



**UNIVERSITA' POLITECNICA DELLE MARCHE**

**FACOLTA' DI INGEGNERIA**

---

Corso di Laurea magistrale in Ingegneria Elettronica

**Rete Lan Cooperlat: simulazione in ambiente GNS3 ed  
implementazione di monitoraggio SNMP.**

**Cooperlat LAN: simulation in GNS3 environment with SNMP  
monitoring implementation.**

Relatore:

Prof. **Gambi Ennio**

Tesi di Laurea di:

**Bucci Marco**

Correlatore:

Ing. **De Santis Adelmo**

A.A. 2020 / 2021



## Sommario

1.	INTRODUZIONE.....	4
2.	IL PROTOCOLLO TCP/IP.....	7
2.1	TCP/IP - <i>Layer</i> Applicazione .....	11
2.2	TCP/IP - <i>Layer</i> di Trasporto .....	12
2.3	TCP/IP – <i>Layer</i> Rete .....	13
2.4	TCP/IP – <i>Layer</i> Data Link e <i>Layer</i> Fisico .....	19
2.5	Incapsulamento dei dati .....	20
3.	FONDAMENTI DI LAN ETHERNET .....	26
3.2	Invio dei dati su reti Ethernet.....	33
3.3	Le Virtual LAN.....	39
3.4	Lo <i>Spanning Tree Protocol</i> .....	43
3.5	Il Per-VLAN <i>Spanning Tree Protocol</i> .....	60
4.	IL SIMULATORE GNS3 .....	64
4.1	Cos'è GNS3, come è strutturato .....	65
4.2	Vantaggi e criticità nell'uso di GNS3 .....	70
4.3	Installazione GNS3 e prima topologia .....	72
5.	LA RETE "COOPERLAT" .....	88
5.1	LAN Cooperlat .....	90
5.2	Ricognizione della LAN .....	93
5.3	Metodo di lavoro .....	95
5.4	Simulazione della rete Cooperlat in GNS3.....	103
6.	MONITORAGGIO DELLE PRESTAZIONI DELLA RETE .....	119

6.1	I vantaggi nella simulazione delle reti .....	131
6.2	Risultati del monitoraggio della rete simulata.....	132
7.	CONCLUSIONI.....	147
8.	RINGRAZIAMENTI .....	148
9.	INDICE DELLE IMMAGINI.....	149
10.	BIBLIOGRAFIA .....	151
11.	SITOGRAFIA.....	152
12.	APPENDICE.....	153
A.	Codice IOS – <i>switchport</i> .....	153
B.	Codice IOS - indirizzo IP su VLAN interface .....	153
C.	Codice IOS - backup su server TFTP .....	154
D.	<i>Command Line Terminal</i> Ubuntu - server FTP .....	154
E.	Codice IOS - <i>PortChannel</i> .....	155
F.	Codice IOS - VTP server e VLAN .....	155
G.	Codice IOS - VTP client .....	158
H.	Codice IOS - <i>Community SNMP name</i> .....	158

# 1. INTRODUZIONE

In uno scenario come un moderno ufficio ogni lavoratore è dotato di un personal computer il quale può comunicare con altri PC all'interno della stessa azienda, grazie ad un'infrastruttura di rete ben organizzata e denominata LAN (*Local Area Network*). Come suggerito dal nome, una LAN consiste in una rete di computer che si estende su aree limitate, quali solitamente un singolo edificio o un gruppo di edifici, e che oltre PC può integrare qualsiasi dispositivo dotato di una scheda di rete come ad esempio stampanti, server, tablet ecc..

La principale caratteristica di una rete LAN è proprio il fatto di essere "locale" ovvero è un sistema proprietario limitato, al quale possono accedere un numero finito di utenti e in genere serve un'area con estensione inferiore al miglio. Una LAN, ad esempio, consente ai lavoratori di operare sullo stesso sistema come se fossero seduti attorno ad un unico computer, che in questo caso è appropriato definire server, ma che effettivamente si possono trovare in uffici più o meno distanti tra loro.

Alcuni vantaggi di una rete LAN sono il poter essere aggiornata, ampliata e riorganizzata senza interruzioni di servizio, ma anche il poter trasmettere rapidamente dati grazie alle elevate disponibilità di banda garantite dalle sempre più performanti tipologie di cablaggio e *switch*.

In questo lavoro di tesi è stata riprodotta mediante il simulatore virtuale di reti GNS3, quella che è la struttura principale della LAN presente nello stabilimento di Jesi (AN) della Azienda Cooperativa Agricola Cooperlat, nella quale ho avuto l'opportunità di svolgere un'attività di tirocinio extracurricolare. L'Azienda mi ha permesso di approfondire la formazione in ambito ICT grazie alla fruizione di un corso atto al conseguimento della certificazione Cisco<sup>1</sup> 200-301 CCNA (*Cisco Certified Network Associate*) e grazie al quale è stato possibile comprendere e approfondire quello che è alla base di questo lavoro.

Il progetto di tesi muove quindi i suoi primi passi partendo da un'attenta e mirata raccolta di informazioni in ambiente di rete LAN Cooperlat per poi passare alla simulazione della stessa in ambiente virtuale. Si procede infine con l'implementazione di un sistema di monitoraggio SNMP, sempre in ambiente virtuale, che coadiuvato ad un'analisi del traffico dati LAN ha lo scopo di evidenziare eventuali criticità e contestualmente l'obiettivo di proporre all'azienda eventuali modifiche atte al miglioramento in termini di robustezza e prestazioni della rete LAN.

---

<sup>1</sup> Cisco System Inc. è leader mondiale nel networking per Internet. L'azienda è stata fondata nel 1984. Oggi, le soluzioni Cisco sono le basi del networking per fornitori di servizi, piccole e medie imprese e clienti aziendali che includono aziende, agenzie governative, servizi pubblici e istituzioni educative.

Nei seguenti capitoli 2 e 3 verranno spiegate le basi fondanti di una LAN, introducendo il modello di rete TCP/IP e lo standard LAN Ethernet.

Al capitolo 4 è presentato il simulatore virtuale di reti GNS3 utilizzato per la riproduzione della topologia LAN Cooperlat. Il capitolo 5 scende invece nell'operatività, esponendo il metodo di lavoro adottato sia nella ricognizione del network aziendale, che nella riproduzione della stessa su simulatore.

Il capitolo 6 tratta l'installazione del sistema di monitoraggio SNMP Nagios XI e i risultati ottenuti dalle modifiche apportate alla topologia, per poi passare al capitolo 7 con le considerazioni e le conclusioni in merito al lavoro di tesi nella sua interezza.

## 2. IL PROTOCOLLO TCP/IP

Oggigiorno tutto il mondo dell'informatica e dell'elettronica utilizza principi e protocolli racchiusi nello standard di rete TCP/IP, ma in passato le cose erano ben diverse. Non esistevano protocolli di rete di nessun tipo, perché erano i principali vendors presenti sul mercato ad elaborare i propri standard e di conseguenza avevano l'enorme limite di essere supportati solo da un numero molto ristretto di dispositivi.

Il risultato fu una enorme frammentazione del mercato ma soprattutto l'impossibilità di integrazione tra i vari sistemi proprietari.

Già a partire dalla fine degli anni '70, l'*International Organization for Standardization* (ISO) si assunse il compito di creare un modello iniziando a lavorare su ciò che sarebbe diventato noto come il modello ISO/OSI (*Open Systems Interconnection*). Tale impresa aveva il nobile obiettivo di standardizzare i protocolli di rete per consentire la comunicazione tra tutti i computer del pianeta.

Durante gli anni '90, prese forma il modello TCP/IP che in breve tempo soppiantò del tutto i modelli proprietari e lo stesso ISO/OSI.

La *Figura 2-a* mostra l'idea generale di quella che fu la transizione storica: da modelli di rete proprietari, allo standard TCP/IP di dominio pubblico.



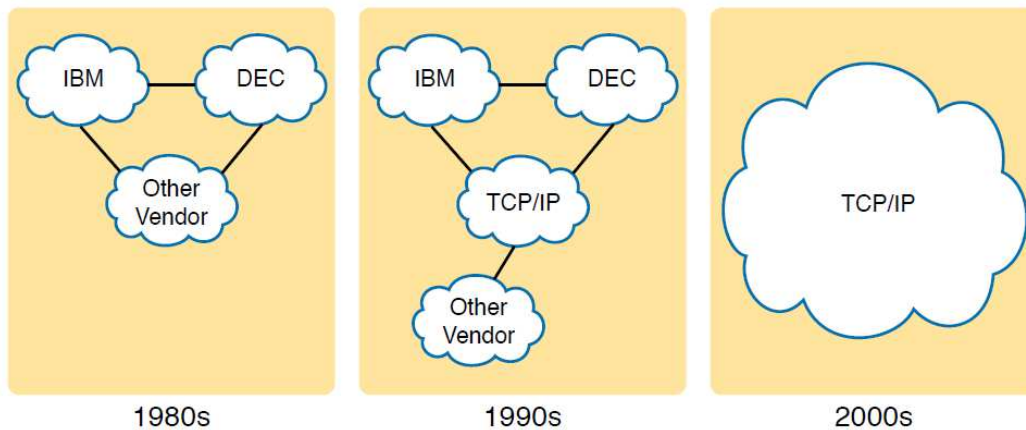


Figura 2-a: Progressione storica, dai modelli di network proprietari allo standard TCP/IP.

Con l'avvento del ventunesimo secolo, domina il modello TCP/IP. Modelli di rete proprietari ancora esistono ma sono stati per lo più scartati a favore di TCP/IP. D'altro canto il modello OSI, il cui sviluppo ha sofferto a causa di un processo di standardizzazione più lento e formale rispetto a TCP/IP, non è mai riuscito ad affermarsi sul mercato.

Il modello TCP/IP definisce e fa riferimento ad un'ampia raccolta di protocolli che consentono ad ogni computer di comunicare con altri; è per questo che il nome stesso del modello è la combinazione di TCP e IP, in quanto sono due dei protocolli cardine che lo costituiscono. Per definire un protocollo, TCP/IP utilizza documenti chiamati *Request For Comments* (RFC), pubblici e facilmente consultabili in internet. Proprio per la sua evidenza pubblica, il modello TCP/IP evita lo spreco di risorse economiche ed intellettuali derivante dal ripetere il lavoro già svolto da altri organi di standardizzazione o consorzi di vendor, semplicemente raggruppando e pubblicando gli standard o i protocolli creati dagli stessi. Ad esempio, *l'Institute of Electrical*

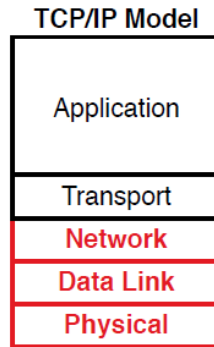
*and Electronic Engineers* (IEEE<sup>2</sup>) definisce le LAN Ethernet, ed effettivamente il modello TCP/IP fa riferimento direttamente alle definizioni date dall'IEEE ad Ethernet.

In estrema sintesi il modello TCP/IP crea un insieme di regole e protocolli che permette a chiunque di prendere un computer, collegarlo con gli appositi cavi o via wireless, e utilizzarlo in rete. Tutto questo è possibile perché il sistema operativo installato sull'hardware, implementa gli standard definiti dal modello.

Per rendere meglio comprensibile il modello TCP/IP, sono state suddivise le sue funzioni in 5 categorie definite *Layers*. Ogni *Layers* include determinati protocolli dediti ad una specifica categoria di funzioni, come mostrato nella *Figura 2-b*.

---

<sup>2</sup> L'*Institute of Electrical and Electronics Engineers* è un'associazione internazionale di scienziati professionisti con l'obiettivo della promozione delle scienze tecnologiche.



*Figura 2-b: Il modello di rete TCP/IP.*

Negli strati inferiori, evidenziati in rosso, il modello TCP/IP concentra tutte le specifiche tecniche su come trasmettere le informazioni da un PC ad un altro, sia che essi siano nella stessa stanza, sia che essi siano agli antipodi del globo terrestre.

Negli strati superiori, invece, vengono definiti i protocolli che forniscono agli utenti un'interfaccia per accedere alle reti.

La logica alla base del modello TCP/IP è che ogni *Layers* svolge dei servizi per il *Layer* superiore ed inoltre può interfacciarsi solo con i *Layers* adiacenti ad esso.

L'unico modo di mettere in comunicazione un medesimo *layer* tra un *host* A e un *host* B, è quello di scendere al *layer* fisico di A, raggiungere B tramite un mezzo trasmissivo, e infine risalire i *Layers* dell'*host* B, fino al *layer* richiesto dalla comunicazione.

Riportiamo ora nella *Tabella 2-a* alcuni esempi dei più utilizzati e conosciuti protocolli facenti parte del modello TCP/IP associati al proprio *Layer* di appartenenza.

Architettura a <i>Layer</i> TCP/IP	Esempi di protocolli
<b>Application</b>	HTTP, POP3, SMTP
<b>Transport</b>	TCP, UDP
<b>Network</b>	IP, ICMP
<b>Data Link &amp; Physical</b>	Ethernet, 802.11 (Wi-Fi)

*Tabella 2-a: Architettura Layer TCP/IP ed esempi di protocolli.*

## 2.1 TCP/IP - *Layer* Applicazione

I protocolli di *layer* di applicazione forniscono alcuni servizi al software applicativo in esecuzione su un computer; in particolare fornisce un'interfaccia tra il software e la rete stessa. Ad esempio il protocollo HTTP, definisce il modo in cui un browser Web possa estrarre il contenuto di una pagina Web da un Web server.

## 2.2 TCP/IP - *Layer* di Trasporto

Il *layer* di trasporto TCP/IP include un numero minore di protocolli rispetto al *layer* Applicazione. In particolare, sono due i più comunemente usati: il *Transmission Control Protocol* (TCP) e lo *User Datagram Protocol* (UDP).

Questa sezione introduce il concetto generale, concentrandosi su un singolo servizio fornito da TCP: il ripristino degli errori di trasmissione.

### 2.2.1 Il ripristino degli errori in TCP

Ogni *layer* che costituisce lo stack TCP/IP fornisce un servizio al *layer* sovrastante e in questo caso il protocollo TCP, appartenente al *Layer* di trasporto, offre un servizio di ripristino degli errori ai protocolli del *layer* applicazione.

Nella *Figura 2-c* ad esempio, Bob e Larry utilizzano il protocollo HTTP per trasferire la home page dal server Larry al browser Bob. Al punto 2 è però indicato un errore di trasmissione e il pacchetto HTTP, contenente parte della pagina web, non può giungere a destinazione.

In questo caso, se non ci fosse un protocollo come TCP, che verifichi che tutte le sequenze di pacchetti giungano correttamente a destinazione, e che (come visibile al punto 4) ripristini l'errore richiedendo una ritrasmissione del

pacchetto corrotto, non sarebbe stato possibile per Bob visualizzare la pagina web.

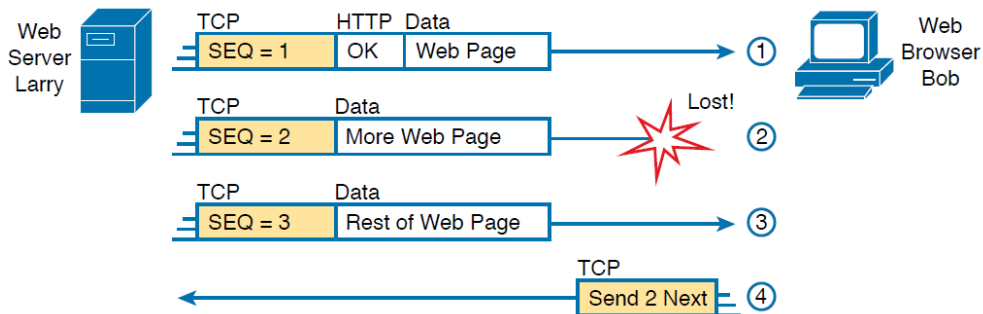


Figura 2-c: Ripristino errore di trasmissione TCP applicato ad HTTP.

L'*header* TCP contiene varie informazioni tra le quali il numero di sequenza (SEQ) di ogni messaggio inviato. La ricezione dei segmenti numero 1 e 3 fa sì che Bob certifichi la non ricezione del segmento numero 2 e che, in automatico, richieda di nuovo l'invio del segmento TCP numero 2 permettendo la correzione dell'errore.

## 2.3 TCP/IP – *Layer* Rete

Il *layer* di rete TCP/IP, analogamente al *layer* trasporto, include un numero ristretto di protocolli, dei quali solo uno spicca per importanza: l'*Internet Protocol* (IP).

Il protocollo IP fornisce diverse funzionalità tra le quali, l'indirizzamento univoco dei PC nella rete mediante l'indirizzo IPv4 (e IPv6), e il routing dei

pacchetti di dati che viaggiano da un PC ad un altro. Per comprendere più facilmente queste due importanti funzioni del protocollo IP, risulta utile un confronto con il sistema di indirizzamento e instradamento del servizio postale.

### **2.3.1 Il protocollo IP, analogia con il servizio postale**

Si supponga di aver scritto due lettere: una da recapitare ad un indirizzo della stessa città da cui viene spedita, e l'altra da recapitare in un altro stato. Scritti gli indirizzi dei destinatari sulle lettere, entrambe sono pronte per essere prese in carico dal servizio postale. Pensandoci non c'è molta differenza nel modo in cui le due lettere sono state preparate, infatti sono state entrambe depositate nella stessa cassetta di posta e ci si aspetta che il servizio postale consegni entrambe le lettere.

Il servizio postale prende in carico ciascuna lettera singolarmente, analizza l'indirizzo del destinatario, e successivamente prende la decisione verso dove inviarle. Per la lettera destinata nella stessa città del mittente, molto probabilmente sarà sufficiente trasferirla su un furgone che provvederà a recapitarla direttamente, mentre per la lettera che deve raggiungere l'altro capo del paese, essa dovrà transitare per molti altri uffici postali prima che si raggiunga l'ufficio più vicino alla destinazione.

La consegna può avere esito positivo solo se vengono prese le giuste decisioni di smistamento in ogni singolo ufficio postale e, per fare questo, è fondamentale che l'indirizzo apposto sulla lettera sia univoco ed universalmente accettato dai servizi postali di tutto il globo.

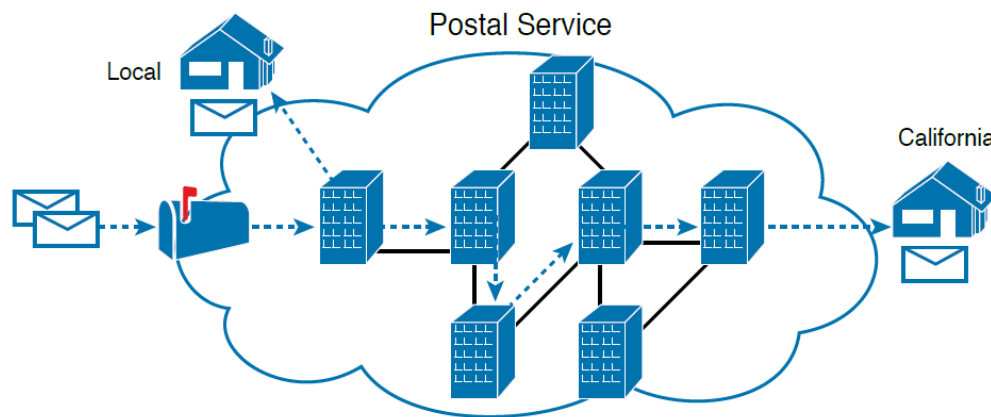


Figura 2-d: Inoltro (Routing) delle lettere da parte del servizio postale.

La persona che invia la lettera si aspetta che venga consegnata e nulla più; essa non ha bisogno di conoscere esattamente i dettagli di quale percorso seguirà la sua lettera. Al contrario il servizio postale non si interessa del contenuto della lettera ma si limita a voler conoscere i dettagli sull'indirizzo e il codice postale del destinatario.

A questo punto è lampante il parallelismo tra *layer* di rete e il servizio postale.

Il *layer* Applicazione è assimilabile all'utente che scrive la lettera, il quale si limita nel comporre il contenuto di un pacchetto IP e si disinteressa del processo di spedizione; è compito del protocollo IP richiedere che ogni *host*



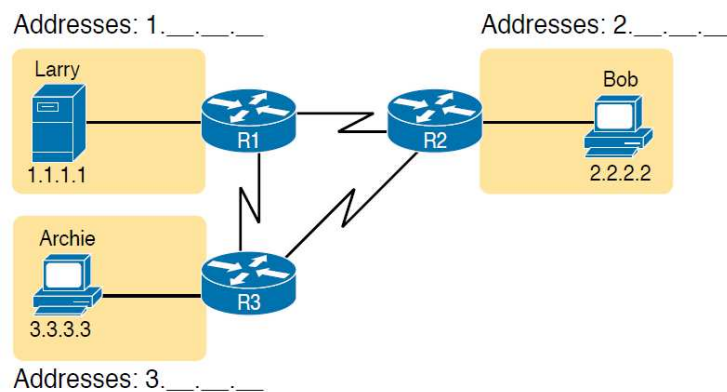
abbia un indirizzo IP univoco, in modo analogo a quanto accade con il servizio postale. L'unicità dell'indirizzamento IP è alla base del processo di routing di un pacchetto IP.

La funzione di smistamento e la scelta di instradamento di un ufficio postale, è l'analogo della funzione svolta dai dispositivi di rete denominati routers.

### 2.3.2 Basi di indirizzamento e routing IP

Ogni dispositivo che utilizza TCP/IP, necessita di un indirizzo IP univoco per poter essere identificato nella rete.

Per comprendere le nozioni di base esaminiamo la *Figura 2-e*, che mostra la situazione del Web server Larry e del browser Web Bob, ora però senza ignorare l'infrastruttura di rete che si interpone tra questi due computer.



*Figura 2-e: Esempio di rete TCP/IP: tre routers con raggruppamento di indirizzi IP.*

Innanzitutto si nota che la *Figura 2-e* mostra alcuni indirizzi IP di esempio associati ai tre *host*. Ogni indirizzo IP è composto da numeri separati da punti. In questo caso specifico Larry utilizza l'indirizzo IP 1.1.1.1 e Bob 2.2.2.2.

Questo stile di numerazione è definito *dotted-decimal notation* (DDN), ovvero notazione decimale puntata, e corrisponde alla traduzione numerica decimale di quattro ottetti binari che compongono l'indirizzo IPv4.

Il protocollo IP fornisce un servizio di inoltro dei pacchetti di dati da un dispositivo all'altro.

In basso a sinistra della *Figura 2-f*, è osservabile come il server Larry abbia già creato un pacchetto di dati HTTP con il proprio header ed apposto l'intestazione TCP. Il protocollo IP si occupa di apporre a questo, un'ulteriore intestazione. L'*header* IP include, tra varie informazioni, un indirizzo IP di origine, che in questo caso è l'indirizzo di Larry (1.1.1.1), e un indirizzo IP di destinazione, in tal caso di Bob (2.2.2.2).

Larry è perciò pronto per inviare il pacchetto IP e come prima cosa non può far altro che inviare il pacchetto al router a lui più vicino ed appartenente alla sua stessa LAN, che prende il nome di *Default Gateway*.

Per Larry risulta quindi superfluo sapere quello che esiste al di là del suo *Default Gateway* perché sarà compito di quest'ultimo scegliere la direzione

verso cui inoltrare il pacchetto affinché raggiunga il destinatario nel minor tempo possibile.

Al passaggio 2, il router R1 riceve il pacchetto IP e lo stesso prende una decisione. Estrapola dall'intestazione IP del pacchetto, l'indirizzo di destinazione (2.2.2.2), lo confronta con i suoi percorsi (*routes*) noti e sceglie di inoltrare il pacchetto verso il router R2. Questo processo di inoltro del pacchetto IP è definito *IP routing*.

Al passaggio 3 il router R2 ripete lo stesso tipo di analisi utilizzata dal router R1, e sceglie di inoltrare il pacchetto alla sua destra, cioè verso la LAN in cui si trova Bob.

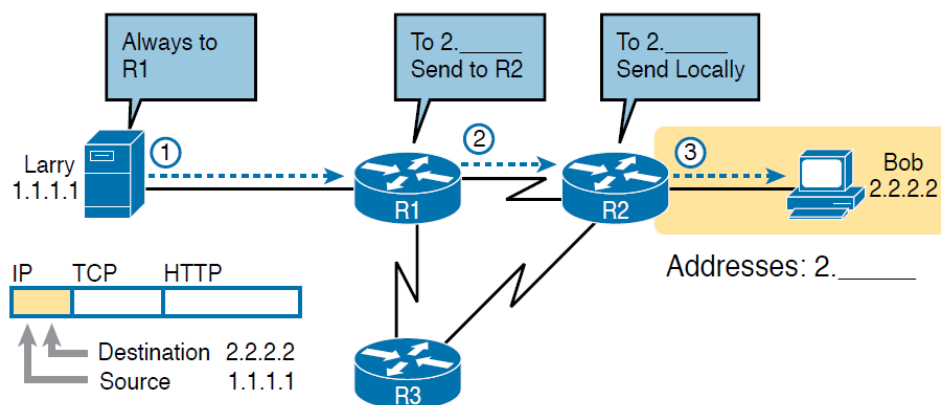


Figura 2-f: Esempio base di routing IP.

## 2.4 TCP/IP – *Layer Data Link* e *Layer Fisico*

Il *layer Data Link* e *Fisico* facenti riferimento al modello TCP/IP, definiscono i protocolli e le specifiche richiesti affinché i dati possano fluire attraverso una rete fisica. Il *Layer Fisico* definisce le caratteristiche del cablaggio di una rete ed il tipo di segnale che viene trasmesso sul mezzo.

Focalizzandoci per un momento sul *layer Data Link*; proprio come ogni altro *layer*, esso fornisce servizi al *layer* direttamente superiore ovvero il *Layer di Rete*. Quando un *host* o un router sceglie di inviare un pacchetto IP a un altro nodo di rete, dovranno incapsulare il pacchetto IP tra un *header* e un *trailer Layer 2* affinché possano inviare tale pacchetto attraverso uno specifico link.

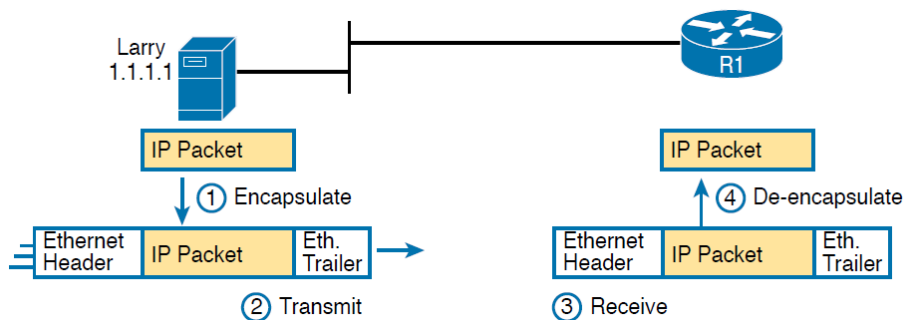


Figura 2-g: Trasmissione di un frame ethernet da Larry verso R1.

Supponendo che il collegamento tra Larry e il router R1 è effettuato mediante un cavo Ethernet, la *Figura 2-g* mostra quattro passaggi che si verificano a *layer Data Link* e che consentono a Larry di inviare un pacchetto IP verso R1.

I passaggi sono i seguenti:

1. Larry incapsula il pacchetto IP tra un'intestazione ed un trailer Ethernet, creando ciò che viene definito un *frame* Ethernet;
2. Larry trasmette fisicamente i bit del *frame* Ethernet, utilizzando segnali elettrici che sono veicolati dal mezzo trasmissivo;
3. Il router R1 riceve il segnale elettrico su una porta e recupera l'informazione trasmessa;
4. Il router R1 organizza i bit ricevuti in modo da ricostruire localmente il *frame* trasmesso da Larry;
5. Il router R1 de-capsula il pacchetto IP dal *frame* Ethernet rimuovendo e scartando l'intestazione e il trailer Ethernet.

In breve possiamo quindi riassumere le funzioni dei livelli Data Link e Fisico rispettivamente in:

- funzioni e servizi relativi alla trasmissione fisica dei dati;
- i protocolli e le regole che controllano l'uso del supporto fisico di trasmissione.

## 2.5 Incapsulamento dei dati

Durante la preparazione per l'invio dei dati, ogni *layer* dello stack TCP/IP aggiunge la propria intestazione (e per i protocolli di Data Link anche un *trailer*) ai dati forniti dal *layer* superiore.

Il processo mediante il quale un *host* TCP/IP invia i dati può quindi essere visto come un processo composto da cinque fasi. I primi quattro passaggi riguardano l'incapsulamento eseguito dai quattro livelli superiori TCP/IP e l'ultimo step è l'effettiva trasmissione fisica.

I passaggi sono riassunti nel seguente elenco:

1. Al *layer* applicazione, l'*host* crea e incapsula i dati di una applicazione con il proprio *header* di *layer*;
2. Il *layer* di trasporto appone un'intestazione al messaggio. A seconda delle tipo di applicazione richiesta, viene in utilizzata un'intestazione TCP o UDP.
3. Al segmento dati ottenuto, viene apposta un'intestazione di *layer* di rete (IP). Il protocollo IP definisce gli indirizzi IP che identificano in modo univoco ogni computer.
4. Il pacchetto IP viene incapsulato all'interno di un *frame* di *layer* Data Link.
5. Lo strato fisico codifica un segnale elettrico, ottico o radio sul mezzo, per trasmettere il *frame*.

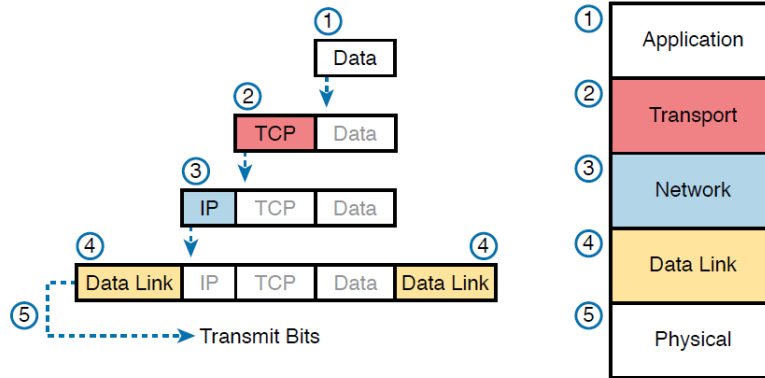


Figura 2-h: Cinque step di incapsulamento dati di TCP/IP.

### 2.5.1 Nomenclatura dei messaggi TCP/IP

Quando si parla e si scrive di networking si utilizzano termini come segmenti, pacchetti, o *frame* per fare riferimento ai messaggi mostrati in Figura 2-i. Ogni termine ha un significato specifico perché riferiti ad un determinato *layer* dello stack TCP/IP.

In particolare la suddivisione è la seguente: **segmento** per il *layer* di Trasporto, **pacchetto** per il *layer* di Rete e **frame** per il *layer* Data Link.

La Figura 2-k mostra ogni *layer* insieme al termine associato.

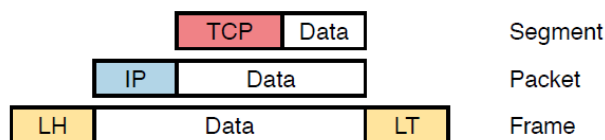


Figura 2-i: Definizione degli incapsulamenti dati.

Le lettere LH e LT stanno ad indicare rispettivamente il *link header* e il *link trailer* del *layer* Data Link.

La *Figura 2-i* mostra anche i dati incapsulati semplicemente come dei "dati" generici questo perché, quando un *Layer* prende in carico dei dati provenienti dal *layer* sovrastante, esso si disinteressa del contenuto e si concentra solo sul processo di incapsulamento di sua competenza.

## **2.5.2 Modello ISO/OSI**

Nonostante il modello di rete ISO/OSI sia stato di fatto superato dal TCP/IP, rimane valida ancora oggi la sua terminologia per cui è importante fare un confronto tra i due.

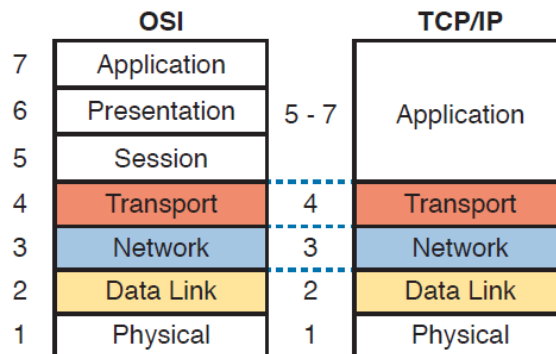
Da un punto di vista concettuale il modello ISO/OSI ha molte somiglianze con il modello TCP/IP.

È composto da livelli e ognuno di questi definisce un insieme di funzioni di rete tipiche e come con TCP/IP, i *Layers* ISO/OSI comprendono ciascuno protocolli e standard. In altri casi proprio come per TCP/IP, il comitato OSI non ha creato nuovi protocolli o standard, ma ha semplicemente fatto riferimento a protocolli già definiti, ad esempio lo standard Ethernet definito dall'IEEE.



Oggi il modello OSI può essere utilizzato come standard di confronto con altri modelli di rete.

La *Figura 2-j* confronta il modello OSI a sette strati con il modello TCP/IP a cinque strati.



*Figura 2-j: Confronto tra modello OSI e TCP/IP.*

Si noti come il modello TCP/IP a destra della *Figura 2-j*, utilizzi esattamente gli stessi nomi dei *Layers* OSI, per quello che riguarda i livelli inferiori, mentre si differenzi nella parte superiore. Nonostante nella pratica ogni nodo di rete implementi una architettura TCP/IP, si tende comunque ad usare la numerazione del modello OSI per scopi didattici e illustrativi. Non trascurabile il fatto che TCP/IP raggruppi più funzioni a *layer* Applicazione mentre OSI le suddivide in Sessione, Presentazione e Applicazione. La maggior parte delle volte non è importante la distinzione tra questi livelli ma è molto probabile trovare riferimenti come "protocollo *Layer* 5–7", utilizzando quindi la numerazione OSI e non TCP/IP.

### 2.5.3 Nomenclatura dei messaggi OSI

OSI utilizza un termine più generico per riferirsi ai messaggi; piuttosto che *frame*, pacchetti e segmenti, OSI utilizza il termine *Protocol Data Unit* (PDU). Una PDU rappresenta l'insieme di bit che includono sia i dati che l'intestazione e l'eventuale trailer per ogni *layer*. Ad esempio un pacchetto IP, utilizzando la terminologia OSI, è una PDU *layer 3* (abbreviato L3PDU) perché IP è un protocollo di *layer 3*.

OSI si riferisce semplicemente al *Layer "x"* PDU (LxPDU), dove la "x" si riferisce al numero del *layer* preso in esame, come mostrato in *Figura 2-k*.

