

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA

CAMPUS DI CESENA

SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA,
INFORMATICA

E TELECOMUNICAZIONI

SPERIMENTAZIONE DI PROTOCOLLI DI ROUTING SU TOPOLOGIE DI RETE IBRIDE

ELABORATO IN

APPLICAZIONI E TECNICHE DI

TELECOMUNICAZIONI

RELATORE

CERRONI WALTER

PRESENTATO DA

FABIO DRADI

CORRELATORE

CALLEGATI FRANCO

I SESSIONE DI LAUREA

ANNO ACCADEMICO 2016-2017

1. Introduzione	4
2. Instradamento delle informazioni	5
2.1 Teoria dell'instradamento	5
2.2 Protocollo OSPF	6
3. Strumenti software e hardware utilizzati	7
3.1 Raspberry Pi	7
3.2 Router Cisco 1801	9
3.3 Switch HP Procurve 2524	12
3.4 Minicom	15
3.5 DHCP	16
3.6 Wireshark	17
4. Creazione nuova network	18
4.1 Confronto tra vecchia e nuova topologia	18
4.1.1 Modifica Raspberry R1.1	20
4.2 Configurazione Router Cisco 1801	23
4.2.1 Accedere al dispositivo	23
4.2.2 Configurazione interfacce	25
4.2.3 Configurazione OSPF nel router Cisco	29
4.2.4 Connessione tra Vlan Router e Raspberry	30
4.3 Configurazione Switch HP Procurve 2524	33
4.3.1 Accedere al dispositivo: configurazione DHCP	33
4.3.2 Configurazione interfacce	37
4.3.3. Connessione tra Vlan Switch e Raspberry	39

4.4 Routing tables e test connettività	43
5. Conclusioni	56
6. Appendice	57
6.1 ISO/OSI	57
6.2 Pacchetto Quagga	59
7. Bibliografia e ringraziamenti	60

1. Introduzione

Al giorno d'oggi la capillarità della rete è uno degli aspetti fondamentali di cui bisogna tenere conto nell'analisi di un progetto topologico. I dispositivi con cui connettersi e operare su rete sono innumerevoli, e quindi diventa indispensabile un valido interfacciamento tra di essi.

Per attuare questo sono necessari diversi protocolli di routing a seconda del progetto e del dispositivo con cui si andrà a gestire il traffico di informazioni previsto.

L'obiettivo di questa tesi, è quello di interfacciare dispositivi di nuova generazione, come i Raspberry Pi, con alcuni di vecchia generazione, come Router Cisco e Switch HP Procurve, e verificare come esso sia possibile grazie ai protocolli di routing supportati da ogni apparato di rete.

Nascono così nuove topologie di rete, denominate "ibride". Con questo termine si intendono reti costituite in parte in modo fisico, e in parte in modo virtuale. La porzione di rete fisica sarà rappresentata dallo switch, dal router, e dai Raspberry, mentre la parte virtuale è generata all'interno dei Raspberry grazie all'uso del tool mininet. Questo software permetterà infatti di creare switch, router e lan virtuali, semplicemente attraverso l'utilizzo di alcuni comandi e di protocolli come OSPF e Zebra.

Si andrà a creare quindi una rete che in cui alla fine saranno presenti diverse interazioni e quindi interfacciamenti tra i mezzi fisici, presenti in quasi tutte le tipologie di rete, e mezzi virtuali, creati con il software nei Raspberry.

Si è deciso quindi di sfruttare la topologia creata dall'ing. Nicola Sparnacci, progettata unicamente con Raspberry Pi, e di sostituire:

- Il primo Raspberry che aveva il compito di Router di Backbone dell'area 0, ossia quella di collegamento tra le 2 aree principali. Al suo posto è stato inserito un Router Cisco 1801 che andrà programmato sfruttando le Vlan interne e i protocolli di routing OSPF.
- Lo switch virtuale che veniva creato dal Raspberry 2.3 con uno switch fisico, che andrà anch'esso configurato tramite Vlan per gestire il traffico proveniente da R2.1 e R2.2.

Questa tesi è divisa in tre capitoli principali, in cui verranno trattati principalmente i temi legati alla teoria delle reti, spiegati e descritti gli strumenti utilizzati e la creazione e sostituzione degli elementi previsti all'interno della nuova network realizzata.

2. Instradamento delle informazioni

2.1 Teoria dell'instradamento

L'instradamento è l'operazione di commutazione nell'ambito delle reti di telecomunicazioni, con cui si decide su quale porta o interfaccia inviare un flusso di dati.

Per fare questo solitamente si utilizza una tabella di indirizzamento, in cui sono presenti tutti i nodi necessari a mettere in comunicazione la sorgente con il destinatario del pacchetto di dati. Un commutatore deve quindi essere in grado di gestire il traffico offerto in ingresso e deve essere dimensionato opportunamente.

Si può differenziare la gestione del traffico di instradamento tra switching e routing. La differenza tra i due modi di gestire il traffico di dati verrà approfondita nei paragrafi successivi.

2.2 Protocollo OSPF

Il protocollo *Open Shortest Path First (OSPF)* è considerato uno dei **protocolli di routing di tipo Link State** più utilizzati su reti IP, ed è stato introdotto per **gestire una grande quantità di apparati** suddivisi in *Autonomous System*, che sono per l'appunto un gruppo di router e reti sotto il controllo di un unico dispositivo. All'interno di un sistema ogni router comunica attraverso un protocollo *IGP* (Interior Gateway Protocol), mentre lo scambio di dati tra router di aree diversi avviene attraverso il protocollo *BGP* (Border Gateway Protocol).

Degli IGP fanno parte i protocolli RIP e OSPF. Il Routing Information Protocol (RIP) è stato l'antenato dell'OSPF. E' un distance-vector protocol e viene impiegato nelle reti di piccola dimensione, attraverso un algoritmo di conteggio di hop che indica la quantità di subnet attraversate per raggiungere il destinatario del pacchetto. La sua evoluzione è l'OSPF dove ogni router all'interno dell'area si crea il grafo della topologia, e utilizza l'algoritmo Dijkstra per trovare il percorso con il minor costo possibile. Un sistema autonomo OSPF può essere configurato in aree gerarchiche e vi è quindi una divisione, in base ai tipi di router che sono presenti all'interno di esse. Per esempio un Area Border Router è un router che interconnette una o più aree OSPF all'area di *BackBone*.

Un'altra area fondamentale è infatti quella di *Backbone*, che è quella del collegamento di base in una rete OSPF. Nel progetto ideato nella tesi, il router di Backbone sarà il router Cisco, che infatti agirà da intermediario tra due aree principali.

3. Strumenti software e hardware utilizzati

3.1 Raspberry Pi

E' stato il **dispositivo base** con cui si è potuta costruire la topologia di rete prefissata. Si è scelto di utilizzare il Raspberry Pi per via delle dimensioni ridotte, che permettono di realizzare reti più o meno complesse ad un costo accessibile. Per questi motivi il Raspberry Pi si rivela un dispositivo molto interessante, che infatti sta venendo implementato a scopo didattico in molti istituti informatici e atenei. Si presenta come un calcolatore su singola board (SoC), ed è stato creato dalla fondazione inglese Raspberry Pi Foundation. Il modello più completo è il Model B+, basato su architettura ARM e con le seguenti caratteristiche tecniche:

- CPU: 700 MHz (con possibilità di OverClock) ARM 1176JZF.S core (famiglia ARM11);
- GPU: Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder;
- Memory (SDRAM): 256 Megabytes (MiB)

E interfaccie Input/Output:

- Video outputs: RCA composito, HDMI;
- Audio outputs: 3.5mm jack, HDMI;
- Slot di memoria microSD;
- Porta Ethernet 10/100RJ45
- 4 porte USB 2.0 hotplug e comportamento di overcurrent;
- 40 GPIO pins.



Molto importanti sono anche le sue dimensioni. Infatti, un Raspberry misura 85.60mm x 56mm x 21mm (che equivalgono circa alla grandezza di una carta di credito) e ha un peso di 45g. Queste caratteristiche ne consentono l'utilizzo anche in situazioni in cui le dimensioni e il peso richiesti siano ridotti senza avere pericoli di rotture poiché è possibile acquistare o creare un case protettivo per ripararlo da urti e agenti esterni.

Nel Febbraio 2015 è stato rilasciato un aggiornamento al modello B+ portando la CPU ad una frequenza di 900 MHz grazie al quad-core ARM Cortex-A7 e arrivando a 1 Gb di memoria RAM statica.

Questo dispositivo per il basso costo, per le piccole dimensioni e per la vasta gamma di possibili utilizzi, rappresenta un punto fermo nei sistemi embedded di nuova generazione ed è stato sostituito ai sistemi più costosi come, ad esempio, le FPGA.

Il punto di forza del sistema Raspberry Pi è, ancor più del basso costo, la flessibilità. Questa è garantita dal sistema operativo Linux supportato che, in ogni sua variante o distribuzione, permette di sfruttare a pieno tutte le potenzialità di un calcolatore. Nella fattispecie, esistono distribuzioni ad-hoc come RaspBian, una versione adattata di Debian. Come si vedrà, la sintassi dei comandi è pressoché identica a quella di un personal computer con sistema operativo Debian.

3.2 Router Cisco 1801



Un router è un dispositivo di **interfacciamento** tra diverse sottoreti e che ha quindi il compito di instradare i dati sotto forma di pacchetti tra subnet diverse. Questo compito è svolto attraverso l'utilizzo del **livello 3** del modello ISO/OSI ossia il TCP/IP, il che lo differenzia dallo switch che instrada a livello locale solo sulla base degli indirizzi fisici (MAC). L'instradamento può avvenire sia in subnet che sono direttamente connesse al router, oppure verso altre sottoreti che hanno però gli indirizzi contenuti nelle tabelle di instradamento della macchina, e che verranno raggiunte attraverso diversi "next-hop" nella rete (indirizzamento indiretto). Queste tabelle denominate "Routing Tables", non hanno come indirizzi solamente quelli di altri apparati di rete, ma possiedono al loro interno anche interi sottoinsieme di reti (Subnet_ID).

Il routing quindi può essere classificato:

Per reti direttamente connesse: quando l'host di un'interfaccia di rete configurata con un indirizzo IP e una netmask, riesce a sapere automaticamente come raggiungere tutti gli host per quella sottorete.

Instradamento statico: in cui le rotte si possono configurare manualmente su ogni router.

Instradamento dinamico: quando le tabelle di instradamento vengono configurate con appositi protocolli di routing di modo che ogni router riesca a scambiarsi informazioni sulla topologia della rete e quindi si riesce a costruire la propria tabella di routing.

Ogni router si può scomporre in base alla funzione di ogni suo componente:

- **CPU:** che ha il compito di attuare il processo di forwarding dei pacchetti, di calcolare le tabelle di instradamento, aggiornando i dati di routing e di supervisionare alcune caratteristiche particolari del router ossia protocolli di gestione come SNMP o OSPF;
- **Memoria:** si suddivide in ROM nella quale è memorizzato il software base del router, includendo alcuni comandi di base tra cui anche quello di bootstrap, in VRAM, che è un tipo particolare di memoria non volatile in cui sono presenti i salvataggi dei file di configurazione del dispositivo, nella flash, in cui è memorizzato il sistema operativo e nella RAM, che è la memoria di lavoro, ed è volatile.
- **Interfacce di configurazione** come la console, un'interfaccia seriale asincrona (RS232) usata per il collegamento di un terminale seriale per la configurazione del router, oppure l'Aux, anch'essa un'interfaccia seriale asincrona, infatti è uguale alla porta console, ma si utilizza per collegare altre periferiche.
- **Interfacce di rete**, che possono essere AUI, 10BaseT, Seriali sincrone, BRI (verso NT di un ISDN basic rate) o ATM (fibra ottica).

Il **vantaggio di un dispositivo Cisco** rispetto ad altri sta nel sistema operativo, che è considerato molto potente e che è in grado di implementare tutti i comandi per le varie funzionalità in modo omogeneo. Sugli apparati Cisco il sistema operativo risiede nella memoria flash e acquisisce i comandi per la configurazione tramite CLI.

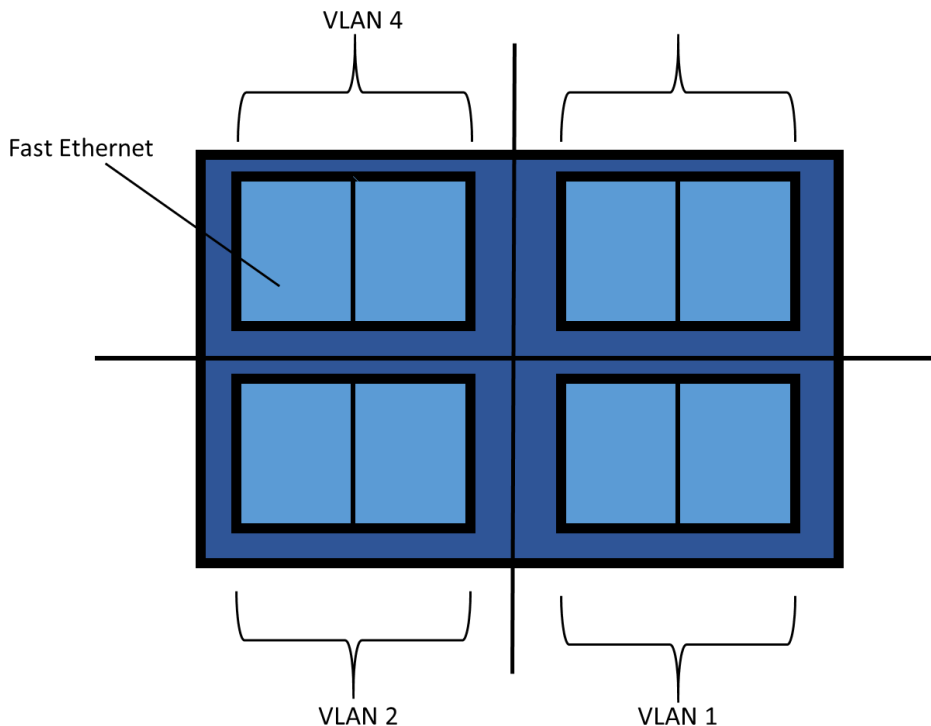
Dal **punto di vista fisico** i router possono essere dei computer con l'unico scopo di indirizzamento/instradamento di pacchetti, oppure possono essere dispositivi creati appositamente per questo utilizzo. In base al modello e al costo sono progettati in maniera diversa, e quelli di fascia più alta, sono costruiti in modo modulare, di modo che si possano aggiungere interfacce a seconda del tipo di rete che occorre.

