP (Spanning Tree Protocol); questo nome è dovuto proprio alla capacità di RSTP di riuscire ad essere molto più veloce in casi di congestione della rete *(4)*. STP, infatti, a causa della sua forte dipendenza dai timer, necessita di circa 50 secondi per poter correttamente adattarsi a danni o cambiamenti nella rete. Per questo motivo è stato scelto RSTP per gestire i vari loop che sono presenti nel segmento di rete, contenente gli switch, connesso al router 4.

Compreso il motivo della scelta di utilizzare RSTP, si mostra ora qual è il suo principio fondamentale. Esso elimina i loop presenti nella rete andando a disabilitare, cioè mettere in stato discarding, alcune interfacce dei dispositivi; queste interfacce, infatti, non sono abilitate a inoltrare traffico utente. Come vengono scelte le interfacce da disabilitare? Innanzitutto, viene scelta una radice, detto root switch, dell'albero che logicamente forma RSTP. Il root inoltra a tutti i dispositivi sottostanti (non-root switch) i messaggi, detti BPDU, che contengono le informazioni per permettere il corretto funzionamento del protocollo. Il root viene impostato con tutte le interfacce, o anche dette porte, con ruolo designated ovvero da cui vengono inoltrate le BPDU. Nei non-root viene selezionata una porta che permette di raggiungere la radice dell'albero con il costo minore, questa porta ha il ruolo di root port. Le porte designated e root possono inoltrare traffico utente, mentre infine vi sono le porte alternate e backup che sono quelle ridondanti e alle quali non è permesso inoltrare alcun tipo di messaggio utente.

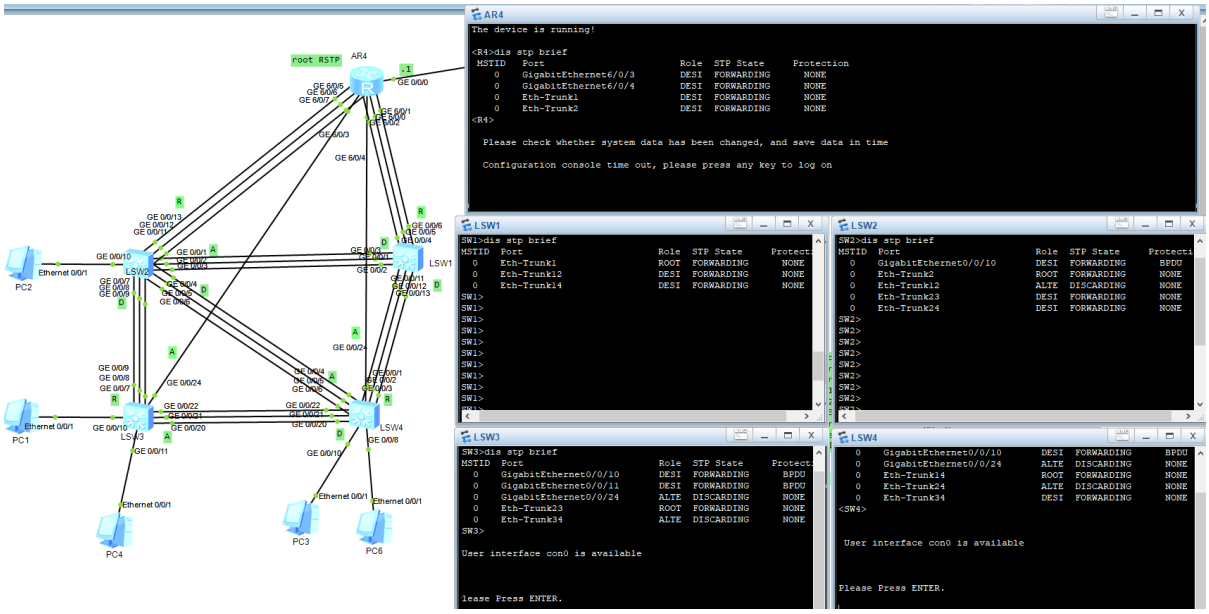


Figura 8: Visualizzazione ruoli delle porte per ogni switch

Una nota prima di analizzare la *figura 8* va fatta su R4. RSTP, come detto in precedenza, lavora a livello data-link, quindi con dispositivi come switch o bridge; quindi, l'albero logico non dovrebbe contenere router perché sono dispositivi di livello superiore (L3). R4 però è un'eccezione perché ad esso è stata aggiunta una card da 24 interfacce GigabitEthernet L2 che a tutti gli effetti lo rende un router con funzionalità anche di switch. Proprio per questo è stato inserito a tutti gli effetti nell'albero RSTP e, anzi, per semplicità e per simulare la classica struttura di albero il root switch è proprio R4. Si può notare dal fatto che tutte le porte collegate al router (in alto nella *figura 8*) sono designated (DESI) e quindi in stato forwarding, cioè possono inoltrare traffico utente. Dalla parte sinistra della figura si possono notare in verde i vari ruoli delle porte cioè: D-Designated, R-Root, A-Alternate. I ruoli delle porte si possono anche osservare sulla destra attraverso il comando dis(play) stp brief dove vengono visualizzate le informazioni principali delle interfacce dello switch tra cui: numero interfaccia, ruolo, stato, protection.

Si può quindi constatare che all'interno della topologia non vi sia alcun loop e quindi non ci sono possibilità che si verifichino problemi come broadcast storm o MAC instability.

3.3. VLAN (Virtual Local Area Network)

Le VLAN sono un insieme di tecnologie molto utili e diffuse che permettono di suddividere logicamente reti locali “flat”, cioè caratterizzate da un singolo dominio di broadcast, in più domini separati l’uno dall’altro. Uno dei motivi principali per cui si utilizzano le VLAN è il comportamento che hanno le reti Ethernet nel momento in cui aumenta considerevolmente il numero dei nodi all’interno di queste ultime; si dice infatti che “Ethernet scala male” proprio per il gran traffico generato dai messaggi broadcast. Per spezzare i domini di broadcast, evitando di dover inserire dispositivi di livello 3 (router), che sono costosi e introducono molto overhead, sono state introdotte queste tecnologie a livello data-link che generano delle sottoreti virtuali. Inoltre, le VLAN sono utili per riuscire a separare il traffico di infrastrutture fisiche che si trovano nella stessa area geografica, che ad esempio non devono poter comunicare tra loro per motivi di sicurezza. Le ACL, che saranno analizzate in un capitolo successivo, ad esempio possono essere applicate per riuscire a filtrare traffico proprio in base alle reti virtuali presenti nella struttura. Ovviamente ogni VLAN ha un proprio spazio degli indirizzi e un default gateway a cui gli host fanno riferimento per poter comunicare con network esterne, esattamente come viene fatto per le normali LAN. Questo discorso verrà approfondito nel capitolo 4, che descrive le tecnologie del livello di rete.

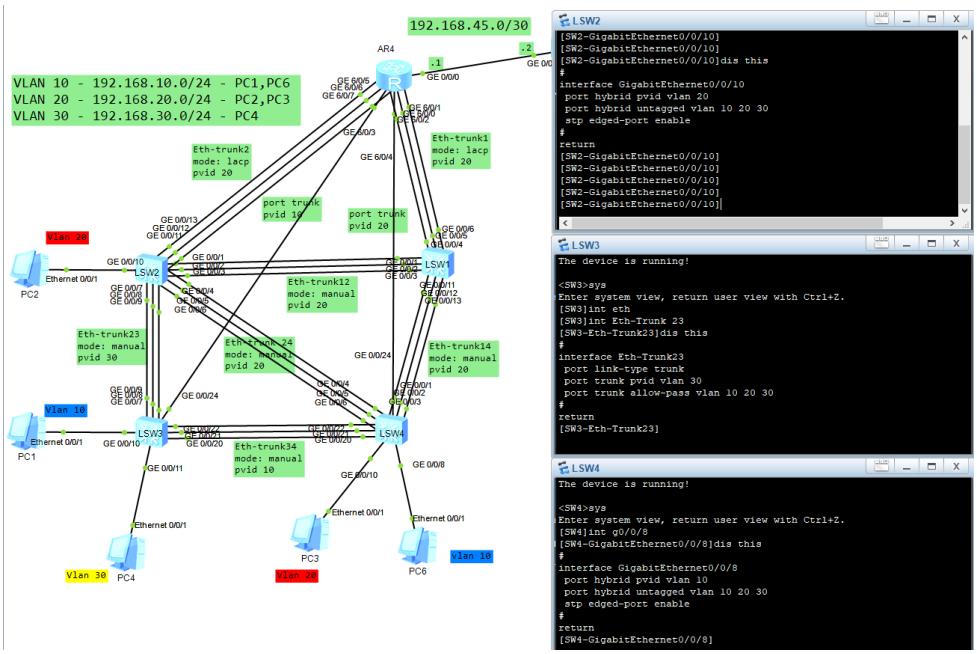


Figura 9:esempi di comandi per abilitare le VLAN nelle porte degli switch

Nel progetto preso come esempio, si nota che la maggior parte dei dispositivi di rete sono concentrati nell'area che sta a valle di R4 ed essa, in condizioni standard, sarebbe caratterizzata da un unico dominio di broadcast. Come già detto in precedenza, questo segmento di rete potrebbe avere problemi di sovraccarico per il grande traffico generato dai messaggi broadcast; quindi, si deve cercare una soluzione per suddividerlo in più domini di broadcast, indipendenti gli uni dagli altri. Inoltre, i calcolatori di questo segmento di rete potrebbero appartenere a dipartimenti distinti nonostante risiedano nella stessa area fisica della struttura, e quindi si deve cercare un modo per separare logicamente questi diversi settori. Per questi motivi sono state implementate tre VLAN diverse (evidenziate dalle label di diversi colori accanto ogni PC) che suddividono in tre domini di broadcast diversi il segmento di rete sopraccitato. Come si può notare dalla *figura 9*, alle 3 VLAN corrispondono altrettanti indirizzi di rete (aspetto che verrà approfondito nel capitolo 4) che le rappresentano:

- Vlan 10 - 192.168.10.0/24
- Vlan 20 – 192.168.20.0/24
- Vlan 30 – 192.168.30.0/24

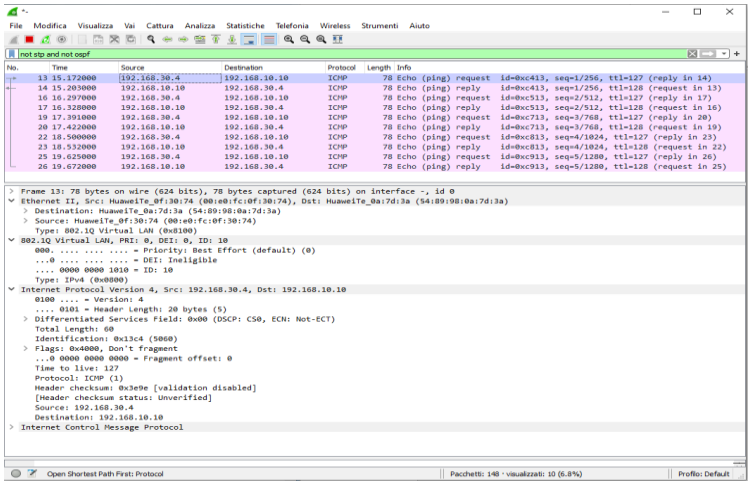
Questa tecnologia si basa su principi di livello data link, ovvero è implementata attraverso la modifica dei frames. Ciò si può notare dalla *figura 10* dove si può notare come il frame Ethernet ha un campo specifico per le VLAN, il cosiddetto "tag". Questi vengono detti frames tagged proprio perché, grazie a questo tag (lungo 4Byte), fanno riferimento alla VLAN di appartenenza. L'aggiunta del tag VLAN segue delle regole ben precise e dipende in larga parte dal tipo di porta a cui si fa riferimento. Ci sono 3 possibili tipologie di porte:

- **Access port**: solitamente sono le interfacce che collegano switch e terminal host; vengono sempre caratterizzate con una vlan tramite il vlan-id. Quando ricevono un frame untagged esse aggiungono a quest'ultimo il tag identificandolo con il vlan id della porta. Mentre quando devono inoltrare un frame all'host, esse rimuovono il tag solo se il vlan-id del messaggio corrisponde con quello della porta.
- **Trunk port:** rappresentano le interfacce tra i vari switch (o router) e anch'esse possono essere associate ad una vlan, tramite il pvid (Port Vlan ID) cioè il numero della vlan associata manualmente dall'amministratore alla porta trunk.

La sostanziale differenza con le porte access è il fatto che le trunk permettono l'inoltro di frame di più reti virtuali attraverso essi. I frames che vengono inoltrati nelle porte trunk sono tutti tagged, tranne per quelli che provengono dalla vlan che corrisponde al pvid.

- **Hybrid port:** sono le interfacce di default, all'interno dei dispositivi Huawei, e possono essere utilizzate, come dice il nome, sia in modo access che trunk (si può sempre configurare un pvid) con alcune differenze. Infatti, le porte ibride se utilizzate come porte access possono inoltrare frames untagged agli host, anche se essi appartengono a una vlan che non corrisponde al pvid; così facendo si può risparmiare traffico in quanto i frame hanno dimensione minore. Invece utilizzate come porte trunk la sostanziale differenza è che tutti i frame, anche quelli che corrispondono al pvid, vengono inoltrati con il tag.

Proprio per cercare di minimizzare il traffico nella rete, sono state implementate le ultime due tipologie di porte: porte trunk tra gli switch e tra switch e router (esempio *figura 10*) mentre porte ibride in tutti i collegamenti tra switch e pc (esempio *figura 11*). Quindi è possibile vedere, attraverso le seguenti catture, il funzionamento delle porte.



Il primo esempio mostra i frame che passano su una porta trunk in cui la vlan sorgente è diversa dal pvid e infatti si può notare come i messaggi siano tagged.

Figura 10: Cattura WIreShark su pacchetti di tipo tagged (porta trunk)

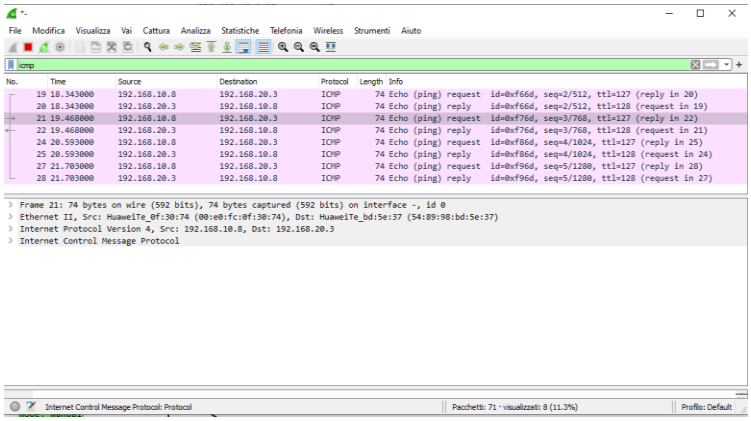


Figura 11: Cattura WIreShark su pacchetti di tipo untagged

A differenza di prima, se si cattura il traffico di una porta ibrida (int g0/0/10 di SW4) si può notare come, nonostante sia un ping tra vlan diverse, i frames passino comunque untagged, andando a ridurre così la dimensione dei messaggi inoltrati nella rete.

