



UNIVERSITA' DEGLI STUDI DI  
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base  
Corso di Laurea in Ingegneria Informatica

Elaborato finale in **Reti di Calcolatori**

***Simulazione di una rete con Routing OSPF  
in Cisco Packet Tracer***

Anno Accademico 2014/2015

Candidato:

**Dario D'Alessandro**

**matr. N46001250**

---

Un ringraziamento ai miei genitori  
e a tutta la mia famiglia che mi ha  
sempre sostenuto per poter  
giungere a questo traguardo.

Un ringraziamento a quegli amici  
di lunga data che oramai chiamo  
fratelli e a quelli più recenti con i  
quali ho stretto un rapporto  
davvero particolare, che mi hanno  
sempre incoraggiato.

---

# Indice

---

Indice.....	III
Introduzione.....	4
Capitolo 1: Instradamento .....	5
1.1 algoritmo di Dijkstra.....	6
1.2 OSPF.....	8
1.3 Aree .....	9
1.4 Comunicazione OSPF .....	10
1.5 Link-State Advertisement.....	11
1.6 Protocolli OSPF .....	12
1.6.1 Header Comune.....	12
1.6.2 Protocollo Hello .....	12
1.6.3 Protocollo Exchange.....	13
1.6.4 Protocollo Flooding .....	14
1.7 Sicurezza.....	15
Capitolo 2: Cisco Packet Tracer .....	16
2.1 Presentazione .....	17
2.2 Livello Hardware.....	18
2.3 Livello Software.....	18
2.4 Simulazioni .....	19
2.4.1 Real-Time .....	19
2.4.2 Simulation .....	19
2.5 PING.....	20
2.6 Traceroute .....	20
Capitolo 3: Simulazione .....	22
3.1 Strutturazione della rete.....	22
3.2 Configurazione indirizzi IP.....	23
3.3 Analisi Tabelle di Routing.....	26
3.4 Configurazione OSPF.....	27
3.4.1 Analisi Tabelle di Routing.....	29
3.5 Testing .....	30
3.6 Link Fail.....	32
Cocclusioni.....	36
Bibliografia.....	37

## Introduzione

---

Attualmente assume sempre più importanza la gestione dei sistemi autonomi, anche di dimensioni notevoli e di elevata complessità.

Per soddisfare tali esigenze sono stati quindi introdotti diversi protocolli per la gestione di questo elevato flusso di dati.

Uno dei più recenti e maggiormente utilizzato è l'OSPF che verrà presentato di seguito nell'elaborato, insieme alla simulazione di una rete basata su tale protocollo.

A tal fine verrà usato l'applicativo "Cisco Packet Tracer", che consentirà di simulare e analizzare varie situazioni e di creare una rete al quanto fedele alla realtà.

Questo lavoro sarà suddiviso in tre capitoli:

- nel CAPITOLO I saranno introdotti i concetti fondamentali dell'instradamento e verrà presentato il protocollo OSPF e il suo funzionamento.
- nel CAPITOLO II sarà introdotto il software con il quale sono state effettuate le simulazioni e verranno presentate alcune funzionalità utili per la simulazione.
- nel CAPITOLO III verrà effettuata la vera e propria simulazione di una rete, sia nell'ipotesi che la rete funzioni correttamente, sia nel caso in cui avvenga un malfunzionamento.

Verranno infine presentate le opportune analisi per comprendere il comportamento del protocollo nelle diverse situazioni.

## Capitolo 1: Instradamento

---

L'*instradamento* è quel processo di rete che determina i percorsi dei pacchetti nel loro viaggio, dall'origine alla destinazione.

La scelta del percorso da seguire viene effettuata tramite un *algoritmo di instradamento*. Solitamente questo algoritmo ha lo scopo di trovare un percorso ottimale in termini di costo, dove con *costo* possiamo intendere la distanza tra due router, la velocità di trasmissione e altre metriche di analisi.