L'**instradamento** in questi dispositivi, come negli switch, può avvenire in più modi :

- **Cut-through:** il router pensa solo a leggere l'indirizzo IP del destinatario e gli manda il contenuto del pacchetto dati in contemporanea alla lettura.

- **Store and forward:** il router memorizza il pacchetto IP prima di trasmetterlo, di modo che ci possa essere un adattamento della velocità del flusso di dati trasmesso in base ai pacchetti in coda in entrata. Questo meccanismo consente di evitare congestioni del traffico e perdite di dati e tempo inutili.

In questo dispositivo le interfacce fisiche di rete sono porte fast ethernet da 10/100 Mbit/s che supportano l'incapsulamento VLAN 802.1Q. Ogni Vlan è divisa a porte di 2 nel modo indicato nell'immagine sottostante.



Inoltre ogni router ha una porta WAN Fast Ethernet onboard e un supporto LAN senza fili opzionale 802.11a/b/g. Per la connettività WAN, il router Cisco 1801 dispone di una porta ADSL su POTS. I router Cisco 1801, Cisco 1802 e Cisco 1803 forniscono connessione a Internet protetta e connessione di backup tramite una porta ISDN S/T in caso di problemi alla connessione principale.

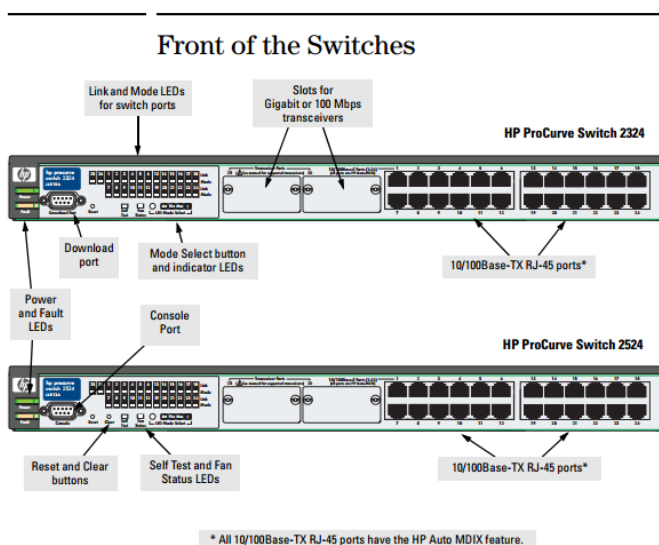
3.3 Switch HP Procurve 2524



E' l'apparato di rete ideale per **gestire il traffico a livello 2** (datalink) del modello ISO/OSI. Infatti permette l'instradamento nelle reti LAN **servendosi dell'indirizzo fisico** (MAC) presente all'interno di ogni dispositivo, che viene letto nei frame ricevuti, e poi interpretato di modo che il flusso di dati venga diretto solo verso il MAC di destinazione desiderato.

In una rete locale grazie ad esso, è possibile operare con più traffici di rete contemporaneamente: sarà sufficiente collegare i dispositivi operanti a porte diverse dello Switch, che se Tagged, faranno in modo di evitare eventuali collisioni tra i frame dei flussi di dati.

Come si può vedere dall'immagine sottostante è formato da 24 porte ethernet 10/100Base-TX e ogni porta è segnalata dal corrispettivo Led, che si accenderà o spegnerà a seconda che la porta sia in funzione o meno. Sulla sinistra vi è una porta seriale che può essere utilizzata come console, e i pulsanti di clear e reset che verranno utilizzati per riportare lo switch in modalità di default, di modo che si possa assegnargli un indirizzo IP dal Dhcp server e accedervi.



Uno degli aspetti fondamentali Nella serie 2500 è che le VLAN sono configurabili in un modo molto utile: se si creano più VLAN all'interno dello switch, ogni VLAN si comporterà come uno switch logico separato contenente solo le porte che gli sono state assegnate. Ogni parte quindi ha un dominio di broadcast isolato dalle altre, come se fossero più switch separati fisicamente, e non ci può quindi essere nessuna comunicazione tra le porte di VLAN diverse, che però rispetteranno le operazioni classiche proprie di ogni apparato di rete di livello 2. Risulta logico comprendere quindi come non sia necessario utilizzare la modalità tagged della rete virtuale, ma sarà sufficiente mantenere tutte le porte in untagged. *

Alla ricezione di ogni pacchetto di dati, lo switch determinerà l'indirizzo di destinazione cercando negli tabella degli indirizzi. Questa tabella viene compilata e aggiornata ogni volta che un dispositivo viene connesso allo switch, ed è composta dagli indirizzi MAC che rendono ogni apparato di rete unico, e quindi indirizzabile univocamente.

Una volta ricevuto il pacchetto, in base alla locazione della porta dell'indirizzo letto nella tabella, lo switch deciderà se spedire il pacchetto o se filtrarlo.

features and benefits

- **9.6 Gbps switch fabric integrated on-chip:** high-performance switch design with a non-blocking architecture
- **hp auto-MDIX:** automatically adjusts for straight-through or crossover cables on all 10/100-TX and 100/1000-T ports
- **stacking capability:** single IP address management for a virtual stack of up to 16 switches including the 1600m, 2400m, 2424m, 2512, 2524, 4000m, 8000m, and 4100gl series
- **RMON and switch monitoring (SMON):** provides monitoring and reporting capabilities for statistics, history, alarms, and events
- **Web interface:** allows you to configure the switch from any Web browser on the network
- **802.3ad Link Aggregation Control Protocol (LACP) and hp trunking:** supports a single trunk with up to 4 links (ports)
- **VLAN support and tagging:** supports up to 30 port-based VLANs and dynamic configuration of 802.1Q VLAN tagging, providing security between workgroups
- **Group VLAN Registration Protocol (GVRP):** allows automatic learning and dynamic assignment of VLANs
- **IP multicast (IGMP):** prevents flooding of IP multicast traffic
- **port security:** prevents unauthorized access using MAC address lockdown
- **Spanning Tree Protocol:** provides redundant links while preventing network loops
- **IEEE 802.1p prioritization:** delivers data to devices based on the priority and type of traffic
- **TACACS+:** eases administration of switch management security by using a password authentication server
- **Cisco Fast EtherChannel®:** supports Cisco's proprietary FEC trunking protocol
- **Rapid Convergence Spanning Tree Protocol (802.1w):** increases network uptime through faster recovery from failed links
- **802.1x and RADIUS network login:** controls port-based access for authentication and accountability
- **Cisco Discovery Protocol (CDP):** enables real-time mapping of nodes to switch ports
- **lifetime warranty:** for as long as you own the product, with next-business-day advance replacement (available in most countries)

Tra le VLAN presenti all'interno dello switch, è sempre presente la Default Vlan, che non può essere infatti cancellata. La sua esistenza è dovuta al fatto che lo switch ha sempre bisogno di almeno una VLAN a cui assegnare le sue porte. Infatti se si vuole cancellare una VLAN con delle porte già assegnate, bisognerà prima ri-assegnare le porte ad un'altra LAN virtuale, e poi si avrà il permesso di cancellare la rete indesiderata ("no vlan <id>").

Inizialmente tutte le porte dello switch sono in modalità untagged member nella VLAN di default che se non cambiato ha ID 1.

Nota*: **Il Tagging.**

Il **tagging** è un espediente per permettere che più Vlan convivino nello stesso apparato. Per necessità infatti, può capitare che una porta sia assegnata a più Vlan, e quindi il traffico dati, che viene opportunamente separato dalle due reti virtuali distinte(obiettivo per cui si sono create queste lan virtuali) sarebbe compromesso. Per evitare ciò, si "tagga" ogni porta che ha in comune più Vlan, di modo che il traffico venga scremato in base al destinatario del pacchetto. Questo si attua inserendo nel frame del pacchetto un apposito codice nel campo Vlan Tag, seguendo il protocollo 802.1Q. Così il traffico delle Vlan verrà comunque diviso nonostante la porta fisica in comune.

3.4 Minicom

Minicom è un **terminale text-based** per i sistemi operativi Unix, utilizzato per la programmazione di porte seriali nella parte riguardante la comunicazione esterna con RS-232.

Tra le sue caratteristiche:

- Impostazioni da remoto di una console seriale;
- Permette l'accesso ad un server o a un computer nel caso in cui la LAN sia down;
- Si connette con i Router Cisco per la configurazione;
- Si connette a sistemi Embedded Linux e dispositivi BSD attraverso il cavo model zero;
- Separa gli interpreti di linguaggio di script;
- Cattura file;
- Compone elenchi con l'auto-redial.

```
Minicom Command Summary
Commands can be called by CTRL-A <key>

Main Functions          Other Functions
Dialing directory..D  run script (Go)...G | Clear Screen.....C
Send files.....S     Receive files.....R | cOnfigure Minicom..O
comm Parameters....P  Add linefeed.....A | Suspend minicom...J
Capture on/off....L  Hangup.....H       | eXit and reset....X
send break.....F     initialize Modem...M | Quit with no reset.Q
Terminal settings..T  run Kermit.....K   | Cursor key mode...I
lineWrap on/off...W  local Echo on/off..E | Help screen.....Z
Paste file.....Y     | scroll Back.....B

Select function or press Enter for none.█

Written by Miquel van Smoorenburg 1991-1995
Some additions by Jukka Lahtinen 1997-2000
```

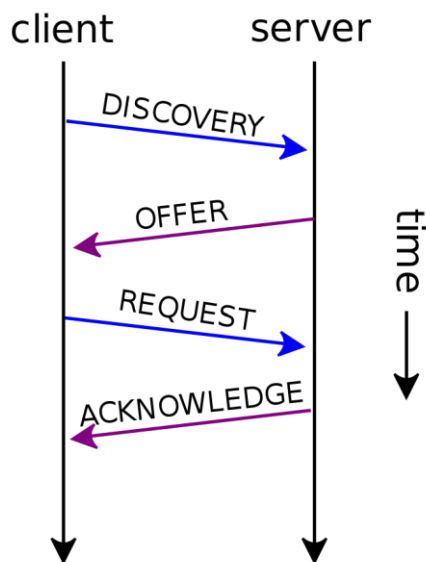
Ci si servirà delle sue caratteristiche infatti per accedere al router Cisco tramite porta seriale e adattatore USB.

3.5 DHCP

Il dhcp è un **protocollo** di configurazione dinamica degli host attraverso cui si possono gestire diversi apparati di rete. Infatti permette l'assegnamento di indirizzi IP in modo casuale ad ogni dispositivo al momento della sua accensione, o anche durante il suo normale funzionamento se richiesto.

Il server dhcp è stato ideato dall' Internet Software Consortium, e per quanto riguarda il pacchetto Debian, è stato implementato come *isc-dhcp-server*.

Nel protocollo sono sempre presenti due oggetti che comunicano. Uno è il client dhcp che è un calcolatore a cui serve un indirizzo IP per la subnet di cui fa parte, e l'altro è il server dhcp, che è l'oggetto che glielo fornirà. Questa funzione di server a volte è presente anche nei router.



Il protocollo si struttura attraverso diverse fasi di *handshake* tra client e server.

Innanzitutto il client invia un pacchetto chiamato DHCPDISCOVER con destinazione l'indirizzo broadcast. Quando questa trama arriva al server DHCP, questo può rispondere con un pacchetto DHCPOFFER, in cui viene proposto un indirizzo IP.

Una volta che il client ha ottenuto l'indirizzo IP desiderato, manda un DHCPREQUEST in broadcast al server scelto.

Il server selezionato conferma l'assegnazione dell'indirizzo con un DHCPACK e la fase di assegnazione termina.

Ora il client può usufruire dell'indirizzo IP solo per un certo periodo di tempo, detto di lease. Questo parametro può essere configurato nel file

dhcpd.conf. Scaduto questo intervallo, il client dovrà richiedere un prolungamento dell'indirizzo con un altro DHCPREQUEST, che sarà convalidato da un altro DHCPACK. Nel caso questa procedura non vada a buon fine, il client perde l'IP acquisito.

3.6 Wireshark



Wireshark è un **tool** ideato per l'**analisi di protocolli di rete** o come **packet sniffer**. Le sue funzionalità sono molto simili a quelle di tcpdump, in quanto permette di osservare il traffico di dati di una rete, ma ha un'interfaccia grafica e maggiori funzionalità.