4. Livello di Rete (L3)

Lo strato di rete si occupa della consegna dei messaggi (a questo livello vengono chiamati pacchetti o datagram) da un nodo mittente ad un nodo destinatario attraverso diverse reti, determinando il percorso migliore per inoltrare i dati. Il livello 3 quindi, si occupa di:

- problemi legati al routing dei pacchetti tra i nodi che vengono identificati tramite l'utilizzo di un protocollo di indirizzamento logico dei pacchetti detto IP (Internet Protocol) *(5)*;
- permettere lo scambio dei pacchetti verso il livello Data-Link (inferiore), in fase di incapsulamento, e verso il livello di trasporto (superiore), in fase di decapsulamento.

In questo capitolo sono approfonditi alcuni aspetti, e protocolli utilizzati, riguardanti proprio il livello di rete, tra cui:

- VLAN routing
- Indirizzamento IP
- IP routing

4.1. VLAN routing

Come è stato detto nel capitolo precedente, le VLAN sono sottoreti virtuali divise tra loro e quindi possono scambiare pacchetti solo attraverso un dispositivo L3. Ci sono 2 possibilità per garantire la comunicazione tra diverse VLAN:

- Sub-interfaces: consiste nel suddividere l'interfaccia fisica, che, collega le VLAN, al router di riferimento in tante sotto interfacce logiche, quante sono le sottoreti definite nel segmento di intersse. Ad ogni sotto interfaccia assegniamo un indirizzo che fungerà da default gateway per gli host di quella VLAN. Questa soluzione necessita che il link che collega VLAN e router sia di livello 3, a cui quindi è possibile assegnare un indirizzo IP.
- VLANIF: si basa su switch che hanno capacità l3. Questa funzione infatti non necessita, obbligatoriamente, di un router ma può essere implementata anche

da uno switch. All'interno del dispositivo si crea un'interfaccia logica, chiamata vlanif, per ogni VLAN a cui si assegna un indirizzo IP che anch'esso, come nel caso precedente, sarà il default gateway per gli host della VLAN.

Nella topologia la comunicazione tra diverse VLAN è stata implementata con la seconda metodologia, ovvero con le interfacce VLANIF. Queste interfacce logiche sono state create all'interno di R4, che può essere visto come uno switch perché ad esso è stata aggiunta una card da 24 interfacce GE di livello 2. Queste 24 interfacce sono state aggiunte per creare i collegamenti tra R4 e i 4 switch che compongono il segmento di rete sottostante. Dato che esse sono interfacce di livello 2, non possono essere assegnati indirizzi IP né direttamente a queste ultime né a sub-interfaces create al loro interno. Proprio per questo la scelta è stata forzata ed è stata implementata la seconda possibilità, cioè sono state configurate le interfacce VLANIF all'interno di R4. Sono state create 3 interfacce, una per ogni VLAN presente nella rete: VLANIF 10, 20, 30.

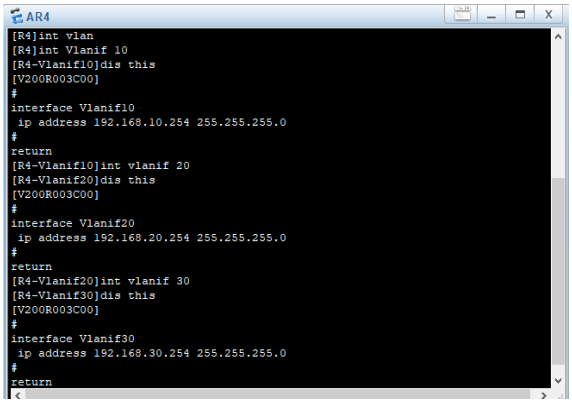


Figura 12: Assegnazione indirizzi IP alle interfacce vlanif di R4

Nella *figura 12* vengono mostrati i comandi per creare e configurare queste interfacce logiche. si creano in system-view con il comando interface Vlanif <id-vlan>. si assegna normalmente l'indirizzo IP all'interfaccia, con ip address <indirizzo> <subnet mask>. I 3 indirizzi mostrati in figura sono i default gateway per gli host delle corrispondenti VLAN.

Quale percorso seguono i pacchetti diretti verso una VLAN diversa dal mittente? Innanzitutto, quando un host vuole inviare un pacchetto che ha come destinazione un indirizzo IP che non appartiene alla stessa rete, esso lo inoltra al proprio default gateway. È stato detto che per ogni VLAN, R4 funge da default gateway quindi tutti i datagrammi, destinati a VLAN diverse dalla propria, sono inoltrati alla interfaccia VLANIF che corrisponde alla sottorete del mittente. Il router poi, visualizza la VLAN di appartenenza del destinatario attraverso il suo indirizzo IP e manda il pacchetto alla interfaccia VLANIF interessata. Dopodiché, il pacchetto è inoltrato al nodo finale.

Il percorso che segue un pacchetto può essere mostrato tramite il comando ICMP tracert *(6)*. Facendo un tracert da PC2 (con indirizzo 192.168.20.253) a PC1 (con indirizzo 192.168.10.253) si può notare come i pacchetti passino per l'interfaccia vlanif 20 di R4 (con indirizzo 192.168.20.254). Dopo di che i dati vengono inoltrati a PC1 passando per l'interfaccia vlanif 10, che però non compare nel percorso del tracert perché è configurata nello stesso dispositivo L3, ossia router 4.

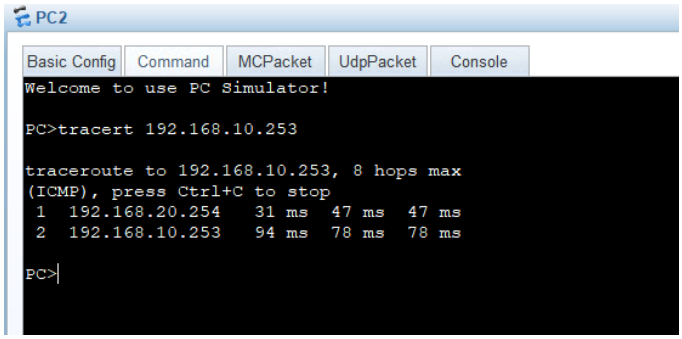


Figura 13: Output del comando ICMP tracert 192.168.10.253 (indirizzo appartenente ad un host di una VLAN diversa rispetto a quella del mittente)

4.2. Indirizzamento IP

Fino ad ora si è sempre parlato di concetti, che si basano sul livello data-link. Salendo di uno strato, si arriva al livello di rete (layer 3) dove si introducono nuovi identificatori logici per i dispositivi, gli indirizzi IP *(7)*. IP è un protocollo di livello 3 che definisce indirizzi logici che identificano univocamente calcolatori e interfacce di dispositivi L3 (router o switch L3). Per permettere a host di diversi segmenti di rete di comunicare, infatti, è necessario stabilire un path logico tra i diversi nodi della rete che permetta appunto ai pacchetti di essere inoltrati nella maniera corretta. Ogni dominio di broadcast ha il proprio indirizzo di rete (caratterizzato dalla subnet mask *(8)*, che serve a identificare il numero di host assegnabili all'interno della rete). Quindi, per prima cosa, si deve assegnare un certo spazio degli indirizzi ad ogni segmento di rete. Questo capitolo approfondisce proprio questo aspetto, ovvero, come e perchè si è suddiviso lo spazio degli indirizzi che contraddistingue la rete.

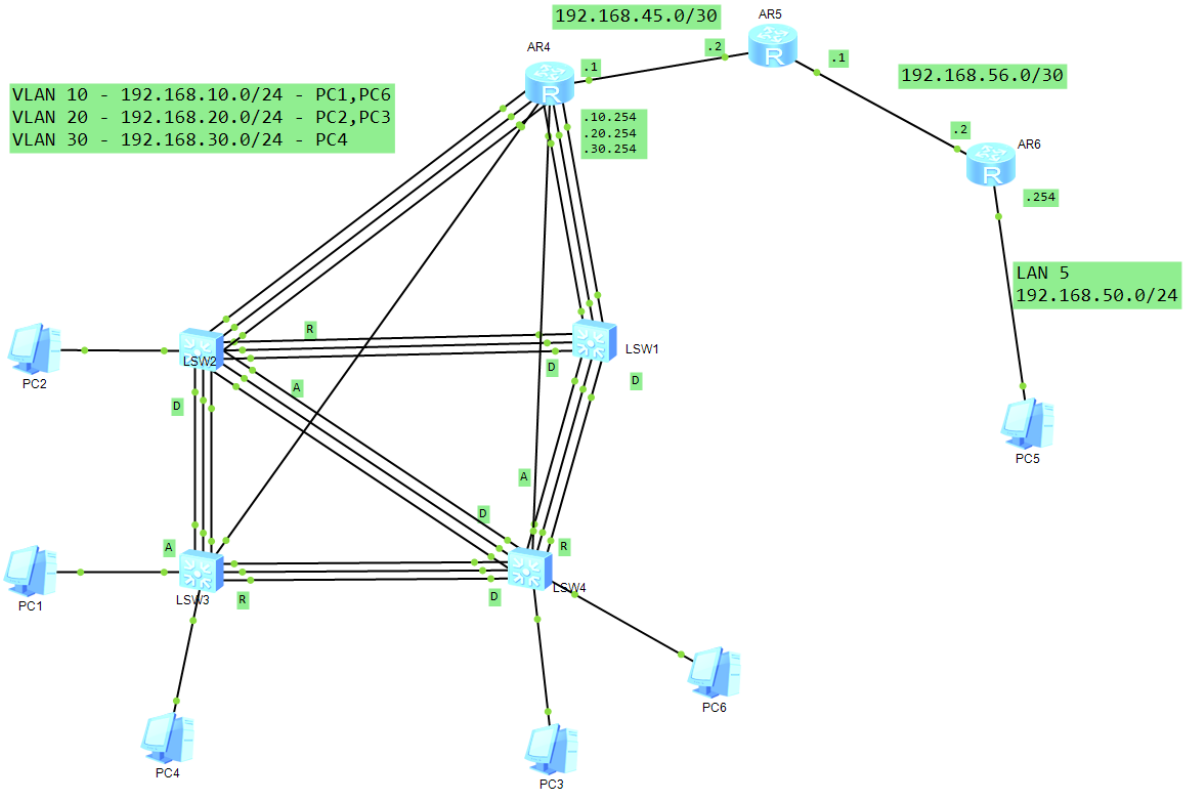


Figura 14: Spazi degli indirizzi assegnati ad ogni sottorete della topologia

Per riassumere le scelte prese viene riportata la *figura 14,* che mostra i vari indirizzi per ogni segmento presente nella topologia. Si può notare che ci sono complessivamente 6 reti distinte, che contraddistinguono altrettanti domini di broadcast, ognuno dei quali è caratterizzato da uno spazio degli indirizzi. Va specificato che lavorando in una rete enterprise, cioè una struttura privata, sono stati utilizzati indirizzi privati, e in particolare gli indirizzi utilizzabili e assegnabili appartenevano alla seguente macro-rete: 192.168.0.0/16.

Partendo da sinistra, si può osservare come sia importante il discorso fatto in precedenza per le VLAN, in quanto avendo create tre reti virtuali sia necessario assegnare tre diversi indirizzi di rete ad ognuna di loro. Nella maniera più intuitiva possibile, è stato assegnato un indirizzo di rete che nel terzo Byte riporta proprio il vlan-id associato, lasciando 8 bit per il campo host in modo che all'interno di ogni VLAN sia possibile indirizzare fino a 2^8-2 (254) host diversi. Così facendo si tiene un certo margine di incremento per quanto riguarda gli host connessi ad ognuna di queste VLAN, senza dover cambiare spazio di indirizzo.

Dopodiché vengono presi in considerazione i due segmenti che collegano i 3 router presenti nella rete. Ogni link che collega direttamente 2 router caratterizza una vera e

propria rete alla quale deve essere assegnato un indirizzo. Per il collegamento tra R4 e R5 è stato assegnato l'indirizzo 192.168.45.0/30 che ricorda, nel terzo Byte, i numeri dei due router che mette in collegamento; è stata utilizzata una subnet mask 30 perché essendo un collegamento p2p sono sufficienti 2 host indirizzabili infatti 2^2-2=2. Il link tra R5 e R6 è stato associato all'indirizzo 192.168.56.0/30 caratterizzato sempre da una maschera /30 perché rappresenta un collegamento p2p tra 2 router.

Infine, alla parte destra della topologia, occupata dalla LAN 5, è stato assegnato lo spazio 192.168.50.0/24 dove sono indirizzabili 254 host diversi.

I gateways delle Vlan e della LAN 5 sono sempre l'ultimo indirizzo assegnabile della rete, cioè il .254, come è evidenziato in figura.

4.3. IP routing

Il precedente capitolo ha trattato dell'indirizzamento, cioè del modo con cui sono stati assegnati gli spazi degli indirizzi per ogni segmento di rete. Il problema è che per ora possono entrare in comunicazione solo host della stessa LAN; quando si tenta di raggiungere segmenti che non sono direttamente collegati al proprio default gateway, i pacchetti non riescono a raggiungere la destinazione. Si deve quindi introdurre una tecnica per permettere ai router di conoscere gli indirizzi di tutte le sottoreti presenti, e quindi consentire l'inoltro di traffico utente verso qualsiasi host presente. Questa tecnica prende il nome di IP routing. Il routing, come ricorda la parola, cioè instradamento, non è nient'altro che la funzione che decide verso quale interfaccia inoltrare un determinato pacchetto ricevuto dal dispositivo L3 che la implementa; in poche parole, è una funzione che decide il percorso che il traffico della rete dovrà percorrere in base all'indirizzo sorgente e di destinazione. Le informazioni riguardanti il routing sono immagazzinate dai dispositivi, i quali le elaborano e le riportano all'interno di una tabella, la cosiddetta tabella di routing. In questa tabella sono riportati i dati principali per ogni rotta presente nel dispositivo; di seguito, nella *figura 15*, si mostra un esempio di tabella di routing presa direttamente da un router della topologia di riferimento (R4).