Il cammino risultante dal routing viene memorizzato in un database interno al router che prende il nome di “tabella di instradamento” (routing table), e contiene informazioni sul next hop per giungere ad una determinata sottorete.

Gli algoritmi di instradamento, possono essere suddivisi in:

- **Algoritmi di instradamento “statici”**: questo tipo di algoritmo prevede, che i cammini vengano modificati raramente e che le dimensioni della rete non siano troppo elevate per cui, una volta che sono stati assegnati i cammini, non cambiano più salvo, per esempio, in seguito all'intervento umano sui router.
- **Algoritmi di instradamento “dinamici”**: questi algoritmi, sono utili per reti che variano nel tempo e le cui dimensioni sono abbastanza elevate. L'instradamento dinamico prevede che l'algoritmo modifichi i percorsi dei pacchetti, in base alle variazioni della topologia della rete e al volume di traffico presente.

\*

Un altro criterio di classificazione è basato sul *tipo* di informazioni che l'algoritmo mette a disposizione dei router. Possiamo quindi avere:

- **Algoritmo di instradamento “globale”**: questa tecnica prevede che l'algoritmo abbia a disposizione una conoscenza globale e completa della rete, in modo da poter calcolare il cammino più breve, tra origine e destinazione. Per fare ciò, ciascun router conosce inizialmente solo il proprio ambito locale (linee e nodi adiacenti) dopodiché trasmette queste informazioni a tutti gli altri nodi della rete, tramite dei Link-State Packet. Questo tipo di algoritmo viene pure chiamato *Link-State Algorithm* (LS).
- **Algoritmo di instradamento “decentralizzato”**: questa tecnica prevede che nessun nodo possieda informazioni complete sul costo di tutti i collegamenti della rete: attraverso un processo iterativo e scambio di informazioni, un nodo può calcolare gradualmente il percorso di costo minimo verso una destinazione, in quanto i nodi inviano a quelli adiacenti le informazioni da loro possedute, in modo da permettere l'aggiornamento delle tabelle di instradamento. Questo tipo di algoritmo prende anche il nome di *Distance-Vector Algorithm* (DV).

\*

In questo elaborato soffermeremo la nostra attenzione su un particolare Link-State Algorithm, cioè l'*algoritmo di Dijkstra* che prende il nome dal suo ideatore Edsger W. Dijkstra.

## 1.1 algoritmo di Dijkstra

Questo algoritmo, essendo Link-State, prevede che ciascun nodo abbia a disposizione informazioni sul grafo della rete e sui costi di tutti i collegamenti. Ciò può essere fatto tramite i Link-State Packet che vengono distribuiti sulla rete tramite un algoritmo Link-State Broadcast.

Utilizzando questo algoritmo, ciascun nodo costruisce uno *Spanning Tree*, cioè l'albero dei cammini di costo minimo verso tutte le destinazioni.

Prima di analizzare la sequenza di operazioni eseguite dall'algoritmo, bisogna formalizzare alcuni elementi.

I nodi sono suddivisi in due insiemi:

- a) l'insieme **PATH** di nodi, per i quali si è già trovato il percorso migliore.
- b) l'insieme **TEMP** di nodi, per i quali si sta cercando un percorso.

Il nodo che sta calcolando l'algoritmo viene detto *nodo radice* (root).

Per ciascun nodo del grafo, viene definita un'etichetta formata da tre valori, che possono essere modificati man mano che l'algoritmo avanza nella sua esecuzione.

Gli elementi dell'etichetta sono:

Un **ID** che definisce il nodo.

- Un valore “**distanza** (V)” che rappresenta il costo noto finora del cammino minimo dal root a V.
- Un indice “**link** (V)” che identifica il predecessore di V nell'attuale cammino minimo dal root a V.