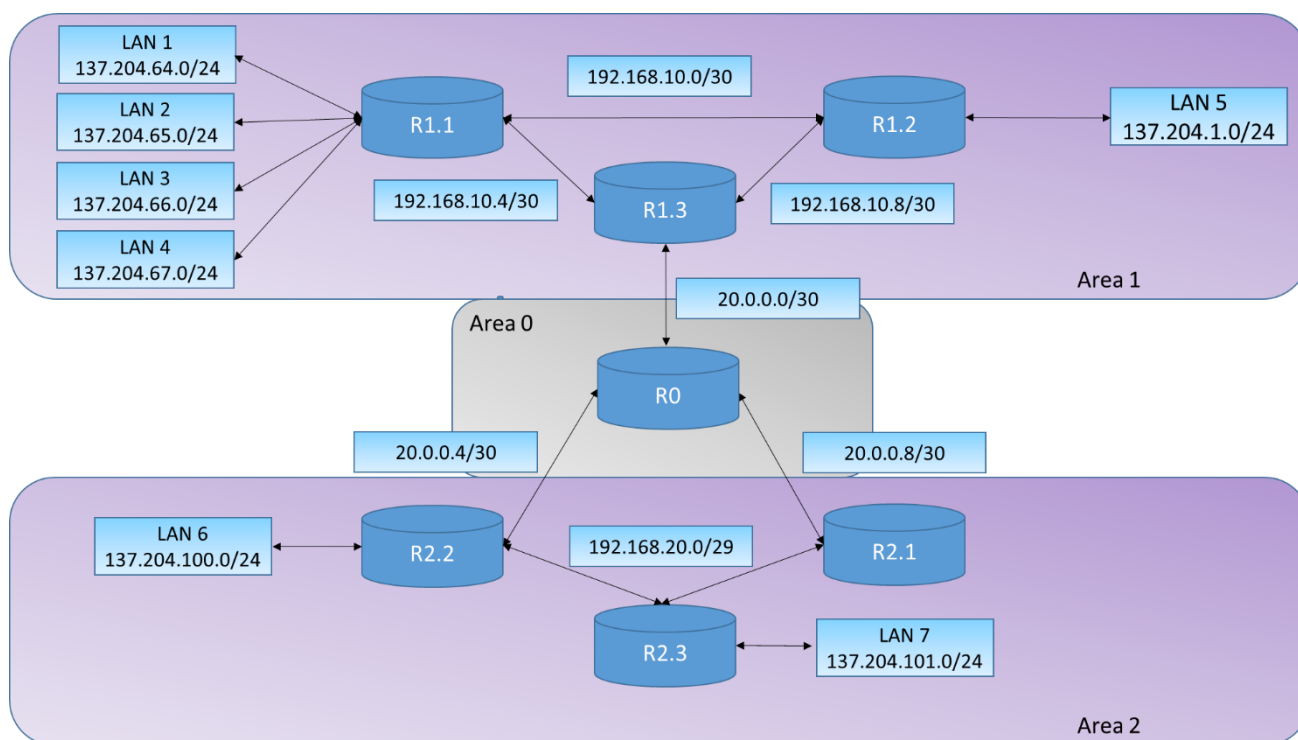
Wireshark è utile per comprendere la struttura di diversi protocolli di rete, che riesce a riconoscere, individuando eventuali incapsulamenti e riconoscendo i singoli campi di un pacchetto. Per la cattura di pacchetti utilizza il codice libcap/winPcap, quindi le reti analizzate devono supportarlo.

E' stato utilizzato nel corso della tesi per osservare il corretto funzionamento del protocollo DHCP, e per catturare l'IP che veniva assegnato allo switch in fase di configurazione, di modo da potervi accedere.

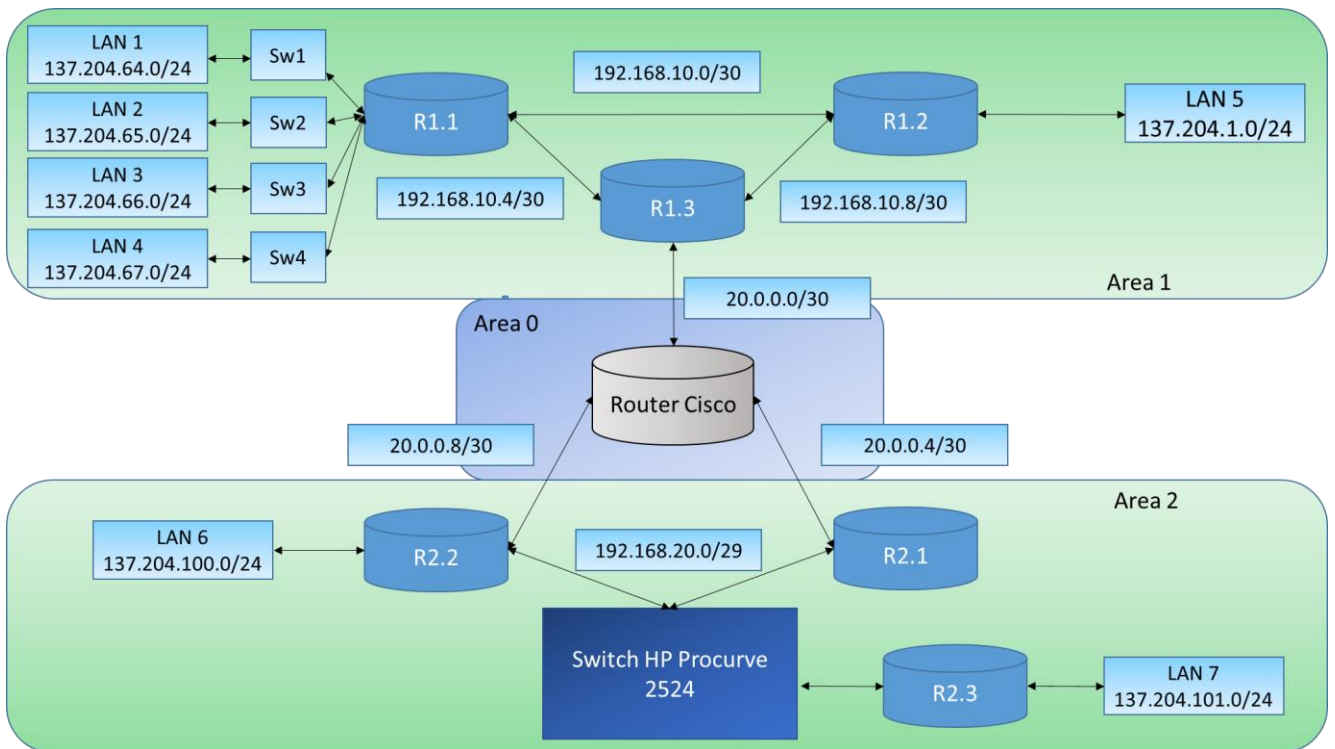
4 Creazione nuova network

4.1 Confronto tra vecchia e nuova topologia

La topologia dell'ing. Nicola Sparnacci prevedeva l'utilizzo unicamente di Raspberry per testare i protocolli di routing in una rete di più recente generazione. La nuova topologia invece avrà lo scopo di testare come si possano implementare reti utilizzando componenti diversi dai soli Raspberry, dando vita quindi a reti più complesse e più adattabili ai dispositivi presenti in qualsiasi rete già costruita. Questo approccio quindi è utile per far comprendere come i Raspberry possano essere sfruttati per qualsiasi utilizzo. Sia dal creare una rete completamente nuova, sia sostituendo parti integranti di una rete già operativa.



Nell'immagine di sopra è illustrata la vecchia configurazione in cui saranno da apportare le modifiche sopra citate, mentre nell'immagine di sotto è come dovrebbe risultare la nuova topologia di rete, con il Router Cisco 1801, lo Switch HP Procurve 2524. Si sono anche voluti mettere in risalto gli switch virtuali creati per ogni rete LAN nel raspberry R1.1.



Si può notare come le differenze sostanziali tra le due topologie dal punto di vista fisico siano ben poche, questo per via della grande adattabilità dei Raspberry, che grazie alla loro personalizzazione, alla facilità di utilizzo del sistema operativo, e al possibile utilizzo del tool mininet, possano implementare quasi ogni tipo di funzione. Ciò non toglie che la maggior parte delle configurazioni di rete attuali siano costruite di base con switch e router fisici, e quindi l'interconnessione tra i due sistemi (virtuali e fisici) è fondamentale per la riuscita di reti funzionali agli scopi richiesti in fase di progetto.

4.1.1 Modifica Raspberry R1.1

A livello progettuale, la scelta di utilizzare un solo switch per 4 Lan differenti non è la migliore. Infatti il primo passo di questa tesi sarà quello di modificare il raspberry R1.1 creando quattro switch virtuali, e facendo in modo che ognuno di essi sia collegato unicamente ad una sola Lan. Per fare questo ci si è serviti di mininet, sfruttando un API di Python che permette tramite uno script, la creazione di topologie di rete complesse in modo molto efficace.

```
from mininet.topo import Topo

class MyTopo( Topo ):
    # Topologia con quattro switch collegati a quattro LAN.

    def __init__( self ):
        # Creazione topologia.

        # Inizializzazione topologia
        Topo.__init__( self )

        # Aggiunta di host e switch
        Host1 = self.addHost( 'h1', ip='137.204.64.1/24', defaultRoute='via
137.204.64.254' )
        Host2 = self.addHost( 'h2', ip='137.204.65.1/24', defaultRoute='via
137.204.65.254' )
        Host3 = self.addHost( 'h3', ip='137.204.66.1/24', defaultRoute='via
137.204.66.254' )
        Host4 = self.addHost( 'h4', ip='137.204.67.1/24', defaultRoute='via
137.204.67.254' )
        Switch1 = self.addSwitch( 's1' )
        Switch2 = self.addSwitch( 's2' )
        Switch3 = self.addSwitch( 's3' )
        Switch4 = self.addSwitch( 's4' )

        # Aggiunta collegamenti
        self.addLink( Host1, Switch1 )
        self.addLink( Host2, Switch2 )
        self.addLink( Host3, Switch3 )
        self.addLink( Host4, Switch4 )
        topos = { 'topo1': ( lambda: MyTopo() ) }
```

Per lanciare il seguente script e far sì che la rete venga implementata, bisognerà inserire l'opzione custom e poi il percorso del file. Per ultimo il nome che si è dato alla topologia che è stato assegnato nell'ultima linea del listato.

```
sudo mn --custom /home/pi/mininet/topo1.py --topo=topo1
```

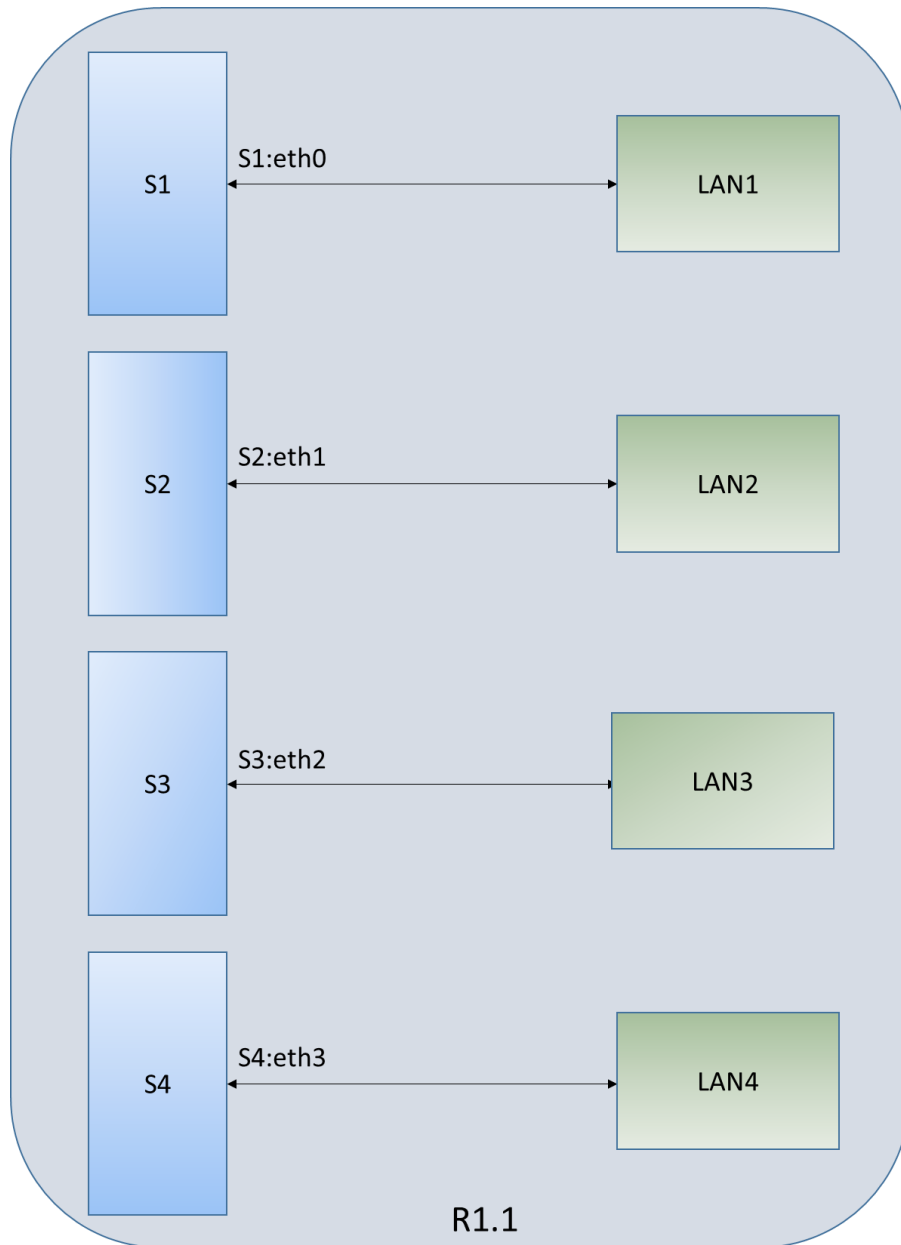
```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ sudo mn --custom /home/pi/mininet/custom/topo1.py --topo=topo1
**** Creating network
**** Adding controller
**** Adding hosts:
h1 h2 h3 h4
**** Adding switches:
s1 s2 s3 s4
**** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4)
**** Configuring hosts
h1 h2 h3 h4
**** Starting controller
c0
**** Starting 4 switches
s1 s2 s3 s4 ...
**** Starting CLI:
mininet>
```

L'ulteriore parte che andrà modificata sarà quella riguardante la creazione di un'interfaccia per ogni switch virtuale, e l'assegnamento di un indirizzo di default gateway per gli host presenti nella lan, che sarà attuato da terminale.

```
sudo ovs-vsctl add-port s1 eth0
sudo ovs-vsctl add-port s1 eth1
```

```
sudo ip addr add 137.204.64.254/24 dev s1
sudo ip addr add 137.204.65.254/24 dev s1
sudo ip addr add 137.204.66.254/24 dev s1
sudo ip addr add 137.204.67.254/24 dev s1
```

La configurazione del Raspberry R1.1 quindi si presenta così alla fine del progetto.