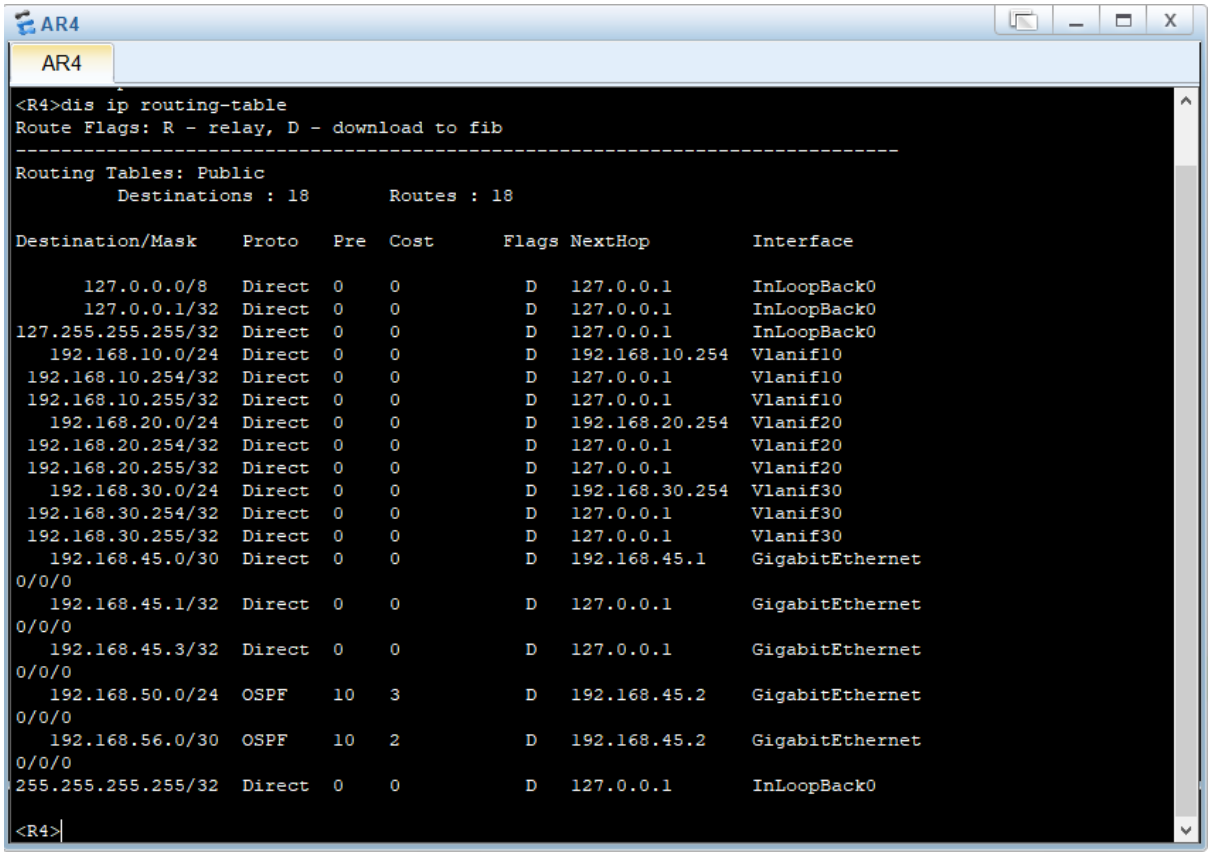


Figura 15: Visualizzazione tabella di routing di un router (R4)

Come si può notare dalla colonna proto della tabella di routing, esistono diverse modalità/protocolli per implementare l'instradamento dei pacchetti. Le rotte, identificate da ogni riga della tabella, che hanno proto=direct sono i segmenti direttamente collegati al router (ad esempio per R4 i segmenti delle Vlan sono direttamente collegati ad esso tramite le interfacce Vlanif). Le altre rotte sono quelle generate dinamicamente dal protocollo **OSPF (Open Shortest Path First)** che è stato ritenuto migliore di tutti gli altri per questa topologia. La scelta di questo protocollo è giustificata anche dal valore del campo preference (pre) della tabella di routing, che determina la priorità dei protocolli con cui la rotta è stata generata; più è basso il valore di preference e più è alta la priorità della rotta. Alcuni esempi del valore di default di pre (preference) sono:

- Direct: 0
- OSPF: 10
- Static: 60
- RIP: 100

Come si può notare, eccezion fatta ovviamente per i segmenti direttamente collegati (Direct), OSPF è il protocollo con la preference più bassa. Si tratta di un protocollo dinamico ovvero riesce automaticamente a ricalcolare le rotte per il traffico in casi di guasti da parte di alcuni dispositivi o link della rete. Perciò, è preferibile rispetto a configurare rotte in maniera manuale attraverso lo static routing perché, in questo caso, l'amministratore dovrebbe aggiornare le tabelle di routing ogni volta che la rete subisce un cambiamento. OSPF inoltre è preferibile a RIP, un altro protocollo di routing dinamico, per diversi motivi:

- il primo (OSPF) supporta reti di dimensioni abbastanza importanti. Il secondo (RIP) riscontra dei limiti da questo punto di vista, infatti presenta un hop count=15 che rappresenta il numero massimo di router che un pacchetto può attraversare nel suo percorso;
- OSPF genera un traffico molto inferiore rispetto a RIP per far in modo che i dispositivi tengano costantemente aggiornate le proprie tabelle di routing;
- grazie ad OSPF si ha un tempo di convergenza minore al verificarsi di modifiche topologiche *(9)* rispetto a RIP.

Vediamo ora come agisce OSPF sui vari router e come riesce a scambiare informazioni tra essi. Innanzitutto, va specificato che, con OSPF, la rete può essere suddivisa in più aree, ognuna delle quali ha i propri dati specifici diversi da quelli delle altre aree. OSPF, però, fa in modo che la topologia adatti un'architettura a stella in cui deve essere presente un'area centrale, detta backbone area o area 0, a cui tutte le altre aree devono essere connesse. Così facendo la backbone area è sempre informata su tutti i cambiamenti della rete e può facilmente aggiornare le aree limitrofe. Nell'esempio riportato, però, la topologia ha dimensioni contenute quindi non è stato necessario suddividerla in sottoaree; infatti, è presente solo l'area centrale ovvero l'area 0. All'interno di questa area i router che ne fanno parte, formano delle relazioni (adiacenze) in modo da scambiarsi le informazioni sulle reti collegate ad ognuno di essi. Così facendo, OSPF elabora (per ogni area) queste informazioni e forma un LSDB (Link State DataBase) da cui i router prelevano le informazioni per popolare la propria tabella di routing. Quindi, ogni area ha il proprio LSDB e invece ogni router della rete ha la propria tabella di routing, diversa ognuna dalle altre (anche router della stessa area hanno tabelle differenti).

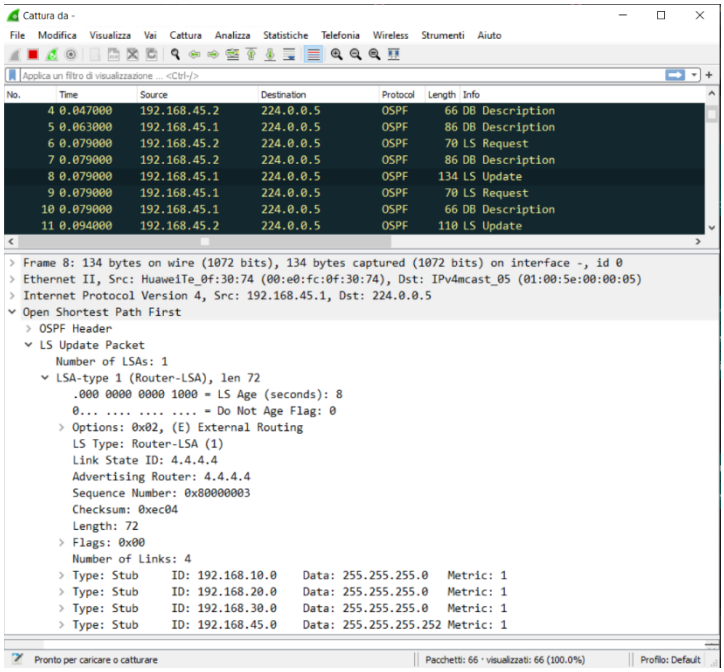


Figura 16: Cattura WireShark dell'interfaccia GE 0/0/0 di R5 sullo scambio di pacchetti OSPF

Dalla *figura 16* è possibile osservare lo scambio di pacchetti (LS Request, DB description, LS update) dei router per scambiarsi le informazioni che venivano citate precedentemente. In questa cattura, specialmente, si evidenziano (in basso nei dettagli di un LS Update) che il router 4 annuncia le reti a lui collegate ovvero le 3 VLAN (192.168.10.0/24 192.168.20.0/24, 192.168.30.0/24) e 192.168.45.0/30 cioè la LAN tra i due router.

Nella pratica, OSPF è molto semplice da implementare. Per poter utilizzare il protocollo si devono specificare, nel router, una serie di parametri, quali:

- l'identificativo (un numero) del processo o istanza OSPF a cui si fa riferimento;
- un eventuale router-id che identifica univocamente i dispositivi all'interno della rete OSPF, espresso in ddn. Se non è indicato viene selezionato automaticamente come router-id l'indirizzo IP maggiore tra le interfacce logiche configurate nel dispositivo. Se non è specificata alcuna interfaccia logica, allora viene preso l'indirizzo IP maggiore tra le interfacce attive.
- il numero dell'area OSPF, espressa in ddn. La backbone area nel caso di esempio sarà 0.0.0.0, anche esprimibile con area 0;
- i segmenti di rete connessi al dispositivo di riferimento; esprimibili con l'indirizzo di rete + wildcard mask, oppure, per essere più specifici ed evitare

errori di routing holes, con l'indirizzo dell'interfaccia del router + wildcard mask settata a 0.

Per fare chiarezza su questi concetti vengono mostrati i comandi assegnati ai tre router presenti nella rete.

AR4

```
The device is running!

<R4>sys
Enter system view, return user view with Ctrl+Z.
[R4]ospf 1 rou
[R4]ospf 1 router-id 4.4.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]dis this
[V200R003C00]
#
 area 0.0.0.0
  network 192.168.10.254 0.0.0.0
  network 192.168.20.254 0.0.0.0
  network 192.168.30.254 0.0.0.0
  network 192.168.45.1 0.0.0.0
#
return
[R4-ospf-1-area-0.0.0.0]
```

Figura 17: comandi OSPF in R4 all'interno dell'area 0

Nella *figura 17* si può notare come in R4 è stato configurato il numero del processo OSPF, ovvero 1, e il router-id 4.4.4.4 che ci ricorda appunto l'identificatore associato al router. Dopodiché all'interno dell'area 0, sono stati esplicitati tutti gli indirizzi delle interfacce che sono direttamente collegate a questo router, rispettivamente:
int vlanif 10 (192.168.10.254),
int vlanif 20 (192.168.20.254),
int vlanif 30 (192.168.30.254) e
int g0/0/0 collegata a R5 (192.168.45.1).

AR4 AR5 AR6

```
The device is running!

<R5>sys
Enter system view, return user view with Ctrl+Z.
[R5]ospf 1 rou
[R5]ospf 1 router-id 5.5.5.5
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]dis this
[V200R003C00]
#
 area 0.0.0.0
  network 192.168.45.2 0.0.0.0
  network 192.168.56.1 0.0.0.0
#
return
```

Figura 18: comandi OSPF in R5 all'interno dell'area 0

In R5 è stata seguita la stessa linea guida, mostrata per R4, e nell'area 0 vi sono rispettivamente gli indirizzi di: int g0/0/0 collegata a R4 (192.168.45.2) e int g0/0/1 collegata a R6 (192.168.56.1).

AR4 AR5 AR6

```
The device is running!

<R6>sys
Enter system view, return user view with Ctrl+Z.
[R6]ospf 1 ro
[R6]ospf 1 router-id 6.6.6.6
[R6-ospf-1]area 0
[R6-ospf-1-area-0.0.0.0]dis this
[V200R003C00]
#
 area 0.0.0.0
  network 192.168.50.254 0.0.0.0
  network 192.168.56.2 0.0.0.0
#
return
```

Figura 19: comandi OSPF in R6 all'interno dell'area 0

Come per R4 e R5, in R6 è stato configurato un router-id che ricordi il numero del router cioè 6.6.6.6, e nell'area 0 vi sono rispettivamente gli indirizzi IP di: int g0/0/0 collegata a R5 (192.168.56.2) e int g0/0/2 connessa alla LAN 5 (192.168.50.254).

All'interno della rete è importante specificare un altro parametro per minimizzare lo scambio di traffico dei dispositivi che fanno parte dell'area OSPF; infatti, in tutti i collegamenti p2p, tra i router, è stato esplicitato un comando a livello di interfaccia che rende più efficiente l'adiacenza formata dai dispositivi che utilizzano il protocollo. Il comando è ospf network-type p2p e in particolare è stato assegnato nelle interfacce GE 0/0/0 di R4 e di R6 e in entrambe le interfacce di R5. Questo perché in una rete p2p le informazioni da scambiare sono minori rispetto, ad esempio, a una rete broadcast. OSPF riconosce una rete p2p solo se si utilizza un cavo seriale, e se invece si impiega un cavo Ethernet per collegare due soli dispositivi, si deve esplicitare il fatto che quel segmento rappresenti non una rete broadcast ma una rete p2p.

Grazie a questi comandi i router entrano in possesso delle informazioni degli altri dispositivi e iniziano a conoscere tutti gli spazi di indirizzi associati ai diversi segmenti di rete, popolando così, la propria tabella di routing. Solo in questo modo gli host di due reti separate sono in grado di pingarsi; nella *figura 20* è riportato un esempio di un comando tracert eseguito da PC2 verso PC5 dove possiamo osservare i vari hop che seguono i messaggi, così da evidenziare il percorso dei pacchetti.

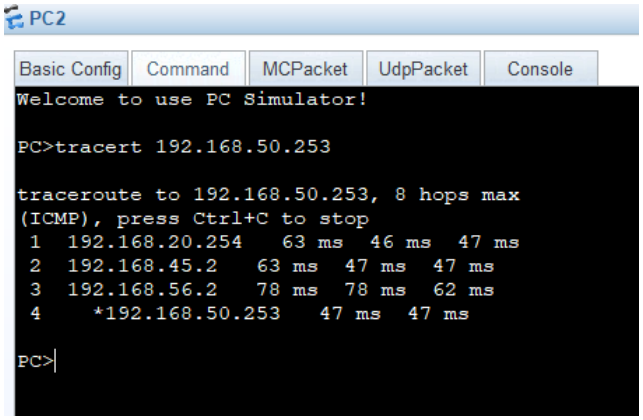


Figura 20: output comando tracert per mostrare gli hop di un pacchetto dalla VLAN 20 alla LAN 5

È possibile notare le interfacce dove passano i dati, tra cui:

1. 192.168.20.254 – R4 (VLAN 20)
2. 192.168.45.2 – R5
3. 192.168.56.2 – R6
4. 192.168.50.253 – PC5 (dest)

5. Livello Applicazione

In cima allo stack protocollare, sia ISO/OSI che TCP/IP, si trova il livello di applicazione che si interfaccia ai processi e fornisce servizi alle applicazioni (programmi) che vengono elaborate nei calcolatori dagli utenti della rete. Nella topologia di esempio, è stato implementato un servizio fondamentale e, che, viene utilizzato nella maggior parte delle reti enterprise, cioè il Dynamic Host Configuration Protocol (DHCP). In questo capitolo viene mostrata la sua implementazione nella rete, come esempio di un servizio fornito a livello applicativo.

