Università degli Studi di Camerino

FACOLTÀ DI SCIENZE E TECNOLOGIA
Corso di Laurea Triennale in Informatica

Tesi di laurea triennale

Implementazione dell'algoritmo OSPFv2 e Tunneling MPLS in una rete WAN

Candidato:
David Pecorella
Matricola 077884

Relatore:
Prof. Fausto Marcantoni

Correlatore:
Ing. Antonio Della Selva

Anno Accademico 2018/2019

Indice

Elenco delle figure

Capitolo 1

Introduzione

Lo scopo di questo lavoro è andare ad espandere l'attuale rete WIRTEK di proprietà della NEWTEC Srl di Matelica, implementando un algoritmo di routing, in questo caso OSPFv2. Al momento la rete è composta per lo più da router con tecnologia MikroTik, ed è gestita in modalità mista con percorsi statici e aree sottoposte ad algoritmi di routing; implementando questo algoritmo andremo ad aggiornare quella parte di rete che sottende ancora a route statiche in modo da gestire in maniera più veloce e ottimale la rete sia quando non ci sono guasti, sia quando c'è un problema. Inoltre è stato deciso di implementare anche la tecnologia MPLS, per creare dei tunnel VPLS in modo da migliorare la gestione delle reti private delle aziende e separarle dal resto della rete.

Capitolo 2

Cenni di teoria

2.1 MikroTik

MikroTik è una compagnia Lettone che sviluppa e fornisce soluzioni hardware e software per connessioni internet; la componente software sviluppata da loro si chiama MikroTik RouterOS, invece la componente hardware si chiama RouterBOARD. La componente software, RouterOS, è basata su kernel Linux ed eredita tutte le funzioni opensource di quest'ultimo, il suo valore è di aver semplificato e standardizzato la configurazione dei molteplici servizi di networking presenti in Linux. Questo sistema operativo è gestibile tramite l'accesso via FTP, Telnet o Secure Shell(SSH) con un linguaggio proprietario, o in alternativa si può usare un'interfaccia grafica tramite un programma Windows chiamato Winbox (Fig. 2.1).

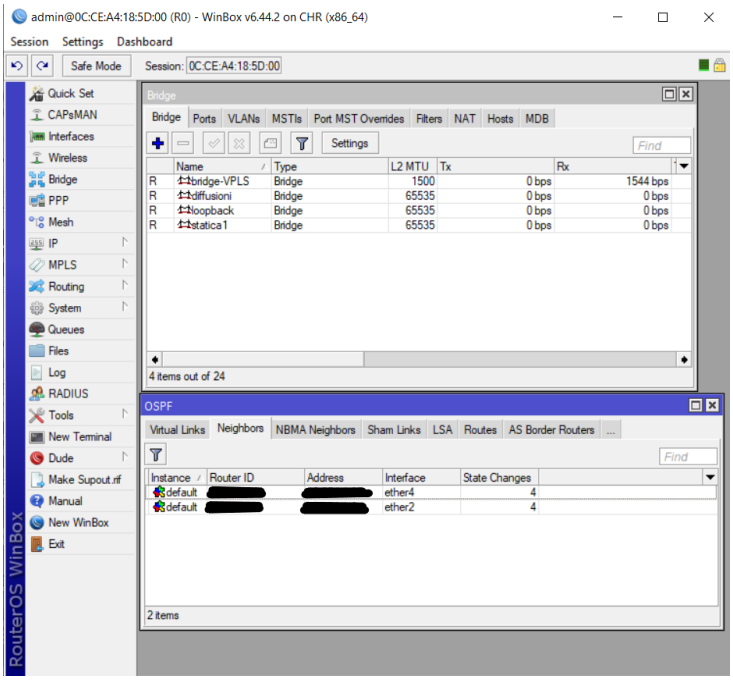

Figura 2.1: Winbox

Figura 2.2: Console MikroTik

Il sistema RouterOS può essere installato sia su hardware proprietario della società, o può essere installato su un computer serie x86, trasformandolo in un router, implementando tutte le funzioni aggiuntive date dal sistema operativo. RouterOS supporta sia multi-core sia multi-CPU, richiede un minimo di 32 MB di ram e 64 MB di memoria sia IDE, SATA, USB che flash; invece per le schede di rete possono essere utilizzate tutte quelle supportate dal kernel linux v3.3.5. Questo sistema operativo inoltre supporta molte applicazioni usate generalmente dai vari fornitori di servizi internet tra i quali troviamo i seguenti servizi di routing:

- Static routing
- Virtual Routing and Forwarding (VRF)
- Policy based routing
- Interface routing
- ECMP routing
- IPv4 dynamic routing protocols: RIP v1/v2, OSPFv2, BGP v4
- IPv6 dynamic routing protocols: RIPng, OSPFv3, BGP
- Bidirectional Forwarding Detection (BFD)
- MPLS e VPLS

Inoltre sono implementati in esso vari servizi di firewalling e le varie funzioni di VPN come IPSec, VLAN e le caratteristiche avanzate del PPP.

2.2 Autonomous System

Un autonomous System (AS) è un insieme di router e reti, gestito da una singola autorità amministrativa o da un dominio che presenta una politica di routing comune e chiaramente specificata, che usa un IGP (Interior Gateway Protocol) per le rotte interne all'AS, ed usa un EGP (Exterior Gateway Protocol) per inviare i pacchetti verso gli altri AS; che corrisponde all'ambiente

in cui andremo ad operare in questo progetto. Ogni AS è completamente indipendente: può decidere l'instradamento interno secondo le proprie preferenze, ed i pacchetti IP sono instradati al suo interno secondo le regole interne. Ogni AS può avere uno o più domini di instradamento interni serviti da protocolli IGP, ogni dominio può adottare il suo protocollo IGP preferito, e grazie alla ridistribuzione può scambiare le informazioni di instradamento con gli altri domini. Un sistema autonomo ad esempio può essere una rete interna di un'azienda o una rete di un provider. Quando un Autonomous System viene registrato da un ISP, gli viene associato un autonomous system number (ASN) unico che identifica ogni rete in internet, che viene usato nel routing BGP (Border Gateway Protocol) nell'interscambio di informazioni tra router di sistemi autonomi differenti. Il numero del sistema autonomo fino al 2007 era definito da un intero a 16 bit, poi da un intero a 32 bit adeguando anche gli AS precedenti al 2007. Alcuni numeri sono però riservati e non devono essere usati (fig. 2.3), tutti gli altri ASN vengono assegnati dallo IANA (Internet Assigned Numbers Authority) ai diversi RIR (Regional Internet Registries), che a loro volta li assegneranno alle varie entità.

Number	Bits	Description	Reference
0	16	Reserved	RFC1930, RFC7607
1 - 23455	16	Public ASN's	
23456	16	Reserved for AS Pool Transition	RFC6793
23457 - 64495	16	Public ASN's	
64496 - 64511	16	Reserved for use in documentation/sample code	RFC5398
64512 - 65534	16	Reserved for private use	RFC1930, RFC6996
65535	16	Reserved	RFC7300
65536 - 65551	32	Reserved for use in documentation and sample code	RFC4893, RFC5398
65552 - 131071	32	Reserved	
131072 - 4199999999	32	Public 32-bit ASN's	
4200000000 - 4294967294	32	Reserved for private use	RFC6996
4294967295	32	Reserved	RFC7300

Figura 2.3: ASN speciali

Gli autonomous system possono essere raggruppati in quattro categorie differenti, a seconda della loro connettività e della politica di operatività:

- Multihomed: un AS che è connesso a più di un solo AS, questo consente all'AS di rimanere connesso ad internet in caso di errore in una delle sue

connessioni. Tuttavia però questo tipo di AS non consente al traffico di passare da un AS ad un altro passando attraverso di lui.

- Stub: un AS che è connesso ad un solo altro AS, considerato un estensione di un altro AS.
- Transit: un AS che connette altre reti attraverso di lui.
- Internet Exchange Point: una infrastruttura attraverso la quale gli ISP o i CDN (Content Delivery Networks) scambiano il traffico internet tra i loro AS.

2.3 Cosa è il routing

Il routing è l'instradamento dei pacchetti che viene fatto dai router a livello di rete (OSI level 3). I router usano tabelle di instradamento composte da blocchi di indirizzi IP, chiamate routes, per sapere dove indirizzare i pacchetti che gli arrivano. L'instradamento è un processo di rete che determina i percorsi dei vari pacchetti nel loro viaggio dalla sorgente alla destinazione. Le tabelle di instradamento o di routing, oltre a fornire le informazioni di routing per le reti collegate direttamente e le reti remote, contengono anche informazioni su come è stata appresa la route, sulla relativa attendibilità e valutazione, sull'ultimo aggiornamento e quale interfaccia utilizzare per raggiungere la destinazione richiesta. Quando un pacchetto arriva all'interfaccia del router, questo analizza l'header del pacchetto per determinare la rete di destinazione. Se la rete di destinazione corrisponde a una route nella tabella di routing, il router inoltra il pacchetto utilizzando le informazioni specificate nella tabella di routing. Se sono presenti due o più route possibili verso la stessa destinazione, viene utilizzata la metrica per decidere quale route inserire nella tabella di routing. Se nella tabella di routing non è presente una route che rappresenta la rete di destinazione, il pacchetto viene scartato o può utilizzare un gateway predefinito per inoltrare un pacchetto a una destinazione sconosciuta.