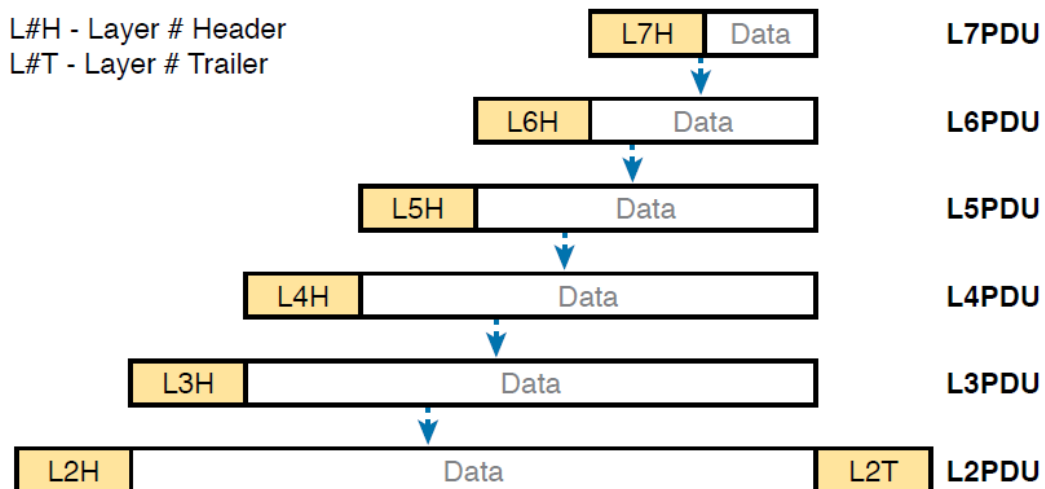


Figura 2-k: Incapsulamento OSI e Protocol Data Units.

## 3. FONDAMENTI DI LAN ETHERNET

Come definito dall'IEEE, con il termine *Ethernet* si comprende un'intera famiglia di standard e specifiche tecniche, sia al *layer* Fisico che al *layer* Data Link, che insieme vanno a costituire un efficace e sicuro metodo di trasmissione dati. La tecnologia Ethernet è sicuramente utilizzata in reti LAN ma può essere anche usata in reti MAN o WAN.

### 3.1.1 Small Office – Home Office LAN

Per iniziare consideriamo una LAN SOHO (*Small Office - Home Office*), ovvero un esempio di rete che è possibile trovare in qualsiasi ufficio o abitazione. In una LAN SOHO innanzitutto è imprescindibile la presenza di un dispositivo chiamato *Switch* Ethernet, il quale presenta svariate porte fisiche a cui è possibile collegare i dispositivi tramite cavi ethernet UTP (*Unshielded Twisted Pair*) e ha il compito di inoltrare correttamente i *frame* ethernet su queste.

La *Figura 3-a* mostra l'esempio di una LAN Ethernet SOHO. In particolare mostra un singolo LAN *switch*, cinque cavi e altrettanti nodi: tre PC, una stampante e un *router*. Il router in questo caso con funzione di *Default Gateway* è quel dispositivo che collega la LAN alla WAN (*Wide Area Network*).

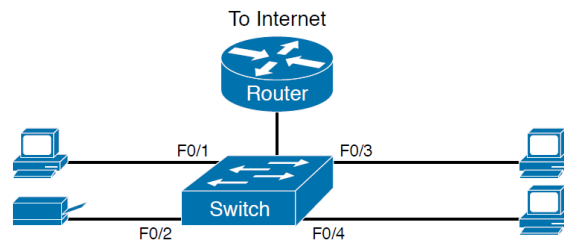


Figura 3-a: Esempio di SOHO LAN Ethernet.

Nelle moderne SOHO LAN, di norma è quasi sempre incluso un ulteriore dispositivo di rete, ovvero l'*Access Point* (AP). L'AP si comporta in qualche modo come uno *switch* Ethernet in quanto tutti i nodi LAN collegati ad esso, comunicano con l'AP in maniera wireless (invece che cablata) con il vantaggio che l'AP ha bisogno di un singolo collegamento Ethernet allo *switch* per connettere sia esso che tutti i dispositivi wireless a lui associati, nonché la possibilità di movimento per i dispositivi collegati come mostrato in *Figura 3-b*.

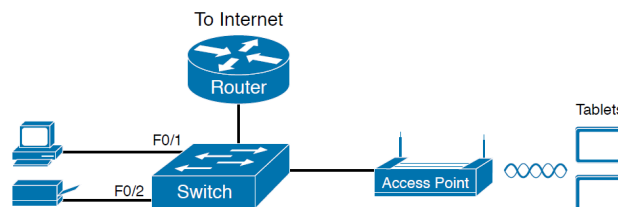


Figura 3-b: Esempio di SOHO LAN Ethernet e Wireless.

Sebbene sia la *Figura 3-a* che *3-b* mostrino *switch*, router e AP come dispositivi separati, la maggior parte delle SOHO LAN odierne utilizzano un unico dispositivo che li combina tutti insieme: l'*Integrated Services Router* (ISR).

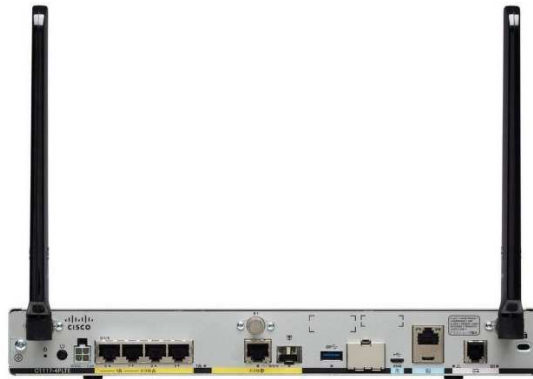


Figura 3-c: Esempio di ISR (Integrated Services Router).

### 3.1.2 LAN Aziendali

Le reti LAN aziendali hanno esigenze simili rispetto ad una rete SOHO, ma con una estensione su larga scala. In linea di massima le LAN Ethernet aziendali sono composte da svariati *switch* custoditi in armadi chiusi sotto chiave e posti in luoghi sicuri e poco frequentati. Da ogni *switch* partono i cavi Ethernet che raggiungono gli uffici e le sale dove potranno essere collegati i dispositivi alla rete LAN. Allo stesso tempo, alcuni dei cavi ethernet potrebbero essere utilizzati per connettere degli AP che, opportunamente configurati, permettono il collegamento wireless alla LAN aziendale.

La *Figura 3-d* mostra l'esempio di una tipica LAN aziendale in un edificio a tre piani. Ogni piano ha uno *switch* e un AP wireless. Per consentire la comunicazione tra i piani, ogni *switch* si collega ad uno *switch* di distribuzione centralizzato. Ad esempio PC3 può inviare dati a PC2, ma

prima, i *frame* ethernet dovranno transitare per lo *switch* di distribuzione (SWD) e poi tornare indietro tramite lo *switch* SW2 al secondo piano.

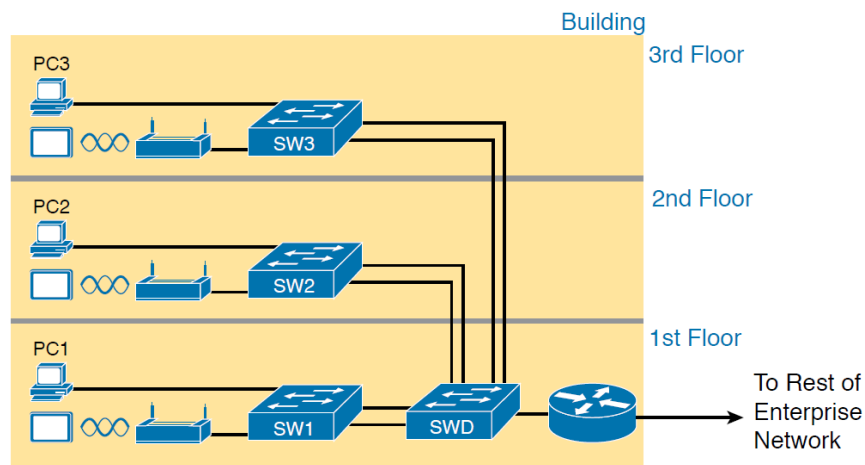


Figura 3-d: Esempio di LAN aziendale cablata e wireless su singolo edificio.

La *Figura 3-d* mostra anche il router, in questo caso un dispositivo ben distinto, e che ha il compito di connettere la LAN alla WAN.

Il resto del capitolo 3.1 si concentra sullo standard Ethernet.

### 3.1.3 Layer Fisico Ethernet

La tecnologia Ethernet racchiude standard che definiscono le specifiche fisiche di come i dati debbano essere inviati su un particolare tipo di cablaggio.

Ethernet supporta un'ampia varietà di collegamenti fisici e include quindi molti standard a seconda della tipologia di cavi utilizzati e la velocità di trasmissione (*throughput*).

La scelta del cablaggio ricade fundamentalmente su due tipologie di materiali: cavi di rame o fibre ottiche. I dispositivi che utilizzano cavi UTP (*Unshielded Twisted Pair*), trasmettono i bit sottoforma di segnali elettrici su conduttori in rame all'interno del cavo; l'alternativa è la Fibra ottica, più costosa <sup>[1]</sup>, ma che consente di ottenere bande trasmissive nettamente superiori rispetto alla trasmissione su rame.

L'IEEE definisce gli standard del *layer* fisico Ethernet, utilizzando delle convenzioni. Il nome formale inizia con l'identificativo numerico 802.3 seguito da alcune lettere a suffisso. Per identificare gli standard vengono utilizzati anche nomi "informali", che aiutano ad identificare la velocità massima garantita nonché un indizio sulla natura del cablaggio cioè se di tipo UTP (con suffisso T) o fibra (con suffisso che include la X).

La *Tabella 3-a* elenca alcuni esempi di standard di *layer* fisico Ethernet, con indicati i corrispondenti nomi comuni utilizzati nella comunità tecnica.

Velocità	Nome comune	Nome informale standard IEEE	Nome formale standard IEEE	Tipo di cavo, lunghezza max
10 Mbps	Ethernet	10BASE-T	802.3	Rame, 100 m
100 Mbps	Fast Ethernet	100BASE-T	802.3u	Rame, 100 m

1000 Mbps	Gigabit Ethernet	1000BASE-LX	802.3z	Fibra, 5000 m
1000 Mbps	Gigabit Ethernet	1000BASE-T	802.3ab	Rame, 100 m
10 Gbps	10 Gig Ethernet	10GBASE-T	802.3an	Rame, 100 m

Tabella 3-a: Esempi di standard Ethernet a layer Fisico.

### 3.1.4 Data Link *layer* Ethernet

Sebbene Ethernet possa essere veicolato su diversi mezzi fisici, i *Layers* superiori al primo non hanno percezione del canale usato per la trasmissione.

Mentre gli standard del *layer* fisico si concentrano sull'invio di bit su un cavo, il protocollo Data Link Ethernet si concentra sull'invio di un *frame* Ethernet dal nodo di origine a quello di destinazione.

La *Figura 3-e* mostra un esempio del processo. In questo caso il PC1 invia un *frame* Ethernet a PC3. Il *frame* viaggia su un collegamento UTP fino allo *switch* SW1, poi su collegamenti in fibra tra gli *switch* SW2 e SW3 e infine su un altro collegamento UTP fino a PC3. Si può notare come i bit viaggino effettivamente a quattro velocità diverse lungo questo percorso: 10 Mbps, 1 Gbps, 10 Gbps e 100 Mbps rispettivamente.



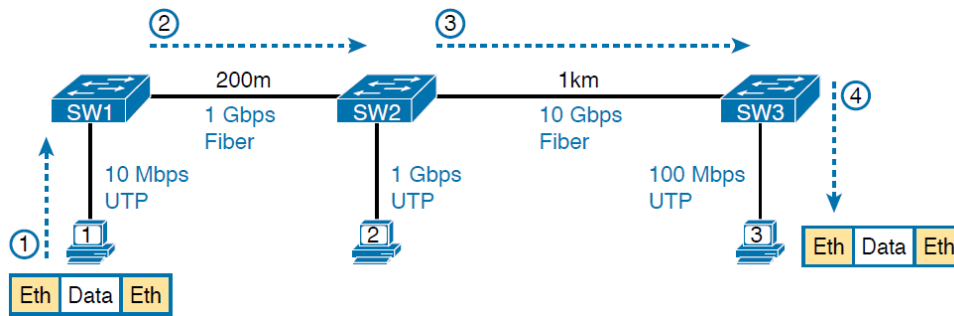


Figura 3-e: Inoltro frame Ethernet su diverse tipologie di link.

Ogni collegamento può quindi utilizzare diversi tipi di cavi a diverse velocità ma il risultato finale è il corretto inoltro di un *frame* Ethernet da un dispositivo mittente ad un destinatario, facenti parte della stessa LAN.

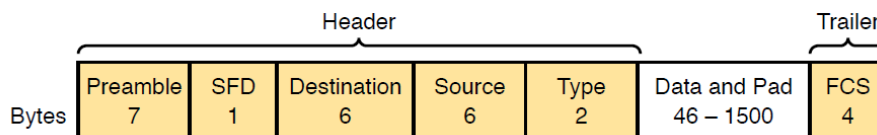
Il resto del capitolo 3 approfondisce il funzionamento del *layer* Data Link di una rete basata su Ethernet, e si conclude con la presentazione di due protocolli essenziali per l'inoltro di *frame* attraverso una LAN Ethernet come quella Cooperlat ovvero il *Vlan Trunking Protocol* e lo *Spanning Tree Protocol*.

## 3.2 Invio dei dati su reti Ethernet

Abbiamo visto come al *layer* fisico, Ethernet abbia molti standard dovuti all'elevato numero di tipologie di cablaggi. Ora analizzeremo la duttilità di Ethernet al *layer* Data Link, dove presenta un unico standard valido per tutte le tipologie di link fisici menzionati in precedenza.

### 3.2.1 Ethernet *Frame*

Il formato del *frame* Ethernet può cambiare, ma di norma quello più diffuso è quello mostrato in *Figura 3-r*.



*Figura 3-f: Formato di frame Ethernet comunemente utilizzato.*

Nella *Tabella 3-e* vengono riassunte le principali funzioni dei campi di *header* e *trailer* Ethernet, che poi verranno analizzate più in dettaglio.

Campo	N° di byte	Descrizione
<i>Preamble</i>	7	Utilizzato per la sincronizzazione.
<i>Start Frame Delimiter</i> (SFD)	1	Indica che il successivo Byte conterrà il <i>MAC address</i> del nodo destinatario del frame.
<b><i>Destination MAC</i></b>	6	Indica il <i>MAC address</i> del destinatario.

<b>Source MAC</b>	6	Indica il <i>MAC address</i> del mittente.
<i>Type</i>	2	Definisce il tipo di protocollo trasportato nel <i>frame</i> (ad esempio IPv4 oppure IPv6)
<i>Data and Pad</i>	46 - 1500	Contiene i dati da un <i>layer</i> superiore, in genere un L3PDU (di solito un pacchetto IPv4 o IPv6). Il mittente aggiunge una stringa "riempitiva" per soddisfare il requisito di lunghezza minima se il pacchetto è inferiore a 46 byte.
<i>Frame Check Sequence (FCS)</i>	4	Fornisce un metodo alla scheda NIC ricevente per determinare se il <i>frame</i> ha subito alterazioni durante la trasmissione.

Tabella 3-b: Campi di intestazione e trailer del frame Ethernet 802.3.

### 3.2.1.1 Indirizzamento Ethernet

Gli indirizzi Ethernet definiti *MAC (Media Access Control) address*, sono lunghi 6 byte ovvero 48 bit. Per comodità, la maggior parte dei computer elenca gli indirizzi MAC come numeri esadecimali a 12 cifre, separati da punti (ad esempio 0000.0C12.3456). Il *MAC address*, all'interno della LAN, svolge il compito che l'indirizzo IP svolge in una WAN, ovvero quello di identificare univocamente, non più un dispositivo, ma una scheda di rete.

Anche se la maggior parte degli indirizzi MAC rappresenta un'unica NIC, esistono tre tipologie di *MAC address*:

- *Unicast address*: identifica un'unica interfaccia di rete;

- *Broadcast address*: i *frame* inviati a questo indirizzo devono essere consegnati a tutti i dispositivi presenti sulla rete LAN ethernet. Ha il valore esadecimale FFFF.FFFF.FFFF, pari a 48 bit impostati a 1;
- *Multicast address*: i *frame* inviati a un indirizzo Ethernet multicast verranno copiati e inoltrati ad un sottoinsieme dei dispositivi presenti sulla LAN, che si “offrono” volontari di ricevere i *frame* inviati ad un indirizzo multicast specifico.

L'inoltro di *frame* Ethernet con l'utilizzo di *MAC address* unicast, può funzionare solo se tali indirizzi sono univoci. Se due schede di rete presentassero lo stesso indirizzo *MAC address* si genererebbe confusione, in quanto un *frame* potrebbe dover raggiungere o l'una o l'altra NIC.

Il protocollo Ethernet risolve questo problema imponendo un rigoroso processo amministrativo in modo che, al momento della produzione, a tutti i dispositivi Ethernet venga assegnato un indirizzo MAC univoco. Prima di produrre un qualsiasi dispositivo Ethernet, è obbligatorio richiedere all'IEEE l'assegnazione di un codice univoco di 3 byte, chiamato *Organizationally Unique Identifier* (OUI), il quale identifica i primi 3 byte del *MAC address* di ogni scheda di rete e che quindi identifica globalmente il produttore. Il compito del produttore è quello di assegnare in maniera univoca i valori degli ultimi 3 byte, ottenendo quindi un numero mai utilizzato prima.

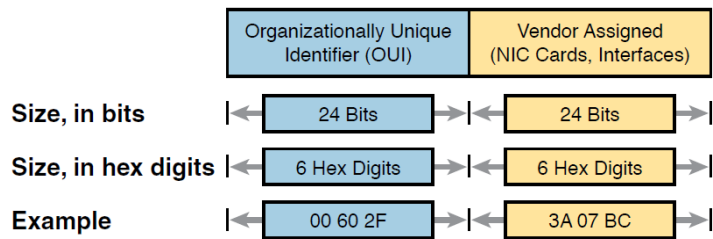


Figura 3-g: Struttura di un indirizzo Ethernet unicast.

### 3.2.1.2 Rilevamento di errori con FCS

Ethernet definisce un metodo per rilevare se i bit contenuti in un *frame* sono stati corrotti durante la trasmissione. Per fare questo, come la maggior parte dei protocolli Data Link, utilizza il campo *Frame Check Sequence* (FCS) posto nel trailer: un valore numerico che è il risultato di una complessa formula matematica applicata dal mittente all'intero *frame* ethernet che sta per inoltrare.

Il mittente memorizza il risultato calcolato nel campo FCS ed invia il *frame*. Il destinatario all'arrivo del *frame* applica la stessa formula matematica e, una volta ottenuto il risultato, lo confronta con quello che il mittente aveva memorizzato nel campo FCS; se i risultati sono gli stessi ciò indica che il *frame* non è mutato durante la trasmissione, altrimenti si è verificato un errore e il *layer* Data Link del destinatario scarta il *frame*.

Importante notare che il rilevamento degli errori a *layer* Data Link non implica anche il ripristino degli stessi. Ethernet definisce che il *frame* errato debba essere eliminato ma non ne richiede la ritrasmissione al mittente. La

funzione di ripristino di errori trasmissivi spetta ai livelli gerarchicamente superiori dello stack TCP/IP, e in particolare al protocollo TCP.

### **3.2.2 Il funzionamento di uno *Switch* Ethernet**

Il ruolo di uno *switch* in una LAN è quello di inoltrare *frame* Ethernet ai corretti destinatari. Per fare ciò, lo *switch* costruisce dinamicamente una *MAC address Table*, nella quale associa ogni sua interfaccia ad uno o più indirizzi MAC.

Questo vuol dire che lo *switch*, all'arrivo di un *frame* ethernet da una delle sue interfacce, compie una sequenza di operazioni. Per prima cosa analizza il contenuto di header e trailer, dove abbiamo già visto sono contenute diverse informazioni, anche se le più importanti sono indubbiamente i *MAC address* sorgente e destinatario. Se lo *switch* riceve per la prima volta un *frame* ethernet da un dispositivo, allora procede immediatamente nell'aggiornamento della sua *Tabella* di indirizzi MAC.

Così facendo, lo *switch* nel momento in cui riceve un *frame* ethernet, confronterà il *MAC address* destinatario con quelli che ha precedentemente memorizzato nella *MAC address Table* e, se trova una corrispondenza, inoltrerà il *frame* verso l'interfaccia evidenziata dal match.

Ad una interfaccia di rete possono corrispondere più di un *MAC address* in quanto se pensiamo ad un link che collega due *switch*, su quella specifica

interfaccia lo *switch* assocerà tutti i *MAC address* dei dispositivi che può raggiungere mediante l'altro *switch*.

### 3.3 Le Virtual LAN

Possiamo dare una definizione alternativa di LAN che può aiutare a comprendere meglio il concetto di LAN Virtuale (VLAN):

“Una LAN include tutti i dispositivi nello stesso *dominio di broadcast*”.

Concettualmente un *dominio di broadcast* è un’area immaginaria che include i dispositivi connessi alla LAN in modo tale che, quando uno qualsiasi invia un *frame* broadcast, tutti gli altri ricevono una copia di tale *frame*.

Per impostazione predefinita uno *switch* considera tutte le sue interfacce nello stesso dominio di broadcast cioè, quando un *frame* di broadcast entra da una sua interfaccia, lo stesso viene inoltrato su tutte le altre interfacce ad eccezione di quella da cui lo ha ricevuto. Continuando sulla linea di tale logica verrebbe da pensare che per creare due diversi domini di broadcast, sia necessario acquistare almeno due *switch* Ethernet, come mostrato in *Figura 3-h*.



*Figura 3-h: Creazione di due domini di broadcast mediante l'utilizzo di due switch fisici separati.*

La tecnologia delle VLAN invece, ha permesso di ottenere una suddivisione logica, e non fisica, dei domini di broadcast.



Ciò è possibile associando opportunamente alcune interfacce dello *switch* ad una VLAN ed altre ad una differente VLAN, ottenendo automaticamente la suddivisione in due diversi domini di broadcast.

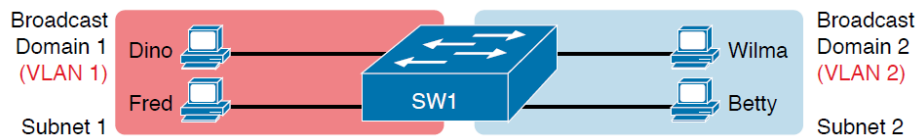


Figura 3-i: Creazione di due domini di broadcast mediante l'utilizzo di uno switch e due VLAN.

Implementare delle VLAN comporta notevoli vantaggi come:

- **Aumento delle prestazioni.** I *frame* non vengono propagati verso destinazioni non necessarie grazie al confinamento del traffico broadcast alla singola VLAN;
- **Aumento della sicurezza.** Gli *host* possono vedere solamente il traffico della loro VLAN e non quello delle altre;
- **Risparmio.** Si realizzano LAN Virtuali sulle stesse strutture fisiche con notevole risparmio di tempo e di denaro;
- **Flessibilità.** Permette di creare progetti più flessibili che raggruppano gli utenti per reparto o per gruppi di lavoro anziché in base alla posizione fisica.

Fino adesso è stato considerato un solo *switch*, ma nel caso se ne avessero due o più interconnessi, e l'obiettivo fosse estendere il collegamento di una stessa VLAN tra i due, in un primo momento potremmo immaginare una situazione come quella mostrata in *Figura 5-v*.

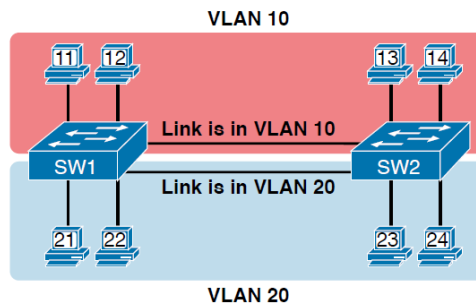


Figura 3-j: Multiswitch VLAN senza collegamento di trunking.

In questo caso vengono utilizzati 2 cavi per interconnettere i due *switch* perché due sono le VLAN attive. Deduciamo che, in questo contesto il numero dei link tra *switch* incrementerebbe in maniera proporzionale al numero di VLAN esistenti, e quindi risulterebbe sconveniente.

La soluzione a questa problematica è stata trovata grazie all'implementazione del *Vlan Trunking Protocol*, che introduce il **VLAN tag** e il collegamento di **trunk** tra *switch*.

L'utilizzo del VTP fa sì che gli *switch* utilizzino un processo chiamato *VLAN tagging* mediante il quale, lo *switch* che invia un *frame* ethernet aggiunge un ulteriore campo all'intestazione del *frame*. Questa intestazione di trunking include un campo identificatore della VLAN (ID VLAN), che specifica a quale VLAN appartiene lo specifico *frame*. In questo modo è possibile collegare gli *switch* con un unico link ma, allo stesso tempo, lo *switch* ricevente può riconoscere a quale VLAN il *frame* appartenga e conseguentemente inoltrarlo alle sole interfacce interessate.

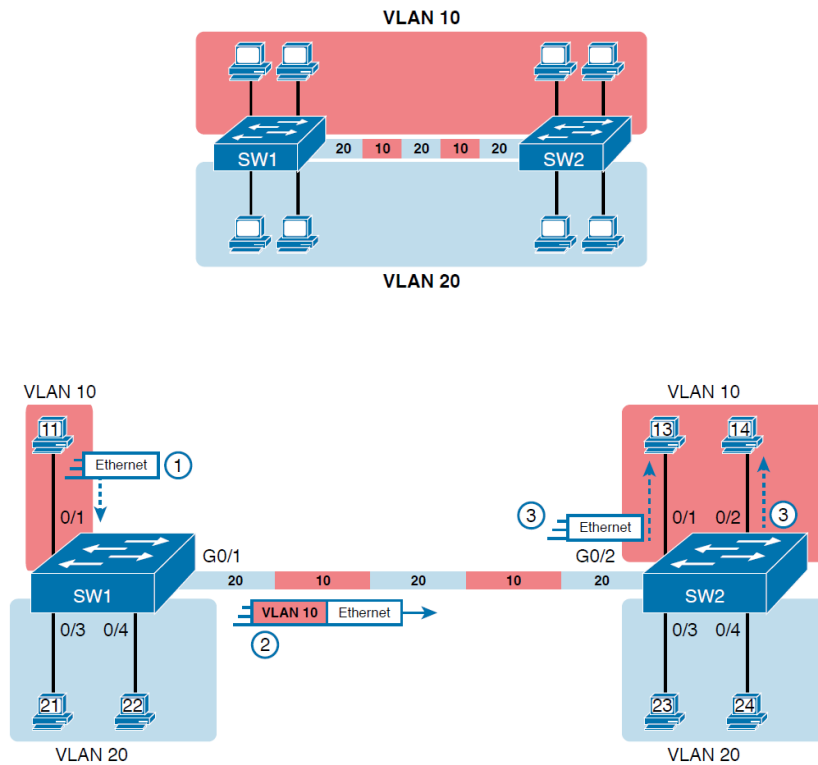


Figura 3-k: Multiswitch VLAN con collegamento di trunking, esempio di inoltrare frame in trunk.

A questo punto sorge legittimo il dubbio su come sia possibile mettere in comunicazione dispositivi appartenenti a due VLAN diverse. Ciascuna VLAN si comporta infatti come una LAN separata dalle altre e per la loro interconnessione è necessario un dispositivo di *Layers 3*, ovvero utilizzando un router o un *multilayer switch*.

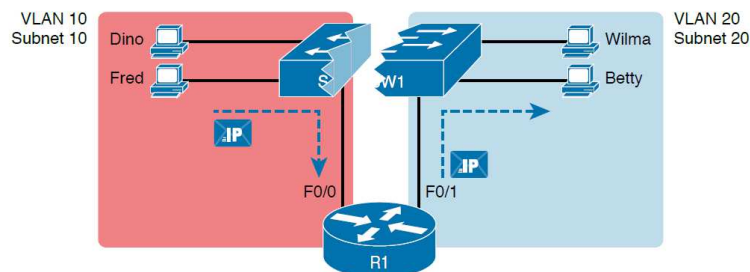


Figura 3-l: Routing tra due VLAN su due interfacce fisiche.

### 3.4 Lo *Spanning Tree Protocol*

In una rete LAN è auspicabile che non vi sia mai interruzione di connettività perciò, è essenziale dotare la rete di ridondanza nei collegamenti tra *switch*.

Questo implica che ogni *switch* debba essere collegato con almeno altri due *switch*, in modo tale da essere resiliente nei confronti di eventuali *fault* su link fisici.

La ridondanza di collegamenti comporta necessariamente la creazione di loop fisici, che costituiscono un punto di cruciale attenzione e per i quali entra in gioco il fondamentale contributo dello *Spanning Tree Protocol* (STP).

Per comprendere meglio la creazione di loop e la loro pericolosità, osserviamo la *Figura 3-y*, che mostra un esempio di rete in cui l'*host* Bob invia un *frame* di *broadcast*. Le linee tratteggiate mostrano come gli *switch* inoltrano il *frame* in assenza di STP.

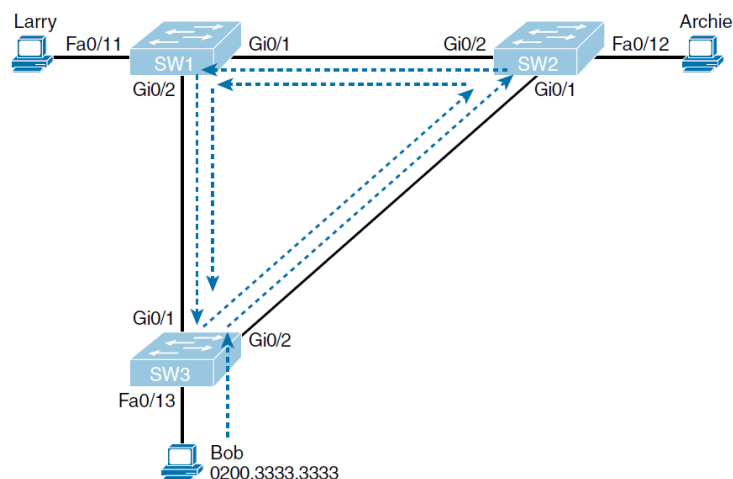


Figura 3-m: Esempio di Broadcast Storm in una LAN senza STP o RSTP.

Trattandosi di un *frame* di broadcast, lo *switch* SW3 inoltrerà lo stesso a tutte le interfacce appartenenti alla stessa VLAN (che in questo caso si suppone essere unica), il che implica che il *frame* viaggerà sia verso lo *switch* SW1 che verso lo *switch* SW2. A loro volta i due *switch* SW1 e SW2 faranno altrettanto e così via, creando un continuo flusso di *frame* broadcast che tende a saturare la banda trasmissiva e può durare ipoteticamente per un tempo infinito, almeno che uno *switch* oppure un collegamento non subisca un *fault*.

Un solo *frame* in loop provoca quella che viene definita ***broadcast storm***, ovvero tempesta di broadcast. Le *broadcast storm* si verificano quando qualsiasi tipo di *frame* Ethernet (*frame* broadcast, *frame* multicast o unicast con destinazione sconosciuta) viene inoltrato in modo indefinito su una rete LAN; questo accade perché, al contrario di un pacchetto IP, il *frame* ethernet non contiene un contatore come il *Time To Live*, e quindi potrebbe non essere mai scartato. Le *broadcast storm* possono saturare tutti i collegamenti con copie di un singolo *frame*, impedendo l'inoltro di *frame* legittimi.

Conseguenze di una *broadcast storm* sono altre problematiche come ***l'instabilità della Tabella degli indirizzi MAC*** oppure ***la trasmissione multipla di uno stesso frame***.

Per *instabilità della MAC address table* si intende che le tabelle degli indirizzi MAC degli *switch*, continuano ad aggiornarsi a causa dei *frame* in

loop con lo stesso *MAC address* sorgente che arriva su più porte dello stesso *switch*. In questo modo, la scelta per lo *switch* di inoltrare un *frame* su di una interfaccia potrebbe non essere più univoca, con conseguente inoltro dei *frame* a destinatari errati.

La *trasmissione multipla di un frame* consiste nel recapito multiplo di uno stesso *frame* all'*host* destinatario causando errori nella ricezione da parte di quest'ultimo. Tale problematica si verifica poiché la presenza di un loop provoca che uno stesso *frame* arrivi a destinazione da due percorsi logici e fisici diversi.

### **3.4.1 Principio di funzionamento di STP**

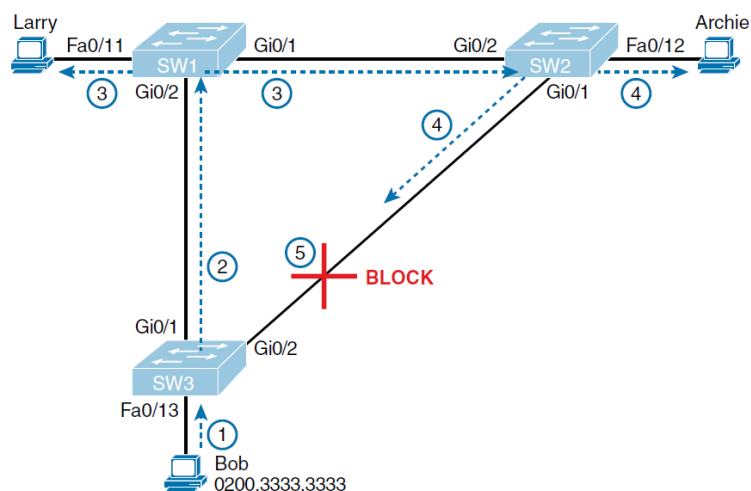
Analizziamo da vicino come i protocolli STP prevengono la formazione di loop. Dal nome estrapoliamo la parola chiave *tree* ovvero albero, e possiamo intuire la finalità per cui lo STP opera, ovvero la creazione di un albero logico sui link che collegano gli *switch* in modo tale che, come per un vero albero, sia sempre possibile tracciare un'unica linea che collega la radice ad ogni suo ramo.

Lo *Spanning Tree Protocol* (STP) o la sua versione *Rapid* (RSTP), che opera nella stessa maniera ma con tempistiche più veloci, è un algoritmo pro-attivo che previene la formazione di loop bloccando specifiche porte degli *switch*, oppure abilitandole all'inoltro nel momento in cui venga rilevato

un *fault* su un link precedentemente in uso, garantendo la continuità di comunicazione.

Le interfacce in stato di blocco non elaborano alcun *frame* ad eccezione dei messaggi di STP/RSTP e quindi, non inoltrano o elaborano *frame* utente e non apprendono gli indirizzi MAC dei *frame* ricevuti.

La *Figura 3-z* mostra lo STP in azione, che risolve il problema mostrato nella precedente *Figura 3-y* ponendo l'interfaccia Gi0/2 dello *switch* SW3 in stato di blocco.