4.2 Configurazione router Cisco 1801

4.2.1 Accedere al dispositivo: configurazione Minicom

Una volta alimentato il dispositivo, si possono utilizzare diverse modalità per accedervi:

- utilizzo della porta Console del router.
- tramite configurazione remota (telnet o SSH).
- utilizzo di Browser Web (funzionalità molto limitate).
- SNMP (necessità di programmi appositi, non molto utilizzato).

Per la configurazione remota è necessario che ci sia un collegamento a livello di IP tra router e l'apparato di gestione. In questo caso, siccome il router è dotato di più interfacce di rete, e quindi di più indirizzi IP, è possibile utilizzare un qualsiasi indirizzo per collegarvisi.

Nel nostro caso per accedervi, si è utilizzata la Console che non presenta problemi di raggiungibilità, ed è ideale per i primi utilizzi di configurazione del router.



Ci si collega quindi da una parte tramite cavo RJ45 alla porta ethernet "Console" e dall'altra con l'adattatore alla porta USB del computer per effettuare la

configurazione.

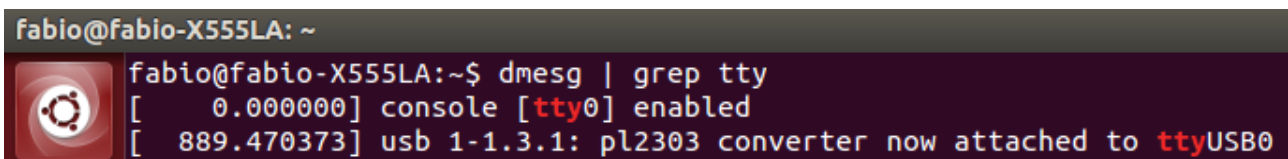
Da terminale di Ubuntu, si installa se non presente Minicom, un tools che tra le molteplici funzioni ha quella di permettere l'interazione con i router della famiglia Cisco. L'installazione avviene tramite comando:

```
$ sudo apt-get install minicom
```

Si avvia poi la modalità di configurazione tramite comando:

```
$ minicom -s
```

Per prima cosa bisogna settare la porta di utilizzo che nel nostro caso è la USB0. Per verificare il settaggio delle porte si sfrutta il comando `$ dmesg | grep tty`.

A terminal window screenshot with a dark background. The prompt is 'fabio@fabio-X555LA: ~'. The user has entered the command 'fabio@fabio-X555LA:~\$ dmesg | grep tty'. The output shows two lines: '[0.000000] console [tty0] enabled' and '[889.470373] usb 1-1.3.1: pl2303 converter now attached to ttyUSB0'.

```
fabio@fabio-X555LA: ~  
fabio@fabio-X555LA:~$ dmesg | grep tty  
[ 0.000000] console [tty0] enabled  
[ 889.470373] usb 1-1.3.1: pl2303 converter now attached to ttyUSB0
```

Questo passaggio è utile per vedere se ci si è attaccati alla porta giusta, di modo che il router Cisco e il computer possano "vedersi".

Si apre quindi Serial Port Setup dal menù di configurazione di minicom e in serial device si andrà a modificare dev/ttyUSB0.


```
+-----+
| A -   Serial Device       : /dev/ttyUSB0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 9600 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : Yes
|
|   Change which setting? █
+-----+
|
|   Screen and keyboard
|   Save setup as dfl
|   Save setup as..
|   Exit
+-----+

Router_R1 con0 is now available
```

Sempre nel medesimo menù, bisogna cambiare la velocità della linea in 9600 bps e inserire la combinazione 8N1 che sta per 8 bit di dati, 1 bit di parità, e 1 bit di stop. L'ultima configurazione da effettuare è quella di verifica che la modalità di controllo delle flow di tipo hardware sia "si" e quella software "no", e lasciare invariato tutto il resto.

Fatto tutto ciò, bisogna salvare tramite "Save setup as ..." e dare un nome che poi andrà richiamato in seguito nella procedura di avvio del dispositivo. Infine si effettua il reboot del software, che ora sarà pronto per essere operativo.

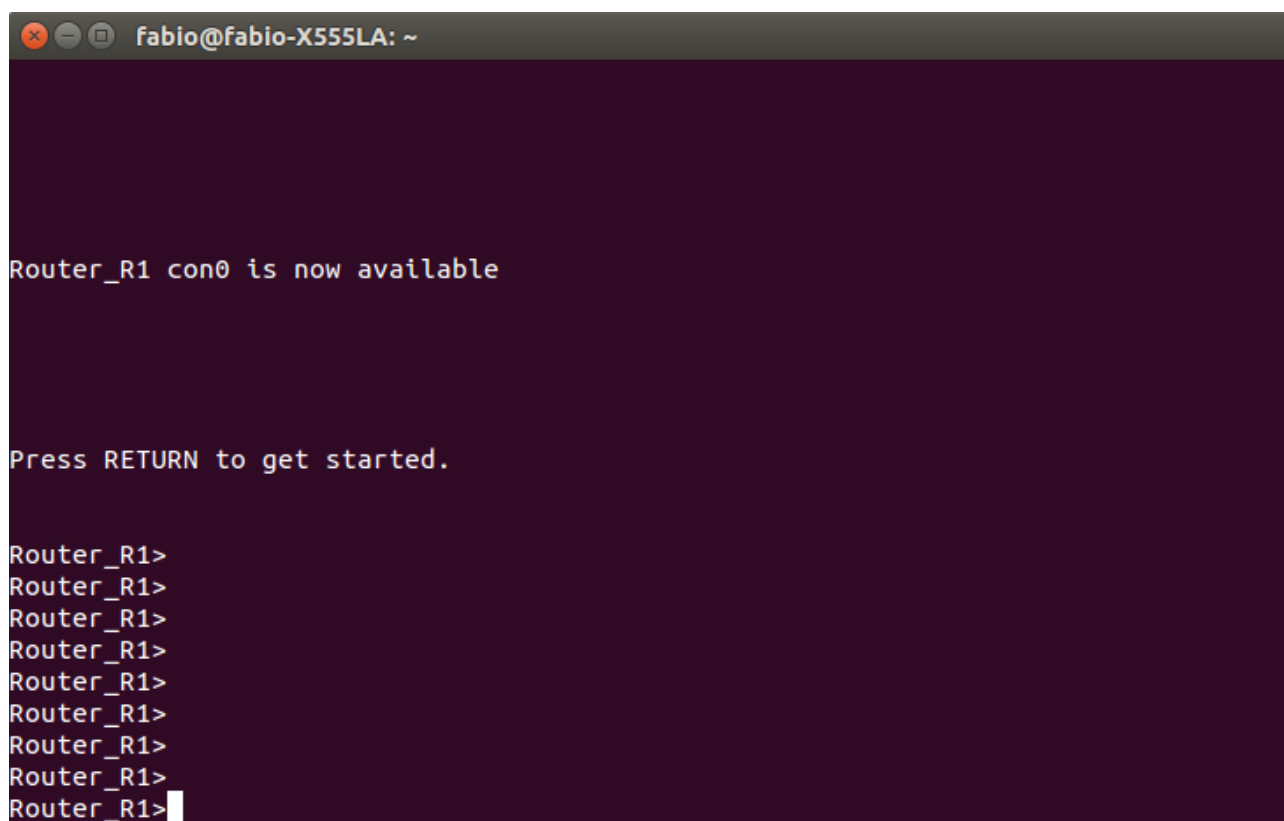
4.2.2 Configurazione interfacce

Ora che si è settato nel modo giusto minicom, si è pronti per accedere al dispositivo.

Da terminale si entra nel software di configurazione del router Cisco con il comando:

```
# minicom "nome configurazione".
```

In seguito sarà sufficiente premere qualche volta il tasto invio per accedere al dispositivo.



```
fabio@fabio-X555LA: ~  
  
Router_R1 con0 is now available  
  
Press RETURN to get started.  
  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>  
Router_R1>
```

Il primo passo è quello di passare alla modalità privilegiata di amministratore. Questo è possibile con il comando "enable" e l'inserimento della password scelta dagli amministratori del router.

Il passo successivo è quello di vedere la configurazione attuale del dispositivo. Questo processo si attua tramite comando "Show running-config".

Con questo comando si osservano le configurazioni delle Fast Ethernet presenti nel router e altri parametri che in questo caso non interessano.

```

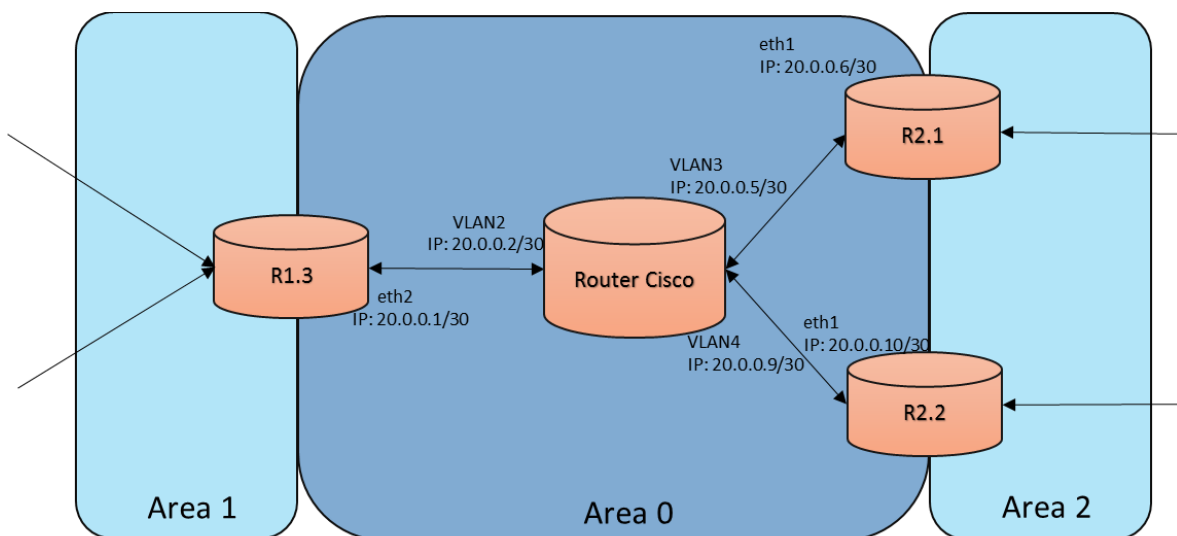
Router_R1>enable
Password:
Router_R1#show running-config
Building configuration...

Current configuration : 1893 bytes
!
version 12.4

```

Ora si entra in modalità di configurazione per modificare i parametri delle interfacce di rete nelle porte Fast Ethernet. Per fare questo si digita la stringa *configure terminal* e successivamente si utilizza il comando *interface vlan n°*. Si andranno quindi ad inserire in 3 vlan, i 3 indirizzi IP con relative netmask con le quali il router R0 comunica con l'esterno. L'obiettivo che ci si è posti è quello di sostituire il router virtuale R0 della topologia che era stato creato tramite raspberry, con un router fisico. Quindi perchè questo accada, occorrerà assegnare gli stessi indirizzi ip della rete analizzata precedentemente. In questo caso si sfrutteranno le VLAN assegnate alle porte FastEthernet del router di modo che funzionino come interfacce di rete e siano visibili quindi dalle altre interfacce presenti nella topologia costruita:

- 20.0.0.2/30 che connette R0 al router R1.3;
- 20.0.0.5/30 per connettersi al router dell'area 2 R2.1;
- 20.0.0.9/30 per connettersi al router dell'area 2 R2.2.



```
Router_R1#
Router_R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router_R1(config)#interface vlan 2
Router_R1(config-if)#ip address 20.0.0.2 255.255.255.252
Router_R1(config-if)#exit
Router_R1(config)#interface vlan 3
Router_R1(config-if)#ip address 20.0.0.5 255.255.255.252
Router_R1(config-if)#exit
Router_R1(config)#interface vlan 4
Router_R1(config-if)#ip address 20.0.0.9 255.255.255.252
Router_R1(config-if)#end
Router_R1#
```

Siccome si sono create più reti slegate tra di loro, si è utilizzato il protocollo OSPF per interfacciarle. Quindi di conseguenza, dato che il router R0 era il router di Backbone e gli era stata assegnata l'area 0 come rete, anche nel router Cisco andrà utilizzato il protocollo OSPF sempre assegnando l'area 0, di modo che anche le altre interfacce di rete fuori dalla sua area possano comunicare con lui.

4.2.3 Configurazione OSPF nel router Cisco

Viene di seguito descritto come implementare il protocollo OSPF all'interno del router Cisco.

OSPF (Open Shortest Path First) è un protocollo di gateway interno (IGP) progettato principalmente per le reti IP, supportando il subnetting e il tagging. Permette inoltre l'autenticazione dei pacchetti e usa IP multicast nell'invio e nella ricezione di quest'ultimi.

Cisco supporta la versione RFC 1253 denominata MIB (Management Information Base). L'OSPF MIB definisce un protocollo di routing IP che fornisce informazioni di gestione relative all'OSPF e supportate dai router Cisco.

Per abilitare la modalità di configurazione di OSPF nel router CISCO si ricorre al comando `router ospf process_id` (ad esempio `router ospf 0`).

Lo step successivo sarà quello di inserire l'indirizzo di ogni network all'interno dell'area OSPF e il numero dell'area assegnatagli.

Nel caso preso in esame sarà quindi:

```
Router_R1(config-router)#network 20.0.0.0 255.255.255.252 area 0
Router_R1(config-router)#network 20.0.0.4 255.255.255.252 area 0
Router_R1(config-router)#network 20.0.0.8 255.255.255.252 area 0
Router_R1(config-router)#end
Router_R1#
```

Si possono quindi osservare i cambiamenti derivanti dalle modifiche appena effettuate rilanciando il comando `$ show running-config`:

```
interface Vlan3
ip address 20.0.0.5 255.255.255.252
!
interface Vlan4
ip address 20.0.0.9 255.255.255.252
!
interface Vlan2
ip address 20.0.0.2 255.255.255.252
!
router ospf 109
log-adjacency-changes
network 20.0.0.0 0.0.0.3 area 0
network 20.0.0.4 0.0.0.3 area 0
network 20.0.0.8 0.0.0.3 area 0
```

Da notare che le netmask nel campo `router ospf 109` sono in logica inversa (logica tradizionale sarebbe `255.255.255.252`).