A questo punto, possiamo vedere quali sono i passi fondamentali di questo algoritmo:

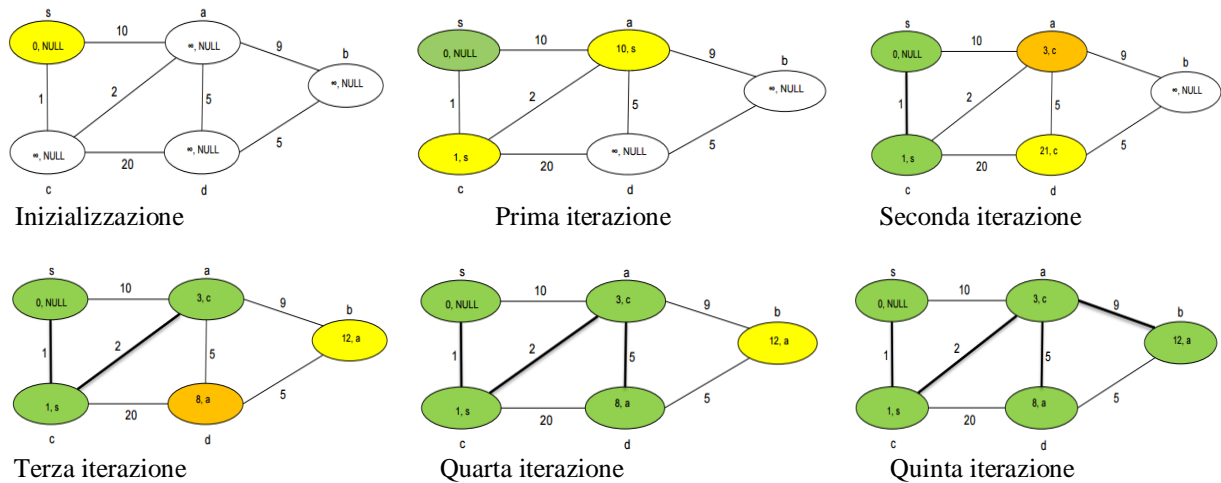
- 1) Viene scelto un nodo radice, che viene inserito in PATH.
- 2) Gli altri nodi della rete vengono inseriti in TEMP.
- 3) Viene prelevato dall'insieme TEMP il nodo N che ha il percorso più breve dal root e viene posto in PATH.
- 4) Ciascun router vicino V al nodo viene passato in PATH:

4.1) Se V non è presente in TEMP viene inserito.

5) Si calcola la distanza tra V e root eseguendo la seguente operazione [distanza ( root-N ) + distanza ( N-V )] se questo valore è minore di quello precedentemente noto per raggiungere V, vengono aggiornati costo e link di quel nodo in TEMP.

6) Si ripete iterativamente dal passo 3 fino a che non si ha la conoscenza globale della rete, cioè lo svuotamento dell'insieme TEMP.

Supposta la seguente rete, vediamo come si comporta questo algoritmo, considerando il nodo *s* quello di root:



Dalle figure si può vedere che si è avuta una prima fase di inizializzazione, dove i nodi non adiacenti sono stati posti pari ad una distanza uguale ad infinito. Dalla seconda figura in poi, ci sono state le iterazioni dell'algoritmo che hanno portato alla conoscenza globale della rete da parte di *s*.

## 1.2 OSPF

Dopo aver richiamato alcuni concetti fondamentali, che stanno alla base del protocollo OSPF, passiamo ora ad analizzare il funzionamento.

Questo protocollo fu sviluppato dall'Internet Engineering Task Force (IETF), in particolare dal Working Group OSPF, al fine di creare un protocollo che superasse i limiti di RIP.

La prima versione fu definita nel 1989 nel RFC 1131. Con il passare degli anni, il protocollo ha avuto molteplici miglioramenti, fino ad arrivare alla versione più recente, OSPF-v2, definita nel RFC 2328, nel 1998.

Il termine **OSPF** sta per "*open shortest path first*" dove con *open* si indica che le specifiche del protocollo sono pubblicamente disponibili.

Ad oggi, è il protocollo più usato in Internet per il routing interno a sistemi autonomi.