*Figura 3-n: Interruzione del loop da parte dello STP.*

In questo modo lo *switch* SW3 non inoltrerà il *frame* di broadcast proveniente da Bob sulla interfaccia Gi0/2 (perché in stato di *Blocking*) e non considererà il *frame* in arrivo dallo *switch* SW2 disinnescando di fatto il loop.

### 3.4.2 Dettaglio protocollo STP

Lo scopo prefissato e presentato precedentemente, lo *Spanning Tree Algorithm* (STA) deve organizzare in modo gerarchico tutti gli *switch* che partecipano alla LAN, e per questo la prima azione intrapresa è l'elezione di uno *switch* "principale" denominato *Root Switch*. Per eleggere il *Root Switch*, anche detto *Root Bridge*, tutti gli *switch* comunicano tra loro mediante particolari messaggi chiamati *Bridge Protocol Data Unit* (BPDU).

I BPDU sono messaggi identificativi, contenenti i parametri che servono allo STP nel suo lavoro di regolatore "gerarchico" tra gli *switch*. Nella *Tabella 3-f* vediamo riassunti i principali campi che lo costituiscono.

Campi di un BPDU	Descrizione
<b>Root Bridge ID</b>	Il Bridge ID del <i>Root Switch</i>
<b>Bridge ID</b>	Il Bridge ID dello <i>switch</i> che invia questo BPDU
<b>Root cost</b>	Il costo STP tra lo <i>switch</i> e il <i>Root Bridge</i>
<b>Timer STP del Root Bridge</b>	Include <i>Hello Timer</i> , <i>MaxAge Timer</i> e <i>Forward Delay Timer</i>

*Tabella 3-c: Campi nello STP BPDU.*

Uno tra i più importanti campi contenuti nel BPDU è il Bridge ID, costituito da 8 byte, i primi 2 di *Priority* e i restanti 6 coincidenti con il *MAC address* dello *switch* generatore della BPDU.



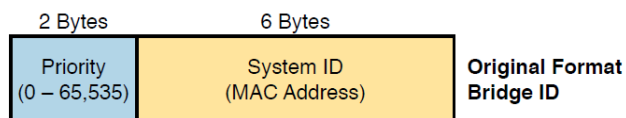


Figura 3-o: Formato del Bridge ID.

Proprio la presenza del *MAC address*, che per definizione è univoco, rende il Bridge ID a sua volta univoco e come vedremo tra poco, potrebbe essere l'unico parametro che garantisce uno sparggio tra due *switch* per decretare chi debba bloccare delle porte e chi no.

### 3.4.2.1 Elezione del Root Bridge

L'elezione del *Root Bridge* è molto semplice. Ogni *switch* non appena viene alimentato e collegato in una rete LAN (dove si presume esista già un *Root Bridge* eletto), per impostazione di default proclama sé stesso come *Root Bridge* ed inizia ad inoltrare dalle sue interfacce le proprie BPDUs, contenenti il proprio BID come nell'esempio mostrato in *Figura 3-bb* e dove lo *switch* SW1 viene connesso per la prima volta alla LAN.

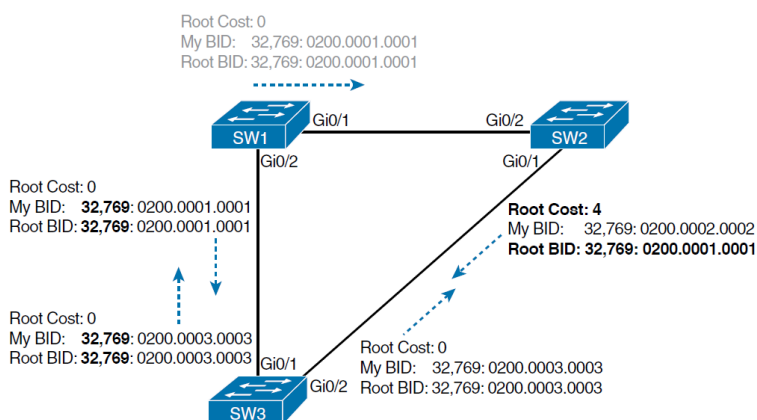


Figura 3-p: Inizio del processo di elezione del Root Bridge.

Tutti gli altri *switch* vengono informati della nuova presenza, ed effettuano un confronto tra il BID ricevuto dal nuovo *switch* e il loro:

- se il BID ricevuto ha un valore più alto del proprio, allora lo *switch* ricevente saprà che il nuovo *switch* non potrà essere il *Root Bridge* e quindi non inoltrerà ulteriormente le BPDU ricevute;
- se il BID ricevuto è invece effettivamente di un valore più basso rispetto al proprio, allora lo *switch* ricevente inoltrerà le BPDU perché tutti gli altri *switch* presenti in LAN sappiano che un nuovo *Root Bridge* della topologia potrebbe essere eletto.

Una volta che le BPDU del nuovo *switch* raggiungono il già eletto *Root Bridge*, avviene un confronto sui due parametri costituenti il BID nell'ordine indicato:

- *Priority*;
- *MAC address*.

Se la *Priority* del nuovo *switch* è più bassa di quella del *Root Bridge*, allora il confronto termina perché il vincente è sempre lo *switch* con *Priority* inferiore; se invece le due *Priority* dovessero essere uguali, allora lo spareggio si incentra sui *MAC address* dei due *switch* che decreteranno necessariamente il vincitore essendo essi univoci.

Riprendendo la *Figura 3-bb*, lo spareggio tra *switch* SW1 e *switch* SW3, in questo caso si conclude proprio con il confronto tra i *MAC address* (in

quanto i due hanno *Priority* identica pari a 32.769), ed essendo quello di SW1 minore di quello di SW3, il vincitore sarà proprio SW1 e tutti gli altri *switch* inizieranno ad inoltrare le sue BPDUs.

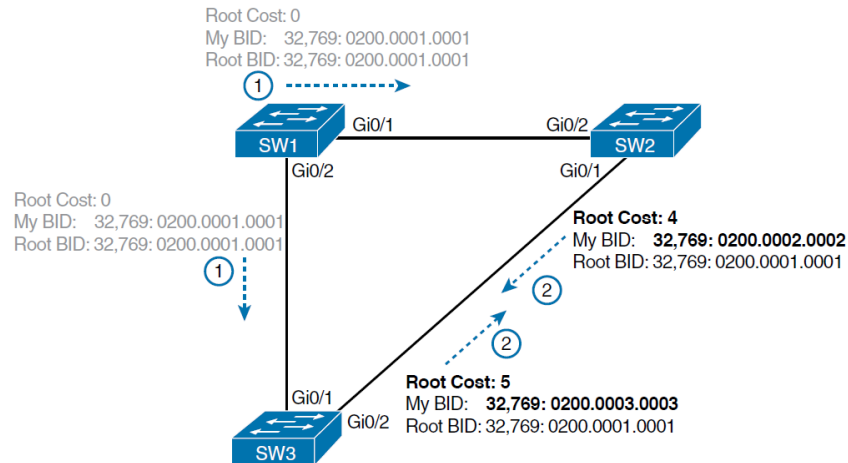


Figura 3-q: SW1 vince l'elezione come Root Bridge.

### 3.4.2.2 Ruoli delle porte nello STP

Lo *Spanning Tree Algorithm*, una volta eletto il *Root Bridge*, stabilisce il ruolo di ciascuna porta presente in topologia. Riassumiamo nella seguente *Tabella 3-g* i possibili ruoli attribuiti dallo STP ad una porta di uno *switch*.

Ruolo della porta	Descrizione
<b>Root port</b>	Tutti gli <i>switch</i> che non sono <i>Root Bridge</i> designano una root port, ovvero la porta dalla quale raggiungono il <i>Root Bridge</i> al minor costo
<b>Designated Port</b>	Porta in stato di <i>Forwarding</i>

<b>Blocked</b>	Porta bloccata dallo STP (non inoltra <i>frame</i> , non impara <i>MAC address</i> , ecc.)
<b>Disabled Port</b>	Porta spenta amministrativamente (shutdown)

Tabella 3-d: Ruoli delle porte nello STP.

L'unico *switch* che ha tutte le porte nello stato di *Designated Port* è il *Root Bridge*.

Dal punto di vista dello STP, il “costo” di un link o una successione di questi, è un parametro che quantifica la “qualità” dei link che separano lo *switch* oggetto della valutazione dal *Root Bridge*. Il costo di un link è legato in maniera inversamente proporzionale alla velocità di *throughput* che il link è in grado di garantire.

Nella *Tabella 3-h* riassumiamo i costi stabiliti dalla IEEE in relazione alla banda sostenibile dai link.

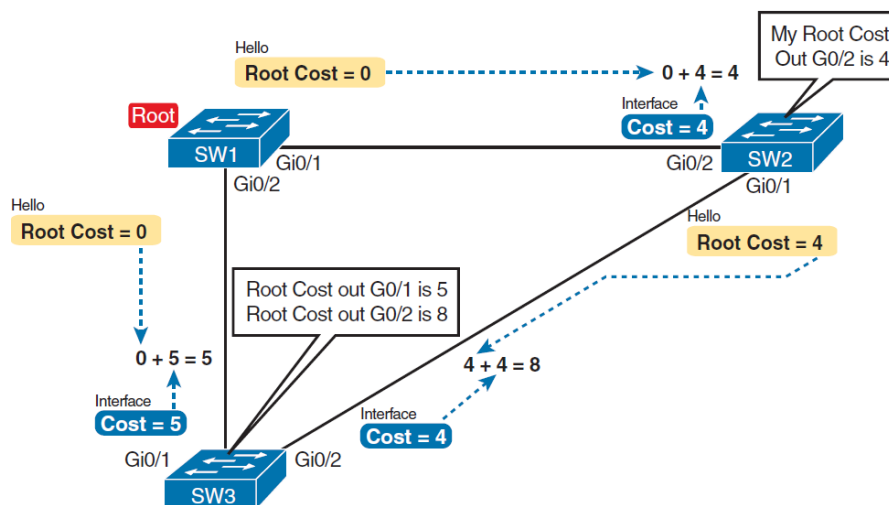
Velocità Ethernet	Costi IEEE prima del 1998	Costi IEEE dopo il 2004
<b>10 Mbps</b>	100	2.000.000
<b>100 Mbps</b>	19	200.000
<b>1 Gbps</b>	4	20.000
<b>10 Gbps</b>	2	2.000
<b>100 Gbps</b>	N/A	200
<b>1 Tbps</b>	N/A	20

Tabella 3-e: Costo delle porte prestabilito dalla IEEE.

Il costo di un link è fondamentale per decretare la root port di uno *switch* perché, come già detto, la root port coincide con il link avente minor *root cost*. Il costo del link però funge anche da discriminante per decretare quale porta bloccare in un link che non sia di root.

La *Figura 3-dd* mostra un chiaro esempio di come gli *switch* calcolino il loro miglior *root cost* e quindi come scelgano la loro *root port*.

Lo STP sullo *switch* SW3, calcola il suo costo per raggiungere il *Root Bridge* sui due possibili percorsi sommando al costo pubblicato dai messaggi *Hello* BPDU inviati sia da SW1 che da SW2, i costi di interfaccia elencati in *Figura 3-r*.



*Figura 3-r: Come lo STP/RSTP calcola il costo di root e decide il Blocking.*

Le discriminanti per decidere quale *switch* debba bloccare un'interfaccia sono:

- Il *root cost*;
- A parità di costo viene bloccata la porta dello *switch* con *bridge ID* più alto.

Nel caso di *Figura 3-dd* verrà bloccata l'interfaccia Gi0/2 di SW3, in quanto il suo *root cost* è 5 mentre quello di SW2 è 4, ma lo stesso si verificherebbe anche se sia SW3 che SW2 avessero *root cost* pari a 4 perché in quel caso il *MAC address*, e quindi il *birdge ID*, di SW3 risulta maggiore di quello di SW2.

### 3.4.2.3 Il cambio topologia nello STP

Esaminiamo come lo STP si comporta di fronte all'esigenza di modificare la propria topologia logica. In questo contesto risulta utile osservare da vicino le tempistiche che lo STP impiega nel suo processo di convergenza. In *Tabella 3-i* sono elencati i principali *Timer* dello STP. Si noti che tutti gli *switch* utilizzano i *Timer* stabiliti dal *Root Bridge* e specificati nelle *Hello BPDU* da lui inoltrate.

Timer	Valore di default	Descrizione
<b>Hello</b>	2 secondi	Il periodo di tempo che intercorre tra l'invio di <i>Hello BPDU</i> da parte del <i>Root Bridge</i> .
<b>Max Age</b>	10 volte il <i>Timer Hello</i>	Quanto tempo deve aspettare uno <i>switch</i> senza ricevere <i>Hello BPDU</i> da parte del

		<i>Root Bridge</i> prima di poter cambiare topologia STP.
<b>Forward Delay</b>	15 secondi	Ritardo che intercorre nella transizione di una porta dallo stato di <i>Blocked</i> allo stato di <i>Forwarding</i> . Una porta passa preventivamente in uno stato di <i>Listening</i> e successivamente di <i>Learning</i> per il tempo specificato dal <i>Forward Delay</i> .

Tabella 3-f: Timer dello STP.

Come risulta evidente dalla *Tabella 3-f*, quando una porta precedentemente in stato di *Blocking* deve passare in *Forwarding*, lo *switch* interessato deve far transitare prima la porta attraverso due stati intermedi. Questi stati STP temporanei sono essenziali per prevenire loop temporanei:

- *Listening*: come nello stato di *Blocking*, l'interfaccia non inoltra i *frame* che riceve. Lo *switch* elimina dalla propria *Tabella MAC address* gli indirizzi dai quali vengono ricevuti *frame* per tutto questo periodo di *Forward Delay*. Tali indirizzi MAC obsoleti potrebbero essere la causa di loop temporanei se non venissero preventivamente eliminati.
- *Learning*: le interfacce dello *switch* in questo stato continuano a non inoltrare *frame* ma lo *switch* inizia comunque ad apprendere gli indirizzi MAC dei *frame* ricevuti sulle interfacce.

Possiamo concludere che i *Timer* e gli stati intermedi di *Listening* e *Learning* dello STP, sono essenziali per evitare che in caso di perturbazione e ricalcolo di una istanza STP si possa creare un loop.

### 3.4.3 Principali differenze tra STP e RSTP

Il protocollo *Rapid Spanning Tree Protocol* (RSTP) funziona in maniera molto simile rispetto lo STP tant'è che in una stessa LAN possono coesistere, ma quello che principalmente li differenzia è la velocità con cui convergono. Con valori di *Timers* predefiniti lo STP converge in 50 secondi mentre il RSTP in soli 10.

Il RSTP include dei metodi che evitano di attendere i *Timer* stabiliti dallo STP, velocizzando il passaggio dallo stato di *Forwarding* allo stato di *Discarding* (che comprende gli stati *Disabled* e *Blocking* dello STP) e viceversa.

Nella *Tabella 3-j* vediamo un confronto tra gli stati delle porte nei due protocolli.

Funzione	Stato STP	Stato RSTP
Porta amministrativamente disabilitata	<b><i>Disabled</i></b>	<b><i>Discarding</i></b>
Stato stabile che ignora i <i>frame</i> in arrivo sulla porta e non inoltra <i>frame</i>	<b><i>Blocking</i></b>	<b><i>Discarding</i></b>



Stato temporaneo senza apprendimento di <i>MAC address</i> e senza inoltro di <i>frame</i>	<b><i>Listening</i></b>	<b>Non utilizzato</b>
Stato temporaneo con apprendimento di <i>MAC address</i> e senza inoltro di <i>frame</i>	<b><i>Learning</i></b>	<b><i>Learning</i></b>
Stato stabile che consente l'apprendimento dei <i>MAC address</i> e l'inoltro di data <i>frame</i>	<b><i>Forwarding</i></b>	<b><i>Forwarding</i></b>

Tabella 3-g: Confronto degli stati delle porte tra STP e RSTP.

### 3.4.4 Funzionalità opzionali per STP

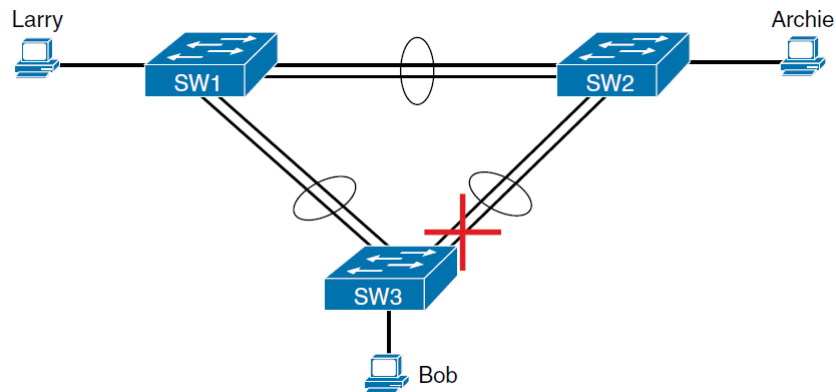
Vediamo in ultimo alcune funzionalità che rendono lo STP (ma anche le altre sue versioni) ancora più efficiente e sicuro: *EtherChannel*, *PortFast* e *BPDU Guard*.

#### 3.4.4.1 *EtherChannel*

Uno dei migliori modi per ridurre il tempo di convergenza dello STP è evitare del tutto il verificarsi della convergenza e quindi il ricalcolo di un'istanza STP. L'*EtherChannel* fornisce un modo per evitare che il ricalcolo dello STP sia necessario quando si verifica un *fault* su una singola porta o cavo.

Esso combina più link di uguale velocità (per un massimo di 8 link). che collegano una coppia di *switch*, creando quello che è definito un *bundle EtherChannel*. Per *bundle* si intende un insieme di link fisici raggruppati tra loro a formare un unico link logico, e di conseguenza se uno dei link fisici

fallisse ma almeno uno degli altri restasse attivo, il link in *bundle* permanerebbe attivo e non si renderebbe necessario il ricalcolo STP. Un esempio di *EtherChannel* è quello mostrato in *Figura 3-ee*.



*Figura 3-s: Esempio di implementazione EtherChannel.*

Per far sì che lo STP converga verso una nuova topologia, entrambi i link fisici, facenti parti del *bundle* che collegano due *switch*, devono interrompersi.

Il fatto di poter accorpare più link fisici in un *bundle* consente anche di raggiungere larghezze di banda superiori perché, il link risultante equivale ad un link con capacità trasmissiva pari alla somma delle singole capacità trasmissive dei link. Senza la configurazione di *EtherChannel* ciò non avverrebbe perché lo STP rileverebbe un bundle come un loop e di conseguenza bloccherebbe delle porte per evitarne la formazione.

#### 3.4.4.2 PortFast

La configurazione *PortFast* su un'interfaccia di uno *switch* consente a quest'ultima di passare immediatamente da uno stato di *Blocking/Discarding* allo stato di *Forwarding*, bypassando gli stati di *Listening* e di *Learning*. L'utilizzo di *PortFast* può risultare pericoloso perché sulle porte in cui viene abilitato è essenziale non collegare alcun bridge, *switch* o altri dispositivi che scambino messaggi STP altrimenti, il rischio è creare dei loop proprio perché vengono aboliti gli stati di *Listening* e *Learning*.

Il *PortFast* è indicato per le connessioni dei dispositivi utenti, in quanto all'avvio di un PC *host*, la porta dello *switch* a cui è collegato può passare immediatamente nello stato di *Forwarding*, senza attendere che lo *switch* confermi che la porta sia una *Designated Port*.

#### 3.4.4.3 BPDU Guard

I protocolli STP e RSTP aprono le LAN a diverse tipologie di possibili rischi per la sicurezza, ad esempio:

- Un utente malintenzionato potrebbe collegare ad uno *switch* (anche periferico) un nuovo *switch* con una priorità STP/RSTP appositamente di basso valore e diventare quindi il *Root Switch*. La nuova topologia STP/RSTP che si verrebbe a creare, porterebbe

senz'altro a prestazioni peggiorative rispetto alla topologia desiderata;

- L'attaccante potrebbe collegarsi a più porte di più *switch*, diventare *Root Switch* ed iniziare ad inoltrare gran parte del traffico in circolo nella rete LAN. L'attaccante potrebbe quindi utilizzare indisturbatamente un analizzatore di rete per copiare un gran numero di *frame* di dati inviati attraverso la LAN;
- Gli utenti potrebbero danneggiare involontariamente la LAN collegando un semplice ed economico *switch* LAN da ufficio. Un tale *switch* senza nessuna funzione STP/RSTP, non bloccherebbe nessuna porta e potrebbe causare un loop.

La funzione Cisco *BPDU Guard* aiuta a risolvere questo tipo di problemi disabilitando immediatamente la porta dello *switch* su cui è attiva, se su di essa vengono ricevute BPDU. Questa funzione è particolarmente utile sulle porte che devono essere utilizzate solo come porte di accesso.

Deduciamo che la funzione *BPDU Guard* crea una forte sinergia con la funzione *PortFast* vista prima. L'utilizzo contemporaneo di entrambe le su una stessa porta, consente di avere interfacce dello *switch* immediatamente pronte in accesso, e quindi in *Forwarding*, senza correre il rischio che vi possa essere collegato uno *switch*.

### 3.5 Il Per-VLAN *Spanning Tree Protocol*

Con l'avvento delle VLAN dalla seconda metà degli anni '90, si cercò sin da subito di integrarle con lo STP, per ottenere una soluzione ancora più performante. Lo STP come lo conosciamo, crea un'istanza logica che è la stessa a prescindere dalle VLAN, e per questo l'IEEE e Cisco si misero al lavoro per creare una nuova tecnica di *Spanning Tree* che integrasse il concetto di VLAN in modo tale che il traffico potesse essere bilanciato su tutti i link disponibili senza cioè lasciare dei link inutilizzati perché mantenuti in *Blocking* dallo STP. Fu così che Cisco brevettò per prima una tecnica rivoluzionaria ovvero il *Per VLAN Spanning Tree Plus* (PvST+), il cui funzionamento è mostrato nella *Figura 3-ff*. Il protocollo PvST+ unisce il concetto di VLAN e STP, con il risultato che per ogni VLAN può essere eletto un *Root Bridge* diverso; questo comporta che le porte in *Blocking* su uno *switch* periferico per via di un'istanza STP dedicata ad una VLAN, non debbano per forza essere in *Blocking* per un'altra.

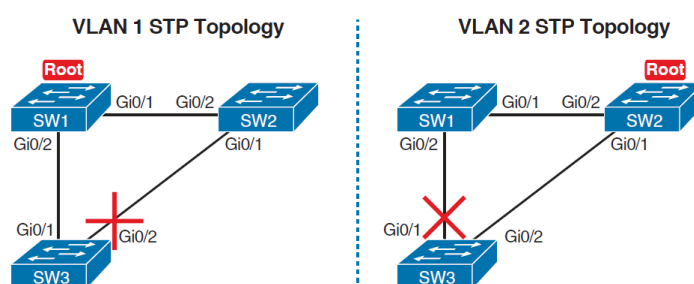


Figura 3-t: Bilanciamento di traffico con due istanze Spanning Tree per le due VLAN 1 e VLAN 2.

Successivamente quando la Cisco introdusse il RSTP, aggiornò anche il suo personalissimo PvST+ in RIPvST+, ancora più efficiente e rapido nei

tempi di convergenza. La IEEE giunse invece all'elaborazione del protocollo *Multiple Spanning Tree Protocol* (MSTP) che racchiude le caratteristiche del RIPvST+ e che tutti i vendor devono includere nei propri apparati di switching.

Gli *switch* Cisco ci offrono varie opzioni per la configurazione dello *Spanning Tree*. Possono utilizzare sia i protocolli proprietari Cisco come PvST+ basato su STP, e RIPvST+ basato su RSTP, oppure gli standard IEEE come MSTP. La *Tabella 3-k* riassume alcune caratteristiche su questi standard e anche, delle parole chiave utilizzate nel comando di configurazione degli *switch* Cisco.

Nome	Basato su STP o RSTP	N° di alberi	IEEE standard	Parametro di config
<b>STP</b>	STP	1	802.1D	N/A
<b>PvST+</b>	STP	1/VLAN	802.1D	pvst
<b>RSTP</b>	RSTP	1	802.1w	N/A
<b>Rapid PvST+</b>	RSTP	1/VLAN	802.1w	Rapid-pvst
<b>MSTP</b>	RSTP	1 o più <sup>3</sup>	802.1s	mst

*Tabella 3-h: Standard STP attualmente sul mercato.*

---

<sup>3</sup> MSTP consente la definizione di tante istanze *Spanning Tree* quante sono le scelte progettuali di rete ma non ne richiede una per ogni VLAN.

### 3.5.1 Il Bridge ID e il System ID Extension

L'intuizione che portò alla creazione di PvST+, RPvST+ e MSTP fu quella di rivedere la definizione del Bridge ID di uno *switch*. Per poter creare istanze *Spanning Tree* diverse per ogni VLAN, è stato pensato di comprendere nel Bridge ID l'identificativo proprio delle VLAN, ovvero il VLAN ID (di 12 bits). Per fare questo il campo *Priority* originale è stato diviso in due parti, come mostrato nella *Figura 3-gg*: un campo di *Priority* ridotto a 4 bit, e un sottocampo a 12 bit chiamato System ID Extension (che rappresenta il VLAN ID).

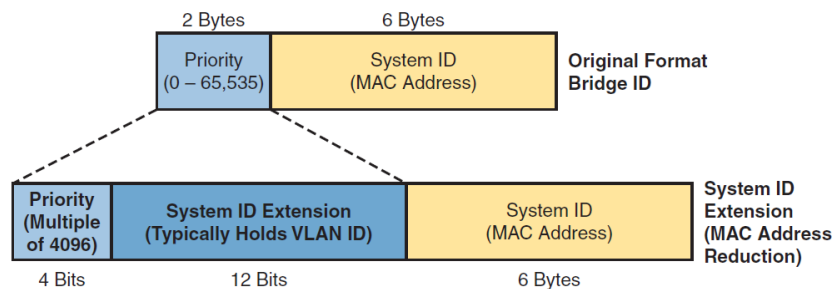


Figura 3-u: STP System ID extension.

Questo fa sì che la *Priority* in un PvST+ non potrà più assumere qualsiasi valore compreso tra 0 e 65.535, bensì sarà dato dalla somma di un multiplo di 4.096 e il VLAN ID delle singole VLAN.

Per capire meglio il motivo di questa affermazione osserviamo *Tabella 3-l*.

<b>Valore decimale</b>	<b>Binario a 16-bit equivalente</b>	<b>Valore decimale</b>	<b>Binario a 16-bit equivalente</b>
<b>0</b>	<b>0000 0000 0000 0000</b>	<b>32768</b>	<b>1000 0000 0000 0000</b>
<b>4096</b>	<b>0001 0000 0000 0000</b>	<b>36864</b>	<b>1001 0000 0000 0000</b>
<b>8192</b>	<b>0010 0000 0000 0000</b>	<b>40960</b>	<b>1010 0000 0000 0000</b>
<b>12288</b>	<b>0011 0000 0000 0000</b>	<b>45056</b>	<b>1011 0000 0000 0000</b>
<b>16384</b>	<b>0100 0000 0000 0000</b>	<b>49152</b>	<b>1100 0000 0000 0000</b>
<b>20480</b>	<b>0101 0000 0000 0000</b>	<b>53248</b>	<b>1101 0000 0000 0000</b>
<b>24576</b>	<b>0110 0000 0000 0000</b>	<b>57344</b>	<b>1110 0000 0000 0000</b>
<b>28672</b>	<b>0111 0000 0000 0000</b>	<b>61440</b>	<b>1111 0000 0000 0000</b>

*Tabella 3-i: Valori di Priority configurabili per PvST+, RPvST+ e MSTP.*



## 4. IL SIMULATORE GNS3

GNS3 (*Graphical Network Simulator*) è un software open source che permette di simulare reti complesse in maniera realistica, senza aver bisogno di hardware di rete dedicato.



*Figura 4-a: Logo GNS3.*

GNS3 è utilizzato per emulare, configurare, testare e risolvere problemi di reti virtuali e reali. A seconda della grandezza del network che si vuole riprodurre, GNS3 può essere eseguito direttamente su un laptop per piccole topologie, ma anche su server dedicati o addirittura in cloud per grandi topologie con molti dispositivi.

Nel presente progetto di tesi il software GNS3 è stato utilizzato per la riproduzione in ambiente virtuale della rete LAN dell'Azienda Cooperativa Agricola Cooperlat di Jesi (AN), con lo scopo di analizzare ed eventualmente proporre migliorie in termini di prestazioni e robustezza del network.

## 4.1 Cos'è GNS3, come è strutturato

GNS3 è costituito da due componenti software di base:

- Il software GNS3 *all-in-one* (GUI);
- La macchina virtuale (VM) GNS3.

GNS3 *all-in-one* può essere definito come la parte client di GNS3, ovvero consiste nell'interfaccia grafica (GUI) con cui l'utente può creare topologie più o meno complesse in estrema semplicità, trascinando e rilasciando i dispositivi nel workspace ed ottenendo un risultato simile a quello mostrato in *Figura 4-b*. Il software GNS3 *all-in-one* è possibile installarlo gratuitamente su tutti i principali sistemi operativi quali Windows, MAC e Linux.

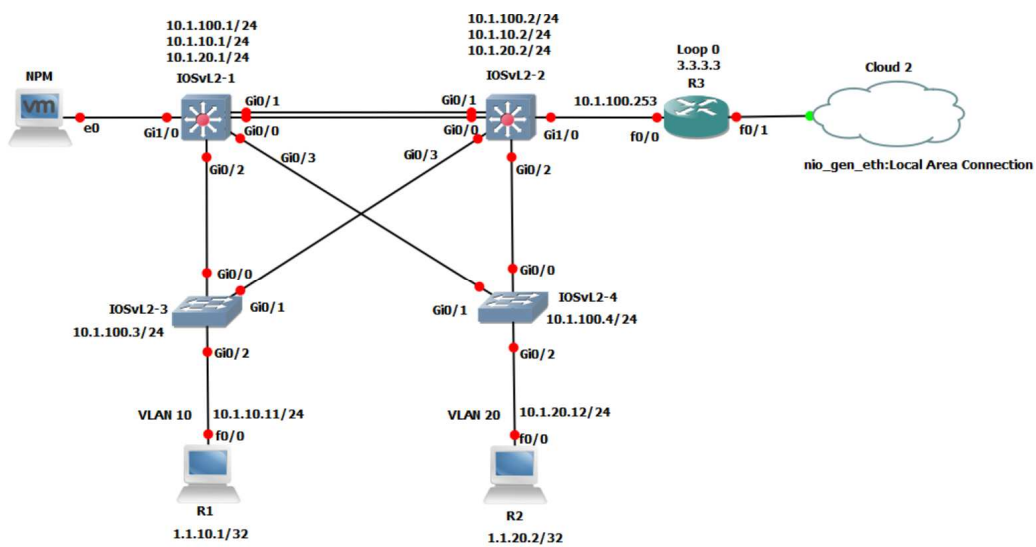


Figura 4-b: Esempio di topologia su Workspace GUI GNS3.

Quando creiamo topologie in GNS3 utilizzando la GUI del software *all-in-one*, i dispositivi devono però essere eseguiti da un server e per fare questo si hanno tre opzioni:

- Server GNS3 locale;
- Macchina virtuale GNS3 locale;
- Macchina virtuale GNS3 remota.

Il server GNS3 locale viene eseguito sullo stesso PC su cui è stato installato il software *all-in-one* GNS3. Questo è un buon metodo per iniziare a muovere i primi passi in GNS3 ma tale configurazione è limitata e non offre molte scelte per quanto riguarda le dimensioni della topologia e i dispositivi supportati.

Per questo è fortemente consigliato l'utilizzo della macchina virtuale GNS3 VM (VM GNS3).

La VM GNS3 può essere eseguita sia localmente su PC, utilizzando ad esempio software di virtualizzazione come VMware Workstation, Virtualbox o Hyper-V, sia in remoto su un server o anche in ambiente cloud.

Per riprodurre il funzionamento di reali dispositivi di rete, GNS3 supporta due tipologie di dispositivi: emulati e simulati.

- Dispositivi emulati: viene eseguito in tutto e per tutto il sistema operativo del dispositivo di network reale. Ad esempio è possibile

virtualizzare il sistema operativo Cisco IOS da un router Cisco fisico reale ed eseguirlo come un router Cisco virtuale e quindi emulato in GNS3;

- Dispositivi simulati: GNS3 simula le caratteristiche e le funzionalità di un dispositivo come uno *switch*, senza quindi eseguire sistemi operativi reali.

### 4.1.1 Cos'è una Macchina Virtuale

È stato già nominato diverse volte il termine “Macchina Virtuale” e altrettanti sistemi di virtualizzazione come VMware, Virtualbox o Hyper-V, ma prima di addentrarci nel funzionamento di GNS è bene definire cosa effettivamente sia una macchina virtuale.

Una macchina virtuale, comunemente abbreviata in VM (dall'inglese *Virtual Machine*), non è diversa da qualsiasi altro computer fisico come laptop, smartphone o server. Ha una CPU, una memoria (RAM), dischi (storage) e può connettersi ad Internet se necessario. Sebbene le parti che compongono un computer siano fisiche e tangibili, le macchine virtuali sono definite dal software all'interno di server fisici.

Alla base di una VM vi è un software chiamato *hypervisor*, o gestore di macchine virtuali, che crea un *layer* di astrazione sopra l'hardware in modo

che le macchine virtuali possano utilizzare le risorse fisiche in modo concorrente secondo precisi algoritmi di scheduling.

### **4.1.2 La VM GNS3**

La VM GNS3 è un requisito fondamentale se lo scopo è eseguire dispositivi basati su emulatore *Qemu* su Windows o Mac OS. È una macchina virtuale il cui file *iso* è facilmente scaricabile dal sito ufficiale di GNS3 e altrettanto semplice da importare in ambiente VMware Workstation o VirtualBox.

Se invece si necessita solo di creare topologie GNS3 basilari e costituite da dispositivi che utilizzano immagini IOS la VM GNS3 non è necessaria. In questo caso è sufficiente il software *all-in-one* GNS3. Quest'ultima è definita modalità *legacy* o modalità *Dynamips*.

### **4.1.3 Emulatori GNS3**

GNS3 supporta più emulatori che possono essere utilizzati nei progetti e ciò offre molta flessibilità durante la creazione di topologie più o meno complesse.

*Dynamips* è la tecnologia sfruttata da GNS3 sin dalle prime versioni, ed emula i router Cisco e lo switching di base. Emula hardware Cisco ormai datato come il router Cisco 3725, e utilizza immagini Cisco IOS reali. Con

questa tipologia di emulatore è possibile copiare un'immagine IOS da un dispositivo di rete fisico, e utilizzarla con GNS3. Cisco, d'altro canto, non supporta l'utilizzo di immagini IOS su hardware non Cisco, e per questo è molto probabile riscontrare bug nell'emulazione.

La maggior parte di aziende produttrici di hardware di networking offre anche immagini *Qemu* che possono essere utilizzate con GNS3 con l'omonimo emulatore. *Qemu* è la miglior scelta per la creazione e l'utilizzo di topologie Cisco in GNS3 in quanto sono portatili e più leggere rispetto alle immagini *Dynamips*.