2.4 Algoritmi di routing

L'algoritmo di routing è la strategia adottata dalla rete per decidere quale percorso o route intraprendere. Se la rete usa i datagrammi questa decisione va ripetuta per ogni pacchetto di dati in arrivo visto che il percorso migliore può variare nel tempo; se invece la rete utilizza circuiti virtuali le decisioni vengono prese solo quando viene impostato un nuovo circuito virtuale, in questo caso viene chiamato anche routing di sessione, perchè un percorso rimane valido per tutta la sessione uscente. Gli algoritmi di routing hanno delle proprietà a prescindere dal tipo di connessione che sono: correttezza, semplicità, robustezza, stabilità, imparzialità ed ottimizzazione. Correttezza e semplicità si descrivono da sole ma le altre sono meno ovvie. Per robustezza si intende che la rete deve essere in grado di far fronte ai cambiamenti di topologia e di traffico, che avverranno inevitabilmente nel tempo, come guasti hardware o software, senza che sia necessario interrompere la rete. Anche la stabilità costituisce un obiettivo importante per l'algoritmo di routing. Un algoritmo stabile raggiunge l'equilibrio e lì rimane. Per imparzialità ed efficienza la situazione cambia perché spesso l'uno va a disturbare l'altro, sono due proprietà che devono essere bilanciate tra di loro e sono necessari dei compromessi tra efficienza globale e imparzialità verso le singole connessioni. Prima di trovare questi compromessi è necessario scegliere cosa ottimizzare, in genere molte reti tentano di minimizzare la distanza da percorrere ossia il numero di salti che il pacchetto deve compiere, migliorando così il tempo di attesa che porta una riduzione della banda consumata e di conseguenza anche la capacità di carico è migliorata. Gli algoritmi di routing si possono suddividere in due gruppi principali: adattivi e non adattivi. Gli algoritmi adattivi cambiano le proprie decisioni secondo le modifiche apportate alla topologia e in genere anche al traffico, chiamati anche algoritmi di routing dinamici. Al contrario gli algoritmi non adattivi calcolano il percorso in anticipo e lo scaricano nei router all'avvio della rete, questa procedura viene chiamata anche routing statico, ma non essendo reattivi agli errori è utile quando le scelte di routing sono chiare. Un altro modo di classificare gli algoritmi di routing riguarda il fatto di essere globali o decentralizzati. Un

algoritmo globale calcola il percorso a costo minimo tra una sorgente e una destinazione avendo una conoscenza globale e completa della rete, ciò richiede che l'algoritmo in qualche modo ottenga tale informazione prima di effettuare il vero e proprio calcolo, che può essere svolto in una locazione (algoritmo di instradamento globale centralizzato) o replicato in più locazioni. Questi algoritmi con informazioni di stato globali sono spesso detti algoritmi link state, dato che l'algoritmo deve essere conscio del costo di ciascun collegamento della rete. In un algoritmo decentralizzato invece il percorso a costo minimo viene calcolato in modo distribuito e iterativo. Inizialmente i nodi conoscono soltanto i costi dei collegamenti adiacenti, poi attraverso un processo iterativo e lo scambio con i nodi vicini, un nodo gradualmente calcola il percorso a costo minimo verso una destinazione o un insieme di destinazioni. Questi tipi di algoritmo sono anche chiamati distance vector, poichè ogni nodo elabora un vettore di stima dei costi verso tutti gli altri nodi della rete.

2.4.1 Algoritmo Link State

Come detto in precedenza in un algoritmo link state la topologia di rete e tutti i costi dei collegamenti sono noti, in tal modo tutti i nodi dispongono di una vista identica e completa della rete e ciascun nodo che lancerà l'algoritmo otterrà gli stessi risultati. Gli algoritmi di link state sono basati sull'algoritmo di Dijkstra, che calcola il percorso a costo minimo da un nodo a tutti gli altri nodi nella rete, partendo da un'inizializzazione seguita da un ciclo che viene eseguito una volta per ogni nodo del grafo, quando termina si avrà calcolato il percorso minimo dal nodo origine a tutti gli altri nodi. Quando l'algoritmo Link State termina, avremo per ciascun nodo il suo predecessore lungo il percorso a costo minimo dal nodo origine, e per ciascun predecessore avremo il rispettivo predecessore, e in questo modo riusciamo a costruire l'intero percorso dall'origine a tutte le destinazioni. Ogni router che utilizza questo tipo di algoritmo compie delle azioni che possono essere riassunte nel seguente modo:

- Scoperta dei vicini: appena acceso un router la prima azione che compie è scoprire i propri vicini e la compie inviando un pacchetto chiamato

HELLO su ogni linea punto a punto; ogni router che riceve questo tipo di pacchetti deve rispondere fornendo il proprio nome, che è globalmente unico. Quando 2 o più router sono connessi tra di loro tramite una connessione broadcast (uno switch, una rete ad anello o altro) si crea un aumento della topologia e causando una serie di messaggi non necessari, in questi casi entra in gioco una figura chiamata router designato che agisce come un singolo nodo della LAN.

- Misurazione del costo dei collegamenti: per calcolare i cammini minimi l'algoritmo richiede che ogni collegamento abbia una metrica di costo o distanza; questo costo può essere configurato manualmente dall'operatore di rete o automaticamente.

- Costruzione dei pacchetti che contengono lo stato dei collegamenti: raccolte tutte le informazioni necessarie per lo scambio, ogni router deve costruire un pacchetto contenente tutti i dati; questo pacchetto è così composto: identità del trasmittente, seguita da un numero di sequenza, dall'età e da una lista dei vicini. Per ogni vicino è riportato il ritardo misurato.

- Distribuzione dei pacchetti che contengono lo stato dei collegamenti: per tenere sotto controllo il flusso di dati, in ogni pacchetto è inserito un numero di sequenza, incrementato per ogni nuovo pacchetto inviato. I router tengono traccia di tutte le coppie (router sorgente - sequenza) rilevate. Quando arriva un nuovo pacchetto contenente le informazioni sullo stato del collegamento, il router confronta i dati con quelli che già ha; se sono nuovi, il pacchetto viene inoltrato su tutte le linee tranne quella di ricezione, altrimenti il pacchetto viene scartato. Per evitare numeri di sequenza ripetitivi si utilizzano numeri di sequenza di 32 bit. Per ovviare a problemi dovuti a corruzione di dati o blocchi del router, che in entrambi casi porterebbero a scartare dei pacchetti, si usa il campo età che subisce un decremento una volta al secondo e decrementato da ogni router durante il processo di flooding iniziale.

- Calcolo di nuovi percorsi: eseguito tramite l'algoritmo di Dijkstra.

Un problema che si può verificare con gli algoritmi di instradamento che utilizzano una metrica dei collegamenti basata sulla congestione o sul ritardo,facendo così cambiare ogni volta il percorso minimo, viene chiamato oscillazione. Una soluzione potrebbe consistere nello stabilire che i costi dei collegamenti non dipendono dalla quantità di traffico trasportato, ma non è accettabile dato che uno scopo dell'instradamento è evitare i congestionamenti. Un'altra soluzione consiste nell'assicurarsi che non tutti i router lancino algoritmi Link State nello stesso istante.

2.4.2 Algoritmo Distance Vector

L'algoritmo distance vector è iterativo, asincrono e distribuito:

- iterativo perchè questo processo si ripete fino a quando non avviene un ulteriore scambio di informazioni tra vicini;
- asincrono perchè non richiede che tutti i nodi operino al passo con gli altri;
- distribuito perchè ciascun nodo riceve parte dell'informazione da uno o più dei suoi vicini direttamente connessi a cui, dopo aver effettuato il calcolo, restituisce i risultati.

Questi algoritmi operano facendo in modo che ogni router conservi una tabella che definisce la distanza conosciuta migliore per ogni destinazione e il collegamento che conduce ad essa. Questa tabella usa come indice i router della rete e vi è memorizzata una voce per ognuno di essi; questa voce è composta dalla linea di trasmissione da utilizzare per quella destinazione e una stima del tempo o della distanza associata ad essa. Questo tipo di algoritmo sebbene converga verso la risposta corretta, può raggiungere l'obiettivo molto lentamente, questo problema è noto anche come conteggio all'infinito, dovuto al fatto che quando un nodo A dice a B che ha un percorso che punta da qualche parte, B non ha modo di sapere se fa parte di quel percorso, quindi nel momento che si crea un problema di collegamento occorre parecchio tempo prima che tutti i nodi della rete se ne accorgano. Si consideri ad

esempio la situazione mostrata nella Figura 2.4, in cui tutte le linee e i router sono inizialmente attivi. La distanza tra A e i router B, C, D ed E è rispettivamente di 1, 2, 3 e 4. Improvvisamente A si spegne oppure la linea tra A e B si interrompe. Al primo scambio di pacchetti B non riceve nulla da

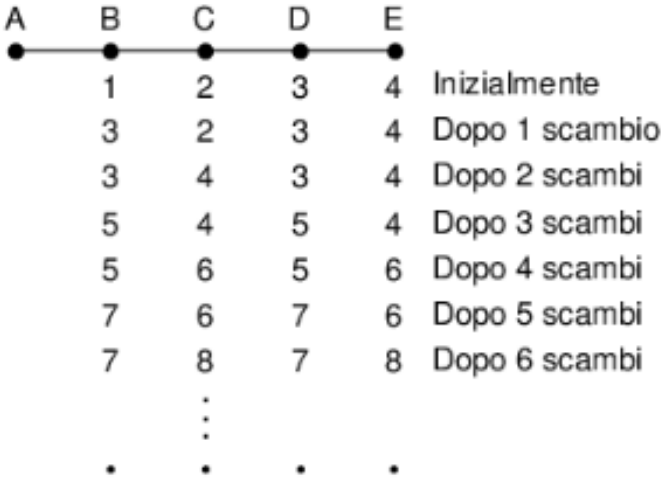

Figura 2.4: Problema del conteggio dell'infinito

A. Fortunatamente C dice: "Non ti preoccupare, io ho un percorso che dista 2 da A." B non può sapere che il percorso di C passa da B stesso. Per quanto ne sa B, C potrebbe avere dieci linee, tutte con percorsi separati diretti ad A di lunghezza 2. Di conseguenza, B pensa di poter raggiungere A attraverso C con un percorso di lunghezza 3. D ed E non aggiornano le loro voci relative ad A durante il primo scambio. Al secondo scambio C nota che tutti i suoi vicini affermano di essere a distanza 3 da A; sceglie quindi a caso uno dei vicini e imposta a 4 la sua nuova distanza da A, come mostrato nella terza riga della Figura 2.4. Gli scambi seguenti producono i risultati mostrati nelle altre righe della Figura 2.4; l'algoritmo va avanti così fino ad arrivare all'infinito.