4.2.4 Connessione tra VLAN router e Raspberry

Ora che sono stati configurati gli indirizzi IP per ogni VLAN del router e sono anche stati settati i parametri per rispettare il protocollo OSPF, si può effettuare il collegamento con i 3 raspberry delle aree 1 e 2.

Per verificare che non siano stati commessi errori e che la funzione di routing di R0 funzioni in modo corretto, si sono effettuate delle prove attraverso l'uso del comando ping.

Nell'attivazione dei raspberry, dopo essere entrati e aver verificato con il comando `"sudo ifconfig"` che le interfacce fisiche di rete a cui ci si è collegati siano quelle giuste e che abbiano assegnato l'indirizzo IP corretto, bisogna creare dove necessario uno switch virtuale tramite il tool *mininet*. Questo accade siccome nella precedente configurazione, gli indirizzi ip delle interfacce di rete erano stati assegnati alle interfacce dello switch virtuale, dato che quasi ogni raspberry doveva implementare questa funzione per via delle più interconnessioni tra LAN diverse.

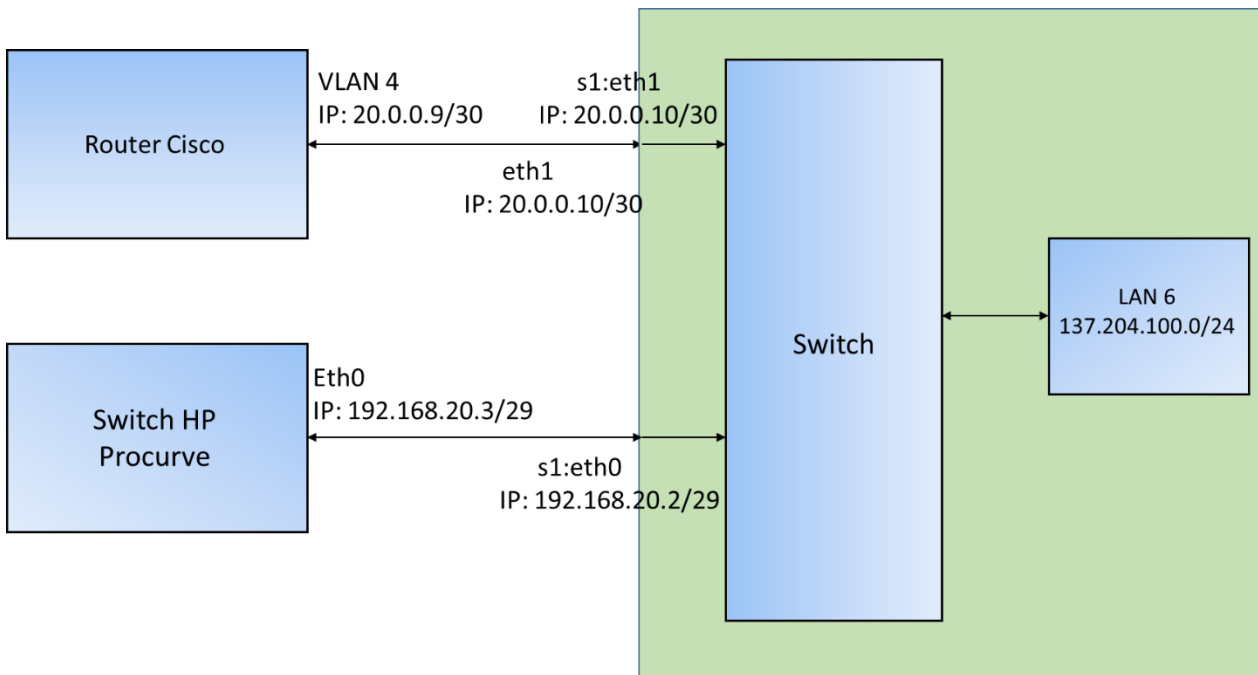
Ad esempio per il raspberry 2.2 si crea uno switch a topologia singola per l'host della lan 5:

```
# mn --switch=ovs --topo=single,1
mininet> py h1.setIP('137.204.100.1/24')
mininet>h1 route add default gw 137.204.100.254
```

Vengono poi aggiunte le interfacce fisiche nello switch s1 attraverso il comando `ovs-vsctl`:

```
# ovs-vsctl add-port s1 eth0
# ovs-vsctl add-port s1 eth1
```

Infine si attiva l'ip forwarding attraverso il comando `sudo sysctl_ipforward = 1`. Si ottiene quindi la seguente configurazione:



Ora si verifica tramite il comando ping che gli apparati di rete riescano nello scambio di informazione. Ecco alcuni esempi:

Comunicazione:

- R2.2 (20.0.0.10/30) ———> Cisco Router (20.0.0.9/30)
- R2.2 (20.0.0.10/30) ———> R1.3 (20.0.0.1/30)

```

pi@raspberrypi ~ $ ping 20.0.0.9
PING 20.0.0.9 (20.0.0.9) 56(84) bytes of data.
64 bytes from 20.0.0.9: icmp_req=1 ttl=255 time=7.33 ms
64 bytes from 20.0.0.9: icmp_req=2 ttl=255 time=1.29 ms
64 bytes from 20.0.0.9: icmp_req=3 ttl=255 time=0.545 ms
64 bytes from 20.0.0.9: icmp_req=4 ttl=255 time=0.493 ms
64 bytes from 20.0.0.9: icmp_req=5 ttl=255 time=0.498 ms
^C
--- 20.0.0.9 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.493/2.033/7.335/2.668 ms
pi@raspberrypi ~ $ ping 20.0.0.1
PING 20.0.0.1 (20.0.0.1) 56(84) bytes of data.
64 bytes from 20.0.0.1: icmp_req=1 ttl=63 time=6.35 ms
64 bytes from 20.0.0.1: icmp_req=2 ttl=63 time=1.42 ms
64 bytes from 20.0.0.1: icmp_req=3 ttl=63 time=0.707 ms
64 bytes from 20.0.0.1: icmp_req=4 ttl=63 time=0.734 ms
64 bytes from 20.0.0.1: icmp_req=5 ttl=63 time=0.722 ms
^C
--- 20.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.707/1.986/6.350/2.198 ms
pi@raspberrypi ~ $

```

Visualizzazione dell'interfaccia eth2 di R1.3 (indirizzo IP 20.0.0.1)

Ping : R1.3 (20.0.0.1) —> Router Cisco (20.0.0.2)

```
eth2    Link encap:Ethernet  HWaddr d0:a6:37:e3:8d:88
        inet addr:20.0.0.1 Bcast:20.0.0.3 Mask:255.255.255.252
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:37 errors:0 dropped:2 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:3418 (3.3 KiB)  TX bytes:9375 (9.1 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:96 errors:0 dropped:0 overruns:0 frame:0
        TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:7776 (7.5 KiB)  TX bytes:7776 (7.5 KiB)

pi@raspberrypi ~ $ ping 20.0.0.2
PING 20.0.0.2 (20.0.0.2) 56(84) bytes of data:
64 bytes from 20.0.0.2: icmp_req=1 ttl=255 time=0.695 ms
64 bytes from 20.0.0.2: icmp_req=2 ttl=255 time=0.589 ms
64 bytes from 20.0.0.2: icmp_req=3 ttl=255 time=0.577 ms
64 bytes from 20.0.0.2: icmp_req=4 ttl=255 time=0.594 ms
64 bytes from 20.0.0.2: icmp_req=5 ttl=255 time=0.582 ms
^C
--- 20.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/ndev = 0.577/0.607/0.695/0.049 ms
pi@raspberrypi ~ $ _
```

Ping : Router Cisco (20.0.0.9) —> R2.2 (20.0.0.10)

```
Router_R1#ping 20.0.0.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Router_R1#
```

Ping : Router Cisco (20.0.0.9) —> R1.3 (20.0.0.1)

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

4.3 Configurazione Switch HP Procurve 2524

4.3.1 Accedere al dispositivo: configurazione DHCP

Una volta alimentato il dispositivo si possono utilizzare diversi metodi per accedervi:

- Menù interface: un'interfaccia che offre un subset di comandi dello switch attraverso la console built VT-100/ANSI;
- CLI: l'interfaccia offerta dalla command line offre l'intero set di istruzioni attraverso la console VT-100/ANSI dello switch;
- Web Browser Interface: in interfaccia che offre informazioni sullo stato dello switch e un subset di comandi attraverso il web browser;
- HP TopTools per Hubs & Switches : semplice da usare, è un tool basato su una gestione della rete tramite browser che funziona bene con le caratteristiche delle reti gestite in uno switch HP.

Si è scelto di servirsi dell'accesso tramite telnet, impostando un indirizzo casuale tramite server dhcp generato da un computer. Per permettere l'assegnazione di un nuovo indirizzo IP allo switch, è necessario resettarlo, cancellando così ogni password memorizzata precedentemente, e ripristinando le impostazioni di default. Si toglie così qualsiasi indirizzo IP assegnatogli in precedenza e ogni configurazione delle VLAN.

Per mandare il dispositivo in modalità di reset bisogna effettuare in ordine i seguenti step:

- Premere contemporaneamente, utilizzando un oggetto appuntito, i pulsanti clear e reset.
- Continuando a tenere premuto clear, rilasciare il pulsante reset.
- Appena il led self test comincia a lampeggiare si può rilasciare anche il clear.

Fatto questo passaggio, ora lo switch è pronto per essere configurato: sarà sufficiente mandargli un indirizzo IP casuale tramite dhcp, che questo verrà memorizzato dall'apparato, e sarà poi raggiungibile tramite il comando #telnet "indirizzo ip". Questo si realizza a livello fisico attraverso un cavo ethernet, che sarà collegato tra l'interfaccia di rete enp2s0 e una qualunque porta ethernet dello switch.

Per far funzionare correttamente il dhcp bisogna configurare 2 file.
Al primo si accede tramite il comando :

```
Sudo nano /etc/default/isc-dhcp-server
```

Bisogna cambiare la parte riguardante le interfacce, siccome si vuole che il dhcp funzioni attraverso l'interfaccia Enp2s0.

```
GNU nano 2.4.2 File: /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts
#
# This is a POSIX shell fragment
#
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPD_CONF=/etc/dhcp/dhcpd.conf
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid
# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="enp2s0"
```

E' necessario poi modificare il file dhcpd.conf a cui si può accedere nel seguente modo:

```
Sudo nano /etc/dhcp/dhcpd.conf
```

I parametri da scrivere sono quelli relativi alla dichiarazione di una subnet, in cui andrà scritto un range di indirizzi IP di cui uno verrà prelevato in modo casuale come indirizzo per lo switch. Inoltre si specifica un gateway e si abilita la direttiva authoritative siccome il server dhcp sarà l'unico della rete locale. Si finisce dando un nome di dominio, un dns primario e secondario, e specificando i tempi nel

quale rimarrà attivo, una volta lanciato il servizio.

Ecco un esempio di dichiarazione di subnet per il server dhcp:

```
# This is a very basic subnet declaration.

subnet 192.168.1.0 netmask 255.255.255.0 {
  authoritative;
  range 192.168.1.150 192.168.1.200;
  option domain-name-servers 192.168.1.1, 192.168.1.2; #pri DNS , sec DNS
  option domain-name "mydomain.example";
  option routers 192.168.1.254; #gateway
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.1.255; #broadcast
  default-lease-time 600;
  max-lease-time 7200;
# option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}
```

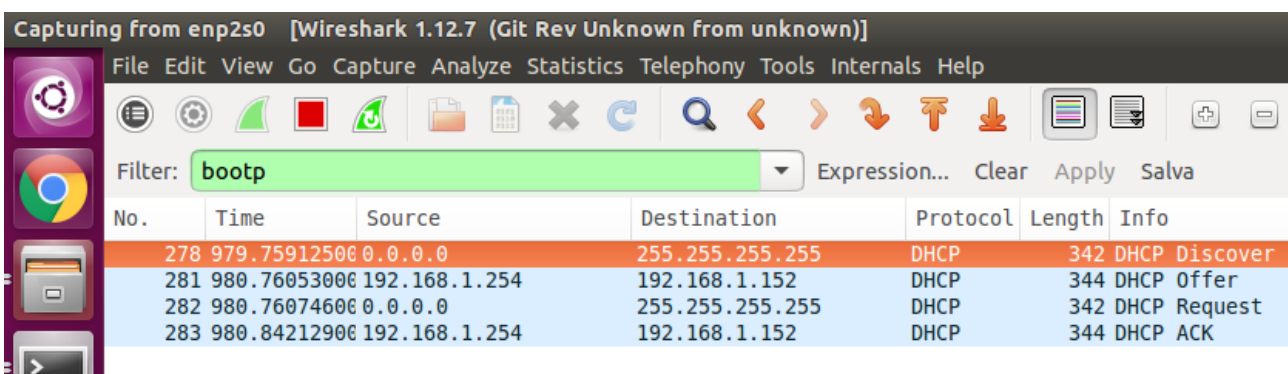
Ora si è pronti per lanciare il server dhcp attraverso Enp2s0.

Per verificare che ci sia comunicazione tra il dhcp e lo switch si apre l'analizzatore di pacchetti wireshark.

Ora si può avviare il dhcp attraverso il comando:

```
Sudo service isc-dhcp-server start
```

Ci si posiziona poi su wireshark, inserendo come filtro "bootp" per vedere lo scambio di pacchetti tra i due apparati di rete.



No.	Time	Source	Destination	Protocol	Length	Info
278	979.759125000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
281	980.760530000	192.168.1.254	192.168.1.152	DHCP	344	DHCP Offer
282	980.760746000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
283	980.842129000	192.168.1.254	192.168.1.152	DHCP	344	DHCP ACK

Come si può vedere dall'immagine, lo switch tramite Enp2s0 manda un DHCP Discover a tutti i dispositivi presenti nella rete (indirizzo di broadcast 255.255.255.255), e attende una risposta dal server. Questa risposta arriva dal server che ha l'indirizzo 192.168.1.254 che manda un DHCP offer all'indirizzo dello switch 192.168.1.152. La transazione viene completata, come si è spiegato nel

paragrafo del DHCP, con un dhcp Request e un DHCP ack, segno che lo scambio di pacchetti è andato a buon fine.