#### **4.1.4 Il toolkit Wireshark**

Durante il setup di installazione di GNS3, viene proposto all'utente di installare o meno alcuni software integrati nelle funzionalità di GNS3.

Nel nostro caso, come mostrato in *Figura 4-c*, è stato compreso nell'installazione il software Wireshark, applicativo per analisi di protocollo o anche *packet sniffer*, utilizzato per la risoluzione di problemi di rete, per il *troubleshooting* ma anche per lo sviluppo di protocolli o software di comunicazione.

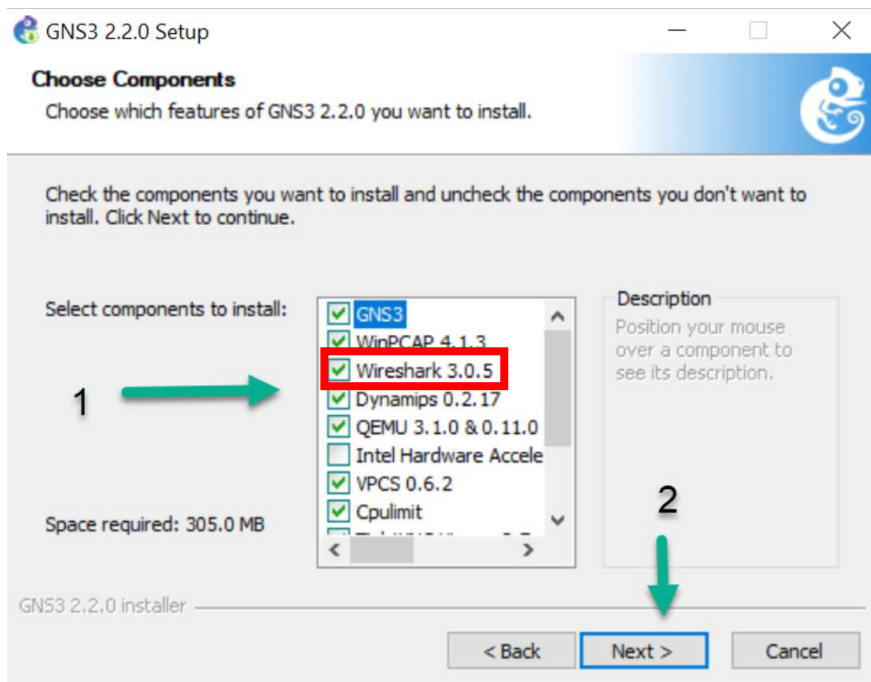


Figura 4-c: Installazione di Wireshark tra i toolkit disponibili nel setup GNS3.

Vedremo in seguito l'utilizzo che ne è stato fatto per i nostri scopi.

## 4.2 Vantaggi e criticità nell'uso di GNS3

Come accennato GNS3 è un software open source che è possibile scaricare e utilizzare gratuitamente. Tra i vari vantaggi che possiamo annoverare oltre la fruibilità gratuita ci sono:

- Nessuna limitazione sul numero di dispositivi supportati (l'unica limitazione è l'hardware: CPU e memoria del PC su cui GNS3 è installato);

- Supporta più opzioni di switching (modulo Etherswitch NM-ESW16, immagini IOU/IOL *Layer 2*, VIRL IOSvL2);
- Supporta tutte le immagini VIRL<sup>4</sup> (IOSv, IOSvL2, IOS-XRv, CSR1000v, NX-OSv, ASAv);
- Supporta ambienti multi-vendor;
- Può essere eseguito con o senza hypervisor;
- Supporta hypervisor sia gratuiti che a pagamento (Virtualbox, workstation VMware, player VMware, ESXi, Fusion);
- In continua evoluzione grazie al codice sorgente open source.

I principali svantaggi sono:

- GNS3 può essere limitato dalle prestazioni del PC a causa dell'installazione locale (le prestazioni della simulazione degradano vertiginosamente per topologie complesse se l'hardware a disposizione non è sufficientemente prestante);
- Impossibilità di simulare stack di *switch* (sono invece molto utilizzati in campo reale, compreso in ambiente Cooperlat).

---

<sup>4</sup> Il Virtual Internet Routing Lab (VIRL) di Cisco è una tecnologia di virtualizzazione a pagamento e di cui è proprietaria Cisco, che consente lo sviluppo di modelli di reti esistenti o immaginarie. Gli utenti hanno la possibilità di progettare, costruire, visualizzare, risolvere i problemi e avviare simulazioni di dispositivi Cisco e di terze parti in un ambiente virtuale



## 4.3 Installazione GNS3 e prima topologia

Per l'installazione della GUI GNS3 e della VM dedicata, ci si è basati sulla documentazione consultabile al sito ufficiale GNS3, dove è riportata passo dopo passo la procedura da seguire in base alla piattaforma di cui si dispone.

Nel nostro caso si è eseguita l'installazione per sistema Windows, dapprima del software GNS3 *all-in-one* e in secondo luogo della VM GNS3 in ambiente VMWare Workstation 16 Player.

Dopo aver installato VMWare Player e importato il file ISO della VM GNS3, è stato molto semplice effettuare l'operazione di *“join”* alla GUI GNS3.

Come mostrato in *Figura 4-d* è sufficiente recarsi nel menu *“Preferences”* alla voce GNS3 VM, abilitare con una tick l'utilizzo della VM e selezionare il motore di virtualizzazione dove si è precedentemente importata la VM.

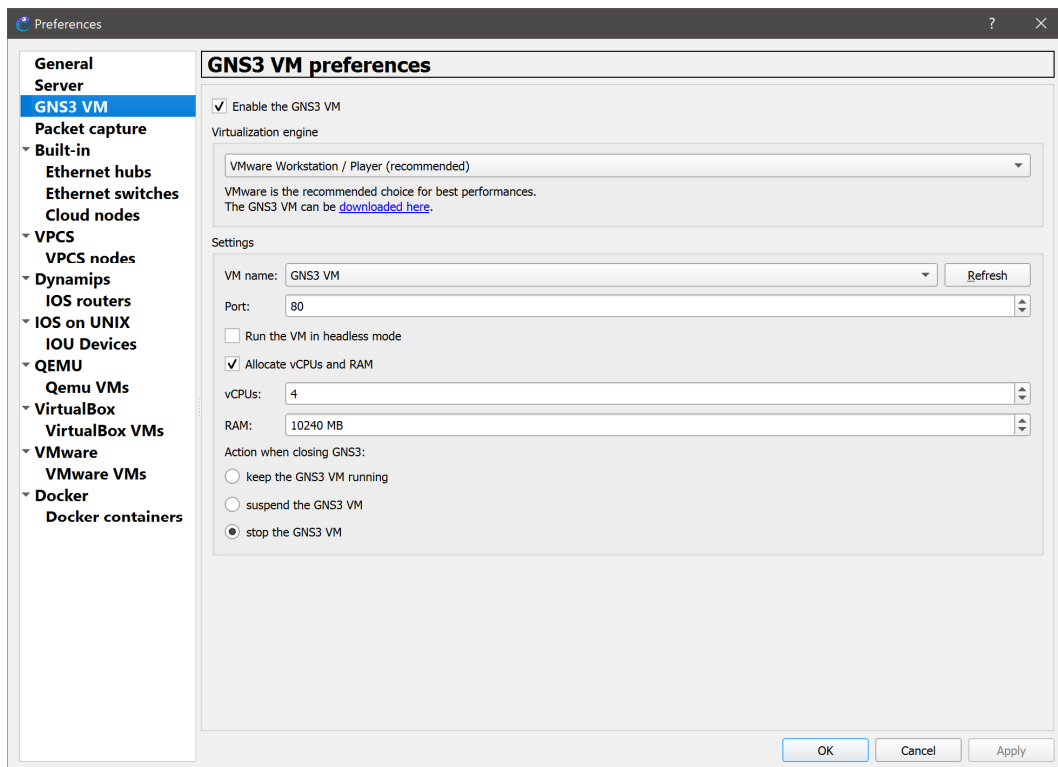


Figura 4-d: Impostazioni per la GNS3 VM utilizzata nel progetto.

In questo modo tutte le volte che verrà lanciata la GUI GNS3, contestualmente verrà accesa anche la GNS3 VM, e potremo tenere sotto controllo il suo stato e i livelli di CPU e RAM, nel comodo riquadro in basso a destra della GUI.

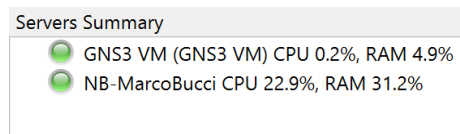
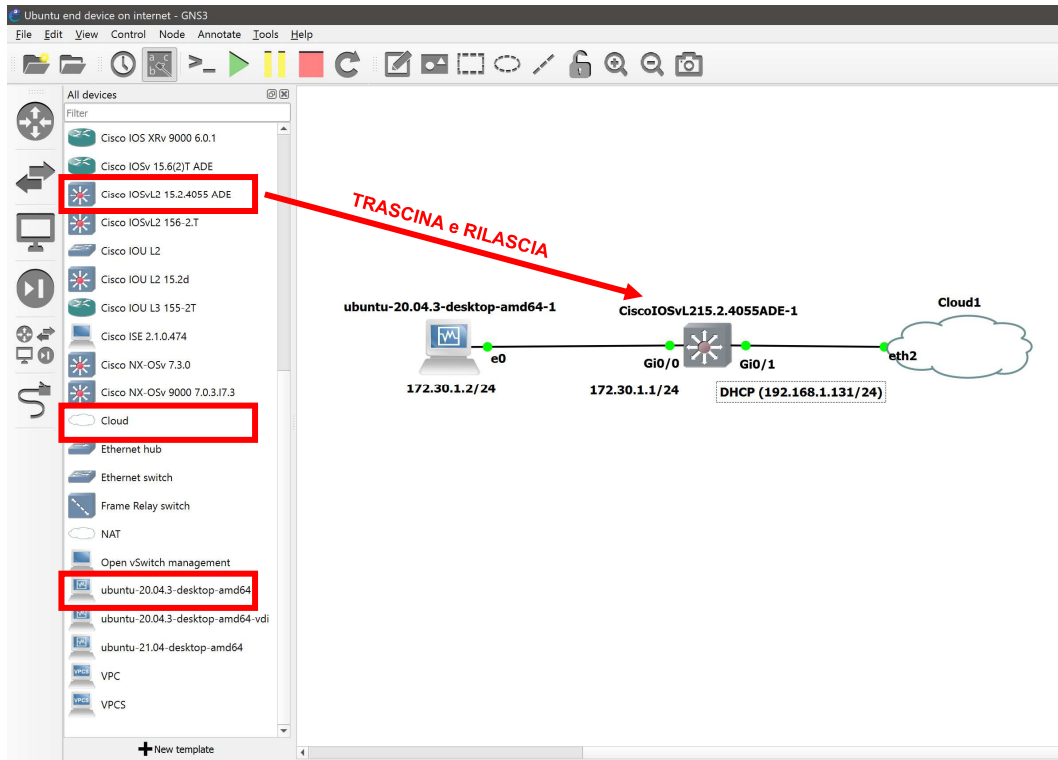


Figura 4-e: Stato della GNS3 VM e del server local host GNS3.

Terminata l'installazione si è passati subito all'implementazione di prime e semplici topologie.

Un esempio è quello riportato in *Figura 4-f*. In questo caso, è stato testato l'utilizzo del nodo Cloud, di un end point con sistema operativo Ubuntu e di uno *switch* Cisco IOSvL2<sup>5</sup> con funzione di intermediario tra i due.



*Figura 4-f: Topologia test per il nodo Cloud e l'utilizzo di end point Ubuntu.*

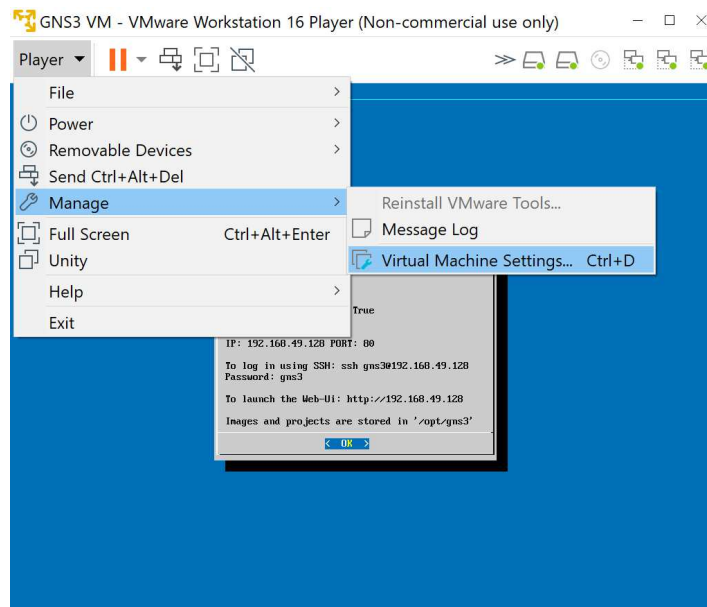
---

<sup>5</sup> IOSvL2 è un'implementazione del codice di switching di *layer 2* di Cisco IOS e che viene eseguito come una macchina virtuale completa. Le immagini IOSvL2 supportano fino a 16 interfacce GigabitEthernet. IOSvL2 è principalmente uno *switch* di *layer 2*, ma nell'immagine sono presenti anche funzionalità di routing tipiche del *layer 3* TCP/IP.

### 4.3.1 Cloud Node e sua configurazione

Il nodo cloud è fondamentale perché permette di collegare la nostra topologia virtuale GNS3 con una rete esterna (potenzialmente la WAN internet) e viceversa. Per farlo, è necessario effettuare una piccola modifica alla VM GNS3.

In particolare, come visibile in *Figura 4-g*, dalla finestra VMWare Player, alla sezione *Manage>Virtual Machine Settings..*, è stata aggiunta una terza scheda di rete virtuale con funzione di *bridge*.



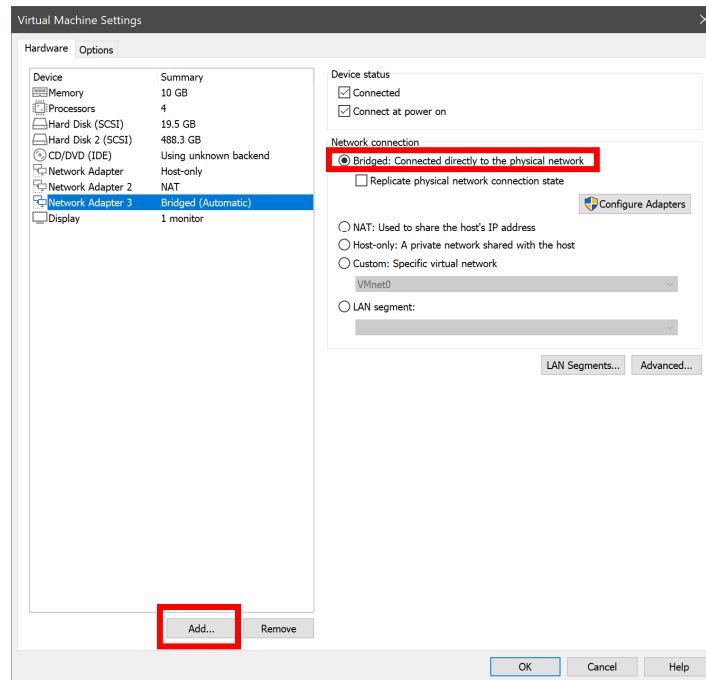


Figura 4-g: Aggiunta interfaccia di rete bridge alla VM GNS3.

A questo punto, una volta posizionato in topologia il nodo cloud, facendo un click con il tasto destro del mouse e successivamente un click su “*Configure*”, viene visualizzata una finestra di configurazione del nodo Cloud dove con un click va prima abilitata la visualizzazione di tutte le schede ethernet (*Show special Ethernet interfaces*) e poi aggiunta la scheda *eth2* corrispondente alla scheda *bridge* aggiunta alla VM.

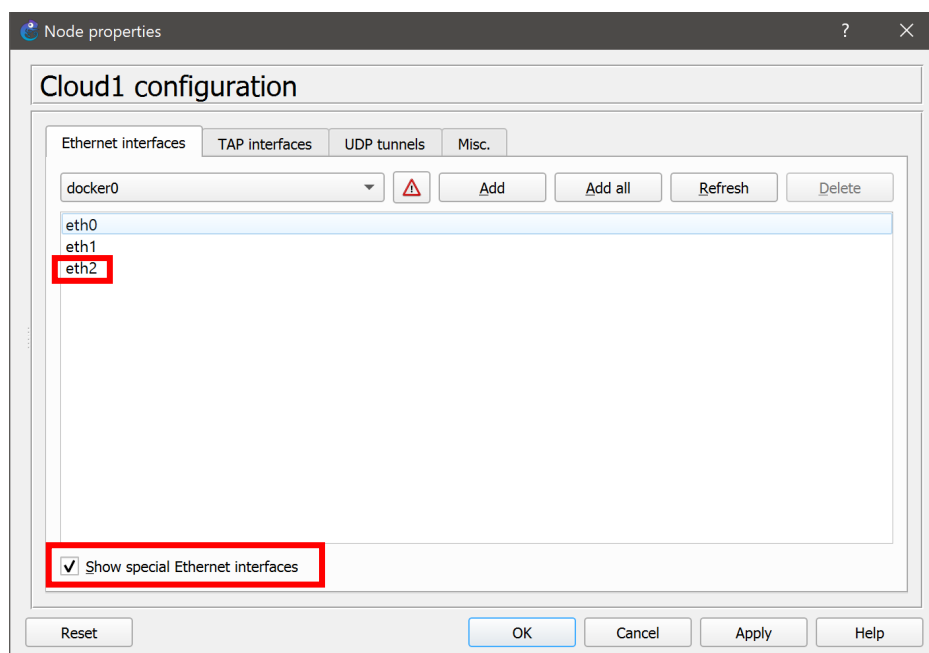
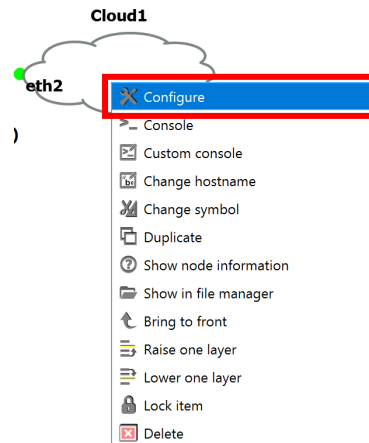


Figura 4-h: Aggiunta della scheda bridge al nodo Cloud.

In tal modo, collegando lo *switch* IOSvL2 all'interfaccia *eth2* del nodo cloud, equivale concettualmente a collegare fisicamente con un cavo di rete lo *switch* all'ISR domestico, descritto nel capitolo 3.1.1 dedicato alle SOHO LAN.

## 4.3.2 VirtualBox End Device

Per poter simulare il funzionamento di un virtual server Linux Ubuntu in topologia, si è sfruttata l'integrazione offerta tra GNS3 e l'ambiente di virtualizzazione Oracle VirtualBox. Dopo aver scaricato e installato VirtualBox, è sufficiente scaricare l'ultima ISO Ubuntu disponibile dal sito *Ubuntu.com*, e successivamente importarla in VirtualBox.

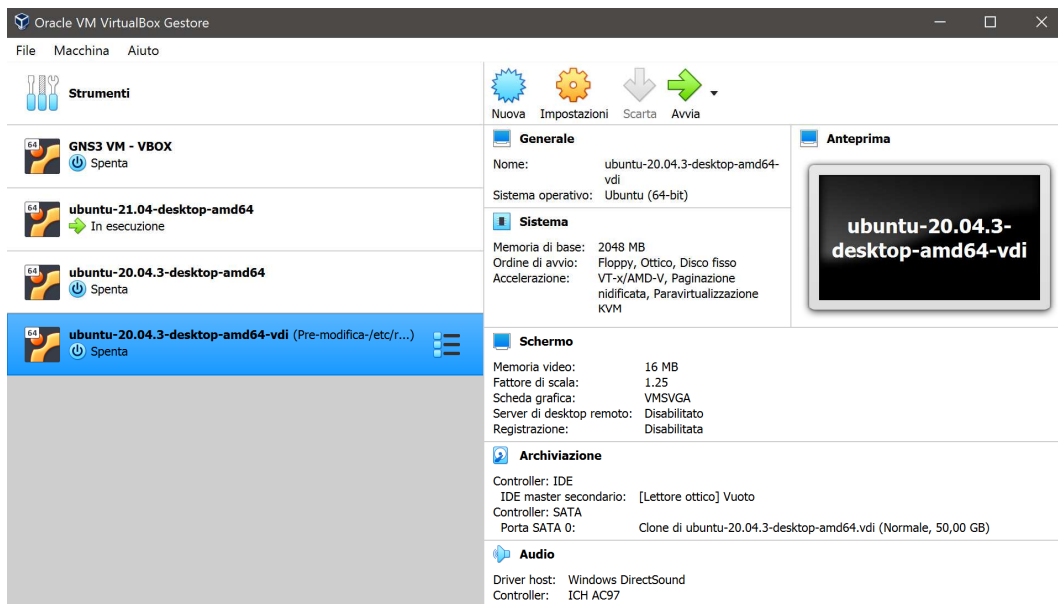


Figura 4-i: Dashboard Oracle VirtualBox con VM Ubuntu.

A questo punto dal menu “*Preferences*” della GUI GNS3, alla voce “*VirtualBox VM*”, con un click sul tab *New* si avvia una procedura guidata che si conclude con l'import della VM Ubuntu tra gli end device fruibili nella GUI GNS3.

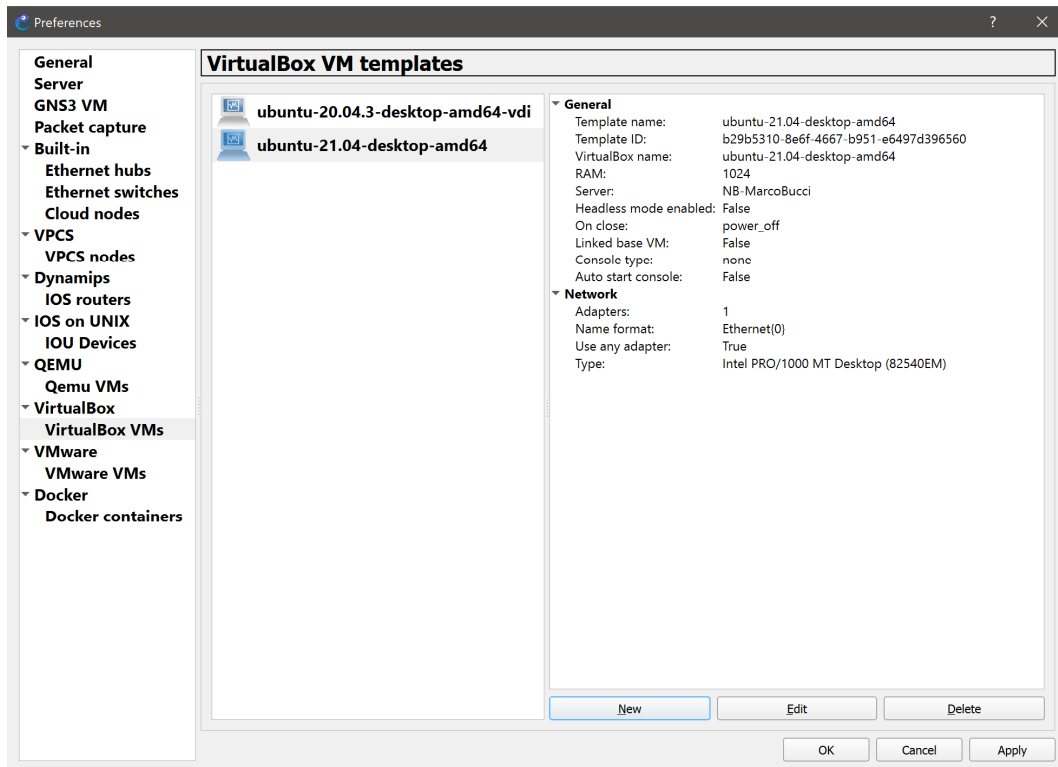


Figura 4-j: Import della VM Ubuntu.

### 4.3.3 Configurazione e test dei dispositivi

Una volta che tutti i device di interesse sono stati posizionati nella dashboard GNS3 e collegati tra loro (come mostrato in *Figura 4-k*), si è passati alla loro configurazione.

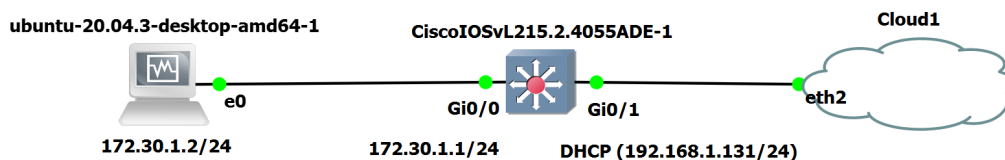


Figura 4-k: Topologia di test Ubuntu end device on Internet.



Partendo dallo *switch* IOSvL2, si è da subito configurata la porta Gi0/1 in modo da non operare come switchport (ovvero non in *Layer 2*), ma come interfaccia *Layer 3* in DHCP, per poter veicolare il traffico IP da e verso internet.

```
interface GigabitEthernet0/1
  description Link to External
  no switchport
  ip address dhcp
  negotiation auto
  no cdp enable
end
```

Si è poi scelto di assegnare alla rete virtuale GNS3 la sottorete 172.30.1.0/24 perciò, sempre dallo *switch*, è stata assegnata all'interfaccia virtuale VLAN 1 l'indirizzo IP 172.30.1.1/24, decretandolo di fatto Default Gateway per la sottorete virtuale, in quanto potrà raggiungere grazie al nodo Cloud entrambe le sottoreti (rete LAN virtuale e rete LAN domestica), ed avere quindi funzione di routing.

```
interface Vlan1
  ip address 172.30.1.1 255.255.255.0
end
```

A questo punto la configurazione dello *switch* risulta essere quella mostrata nelle Figure 4-k, dove possiamo notare l'indirizzo IP assegnato staticamente all'interfaccia VLAN 1 dello *switch* (172.30.1.1/24), mentre

l'interfaccia GigabitEthernet0/1, configurata in assegnazione DHCP, ha ricevuto dinamicamente l'indirizzo 192.168.1.131/24 da parte dell'ISR domestico.

```
Switch#show ip interface brief
Interface          IP-Address      OK? Method Status Protocol
GigabitEthernet0/0 unassigned      YES unset  up      up
GigabitEthernet0/2 unassigned      YES unset  down    down
GigabitEthernet0/3 unassigned      YES unset  down    down
GigabitEthernet0/1 192.168.1.131  YES DHCP   up      up
GigabitEthernet1/0 unassigned      YES unset  down    down
GigabitEthernet1/1 unassigned      YES unset  down    down
GigabitEthernet1/2 unassigned      YES unset  down    down
GigabitEthernet1/3 unassigned      YES unset  down    down
GigabitEthernet2/0 unassigned      YES unset  down    down
GigabitEthernet2/1 unassigned      YES unset  down    down
GigabitEthernet2/2 unassigned      YES unset  down    down
GigabitEthernet2/3 unassigned      YES unset  down    down
GigabitEthernet3/0 unassigned      YES unset  down    down
GigabitEthernet3/1 unassigned      YES unset  down    down
GigabitEthernet3/2 unassigned      YES unset  down    down
GigabitEthernet3/3 unassigned      YES unset  down    down
Vlan1              172.30.1.1     YES manual up      up
```

```
Switch#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

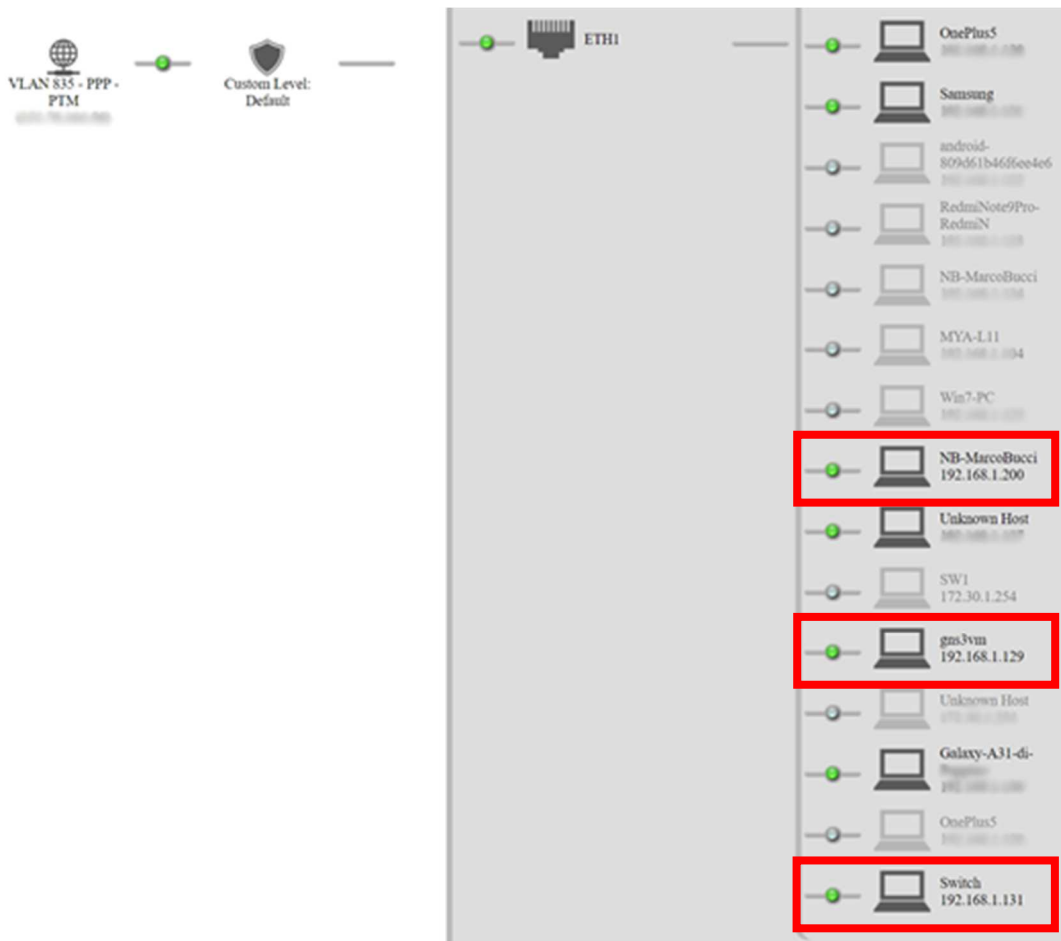
Gateway of last resort is 192.168.1.1 to network 0.0.0.0

S*   0.0.0.0/0 [254/0] via 192.168.1.1
     172.30.0.0/16 is variably subnetted, 2 subnets, 2 masks
C    172.30.1.0/24 is directly connected, Vlan1
L    172.30.1.1/32 is directly connected, Vlan1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, GigabitEthernet0/1
L    192.168.1.131/32 is directly connected, GigabitEthernet0/1
```

Figura 4-I: Risultato della configurazione dello switch.

Nella seconda parte della *Figura 4-I* è anche possibile osservare la *Tabella di routing* che lo *switch Layer 3* ha creato autonomamente, e dove è possibile osservare che le due sottoreti 172.30.1.0/24 e 192.168.1.0/24, sono rispettivamente raggiungibili dalla VLAN1 e dall'interfaccia Gi0/1.

A riprova della “connessione” tra ambiente virtuale e ambiente reale offerta dal nodo cloud, dall’interfaccia web dello stesso ISR come mostrato in *Figura 4-l*, è possibile notare l’assegnazione tramite servizio DHCP degli indirizzi IP sia alla GNS3VM che all’interfaccia GigabitEthernet0/1 dello switch IOSvL2.



*Figura 4-m: Web Interface ISR D-Link dove sono visibili gli indirizzi IP assegnati al NB-MarcoBucci, alla GNS3VM e allo Switch IOSvL2.*

A questo punto si è passati alla configurazione dell’end point Ubuntu direttamente dalla sua interfaccia desktop in ambiente VirtualBox.

Dalla sezione *Impostazioni>Via Cavo>IPv4*, è stato sufficiente assegnare un indirizzo appartenente alla sottorete della VLAN 1 (in questo caso 172.30.1.2/24), il gateway corrispondente all'indirizzo 172.30.1.1/24, e un indirizzo per il server DNS (8.8.8.8, Google DNS).

The screenshot shows the 'Cavo' (Wired) network configuration window. At the top, there are buttons for 'Annulla' (Cancel), 'Cavo', and 'Applica' (Apply). Below this is a navigation bar with tabs for 'Dettagli' (Details), 'Identità' (Identity), 'IPv4', 'IPv6', and 'Sicurezza' (Security). The 'Dettagli' tab is selected and underlined. The configuration details are as follows:

Velocità collegamento	1000 Mb/s
Indirizzo IPv4	172.30.1.2
Indirizzo IPv6	fe80::bccb:d9f1:923f:3e3d
Indirizzo hardware	08:00:27:E4:17:F6
Instradamento predefinito	172.30.1.1
DNS	8.8.8.8

Below the configuration details, there are three checkboxes:

- Connettere automaticamente
- Rendere disponibile agli altri utenti
- Connessione a consumo: ha un limite sui dati o può avere costi aggiuntivi  
Gli aggiornamenti software, e altri scaricamenti di dati, non verranno avviati automaticamente.

At the bottom right, there is a red button labeled 'Rimuovi profilo connessione' (Remove connection profile).

*Figura 4-n: Configurazione parametri di rete dell'End Point Ubuntu.*

Giunti qui però l'end device non può ancora navigare in internet, perché il "mondo esterno" (WAN internet) non sa come poter raggiungere la rete 172.30.1.0/24; per far questo è stato necessario implementare una regola di routing direttamente nell'ISR e mostrata in *Figura 4-o*.

Destination IP / SubnetMask	Gateway IP Address	Interface	Traffic Classes	Metric	Type	Status	Enabled
Network 172.30.1.1/255.255.255.0		Bridge Ethernet WiFi (192.168.1.1)	None		Static	Enabled	<input checked="" type="checkbox"/>
Default	151.6.152.24	VLAN 835 - PPP - PTM (151.76.164.56)	None		Dynamic	Enabled	<input checked="" type="checkbox"/>

Figura 4-o: Regola di routing aggiunta all'ISR.

In questo modo l'ISR è in grado di inoltrare il traffico proveniente da internet con destinatario un end device nella sottorete 172.30.1.0/24, utilizzando l'interfaccia Bridge Ethernet 192.168.1.1.

Ora è possibile apprezzare l'end point Ubuntu navigare in internet e contattare web server esterni.