2.5 Protocollo OSPF

OSPF è un protocollo basato sull'algoritmo Link State ed è nato per avere delle caratteristiche specifiche:

- la prima tra tutte è che deve essere open, nel senso che non doveva avere vincoli di brevetto;
- la seconda è che deve supportare diverse metriche di distanza;
- la terza è che deve essere un algoritmo dinamico in grado di modificarsi rapidamente e automaticamente in base alla topologia;
- la quarta è che deve supportare il routing basato sul tipo di servizio cioè doveva essere in grado di instradare il traffico in tempo reale in un modo e il traffico di altro genere in un altro;
- la quinta è che deve eseguire il bilanciamento del carico, ossia suddividere il carico su più linee;
- la sesta è che deve necessariamente supportare i sistemi gerarchici;
- la settima riguarda la possibilità di implementare la sicurezza della rete per evitare intrusioni;
- infine la necessità di provvedere alla gestione dei router collegati ad internet tramite tunnel.

OSPF è classificato come Interior Gateway Protocol (IGP), perché distribuisce le informazioni di routing tra i router appartenenti ad un singolo AS. OSPF supporta sia collegamenti punto a punto sia reti broadcast, OSPF opera riassumendo l'insieme di reti reali, router e linee in un grafo orientato in cui ad ogni arco è assegnato un costo, quindi calcola il percorso più breve in base al valore degli archi. Come detto in precedenza quando sono presenti più percorsi con la stessa lunghezza, OSPF ricorda l'intero insieme dei percorsi più brevi e durante l'inoltro del pacchetto il traffico viene suddiviso su di essi; questa procedura che aiuta a bilanciare il carico è chiamata ECMP

(equal cost multipath). Molte reti o AS (autonomous system) sono difficili da gestire quando sono molto grandi, e in questo OSPF mette a disposizione la possibilità di dividere tali AS in più aree numerate, dove ogni area è una rete o un insieme di reti contigue. Queste aree non si sovrappongono, ma non hanno bisogno di coprire tutta la rete. Ogni AS possiede un'area dorsale (backbone area), denominata area 0 e i router presenti in questa area sono chiamati router di dorsale; tutte le altre aree sono collegate alla backbone, quando questo non è possibile OSPF mette a disposizione una funzione chiamata Virtual Link, che altro non è che un tunnel tra i router di confine (Fig. 2.5). Per passare da un'area ad una qualunque altra area

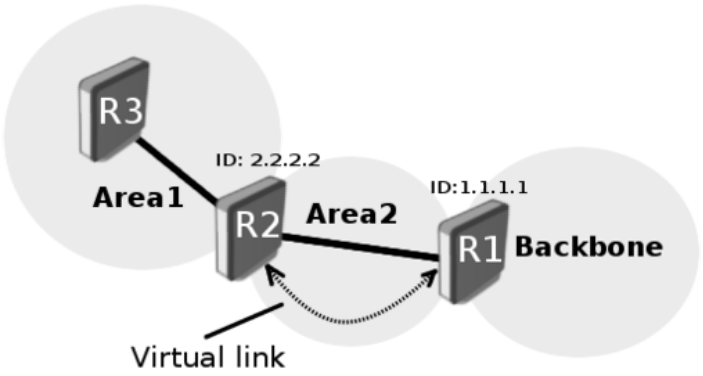

Figura 2.5: Virtual Links

dello stesso AS di conseguenza bisogna passare per la backbone. Come per le altre aree, anche la topologia della backbone non è visibile al di fuori della backbone area. I router che si trovano completamente in un'area sono detti router interni (internal router). Ogni router connesso a due o più aree viene chiamato router di area di confine (area border router) e deve quindi far parte della backbone; il suo compito è di elencare le destinazioni di un'area e inviarle alle aree a cui è connesso, questo elenco contiene anche informazioni di costo, ma non della topologia all'interno dell'area. Tuttavia se esiste un solo router di confine fuori da un'area, anche tale elenco non ha bisogno di essere trasmesso, ed i percorsi verso destinazioni fuori dall'area iniziano sempre con l'indicazione di andare verso il router di confine, e generalmente queste aree

sono definite stub area. Passare le informazioni di costo permette agli host nelle altre aree di trovare il miglior router di confine da usare per entrare in un'area. Non passare le informazioni sulla topologia riduce il traffico e semplifica l'elaborazione dei percorsi più brevi verso i router delle altre aree. L'ultima tipologia di router (Fig. 2.6) che prenderemo in considerazione è detto router di confine dell'AS (AS boundary router), ed il suo scopo è trasmettere nell'area i percorsi verso destinazioni esterne poste in altri AS, e quindi i percorsi esterni appaiono come le destinazioni e possono essere raggiunti tramite il router di confine dell'AS con un costo. Un router può avere uno o più ruoli ad esempio un router di confine è anche un router di backbone. Il compito principale di ogni router è calcolare il percorso più breve da sé stesso ad ogni altro router nell'AS; un router di confine dell'area ha bisogno dei database di tutte le aree a cui è connesso ed esegue separatamente un diverso algoritmo di ricerca del percorso più breve per ogni area. Come

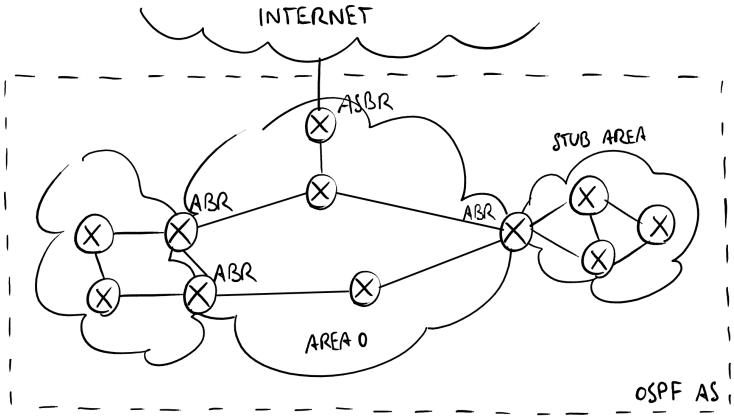

Figura 2.6: Tipologia di router OSPF

detto precedentemente i router in diverse aree, per trasmettere tra di loro, devono passare per la backbone area e quindi i router possono usare diversi router di confine dell'area per entrare nella backbone ed infine nell'area di destinazione. I pacchetti sono inoltrati dalla sorgente alla destinazione così come sono, a meno che non venga usato un altro protocollo ad esempio MPLS. Inoltre i percorsi verso destinazioni esterne possono includere il costo esterno dal router di confine sul cammino esterno oppure avere solo il costo interno fino all'AS. OSPF funziona scambiando informazioni tra i router

adiacenti tra loro, e per evitare che ogni router su una LAN comunichi con ogni altro router sulla LAN, entra in gioco un router chiamato router designato. Il router designato è detto adiacente agli altri router della sua LAN e scambia informazioni con gli altri apparati, agendo come un singolo nodo che rappresenta la LAN; insieme al router designato viene eletto anche un router designato di backup che in caso di problemi al router designato, diventa lui stesso il router designato. Durante il normale funzionamento ogni router invia periodicamente in flooding messaggi LINK STATE UPDATE, ognuno con un numero di sequenza, ad ogni router adiacente, indicando così lo stato del dispositivo ed i costi utilizzati nel database della topologia; ad ognuno di questi messaggi segue una risposta di LINK STATE ACK che rende questo tipo di messaggi affidabili. I messaggi DATABASE DESCRIPTION sono utilizzati quando una linea viene attivata, e contengono i numeri di sequenza di tutte le voci relative allo stato dei collegamenti possedute in quel momento dal trasmittente, in questo modo il ricevente può confrontare i propri valori con quelli del trasmittente e scoprire chi ha i valori più recenti. Ogni router può richiedere ad un router adiacente informazioni sullo stato dei collegamenti usando i messaggi LINK STATE REQUEST, il risultato di questo algoritmo è che ogni coppia di router adiacenti controlla chi dei due abbia i dati più recenti, così facendo le nuove informazioni si propagano attraverso l'area. Tutti questi tipi di messaggi sono inviati direttamente come pacchetti IP.
Riepilogando il flooding permette ad ogni router di comunicare a tutti gli altri router della sua area chi sono i vicini e i costi utilizzati, così facendo ogni router si costruisce un grafo dell'area e quindi di calcolare il percorso più breve; anche la backbone area fa la stessa cosa ed inoltre i router di backbone accettano informazioni dai router di confine d'area in modo da poter calcolare il percorso migliore diretto da tutti i router di backbone ad ogni altro router. Queste informazioni si propagano indietro verso i router di confine d'area, che le annunciano dentro le proprie aree. Una caratteristica molto importante di questo protocollo di routing è la sicurezza, cioè che gli scambi tra router OSPF possono essere autenticati, in questo modo soltanto router fidati possono prendere parte al protocollo OSPF in un AS, evitando

che malintenzionati immettano informazioni errate nelle tabelle di routing. Di base nei router OSPF questa funzione è disattivata. Esistono 2 tipi di autenticazione:

- Semplice, la meno sicura, su tutti i router si configura la stessa password che deve essere inclusa, in chiaro, nei pacchetti OSPF.
- MD5, più sicura e si basa su chiavi segrete condivise configurate in ogni router; per ogni pacchetto OSPF che viene inviato i router calcolano l'hash MD5 del contenuto a cui è stata aggiunta la chiave segreta e includono il risultato nel pacchetto, il ricevente calcola la funzione hash MD5 del pacchetto, usando la chiave segreta preconfigurata, e la confronta con il valore inserito nel pacchetto per verificarne l'autenticità.