5.1. DHCP (Dynamic Host Control Protocol)

Un servizio fondamentale per le reti informatiche è quello di assegnazione dinamica dell'indirizzo IP ai propri calcolatori/smartphones. Come è stato detto in precedenza, per poter comunicare con altri calcolatori, un dispositivo necessita di un indirizzo IP; quest'ultimo può essere configurato attraverso due diverse modalità, ovvero:

- indirizzamento statico che consiste nel configurare manualmente ad ogni nodo di rete il proprio indirizzo IP;
- indirizzamento dinamico che permette ai calcolatori di poter ottenere un indirizzo IP automaticamente, senza che l'amministratore lo debba assegnare manualmente. Esistono diversi protocolli che implementano l'indirizzamento dinamico come BOOTP, RARP e DHCP. Nella topologia è stato utilizzato proprio quest'ultimo che, quindi sarà l'unico ad essere analizzato.

DHCP utilizza un'architettura client/server in cui i client, ovvero i terminali, richiedono un indirizzo IP al server. I server possono essere rappresentati sia direttamente da router, come nel caso della topologia del progetto, oppure da server appositi a fornire questo, e anche molti altri servizi. Proprio il server provvederà a processare la richiesta dei client e a fornire una risposta. Essa consiste nell'assegnazione di un indirizzo IP e di altre, eventuali, informazioni di configurazione, quali possono essere ad esempio l'indirizzo del server DNS.

All'interno della topologia di esempio, DHCP viene utilizzato per non dover manualmente configurare gli indirizzi a tutti i calcolatori. Infatti, all'avvio dei dispositivi a tutti i nodi viene assegnato automaticamente un indirizzo appartenente allo spazio

degli indirizzi assegnato al segmento a cui essi fanno parte. All'interno della rete il DHCP server è stato abilitato in R6. Per la LAN 5 non ci sono problemi per raggiungere il server, in quanto è direttamente collegato ad essa; perciò, con una richiesta broadcast un host può facilmente avvisare il router della sua presenza e può, quindi, richiedere un indirizzo. Il discorso è leggermente più complicato per le VLAN 10, 20 e 30 in quanto i PC non riescono a raggiungere il server (R6) direttamente con un messaggio broadcast perché appartengono a domini di broadcast differenti. Entra in gioco una funzionalità molto utile che mette a disposizione DHCP, ovvero quella del DHCP relay. Il DHCP relay è un dispositivo che fa da tramite tra il server e i client che non appartengono allo stesso dominio di broadcast. Il DHCP relay conosce l'indirizzo del DHCP server e quindi può inoltrare le richieste dei client, a lui collegati, verso il server così che lui possa processarle. A questo punto il server elabora le risposte e le inoltra di nuovo al relay che si occupa di consegnare i messaggi a tutti i client che abbiano fatto precedentemente richiesta. Il relay è interpretato nella topologia da R4 e grazie ad esso anche gli host appartenti alle VLAN possono ottenere un indirizzo IP automaticamente.

I comandi per implementare questo protocollo sono riportati, passo per passo, nelle figure successive:

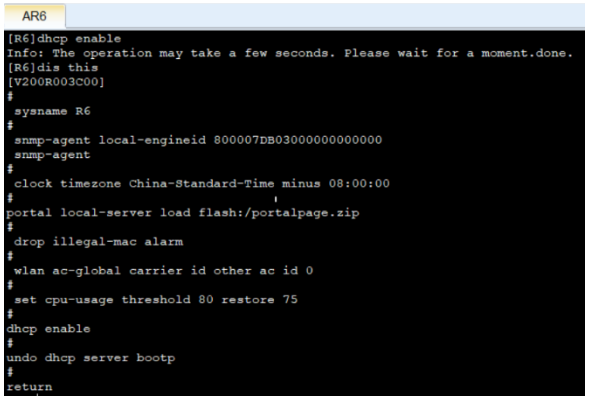


Figura 21: output comando display per evidenziare che è stato abilitato il protocollo DHCP in R6 (server DHCP)

1. innanzitutto, si deve abilitare DHCP nel router attraverso il comando dhcp enable in system-view. Nella *figura 21* si può notare che in R6 è stato abilitato DHCP dalla penultima riga dell'output del comando display this

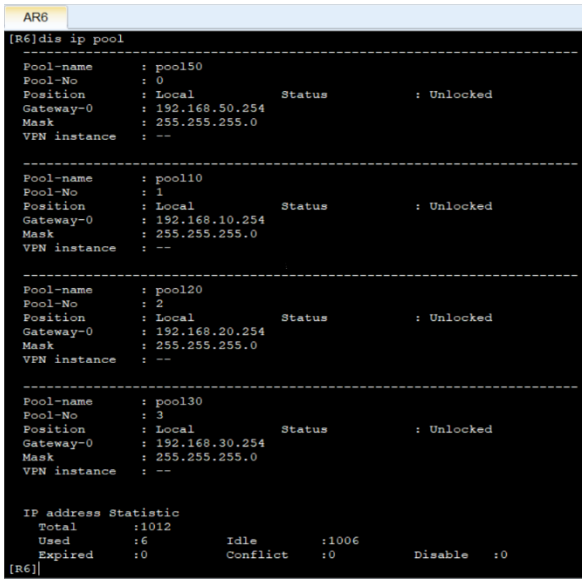


Figura 22: definizione degli ip pool da cui verranno selezionati gli indirizzi IP da assegnare ai DHCP client

2. si devono definire dei "pool" di indirizzi, ovvero gli spazi di indirizzamento, da cui poi il server prenderà gli indirizzi IP da assegnare ai client che ne fanno richiesta. Questi pool vengono creati con il comando ip pool <name>; ad ognuno di essi viene assegnato uno spazio di indirizzi tramite il comando network <indirizzo_di_rete> mask <subnet_mask> e il proprio gateway con gateway-list <indirizzo_DG>

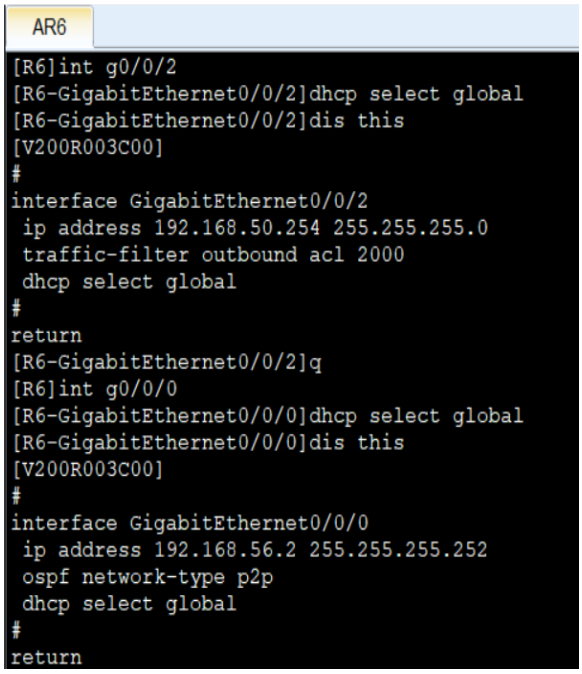


Figura 23: dhcp select global, comando per l'attivazione del server DHCP sull'interfaccia di interesse

3. nell'interfaccia interessata, cioè quella collegata al segmento di rete contenente i client, deve essere abilitato il servizio. Ciò può essere fatto in 2 diversi modi:

- con il comando dhcp select global dove il server assegna un indirizzo al client tra quelli disponibili nel pool definito per quel segmento specifico (utilizzata nell'esempio);
- con il comando dhcp select interface, tramite il quale non è necessario definire un pool perché ai client verrà assegnato un indirizzo dello stesso spazio di quello configurato nell'interfaccia.

Nelle seguenti catture (dalla *figura 24* alla 27) Wireshark vengono mostrati in dettaglio i 4 messaggi che si scambiano client e server per accordarsi e per assegnare, al primo di questi, un indirizzo.

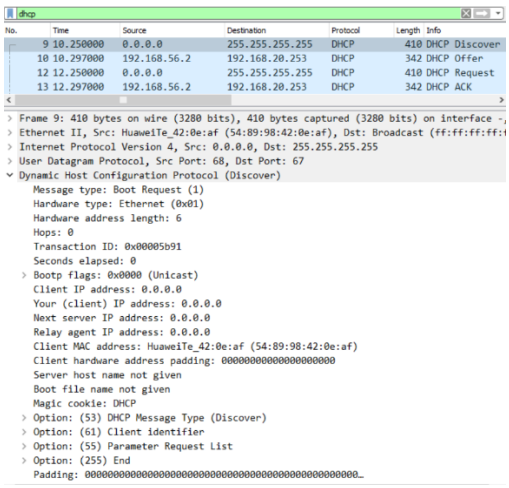


Figura 24: cattura WireShark sul DHCP discover

Il primo è il DHCP discover ovvero il primo messaggio del client, inoltrato in broadcast, per cercare un DHCP server. Si può notare che il discover ha come mittente IP 0.0.0.0 proprio perché il client non ha alcun indirizzo IP configurato.

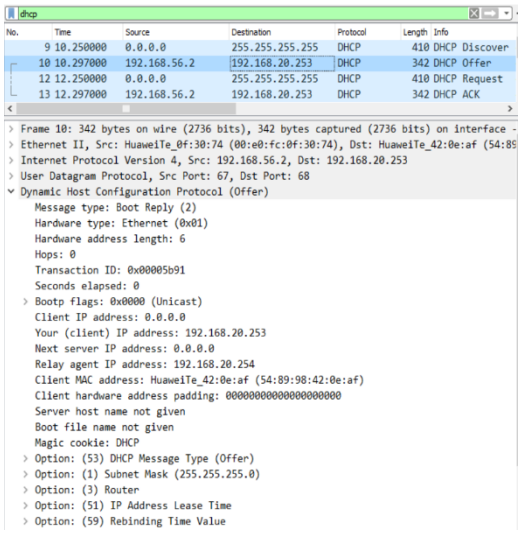


Figura 25: cattura WireShark sul DHCP offer

Quando il DHCP discover raggiunge il server, quest'ultimo risponde con un DHCP offer. Questo messaggio è particolare perché è unicast, e ha destination IP l'indirizzo che il server sta proponendo al client ma che ancora non è attivo. Il client è in grado di processare questo messaggio per com'è definito DHCP; infatti, RFC1122 definisce il processo per il quale un host con indirizzo 0.0.0.0 accetti e processi tutti i pacchetti IP che gli arrivino.

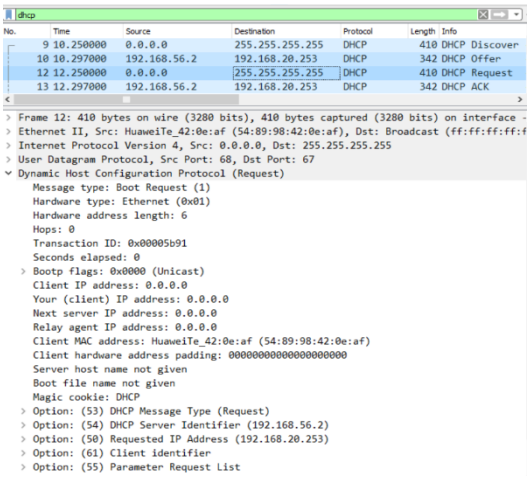


Figura 26: cattura WireShark sul DHCP request

Il client a questo punto richiede ufficialmente i parametri offerti dal server con un messaggio DHCP request, inoltrato anch'esso in broadcast, come il discover.

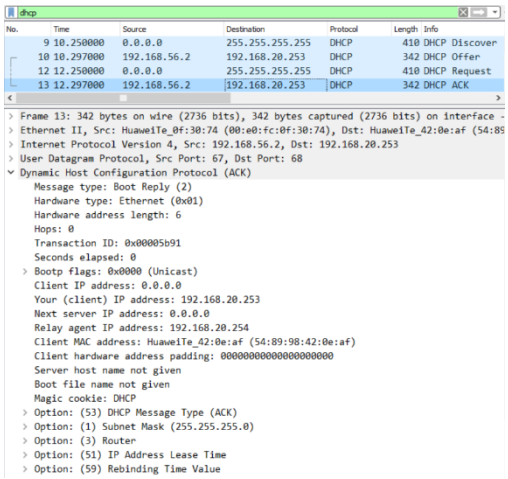


Figura 27: cattura WireShark sul DHCP ack

L'ultimo passo è l'effettiva approvazione da parte del server sui parametri richiesti dal client con la DHCP request. La conferma avviene con il messaggio DHCP ack. Una volta che il client processa questo messaggio l'indirizzo viene ufficialmente assegnato al calcolatore e quest'ultimo può utilizzarlo per ricevere e inoltrare traffico. .

6. Sicurezza

Questo capitolo tratta delle tematiche, legate alla sicurezza informatica, che sono state adottate all'interno della topologia. Infatti, sono state applicate tecniche per il filtraggio del traffico che vanno a rendere più sicure alcune sotto-aree vulnerabili. Questi aspetti sono analizzati nella sezione specifica delle Access Control List (ACL).

Inoltre, in questo capitolo verrà mostrata un'ulteriore implementazione per la topologia che permette di proteggere informazioni riguardanti l'ip routing delle reti private aziendali verso l'esterno. Si supponga, ad esempio, di sottostare alle seguenti ipotesi:

- R5 (router centrale) sia un dispositivo che non appartiene direttamente all'infrastruttura di riferimento ma ad esempio all'Internet Service Provider (ISP);
- R4 e R6 siano due router che collegano due filiali fisicamente distaccate della stessa azienda. Una filiale è rappresentata dalle VLAN 10, 20 e 30 mentre l'altra dalla LAN 5;
- Si voglia utilizzare un protocollo di routing dinamico per far comunicare le due filiali ma senza cedere informazioni direttamente all'ISP riguardanti la gestione delle reti private.

Sotto queste ipotesi, e per soddisfare l'ultima richiesta, si è pensato di implementare un tunnel GRE che mettesse direttamente in collegamento R4 e R6 e che permetta di veicolare pacchetti OSPF senza coinvolgere "logicamente" R5 e rilevare ad esso la struttura interna aziendale. All'interno del paragrafo 6.2 (Tunneling GRE) verranno mostrate le differenze tra quest'ultima implementazione e la configurazione analizzata fino ad ora, e la sua utilità per rendere più sicura ed efficiente la rete.