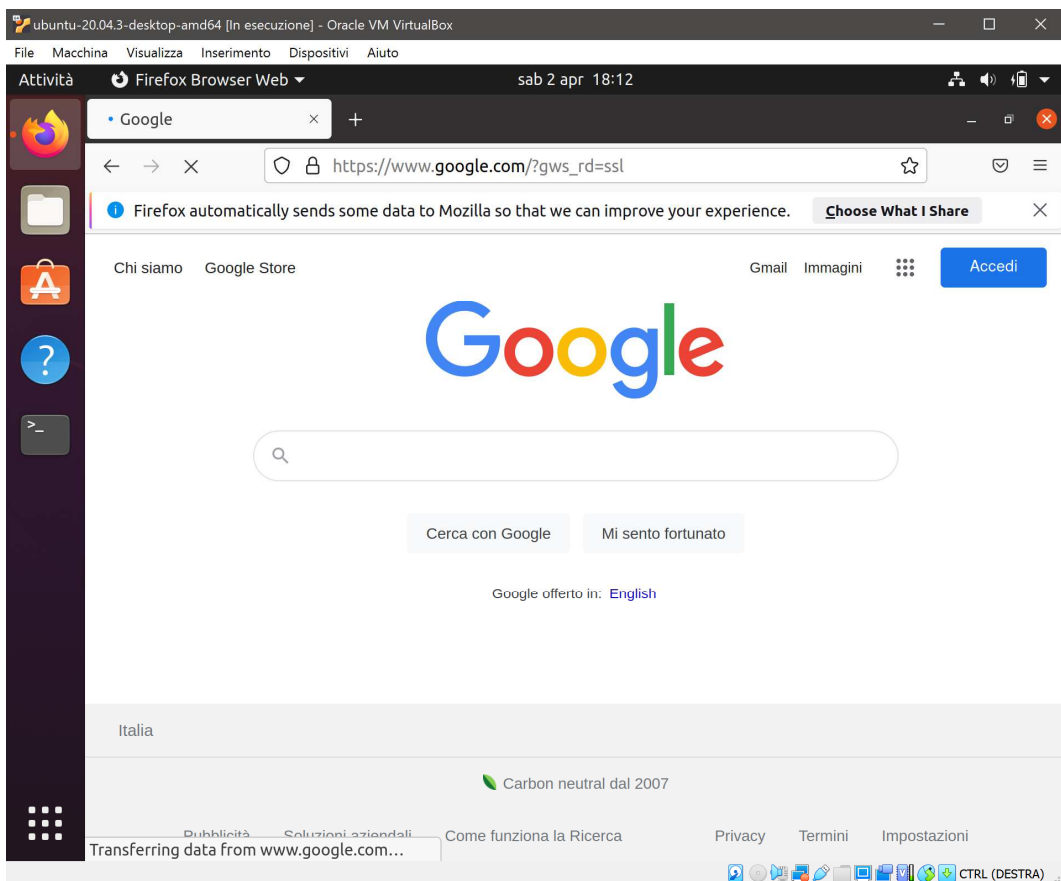


Figura 4-p: Console VirtualBox con Ubuntu Desktop connesso ad internet.

In questa fase è stato riscontrato un bug a *layer* di interazione tra GNS3 e VirtualBox, che impedisce la navigazione tramite il nodo cloud in esecuzione su GNS3VM. La risoluzione sta nel porsi all'interno del pannello di controllo Windows, alla voce "Connessioni di rete", e disabilitare/abilitare l'adattatore virtuale "VirtualBox Host-Only Network".

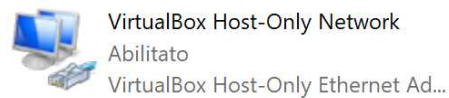


Figura 4-q: Bug di integrazione tra VirtualBox e GNS3VM.

Questo si rende necessario ogni qual volta si avvia l'esecuzione della topologia, e quindi tendenzialmente solo ad inizio simulazione.

#### 4.3.4 NAT node

A partire dalla versione 2.0, GNS3 ha reso disponibile l'utilizzo del nodo NAT. Questo nodo consente di connettere una topologia ad Internet tramite NAT in maniera molto semplice ed intuitiva rispetto al nodo cloud visto precedentemente.

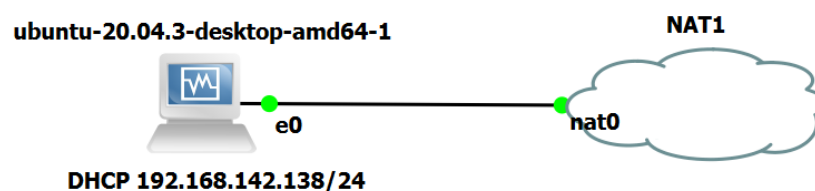


Figura 4-r: Esempio di utilizzo del nodo NAT.

Tale nodo risulta quindi molto utile quando è necessario scaricare files da Internet come aggiornamenti di OS, librerie oppure l'attivazione di una licenza.

Il nodo NAT per poter funzionare richiede la VM GNS3 o un computer con OS Linux e pacchetto *libvirt* installato.

Per impostazione predefinita, il nodo NAT esegue un server DHCP, nel nostro caso con un pool di indirizzi della sottorete 192.168.142.0/24, per cui basterà configurare la porta del dispositivo, direttamente collegato al NAT node, in configurazione DHCP per poter ricevere i parametri di rete ed iniziare da subito a navigare in internet.

Alla base del nodo NAT vi è una scheda di rete virtuale direttamente creata in ambiente Windows e individuabile in *Figura 4-s*. Ecco quindi la sostanziale differenza dal nodo Cloud, in quanto la topologia non sarà direttamente accessibile da Internet (come invece avviene nel caso Cloud), ma comunque permette uno scambio di dati tra topologia virtuale GNS3 e PC *host* su cui è installato GNS3 (in quanto il PC può raggiungere la sottorete NAT).

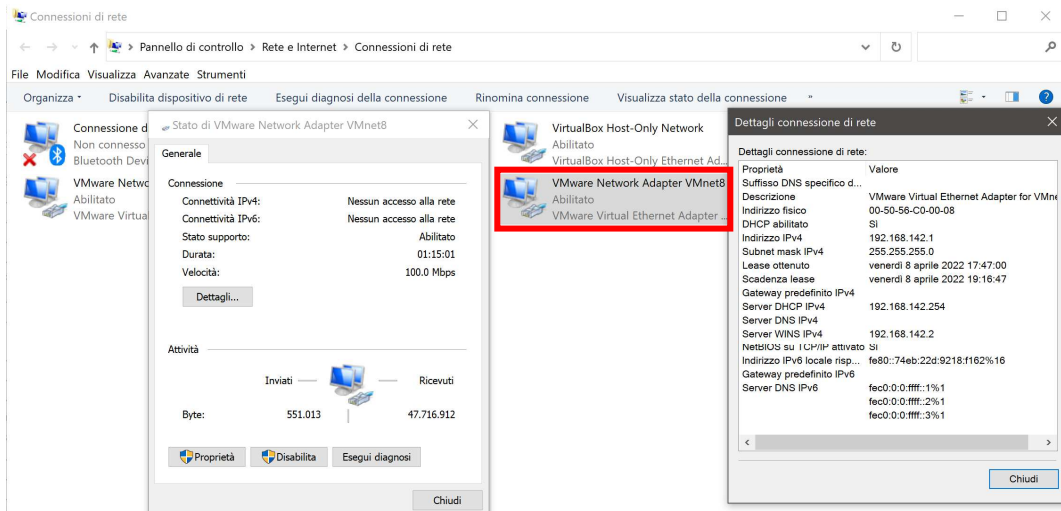


Figura 4-s: Scheda di rete virtuale alla base del nodo NAT.



## 5. LA RETE "COOPERLAT"

L' Azienda Cooperativa Cooperlat fonda le sue radici a Jesi, nel cuore delle Marche, ma nel corso degli anni ha ampliato il proprio raggio di azione con depositi logistici che vanno dal Nord al Sud Italia e stabilimenti di produzione e trasformazione del latte quasi tutti presenti in suolo Marchigiano e Piemontese, posizionandosi tra i primi gruppi del settore lattiero-caseario in Italia.

Osservando l'azienda da un punto di vista tecnico e di networking è da subito palese come la struttura sia un chiaro esempio di network WAN a stella ovvero tutti i depositi su suolo italiano sono collegati mediante tecnologia MPLS alla sede di Jesi che funge da centro stella e dove quindi confluiscono le connessioni.

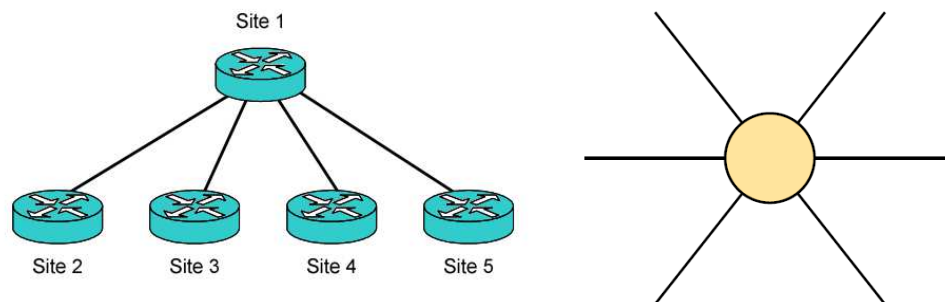


Figura 5-a: Rappresentazione grafica del concetto di rete a Stella.

La tecnologia MPLS (*Multiprotocol Label Switching*) è una tecnologia di data *Forwarding* di *layer 3* (*layer* di rete) ad alte prestazioni per l'instradamento

di pacchetti IP attraverso una rete pubblica e condivisa ed è gestita interamente da un provider.

L'idea di base consiste nell'associare a ciascun pacchetto *Layer 3* un breve identificativo di lunghezza fissa (*Label*) che gli apparati di networking (Routers) utilizzano per effettuare un instradamento veloce basato sulla commutazione d'etichetta.

Con l'MPLS i dati vengono indirizzati in un percorso ben preciso e prestabilito su rete pubblica e riconosciuti grazie alle etichette apposte al pacchetto IP.

I principali vantaggi della tecnologia MPLS sono:

- *Sicurezza*: Il protocollo MPLS operando in modalità Label Switching offre le massime garanzie di sicurezza senza alcun ulteriore servizio aggiuntivo (Firewall, IDS, etc.). I livelli di sicurezza sono paragonabili a reti di *Layers 2*;
- *Scalabilità*: Il dimensionamento delle VPN MPLS è estremamente flessibile e l'aggiunta di una nuova sede non comporta alcuna variazione e maggior costi per HW e SW presso le altre sedi in quanto la configurazione della VPN è implementata sulle apparecchiature del Provider garantendo una scalabilità illimitata;
- *Personalizzazione e flessibilità*: La tecnologia MPLS supporta un'ampia gamma di servizi personalizzabili con diversi livelli di priorità, classe e garanzia in modo da poter implementare e

differenziare i diversi utilizzi applicativi come ad esempio il VOIP (*Voice Over IP*).

In questo lavoro di tesi ci si è però incentrati su quello che riguarda il *Layer* 2 dello stack protocollare TCP/IP ovvero il *layer* Data Link, escludendo i collegamenti MPLS con le sedi periferiche ed incentrando l'analisi sulla rete LAN interna allo stabilimento di Jesi.

## **5.1 LAN Cooperlat**

Di una LAN è importante prendere in analisi anche la disposizione dei fabbricati in cui si sviluppa l'azienda.

Lo stabilimento Cooperlat di Jesi include diversi edifici tra cui uno dedicato agli uffici di amministrazione e reparto commerciale, un secondo dedicato alla programmazione di produzione, laboratorio chimico e controllo di produzione, e infine un terzo dove è racchiuso tutto il polo produttivo comprendente magazzino automatico e impianti di cogenerazione energetica.

Tutti gli edifici e l'eterogeneità di applicazioni che ne deriva sono interconnessi tra loro da un'unica rete LAN che comprende un totale di 34 *switch* Cisco dislocati in vari punti degli edifici.

La convivenza di una tale varietà di sistemi e applicazioni è possibile mediante l'utilizzo di Virtual LAN (VLAN), che garantiscono sicurezza ed e migliori prestazioni.

### **5.1.1 Le Virtual LAN**

La LAN Cooperlat conta ben 16 VLAN, ognuna dedicata ad una specifica classe di servizio.

Come definito nel capitolo 3.5 le VLAN sono nate proprio con l'intento di minimizzare il dominio di broadcast di una rete LAN o, più semplicemente, per distinguere il traffico *Layer 2* delle principali applicazioni che coesistono in un ambiente LAN.

### **5.1.2 Lo *Spanning Tree Protocol***

Nella LAN Cooperlat viene adottato il protocollo proprietario Cisco Rapid-PvST+, anche se non nel pieno delle sue potenzialità.

La particolarità del protocollo PvST+ e della sua versione Rapid, è proprio quella di poter eleggere istanze STP diverse per ogni VLAN ma, nel contesto Cooperlat, tutte le VLAN fanno capo ad un unico *Root Bridge* che coincide con uno *switch* Cisco Catalyst 4507r. Come possiamo dedurre dalla configurazione estratta dal medesimo *switch*, tutte le VLAN fanno

riferimento al 4507r come *Root Bridge* in quanto la *Priority* è impostata al valore 4096 per l'intero intervallo di VLAN da 1 a 1005.

```
Spanning Tree mode rapid-pvst
Spanning Tree extend system-id
Spanning Tree vlan 1-1005 Priority 4096
```

Lo *switch* che funge da backup al *Root Bridge* è un Cisco Catalyst 9300 il quale ha *Priority* pari a 8192 per tutte la VLAN attive.

```
Spanning Tree mode rapid-pvst
Spanning Tree extend system-id
Spanning Tree vlan 1-4094 Priority 8192
```

I restanti *switch* possiedono *Priority* maggiori e pari al valore di default di 32768 in quanto sono di fatto *switch* con funzioni di distribuzione e accesso.

Per proteggersi da incauti collegamenti di *switch* o hub da parte di utenti o malintenzionati, tutte le porte in access sono configurate con *bpdu-guard*, in modo da transitare nello stato di *error disable* nell'eventualità che uno di questi dispositivi venga collegato ad una di tali porte.

## 5.2 Ricognizione della LAN

Possiamo già dedurre che gli *switch* Cisco Catalyst 4507r e 9300 sono due *switch* particolari e di importanza cruciale all'interno della LAN Cooperlat. Essi sono gli unici *switch Layer 3* presenti in topologia e svolgono perciò anche funzioni di routing tra VLAN. Tali *switch* sono legati da ridondanza virtuale grazie al protocollo FHRP (*First Hop Redundancy Protocol*) nella versione 3 VRRP (*Virtual Router Redundancy Protocol*), proprio per evitare il *single point of failure* e quindi rendere trasparente l'eventuale *failover* dello *switch* VRRP primario, che in questo caso corrisponde al Cisco Catalyst 4507r.

Tutti gli *host* presenti in LAN Cooperlat, per poter comunicare con gli *host* appartenenti ad altre VLAN oppure per raggiungere la WAN Internet, debbono per forza transitare attraverso il Default-Gateway che per tutte le VLAN, in condizioni normali, coincide con lo *switch* Cisco 4507r mentre, in assenza di esso, con il suo backup ovvero lo *switch* Cisco 9300. Questo è reso possibile dal protocollo VRRP il quale permette di raggruppare i due *switch* centro stella sotto un unico indirizzo IPv4 virtuale. Nella pratica ciò si traduce nel fatto che ogni *host* ha impostato come Default-Gateway un indirizzo IP virtuale che non corrisponde a nessuna interfaccia fisica ma che viene appositamente assegnato dal protocollo VRRP allo *switch* centro stella primario e, nel caso di guasto, al suo corrispondente backup. Il lavoro di questo protocollo è fondamentale per evitare il *single point of failure* menzionato precedentemente, ovvero la possibilità che tutta LAN possa

risultare “isolata” sia internamente che esternamente a causa del *fault* di un singolo *switch* con funzione di Default-Gateway. In assenza di Default-Gateway, un *host* può comunicare solo con *host* appartenenti alla sua stessa VLAN e di conseguenza non può nemmeno navigare in Internet o comunicare con *host* appartenenti a VLAN diverse.

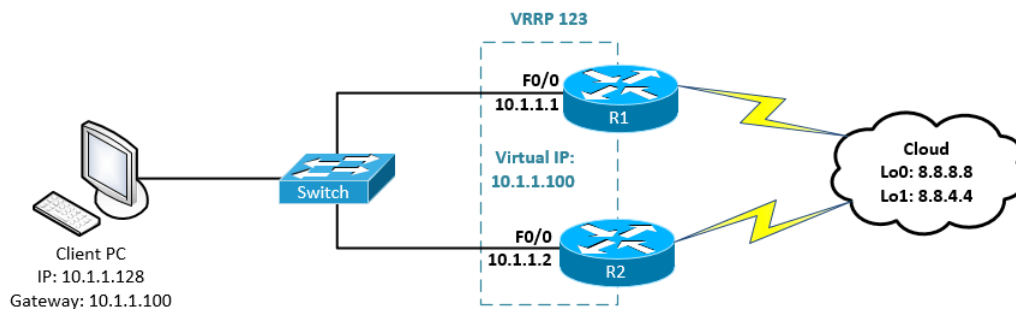


Figura 5-b: Esempio di applicazione di VRRP. L'host utilizza come indirizzo di default-gateway l'indirizzo IP virtuale gestito dal VRRP.

La ridondanza dei collegamenti fisici tra *switch* e lo sforzo per evitare un *single point of failure* sono senz'altro aspetti positivi e sinonimi di robustezza e resilienza della rete LAN Cooperlat.

Manca invece la diversificazione di un *Root Bridge* per ogni VLAN, che garantirebbe un miglior bilanciamento di carico sui link ma anche un alleggerimento del carico computazionale richiesto ai due *switch* di centro stella e di conseguenza anche un guadagno prestazionale, aspetto che verrà approfondito nella sezione relativa alla simulazione.

## 5.3 Metodo di lavoro

Come metodo di lavoro si è scelto di iniziare con l'estrazione e il salvataggio di tutte le configurazioni degli *switch* presenti in topologia così da avere una base di partenza per la riproduzione virtuale della LAN Cooperlat.

Per prima cosa è necessario collegarsi all'interfaccia di comando (CLI – *Command Line Interface*) di ogni singolo *switch* presente in topologia. Per farlo ci sono essenzialmente due metodi:

1. Ponendosi nelle immediate vicinanze di ogni *switch* e collegandosi con un cavo chiamato cavo console;
2. Mediante applicazione SSH/Telnet client conoscendo a priori l'indirizzo IP dello *switch* (nel nostro caso si è utilizzato il software opensource Putty).

Per praticità abbiamo scelto il secondo metodo che permette, da un singolo PC connesso alla rete LAN, di raggiungere tutti gli *switch* mediante i loro indirizzi IP di management.

Una volta collegati con successo ad uno *switch* ed autenticati con utente e password di amministratore possiamo seguire due modalità per copiare la configurazione dello *switch*:

1. Dopo aver eseguito un comando di show della running-config (ovvero il file contenente tutti i parametri di configurazione che lo



*switch* sta utilizzando al momento), copiare semplicemente la configurazione su un documento di testo e salvarlo;

2. Utilizzare il comando *copy running-config tftp*, che in questo caso invia su sever TFTP la running-config sottoforma di file.

Ovviamente si è optato per la seconda modalità perché più rapida e professionale.

A questo punto si è reso però indispensabile l'installazione del software open source Tftpd64<sup>6</sup> su PC.

In *Figura 5-c* è mostrata la finestra iniziale all'avvio del programma.

---

<sup>6</sup> Tftpd64 è un'applicazione gratuita che può svolgere compiti come server DHCP, TFTP, DNS, SNTP e Syslog, nonché come client TFTP. Tftpd64 non deve essere per forza installato ma basta lanciare il file eseguibile tftpd64.exe e si è già pronti e in ascolto per le richieste TFTP.

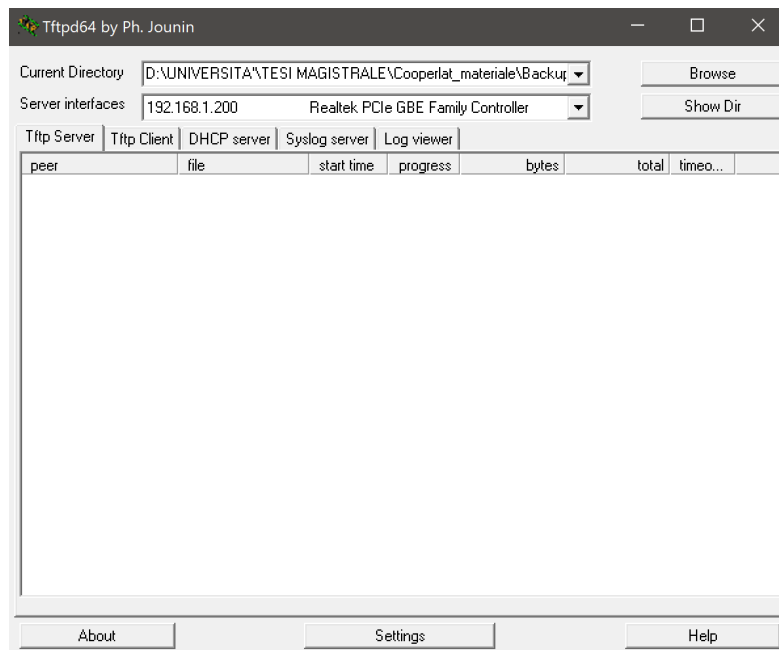


Figura 5-c: Finestra iniziale all'avvio del programma Tftpd64.

Grazie al software Tftpd64 abbiamo potuto creare un server TFTP al quale gli *switch* potessero inviare le proprie *running-configuration* semplicemente mediante il comando TFTP che vediamo sotto.

**copy running-config tftp:**

Address or name of remote *host* [ ]? **xxx.xxx.xxx.xxx**

Destination filename [ce\_2-config]? **backup\_cfg\_for\_my\_switch**

!!

1030 bytes copied in 2.489 secs (395 bytes/sec)

Analizzando il comando *copy running-config tftp*: vediamo che sono richiesti due parametri:

1. L'indirizzo IP del server TFTP (o nome macchina);

2. Il nome che vogliamo assegnare al file di backup contenete la config dello *switch* (tra parentesi quadre il nome che il sistema attribuisce in automatico).

Una volta confermati i due parametri richiesti, il client TFTP dello *switch* invia il file di configurazione attualmente in esecuzione sullo stesso al server TFTP specificato (un semplice PC aziendale in questo caso). A questo punto ritroveremo una copia esatta del file di configurazione nella cartella specificata dal campo *Current Directory* nell'applicazione Tftpd64.

Compresi i passaggi da eseguire per poter estrarre la configurazione di ogni singolo *switch*, si è pensato di realizzare una procedura automatizzata che fosse poi riutilizzabile da chiunque in azienda per effettuare periodicamente e rapidamente un backup completo degli *switch*.

A tal proposito si è implementata una routine ad hoc, o più propriamente un flusso, mediante il software Microsoft PowerAutomate<sup>7</sup>.

PowerAutomate è uno strumento molto potente che permette di automatizzare procedure ripetitivi, velocizzando di conseguenza la loro esecuzione. È possibile collegare diverse azioni l'una con l'altra, elaborare

---

<sup>7</sup> PowerAutomate è un servizio che consente di creare flussi di lavoro automatizzati tra le app e i servizi preferiti per sincronizzare file, ricevere notifiche, raccogliere dati e molto altro ancora.

automaticamente l'output ottenuto e utilizzare tali informazioni per eseguire altre operazioni. È inoltre possibile prevedere cicli automatizzati e impostare istanze *if...then...else* per stabilire che cosa PowerAutomate deve fare a seconda dell'output ricevuto. Il software è dotato anche di uno strumento per il debug che, come i più comuni ambienti di programmazione, permette di trovare eventuali errori nel codice e di eseguire passo passo i vari step di codice.

Di seguito possiamo osservare la finestra del programma alla sua apertura. Qui sono racchiusi tutti i flussi creati ed è possibile crearne di nuovi, eseguirli, eliminarli, duplicarli oppure di modificarli.

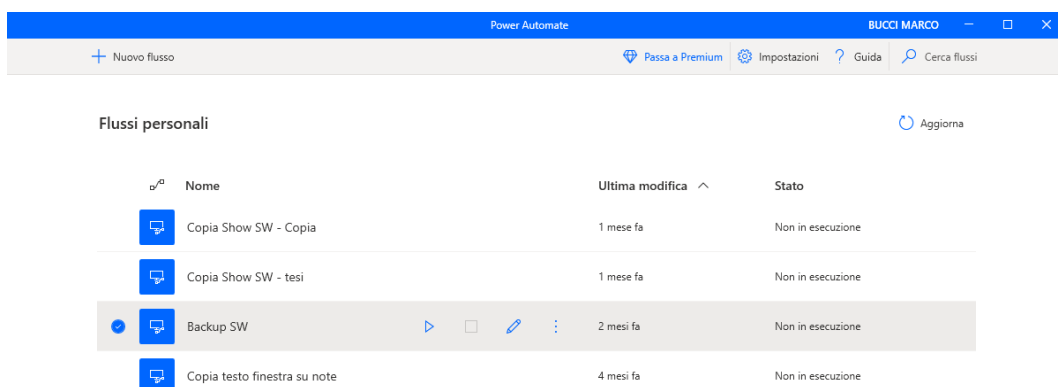


Figura 5-d: Dashboard iniziale del programma PowerAutomate.

Entrando in modifica all'interno di un flusso, come in *Figura 5-e*, sulla sinistra si hanno a disposizione moltissime azioni predefinite e raggruppate per categoria di utilizzo. Ogni azione corrisponde ad un blocco di codice e concatenando opportuni blocchi si ottiene un flusso che automatizza ciò che avrebbe richiesto notevole tempo, ma che soprattutto è ripetibile in futuro

ogni qual volta l'azienda richiede di effettuare un completo backup degli apparati di rete.

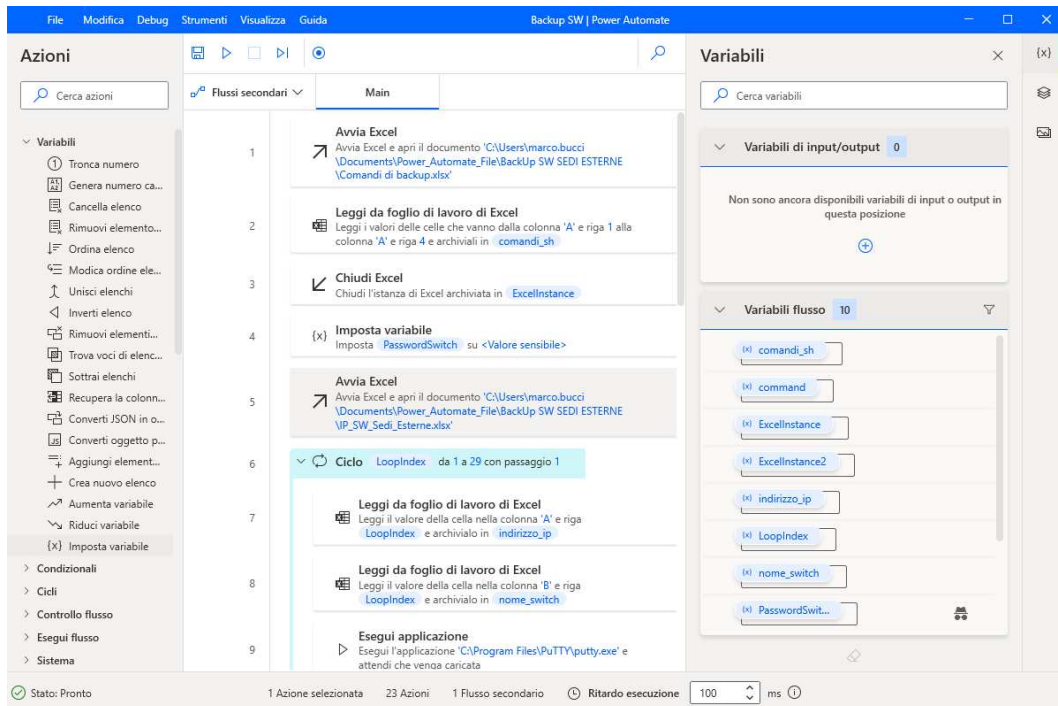


Figura 5-e: Finestra di modifica e debug di un flusso PowerAutomate.

Sempre in *Figura 5-e* vediamo una parte del flusso implementato per il nostro scopo. Si è scelto di impiegare due fogli Excel come database per racchiudere tutti gli indirizzi IP degli *switch* a cui ci si desidera collegare e un secondo foglio di calcolo, che racchiude i comandi da inviare agli stessi una volta aperta la connessione.

La routine comincia con l'apertura dei due file Excel, la relativa estrazione dei dati e caricamento in variabili di ambiente (visibili nella colonna di destra della dashboard). Dopo di che si apre un ciclo *for*, con contatore che incrementa di una unità ad ogni ciclo fino a raggiungere il numero totale di

*switch*. Per ogni ciclo *for*, PowerAutomate apre un'istanza Putty, compila in autonomia il campo IP *address* e avvia il collegamento in SSH<sup>8</sup> allo *switch*.

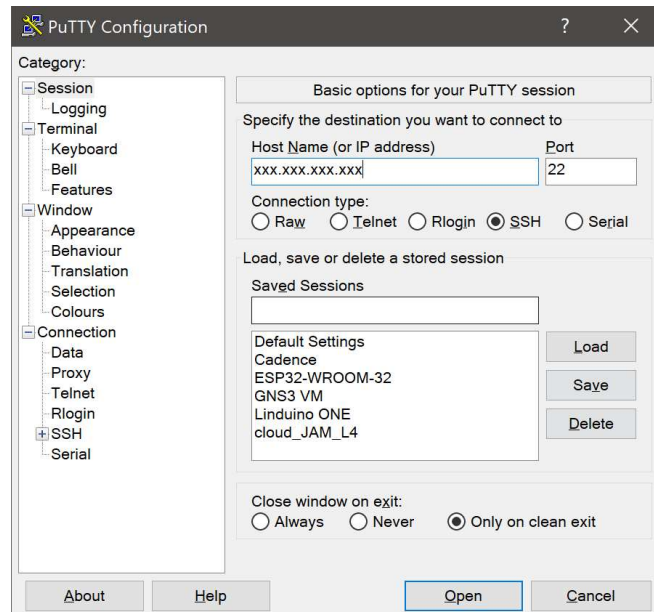


Figura 5-f: Interfaccia Putty, dove al campo IP address viene inserito l'indirizzo dello switch.

Una volta stabilita la connessione si passa all'invio dei comandi, prima le credenziali di autenticazione (con password che PowerAutomate permette di criptare in modo da evitare che nel codice possa restare in chiaro), poi al comando *copy running-config tftp:* verso il server TFTP precedentemente installato su PC aziendale.

---

<sup>8</sup> SSH (*Secure Shell*) è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro *host* di una rete informatica. È il protocollo che ha sostituito l'analogo, ma insicuro, Telnet.

Conclusa l'esecuzione del flusso possiamo comodamente ritrovare tutte e 34 le configurazioni di cui necessitiamo, nella cartella indicata dal server TFTP come *current directory*, dove sono stati immagazzinati i file che mano mano venivano inviati dagli *switch*.

Un ulteriore flusso PowerAutomate è stato poi implementato per estrarre dagli *switch* alcune importanti informazioni non presenti nei files di configurazione. In particolare si tratta di alcuni comandi di *show*, visibili nel codice riportato di seguito, che si sono resi molto utili per capire gli attuali collegamenti tra *switch* e aggiornare alcune piantine della topologia LAN Cooperlat che nel tempo non erano state revisionate.

```
sh vlan brief
sh cdp neighbor
sh EtherChannel summary
sh int status
sh ip interface brief
sh ip route
sh Spanning Tree summary
sh Spanning Tree
```

Uno dei comandi di show che si è rivelato molto utile è lo *show cdp neighbor*, il quale sfrutta il protocollo proprietario Cisco CDP per rivelare molte informazioni riguardanti i dispositivi Cisco direttamente collegati allo *switch*; ad esempio il numero di porta alla quale sono collegati ma anche l'indirizzo IP e la tipologia di link utilizzata (se utilizza un modulo SFP, se ethernet

ecc.). L'alternativa proposta da IEEE al protocollo proprietario Cisco CDP è il protocollo LLDP ma in questo caso non è stato utilizzato visto che la topologia Cooperlat comprende solo dispositivi Cisco.

## 5.4 Simulazione della rete Cooperlat in GNS3

Conclusa la fase di raccolta informazioni e collezionate tutte le configurazioni degli *switch* presenti in LAN Cooperlat, si è passati all'ambiente virtuale GNS3.

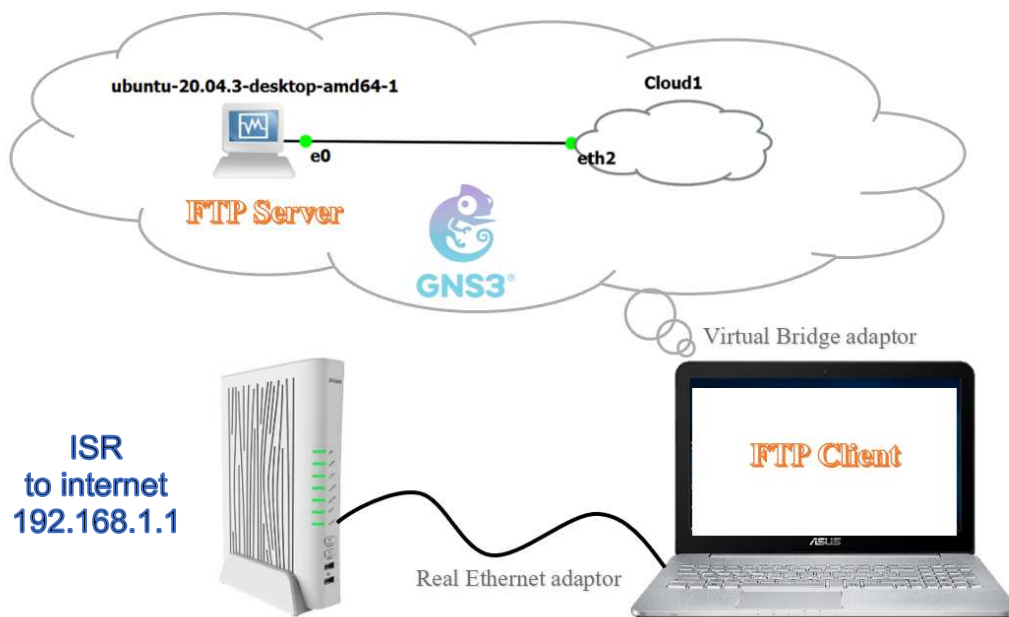
Per prima cosa ci si è posti il quesito di come trasferire i file di configurazione degli *switch* da una directory locale in ambiente Windows, all'interno della topologia virtuale. Appurato che tra ambiente *host* Windows e VM VirtualBox un'azione di *copy/paste* risulta impossibile, si optato per sfruttare la connettività offerta dai nodi Cloud e NAT congiuntamente all'utilizzo del protocollo FTP<sup>9</sup> come vedremo a breve.