In OSPF tra i vari router vengono scambiati diversi pacchetti di messaggi chiamati LSP (Link State Packet), questi pacchetti possono contenere anche uno o più messaggi chiamati LSA (Link State Advertisiment), inviati in multicast agli altri router della rete OSPF. I messaggi LSA più comuni sono di sei tipi (Fig. 2.7):

Value	Link Type	Description
1	*Router-LSA*	Link to a router
2	*Network-LSA*	Link to a network
3	*Summary-LSA (IP Network)*	When areas are used, summary information generated about a network.
4	*Summary-LSA (ASBR)*	When areas are used, summary information about a link to an AS boundary router.
5	*AS-External-LSA*	An external link outside the autonomous system.

Figura 2.7: LSA type

- LSA di tipo 1 (router): contiene informazioni sui router adiacenti a chi genera il LSA, nello specifico definisce lo stato delle interfacce e i relativi costi associati.

- LSA di tipo 2 (network): generato dai designated router per descrivere la topologia della LAN dove esso è collegato, e inviato agli altri router nella sua area.
- LSA di tipo 3 (summary): generati dai router di confine (ABR) per comunicare ai router interni, delle aree a cui sono collegati, come raggiungere le altre aree.
- LSA di tipo 4 (ASBR Summary): generati dai router di confine per annunciare la raggiungibilità di un router di confine dell'AS.
- LSA di tipo 5 (AS external): generati da un ASBR per annunciare destinazioni esterne al dominio di routing e vengono inviati a tutte le aree ma possono essere filtrati in aree come le stub area e le totally stub area.
- LSA di tipo 7 (NSSA): vengono generati dai router di confine all'ingresso di un area NSSA o TNSSA, per attraversare l'area.

Abbiamo accennato al fatto che esistono delle aree in OSPF e ne abbiamo accennate alcune, qui di seguito andremo a trattarle tutte nello specifico:

- Stub Area: queste aree hanno come caratteristica il fatto che gli LSA (Link State Advertisment) di tipo 5, cioè i percorsi esterni, né entrano né passano attraverso queste aree; il routing verso un percorso esterno in queste aree è indirizzato verso quello di default dell'area. Questo fa si che le dimensioni del database sia ridotto e quindi minor richiesta di memoria per i router dentro queste aree. Un'area può essere configurata come stub quando esiste un solo punto di uscita, o quando la scelta del punto di uscita non può ricadere su un percorso esterno. Inoltre non si possono configurare virtual links attraverso queste aree e non può esserci un router di confine (AS boundary router).
- Totally stub area: come la stub area non permette percorsi esterni (LSA tipo 5) ed inoltre neanche percorsi riassuntivi (LSA tipo 3) quindi i router dentro queste aree sanno solo i percorsi interni, per uscire da queste aree i router conoscono solo un percorso di default.

- Not so Stubby Area (NSSA): è simile alla Stub Area ma può importare percorsi esterni e quindi può esserci un router di confine. Quando un LSA di tipo 5 deve passare attraverso queste aree, i router di confine dell'area NSSA convertono gli LSA di tipo 5 in LSA di tipo 7 (viaggiano solo attraverso un'unica area NSSA), e viceversa quando un LSA di tipo 7 raggiunge un router di confine dell'area NSSA viene riconvertito in un LSA di tipo 5.

- Totally Not so Stubby area: come le aree NSSA blocca gli LSA di tipo 5 ed in più blocca anche gli LSA di tipo 3, comportandosi allo stesso modo delle aree NSSA, cioè convertendoli in LSA di tipo 7. (Fig. 2.8)

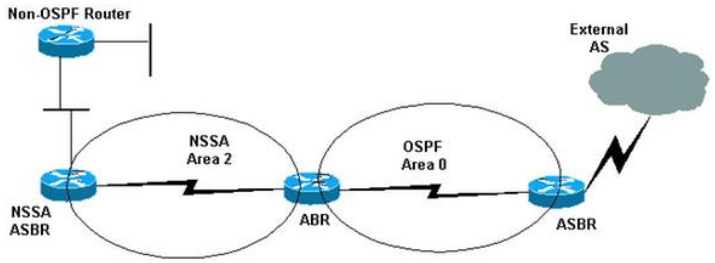

Figura 2.8: NSSA

2.5.1 Differenze tra OSPF v1 e v2

Tra le due versioni le differenze che ci sono hanno reso obsoleta la prima versione e incompatibile con la nuova versione. Un'importante novità che è stata implementata con la nuova versione è l'aver introdotto le stub area, dovuto ad un'esigenza di alcuni AS di non far entrare in determinate aree gli LSA 5 riducendo così le risorse necessarie in quell'area. Un'altra importante novità è come vengono gestiti i TOS (Type of Service), prima ogni router OSPF aveva una tabella di routing separata per ogni tipologia di servizio, impiegando molte risorse in termini di memoria e processore; con la nuova versione invece l'amministratore di sistema può configurare i router in modo da avere una singola tabella di routing (la tabella TOS 0), ottimizzando così le risorse dei router, anche se del traffico potrebbe prendere dei percorsi

non ottimali. Un'altra funzionalità che è stata introdotta è perchè in alcuni casi si verificava che veniva calcolato un extra-hop quando si calcolavano i percorsi verso destinazioni esterne, questo succede quando più router OSPF condividono una LAN con un router esterno (router Y) e solo un router OSPF (router X) scambia informazioni di routing con Y, i router OSPF nella LAN diversi da X inoltreranno i pacchetti destinati a Y attraverso X, generando un hop aggiuntivo. Per risolvere questo problema è stato aggiunto un campo agli LSA 5, chiamato indirizzo di inoltro, che indicherà l'indirizzo del router al quale devono essere inoltrati i pacchetti.

2.6 Border Gateway Protocol

Questa tipologia di protocollo è di tipo distance vector, e si occupa principalmente degli aspetti politici, a differenza degli IGP come OSPF che si occupano solo di trasportare i pacchetti nella maniera più efficiente possibile dalla sorgente alla destinazione. Per esempio, un AS aziendale potrebbe voler inviare pacchetti a qualunque sito di Internet e ricevere pacchetti da qualunque altro sito di Internet; ma potrebbe anche non essere disposto a trasportare pacchetti generati in un AS estraneo e diretti ad un altro AS estraneo, anche quando il suo AS si trova sul percorso più breve tra i due. D'altro canto, potrebbe essere pronto a trasportare il traffico di passaggio per i suoi vicini o anche per altri AS specifici che hanno pagato per questo servizio. Le aziende telefoniche, per esempio, potrebbero essere felici di svolgere la funzione di carrier per i loro clienti, ma non per i clienti di altre aziende. I protocolli di routing interdominio in generale, e BGP in particolare, sono stati progettati per consentire il rispetto di molti tipi di criteri di routing nel traffico tra AS. Le politiche di routing possono essere molto individuali e vengono implementate decidendo quale traffico può fluire e su quali linee tra AS. Una politica comune consiste nel fatto che un cliente di un ISP paghi un altro ISP per consegnare pacchetti a qualunque altra destinazione di Internet e ricevere i pacchetti inviati da qualunque altra destinazione. Si dice che l'ISP cliente compra un servizio di transito (transit service) da un ISP fornitore. Nel caso due o più AS decidano di scambiarsi il traffico direttamente senza

passare per un altrro AS, possono farlo tramite una politica chiamata peering; che consiste nello scambiarsi le tabelle di routing per gli indirizzi che stanno nelle loro reti. Si consideri comunque che il peering non è transitivo, quindi se un AS (AS1) che fa peering con un altro (AS2), è collegato esso stesso con un ulteriore AS (AS3) non è detto che consenta il traffico attraverso di lui tra i due AS (cioè tra AS 2 ed AS 3). Come detto prima BGP è un protocollo di tipo distance vector, ma è abbastanza diverso dai protocolli interdominio basati anche loro su distance vector, quale RIP. Abbiamo già visto che le politiche e non la distanza vengono usate per selezionare i percorsi da impiegare. Un'altra grande differenza è che i router BGP, invece di tenere traccia solo del costo di un cammino verso ogni destinazione, memorizzano tutto il cammino usato. Questo approccio è chiamato path vector protocol (protocollo a vettore di percorso). Il cammino consiste nel router del prossimo salto (che potrebbe trovarsi dall'altra parte dell'ISP, non adiacente) e la sequenza di AS, chiamata anche AS path, che il percorso ha seguito (data in ordine inverso). Infine coppie di router BGP comunicano tra di loro stabilendo connessioni TCP. Questo modo di operare fornisce comunicazioni affidabili e nasconde tutti i dettagli della rete che si attraversa. La regola per propagare i percorsi all'interno di un ISP è che i router di confine dell'ISP imparino, per consistenza, tutti i cammini visti da tutti gli altri router di confine. Se un router di confine di un ISP viene a conoscenza di un prefisso verso IP 128.208.0.0/16, gli altri router verranno a conoscenza di questo prefisso. Il prefisso sarà quindi raggiungibile da tutti i punti dell'ISP, indipendentemente da come i pacchetti siano entrati nell'ISP da altri AS. I router BGP scelgono quale percorso usare per ogni destinazione nel seguente modo: ogni router BGP può venire a conoscenza di un percorso per una certa destinazione da un router connesso all'ISP successivo e da tutti i router di confine (che hanno avuto notizie di cammini differenti dai router connessi ad altri ISP). Il router deve decidere quale percorso tra tutti è il migliore da utilizzare. In conclusione è l'ISP a decidere la politica con cui scegliere i percorsi preferenziali.
Alcune strategie comuni sono:

- La prima strategia afferma che i percorsi tramite reti in peering sono preferibili a quelli tramite provider di transito; i primi non costano nulla,

mentre i secondi sono a pagamento. Una strategia simile indica che i percorsi dei clienti hanno la preferenza più alta. Spedire il traffico dei clienti paganti direttamente è una buona politica commerciale.