L'unico problema di questa nuova soluzione riguarda la riservatezza e l'autenticità dei dati. Questo perché GRE è un protocollo molto generico, in quanto riesce a veicolare più di 20 protocolli all'interno dei suoi tunnel, ma a suo discapito le informazioni non sono né autenticate né criptate in alcun modo. Per risolvere questo problema e rendere la rete inaccessibile dall'esterno, mantenendo comunque la possibilità di poter veicolare tecnologie diverse, si deve integrare un nuovo protocollo: IPSec. Come dice la parola, esso è un protocollo che permette di rendere sicure comunicazioni IP, infatti

IPSec è stato creato in modo tale da poter veicolare solo pacchetti IP. In definitiva quindi si è deciso di realizzare un tunnel IPSec che veicoli GRE in modo tale da poter trasportare “tutto”, grazie alla generalità di GRE, in modo sicuro, con le funzionalità di IPSec. Tutto ciò verrà analizzato in maniera più dettagliata nel paragrafo 6.3 (GRE over IPSec).

6.1. ACL (Access Control List)

Fino ad ora non sono stati presi in considerazione aspetti legati alla sicurezza della rete; un metodo sicuramente molto efficace per rendere segmenti di rete più sicuri è quello di filtrare il traffico che circola su di essi. Esistono dei meccanismi appositi che permettono di implementare un processo di autorizzazione verso una data risorsa, in questo caso un segmento di rete; questi meccanismi prendono il nome di ACL (Access Control List). Uno dei molteplici utilizzi delle ACL è proprio quello di filtrare i messaggi in entrata o in uscita da una determinata interfaccia di un dispositivo l3, specialmente nei router. Sostanzialmente esse sono una lista di regole che, in base a dei parametri specifici settati dall’amministratore di rete, catturano pacchetti e decidono quale sia l’azione da attuare su questi ultimi. Sono molto importanti nelle reti enterprise, soprattutto per la cosiddetta DMZ (DeMilitarized Zone) ovvero una sottorete (logica o fisica) che contiene ed espone server verso reti esterne. Nella DMZ è molto importante poter scegliere quale sia il traffico da permettere e quale sia quello da filtrare perché ritenuto pericoloso. Un approccio naive per le DMZ potrebbe essere proprio quello di utilizzare le ACL per negare l’accesso a queste aree da parte di sottoreti non conosciute o ritenute poco sicure.

Nei dispositivi Huawei, esistono 3 diverse tipologie di ACL, e sono essenzialmente caratterizzate ognuna dai parametri che esse adottano per poter intercettare i pacchetti. Esse si distinguono tra loro da un range di valori associati a ogni tipologia:

- **basic ACL** (range da 2000 a 2999), sono le più semplici da implementare ma meno precise perché permettono di selezionare i pacchetti solo in base alla source IP;
- **advanced ACL** (range da 3000 a 3999), sono più specifiche in quanto consentono di filtrare il traffico in base a source e destination IP, protocolli L3 e L4 utilizzati, source e destination port L4 e anche rispetto a flag presenti nei segmenti (utilizzati ad esempio nell’header TCP).

- **Layer 2 ACL** (range da 4000 a 4999) che basano il match su informazioni di livello 2, come ad esempio source e destination MAC e protocolli L2 utilizzati.

Nella topologia di riferimento, l'utilizzo principale delle ACL è quello di filtrare il traffico verso un determinato segmento di rete. Infatti, la LAN 5 contiene calcolatori e informazioni ritenute importanti e può essere acceduta solo da host di una determinata network. Era importante permettere l'accesso alla LAN 5 solamente agli host che appartenessero alla VLAN 20 e negarlo invece alla VLAN 10 e 30. Essendo questa la richiesta, nella rete è stata adottata una basic ACL (si nota dalla *figura 28* dal numero associato alla ACL cioè 2000) che seleziona il traffico in base all'indirizzo IP sorgente. Prima di mostrare l'effettiva implementazione dell'ACL, deve essere chiarita la scelta del dispositivo dove quest'ultima è stata applicata. Avendo deciso di impiegare una basic ACL, che risulta essere molto generica, essa rischia di filtrare più traffico del dovuto specialmente se applicata vicino alla sorgente perché andrà a bloccare tutti i pacchetti delle VLAN 10 e 30, anche se non destinati verso la LAN 5. Perciò è molto più funzionale, per il corretto inoltro dei pacchetti, attivare l'ACL nel router più vicino alla destinazione, ovvero R6. Così facendo infatti si andranno a bloccare solo i pacchetti aventi come destinazione un nodo appartenente alla LAN 5, segmento di rete che si intende proteggere. Un altro aspetto molto importante da tenere in considerazione quando si applica una ACL è il "verso" con cui essa andrà a catturare i pacchetti. Ci sono due possibilità ovvero in ingresso (inbound) e in uscita (outbound). In questo caso l'ACL deve essere applicata in uscita rispetto all'interfaccia G0/0/2 di R6 perché deve catturare solo i pacchetti che sono diretti verso il segmento collegato al link, cioè la LAN 5. Se infatti fosse stato specificato inbound essa avrebbe perso completamente il suo senso perché avrebbe selezionato solo i pacchetti ricevuti dalla GE 0/0/2 di R6, con source IP 192.168.10.0/24 e 192.168.30.0/24; cosa ovviamente impossibile per come sono stati assegnati gli indirizzi ai segmenti di rete. Infatti, l'unica interfaccia di R6 che può ricevere pacchetti con indirizzo IP mittente appartenenti alle tre Vlan è la GigabitEthernet 0/0/0.

AR6

```
[R6]acl 2000
[R6-acl-basic-2000]rule 5 deny source 192.168.10.0 0.0.0.255
[R6-acl-basic-2000]rule 10 deny source 192.168.30.0 0.0.0.255
[R6-acl-basic-2000]rule 20 permit source 192.168.20.0 0.0.0.255
[R6-acl-basic-2000]rule 25 deny source any
[R6-acl-basic-2000]dis this
[V200R003C00]
#
acl number 2000
 rule 5 deny source 192.168.10.0 0.0.0.255
 rule 10 deny source 192.168.30.0 0.0.0.255
 rule 20 permit source 192.168.20.0 0.0.0.255
 rule 25 deny
#
return
[R6-acl-basic-2000]q
[R6]int g0/0/2
[R6-GigabitEthernet0/0/2]dis this
[V200R003C00]
#
interface GigabitEthernet0/0/2
 ip address 192.168.50.254 255.255.255.0
 traffic-filter outbound acl 2000
 dhcp select global
#
return
```

Figura 28: visualizzazione comandi per la creazione dell'acl 2000

Come si può notare dalla *figura 28* le ACL si presentano proprio come una lista di regole che l'amministratore specifica e ognuna di esse è identificata da un numero, che ne caratterizza anche la priorità. Infatti, le regole definite nelle ACL sono prese in considerazione e verificate in ordine crescente in base a questo numero identificativo. Tramite le regole 5 e 10 quindi si nega, attraverso la parola chiave deny, l'accesso alle VLAN 10 e 30 che, si ricorda, hanno uno spazio di indirizzi associato rispettivamente di 192.168.10.0/24 e 192.168.30.0/24 (si utilizza la wildcard mask nelle ACL, come in OSPF). Con la terza regola (rule 20) viene permesso, con la parola chiave permit, l'accesso alla VLAN 20 a cui è assegnato lo spazio 192.168.20.0/24. Infine, attraverso l'ultima regola (rule 25) viene negato l'accesso a qualsiasi altro pacchetto contente una source IP diverso da quelli precedentemente specificati con la parola chiave source any. Una volta definita la ACL deve essere attivata sulla corretta interfaccia in modo tale da rispettare il filtro sulla source IP. Nella parte finale della figura si può osservare come effettivamente l'acl 2000 sia stata applicata all'interno dell'interfaccia GigabitEthernet 0/0/2 di R6 attraverso il comando traffic-filter outbound acl 2000. Tramite la parola chiave outbound è stato specificato che l'ACL deve filtrare i pacchetti in uscita verso questa interfaccia, data la motivazione della pagina precedente

Con le successive figure (*figura 29, 30,* 31) viene mostrata l'effettiva utilità dell'ACL implementata:

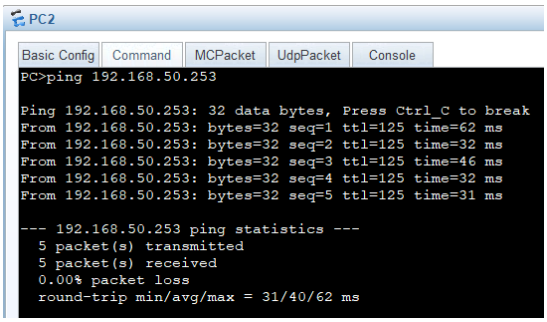


Figura 29: ping con esito positivo da PC2 (VLAN 20) a PC5 (LAN 5)

in questo caso si effettua un ping per testare l'accesso verso l'host 192.168.50.253, cioè PC5, da PC2 (192.168.20.253) che fa parte della VLAN 20. Come è evidenziato dal ping, i due calcolatori riescono a comunicare perfettamente perché l'ACL lo permette tramite la rule 20.

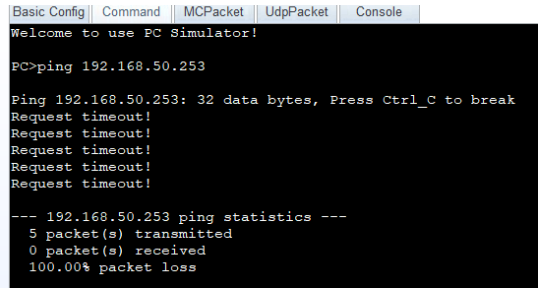


Figura 30: ping con esito negativo da PC1 (VLAN 10) a PC5 (LAN 5)

A differenza di prima, in questo esempio viene mostrato un tentativo fallimentare di ping da PC1, della VLAN 10. Questo avviene perché nell'ACL la rule 5 specifica proprio come alla VLAN 10 sia negato l'accesso alla LAN 5 e di conseguenza a PC5.

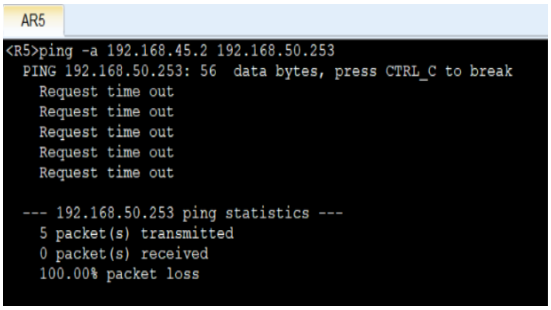


Figura 31: ping con esito negativo da R5 (192.168.45.2) a PC5 (LAN 5)

In quest'ultima figura viene mostrata l'applicazione della regola 25 che nega l'accesso alla LAN 5 a qualsiasi sorgente IP. Infatti, viene effettuato un ping verso PC5 dal router 5, in particolare dalla interfaccia GigabitEthernet 0/0/0 definita dallo switch -a 192.168.45.2. Si può notare come i pacchetti non riescano a raggiungere la destinazione proprio perché nell'acl 2000 è stata utilizzata la parola chiave deny per scartare tutti i pacchetti che non matchino altre regole.

6.2. Tunneling GRE

In questa sezione viene mostrata una diversa implementazione della rete rispetto a quella vista fino ad ora. Ciò è fatto per analizzare una situazione, piuttosto comune, dove una impresa vuole nascondere la struttura interna della rete grazie ad un tunnel logico che veicoli il protocollo di routing OSPF. Così facendo è possibile tenere sempre aggiornate le tabelle di routing dei due router interni, mascherandone le informazioni a R5 che li collega e che può rappresentare, in una situazione reale, l'ISP.

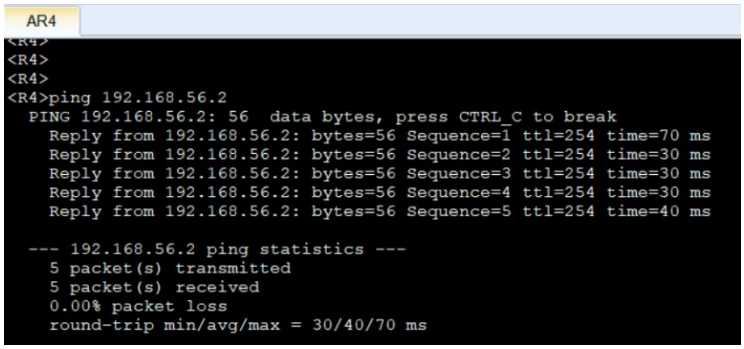
Il tunnel virtuale che implementa questo processo è stato gestito attraverso il protocollo GRE (Generic Routing Encapsulation). Esso è principalmente utilizzato per il trasporto di pacchetti che utilizzano protocolli diversi rispetto a IP; l'utilizzo più comune è proprio quello dell'incapsulamento di protocolli di routing per permettere ad aree distaccate di rimanere costantemente aggiornate sulla struttura della rete. Il principio che sta alla base di questo protocollo è, come dice il nome, l'incapsulamento di pacchetti in altri per nascondere il contenuto di quelli più interni. GRE genera un collegamento logico p2p tra due dispositivi che li collega direttamente nonostante essi siano distanti tra loro. I 2 peer del tunnel per simulare questo link diretto, inseriscono il pacchetto, che deve essere inoltrato all'interno del tunnel, in un altro pacchetto IP e dopodiché aggiungono un riferimento al protocollo GRE, così che l'altro peer sa che dovrà decapsulare i dati secondo lo standard del protocollo.

Per poter costituire un tunnel GRE è necessario che i due peer siano raggiungibili. Qui sorge la prima differenza sostanziale rispetto quello visto sinora, infatti, i 3 router della topologia non formano più un'adiacenza OSPF poiché R5 non appartiene più a tale processo. Per questo sono state configurate delle rotte statiche che permettano la comunicazione dei tre dispositivi *(10)*. Di seguito sono mostrati i comandi assegnati per ognuno dei router, dove si è specificato rispettivamente:

1. Indirizzo della rete di destinazione a cui si riferisce la rotta;
2. Subnet mask della rete (esprimibile sia in ddn che come numero di bit a 1);
3. Indirizzo del next-hop, ovvero il dispositivo a cui sono mandati i pacchetti che hanno come destination IP un indirizzo che combaci con la rotta.

R4: ip route-static 192.168.56.0 30 192.168.45.2

R6: ip route-static 192.168.45.0 30 192.168.56.1



In questo modo, come si può osservare dalla *figura 32*, R4 con indirizzo 192.168.45.1 riesce a pingare R6, che ricordiamo avere indirizzo 192.168.56.2.