---

<sup>9</sup> File Transfer Protocol (FTP) (protocollo di trasferimento file), è un protocollo di livello applicativo per la trasmissione di dati tra *host* basato su TCP e con architettura di tipo client-server.



Lo scopo è quindi installare un server FTP in ambiente virtuale GNS3 che sia raggiungibile da rete locale domestica e al quale trasferire tutte le configurazioni degli *switch* Cooperlat precedentemente ottenute. In *Figura 5-g* è mostrata l'architettura così ideata.

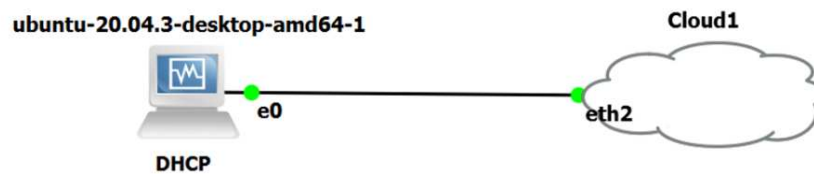


*Figura 5-g: Architettura software ideata per l'installazione del server FTP in GNS3 e passaggio dei file di configurazione switch mediante client FTP.*

---

Il protocollo usa connessioni TCP distinte per trasferire i dati e per controllare i trasferimenti e richiede autenticazione del client tramite nome utente e password, sebbene il server possa essere configurato per connessioni anonime con credenziali fittizie.

Il primo step è stato importare nella dashboard GNS3 una VM Desktop Ubuntu e un nodo Cloud (in esecuzione su GNS3 VM) come mostrato in *Figura 5-h*.



*Figura 5-h: Primo step per l'installazione di un server*

Collegati tra loro VM e nodo Cloud (sempre utilizzando l'interfaccia *eth2* ovvero la scheda di rete virtuale con funzione di *Bridge*), l'interfaccia di rete del virtual desktop Ubuntu, riceve la configurazione di rete dal server DHCP (in questo caso l'ISR domestico) e sostanzialmente risulta già pronto alla navigazione in internet.

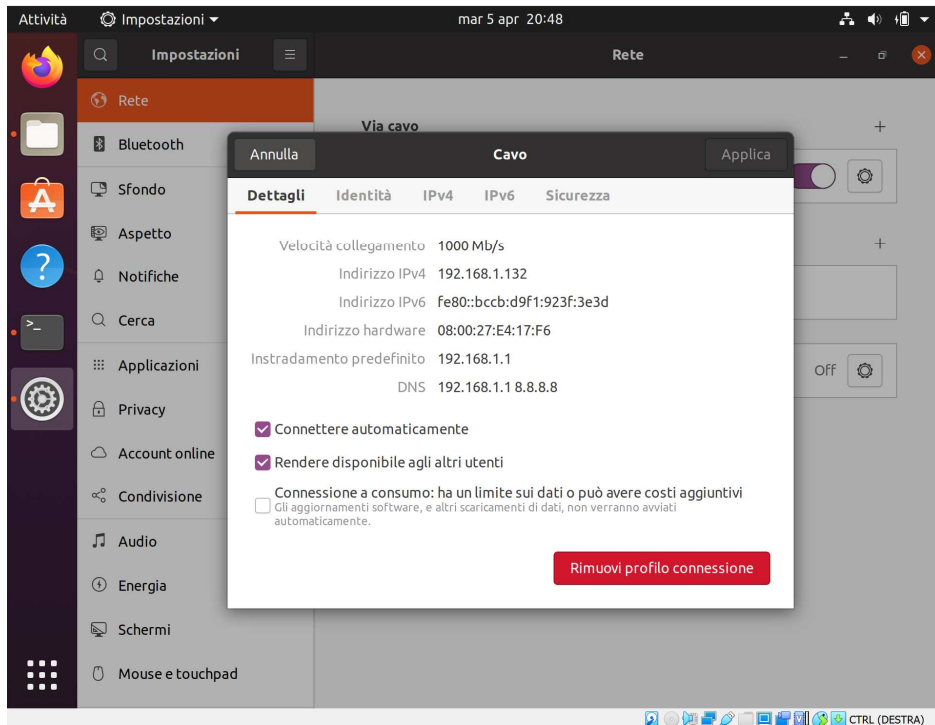


Figura 5-i: Parametri di rete ereditati dal server DHCP con connessione mediante nodo Cloud.

Si procede quindi all'installazione del servizio di server FTP, scaricabile direttamente dai repositories Linux con i seguenti comandi impartiti direttamente dal *Command Line Terminal* per adibire la virtual machine Ubuntu a server FTP.

```

# Innanzitutto si inizia con un update generico dell'OS Ubuntu.
user@user-VirtualBox:~$ sudo apt update
# Ora si installa il servizio server FTP.
user@user-VirtualBox:~$ sudo apt install vsftpd
# Terminata l'installazione si avvia il servizio
user@user-VirtualBox:~$ sudo systemctl start vsftpd
# e si abilita lo stesso.
user@user-VirtualBox:~$ sudo systemctl enable vsftpd
# Per precauzione si effettua una copia del file di configurazione del
servizio.
user@user-VirtualBox:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
# E' possibile creare nuovi utenti che si possano poi collegare al server
FTP.
user@user-VirtualBox:~$ sudo useradd -m testuser
user@user-VirtualBox:~$ sudo passwd testuser
# Per sicurezza si modificano le regole firewall del OS aprendo le porte
20 e 21 TCP.
user@user-VirtualBox:~$ sudo ufw allow 20/tcp
user@user-VirtualBox:~$ sudo ufw allow 21/tcp

```

A questo punto l'installazione è completata e per verificare la corretta installazione è possibile collegarsi al server FTP in *localhost* come mostrato nella seguente *Figura 5-j*.

```

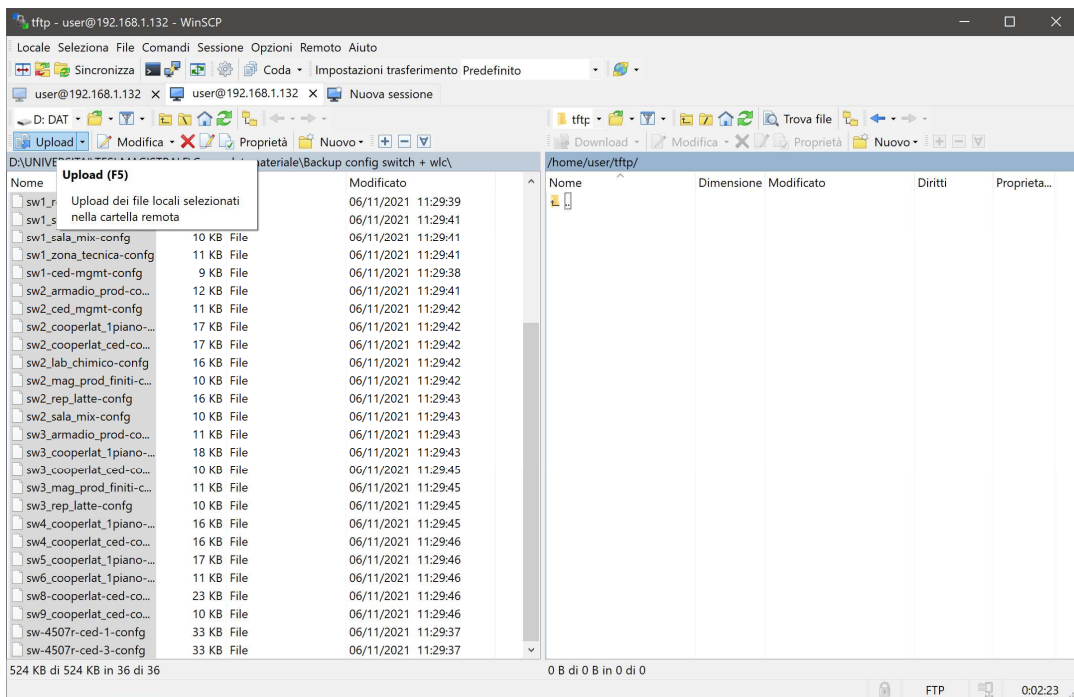
user@user-VirtualBox:~$ sudo ftp user-VirtualBox
Connected to user-VirtualBox.
220 (vsFTPd 3.0.3)
Name (user-VirtualBox:user): testuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

*Figura 5-j: Collegamento in localhost al server FTP.*

Ora che il server FTP è attivo e funzionante si ha la necessità di un client FTP lato Notebook Windows affinché si possa stabilire una connessione al server FTP su porta TCP 21, e per questo si è optato per il software opensource WinSCP.

Come osservabile dalla *Figura 5-l* la sua interfaccia è molto intuitiva: sulla sinistra è possibile sfogliare le directory locali mentre sulla destra è possibile navigare le directory del server FTP al quale ci si è collegati ed autenticati e a cui inviare i file di configurazione degli *switch*.



*Figura 5-k: Interfaccia WinSCP al collegamento con il server FTP in esecuzione su VM Ubuntu.*

Al momento però dell'effettivo trasferimento file, con connessione mediante nodo Cloud, da WinSCP si nota estrema lentezza e a volte la connessione stessa tra client e server FTP fallisce per time-out raggiunto e perciò si è

scelto di cambiare metodo di interconnessione, optando per l'utilizzo del nodo NAT.

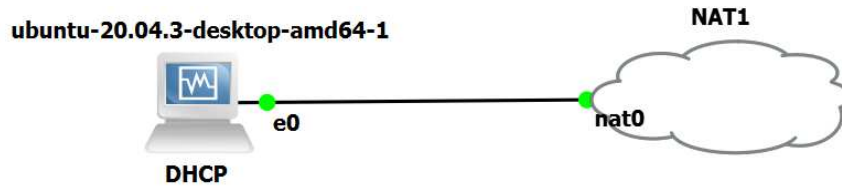


Figura 5-l: Connessione al nodo NAT.

Con il NAT il client Ubuntu eredita delle impostazioni di rete diverse come visibili in *Figura 5-m* e da subito la connessione risulta più stabile.

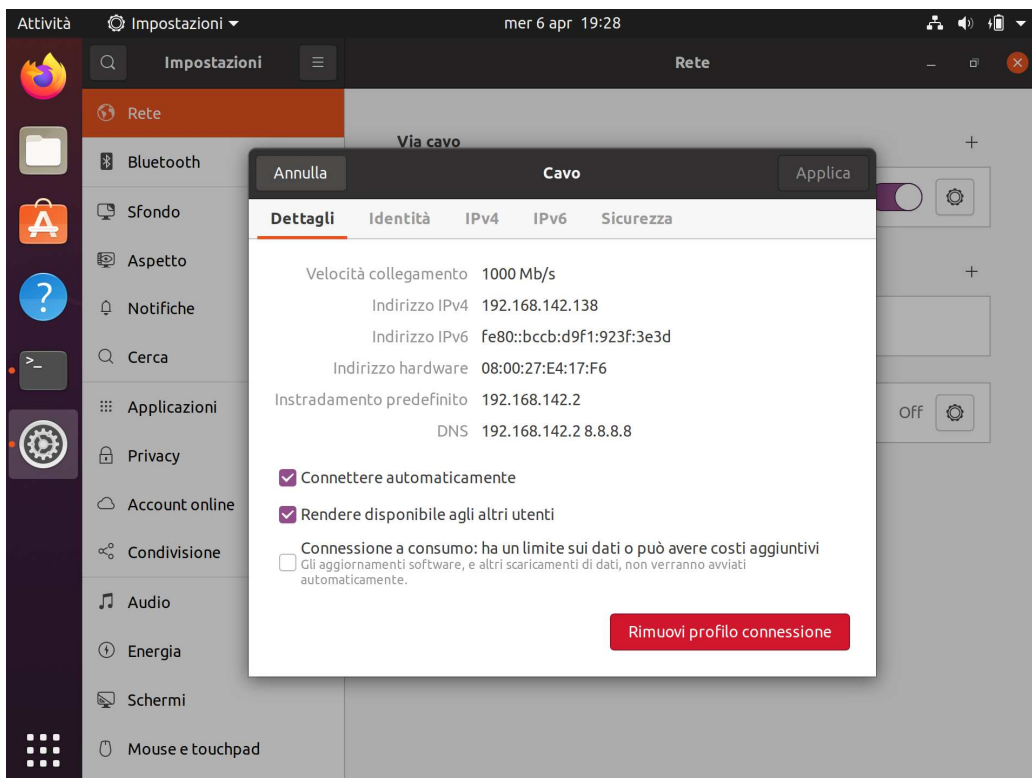
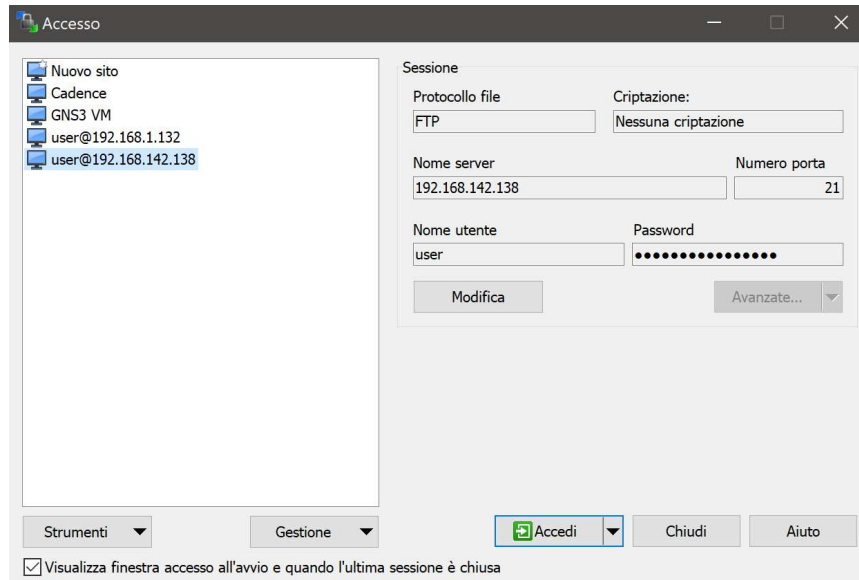


Figura 5-m: Parametri di rete del client Ubuntu connesso tramite nodo NAT.

Si passa quindi alla creazione di una nuova connessione WinSCP sempre con protocollo FTP su porta TCP 21, ma con indirizzo IP differente per quello che è il server FTP di destinazione.



*Figura 5-n: Configurazione delle connessioni WinSCP.*

Una volta stabilita la connessione dalla shell WinSCP, ci si porta a sinistra nella cartella locale contenete tutti i file di configurazione, e a destra nella directory di default `/srv/ftp` del server FTP Ubuntu, e selezionando tutti i file è possibile effettuare l'upload contemporaneamente in pochi istanti.

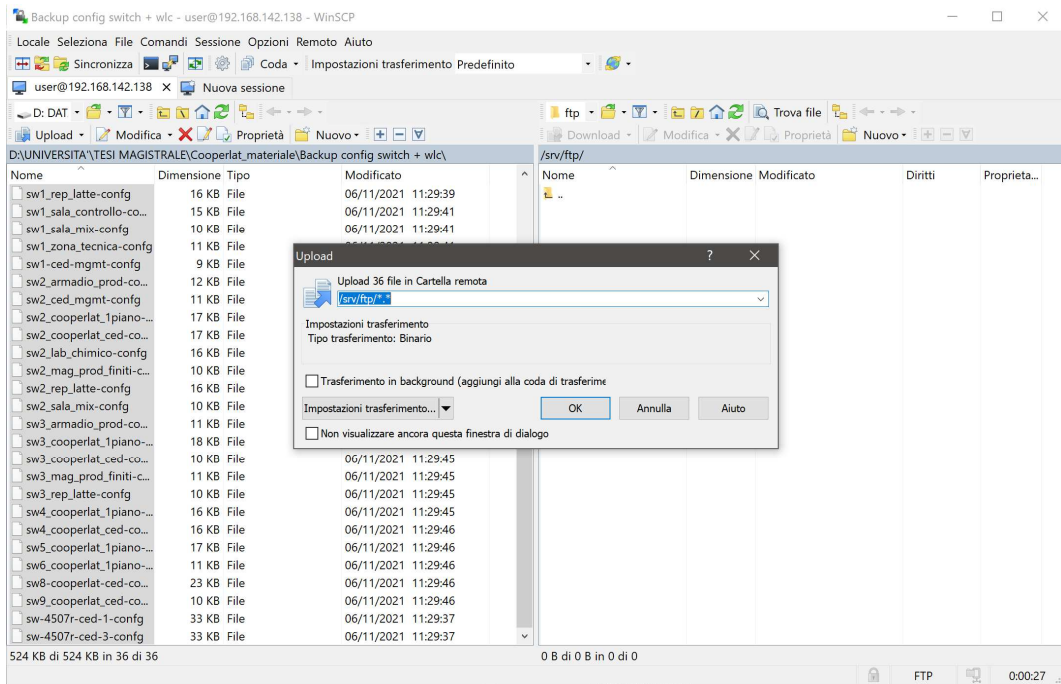


Figura 5-o: Upload dei file su server FTP.

Ora dal virtual server Ubuntu è possibile ritrovare tutti i file nella cartella /srv/ftp specificata e come apprezzabile in Figura 5-p.

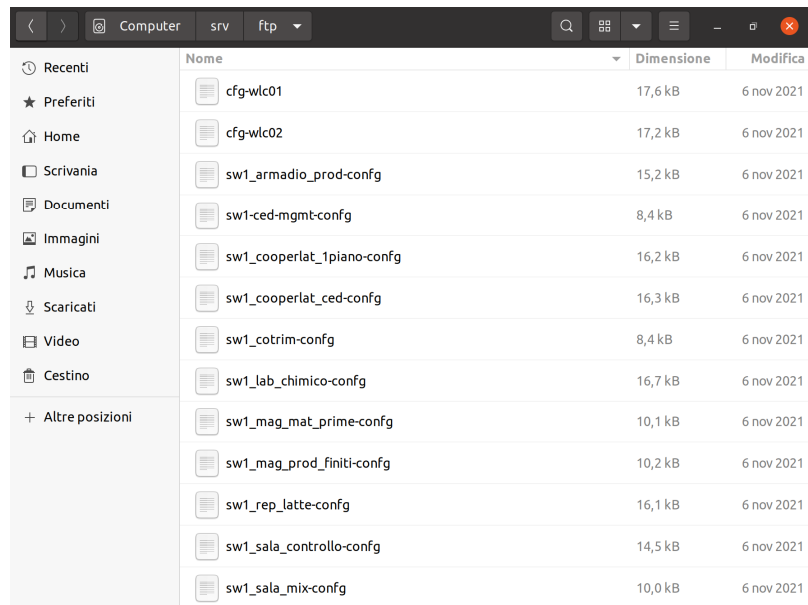


Figura 5-p: File caricati nella directory /srv/ftp.



Si passa ora al secondo step che consiste nell'import delle configurazioni ad opera degli *switch* che vengono aggiunti alla topologia.

Si procede quindi alla rimozione del nodo NAT e all'importazione di uno *switch* IOSvL2.

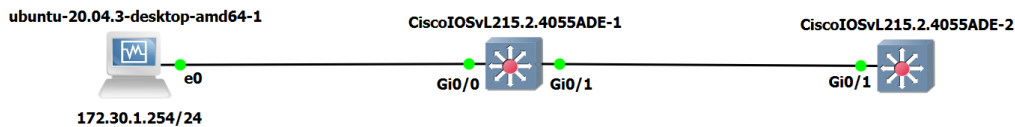


Figura 5-q: Rimozione del nodo NAT e aggiunta di switch IOSvL2.

Il server FTP, avendo rimosso il nodo NAT, non possiede più un indirizzo IP quindi è necessario attribuirne uno manualmente e, conoscendo già l'indirizzamento della sottorete di management degli *switch* Cooperlat, si procede con l'assegnare un indirizzo appartenente alla stessa e in particolare il 172.30.1.254/24. In questo modo il server sarà sempre raggiungibile dagli *switch*.

The screenshot shows the 'Cavo' network configuration window in Ubuntu. The 'IPV4' tab is selected. Under 'Metodo IPV4', the 'Manuale' option is selected. Under 'Indirizzi', the following values are entered:

Indirizzo	Netmask	Gateway
172.30.1.254	255.255.255.0	172.30.1.1

Figura 5-r: Configurazione manuale dei parametri di rete del server Ubuntu FTP.

Sugli *switch* che vengono importati in dashboard è sufficiente assegnare all'interfaccia VLAN 1 un indirizzo IP facente parte della stessa sottorete di management, così che sia possibile contattare il server FTP e scaricare da esso la configurazione.

```
configure terminal
interface vlan 1
ip address 172.30.1.250 255.255.255.0
no shutdown
```

Di seguito è possibile osservare la configurazione utilizzata per l'interfaccia VLAN 1 e i test di *ping* tra server FTP e *switch* e viceversa.

```
Switch#sh run int vlan 1
Building configuration...

Current configuration : 62 bytes
!
interface Vlan1
 ip address 172.30.1.250 255.255.255.0
end
```

```
Switch#ping 172.30.1.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.1.254, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/3/6 ms
```

```

user@user-VirtualBox:~$ ping 172.30.1.250
PING 172.30.1.250 (172.30.1.250) 56(84) bytes of data.
64 bytes from 172.30.1.250: icmp_seq=1 ttl=255 time=12.2 ms
64 bytes from 172.30.1.250: icmp_seq=2 ttl=255 time=8.14 ms
64 bytes from 172.30.1.250: icmp_seq=3 ttl=255 time=7.00 ms
64 bytes from 172.30.1.250: icmp_seq=4 ttl=255 time=7.57 ms
64 bytes from 172.30.1.250: icmp_seq=5 ttl=255 time=6.59 ms
^C
--- 172.30.1.250 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 6.592/8.306/12.234/2.032 ms

```

Figura 5-s: Configurazione interfaccia VLAN 1 e test di ping.

Giunti a questo punto è tutto pronto per l'import della configurazione.

Il comando FTP impartito dallo *switch* è il seguente:

```
copy ftp://user:****@172.30.1.254//srv/ftp/***** running-config
```

dove:

- **copy ftp:** indica il protocollo utilizzato per la connessione al server;
- **user:\*\*\*\*@** sono utente e password dell'utente FTP con il quale ci si vuole collegare;
- **172.30.1.254** è l'indirizzo IP del server;
- **//srv/ftp/\*\*\*\*\*** indica il file che si intende prelevare ed importare;
- **running-config** la destinazione del file di configurazione.

In *Figura 5-t* è possibile osservare un esempio di esecuzione del comando.

A questo punto, è bene sottolineare il fatto che l'immagine dello *switch* IOSvL2 utilizzato per rappresentare in GNS3 tutti gli *switch* della topologia Cooperlat, è l'immagine di uno *switch* virtuale che non esiste nella realtà,

quindi il passaggio dei file di *config* non può concludersi al 100%: un motivo tra i tanti il fatto che uno *switch* IOSvL2 supporta un massimo di 16 interfacce *GigabitEthernet*, mentre di norma uno *switch* reale ha 24 porte più eventuali moduli aggiuntivi.

Il risultato è visibile sempre in *Figura 5-t* dove, dopo aver impartito il comando di *copy*, lo *switch* restituisce errori di *Invalid Input*. Ad esempio non vengono importate le configurazioni delle interfacce motivo per cui la configurazione delle porte in trunk, dove sono presenti i link tra i vari *switch*, è stata effettuata ex-novo su ogni *switch* IOSvL2 dopo aver terminato l'import dei file di config.

```
Switch#Copy ftp://user:user@172.30.1.254//srv/ftp/sw1_armadio_prod-config running-config
Destination filename [running-config]?
Accessing ftp://*****:****@172.30.1.254//srv/ftp/sw1_armadio_prod-config...
Loading /srv/ftp/sw1_armadio_prod-config !
[OK - 15179/4096 bytes]

service call-home
^
% Invalid input detected at '^' marker.

platform punt-keepalive disable-kernel-core
^
% Invalid input detected at '^' marker.

Translating "all"

boot system switch all flash:packages.conf
^
% Invalid input detected at '^' marker.