- Una strategia di tipo diverso è rappresentata dalla regola per cui gli AS path sono i migliori. Questa strategia è opinabile, dato che un AS potrebbe essere una rete di qualunque dimensione, quindi un cammino attraverso tre piccoli AS potrebbe in pratica essere più breve di un cammino attraverso un solo grande AS. Tuttavia quelli brevi in media tendono a essere migliori e questa regola è un modo comune di vincere le indecisioni.

- Un'ultima strategia è preferire il percorso con un costo più basso all'interno dell'ISP. Questa strategia è chiamata early exit (uscita precoce) o hot-potato routing (routing della patata bollente) e ha il curioso effetto collaterale di rendere i percorsi asimmetrici.

2.7 MPLS

Questo protocollo è detto multi-protocollo perchè non è propriamente un protocollo di livello 3, dato che per impostare le etichette dei percorsi dipende da IP o da altri indirizzi a livello di rete; ma non è neanche di livello 2, perchè inoltra i pacchetti attraverso più salti e non attraverso un singolo collegamento. MPLS è un servizio orientato alla connessione usato in internet all'interno delle reti degli ISP; con MPLS i pacchetti IP sono incapsulati in un'intestazione MPLS avente un identificatore di connessione a 20 bit (Fig. 2.9), che consente ai router di inoltrare i dati in base a questa intestazione, invece che in base all'indirizzo di destinazione. I benefici maggiori di questa tecnica sono un routing flessibile e un inoltro adatto alla qualità del servizio, oltre che veloce.
L'intestazione MPLS generica ha quattro campi:

- il campo label, che contiene l'indice;

- il campo QoS che indica la classe di servizio;

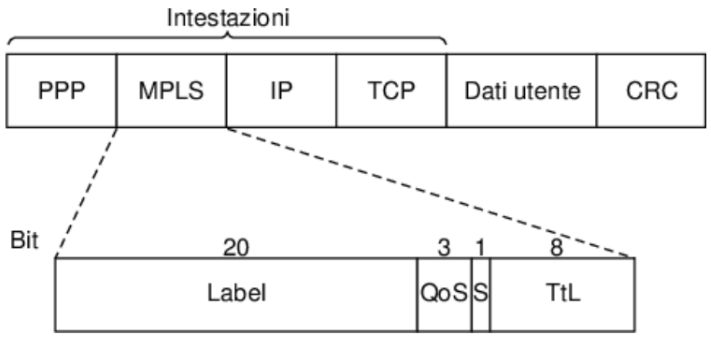

Figura 2.9: Pacchetto MPLS

- il campo S che si riferisce allo stacking di più etichette;
- il campo TtL che indica quante volte il pacchetto deve essere ancora inoltrato, questo campo viene decrementato a ogni router e quando raggiunge lo 0 viene scartato, così facendo si impedisce la formazione di cicli infiniti nei casi di instabilità nel routing.

Quando un pacchetto potenziato da MPLS raggiunge un router LSR (label switching router), l'etichetta viene utilizzata come indice in una tabella per determinare la linea di trasmissione e la nuova etichetta da utilizzare; questo scambio è utilizzato in tutte le reti a circuito virtuali. Per essere distinguibili una volta arrivate a destinazione, le etichette devono essere riassegnate a ogni salto. Dato che molti host e router non comprendono MPLS, quando un pacchetto IP raggiunge il bordo di una rete MPLS, il LER (label edge router) ispeziona l'indirizzo IP di destinazione e gli altri campi, così facendo è in grado di capire quale percorso MPLS dovrebbe seguire il pacchetto e di conseguenza attaccare la giusta etichetta.

Una volta che il pacchetto raggiunge l'altro capo della rete MPLS, l'etichetta termina il suo compito e viene rimossa (Fig. 2.10).

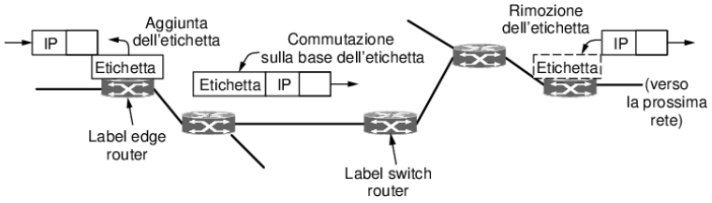

Figura 2.10: Ciclo etichetta MPLS

Le etichette possono sovrapporsi creando una stack di etichette che vengono rimosse mano a mano che raggiungono la loro destinazione ed ovviamente quella applicata più recentemente guida il pacchetto; il bit S nell'intestazione permette ad un router che rimuove un etichetta di sapere se dietro ce ne sono altre. MPLS è in grado di raggruppare i flussi sotto una singola etichetta ed appartengono così alla stessa FEC (forwarding equivalence class), così facendo questa classe copre oltre alla destinazione dei pacchetti anche la loro classe di servizio, perchè tutti i pacchetti di un flusso sono trattati nello stesso modo durante l'inoltro. MPLS risulta utile quando esiste più di un percorso per raggiungere la destinazione, può smistare il traffico in questi percorsi piuttosto che prendere il percorso più breve; può anche essere utilizzato per reindirizzare il traffico su un percorso alternativo precedentemente calcolato in risposta ad un malfunzionamento di un collegamento. Un altro suo utilizzo è quello di implementare le VPN, utile ad un ISP, in modo da utilizzare la rete MPLS per connettere tra loro le reti dei clienti, in modo da isolare le risorse e l'indirizzamento delle VPN dei clienti da quelle degli altri utenti che utilizzano la rete del provider, in questo caso si usano i tunnel MPLS chiamati VPLS. Quando due router LSR stabiliscono una connessione tra di loro utilizzano un protocollo chiamato LDP, che consiste in una serie di procedure e messaggi attraverso le quali i router stabiliscono il percorso attraverso una rete, mappando le informazioni di routing a livello di rete direttamente al livello data-link.

2.7.1 VPLS

VPLS (Virtual Private LAN service) è un metodo per creare tunnel trasparenti basati su MPLS aggiungendo un'etichetta VPLS nei pacchetti MPLS. Un tunnel VPLS si presenta come un'interfaccia separata del router. Un tunnel nella rete serve a collegare due router tra di loro che non sono direttamente connessi, in modo che i pacchetti non vadano in giro per la rete o per motivi di protocollo, ad esempio due router IPv6 che devono comunicare tra di loro ma che in mezzo c'è una rete IPv4 allora tra di loro si crea un tunnel dove il pacchetto IPv6 viene incapsulato in un pacchetto IPv4 e attraversa così la rete fino ad arrivare a destinazione (Fig. 2.11). Uno svantaggio del tunneling

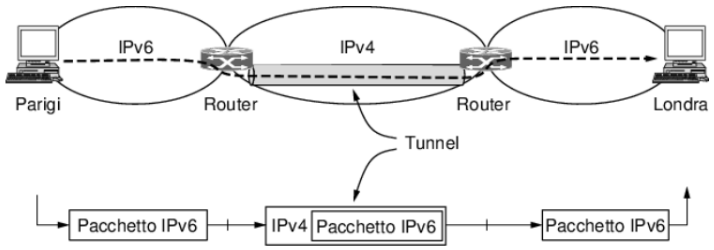

Figura 2.11: Tunnel

è che nessuno degli host della rete su cui viene fatto è raggiungibile, perchè i pacchetti non possono uscire durante l'attraversamento del tunnel, ma questa limitazione è un vantaggio per le VPN (Virtual Private Network) perchè i dati non viaggiano per le rete con il rischio di essere intercettati, in questo modo un tunnel aumenta anche la sicurezza delle VPN; nel nostro caso l'azienda userà questi tunnel per gestire il traffico del protocollo PPPoE.

2.7.2 PPPoE

Il protocollo PPPoE (Point to Point Protocol over Ethernet) fornisce dei benefici per gestire i clienti, per amministrare la rete e un sistema di gestione account per gli ISP e gli amministratori di rete. PPPoE è un estensione dello standard PPP (Point to Point Protocol); le differenze tra di loro sono nel metodo di trasporto dove PPPoE impiega Ethernet invece di collegamenti

con modem seriali, e questo permette di gestire gli accessi ad internet tramite username e password. I client ed i server PPPoE lavorano su ogni interfaccia Ethernet Layer2 nel router (Wireless 802.11, Ethernet 10/100/1000 Mbit/s, radiolan ed EoIP). PPPoE ha due step:

- Scoperta: un client scopre tutti gli accessi disponibili ai concentratori PPPoE e ne seleziona uno per stabilire una sessione PPPoE. Per effettuare questo passaggio PPPoE usa dei frame Ethernet speciali con il suo Tipo di frame Ethernet 0x8863. Questo step a sua volta ha quattro passaggi:
  - Inizializzazione: il client PPPoE invia un frame PADI (PPPoE Active Discovery Initialization) all'indirizzo Ethernet di broadcast e opzionalmente può specificare un nome del servizio.
  - Offerta: quando un server riceve un frame PADI, risponde inviando un frame PADO (PPPoE Active Discovery Offer) all'indirizzo Ethernet del client in modalità unicast. Se ci sono più server nel range del client, prenderà tutti i frame PADO e ne sceglierà uno per iniziare la sessione.
  - Richiesta: il client invia un frame PADR (PPPoE Active Discovery Request) all'indirizzo del server scelto in unicast, e se il server acconsente di stabilire una sessione con il client, inizierà una sessione PPP ed assegnerà un Session ID.
  - Conferma della sessione: il server invia un frame PADS (PPPoE Active Discovery Session confirmation) al client che una volta ricevuto, conoscerà l'indirizzo mac del server ed il Session ID.

- Sessione: una volta finito lo step della scoperta, entrambi i peers conosceranno l'ID della Sessione e l'indirizzo MAC dell'altro. I frame PPP sono incapsulati in fram PPPoE con tipo di frame 0x8864.
  La sessione consiste in una negoziazione LCP, un'autenticazione e una negoziazione IPCP dove viene assegnato un indirizzo IP al client.