Figura 32: conferma della raggiungibilità dei due peer del tunnel GRE attraverso un ping

A questo punto è possibile realizzare a tutti gli effetti il tunnel GRE all'interno di R4 e R6. Per fare ciò si deve creare l'interfaccia logica tunnel attraverso il comando: interface Tunnel <num_interfaccia>. Una volta creata l'interfaccia si deve assegnare un indirizzo IP che identifica l'interfaccia del dispositivo in questo nuovo link. Per il tunnel GRE è stato assegnato quindi un nuovo spazio di indirizzi ovvero 192.168.46.0/30; ha una subnet /30 perché è un collegamento p2p che collega due soli router e quindi non necessita di più di due host indirizzabili. Dopo di che con il comando tunnel-protocol gre si attiva effettivamente GRE come protocollo che gestisce il tunnel. A questo punto rimangono da specificare gli indirizzi delle interfacce fisiche che simulano il tunnel GRE; infatti, i pacchetti veicolati con GRE non passeranno fisicamente sul tunnel ma saranno incapsulati all'interno di un altro pacchetto che sarà inoltrato su collegamenti reali che connettono i due peer. Nell'esempio infatti il pacchetto originale, il quale risulterà passare nel tunnel, sarà contenuto all'interno di un pacchetto esterno; quest'ultimo passerà per il router 5 seguendo il classico percorso formato dalle interfacce GigabitEthernet. Quindi gli indirizzi "fisici" che sono stati specificati come source e destination per R4 e R6 sono stati configurati rispettivamente tramite i seguenti comandi:

- R4: source 192.168.45.1
  destination 192.168.56.2
- R6: source 192.168.56.2
  destination 192.168.45.1

A questo punto a livello di interfaccia tunnel sono state date due ulteriori indicazioni specifiche. La prima riguarda proprio l'incapsulamento che effettua GRE; infatti come già specificato il pacchetto originale è incapsulato all'interno di un altro pacchetto IP. Per questo la dimensione massima MTU (Maximun Transmission Unit) del pacchetto

interno non può essere 1500 Byte come per tutti gli altri pacchetti a livello 3. GRE, infatti, genera un overhead di 24 Byte per poter incapsulare correttamente i dati e quindi si deve indicare un MTU massimo di 1500B – 24B =1476B. Questo è stato fatto attraverso il semplice comando mtu 1476. La seconda e ultima opzione si riferisce al processo OSPF. Come analizzato nel capitolo di OSPF, un collegamento p2p va specificato in modo tale da minimizzare il traffico di cui ha bisogno il protocollo per tenere aggiornati i dispositivi. Quindi si assegna il comando ospf network-type p2p.

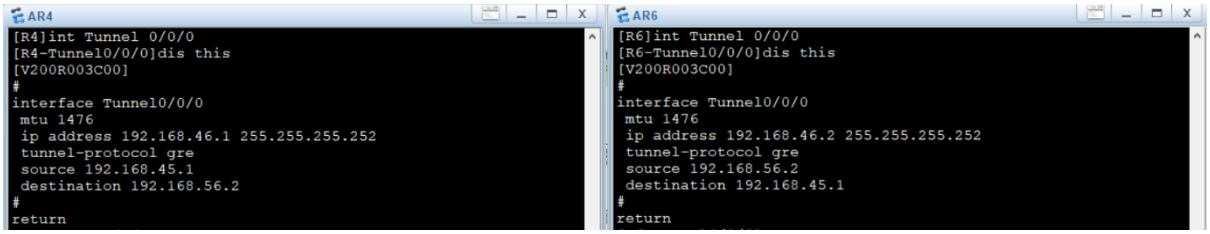


Figura 33: comandi assegnati all'interno dell'interfaccia Tunnel 0/0/0 di R4 e di R6

Una volta creata l'interfaccia logica che simula il tunnel GRE, si può verificare il corretto funzionamento di quest'ultimo andando a mostrare le informazioni riguardanti l'interfaccia Tunnel. Ciò è fatto attraverso il comando display interface Tunnel <num_int> come mostrato dalla *figura 34*.

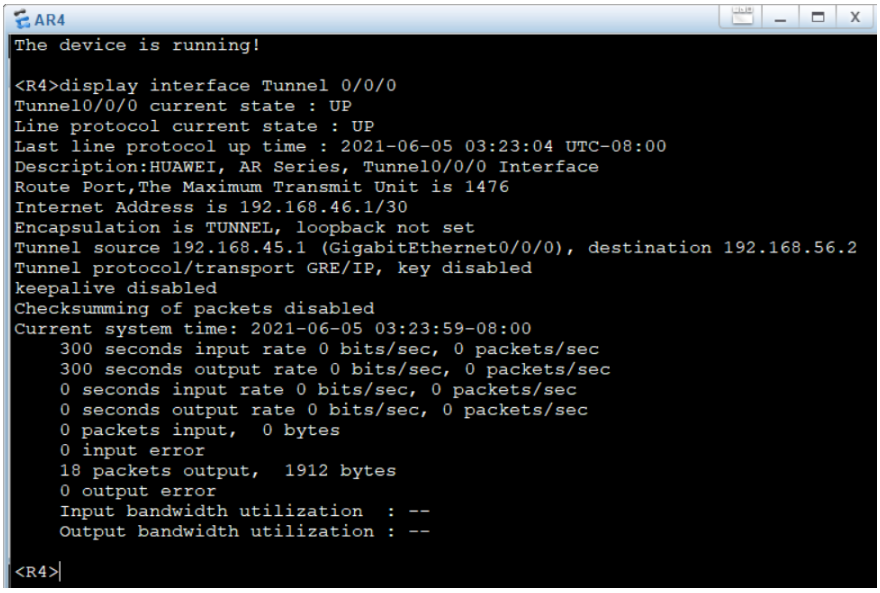


Figura 34: output comando display interface Tunnel 0/0/0

La prima riga dell'output, infatti, segnala che il tunnel è in stato UP, cioè è attivo e funzionante; dopo di che è segnalato che il mtu è settato a 1476 Byte. La voce Internet Address rappresenta l'indirizzo assegnato al peer del tunnel; nella figura l'output è in riferimento a R4 e quindi si ha come indirizzo 192.168.46.1/30. Invece sotto sono

specificati gli indirizzi delle interfacce fisiche dei due peer del tunnel che permettono di incapsulare i pacchetti GRE. Infatti, nel caso di R4 si ha Tunnel source 192.168.45.1, che rappresenta l'indirizzo dell'interfaccia G0/0/0 di R4, mentre Tunnel destination 192.168.56.2, che rappresenta l'indirizzo dell'interfaccia G0/0/0 di R6.

Una volta che si ha il tunnel perfettamente funzionante si deve aggiungere la nuova network, ovvero la 192.168.46.0/30, all'interno dell'area 0.0.0.0 di OSPF. Quindi, sostanzialmente, in R4 e in R6 non vengono più indicati i link che collegano R5 ma al loro posto deve essere precisato l'indirizzo dell'interfaccia Tunnel.

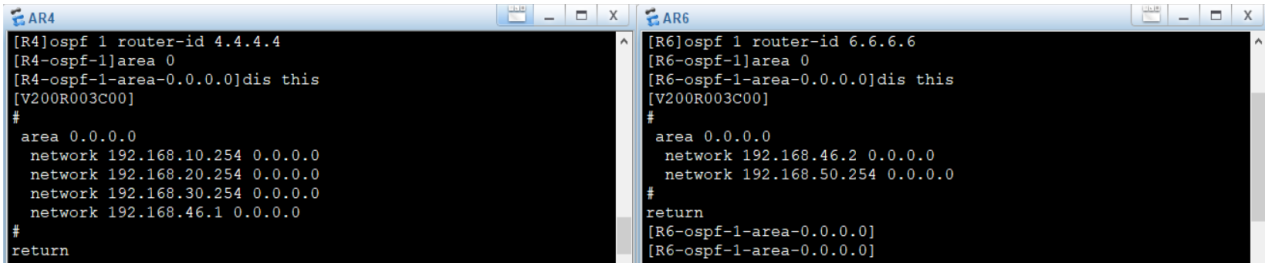


Figura 35: differenze comandi OSPF in R4 e in R6 all'interno dell'area 0

Come si può notare dalla *figura 35*, all'interno dell'area 0 sono rimaste specificate le network delle VLAN per R4 e della LAN 5 per R6. Il cambiamento è avvenuto con l'aggiunta della network riguardante il tunnel GRE ovvero rispettivamente:

- R4 – network 192.168.46.1 0.0.0.0
- R6 – network 192.168.46.2 0.0.0.0

Grazie a queste due informazioni i pacchetti OSPF riescono a passare attraverso il tunnel GRE e a legare i due router attraverso una adjacency. Quindi, attraverso GRE i due dispositivi sono in grado di scambiarsi informazioni sulle proprie sottoreti direttamente collegate ad essi e fare in modo di popolare le tabelle di routing automaticamente grazie ad un protocollo dinamico. Così facendo infatti le due filiali sono perfettamente consapevoli di eventuali cambiamenti della rete e i due router sono in grado di adattarsi a qualsiasi convergenza. Tutto questo è possibile senza dover minimamente coinvolgere R5, che si ricorda rappresentare l'ISP, e quindi mantenendo sicura la struttura interna aziendale.

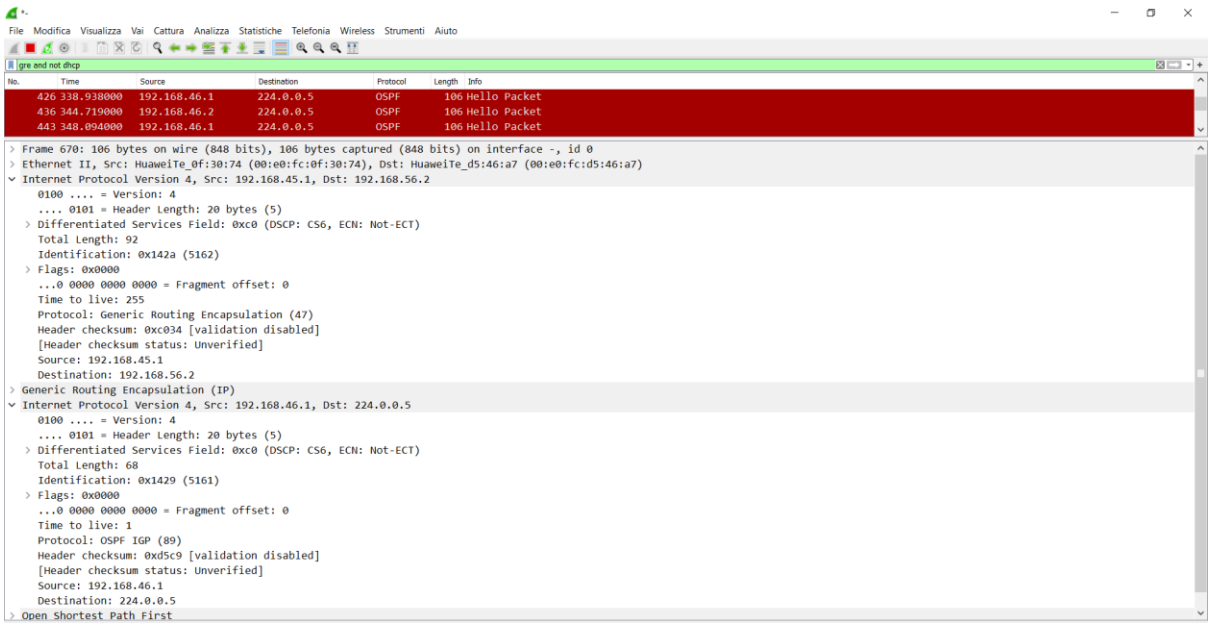


Figura 36: cattura WireShark su pacchetti OSPF incapsulati tramite GRE

Nella *figura 36*, si riporta una cattura WireShark che mostra come i pacchetti vengano effettivamente incapsulati uno dentro l'altro attraverso il protocollo GRE. Come si può notare nel pacchetto è stato aggiunto l'header GRE che notifica al peer di destinazione del tunnel che i dati sono stati incapsulati con il seguente protocollo. In questo esempio è evidenziato un Hello Packet di OSPF, il secondo pacchetto, ovvero quello più interno che viene gestito da GRE. L'hello packet è incapsulato all'interno di un normale pacchetto IP, il primo, che ha come source IP 192.168.45.1 (interfaccia G0/0/0 di R4) e destination IP 192.168.56.2 (interfaccia G0/0/0 di R6) che non sono altro che la source e la destination del tunnel GRE. In questo pacchetto si può notare come nel campo protocol non ci sia direttamente OSPF ma bensì GRE (protocol=47); infatti una volta che il pacchetto raggiunge R6 e il livello 3 nota questo valore di protocol, i dati vengono inviati al controller GRE che decapsula il pacchetto interno OSPF e dopodichè R6 lo elabora normalmente. Ultima cosa da evidenziare è che nel pacchetto interno OSPF il mittente IP è l'indirizzo di R4 assegnato all'interfaccia Tunnel ovvero 192.168.46.1.