switch 1 provision c92001-24p-4g
^
% Invalid input detected at '^' marker.
```

*Figura 5-t: Esempio di import della configurazione da server FTP.*

L'utilità dell'import delle configurazioni è comunque senz'altro di aiuto in quanto ha permesso di importare molte altre informazioni come i profili degli utenti amministratori, il nome dello *switch*, gli indirizzi IP assegnati alle interfacce fisiche o VLAN, gli SNMP group (fondamentali poi per il

monitoraggio SNMP), gli indirizzi dei server DNS, e molte altre configurazioni.

Terminata l'importazione si procede al cambio dell'*hostname* a livello GNS3 semplicemente intervenendo da GUI.

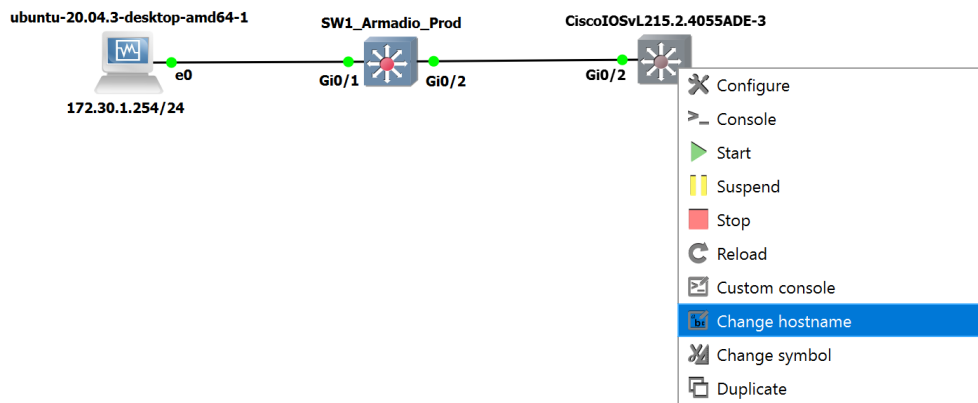


Figura 5-u: Cambio dell'*hostname* da GUI GNS3.

Si prosegue in questo modo finché non vengono riportati tutti gli *switch* in dashboard GNS3, e collegati tra loro rispettando i link reali fino ad ottenere l'intera topologia mostrata in *Figura 5-v*.

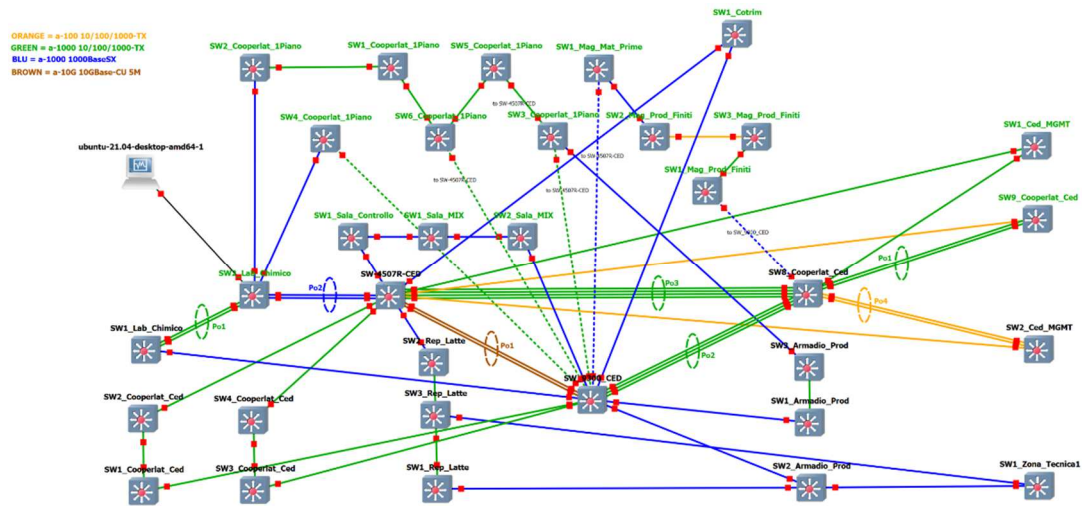


Figura 5-v: Topologia Cooperlat riprodotta in GNS3.

Come possibile osservare, si è differenziato per colore le tipologie di link che effettivamente vi sono tra gli *switch* in azienda (anche se rimane una distinzione prettamente grafica, in quanto in GNS3 non è possibile simulare connessioni in fibra ottica).

Ci sono poi dei link tratteggiati i quali indicano dei collegamenti che in realtà confluiscono al Catalyst 4507r, ma che per ragioni di limitatezza di porte allo *switch* IOSvL2 (totale 16 porte), è stato necessario “dirottare” all’altro *switch* di centro stella Catalyst 9300.

I link affasciati indicano i port channel tra gli *switch* e rimandiamo all’Appendice E per il codice utilizzato nella loro implementazione.

Per non saturare le risorse del Notebook, si è proceduto per step, accendendo solamente singole porzioni di topologia ma lasciando sempre

in esecuzione lo *switch* di centro stella 4507r, il server FTP e lo *switch* lo SW2\_Lab\_chimico che interconnette i due.

Nelle topologie LAN con molti *switch*, come in questo caso, il protocollo VTP<sup>10</sup> è utilizzato per creare rapidamente delle VLAN a partire dallo *switch* incaricato di essere il VTP server, e da esso si propagano a tutti gli *switch* tramite link di *trunk* senza la necessità di dover riconfigurare la stessa VLAN manualmente.

Nella topologia Cooperlat e quella riprodotta in GNS3, è proprio lo *switch* Catalyst 4507r il VTP Server, mentre tutti i restanti *switch* sono configurati come VTP Client; nel momento in cui un nuovo *switch* viene configurato come VTP client e possiede un link in *trunk* tramite il quale può contattare il VTP server, allora eredita tutte le stesse VLAN e relative caratteristiche.

Nelle appendici F e G è possibile visionare il codice IOS utilizzato per la configurazione rispettivamente di VTP server e VTP client sugli *switch*.

---

<sup>10</sup> il VLAN Trunking Protocol (acronimo VTP) è un protocollo di rete proprietario di Cisco utilizzato nelle reti dati per distribuire le informazioni relative alle VLAN (virtual local area network). Il protocollo MRP, che ha rimpiazzato il protocollo GVRP, è il suo analogo standardizzato da IEEE.

## 6. MONITORAGGIO DELLE PRESTAZIONI DELLA RETE

Conclusa la riproduzione della topologia LAN Cooperlat, si è passati all'implementazione di un servizio di monitoraggio della rete basato su protocollo SNMP<sup>11</sup>.

La presenza di un NMS (*Network Monitoring System*) è uno dei componenti chiave per il management di una rete. Esso monitora costantemente la rete e i suoi principali apparati come router, *switch*, firewall, server e macchine virtuali per rilevare eventuali guasti e problemi per 24 ore al giorno e 7 giorni su 7, coprendo quindi anche archi temporali in cui nessun membro del reparto IT è fisicamente presente in azienda.

Oggigiorno quasi tutti i sistemi di monitoraggio offrono l'importante servizio di reportistica e alerting via email, che permette di ricevere email al

---

<sup>11</sup> Il *Simple Network Management Protocol* (SNMP) è un protocollo di rete che appartiene alla suite di protocolli Internet TCP/IP. Opera al *layer 7* del modello OSI, utilizzando come protocollo del *layer* trasporto UDP sulle porte 161 e 162, consentendo di semplificare la configurazione, gestione e supervisione (monitoring) di apparati collegati in una rete (siano essi nodi interni di commutazione come i dispositivi di rete o nodi terminali di utenza), riguardo a tutti quegli aspetti che richiedono azioni di tipo amministrativo (management).



verificarsi di *fault* o comunque di ricevere report schedulati in formato PDF, CSV, XLSX ecc. sullo stato della rete.

Per il nostro progetto di tesi la scelta è ricaduta sul software opensource NagiosXI, basato su protocollo SNMP.

The logo for Nagios, featuring the word "Nagios" in a bold, black, sans-serif font. The letter 'N' is underlined with a horizontal line.

*Figura 6-a: Logo Nagios.*

Per prima cosa dalla console VirtualBox si crea una nuova virtual machine Ubuntu con release 20.04.03, in quanto l'ultima versione NagiosXI non supporta build successive di OS.

Successivamente la nuova VM viene importata nella dashboard GNS3 e connessa alla topologia. In ultimo viene connesso anche un nodo cloud per permettere la navigazione in Internet.

La topologia Cooperlat riprodotta in GNS3, raggiunge così il suo aspetto finale come mostrato in *Figura 6-b*.

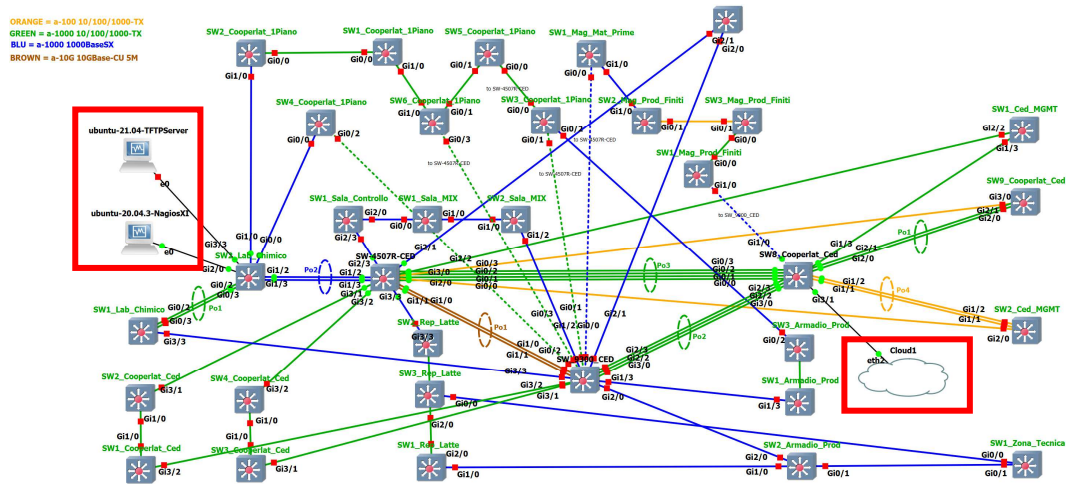


Figura 6-b: Topologia Cooperlat comprensiva di server NagiosXI e nodo Cloud.

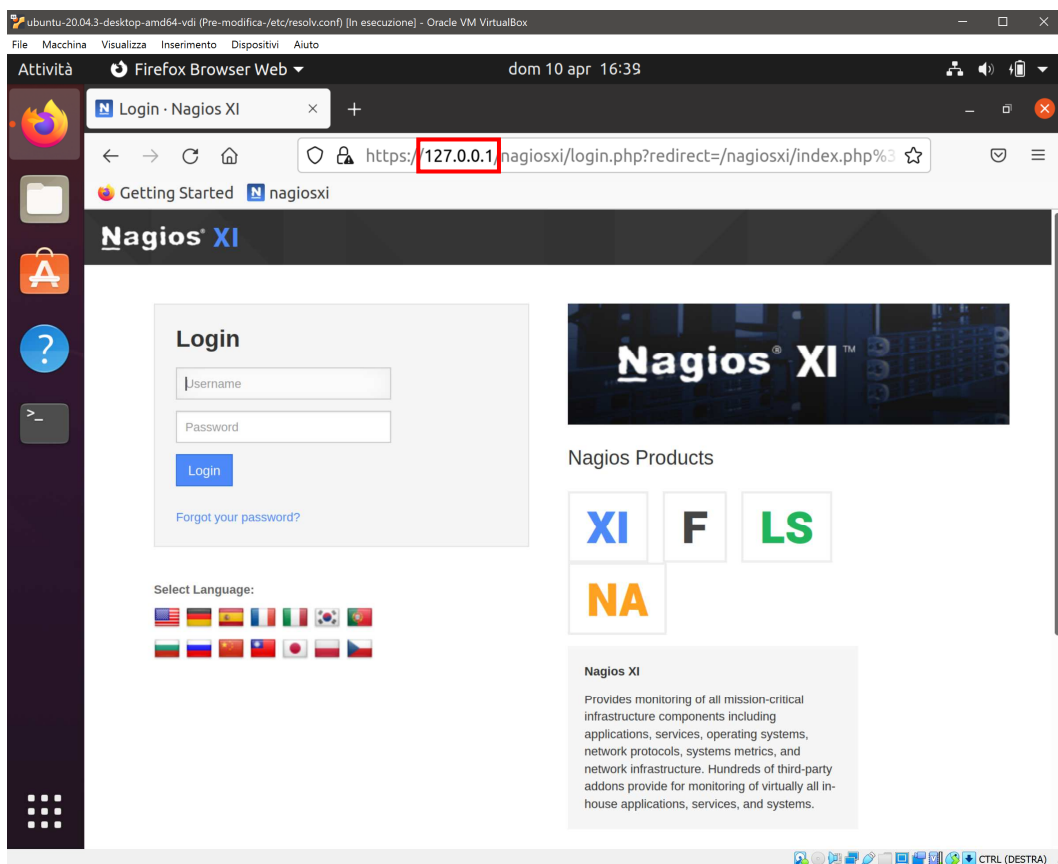
A questo punto assegnando manualmente al nuovo server NagiosXI un indirizzo IP della sottorete di management, è possibile verificare la possibilità di accedere ad internet (grazie alla regola di routing implementata lato ISR e descritta al capitolo 4.3.3) e di conseguenza iniziare l'installazione manuale del software NagiosXI.

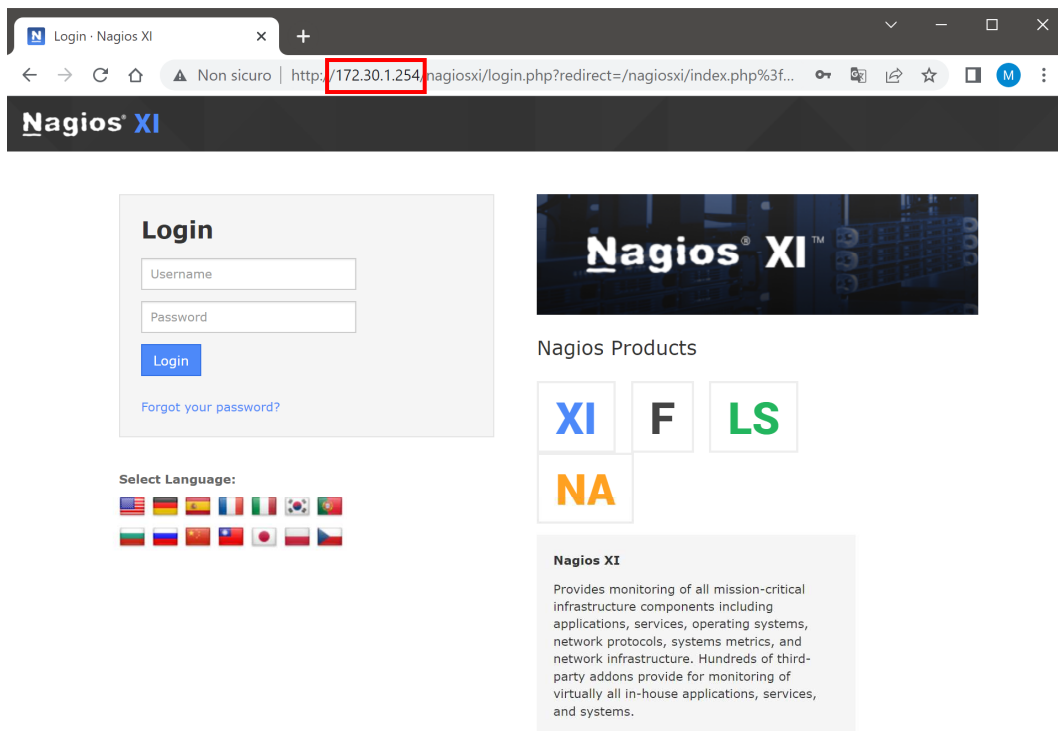
Cavo	
Dettagli	Identità
Velocità collegamento	1000 Mb/s
Indirizzo IPv4	172.30.1.254
Indirizzo IPv6	fe80::7c64:a607:50b0:403d
Indirizzo hardware	08:00:27:0A:12:AE
Instradamento predefinito	172.30.1.32
DNS	8.8.8.8 8.8.4.4
<input checked="" type="checkbox"/> Connettere automaticamente	
<input checked="" type="checkbox"/> Rendere disponibile agli altri utenti	
<input type="checkbox"/> Connessione a consumo: ha un limite sui dati o può avere costi aggiuntivi <small>Gli aggiornamenti software, e altri scaricamenti di dati, non verranno avviati automaticamente.</small>	
<input type="button" value="Rimuovi profilo connessione"/>	

Figura 6-c: Configurazione manuale dei parametri di rete del server NagiosXI.

Per tutti i passaggi dell'installazione è possibile fare riferimento alla documentazione ufficiale disponibile al sito [www.library.nagios.com](http://www.library.nagios.com).

Terminata l'installazione è possibile accedere all'applicativo NagiosXI tramite web interface sia in *localhost* da VM Ubuntu, oppure direttamente dal browser di un qualsiasi dispositivo collegato alla rete domestica e in grado di raggiungere l'indirizzo del server NagiosXI (172.30.1.254).





*Figura 6-d: Web Interface NagiosXI raggiunta in localhost (Mozilla Firefox sopra) e da notebook Windows all'indirizzo IP 172.30.1.254 (Google Chrome sotto).*

Effettuando il login con le credenziali amministrative scelte in fase di installazione, si presenta la dashboard visibile in *Figura 6-e*, e spostandoci nella sezione “*Configure*”, vengono proposti alcuni servizi per iniziare sin da subito la configurazione per il monitoraggio di dispositivi di rete.

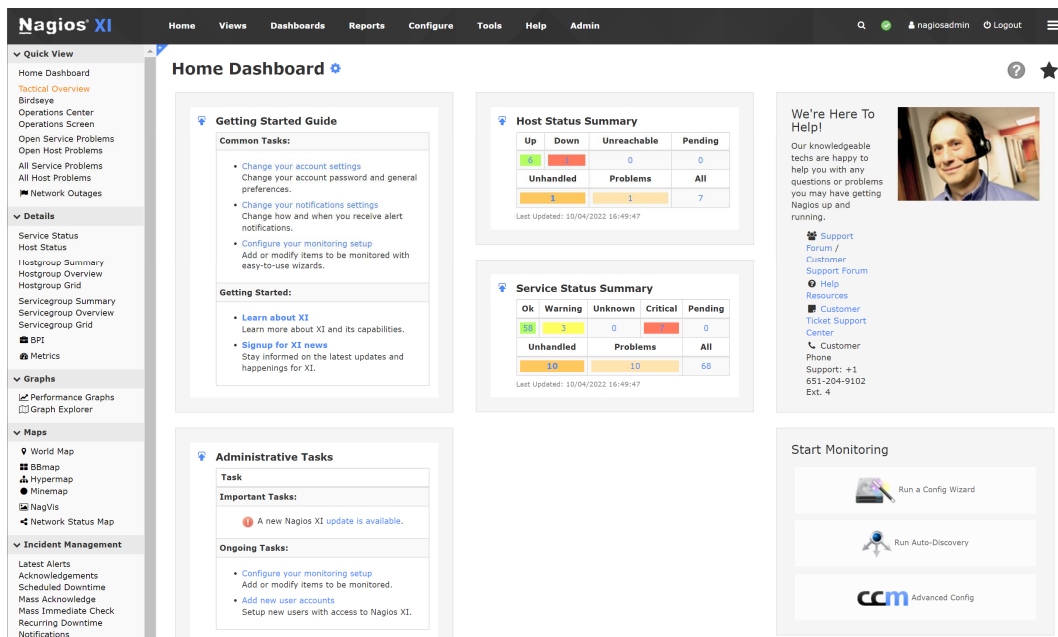


Figura 6-e: Dashboard iniziale NagiosXI.

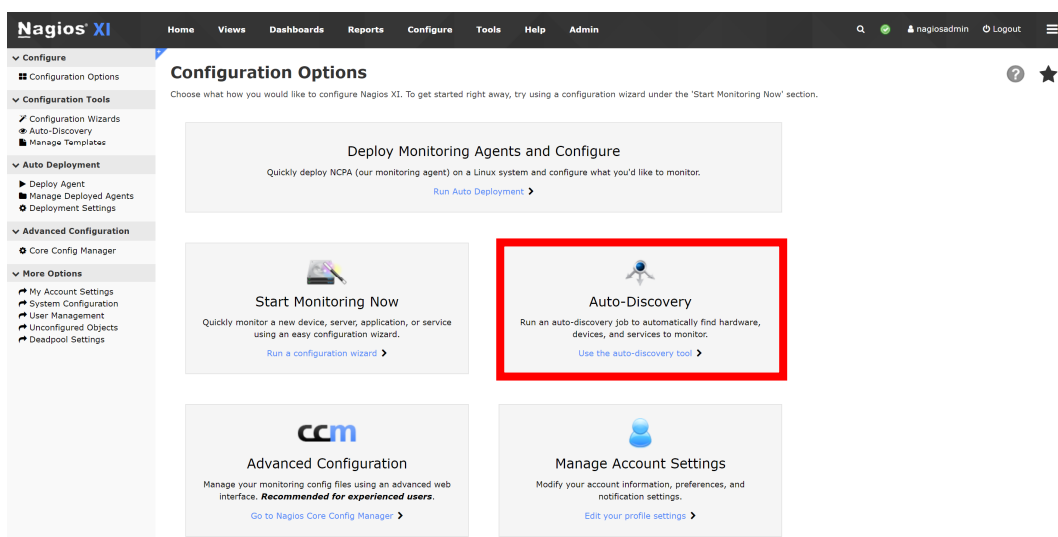


Figura 6-f: Pannello Configure di NagiosXI.

Per ottenere una prima configurazione, si è usufruito del comodo *tool Auto-Discovery*, un servizio che come visibile dalla *Figura 6-g*, richiede alcune semplici informazioni per poter configurare il monitoraggio ipoteticamente

di tutti i dispositivi accesi e connessi in rete. Il tool richiede informazioni basilari come la sottorete target oggetto del monitoraggio, eventuali indirizzi IP da escludere, e in ultimo la possibilità di programmare in maniera automatica lo scanning della sottorete in modo tale da scoprire eventuali nuovi dispositivi collegati e automaticamente porli sotto monitoraggio.

## New Auto-Discovery Job

Use this form to configure an auto-discovery job.

**Scan Target:**

Enter an network address and netmask to define the IP ranges to scan.

**Exclude IPs:**

An optional comma-separated list of IP addresses and/or network addresses to exclude from the scan.  
**Note:** The excluded addresses may be pinged, but they will not be scanned for open/available services via nmap.

**Schedule:** **Frequency:**

Specify the schedule you would like this job to be run.

[Show Advanced Options +](#)

*Figura 6-g: Configurazione dell'Auto-Discovery Job.*

Il Job di Auto-Discovery automatizza la configurazione del monitoraggio per quelli che sono parametri base, ovvero che possono essere monitorati in qualsiasi dispositivo di rete come ad esempio la risposta al ping ICMP e altri servizi definiti nel protocollo SNMP.

NagiosXI offre poi anche una serie di altri *wizard* semplici ed intuitivi per configurare il monitoraggio SNMP di specifici apparati, uno dei quali specifico per Router e *Switch*.

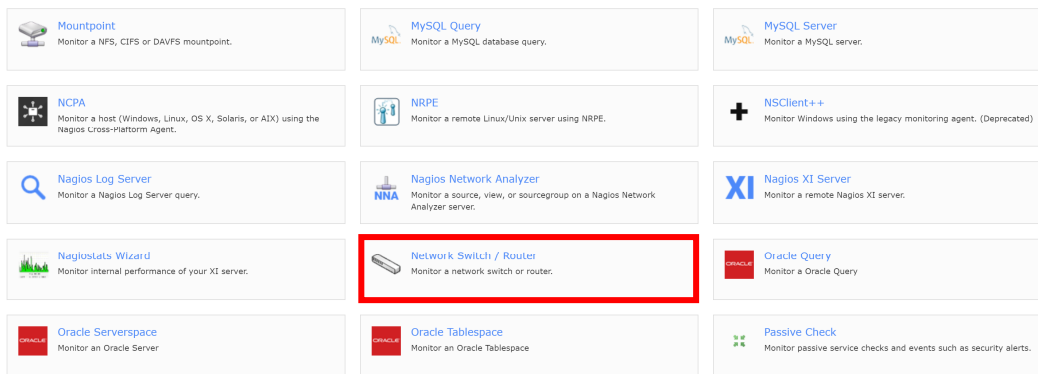


Figura 6-h: Wizard per la configurazione del monitoraggio.


Facendo doppio click sull'icona mostrata in *Figura 6-h*, si avvia una procedura guidata che, con poche e semplici informazioni, conduce alla configurazione di un monitoraggio dettagliato dell'apparato designato tramite protocollo SNMP. In questo caso però, è richiesta anche una configurazione lato *switch* affinché il protocollo SNMP possa comunicare opportunamente.

Nel dispositivo *switch* deve essere configurato il “*SNMP Community name*”, che permette all'applicativo NagiosXI di poter effettuare operazioni di *Traps SNMP*, ovvero interrogare il dispositivo sullo stato dei suoi componenti e delle sue performance.

Dallo *switch* la configurazione implementata è la seguente:

```
configure terminal
# Per abilitare il protocollo SNMP e l'accesso in READ-ONLY settare
la seguente community
set snmp-server community cooperlatro RO
```

A questo punto dalla procedura guidata del wizard è necessario inserire l'indirizzo IP dello *switch* e il nome della *community SNMP* scelto.



### Configuration Wizard: Network Switch / Router - Step 1

Router/Switch Information

**IP Address:**   
The IP address of the network device you'd like to monitor


**Port:**   
The port of the network device

SNMPv1  SNMPv2c  SNMPv3

**SNMP Community:**   
The SNMP community string required used to to query the network device

Figura 6-i: Configuration Wizard per switch e router.

Il *Wizard* poi continua dando la possibilità di selezionare nel dettaglio quali porte monitorare e anche impostare le soglie di *throughput* oltre le quali segnalare alert, oppure impostare dei *Timer* di autocheck.



### Configuration Wizard: Network Switch / Router - Step 3

Monitoring Settings

Define basic parameters that determine how the host and service(s) should be monitored.

**Under normal circumstances:**

Monitor the host and service(s) every  minutes.

**When a potential problem is first detected:**

Re-check the host and service(s) every  minutes up to  times before sending a notification.

Figura 6-j: Impostazione Timer di autocheck.



Essendo la versione NagiosXi installata una versione *trial*, ci sono delle limitazioni nell'utilizzo dell'applicazione, una delle quali è l'impossibilità di monitorare un numero maggiore di 7 *host* e un tetto sul numero massimo di servizi monitorabili pari a 100. Per cui il monitoraggio completo, è stato configurato solo per lo *switch* emulatore del Catalyst 4507r.

Di seguito è possibile osservare alcuni grafici e tabelle disponibili di default, che permettono di avere un colpo d'occhio su tutti i dispositivi monitorati e di risaltare eventuali problemi, così da poter intervenire nell'immediato.

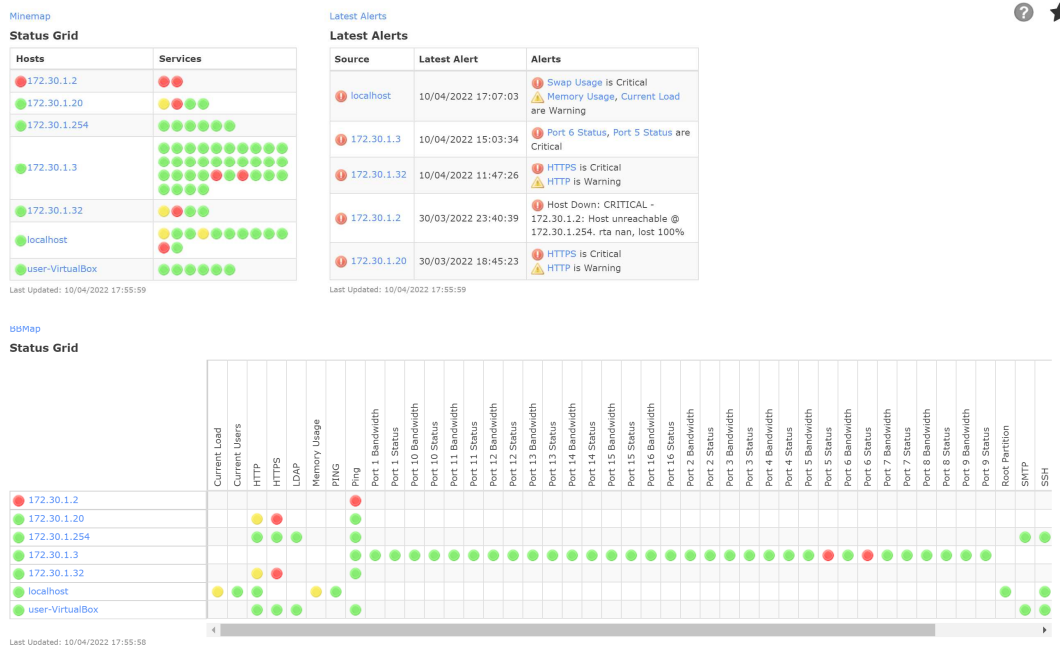


Figura 6-k: Grafici e tabelle per la visualizzazione degli apparati monitorati.

Dalla *Figura 6-k* è possibile osservare come per lo *switch* Catalyst 4507r, sono molti i servizi monitorati rispetto agli altri *host*, questo perché solo in tale caso è stato abilitato il monitoraggio SNMP per tutte le sue interfacce.

In ultimo è stato configurato il servizio di alerting via mail: è sufficiente dalla sezione “Admin” recarsi alla voce “Email Settings” e configurare i campi mostrati in *Figura 6-l*.

## Email Settings

Modify the settings used by your Nagios XI system for sending email alerts and informational messages.

The screenshot shows the 'Email Settings' configuration page in Nagios XI. At the top, there is a blue button labeled 'Send a Test Email'. Below it are two tabs: 'Outbound' (selected) and 'Inbound'. A light blue note box states: 'Note: Mail messages may fail to be delivered if your XI server does not have a valid DNS name. For more information, read Understanding Email Sending in Nagios XI.' Under the 'Outbound Mail Settings' section, the 'Send From' field contains 'Nagios XI <s1086542@studenti.univpm.it>'. The 'Send Method' section has two radio buttons: 'Sendmail' (selected) and 'SMTP'. A light blue note box below this section says: 'Note: On some systems, sendmail may not be configured to send emails outside of localhost. We highly recommend using SMTP configuration.' The 'Logging' section has a checked checkbox for 'Enable logging of mail sent with the internal mail component (PHPMailer) /usr/local/nagiosxi/tmp/phpmailer.log'. At the bottom, there are two buttons: 'Update Settings' (blue) and 'Cancel' (white).

*Figura 6-l: Configurazione Email Alerting.*

Il sistema, essendo connesso ad internet mediante nodo cloud, invia email mediante l'indirizzo di posta elettronica configurato in fase di registrazione (che è comunque possibile modificare in un secondo momento) per riportare eventuali problemi e criticità emersi in fase di monitoraggio.

Un esempio delle email ricevute sono quelle mostrate in *Figura 6-m*, ed in particolare è possibile osservare un esempio di messaggio *plain text* segnalante l'alert di warning per quello che riguarda l'interfaccia GigabitEthernet 0/1 dello *switch* con IP 172.30.1.20 in stato di *down*.



## 6.1 I vantaggi nella simulazione delle reti

Il software GNS3 è uno strumento fondamentale per un Network Engineer alle prime armi, perché permette senza ombra di dubbio di entrare in confidenza con quello che riguarda la configurazione di *switch*, router e firewall senza la necessità di disporre effettivamente di apparati fisici di elevato costo ma non solo: riproducendo in toto una topologia pre-esistente è possibile sfruttare l'ambiente virtuale per testare eventuali progetti atti alla modifica della stessa, in modo tale da evidenziare possibili anomalie e quindi riprogrammare l'intervento, affinché queste non si verifichino all'atto pratico.

Un approccio simulato consente di minimizzare l'impatto di eventuali modifiche apportate alla topologia o alla configurazione degli apparati, inoltre consente di testare nuovi applicativi come sistemi di monitoraggio quali NagiosXI, ed apprezzarne l'utilità e i benefici.

Per ragioni prettamente legate alle modeste risorse del Notebook dove è stato installato l'applicativo GNS3, non è stato possibile effettuare una simulazione della topologia totale però, anche se simulata dividendo per aree di interesse, è comunque risultato di grande aiuto per la comprensione di dinamiche che si verificano tra gli *switch* presenti in azienda.

## 6.2 Risultati del monitoraggio della rete simulata

La virtualizzazione soffre della scarsità di risorse hardware su PC, infatti nella simulazione della topologia Cooperlat si è inizialmente notata lentezza di risposta da parte soprattutto degli *switch* di centro stella Catalyst 4507r e 9300.

Grazie al monitoraggio offerto da NagiosXI, ed eseguendo alcuni comandi dedicati da CLI riportati di seguito, abbiamo notato che l'utilizzo delle risorse (in particolare CPU) degli *switch* di centro stella, era per la quasi totalità occupata per l'esecuzione dei servizi VRRP<sup>12</sup> e *Spanning Tree*.

Lo *switch* Catalyst 4507r si aggirava su una media all'82% di occupazione di CPU, mentre lo *switch* Catalyst 9300 all'incirca il 63%, quando uno *switch* reale in media non supera il 10% di utilizzazione.

---

<sup>12</sup> Il *Virtual Router Redundancy Protocol* (VRRP) elimina il *single point of failure* inerente al routing statico. Il VRRP è un protocollo di selezione che opera su un cluster di router (solitamente due) ed assegna dinamicamente un indirizzo IP virtuale all'interfaccia del router eletto come primario. Se l'interfaccia del router primario per qualsiasi ragione non dovesse più essere raggiungibile allora il protocollo in automatico assegnerà il ruolo di primario e l'indirizzo IP virtuale al router che precedentemente era secondario.

- CON VRRP attivo sulle interface VLAN

SW-4507R-CED#sh processes cpu sorted

CPU utilization for five seconds: 89%/0%; one minute: 88%; five minutes: 82%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	510621	33337	15316	66.30%	64.42%	60.52%	0	Spanning Tree
86	130836	3149	41548	17.03%	17.41%	16.07%	0	VRRP Main thread
70	11400	880	12954	1.43%	1.37%	1.31%	0	Per-Second Jobs
3	3330	364	9148	0.79%	1.20%	0.54%	0	Exec

SW\_9300\_CED#sh processes cpu sorted

CPU utilization for five seconds: 71%/0%; one minute: 67%; five minutes: 63%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	468893	70778	6624	61.77%	57.53%	53.92%	0	Spanning Tree
70	11354	872	13020	1.30%	1.43%	1.32%	0	Per-Second Jobs
87	6169	10761	573	1.14%	0.80%	0.72%	0	VRRS Main thread
86	6914	3039	2275	0.89%	0.91%	0.81%	0	VRRP Main thread

I due *switch* hanno un ruolo chiave nella topologia Cooperlat poiché sono gli *switch* a più alto livello della LAN e svolgono routing tra VLAN, funzione di default-gateway e, a livello STP, lo *switch* 4507r è il *Root Bridge* mentre lo *switch* 9300 è il secondario.

Avendo importato in GNS3 le configurazioni reali degli *switch* Cooperlat, sono stati abilitati anche servizi di cui effettivamente la simulazione GNS3 non necessita.

Non c'è infatti la necessità di evitare un *single point of failure* a livello di routing, per cui abbiamo optato per la disabilitazione del protocollo VRRP, notando fin da subito un miglioramento delle prestazioni e raggiungendo percentuali di utilizzo CPU del 66% per lo *switch* 4507r e del 53% per lo *switch* 9300 come mostrato in seguito.

- CON VRRP disattivato sulle interface VLAN

SW-4507R-CED#sh proc cpu sorted

CPU utilization for five seconds: 75%/0%; one minute: 76%; **five minutes: 66%**

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	299509	25111	11927	68.75%	66.13%	51.41%	0	Spanning Tree
70	6857	497	13796	1.49%	1.34%	1.09%	0	Per-Second Jobs
3	8164	770	10602	0.74%	0.96%	1.17%	0	Exec
62	1662	2298	723	0.74%	0.77%	0.39%	0	IOSv e1000
87	2056	2746	748	0.58%	0.35%	0.31%	0	VRRS Main thread

SW\_9300\_CED#sh proc cpu sorted

CPU utilization for five seconds: 54%/0%; one minute: 58%; **five minutes: 53%**

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	267873	40911	6547	46.54%	49.11%	42.57%	0	Spanning Tree
70	7089	526	13477	1.28%	1.32%	1.10%	0	Per-Second Jobs
108	2008	4534	442	1.20%	1.00%	0.71%	0	UDLD
74	3460	31241	110	1.12%	0.77%	0.55%	0	Ethernet Msec Ti
87	3937	6201	634	1.04%	0.77%	0.64%	0	VRRS Main thread

Tali valori sono ancora lontani da quelli auspicabili, ma questo è comprensibile in quanto in una topologia fisica, la capacità di calcolo è distribuita per ogni singolo *switch*, mentre in una topologia virtuale, si sfruttano quelle dell'unico hardware che la sostiene.

Resta però il fatto che tali soglie di utilizzo CPU risultano ancora molto elevate, e quindi, si è tentata una ulteriore modifica riguardante questa volta il modo di operare del protocollo STP.

Di default gli *switch* Cisco, come anche gli *switch* Cooperlat, sono configurati per lavorare con il protocollo proprietario PvST+. Come approfondito nel Capitolo 3.7, la principale funzionalità del PvST+ è la

capacità di gestire istanze STP diverse per ogni VLAN, permettendo quindi di differenziare quello che è il *Root Bridge* per ogni singola VLAN e bilanciando perciò il carico computazionale degli *switch* e la banda dei collegamenti tra questi.

Nella LAN Cooperlat questa funzionalità non è sfruttata, in quanto è lo *switch* Catalyst 4507r ad essere eletto *Root Bridge* per tutte le VLAN, come visibile da un estratto della configurazione mostrato di seguito.

```
SW-4507R-CED#sh Spanning Tree summary
Switch is in pvst mode
Root bridge for: VLAN0001-VLAN0002, VLAN0004-VLAN0005, VLAN0015, VLAN0030
VLAN0040, VLAN0049-VLAN0052, VLAN0100-VLAN0113, VLAN0115-VLAN0116
VLAN0200-VLAN0203, VLAN0240-VLAN0242, VLAN0252
```

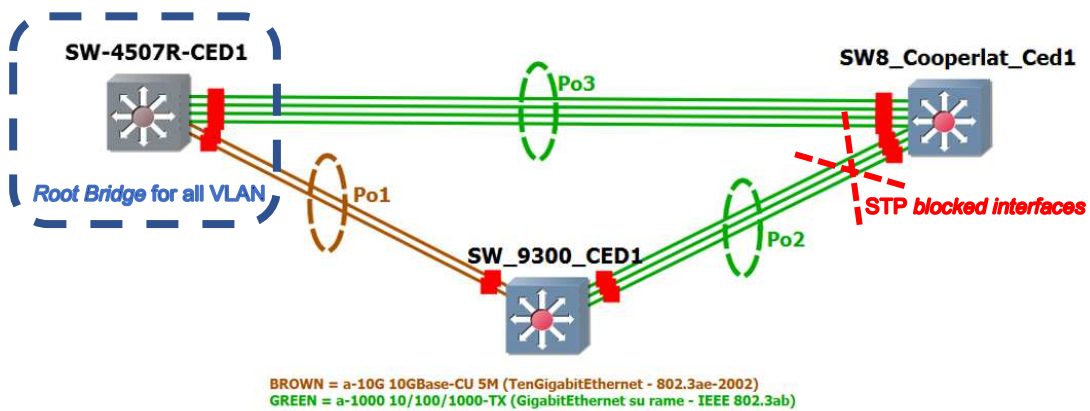


Figura 6-n: Core distribution nella LAN Cooperlat.

Si è quindi pensato di adottare una diversificazione dei *Root Bridge* sui 3 *switch* che formano il core distributivo della topologia, ed osservare la variazione di occupazione di risorse a seguito di tale modifica.



Le modifiche apportate ai tre *switch* sono di fatto due:

- È stato cambiato lo **Spanning Tree mode**, passando alla versione Rapid-PvST+ così da ottenere tempi di convergenza più rapidi;
- Modificata la **Priority di Spanning Tree** relative alle VLAN, in modo tale da eleggere:
  - lo **SW-4507R-CED Root Bridge** per le VLAN dalla 1 alla 40;
  - lo **SW\_9300\_CED Root Bridge** per le VLAN dalla 49 alla 103;
  - lo **SW8-COPERLAT-CED Root Bridge** per le VLAN restanti fino alla 252.

Riportiamo ora il codice IOS per la configurazione sopramenzionata, e di seguito i risultati delle rielezioni per i *Root Bridge*.

```
SW-4507R-CED#sh run | i spanning
Spanning Tree mode rapid-pvst
Spanning Tree extend system-id
Spanning Tree vlan 1-48,53-99,114,117-199,253-1005 Priority 4096
Spanning Tree vlan 49-52,100-113,115-116 Priority 8192
Spanning Tree vlan 200-252 Priority 12288
```

```
SW_9300_CED#sh run | i spanning
Spanning Tree mode rapid-pvst
Spanning Tree extend system-id
Spanning Tree vlan 1-48,53-99,117-4094 Priority 8192
Spanning Tree vlan 49-52,100-103 Priority 4096
Spanning Tree vlan 104-116 Priority 28672
```

```
sw8-cooperlat-ced#sh run | i spanning
Spanning Tree mode rapid-pvst
Spanning Tree extend system-id
Spanning Tree vlan 1-103,117-199,253-1005 Priority 12288
Spanning Tree vlan 104-116,200-252 Priority 4096
```

```
SW-4507R-CED#sh Spanning Tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001-VLAN0002, VLAN0004-VLAN0005, VLAN0015, VLAN0030
VLAN0040
```

```
SW_9300_CED#sh spann summ
Switch is in rapid-pvst mode
Root bridge for: VLAN0049-VLAN0052, VLAN0100-VLAN0103
```

```
sw8-cooperlat-ced#sh spann summ
Switch is in rapid-pvst mode
Root bridge for: VLAN0104-VLAN0113, VLAN0115-VLAN0116, VLAN0200-VLAN0203
VLAN0240-VLAN0242, VLAN0252
```

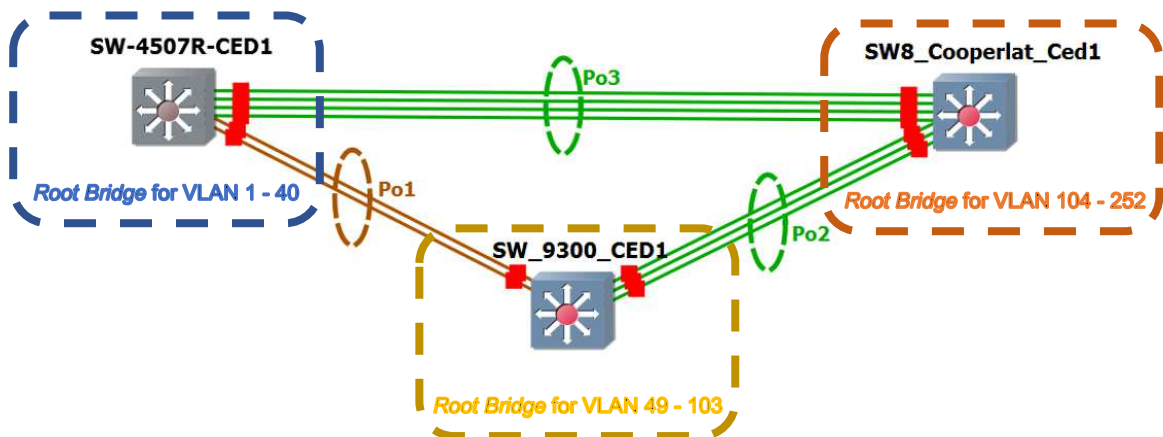


Figura 6-o: Core distribution nella LAN Cooperlat post ricalcolo Rapid-PvST+.

Per poter apprezzare i risultati delle modifiche annunciate, si sono effettuate due tipologie di confronti:

- Il primo è un confronto tra i valori di vCPU degli *switch*;
- Il secondo utilizza il software Wireshark, prima per un'analisi del *Frame rate (Frame/s)* di pacchetti scambiati relativi ai protocolli STP, e successivamente per una analisi sui *response time (ms)* relativi al protocollo ICMP<sup>13</sup>.

### 6.2.1 Confronto dei valori di vCPU

Per questo confronto si è operato direttamente da CLI utilizzando gli appositi comandi di *show* riportati di seguito.

- Valori pre modifica, con lo SW-4507R-CED eletto unico *Root Bridge* e modalità *Spanning Tree PvST+* attiva per tutti e tre gli *switch core distribution*;

---

<sup>13</sup> L'*Internet Control Message Protocol (ICMP)* è un protocollo di servizio per reti a pacchetto che si occupa di trasmettere informazioni riguardanti malfunzionamenti, informazioni di controllo o messaggi tra i vari componenti di una rete.