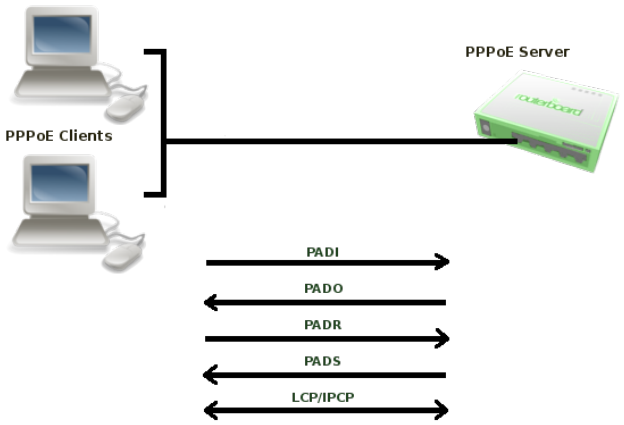

Figura 2.12: PPPoE scoperta

Quando una sessione viene terminata o dal client o dal server viene inviato un frame PADT (PPPoE Active Discovery Terminate).

Capitolo 3

Laboratorio

3.1 Simulazione in Laboratorio

Prima di implementare l'algoritmo nella rete, sono state fatte delle prove in laboratorio sia in virtuale, usando il software di simulazione di reti GNS3 (Fig. 3.1), sia fisicamente prendendo 5 router mikrotik. Verificati i test e la funzionalità dell'implementazione si è passati a fare una pianificazione degli indirizzi IP da usare in una parte della rete scelta insieme all'azienda, riproducendo, prima di passare all'implementazione, quella parte di rete in virtuale usando sempre il software GNS3 e delle macchine virtuali gestite tramite il software Virtualbox (Fig. 3.2), fornite da GNS3 ed in seguito è stato caricato, dentro le macchine virtuali, il routerOS 6.44.2 CHR, che corrisponde all'ultima versione per router di test per simulare un reale router MikroTik fisico con licenza; grazie a questa macchina virtuale si possono creare "infiniti" router (dipendendo dalle caratteristiche tecniche della macchina su cui viene installato).

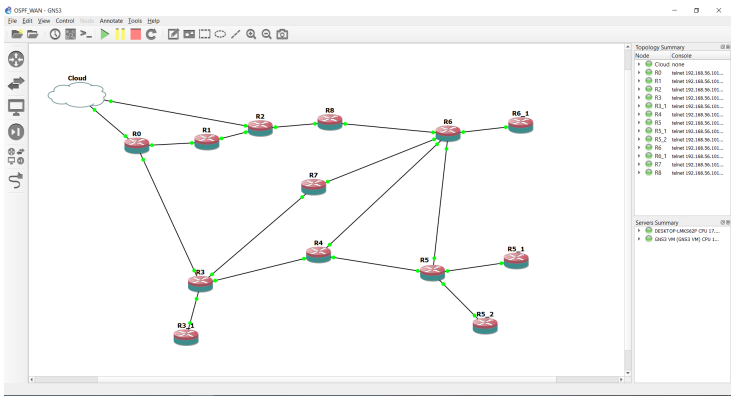

Figura 3.1: Esempio GNS3

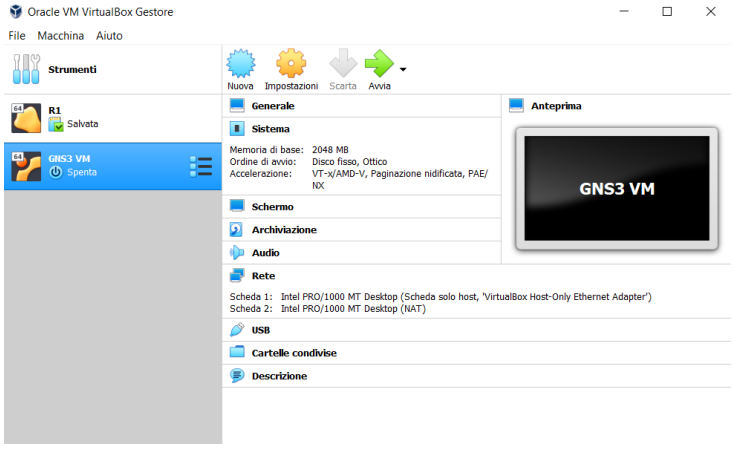

Figura 3.2: Virtualbox

3.1.1 Creazione rete virtuale e fisica

Come prima cosa è stato creato un ambiente di rete virtuale usando il software GNS3 con 5 router mikrotik connessi tra di loro come mostrato in figura 3.3.

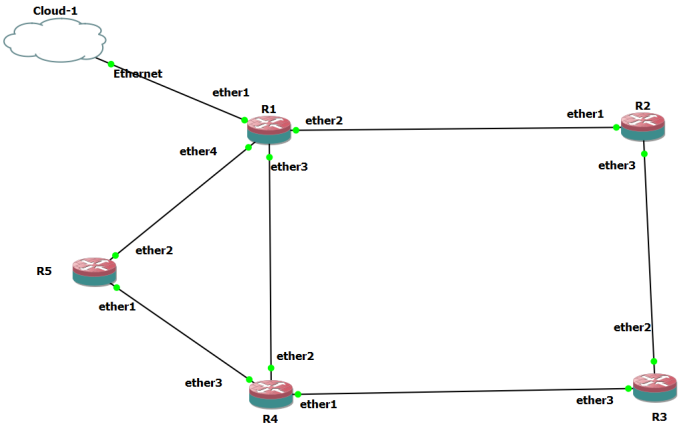

Figura 3.3: Test con 5 router

I router sono stati configurati in modo che ognuno di essi avesse:

- ogni interfaccia utilizzata con un indirizzo IP con subnet mask /29; si è scelta questa sotto rete perchè nella rete aziendale tra un router ed un altro possono esserci dispositivi come antenne o ripetitori che necessitano anche loro di indirizzi IP.

```
ip address add address=192.0.2.3 netmask
    =255.255.255.240 interface=eth1
```

- un indirizzo IP di loopback con subnet mask /32.

```
ip address add address=192.0.2.0
netmask=255.255.255.255 interface=loopback
```

- 3 tipi di bridge:
  - un bridge di loopback

```
interface bridge add name=loopback
```

  - un bridge per gli indirizzi statici

```
interface bridge add name=statica1
```

  - un bridge per le antenne di diffusione

```
interface bridge add name=diffusioni
```

- un DHCP server: per assegnare gli indirizzi alle antenne che sono montate sulle case dei clienti.

```
/ip pool
add name=dhcp ranges=192.0.2.2-192.0.2.49
/ip dhcp-server
add address-pool=dhcp disabled=no interface
   =bridge1 name=dhcp1 lease-time=1d
/ip dhcp-server network
add address=192.0.2.0/24 dns-server
   =192.0.2.254 gateway=192.0.2.254
```

- implementato l'algoritmo OSPF
- riservate le porte di bridge dalla ethernet 6 alla 9 comprese, per le 4 antenne di diffusione, che in genere sono montate sui loro ripetitori in modo da coprire tutte le direzioni, quindi in genere montate direzionate ognuna verso un punto cardinale Nord,Sud,Ovest ed Est.
- un indirizzo IP per le diffusioni con subnet mask /24, usato per il DHCP server

- un indirizzo IP per gli indirizzi statici con subnet mask /29; questi indirizzi vengono usati dall'azienda per monitorare dei sensori di controllo, per creare delle reti VLAN e per degli indirizzi riservati per eventuali necessità dell'azienda.

Inoltre è stato anche attivato il servizio MikroTik chiamato RoMON (Router Management Overlay Network), che permette di creare un discovery della rete utilizzando un "Peer Mac discovery" che lavora sia nel Layer 2, sia nel Layer 3 (attraverso un data forwarding). Quando RoMON è abilitato i pacchetti non vengono visualizzati da strumenti per lo sniffing. Per configurarlo è stato immesso il seguente comando:

```
/tool romon set enabled=yes
```

Così facendo è possibile accedere agli router della rete su cui è attivo RoMON senza essere collegati fisicamente ad essi.

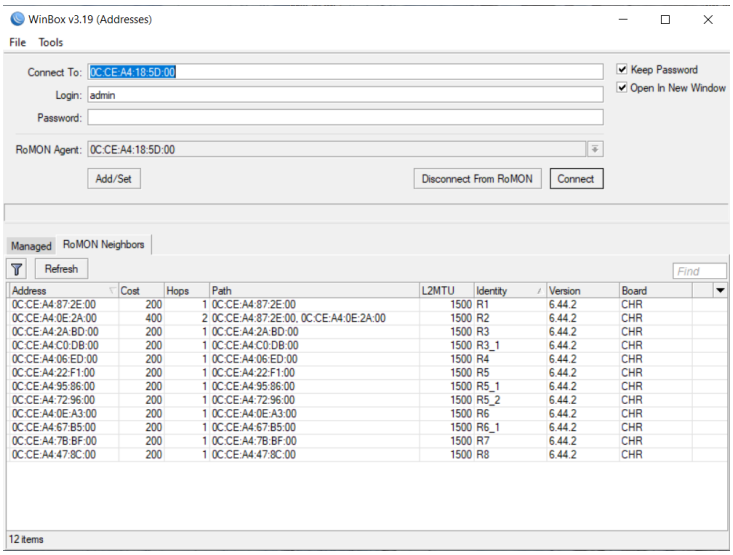

Figura 3.4: RoMON

Queste procedure sono poi state eseguite sui router fisici per un ulteriore prova, mantenendo comunque la stessa topologia delle interfacce.