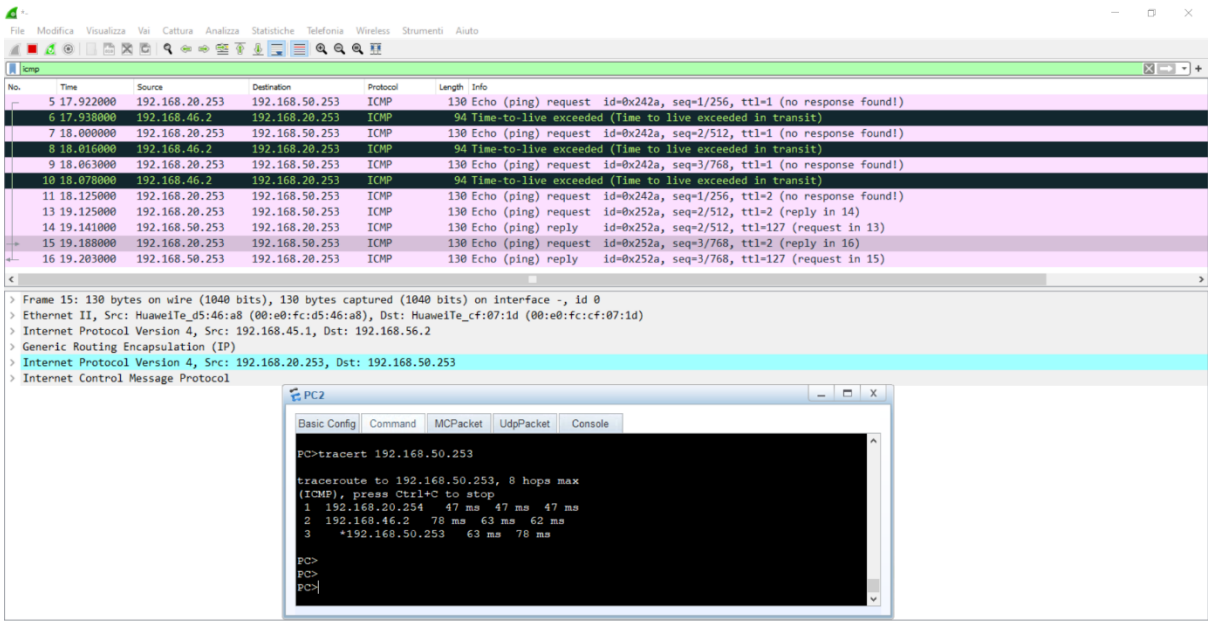


Figura 37: cattura WireShark su pacchetti ICMP (tracert) incapsulati tramite GRE

Un altro esempio per far comprendere meglio il funzionamento del tunnel GRE può essere quello mostrato nella *figura 37*. Nella parte bassa è mostrato l'output del comando tracert 192.168.50.253 (indirizzo di PC5 della LAN 5) eseguito da PC2, appartenente alla VLAN 20. Attraverso il tracert si può notare come il pacchetto interno, veicolato con GRE, passi virtualmente sul tunnel GRE; ciò si può notare dal fatto che il secondo hop è dato dall'indirizzo IP di R6 nell'interfaccia Tunnel 0/0/0 ovvero 192.168.46.2. Quindi fondamentalmente il percorso che segue il pacchetto ICMP è il seguente:

1. I dati vengono inviati al default gateway assegnato in PC2, cioè all'interfaccia vlanif 20 di R4 che ha indirizzo 192.168.20.254;
2. Da qui il pacchetto deve essere inoltrato sull'interfaccia tunnel 0/0/0, come definito dalla tabella di routing. A questo punto il pacchetto ICMP è incapsulato in un altro pacchetto IP il quale viene inviato fisicamente verso R5; R5 poi lo inoltrerà a R6 tramite l'interfaccia G0/0/0. Questo processo è evidenziato dalla cattura Wireshark, che permette di osservare entrambi i pacchetti, uno all'interno dell'altro. Il pacchetto interno però non vede il percorso fisico ed esso viene decapsulato direttamente quando raggiunge la destinazione ovvero R6. Infatti, nel tracert è evidenziato come il secondo hop sia proprio 192.168.46.2; nient'altro che l'indirizzo del peer di destinazione del tunnel.
3. A questo punto la ICMP request raggiunge la sua destinazione cioè 192.168.50.253 (PC5).

Nella *figura 38* sono evidenziati gli effettivi percorsi di entrambi i pacchetti. In rosso è mostrato il path del pacchetto interno, incapsulato con GRE, ovvero quello che risulta dal comando tracert precedente. In nero invece si può osservare il percorso del pacchetto esterno che permette di virtualizzare il tunnel GRE. Quest'ultimo è inoltrato sui link fisici che compongono la rete e che connettono i due peer del tunnel.

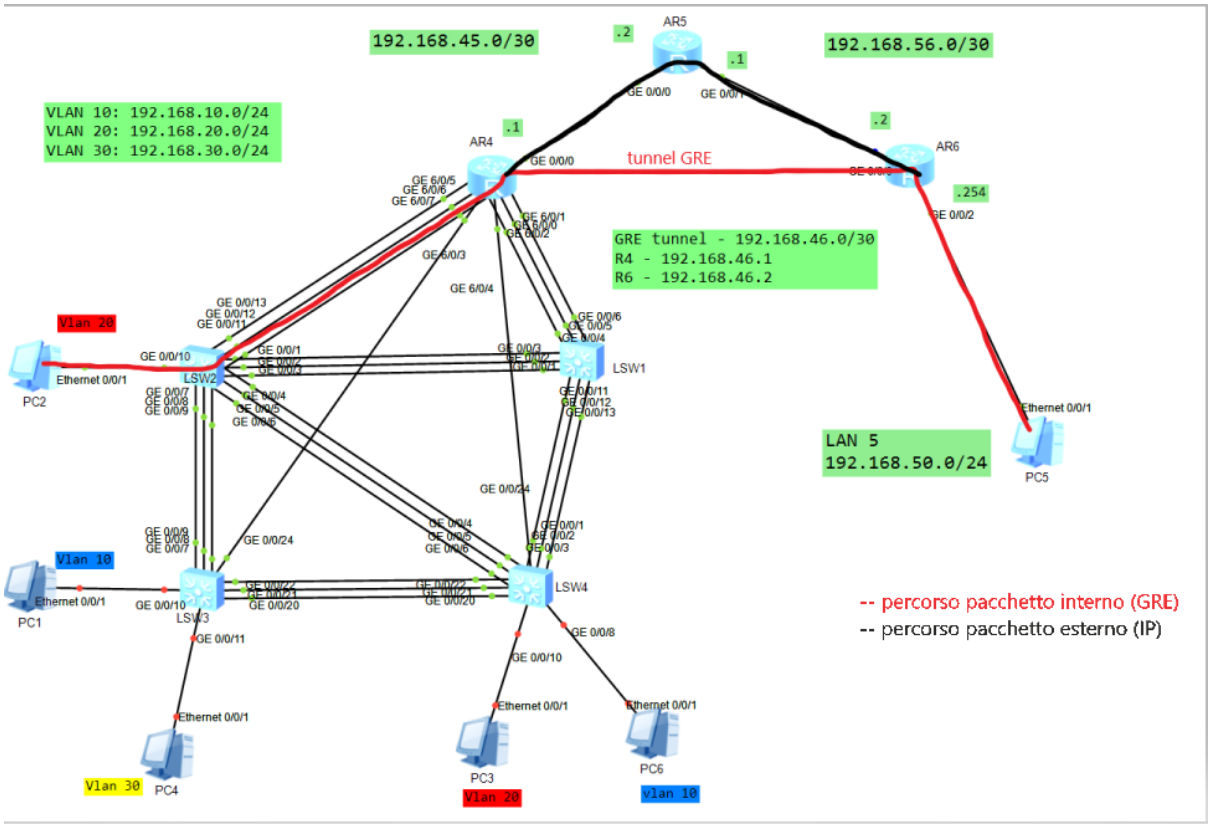


Figura 38: Visualizzazione percorso virtuale (in rosso) del pacchetto interno e percorso reale (in nero) del pacchetto esterno che incapsula l'altro

6.3. GRE over IPSec

Come già accennato all'inizio del seguente capitolo, il problema sta nel fatto che le informazioni che passano sul tunnel GRE non sono riservate in alcun modo. GRE non permette di specificare alcun algoritmo di crittografia o di autenticazione. Per risolvere questo problema, è stato applicato un nuovo protocollo: IPSec, il quale è utilizzato per creare tunnel sicuri. I tunnel IPSec garantiscono le seguenti proprietà di sicurezza:

- Confidenzialità: impedisce che il contenuto della comunicazione possa essere rilevato da terze parti, grazie a protocolli esterni che mettono a disposizione algoritmi di crittografia per i dati che circolano nel tunnel.
- Integrità: assicura che i dati che passano nel tunnel non siano stati modificati da enti esterni e mantiene l'ordine di essi inalterato.
- Autenticazione: prima di rendere operativo il tunnel vengono messi in atto meccanismi per verificare l'autenticità dei due peer che lo compongono.

IPSec quindi fornisce molti meccanismi, tra cui i più importanti sono sicuramente quelli di crittografia e di autenticazione. Per fare ciò si serve di due ulteriori protocolli: AH (Authentication Header) ed ESP (Encapsulating Security Payload). Essi permettono al tunnel IPSec di autenticare i dispositivi che lo utilizzano e di criptare i dati che passano al suo interno grazie ad algoritmi appositi. Il limite di IPSec però sta nel fatto che esso può veicolare solamente pacchetti IP al suo interno; nel caso di esempio, invece si vuole che il tunnel sia generico e soprattutto che esso riesca a veicolare pacchetti OSPF per mantenere in contatto le due componenti della rete. Per fare ciò è necessario creare un tunnel IPSec a livello più basso che garantisca la riservatezza della comunicazione e che possa trasportare solo IP. Dopodiché, grazie al fatto che GRE è a tutti gli effetti un protocollo basato su IP, è possibile trasportare tutto il traffico di quest'ultimo attraverso il tunnel IPSec, così da riuscire ad avere un collegamento sicuro e generico allo stesso tempo.

Ora c'è da capire come si crea un tunnel IPSec che riesca a trasmettere al suo interno non solo IP; infatti, dato che esso offre più servizi di sicurezza rispetto a GRE, anche la sua implementazione è leggermente più complessa. La sua configurazione si divide in diversi step, ovvero:

1. **Individuare il traffico da proteggere tramite un ACL**

Per prima cosa all'interno dei router R4 e R6 sono state configurate due ACL advanced che catturano tutto il traffico GRE. Sono state scelte le ACL advanced perché le ACL basic non permettono di filtrare il traffico in base al protocollo. La regola che è stata specificata all'interno dell'ACL è la seguente: rule 5 permit gre, come è evidenziata nella *figura 39, e che va a catturare tutti i pacchetti che sono incapsulati con GRE*.

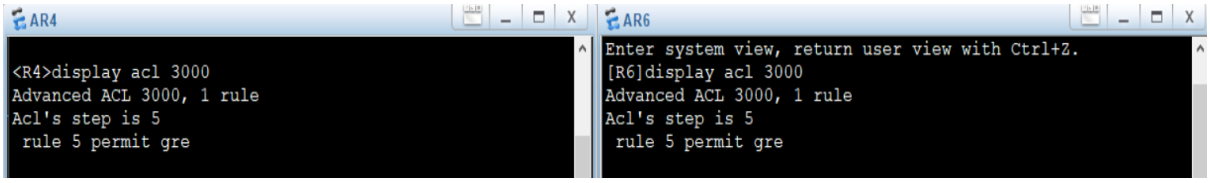


Figura 39: creazione acl 3000 per catturare tutto il traffico GRE per poter veicolare nel tunnel IPSec

2. **Definire una ipsec proposal**

La proposal è utilizzata dai due peer per definire le caratteristiche del tunnel. All'interno di essa sono stati specificati i seguenti parametri:

- Encapsulation mode – settata a Tunnel per specificare la modalità di incapsulamento del pacchetto originale all'interno di quello esterno.
- Transform – settata con il valore ah-esp dove si definiscono i protocolli utilizzati per l'autenticazione e la crittografia dei dati. Si può utilizzare anche un solo protocollo tra i due però per avere maggiore sicurezza sono stati adottati entrambi.
- ESP authentication – specifica l'algoritmo di autenticazione utilizzato dal protocollo ESP, cioè SHA2-256.
- ESP encryption – specifica l'algoritmo di crittografia utilizzato dal protocollo ESP, cioè AES-256.
- AH authentication - specifica l'algoritmo di autenticazione utilizzato dal protocollo AH, cioè SHA2-256.

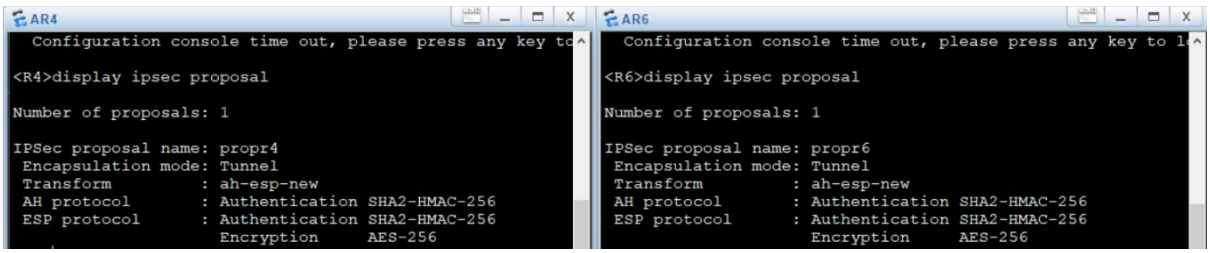


Figura 40: output comando display ipsec proposal per mostrare i parametri all'interno della proposal

3. Stabilire la policy

Una volta specificati i protocolli e gli algoritmi che saranno utilizzati all'interno del tunnel, si deve stabilire la policy che permette ai due peer di scambiarsi le chiavi e le informazioni necessarie per autenticarsi e assicurarsi che il tunnel sia affidabile. Ci sono due possibilità per definire una policy:

- manualmente, cioè andando a configurare tutte le password e i parametri necessari per il corretto funzionamento dei protocolli scelti;
- attraverso un protocollo esterno, ISAKMP (Internet Security Association and Key Management Protocol), che genera e scambia automaticamente le chiavi e le informazioni necessarie per avviare il tunnel.

Per costituire la policy della rete di esempio è stato utilizzato ISAKMP che è molto comodo in quanto semplifica parecchio la configurazione del tunnel. Per realizzare una policy attraverso ISAKMP, è necessario innanzitutto definire in ognuno dei due router il peer che andrà a comporre il tunnel. Per prima cosa, quindi, si crea il peer con il comando ike peer <nome_peer> v2 dove v2 rappresenta la versione IKE (entrambi i peer devono avere stessa versione). Dopo di che si autentica il peer con una chiave tramite il comando pre-share-key cipher <chiave> e infine si specifica l'indirizzo IP dell'altro peer che compone il tunnel con remote-address <IP_peer_remoto>.

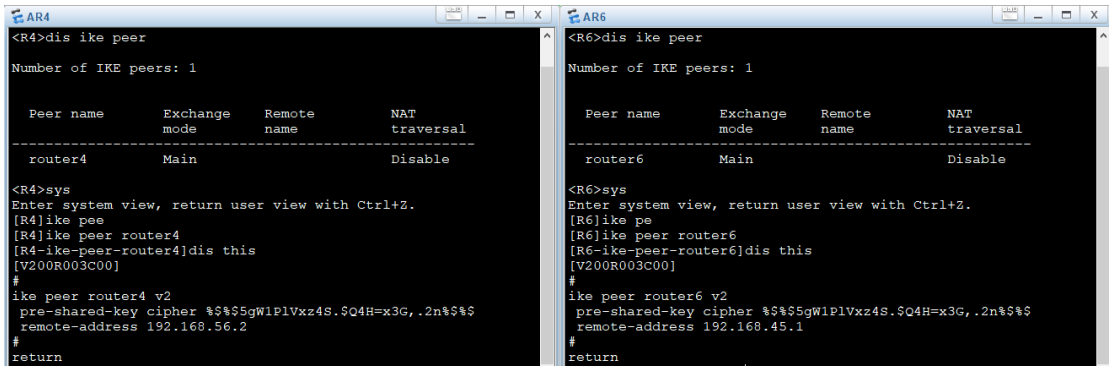


Figura 41: definizione ike peer per identificare e autenticare il dispositivo di riferimento

A questo punto si può creare a tutti gli effetti la IPSec policy con il comando ipsec policy <nome_policy> 1 isakmp dove si specifica appunto la volontà di utilizzare il protocollo ISAKMP per lo scambio delle chiavi. All'interno della policy vanno specificati i parametri precedentemente definiti, ovvero:

- l'acl con il comando security acl 3000;
- la proposal con il comando proposal <nome_proposal>;
- il peer con il comando ike-peer <nome_peer>.