Ora si entrerà nello switch tramite l'utilizzo del comando telnet:

```
Sudo telnet 192.168.1.153
```

Si nota che l'ip è diverso da quello precedente, perché si utilizza la configurazione effettuata la prima volta che è stato utilizzato il dhcp dopo il reset dello switch. Sotto invece si può vedere lo scambio di pacchetti che si ha con il comando telnet nell'ambiente wireshark.

28	388.83218200	192.168.1.254	192.168.1.153	TCP	74	60172-23	[SYN]
29	388.84327600	192.168.1.153	192.168.1.254	TCP	62	23-60172	[SYN,
30	388.84333800	192.168.1.254	192.168.1.153	TCP	54	60172-23	[ACK]
31	388.84351800	192.168.1.254	192.168.1.153	TELNET	81	Telnet Data ..	
32	388.88744200	192.168.1.153	192.168.1.254	TELNET	60	Telnet Data ..	
33	388.88746200	192.168.1.254	192.168.1.153	TCP	54	60172-23	[ACK]
34	388.89258800	192.168.1.153	192.168.1.254	TELNET	656	Telnet Data ..	
35	388.89260600	192.168.1.254	192.168.1.153	TCP	54	60172-23	[ACK]
36	388.89266600	192.168.1.254	192.168.1.153	TELNET	57	Telnet Data ..	
37	388.89937100	192.168.1.153	192.168.1.254	TCP	60	23-60172	[ACK]

Dopo qualche secondo, apparirà la seguente schermata, segno che l'accesso al device è andato a buon fine.

```
fabio@fabio-X555LA: ~  
HP J4813A ProCurve Switch 2524  
Software revision F.05.79  
Copyright (C) 1991-2014 Hewlett-Packard Co. All Rights Reserved.  
RESTRICTED RIGHTS LEGEND  
Use, duplication, or disclosure by the Government is subject to restrictions  
as set forth in subdivision (b) (3) (ii) of the Rights in Technical Data and  
Computer Software clause at 52.227-7013.  
HEWLETT-PACKARD COMPANY, 3000 Hanover St., Palo Alto, CA 94303  
Press any key to continue
```

Per validare quanto detto sopra si riporta uno screenshot del lancio dei comandi `arp -n` e `route -n`, che dimostrano come l'indirizzo assegnato allo switch sia quello prefissato e che vi è la connessione desiderata.

```
fabio@fabio-X555LA:~$ arp -n
Indirizzo TipoHW IndirizzoHW Flag Maschera Interfaccia
192.168.1.153 ether 00:30:6e:67:13:c0 C enp2s
0
```

4.3.2 Configurazione interfacce

Una volta all'interno dello switch la prima cosa da fare è quella di osservare lo stato delle vlan per vedere quali sono attive, e quali porte gli sono assegnate. Per fare questo ci si può servire del comando `show vlan` e nello specifico `show vlan "n°della vlan"`.

```
HP ProCurve Switch 2524(config)# show vlan

Status and Counters - VLAN Information

Maximum VLANs to support : 8
Primary VLAN : DEFAULT_VLAN
Management VLAN :

802.1Q VLAN ID Name          Status
-----
1          DEFAULT_VLAN  Static
3          to_R21&R22  Static

HP ProCurve Switch 2524(config)#
```

Nell'immagine di sotto si può vedere come alla VLAN 3 siano state assegnate le porte 12,13,14 in modalità untagged, e come lo status delle porte sia Up.

```
HP ProCurve Switch 2524# show vlan 3

Status and Counters - VLAN Information - Ports - VLAN 3

802.1Q VLAN ID : 3
Name           : to_R21&R22
Status         : Static

Port Information Mode      Unknown VLAN Status
-----
12              Untagged Learn        Up
13              Untagged Learn        Up
14              Untagged Learn        Up
```

Ora si descrivono i passi effettuati come procedura per creare la VLAN 3 e assegnargli le porte e gli indirizzi necessari per il funzionamento della rete con lo Switch HP Procurve.

Per prima cosa si entra nella modalità di configurazione del dispositivo. Questo si attua tramite il comando "configure".

Ora si può creare la vlan desiderata semplicemente digitando : "vlan <n°vlan>". Fatto questo, si può dare un nome alla vlan , assegnarle le porte specificandone la modalità e assegnarle un indirizzo IP.

```
HP ProCurve Switch 2524> enable
HP ProCurve Switch 2524# configure
HP ProCurve Switch 2524(config)# vlan 3
HP ProCurve Switch 2524(vlan-3)# name to_R1.1&R2.2
HP ProCurve Switch 2524(vlan-3)# show vlan 3

Status and Counters - VLAN Information - Ports - VLAN 3

802.1Q VLAN ID : 3
Name           : to_R1.1&R2.2
Status         : Static
```

Come indirizzo IP si assegnerà il 192.168.20.3/29 di modo che lo Switch possa comunicare con i Raspberry dell'area 2.

Saranno sufficienti 3 porte che si utilizzano in modalità untagged di modo che siano utilizzate tutte per la stessa vlan.

```

HP ProCurve Switch 2524# config
HP ProCurve Switch 2524(config)# vlan 3
HP ProCurve Switch 2524(vlan-3)# ip address 192.168.20.3 255.255.255.248
HP ProCurve Switch 2524(vlan-3)# exit
HP ProCurve Switch 2524(config)# exit
HP ProCurve Switch 2524# show ip

```

Internet (IP) Service

```

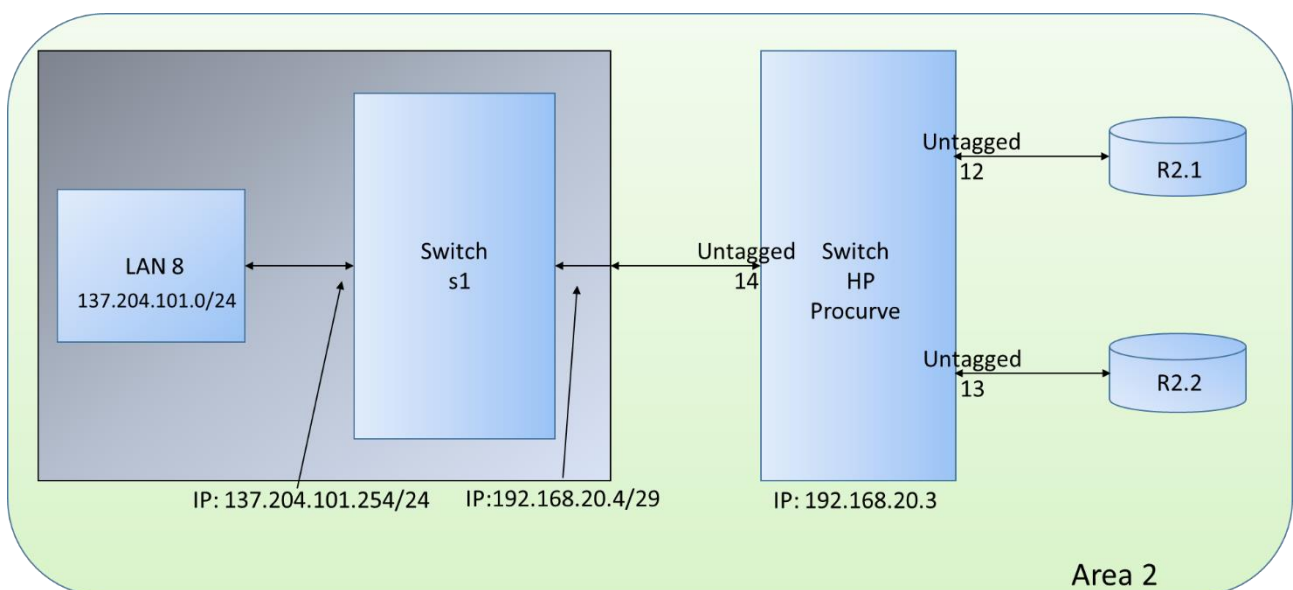
Default Gateway :
Default TTL      : 64

```

VLAN	IP Config	IP Address	Subnet Mask
-----+-----			
DEFAULT_VLAN	Manual	192.168.1.153	255.255.255.0
to_R21&R22	Manual	192.168.20.3	255.255.255.248

4.3.3 Connessione tra VLAN Switch e Raspberry

Il prossimo passo sarà quello di andare a modificare il raspberry R2.3 siccome la precedente topologia prevedeva che lo Switch fisico, fosse virtuale e all'interno del Raspberry. Invece bisognerà mantenere l'altro Switch che serviva per collegare il Raspberry con un host della LAN 8.



Per fare questo si modificano i file di configurazione del pacchetto Quagga con i

quali si era strutturata la base per la comunicazione. Si accede quindi al Raspberry R2.3, e poi al file Zebra tramite il seguente script:

```
Sudo nano /etc/quagga/zebra.conf
```

Qui bisognerà cancellare la parte riguardante il vecchio switch virtuale s2 siccome non più presente, e occorrerà aggiungere all'interfaccia di s1, l'indirizzo per poter comunicare con la vlan dello Switch HP.

Nel file di configurazione OSPF invece sarà sufficiente cancellare la stringa "interface s2" per lo stesso motivo spiegato precedentemente.

```
! *- zebra *-  
  
hostname Router23  
password zebra  
enable password zebra  
  
interface lo  
description loopback  
ip address 127.0.0.1/8  
ip forwarding  
  
interface s1  
description LAN8  
ip address 137.204.101.254/24  
ip address 192.168.20.4/29  
ip forwarding  
  
log file /var/log/quagga/zebra.log
```



```
! *- ospf *-  
hostname Router23  
password zebra  
interface s1  
interface s2  
router ospf  
network 137.204.101.0/24 area 2  
network 192.168.20.0/29 area 2  
log stdout  
log file /var/log/quagga/ospfd.log
```

Infine si aggiunge l'indirizzo nella tabella di route del dispositivo e si abilita l'ip forwarding.

```
Sudo ip addr add 192.168.20.4/29 dev s1  
Sudo sysctl -w net.ipv4.ip_forward = 1
```

Fatto questo si può lanciare il tool mininet per creare il primo switch e settare l'IP dell'host della LAN 8 e aggiungere la route di default gateway allo switch.

```
Sudo mn -switch=ovs -topo=single,1  
Mininet>py h1.setIP('137.204.101.1/24')  
Mininet>h1 route add default gw 137.204.101.254
```

Ora si verifica la raggiungibilità tra i Raspberry dell'area 2 e lo switch tramite il comando ping.

Ping: Switch HP (192.168.20.3) ———▶ R2.2 (192.168.20.2)

```
HP ProCurve Switch 2524# ping 192.168.20.2  
192.168.20.2 is alive, time = 10 ms
```

Ping: Switch HP (192.168.20.3) ———▶ R2.1 (192.168.20.1)

```
HP ProCurve Switch 2524# ping 192.168.20.1  
192.168.20.1 is alive, time = 10 ms
```

Ping: R2.3 (192.168.20.4) ———▶ Switch HP (192.168.20.3)

```
pi@raspberrypi ~ $ ping 192.168.20.3
PING 192.168.20.3 (192.168.20.3) 56(84) bytes of data.
64 bytes from 192.168.20.3: icmp_req=1 ttl=64 time=37.7 ms
64 bytes from 192.168.20.3: icmp_req=2 ttl=64 time=6.63 ms
64 bytes from 192.168.20.3: icmp_req=3 ttl=64 time=6.11 ms
64 bytes from 192.168.20.3: icmp_req=4 ttl=64 time=5.87 ms
64 bytes from 192.168.20.3: icmp_req=5 ttl=64 time=5.85 ms
64 bytes from 192.168.20.3: icmp_req=6 ttl=64 time=3.94 ms
64 bytes from 192.168.20.3: icmp_req=7 ttl=64 time=3.80 ms
^C
--- 192.168.20.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 3.803/10.001/37.790/11.390 ms
pi@raspberrypi ~ $ _
```

4.4 Routing tables e test connettività

In questo capitolo vengono riportate alcune tabelle di routing dei principali router che sono stati modificati nella nuova topologia, i relativi file di configurazione di mininet e di quagga (ospfd e zebra), la running-config completa del router Cisco e dello switch HP. Vengono inoltre verificate la raggiungibilità di ogni nodo attraverso il comando traceroute tra host lan, e altri comandi utili per verificare la consistenza della rete implementata.