3.1.2 Implementazione dell'algoritmo in area virtuale

Dopo aver fatto le configurazioni di base ai router è stato implementato l'algoritmo OSPF con le seguenti impostazioni:

- Router ID: utilizzato nella rete OSPF per identificare i vari router, che nel nostro caso corrisponde all'indirizzo di loopback.

```
/routing ospf instance
set [ find default=yes ] router-id=192.0.2.0
```

- Le interfacce con tipo connessione Point to Point e autenticazione su di esse di tipo MD5: come visto precedentemente, è il tipo di sicurezza più alta che si può implementare con questo tipo di tecnologia.

```
/routing ospf interface
   add authentication=md5
   authentication-key=prova123
   network-type=point-to-point
```

- Aggiunte due network relative all'indirizzo di loopback e all'indirizzo delle interfacce e inserite nelle rispettive aree OSPF.

```
/routing ospf network
add area=backbone network=192.0.2.0/24
add area=backbone network=192.0.2.0/24
```

Effettuati questi passaggi sono stati verificati i seguenti punti:

- Che i router configurati vedessero i propri vicini (neighbors) come in figura 3.5.
- Che ogni router avesse i percorsi per tutti i router della rete OSPF (Fig. 3.6).
- Il tempo di risposta a dei guasti simulati e il comportamento delle routes dopo il ripristino del guasto.

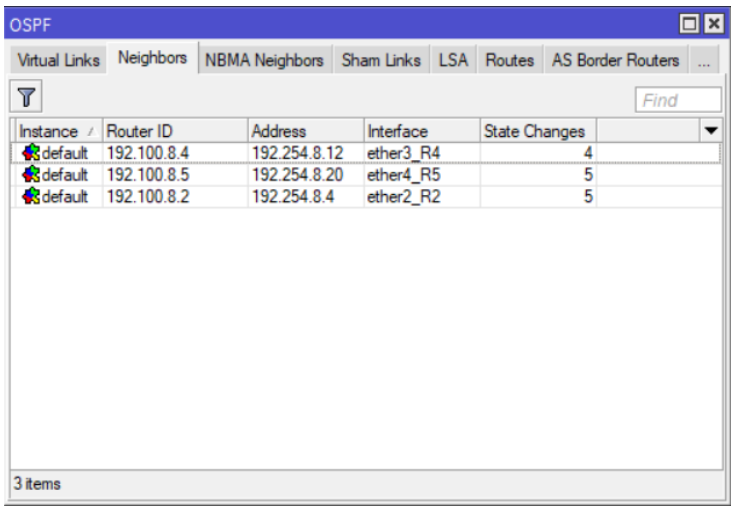

Figura 3.5: OSPF Neighbour

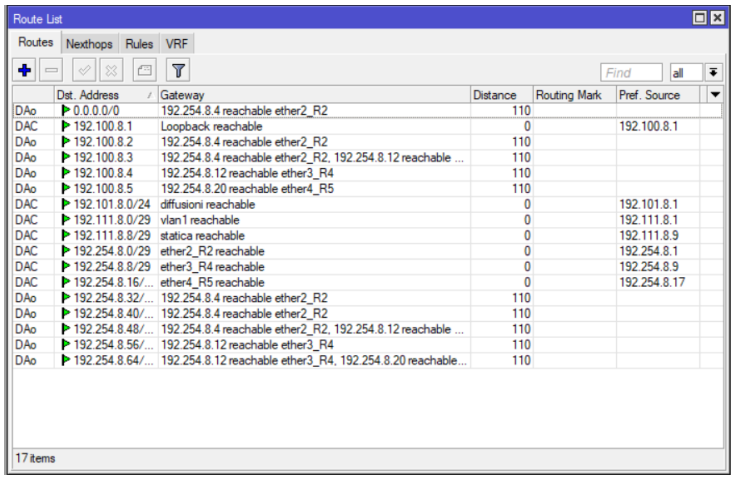

Figura 3.6: Route

Per quest'ultimo punto quello che è stato fatto in concreto è verificare dove passasse la rete per un determinato router R5, pingare l'indirizzo IP di loopback di quel router R5 e poi simulare un problema troncando la linea che collegava R1 ad R5. Si è notata una perdita di pacchetti prima che il router R1 si accorgesse del problema nel percorso più breve conosciuto da lui, nel momento che nota questo problema immediatamente cambia le sue tabelle dei percorsi più brevi verso i router della sua rete OSPF e i pacchetti ritornano ad arrivare a destinazione. Nel momento poi che il "guasto" alla rete viene sistemato, R1 cambia di nuovo la sua tabella dei percorsi, stavolta però senza perdere nessun pacchetto. (Fig. 3.7)

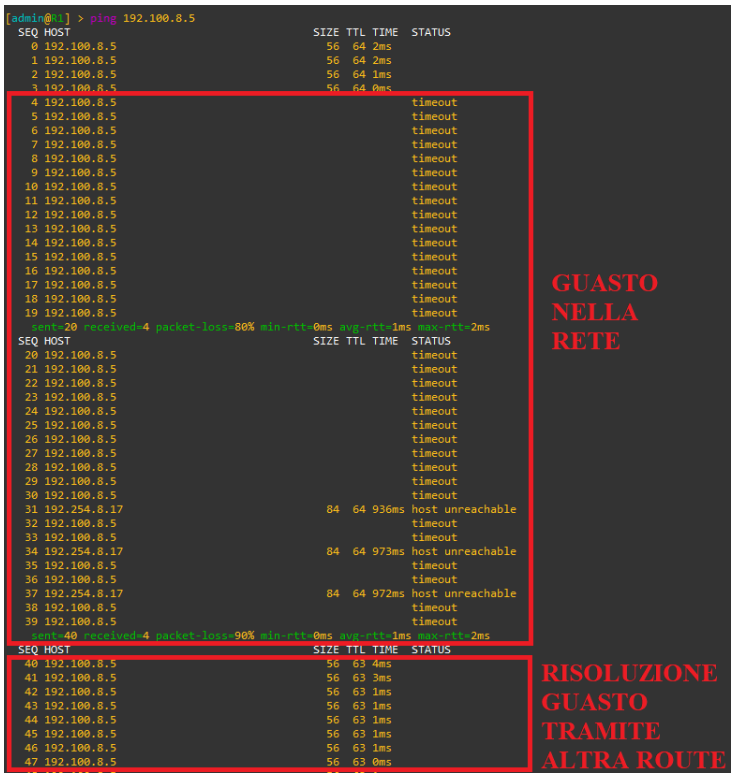

Figura 3.7: Simulazione guasto

3.2 Dal planning IP all'implementazione sulla rete

3.2.1 Scelta dell'area di rete

La scelta dell'area è stata fatta in modo da creare per prima cosa una parte dell'area di backbone nella rete, fondamentale per il corretto funzionamento di OSPF, scegliendo quindi i router principali della rete che gestiscono il traffico fino alle aree più piccole, ed inoltre sono state selezionate anche tre aree non di backbone che sono state impostate come Totally Stub e collegate ai router dell'area di backbone selezionata. Questa parte della rete è solo una piccola area che, se il progetto risponde bene alle esigenze dell'azienda, verrà mano a mano ingrandita con gli stessi criteri usati per questa porzione di rete.

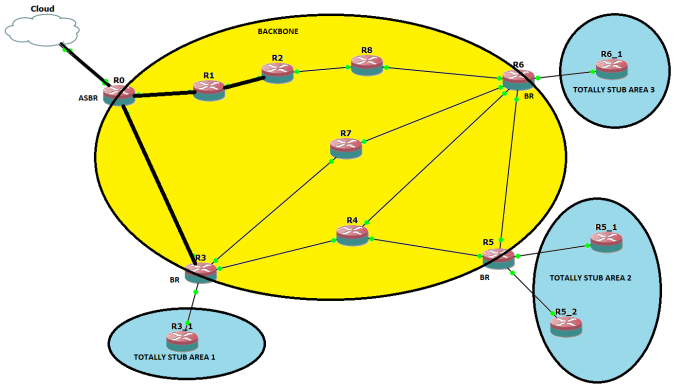

Figura 3.8: Area della rete scelta

3.2.2 Planning IP

La pianificazione degli indirizzi IP è stata fatta in modo da non dover, in futuro, rimettere mano al lavoro che è stato fatto. Sono stati presi degli indirizzi non utilizzati dall'azienda e divisi per aree OSPF in cui il router si trova, si è utilizzato il criterio di identificare queste aree tramite il terzo ottetto degli indirizzi IP usati; sono stati riservati gli indirizzi IP da x.x.1.x a

x.x.10.x per l'area di backbone ed invece per le stub aree gli indirizzi maggiori di x.x.10.x

3.2.3 Implementazione algoritmo e test

Per i router di backbone la configurazione è per tutti simile a quella fatta nei cinque router tranne per il router ASBR (router di confine del sistema), i tre router BR (router di confine con le stub) e per i router presenti nelle stub. Qui di seguito verranno elencati le configurazioni principali che ogni tipologia di router ha di differente dagli altri.
ASBR: questo è stato il primo router ad essere configurato perché è quello che fa da tramite tra i router da cui arriva la linea internet e i router delle rete di proprietà della Newtec. Con i seguenti comandi il router fa in modo di assegnare un id al router (corrispondente all'indirizzo di loopback), distribuire il suo percorso di default con l'impostazione "always-as-type-1", il che significa che nei costi dei percorsi viene incluso anche quello per uscire dalla rete; ed infine vengono ridistribuite anche le rete locali connesse ed inserite nel protocollo di routing.

```
/routing ospf instance
set [ find default=yes ] router-id=192.0.2.0
set distribute-default=always-as-type-1
set redistribute-connected=as-type-1
```

Router di backbone: questi sono i router che si trovano esclusivamente all'interno dell'area di backbone e non sono collegati con altre aree. Per questi router la configurazione segue quella standard, con l'unica eccezione di ridistribuire le reti locali con il comando:

```
/routing ospf instance
set redistribute-connected=as-type-1
```

Router BR: questi sono i router che fanno parte della backbone ma che sono collegati con altre aree, nel nostro caso delle totally stub. Questi BR hanno la particolarità di appartenere a più aree, nel nostro caso, quella di

backbone e quella totally stub; di conseguenza le impostazioni di routing OSPF devono essere aggiornate per questa tipologia di router con i seguenti comandi:

```
/routing ospf network
add area=backbone network="network loopback"
add area=backbone network="network ptp della backbone"
add area=stub1 network="network ptp BR e router stub"
```

Router totally stub: questi sono i router presenti nelle aree totally stub e sono connessi con i router BR. Loro appartengono esclusivamente alla loro area che in questo caso è totally stub, che come detto precedentemente, non fa passare i messaggi LSA di tipo 3 e 5. Per impostarli come router stub sono stati eseguiti i seguenti comandi:

```
/routing ospf area add area-id=0.0.0.1 name=stub1 type=
   stub inject-summary-lsas=no
/routing ospf network
add area=stub1 network="network loopback"
add area=stub1 network="network statiche"
add area=stub1 network="network ptp della stub"
add area=stub1 network="network DHCP"
```

Implementate queste configurazioni sono state fatte delle prove analoghe a quelle fatte nei 5 router, verificando inoltre anche il comportamento delle 3 aree stub e dei relativi border router, verificandone i percorsi conosciuti, rispetto a quelli della backbone (Fig. 3.9).