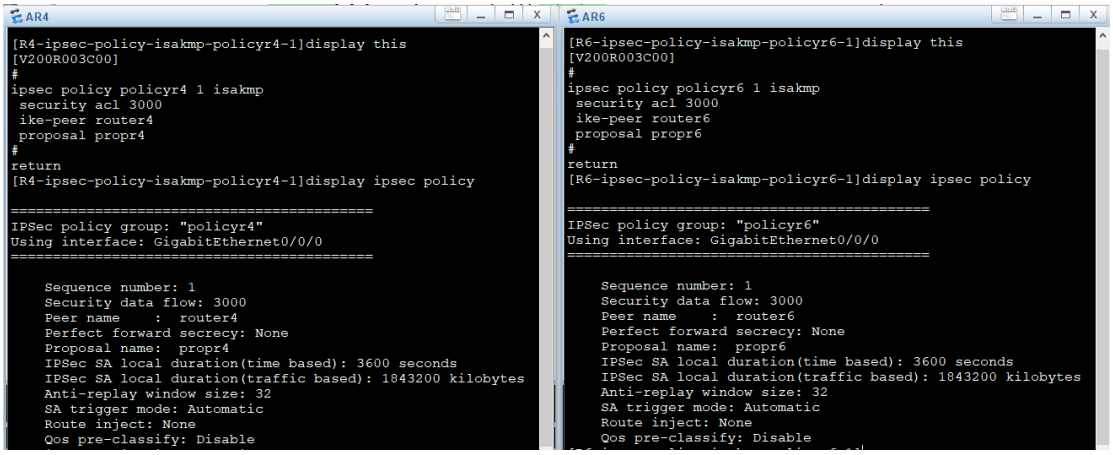


Figura 42: definizione della policy ipsec di R4 e di R6 attraverso il protocollo ISAKMP

Con la *figura 42* si mostrano i comandi eseguiti nella policy e le informazioni che vengono scambiate con essa.

4. **Applicare la policy sull'interfaccia interessata**

Infine, per poter attivare il tunnel, si deve applicare la policy all'interno dell'interfaccia fisica che veicola i pacchetti sicuri. Quindi sia su R4 che su R6 nell'interfaccia GigabitEthernet 0/0/0 è stato assegnato il seguente comando: ipsec policy <nome_policy>. Attraverso questo comando, quindi, inizierà la negoziazione delle chiavi tramite ISAKMP e una volta stabiliti e definiti tutti i parametri che configurano il tunnel, tutti i pacchetti GRE che passano sul tunnel saranno criptati e quindi non accessibili all'esterno.

Nella *figura 43* è mostrata una cattura WireShark dove viene evidenziato proprio questo processo attraverso lo scambio di messaggi da parte dei due peer. C'è una prima parte in cui R4 e R6 configurano la Security Association (i parametri del tunnel) attraverso il protocollo ISAKMP e una volta che i due peer sono stati autenticati, gli algoritmi di crittografia e autenticazione entrano effettivamente in funzione. Come si può notare dal contenuto di un pacchetto qualsiasi, inoltrato sul tunnel, il contenuto è completamente nascosto all'esterno e quindi la comunicazione è assolutamente sicura da attacchi per reperire informazioni dai messaggi.

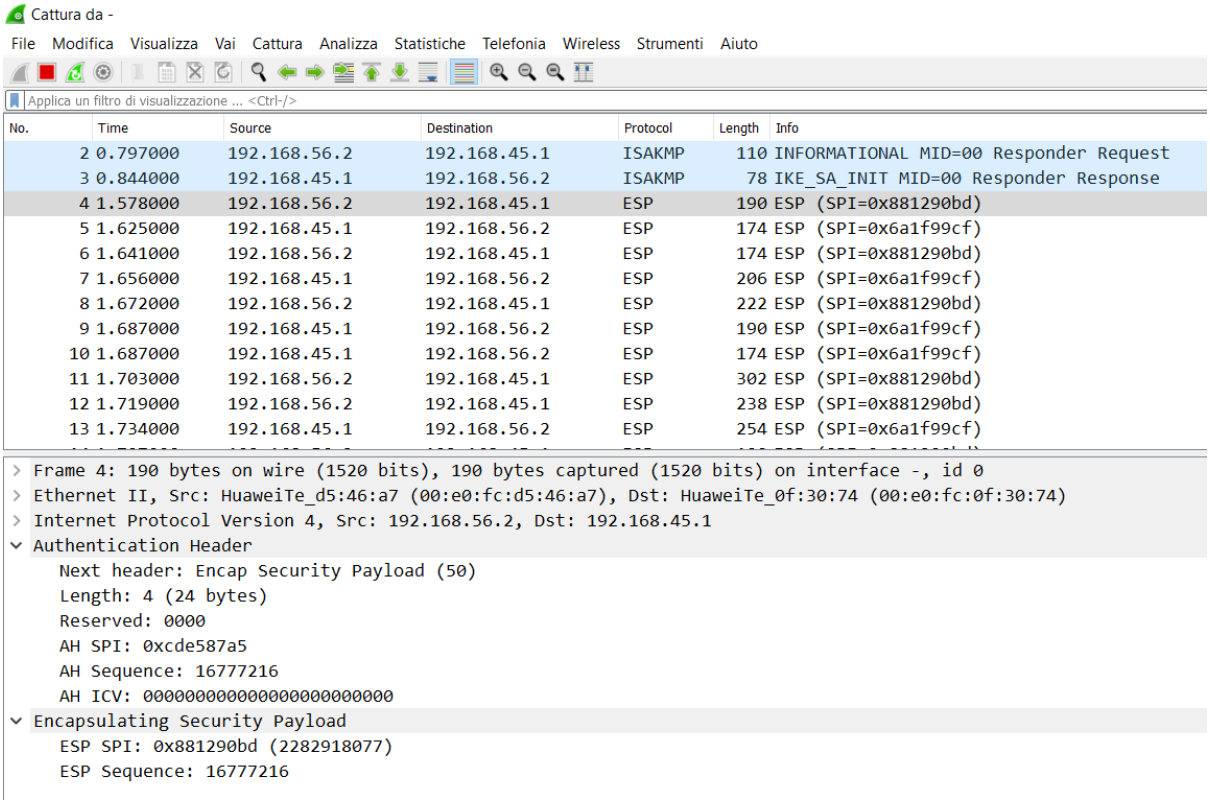


Figura 43: cattura WireShark per evidenziare i messaggi ISAKMP (per negoziare le chiavi) e dopo di che la riservatezza del contenuto dei pacchetti che passano sul tunnel IPSec

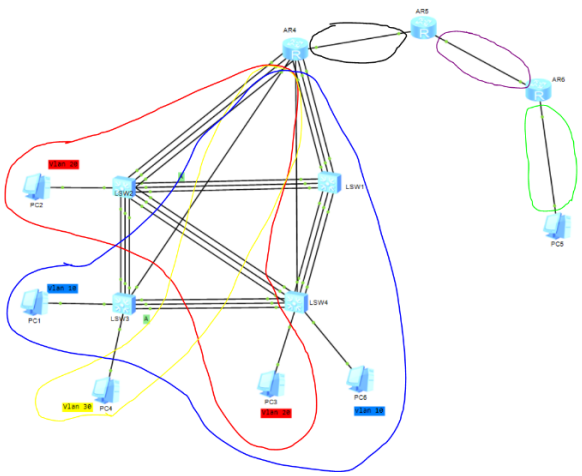
7. Conclusioni

7.1. Risultati e conclusioni

Per concludere l'elaborato si evidenzia il corretto funzionamento della rete progettata e configurata dal candidato, in tutti i suoi punti essenziali. Per fare ciò verranno ripercorse tutte le tecnologie, i protocolli utilizzati e i servizi offerti al fine di far comunicare al meglio tutti i nodi della rete.

Per prima cosa all'avvio della rete a tutti i terminali, rappresentati dai PC, viene assegnato automaticamente un indirizzo IP attraverso il DHCP dal server apposito, cioè R6. Grazie a questo protocollo, non è necessario che l'amministratore configuri manualmente nessun indirizzo IP ai nodi della rete.

La topologia è stata configurata in modo tale da suddividere, in 3 domini di broadcast diversi, il segmento di rete sottostante il router 4 grazie all'implementazione delle Virtual Local Area Networks. Nel complesso, infatti, esistono 6 diversi domini di broadcast tra cui:



- Vlan 10 – 192.168.10/24;
- Vlan 20 – 192.168.20.0/24;
- Vlan 30 – 192.168.30.0/24;
- Sottorete R4-R5 – 192.168.45.0/30;
- Sottorete R5-R6 – 192.168.56.0/30;
- LAN 5 – 192.168.50.0/24.

Figura 44: visualizzazione dei diversi domini di broadcast che contraddistinguono la topologia

Per popolare le tabelle di routing è stato utilizzato il protocollo OSPF (un protocollo di routing dinamico), che si adatta automaticamente a cambiamenti della rete. Grazie a quest'ultimo tutti i nodi della rete possono raggiungersi e ciò è stato evidenziato nel corso dell'elaborato grazie a comandi ICMP come ping e tracert.

Non tutti i nodi della rete possono raggiungersi perché è stata implementata un ACL che permette di filtrare il traffico verso la LAN 5. Infatti, solo agli host e all'interfaccia

di R4 che appartengono alla VLAN 20 è permesso di raggiungere la LAN 5, mentre per tutti le altre sottoreti è negato l'accesso. Inoltre, per migliorare la riservatezza dei dati, è stata analizzata una diversa implementazione dove viene creato un tunnel IPSec che veicola al suo interno il traffico GRE per ottenere un collegamento diretto tra R4 e R6 che cripti e autentichi i pacchetti di diversi protocolli, e non solo IP.

Come mostrato per tutto l'elaborato e in questo breve riassunto, gli obiettivi del progetto sono stati raggiunti in quanto la rete funziona correttamente. Non ci sono picchi di traffico che rallentano i dati trasmessi e tutti i dispositivi svolgono il loro compito come dovrebbero. I servizi (DHCP) e le tecnologie (Link aggregation, VLAN) richieste sono perfettamente funzionanti e rendono più efficiente la topologia sia per quanto riguarda la gestione, che per la velocità di quest'ultima. In merito alla manutenzione della rete, essa si comporta in maniera accettabile in quanto utilizza un protocollo di routing dinamico: OSPF che ha un tempo di convergenza relativamente basso. Oltre al routing è stato analizzato il protocollo RSTP, utilizzato per gestire gli switching loops nel segmento che contiene le VLAN 10, 20 e 30; RSTP ha tra i suoi punti di forza il fatto che riesce ad adattarsi a failure della rete in maniera molto veloce.

7.2. Competenze acquisite e riflessioni

Questo progetto è stato molto utile per il mio percorso formativo, infatti con esso ho ampliato di gran lunga le mie conoscenze acquisite nei tre anni di studi. Ho potuto affrontare un argomento che è al di fuori del piano formativo offerto dall'Università e approfondire tematiche molto interessanti e richieste soprattutto nel mondo del lavoro. Mi ritengo molto soddisfatto dal percorso fatto perchè nella prima parte ho appreso le basi teoriche, che poi ho applicato in questo progetto con cui ho potuto approcciare i processi e i dispositivi che rendono funzionante una rete informatica. Oltre alle competenze teoriche e pratiche strettamente legate al progetto, ho acquisito altre capacità molte importanti. La prima tra tutte è quella di problem solving; durante lo sviluppo della rete ho dovuto affrontare difficoltà e problemi che dovevano essere risolti. In questo contesto, inoltre, non esiste una unica soluzione ad un problema e a volte alcune scelte sono computazionalmente migliori di altre quindi preferibili; proprio per questo ho dovuto sperimentare e verificare diverse soluzioni plausibili e scegliere la migliore tra esse. Tutto ciò è servito per ampliare il bagaglio di competenze acquisite in questa esperienza.

Spero di riuscire a mettere in pratica in futuro le competenze assimilate in questi mesi di lavoro perché sono concetti molto interessanti che mi piacerebbe approfondire in un eventuale sbocco lavorativo futuro.

8. Bibliografia

(1) *4 reasons why you should be using Huawei eNSP,* https://forum.huawei.com/enterprise/en/4-reasons-why-you-should-be-using-huawei-ensp/thread/600564-861

(2) *LIVELLO 2: DATA-LINK,* http://didatticainfo.altervista.org/Quinta/DataLink.pdf

(3) *Dominio di broadcast,* https://it.wikipedia.org/wiki/Dominio_di_broadcast

(4) *Congestione (reti),* https://it.wikipedia.org/wiki/Congestione_(reti)

(5) *Che cos'è il protocollo Internet (IP),* https://www.ionos.it/digitalguide/server/know-how/che-cose-il-protocollo-internet-definizione-di-ip-co/

(6) *ICMP, ping, traceroute (tracert),* https://corsocescot.files.wordpress.com/2012/04/icmp_ping_traceroute_20120402.pdf

(7) *Che cos'è un indirizzo IP e a che cosa serve?,* https://www.avg.com/it/signal/what-is-an-ip-address#topic-1

(8) *Come si calcolano il prefisso, la rete, la sottorete e i numeri host?,* https://qastack.it/networkengineering/7106/how-do-you-calculate-the-prefix-network-subnet-and-host-numbers

(9) *Convergenza di una internetwork,* https://www.eforhum.it/news/convergenza-internetwork-cisco-ccna/

(10) *Instradamento statico*, https://it.wikipedia.org/wiki/Instradamento_statico

Ringraziamenti

Alla fine di questo lungo e impegnativo percorso, vorrei ringraziare tutte le persone a me care che mi hanno aiutato e sostenuto durante questi tre anni.

Innanzitutto, ringrazio la mia famiglia sia per il sostegno economico che, soprattutto, per aver sempre creduto in me e per avermi fatto capire quanto lo studio sia importante per il mio futuro.

Un ringraziamento speciale va ad Alessia per essermi stata sempre accanto, in qualsiasi momento, sia di gioia che di difficoltà e per avermi trasmesso positività e serenità nei momenti duri di questo percorso. Ha condiviso, a pieno con me, questo percorso, perciò la ringrazio di cuore e condivido con lei questo bellissimo traguardo.

Ringrazio i miei amici, Giovanni, Dalila, Andrea, Filippo, Fabrizio, Devis, Lorenzo, Edoardo e Sergio per avermi fatto divertire, sorridere e per aver reso più leggeri i periodi intensi di studio. Ringrazio, inoltre, i miei compagni di corso Giovanni e Daniele che sono stati la mia spalla durante questi tre anni e perché condividendo con loro questo percorso ho trovato due nuovi amici importanti.

Infine, ringrazio i professori Ennio Gambi e Adelmo De Santis che mi hanno seguito con grande impegno e pazienza durante lo sviluppo del tirocinio e della tesi, aiutandomi a sviluppare e raggiungere questo (spero) bel risultato.