OSPF è di tipo *Link-State*, cioè ciascun router costruisce una mappa topologica del sistema autonomo, ponendo se stesso come root, ed eseguendo l'algoritmo di Dijkstra per determinare l'albero delle rotte minime verso tutte le sottoreti: se due percorsi hanno lo stesso costo, il carico verrà distribuito equamente tra i due.



Sarà compito dell'amministratore di rete impostare i costi dei collegamenti, in modo da favorire o meno il passaggio per un hop.

### 1.3 Aree

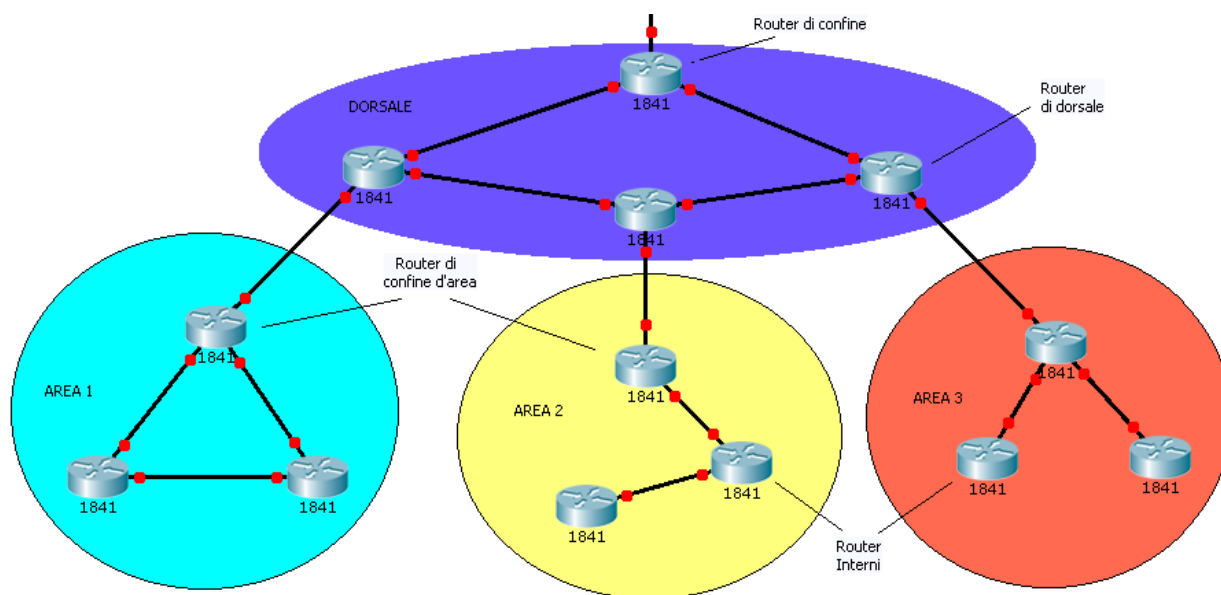
Essendo richiesto per ciascun router la conoscenza globale della rete, si potrebbe andare incontro ad un overhead nell'esecuzione dell'algoritmo di Dijkstra. Per risolvere questo problema, sono state introdotte le *aree*.

Ciascuna area è indipendente dalle altre nell'esecuzione dell'algoritmo di instradamento, per cui ciascun router invia lo stato dei suoi collegamenti solo a quelli della propria area e, i dettagli interni di un area, rimangono invisibili ai router esterni.

Per poter far comunicare tra loro le varie aree, vi sono appositi router di confine che si occupano dell'instradamento dei pacchetti verso l'esterno. Ciascun sistema autonomo, suddiviso in più aree, richiede una dorsale (*backbone area*), il cui ruolo è quello di instradare il traffico tra le varie aree.

Quindi possiamo suddividere i router nel seguente modo:

- **Router “interni”** : si trovano in aree non dorsali ed effettuano soltanto l'instradamento interno al sistema autonomo.
- **Router di “confine d'area”** : appartengono sia ad una generica area sia alla dorsale.
- **Router di “dorsale (non di confine)”** : Effettuano l'instradamento all'interno della dorsale, ma non sono router di confine.
- **Router di “confine”** : Scambiano informazioni d'instradamento con i router di altri sistemi autonomi.