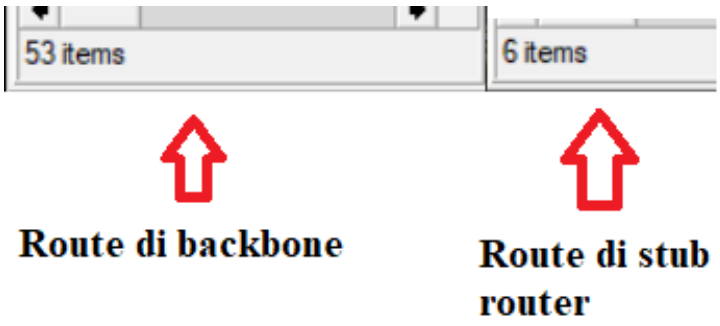

Figura 3.9: Differenza route conosciute tra router di backbone e router totally stub

3.3 MPLS over OSPF

Per permettere ai tunneling VPLS di funzionare correttamente, nei router deve essere configurato correttamente il protocollo MPLS. Per farlo è stato attivato il protocollo impostando nella configurazione LDP, come lsr-id e transport-address, l'indirizzo ip di loopback, con i seguenti comandi:

```
/mpls ldp
set enabled=yes lsr-id=192.0.2.0 transport-address
    =192.0.2.0
```

In seguito sono state inserite le interfacce con cui ogni router è collegato con quelli adiacenti, in questo modo:

```
/mpls ldp interface
add interface=ether1
add interface=ether2
...
```

Dopo aver configurato correttamente MPLS in tutti i router della rete controllando se i router avessero tutti i router adiacenti come LDP neighbors; è stato attivato VPLS su tutti i router. A seconda del router ci sono due principali configurazioni: quella dei router non concentratori e quella del router concentratore. Per i router non concentratori è stata inserita una vpls impostando come remote peer l'indirizzo IP di loopback del router concentratore, e un VPLS ID univoco utilizzato per connettere il router a quello concentratore.

```
/interface vpls
add name=prova remote-peer=1.1.1.1 vpls-id=0:1 disabled
    =no
```

Inoltre sui router delle totally stub area è stato aggiunto come percorso l'indirizzo di loopback del router concentratore, altrimenti non riuscivano a trovare la destinazione. Per il router concentratore la configurazione è simile, ma devono essere create tante interfacce VPLS quanti sono i router su cui si

vuole configurare il tunneling, dove il remote peer corrisponde all'indirizzo IP di loopback del router non concentratore, e il VPLS id deve corrispondere a quello impostato nell'interfaccia VPLS del router non concentratore.

```
/interface vpls
add name=1 remote-peer=1.1.1.1 vpls-id=0:1 disabled=no
add name=2 remote-peer=2.2.2.2 vpls-id=0:2 disabled=no
...
```

Inoltre il router concentratore ha un bridge vpls, e ad esso sono collegate tutte le interfacce VPLS con horizon settato ad 1, così facendo i tunnel VPLS rimangono separati impedendo ai pacchetti di mischiarsi tra di loro.

```
/interface bridge add name=bridge-vpls
/interface bridge port add bridge=bridge-vpls interface
   =vpls-1 horizon=1
/interface bridge port add bridge=bridge-vpls interface
   =vpls-2 horizon=1
...
```

In questo modo ora nella rete è stato configurato il tunnelling VPLS, e per verificarlo basta vedere nelle vpls del concentratore se accanto alle varie vpls se risulta l'etichetta RS, che sta per Running (funzionante) e slave (questo solo per il concentratore vedi i client come slave), e se c'è trasmissione di dati sia in trasferimento (Tx) sia in ricezione (Rx) (Fig. 3.10)

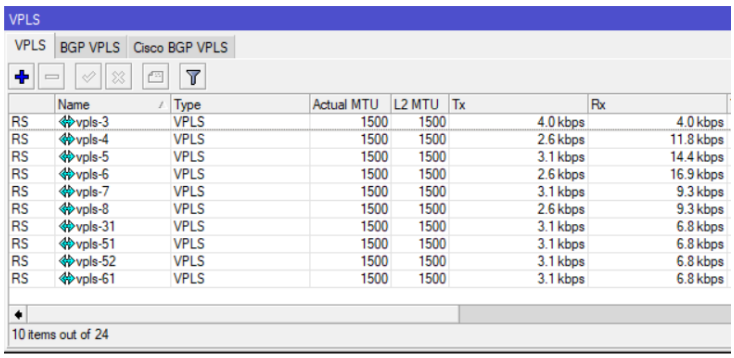

Figura 3.10: Corretto funzionamento VPLS

3.3.1 PPPoE

Dopo aver configurato correttamente VPLS si passa alla configurazione del server PPPoE. Tramite i seguenti comandi viene:

- configurato un server PPPoE a cui viene assegnato un nome, un'interfaccia che nel nostro caso è il bridge VPLS, e vengono settati un MTU (Maximum Trasmission Unit) ed un MRU (Maximum Receive Unit).

```
interface pppoe-server server add service-name
    =service2 interface=bridge-VPLS max-mtu
    =1500 max-mru=1500
```

- configurato un profilo PPP a cui viene assegnato come local-address, l'indirizzo di loopback del server

```
ppp profile set default local-address
    =192.0.2.128
```

- impostato che deve esserci un sistema di account, che deve essere gestito tramite RADIUS ed inoltre impostato il tempo di aggiornamento della connessione attiva a cinque minuti

```
ppp aaa set accounting=yes use-radius=yes
    interim-update=5m
```

In seguito vengono configurati i PPPoE client direttamente nei router dei clienti con questo script:

```
/interface pppoe-client
add ac-name="" add-default-route=yes allow=pap,chap,
   mschap1,mschap2 \
    default-route-distance=1 dial-on-demand=no disabled
       =no interface=ether1 \
    keepalive-timeout=60 max-mru=1480 max-mtu=1480 mrru
       =disabled name=Gateway \
    password=prova profile=default service-name="" use-
       peer-dns=yes user=\
    prova
```

In cui vengono impostate le seguenti opzioni:

- Il nome del concentratore di accesso che viene lasciato vuoto in modo che il client si connetterà a qualsiasi concentratore.
- Impostato automaticamente un percorso di default.
- Permessi i vari metodi di autenticazione in questo caso tutti.
- Impostato il valore di distanza verso il percorso di default ad 1.
- L'interfaccia Ethernet su cui il client è impostato, nel nostro caso Ethernet1.
- Il nome del servizio PPPoE.
- Nome utente e password che verranno poi autenticati dal PPPoE server tramite RADIUS.

Ora questa parte della rete è interamente configurata con OSPF, tunneling VPLS e il protocollo PPPoE per permettere ai clienti di connettersi ad internet.

Capitolo 4

Conclusioni

Il lavoro effettuato ha dimostrato che l'algoritmo sarà in grado di espandere una parte dell'attuale rete WIRTEK permettendo di aggiornare la stessa, gestendo in maniera più veloce ed ottimale la rete in presenza sia di guasti che di problemi. Si implementerà anche la tecnologia MPLS, creando dei tunnel VPLS che andranno a migliorare la gestione delle reti private separandole dal resto della rete, quindi aumentandone in generale anche la sicurezza. Questa parte della rete è solo una piccola area che, se il progetto risponde bene alle esigenze dell'azienda, verrà mano a mano ingrandita con gli stessi criteri usati per questa porzione di rete.

Bibliografia

[1] Andrew S. Tanenbaum, David J.Wetherall, *Reti di calcolatori*, (2015)

[2] John T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*

[3] Rob Cameron, Chris Cantrell, Anne Hemni, Lisa Lorenzin, *Configuring Juniper Networks NetScreen and SSG Firewalls*

[4] James F. Kurose, Keith W. Ross, *Reti di Calcolatori e Internet*, (2017)

[5] J.Moy (1991), *OSPF v2 RFC 1247*

[6] J.Moy (1998), *OSPF v2 RFC 2328*

[7] P. Murphy (2003), *The OSPF Not-So-Stubby Area (NSSA) Option RFC 3101*

[8] M. Lasserre, V. Kompella, (2007) *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling RFC 4762*

[9] L. Andersson, I. Minei, B. Thomas, (2007) *LDP Specification RFC 5036*

[10] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, R. Wheeler (1999) *A Method for Transmitting PPP Over Ethernet (PPPoE) RFC 2516*

[11] J. Hawkinson, T. Bates (1996) *Guidelines for creation, selection, and registration of an Autonomous System (AS) RFC 1930*

[12] Y. Rekhter, T. Li, S. Hares, (2006) *A Border Gateway Protocol 4 (BGP-4) RFC 4271*