```
SW-4507R-CED#sh processes cpu sorted
```

```
CPU utilization for five seconds: 69%/0%; one minute: 67%; five minutes: 44%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	163291	17756	9196	60.91%	57.33%	35.59%	0	Spanning Tree

```
SW_9300_CED#sh processes cpu sorted
```

```
CPU utilization for five seconds: 84%/0%; one minute: 78%; five minutes: 78%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	1426085	147800	9648	80.01%	72.41%	71.23%	0	Spanning Tree

```
sw8-cooperlat-ced#sh processes cpu sorted
```

```
CPU utilization for five seconds: 46%/0%; one minute: 55%; five minutes: 33%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	98936	11310	8747	35.19%	42.65%	23.29%	0	Spanning Tree

- Valori post ricalcolo dei *root bridge* e passaggio alla modalità Rapid-PvST+.

```
SW-4507R-CED#sh proc cpu sort
```

```
CPU utilization for five seconds: 63%/0%; one minute: 61%; five minutes: 62%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	2103116	304382	6909	55.64%	53.78%	54.39%	0	Spanning Tree

```
SW_9300_CED#sh proc cpu sort
```

```
CPU utilization for five seconds: 55%/0%; one minute: 62%; five minutes: 64%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
132	2236945	196981	11356	47.56%	55.31%	56.78%	0	Spanning Tree

```
sw8-cooperlat-ced#sh proc cpu sort
CPU utilization for five seconds: 48%/0%; one minute: 48%; five minutes: 48%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
132   1616245       276633    5842 39.23% 38.75% 38.57%  0 Spanning Tree
```

Il risultato più rappresentativo è l'alleggerimento di risorse notato nello *switch* SW\_9300\_CED dove, anche dal grafico temporale mostrato sotto, si nota come in corrispondenza del cambiamento apportato (segnalato dalla freccia in rosso), si sia creato un gap negativo e protratto nel tempo nell'utilizzazione della CPU virtuale dello *switch*.

```
SW_9300_CED#sh proc cp hi

76677768988889998889898777888989889889899
4890309585829980785490787989178341377140521399
100      *   **   *           *           *#
90      *##* *#####*#*   *** * ** * * * *#
80      #####*#####*#####*#####*#####*##
70      *****#####
60      #####
50      #####
40      #####
30      #####
20      #####
10      #####
0...5...1...1...2...2...3...3...4...4...5...5...6
      0  5  0  5  0  5  0  5  0  5  0
      CPU% per minute (last 60 minutes)
      * = maximum CPU%  # = average CPU%
```

## 6.2.2 Confronto dei valori di *Frame rate* STP e VRRP e *Response Time* ICMP

Per il secondo confronto è stato utilizzato il toolkit Wireshark, installato in fase di setup GNS3 e per cui rimandiamo al capitolo 4.1.4.

Anche in questo caso si è proceduto con una raccolta dati *pre* e *post* modifiche proposte. In particolare con Wireshark, è possibile analizzare tutto il traffico Ethernet in transito su ogni singolo collegamento presente in topologia GNS3, e per il nostro scopo, sono stati monitorati i tre link in *PortChannel* che collegano i tre *core switch* Cooperlat.

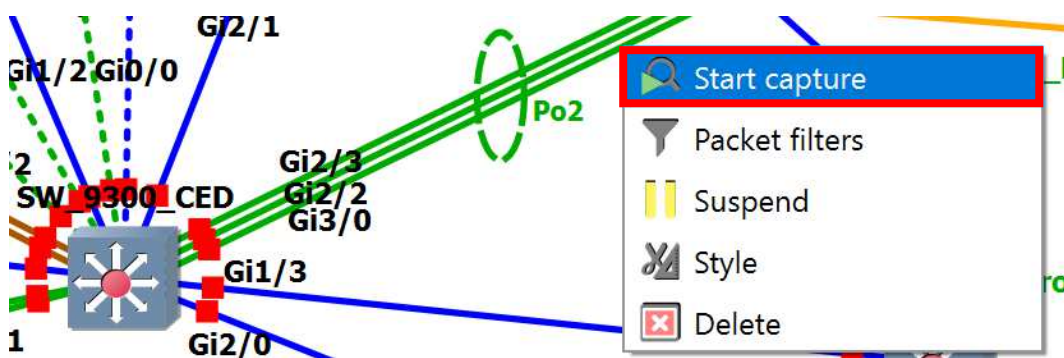
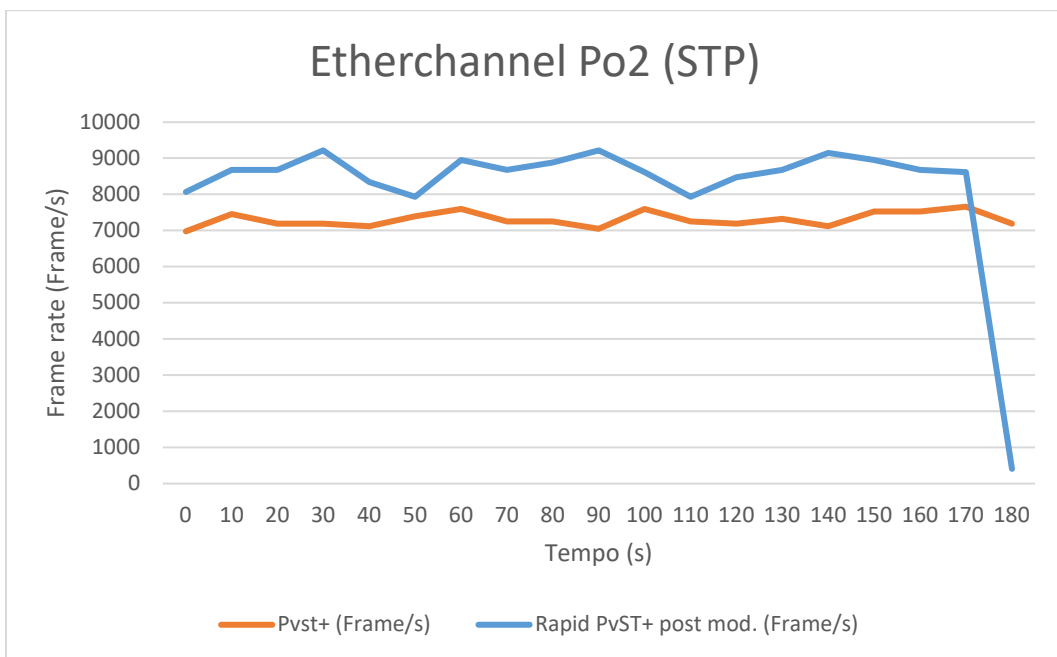
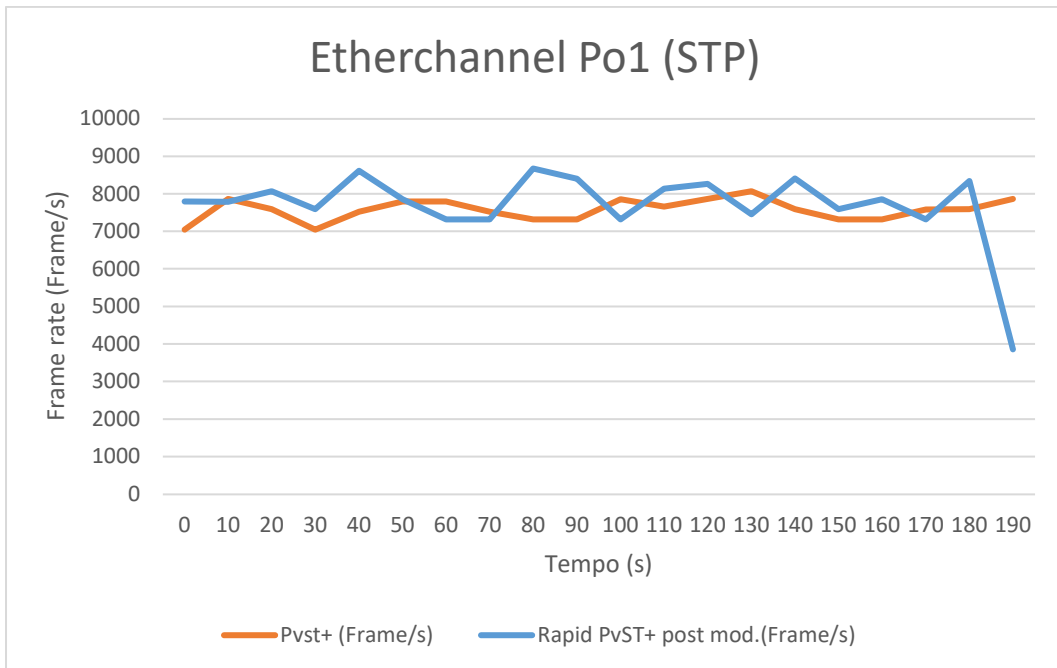
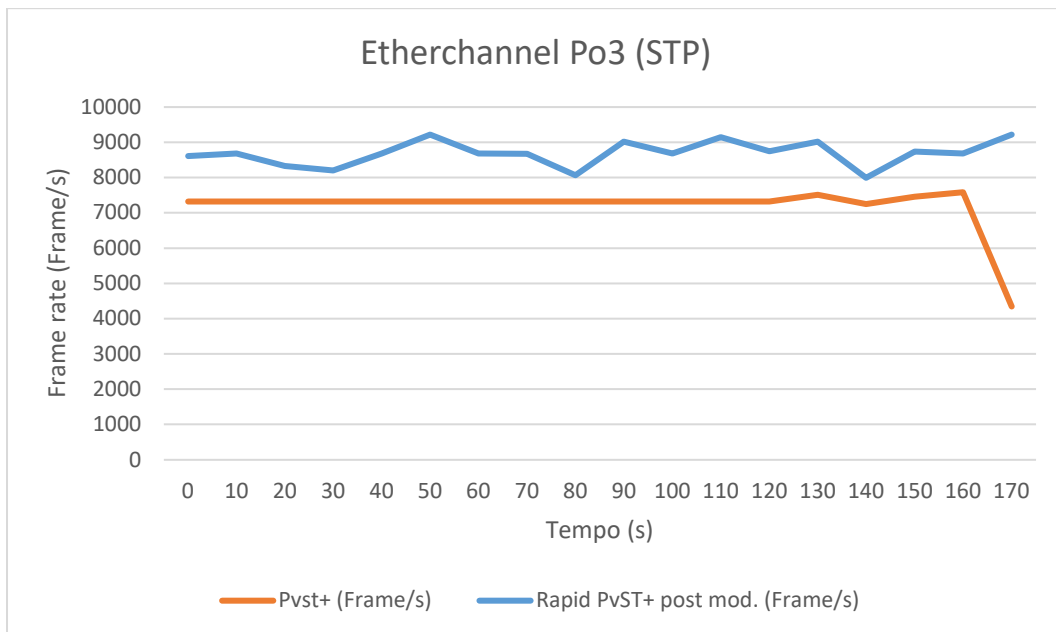


Figura 6-p: Sniffing del traffico di rete su un link GNS3 con Wireshark.

Una volta compiuto lo *sniffing* dei pacchetti in transito sui link per un tempo adeguato, si è interrotta la cattura e si è passati all'analisi dei dati raccolti filtrando prima la parte di pacchetti relativi allo STP, e poi i pacchetti relativi al protocollo ICMP. I dati filtrati ottenuti con Wireshark, sono stati direttamente copiati in Excel per poterne ricavare dei grafici di confronto.

Di seguito osserviamo tre grafici relativi ai tre *PortChannel*, che riportano il *Frame rate* (Frame/s) registrato e attinente ai frame STP.



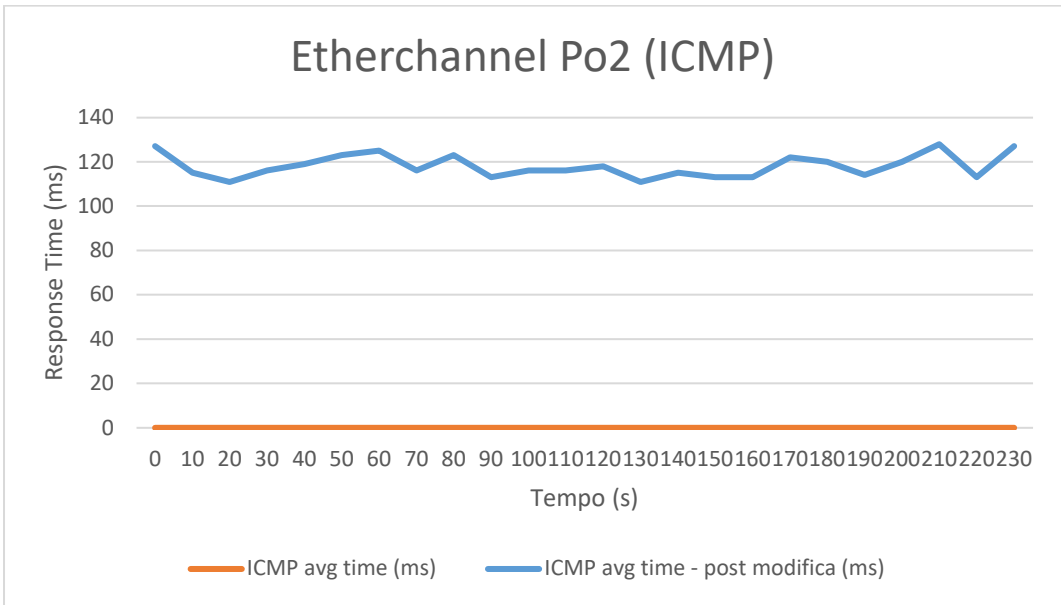
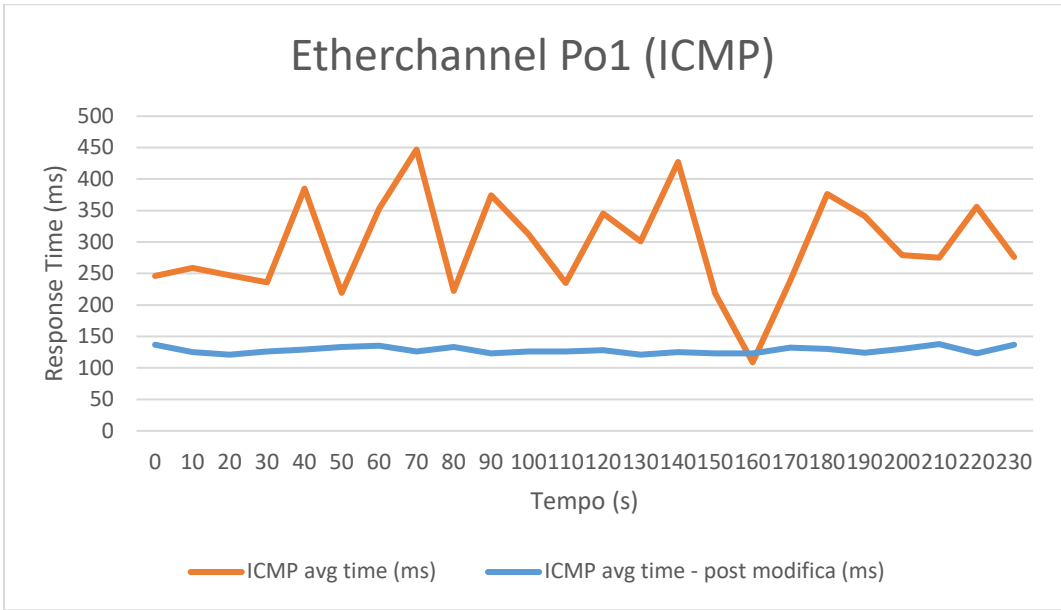


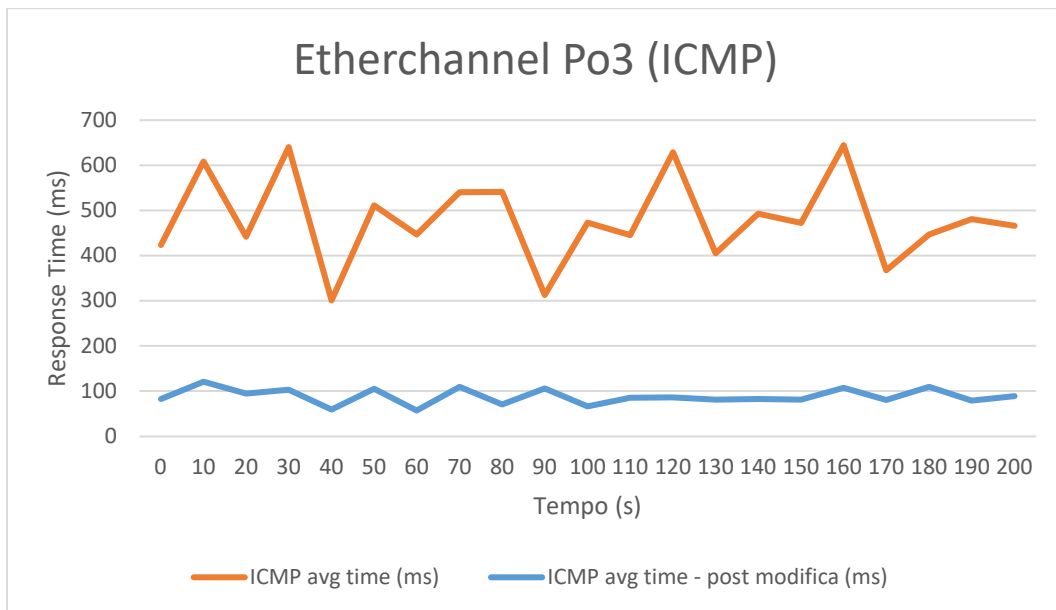
Dai grafici è possibile estrapolare le seguenti informazioni:

- Dai tracciati di *Frame rate* (Frame/s) relativi allo STP, al contrario di come ci si aspettava, si nota un leggero aumento di traffico, probabilmente dovuto al fatto che tutti e tre gli *switch* sono divenuti *root bridge* di alcune VLAN, e di conseguenza ognuno di essi possiede tutte le interfacce in stato *designated port* (per le sole VLAN di cui sono *root bridge*), inoltrando quindi le proprie BPDU.

Passando all'analisi dei *Response Time* medi ICMP, si osserva quanto segue.







- Nei *PortChannel Po1* e *Po3* si rileva un notevole abbassamento dei valori medi di *Response Time* ma anche una “stabilizzazione” di quest’ultimi, a conferma del miglioramento prestazionale degli *switch*;
- Nel *PortChannel Po2*, non si rileva alcun traffico ICMP *pre* modifica perché il link è in stato di *blocking*, mentre *post* modifica si registra un traffico ICMP con *Response Time* nella media.

Possiamo concludere l’analisi affermando che in controtendenza a quanto ci si aspettava, il *Frame rate* STP aumenta, ma i tempi di risposta ICMP migliorano nettamente, confermando il miglioramento prestazionale dello switching e la minore occupazione di vCPU degli *switch* in topologia GNS3.

Nonostante la simulazione non possa rispecchiare perfettamente il funzionamento reale dei dispositivi, è stato comunque possibile effettuare delle valutazioni oggettive su possibili soluzioni migliorative lato prestazionale e lato robustezza del networking aziendale.

## 7. CONCLUSIONI

Il progetto di tesi correlato al lavoro svolto in azienda Cooperlat, mi ha permesso di approfondire quello che è l'ambito del networking che risulta essere sempre più al centro delle dinamiche aziendali visto l'avanzamento di industria 4.0 e di *Internet of Things*, ma anche la comprensione e l'utilizzo di strumenti informatici come le *Virtual Machine*.

Oltre ad una crescita personale nell'ambito di un settore altamente professionalizzante, il progetto, seppur nei limiti intrinseci della simulazione in quanto tale, ha permesso di evidenziare quelle che potrebbero essere delle azioni migliorative da apportare alla rete aziendale per poterne ottimizzare le performance a livello di robustezza ma anche la sicurezza. In particolare si è data prova che la revisione del protocollo di *Spanning Tree* PvST+, differenziando quelli che sono i *Root Bridge* per le VLAN attive, si traduce in un miglioramento prestazionale degli *switch*.

Si è potuto poi apprezzare l'importanza di un servizio di monitoraggio SNMP con l'innegabile vantaggio di avere l'intera l'infrastruttura sotto controllo 24/7 e di ricevere *alerting* via mail in tempo reale nel caso di *fault* o malfunzionamenti, riducendo quindi i tempi di fermi produttivi e traducendosi quindi in un ottimo investimento nel medio e lungo periodo.

## 8. RINGRAZIAMENTI

Questo elaborato di tesi ha potuto prendere forma e sostanza grazie al tirocinio extracurricolare svolto presso l'azienda Cooperlat e che fortunatamente si è già tramutato in lavoro per me. Ringrazio quindi l'azienda per avermi dato fiducia sin dal primo momento e le basi per formarmi in questo ambito che reputo fondamentale e affascinante.

Se concludo il mio percorso di studi lo devo alla mia famiglia che da sempre mi ha sostenuto in tutte le mie decisioni e alla mia dolce e preziosa metà Costanza che mi ha dato sempre la forza di affrontare i momenti più bui e di puntare sempre più in alto credendo in me più di quanto non lo faccia io stesso, permettendomi di migliorarmi giorno dopo giorno.

Ringrazio infine i miei amici "Semolini", gli amici di una vita per l'affetto e il bene che mi hanno sempre dimostrato e che raramente si riceve da un gruppo di amici.

Come spesso si è soliti dire, mi piace ricordare che questo non è un traguardo, ma un importante inizio verso quella che sarà la mia futura vita.

## 9. INDICE DELLE IMMAGINI

Figura 2-a: Progressione storica, dai modelli di network proprietari allo standard TCP/IP.	8
Figura 2-b: Il modello di rete TCP/IP.	10
Figura 2-c: Ripristino errore di trasmissione TCP applicato ad HTTP.	13
Figura 2-d: Inoltro (Routing) delle lettere da parte del servizio postale.	15
Figura 2-e: Esempio di rete TCP/IP: tre routers con raggruppamento di indirizzi IP.	16
Figura 2-f: Esempio base di routing IP.	18
Figura 2-g: Trasmissione di un frame ethernet da Larry verso R1.	19
Figura 2-h: Cinque step di incapsulamento dati di TCP/IP.	22
Figura 2-i: Definizione degli incapsulamenti dati.	22
Figura 2-j: Confronto tra modello OSI e TCP/IP.	24
Figura 2-k: Incapsulamento OSI e Protocol Data Units.	25
Figura 3-a: Esempio di SOHO LAN Ethernet.	27
Figura 3-b: Esempio di SOHO LAN Ethernet e Wireless.	27
Figura 3-c: Esempio di ISR (Integrated Services Router).	28
Figura 3-d: Esempio di LAN aziendale cablata e wireless su singolo edificio.	29
Figura 3-e: Inoltro frame Ethernet su diverse tipologie di link.	32
Figura 3-f: Formato di frame Ethernet comunemente utilizzato.	33
Figura 3-g: Struttura di un indirizzo Ethernet unicast.	36
Figura 3-h: Creazione di due domini di broadcast mediante l'utilizzo di due switch fisici separati.	39
Figura 3-i: Creazione di due domini di broadcast mediante l'utilizzo di uno switch e due VLAN.	40
Figura 3-j: Multiswitch VLAN senza collegamento di trunking.	41
Figura 3-k: Multiswitch VLAN con collegamento di trunking, esempio di inoltro frame in trunk.	42
Figura 3-l: Routing tra due VLAN su due interfacce fisiche.	42
Figura 3-m: Esempio di Broadcast Storm in una LAN senza STP o RSTP.	43
Figura 3-n: Interruzione del loop da parte dello STP.	46
Figura 3-o: Formato del Bridge ID.	48
Figura 3-p: Inizio del processo di elezione del Root Bridge.	48
Figura 3-q: SW1 vince l'elezione come Root Bridge.	50
Figura 3-r: Come lo STP/RSTP calcola il costo di root e decide il Blocking.	52
Figura 3-s: Esempio di implementazione EtherChannel.	57
Figura 3-t: Bilanciamento di traffico con due istanze Spanning Tree per le due VLAN 1 e VLAN 2.	60
Figura 3-u: STP System ID extension.	62
Figura 4-a: Logo GNS3.	64
Figura 4-b: Esempio di topologia su Workspace GUI GNS3.	65
Figura 4-c: Installazione di Wireshark tra i toolkit disponibili nel setup GNS3.	70
Figura 4-d: Impostazioni per la GNS3 VM utilizzata nel progetto.	73
Figura 4-e: Stato della GNS3 VM e del server local host GNS3.	73
Figura 4-f: Topologia test per il nodo Cloud e l'utilizzo di end point Ubuntu.	74
Figura 4-g: Aggiunta interfaccia di rete bridge alla VM GNS3.	76
Figura 4-h: Aggiunta della scheda bridge al nodo Cloud.	77
Figura 4-i: Dashboard Oracle VirtualBox con VM Ubuntu.	78
Figura 4-j: Import della VM Ubuntu.	79
Figura 4-k: Topologia di test Ubuntu end device on Internet.	79
Figura 4-l: Risultato della configurazione dello switch.	81
Figura 4-m: Web Interface ISR D-Link dove sono visibili gli indirizzi IP assegnati al NB-MarcoBucci, alla GNS3VM e allo Switch IOSvL2.	82
Figura 4-n: Configurazione parametri di rete dell'End Point Ubuntu.	83
Figura 4-o: Regola di routing aggiunta all'ISR.	84

Figura 4-p: Console VirtualBox con Ubuntu Desktop connesso ad internet.....	84
Figura 4-q: Bug di integrazione tra VirtualBox e GNS3VM.....	85
Figura 4-r: Esempio di utilizzo del nodo NAT.....	85
Figura 4-s: Scheda di rete virtuale alla base del nodo NAT.....	87
Figura 5-a: Rappresentazione grafica del concetto di rete a Stella.....	88
Figura 5-b: Esempio di applicazione di VRRP. L'host utilizza come indirizzo di default-gateway l'indirizzo IP virtuale gestito dal VRRP.....	94
Figura 5-c: Finestra iniziale all'avvio del programma Tftpd64.....	97
Figura 5-d: Dashboard iniziale del programma PowerAutomate.....	99
Figura 5-e: Finestra di modifica e debug di un flusso PowerAutomate.....	100
Figura 5-f: Interfaccia Putty, dove al campo IP address viene inserito l'indirizzo dello switch.....	101
Figura 5-g: Architettura software ideata per l'installazione del server FTP in GNS3 e passaggio dei file di configurazione switch mediante client FTP.....	104
Figura 5-h: Primo step per l'installazione di un server.....	105
Figura 5-i: Parametri di rete ereditati dal server DHCP con connessione mediante nodo Cloud.....	106
Figura 5-j: Collegamento in localhost al server FTP.....	107
Figura 5-k: Interfaccia WinSCP al collegamento con il server FTP in esecuzione su VM Ubuntu.....	108
Figura 5-l: Connessione al nodo NAT.....	109
Figura 5-m: Parametri di rete del client Ubuntu connesso tramite nodo NAT.....	109
Figura 5-n: Configurazione delle connessioni WinSCP.....	110
Figura 5-o: Upload dei file su server FTP.....	111
Figura 5-p: File caricati nella directory /srv/ftp.....	111
Figura 5-q: Rimozione del nodo NAT e aggiunta di switch IOSvL2.....	112
Figura 5-r: Configurazione manuale dei parametri di rete del server Ubuntu FTP.....	112
Figura 5-s: Configurazione interfaccia VLAN 1 e test di ping.....	114
Figura 5-t: Esempio di import della configurazione da server FTP.....	115
Figura 5-u: Cambio dell'hostname da GUI GNS3.....	116
Figura 5-v: Topologia Cooperlat riprodotta in GNS3.....	117
Figura 6-a: Logo Nagios.....	120
Figura 6-b: Topologia Cooperlat comprensiva di server NagiosXI e nodo Cloud.....	121
Figura 6-c: Configurazione manuale dei parametri di rete del server NagiosXI.....	121
Figura 6-d: Web Interface NagiosXI raggiunta in localhost (Mozilla Firefox sopra) e da notebook Windows all'indirizzo IP 172.30.1.254 (Google Chrome sotto).....	123
Figura 6-e: Dashboard iniziale NagiosXI.....	124
Figura 6-f: Pannello Configure di NagiosXI.....	124
Figura 6-g: Configurazione dell'Auto-Discovery Job.....	125
Figura 6-h: Wizard per la configurazione del monitoraggio.....	126
Figura 6-i: Configuration Wizard per switch e router.....	127
Figura 6-j: Impostazione Timer di autocheck.....	127
Figura 6-k: Grafici e tabelle per la visualizzazione degli apparati monitorati.....	128
Figura 6-l: Configurazione Email Alerting.....	129
Figura 6-m: Alerting email inviate dall'applicazione NagiosXI direttamente su casella di posta Gmail.....	130
Figura 6-n: Core distribution nella LAN Cooperlat.....	135
Figura 6-o: Core distribution nella LAN Cooperlat post ricalcolo Rapid-PvST+.....	137
Figura 6-p: Sniffing del traffico di rete su un link GNS3 con Wireshark.....	141

## 10. BIBLIOGRAFIA

- [1]. CCNA 200-301 Official Cert Guide, Wendell Odom, Copyright © 2020 Pearson Education, Inc.
- [2]. P. Gil, G. J. Garcia, A. Delgado, R. M. Medina, A. Calderón and P. Marti, "Computer networks virtualization with GNS3: Evaluating a solution to optimize resources and achieve a distance *Learning*," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, 2014, pp. 1-4, doi: 10.1109/FIE.2014.7044343.
- [3]. S. Liu, H. Wang, J. Liu and M. Xian, "Feasibility analysis of network security teaching platform based on KVM and GNS3," 2019 International Conference on Information Technology and Computer Application (ITCA), 2019, pp. 310-313, doi: 10.1109/ITCA49981.2019.00075.
- [4]. J. Renita and N. E. Elizabeth, "Network's server monitoring and analysis using Nagios," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 1904-1909, doi: 10.1109/WiSPNET.2017.8300092.
- [5]. M. A. A. bin Mohd Shuhaimi, Z. binti Zainal Abidin, I. binti Roslan and S. binti Anawar, "The new services in Nagios: Network bandwidth utility, email notification and sms alert in improving the network performance," 2011 7th International Conference on Information Assurance and Security (IAS), 2011, pp. 86-91, doi: 10.1109/ISIAS.2011.6122800.



## 11. SITOGRAFIA

- [1]. [www.gns3.com](http://www.gns3.com)
- [2]. [www.Learningnetwork.cisco.com](http://www.Learningnetwork.cisco.com)
- [3]. [www.ubuntu.com](http://www.ubuntu.com)
- [4]. [www.nagios.com](http://www.nagios.com)
- [5]. [www.phoenixnap.com](http://www.phoenixnap.com)
- [6]. [www.winscp.net](http://www.winscp.net)
- [7]. [www.pjo2.github.io/tftpd64/](http://www.pjo2.github.io/tftpd64/)

## 12. APPENDICE

### A. Codice IOS – switchport

1) Prima si seleziona l'interfaccia desiderata

```
interface GigabitEthernet0/1
```

2) Una volta all'interno della configurazione di interfaccia è possibile dotarla di una descrizione e negare la funzione della stessa in switchport in modo tale da ottenere funzione di Layers 3

```
description Link to External
```

```
no switchport
```

3) Si imposta l'interfaccia in modo tale da ereditare le impostazioni di rete dal server DHCP

```
ip address dhcp
```

```
negotiation auto
```

```
no cdp enable
```

```
end
```

### B. Codice IOS - indirizzo IP su VLAN interface

1) Prima si seleziona l'interfaccia virtuale di interesse, in questo caso la VLAN 1

```
interface Vlan1
```

2) si assegna indirizzo IPv4 e subnet mask

```
ip address 172.30.1.1 255.255.255.0
```

```
end
```

## C. Codice IOS - backup su server TFTP

1) Viene dapprima impartito il comando

```
copy running-config tftp:
```

2) Si specifica l'indirizzo IP del server TFTP

```
Address or name of remote host []? xxx.xxx.xxx.xxx
```

3) È possibile personalizzare il nome del file che si andrà ad inviare

```
Destination filename [ce_2-config]? backup_cfg_for_my_router
```

```
!!
```

```
1030 bytes copied in 2.489 secs (395 bytes/sec)
```

## D. Command Line Terminal Ubuntu - server FTP

# Innanzitutto si inizia con un update generico dell'OS Ubuntu.

```
user@user-VirtualBox:~$ sudo apt update
```

# Ora si installa il servizio server FTP.

```
user@user-VirtualBox:~$ sudo apt install vsftpd
```

# Terminata l'installazione si avvia il servizio

```
user@user-VirtualBox:~$ sudo systemctl start vsftpd
```

# e si abilita lo stesso.

```
user@user-VirtualBox:~$ sudo systemctl enable vsftpd
```

# Per precauzione si effettua una copia del file di configurazione del servizio.

```
user@user-VirtualBox:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

# E' possibile creare nuovi utenti che si possano poi collegare al server FTP.

```
user@user-VirtualBox:~$ sudo useradd -m testuser
```

```
user@user-VirtualBox:~$ sudo passwd testuser
```

# Per sicurezza di modificano le regole firewall del OS aprendo le porte 20 e 21 TCP.

```
user@user-VirtualBox:~$ sudo ufw allow 20/tcp
```

```
user@user-VirtualBox:~$ sudo ufw allow 21/tcp
```

## E. Codice IOS - PortChannel

1) Prima di tutto spegnere le singole interfacce ed eliminare eventuali port channel già esistenti.

```
conf t
interface range GigabitEthernet 1/0-1
shout down
exit
```

2) Resettare le interfacce con il comando default.

```
default interface range GigabitEthernet 1/0-1
```

3) Configurare il port channel.

```
interface range GigabitEthernet 1/0-1
channel-group 1 mode desirable
```

4) Configurare le singole interfacce in trunk (ma NON ACCENDERLE).

```
interface range GigabitEthernet 1/0-1
description Po1 to SW-4507
switchport trunk encapsulation dot1q
switchport mode trunk
```

5) configurare il trunk sul Po1.

```
interface Po1
description trunk to SW-4507
switchport trunk encapsulation dot1q
switchport mode trunk
```

6) A questo punto accendere le interfacce facenti parte del port-channel.

```
int range GigabitEthernet 1/0-1
no shout down
```

## F. Codice IOS - VTP server e VLAN

```
vtp domain cooperlat.it
vtp mode server
vtp password *****
vtp version 2
```

```
vlan 1
name default
vlan 2
name lan_Cooperlat
vlan 4
name tetrapack
vlan 5
name elettrici
vlan 15
name mgmt_nuova
vlan 30
name Vodafone-1
vlan 40
name Vodafone-2
vlan 49
name CGS
vlan 50
name DMZ
vlan 51
name outside
vlan 52
name DMZ-Servizi
vlan 100
name Wifi
vlan 101
name SIM
vlan 102
name MES
vlan 103
name ospiti
vlan 104
name WiFi-Tetrapack
vlan 105
name SI
```

```
vlan 106
name Trascar
vlan 107
name tecnova
vlan 108
name auteco
vlan 109
name wifi-cellulari
vlan 110
name 3Com
vlan 111
name videosorveglianza
vlan 112
name prasmatic
vlan 113
name citrix
vlan 115
name Cogenerazione
vlan 116
name sql_cluster
vlan 200
name HMC
vlan 201
name VMotion
vlan 202
name PTP-ASA
vlan 203
name ASA-Failover
vlan 240
name voip-server
vlan 241
name voip
vlan 242
name ip_phones
vlan 252
name DMZ-guest
```

## G. Codice IOS - VTP client

```
vtp domain cooperlat.it  
vtp mode client  
vtp password *****
```

## H. Codice IOS - *Community SNMP name*

```
configure terminal  
# Per abilitare il protocollo SNMP e l'accesso in READ-ONLY settare  
la seguente community  
set snmp-server community cooperlatro RO
```