## 1.4 Comunicazione OSPF

Ciascun router invia periodicamente lo stato dei collegamenti anche se non sono cambiati, al fine di aumentare la robustezza del protocollo. Solitamente il periodo che intercorre tra un invio e il successivo è di 30 minuti, anche se questo parametro può essere modificato dall'amministratore di rete Designated Router

OSPF richiede la *sincronizzazione* tra gli N router dell'area, il che comporta un elevato traffico sulla rete con una complessità pari a  $N*(N-1)$  ogni volta che ci sarà un cambiamento di stato. Al fine di ridurre questa complessità sono stati introdotti i *Designated Router* (DR).

Il Designated Router è un router che viene eletto tra quelli facenti parte di un'area, il cui compito principale è quello di essere un riferimento per gli altri router. Tramite il DR, i restanti router possono sincronizzare i propri database senza dover richiedere singolarmente le informazioni, riducendo quindi la quantità di messaggi trasmessi.

Le modalità di funzionamento, variano se abbiamo una rete Broadcast o meno:

- Se la rete è *Broadcast*, quando un router deve inviare un aggiornamento, lo trasmette al Designated Router usando l'indirizzo multicast 224.0.0.6. Sarà poi compito di quest'ultimo eseguire un flooding a tutta la rete sull'indirizzo multicast 224.0.0.5.

- Se la rete *non è Broadcast*, quando un router invia un aggiornamento, lo invia al Designated Router, su un collegamento punto-punto. Quest'ultimo a sua volta invia una copia separatamente ad ogni altro router.

Oltre ad eleggere il Designated Router, viene pure eletto il *Backup Designated Router* (BDR), il cui compito è quello di sostituire il DR nel caso in cui subisca una disconnessione, finché non viene eletto il nuovo DR.

## 1.5 Link-State Advertisement

Ogni router mantiene in memoria un *Link-State database*, composto da Record Link-State per ciascuna area su cui è operativo, il quale altro non è che la descrizione topologica dell'area a cui il router è collegato.

I vari Record Link-State, vengono aggiornati tramite degli annunci detti *Link-State Advertisement (LSA)*. Gli LSA vengono generati da ciascun router dell'autonomous system, a seconda delle competenze ad esso assegnate, e vengono trasportati all'interno di pacchetti IP. Esistono cinque tipi di LSA:

- **Tipo 1 – “Router” Link:** sono generati da ciascun router all'interno dell'AS e per ciascun area a cui il router è attaccato: essi servono a descrivere lo stato e l'identità delle connessioni di un router all'interno di un area.
- **Tipo 2 – “Network” Link:** sono generati, per ciascuna rete multi-accesso, dal Designated Router e sono propagati all'interno dell'area di cui appartiene. Essi hanno lo scopo di descrivere lo stato delle connessioni per quella rete.
- **Tipo 3 – “Network Summary” Link:** sono generati dai router di confine d'area e propagati all'interno di una sola area per volta e contengono il summary delle route a cui sono connessi.
- **Tipo 4 – “AS external ASBR Summary” Link:** è generato da router di confine per notificare la propria presenza.
- **Tipo 5- “External” Link:** sono generati dai router di confine contenenti informazioni relative a destinazioni su Autonomous System esterni.

## 1.6 Protocolli OSPF

L'**OSPF** in realtà è composto da tre sotto-protocolli che permettono l'instaurazione di una comunicazione tra i router e lo scambio degli LSA necessari per poter sincronizzare i database.