Router R1.1:

```
raspberrypi# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
O - OSPF, I - IS-IS, B - BGP, A - Babel,
> - selected route, * - FIB route

O>* 20.0.0.0/30 [110/20] via 192.168.10.6, s1, 00:42:36
O>* 20.0.0.4/30 [110/31] via 192.168.10.6, s1, 01:24:02
O>* 20.0.0.8/30 [110/31] via 192.168.10.6, s1, 01:24:02
C>* 127.0.0.0/8 is directly connected, lo
O>* 137.204.1.0/24 [110/20] via 192.168.10.2, s1, 00:02:04
O 137.204.64.0/24 [110/10] is directly connected, s1, 00:00:40
C>* 137.204.64.0/24 is directly connected, s1,
O 137.204.65.0/24 [110/10] is directly connected, s2, 00:00:40
C>* 137.204.65.0/24 is directly connected, s2,
O 137.204.66.0/24 [110/10] is directly connected, s3, 00:00:40
C>* 137.204.66.0/24 is directly connected, s3,
O 137.204.67.0/24 [110/10] is directly connected, s4, 00:00:40
C>* 137.204.67.0/24 is directly connected, s4,
O>* 137.204.100.0/24 [110/41] via 192.168.10.6, s1, 00:05:14
O>* 137.204.101.0/24 [110/61] via 192.168.10.6, s1, 00:09:41
O>* 192.168.10.0/30 [110/10] is directly connected, s1, 00:00:40
O>* 192.168.10.4/30 [110/10] is directly connected, s1, 00:00:40
O>* 192.168.10.8/30 [110/20] via 192.168.10.6, s1, 00:01:18
O>* 192.168.20.0/29 [110/31] via 192.168.10.6, s1, 00:27:48
```

```
! *- zebra *-  
hostname Router1  
password zebra  
enable password zebra  
  
interface lo  
description loopback  
ip address 127.0.0.1/8  
ip forwarding  
  
interface s1  
ip address 192.168.10.1/30  
ip address 192.168.10.5/30  
ip address 137.204.64.254/24  
ip forwarding  
  
interface s2  
ip address 137.204.65.254/24  
  
interface s3  
ip address 137.204.66.254/24  
  
interface s4  
ip address 137.204.67.254/24  
  
log file /var/log/quagga/zebra.log
```

```
*- ospf *-  
  
hostname Router1  
password zebra  
  
interface s1  
interface s2  
interface s3  
interface s4  
  
router ospf  
network 192.168.10.0/30 area 1  
network 192.168.10.4/30 area 1  
network 137.204.64.0/24 area 1  
network 137.204.65.0/24 area 1  
network 137.204.66.0/24 area 1  
network 137.204.67.0/24 area 1  
  
log stdout  
log file /var/log/quagga/ospfd.log
```

Router R2.2:

```
raspberrypi# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
O - OSPF, I - IS-IS, B - BGP, A - Babel,
> - selected route, * - FIB route

O>* 20.0.0.0/30 [110/21] via 20.0.0.9, s1, 00:02:34
O>* 20.0.0.4/30 [110/21] via 20.0.0.9, s1, 00:02:34
O 20.0.0.8/30 [110/10] is directly connected, s1, 00:03:24
C>* 20.0.0.8/30 is directly connected, s1
C>* 127.0.0.0/8 is directly connected, lo
O>* 137.204.64.0/24 [110/41] via 20.0.0.9, s1, 00:02:05
O>* 137.204.65.0/24 [110/41] via 20.0.0.9, s1, 00:02:05
O>* 137.204.66.0/24 [110/41] via 20.0.0.9, s1, 00:02:05
O>* 137.204.67.0/24 [110/41] via 20.0.0.9, s1, 00:02:05
O 137.204.100.0/24 [110/10] is directly connected, s1, 00:00:18
C>* 137.204.100.0/24 is directly connected, s1
O>* 137.204.101.0/24 [110/20] via 192.168.20.3, s1, 00:01:24
O>* 137.204.1.0/24 [110/41] via 20.0.0.9, s1, 00:02:05
O>* 192.168.10.0/30 [110/41] via 20.0.0.9, s1, 00:02:05
O>* 192.168.10.4/30 [110/31] via 20.0.0.9, s1, 00:02:05
O>* 192.168.10.8/30 [110/41] via 20.0.0.9, s1, 00:02:05
O 192.168.20.0/29 [110/10] is directly connected, s1, 00:03:24
C>* 192.168.20.0/29 is directly connected, s1
```

Router Cisco:

```
Router_R1#show running-config
Building configuration...

Current configuration : 1893 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router_R1
!
```

```
boot-start-marker
boot-end-marker
!
enable secret 5 $1$gERf$PzViSorWPVXgJPkWG2uWz.
!
aaa new-model
!
!
aaa session-id common
!
!
dot11 syslog
!
!
ip cef
!
!
no ip domain lookup
ip domain name netlab.ingce.unibo.it
!
multilink bundle-name authenticated
mpls label range 4000 4999
!
!
username admin privilege 15 secret 5 $1$3473$vTN3d8Fw.Rp24bo2wGzAG/
username labreti secret 5 $1$7WNU$xkxGKsshPo28njv2cCh2J1
!
!
!
```

```
archive
log config
  hidekeys
!
!
ip ssh time-out 60
ip ssh authentication-retries 2
ip ssh version 2
!
!
!
interface Loopback0
  ip address 10.0.0.1 255.255.255.255
!
interface FastEthernet0
  no ip address
  ip ospf cost 1
  duplex auto
  speed auto
!
interface BRI0
  no ip address
  encapsulation hdlc
  shutdown
!
interface FastEthernet1
!
interface FastEthernet2
!
```

```
interface FastEthernet3
  switchport access vlan 2
!
interface FastEthernet4
  switchport access vlan 2
!
interface FastEthernet5
  switchport access vlan 3
!
interface FastEthernet6
  switchport access vlan 3
!
interface FastEthernet7
  switchport access vlan 4
!
interface FastEthernet8
  switchport access vlan 4
!
interface ATM0
  no ip address
  shutdown
  no atm ilmi-keepalive
  dsl operating-mode auto
!
!
interface Vlan1
  ip address 137.204.64.254 255.255.255.0
!
```



```
interface Vlan3
ip address 20.0.0.5 255.255.255.252
!
interface Vlan4
ip address 20.0.0.9 255.255.255.252
!
interface Vlan2
ip address 20.0.0.2 255.255.255.252
!
router ospf 109
log-adjacency-changes
network 20.0.0.0 0.0.0.3 area 0
network 20.0.0.4 0.0.0.3 area 0
network 20.0.0.8 0.0.0.3 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
control-plane
!
!
```

```
line con 0
line aux 0
line vty 0 4
  transport input ssh
line vty 5 15
  transport input ssh
!
end
```

```
Router_R1#show ip ospf route
```

```
      OSPF Router with ID (10.0.0.1) (Process ID 109)
```

```
Area BACKBONE(0)
```

```
Intra-area Route List
```

- * 20.0.0.4/30, Intra, cost 1, area 0, Connected
 via 20.0.0.5, Vlan3
- * 20.0.0.0/30, Intra, cost 1, area 0, Connected
 via 20.0.0.2, Vlan2
- * 20.0.0.8/30, Intra, cost 1, area 0, Connected
 via 20.0.0.9, Vlan4

```
Intra-area Router Path List
```

- i 20.0.0.6 [1] via 20.0.0.6, Vlan3, ABR, Area 0, SPF 3
- i 20.0.0.1 [1] via 20.0.0.1, Vlan2, ABR, Area 0, SPF 3

Inter-area Route List

```
*> 192.168.10.0/30, Inter, cost 21, area 0
    Via 20.0.0.1, Vlan2
*> 192.168.10.4/30, Inter, cost 11, area 0
    via 20.0.0.1, Vlan2
*> 192.168.10.8/30, Inter, cost 11, area 0
    via 20.0.0.1, Vlan2
*> 192.168.20.0/29, Inter, cost 11, area 0
    via 20.0.0.6, Vlan3
```

```
Router_R1#show ip ospf database
```

OSPF Router with ID (10.0.0.1) (Process ID 109)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.0.0.1	10.0.0.1	1588	0x8000000C	0x00A1D7	3
20.0.0.1	20.0.0.1	1594	0x8000027F	0x00284C	1
20.0.0.6	20.0.0.6	1593	0x8000017F	0x00530E	1
20.0.0.10	20.0.0.10	1592	0x8000007F	0x00753A	1

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
20.0.0.1	20.0.0.1	1594	0x8000000F	0x009C68
20.0.0.6	20.0.0.6	1593	0x80000005	0x00CE31
20.0.0.10	20.0.0.10	1592	0x80000003	0x00AD92

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.10.0	20.0.0.1	1432	0x8000028C	0x00F4B7
192.168.10.4	20.0.0.1	1779	0x8000026F	0x00D781
192.168.10.8	20.0.0.1	180	0x8000026E	0x00B1A4
192.168.20.0	20.0.0.6	1617	0x8000017C	0x004402

Router_R1#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

137.204.0.0/24 is subnetted, 7 subnets

O IA 137.204.64.0 [110/31] via 20.0.0.1, 01:32:17, Vlan2

O IA 137.204.65.0 [110/31] via 20.0.0.1, 01:32:17, Vlan2

O IA 137.204.66.0 [110/31] via 20.0.0.1, 01:32:17, Vlan2

O IA 137.204.67.0 [110/31] via 20.0.0.1, 01:32:17, Vlan2

O IA 137.204.1.0 [110/31] via 20.0.0.1, 1:32:17, Vlan2

O IA 137.204.100.0 [110/21] via 20.0.0.10, 00:27:28, Vlan4

O IA 137.204.101.0 [110/41] via 20.0.0.6, 01:57:42, Vlan3

20.0.0.0/30 is subnetted, 3 subnets

C 20.0.0.4 is directly connected, Vlan3

C 20.0.0.0 is directly connected, Vlan2

C 20.0.0.8 is directly connected, Vlan4

192.168.10.0/30 is subnetted, 2 subnets

O IA 192.168.10.4 [110/11] via 20.0.0.1, 00:34:01, Vlan2

O IA 192.168.10.8 [110/11] via 20.0.0.1, 00:34:01, Vlan2

192.168.20.0/29 is subnetted, 1 subnets

O IA 192.168.20.0 [110/11] via 20.0.0.6, 00:27:40, Vlan3

10.0.0.0/32 is subnetted, 1 subnets

C 10.0.0.1 is directly connected, Loopback0

Switch HP:

HP ProCurve Switch 2524# show running-config

Running configuration:

; J4813A Configuration Editor; Created on release #F.05.79

hostname "HP ProCurve Switch 2524"

cdp run

interface 15

disable

exit

snmp-server community "public" Unrestricted

```

vlan 1
  name "DEFAULT_VLAN"
  untagged 1-11,15-26
  ip address 192.168.1.153 255.255.255.0
  no untagged 12-14
  exit
vlan 3
  name "to_R21&R22"
  untagged 12-14
  ip address 192.168.20.3 255.255.255.248
  exit
no aaa port-access authenticator active

```

Si applica ora il comando *traceroute* da un host di una lan, ad esempio la 1, ad uno di un'altra lan, per esempio la 7, di modo da vedere la comunicazione tramite hop tra le 2 lan più esterne.

```

mininet>h1 traceroute -n 137.204.101.1
traceroute to 137.204.100.1 (137.204.100.1) 30 hops max, 60 byte packets
 1 137.204.64.254 14.7648ms 15.339ms 17.828ms
 2 192.168.10.6 63.662ms 56.342ms 67.759ms
 3 20.0.0.2 81.314ms 86.973ms 90.436ms
 4 20.0.0.6 85.463ms 88.942ms 92.346ms
 5 192.168.20.4 242.683ms 253.583ms 251.124ms
 6 137.204.101.1 272.991ms 285.534ms 289.895ms

```

Come si può vedere dagli hop passati, il pacchetto percorre correttamente tutti i nodi della rete, arrivando alla destinazione della rete virtuale di R2.3. Sempre con il comando *traceroute* è interessante eseguire altre prove per verificare la validità del progetto. Traceroute da host h1 della lan1 a host h1 della lan2:

```
mininet>h1 traceroute -n 137.204.65.1
traceroute to 137.204.65.1 (137.204.65.1) 30 hops max, 60 byte packets
 1 137.204.64.254 8.390ms 34.515ms 30.102ms
 2 137.205.65.1 162.947ms 160.600ms 155.837ms
```

E' interessante notare come ora il flusso di dati passi attraverso i due switch s1 e s2 anziché arrivare direttamente dalla lan1 alla lan2.

Ultima prova che si effettuerà sarà il traceroute dalla lan1 alla lan5, rimanendo così all'interno della stessa area ospf.

Ecco il risultato a comando eseguito:

```
mininet>h1 traceroute -n 137.204.1.1
traceroute to 137.204.101.1 (137.204.101.1) 30 hops max, 60 byte packets
 1 137.204.64.254 13.728ms 14.374ms 17.452ms
 2 192.168.10.2 28.413ms 123.081ms 151.833ms
 3 137.204.1.1 241.041ms 259.561ms 259.212ms
```

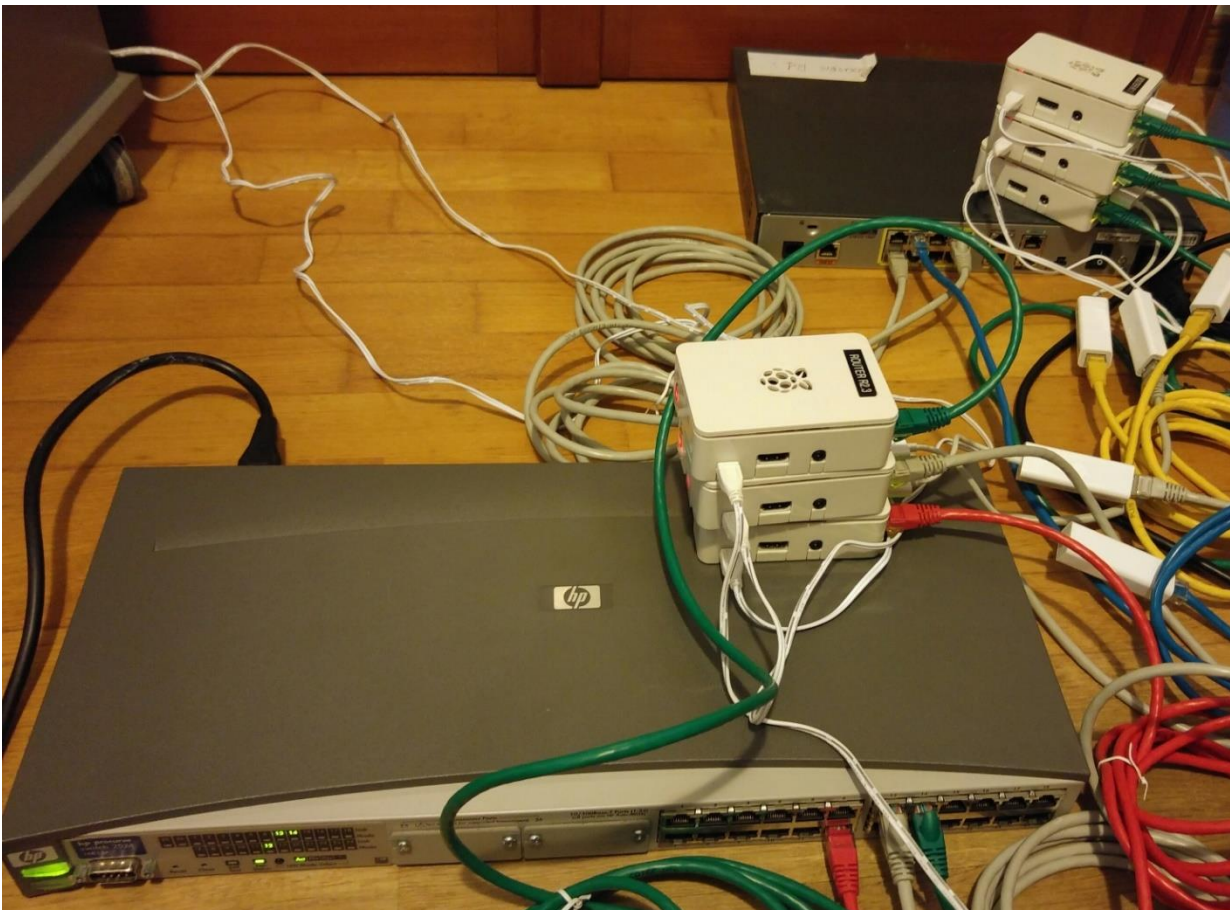
Si nota che il pacchetto esce dal nodo R1.1 e arriva al router R1.2 nella porta eth0(192.168.10.2) e così facendo entra nella rete virtuale arrivando a destinazione.