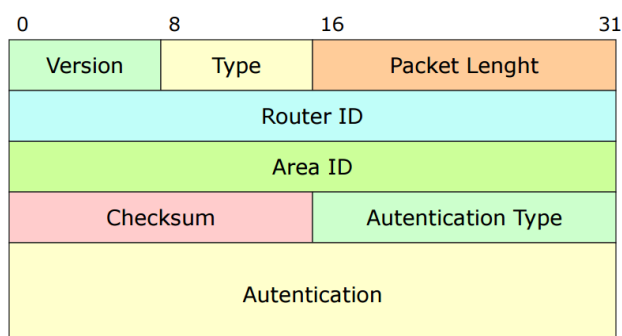
I tre sotto-protocolli sono **HELLO**, **EXCHANGE** e **FLOODING**. Tutti e tre sono accomunati da un *header* comune.

Analizziamo ora i singoli sotto-protocolli e l'header comune

### 1.6.1 Header Comune

L'header sarà composto dai seguenti campi:

- **Version:** indica la *corrente versione* OSPF, attualmente la 2.
- **Type:** il *tipo di pacchetto* trasportato (Hello, Exchange, Flooding).
- **Router\_ID:** l'*indirizzo IP* scelto per identificare il router.
- **Area\_ID:** *numero* che identifica univocamente l'area all'interno del dominio OSPF.
- **Checksum:** viene calcolato sull'intero pacchetto OSPF, ad esclusione del campo Authentication.
- **Authentication:** identifica l'*algoritmo* di autenticazione.

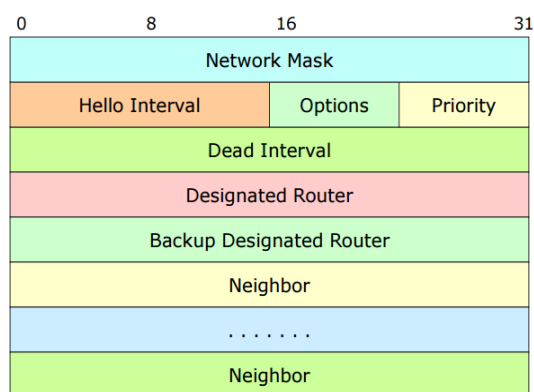


### 1.6.2 Protocollo Hello

Al fine di conoscere i propri vicini e verificarne l'operatività, ciascun Router manda in esecuzione il protocollo Hello, che assicura una comunicazione bidirezionale con essi.

Viene utilizzato anche per eleggere il *Designated Router* e il *Backup Designated Router* per quella rete. La struttura dei pacchetti Hello è la seguente:

- **Network Mask:** netmask associata all'interfaccia, da cui viene emesso il pacchetto.
- **Hello Interval:** comunica ogni quanti secondi viene emesso un pacchetto.
- **Options:** vengono definiti solo gli ultimi 2 bit i quali sono: **E** per vedere se il router è in grado di inviare o ricevere route esterne, e **G** se il router è in grado di gestire il routing TOS.
- **Priority:** serve per l'elezione del Designated Router: il router a priorità più alta diventerà quello designato.
- **Dead\_Intervall:** intervallo di validità dei pacchetti di Hello ricevuti.
- **Designated/Backup Designated Router:** indirizzo del Designated Router e del Backup Designated Router.
- **Neighbor:** lista di router di cui è stato ricevuto il pacchetto Hello negli ultimi Dead\_Intervall.



### 1.6.3 Protocollo Exchange

Per ridurre il traffico sulla rete, la sincronizzazione dei database dei router non avviene verso tutti i router, ma solo verso il Designated Router. La sincronizzazione iniziale avviene tramite il protocollo EXCHANGE.