5. Conclusioni

Costruire questa nuova topologia è stato molto interessante per via della varietà di configurazioni che si sono dovute affrontare, e per osservare come grazie al modello ISO/OSI sia possibile implementare strutture di reti con dispositivi di natura diversa, sfruttando in questo caso i livelli 2 e 3 del suddetto modello.

Questo è un punto di partenza per creare nuove reti sempre più complesse, facendo uso di nuovi componenti come i Raspberry, che grazie alle dimensioni e al basso costo, possono essere inseriti in qualsiasi ambito, dando vita a nuove reti sempre più duttili.

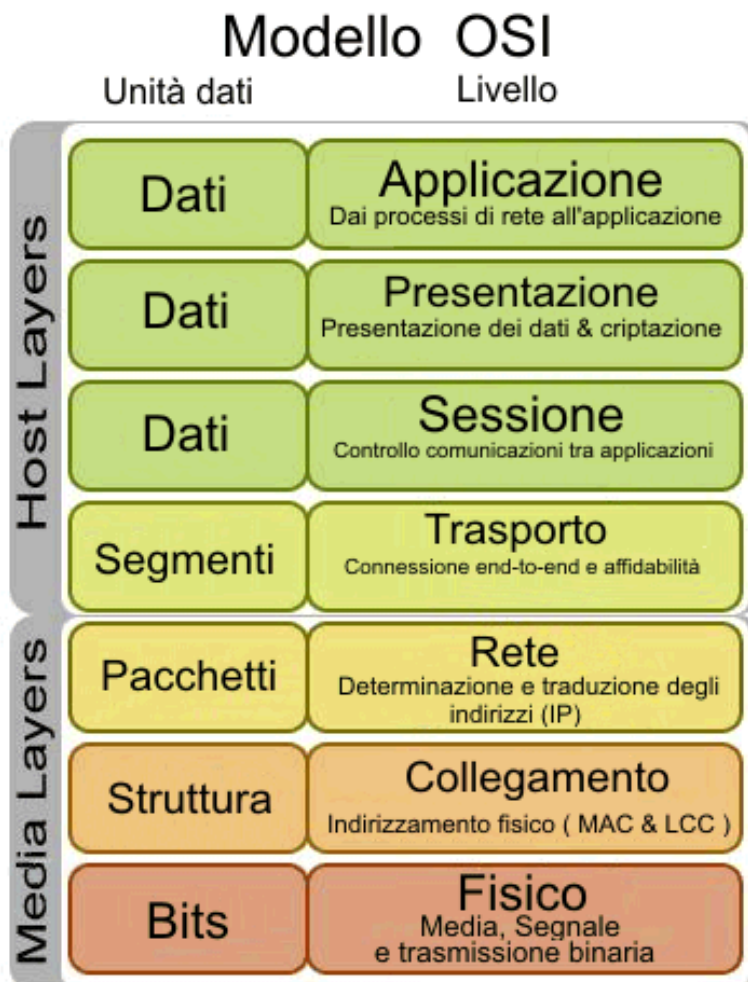
Si possono quindi creare nuove reti completamente da zero, oppure modificarne di pre-esistenti a seconda dello scopo prefissato e degli strumenti a disposizione, abbassando così i costi grazie alla facilità di implementazione e al basso costo dei Raspberry.



6. Appendice

6.1 ISO/OSI

L'Open System Interconnection (OSI) è uno standard per la gestione di calcolatori ideato dalla International Organization for Standardization (ISO). E' stato impiegato nel 1978, e si basa su un modello formato da **7 strati**, ognuno composto da un protocollo per **regolare il trasferimento in blocchi di rete a commutazione di pacchetto**.



Livello 1 – Fisico:

A questo livello si controllano le trasmissioni dei dati sulla base di tensione e forma del segnale. Vengono quindi stabiliti i valori logici dei dati abbinati alle tensioni e ci si preoccupa ad esempio della modulazione e della codificazione nella trasmissione dei bit.

Livello 2 – DataLink:

Il suo compito è quello di creare i pacchetti per trasferire i dati attraverso la rete. Ogni dato viene quindi frammentato, e modificato con un header e una tail, con la funzione di controllo, e ad ogni pacchetto spedito viene fatto corrispondere un segnale che viene denominato Ack. E' il livello su cui agisce lo switch di cui ci si è occupati in questa tesi.

Livello 3 – Rete:

E' il livello che permette che gli strati superiori siano indipendenti da quelli fisici. Si occupa del routing (instradamento) dei pacchetti e della conversione di quest'ultimi. Per gestire l'instradamento sfrutta i modelli di TCP/IP. Nel corso di questa tesi si è utilizzato molto questo protocollo sia per la progettazione della rete tramite Raspberry, che con il router Cisco, che ha infatti la funzione base di instradare i pacchetti.

Livello 4 – Trasporto:

Il suo compito è quello del trasporto fisico dei dati. Si occupa quindi di tutto ciò che è inerente alla connessione tra due host nello scambio dei dati, infatti la stabiliscono, la mantengono e la terminano una volta avvenuto il trasferimento.

Livello 5 – Sessione:

Il suo obiettivo è quello di gestire la connessione tra applicazioni cooperanti (sessioni). Consente anche di aggiungere funzioni avanzate come i tokens per riconoscere univocamente i dispositivi.

Livello 6 – Presentazione:

I protocolli di questo layer si occupano di adattare ad un formato standard le applicazioni e di fornire servizi per le applicazioni come crittografia e formattazione.

Livello 7 – Applicazione

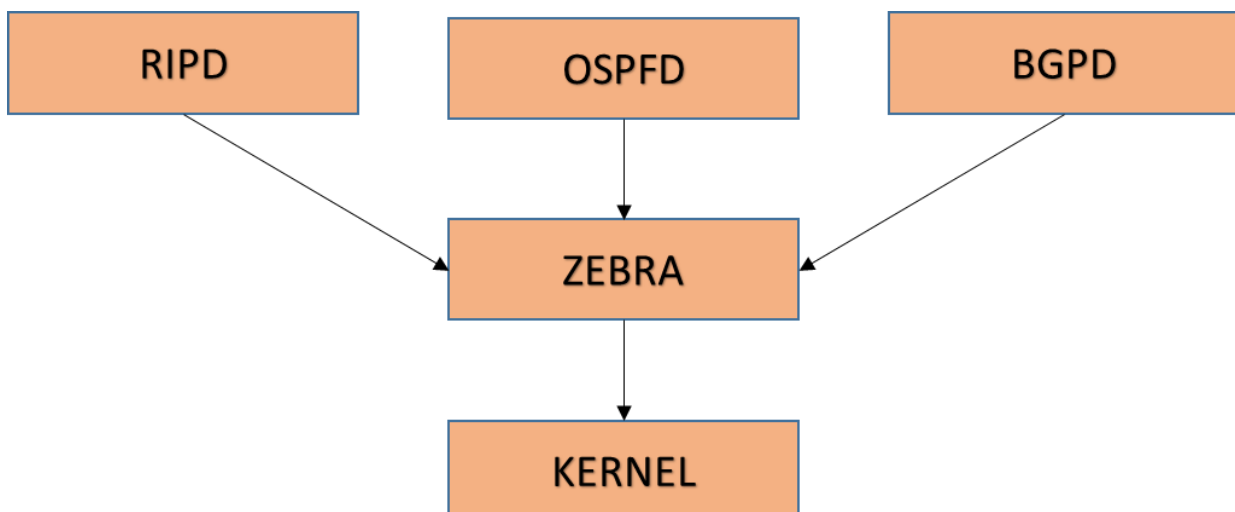
E' l'ultimo livello e quello più vicino all'utente. Qui i protocolli operano direttamente con i programmi e i software che hanno al loro interno moduli di rete. Il compito principale di questo layer è quindi la gestione della comunicazione a livello applicativo.

6.2 Pacchetto Quagga

Il pacchetto quagga è un software opensource di cui si è serviti per programmare i Raspberry collocandoli in aree diverse. Questo è stato possibile siccome questo tool supporta il routing di livello 3 con protocolli come RIPv1, OSPFv2, RIPv2, BGP-4. Oltre ai protocolli classici di routing IPv4, supporta anche quelli più recenti come l'IPv6.

Quagga funziona come un router dedicato in cui vengono scambiate informazioni con gli altri routing tramite una tabella speciale di instradamento presente nel kernel (kernel routing table). Per modificare questa tabella si fa uso del demone Zebra, che è il software di gestione degli indirizzi. Una volta installato il programma, sarà necessario attivare il demone che si vuole utilizzare attraverso la modifica del file presente al percorso `/etc/quagga/daemons`, dove bisognerà attivare i demoni desiderati.

Una volta fatto questo passaggio, sarà sufficiente accedere ai file di configurazione del demone, che nel nostro caso saranno per zebra `/etc/quagga/zebra.conf` mentre per ospf sarà `/etc/quagga/ospfd.conf`, e si potrà modificare la rete secondo le esigenze stabilite in fase di progetto.



7. Bibliografia e Ringraziamenti

Libri

Pattavina A., Reti di telecomunicazioni. Networking e Internet, Milano, McGraw-Hill, 2003

Tesi

Sperimentazione di protocolli di routing Ip su piattaforme raspberry pi, di Nicola Sparnacci

Manuali

Router Cisco 1801. Disponibile online:

<http://www.cisco.com/c/en/us/products/routers/1801-integrated-services-router-isr/index.html>.

Switch HP Procurve 2524. Disponibile online: <http://whp->

[aus1.cold.extweb.hp.com/pub/networking/software/2500-MgmtConfig-Oct2005-59692354.pdf](http://whp-aus1.cold.extweb.hp.com/pub/networking/software/2500-MgmtConfig-Oct2005-59692354.pdf)

Online

Comandi principali per la configurazione di router Cisco. Disponibile online:

<http://netgroup.polito.it/teaching/prlc/Cisco - Interfaces and routing.pdf>

HP Procurve Networking Advanced CLI Command Reference. Disponibile Online:

<http://www.sysadmintutorials.com/tutorials/hp/hp-procurve-advanced-cli-commands-reference/>

Configuring HP ProCurve Switch. Disponibile Online:

<http://blog.petrilopia.net/info/configuring-hp-procurve-switch/>

Install and configure DHCP server on CentOS 7 / Ubuntu 14.04. Disponibile Online:

<http://www.itzgeek.com/how-tos/linux/ubuntu-how-tos/install-and-configure-dhcp-server-on-centos-7-ubuntu-14-04.html>

Chiarimenti sulle VLAN: default vlan, PVID, porte tagged e untagged. Disponibile

Online: <http://www.netsetup.it/networking-livello2/porte-tagged-untagged>

Walkthrough Mininet: <http://mininet.org/walkthrough/>

Router e Switch, Instradamento, Protocollo OSPF. Disponibile online:

<https://it.wikipedia.org/wiki/Instradamento>

<https://it.wikipedia.org/wiki/Router>

<https://it.wikipedia.org/wiki/Switch>

Probabilmente ci vorrebbero intere pagine per parlare di chi mi ha sostenuto in questi anni trascorsi sui libri, ma anche ad imparare ad essere uomo, tra vittorie e sconfitte, tra conquiste e delusioni. Volevo cominciare ringraziando la mia famiglia, che è un punto di fondamentale importanza, perché nonostante tutto, so che posso contare sempre sui miei genitori, e i miei parenti più stretti. Poi viene Chiara, la mia ragazza che mi è stata vicina in questi ultimi mesi in cui si è finalmente realizzato il mio obiettivo, e credo sia anche un po' merito suo se finalmente ce l'ho fatta.

Poi volevo ringraziare gli amici di una vita: Mattia Ravaglia, Mattia Errani, Matteo Picchietti, Tommaso Allegri, Nicola Sparnacci, Chiara Fariselli, Eleonora Bucci, Camilla Casadio, Matilde Scanferla, Carlotta Casadio. Loro mi hanno assistito durante questa avventura, e mi hanno fatto tirare fuori la forza che spesso non pensavo nemmeno di avere. Molto del merito è anche loro se ho raggiunto questo traguardo, perché mi hanno insegnato quanto sia importante il valore dell'amicizia, e quanto sia fondamentale sapere di poter contare su delle persone che nei momenti di difficoltà semplicemente sanno esserci. Sono la mia seconda famiglia.

Ci sono poi tante altre persone che dovrei citare, che sono state parte integrante di questo pezzo della mia vita, durato quasi cinque anni. Molti di loro non li sento quasi più, ma li porto sempre con me, come i ricordi che ho con loro, li porto sempre dentro il mio cuore.

L'ultimo ringraziamento va a Walter Cerroni, il prof che mi ha seguito con grande pazienza durante lo sviluppo del tirocinio e della tesi, e per questo non posso fare a meno di ringraziarlo.