Durante questo processo, tra due router, viene stabilita una relazione Master/Slave. Ogni pacchetto inviato ha un proprio *Sequence Number* concordato all'inizio della procedura. Il Master invia pacchetti di Database Description Packet, contenente una lista di Link State Advertisements, allo Slave che risponde con un pacchetto dello stesso tipo contenente una

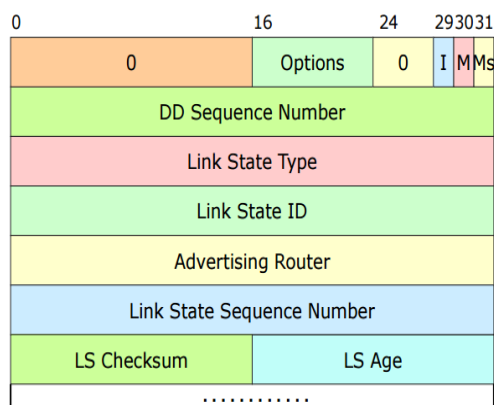
lista di LSA riguardanti il proprio database e con lo stesso Sequence Number onde inviare anche un Acknowledge al Master. Durante questo scambio, ogni router prende nota degli advertisements più recenti (rispetto ai propri) dell'altro router. Questi advertisements verranno poi richiesti uno ad uno appena finito il processo di scambio dei Database Description Packet. Questi aggiornamenti verranno richiesti tramite Link-State Request cui si risponde con pacchetti di Link-State Update. Concluso l'intero processo di sincronizzazione, l'adiacenza viene considerata pienamente funzionante e mantenuta attiva tramite l'invio ad intervalli regolari di pacchetti Hello contenenti o meno informazioni.

Di seguito i campi del Database Description Packet:

**Options:** Equivalenti al pacchetto Hello

- **I:** Initialize
- **M:** More
- **MS:** Master – Slave
- **DD:** Sequence Number: Numero di sequenza del pacchetto

I campi successivi, sono la descrizione dell'header:



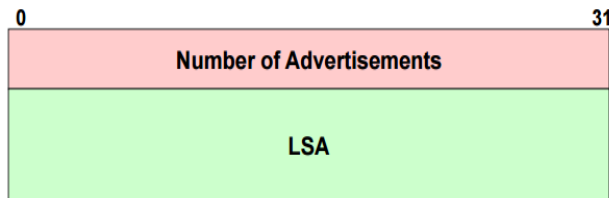
#### 1.6.4 Protocollo Flooding

Il protocollo di *flooding* ha lo scopo di diffondere a tutta la rete il nuovo stato di un link. Questi aggiornamenti vengono trasmessi dal Designated Router, attraverso pacchetti Link-State Update.

L'esecuzione di questo protocollo, oltre ad avvenire in seguito ad un cambiamento di link, avviene anche in seguito allo scadere di un timer, di solito impostato a 30 minuti.

I campi del Link-State Update sono:

- **Number of Advertisements:** indica il numero di LSA che vengono trasportati dal pacchetto in esame.
- **LSA:** è il Link-State vero e proprio.



## 1.7 Sicurezza

Le comunicazioni tra i router possono essere autenticate in modo da permettere, solo ad alcuni router, di accedere ai pacchetti che viaggiano sulla rete.

Esistono *due tipi* di autenticazione disponibili per il protocollo OSPF:

- **“Simple” authentication:** in questo caso il campo authentication prevede una password di otto caratteri che va configurata su tutti i router interessati. L'amministratore può configurare una password diversa per ciascuna sotto-rete o collegamento punto-punto. Questa tecnica è molto semplice e più esposta a manomissioni.
- **“Cryptographic” authentication:** quando viene usata questa procedura, gli 8 bytes del campo Authentication vengono ridefiniti per contenere una chiave, un campo lunghezza e un numero di sequenza. Questa tecnica si basa su chiavi segrete condivise, configurate in ogni router. L'algoritmo standard è denominato *MD5* ; quando viene utilizzato questo algoritmo, il mittente calcola un checksum MD5 sulla concatenazione dei pacchetti OSPF, sulla chiave segreta e sul campo lunghezza, producendo un messaggio di sommario lungo 16 bytes. Questo messaggio viene aggiunto al pacchetto OSPF. Alla ricezione, il ricevente esegue lo stesso calcolo usando la chiave segreta e compara il risultato al sommario ricevuto: se i valori sono uguali il pacchetto viene preso in considerazione, altrimenti scartato.

## Capitolo 2: Cisco Packet Tracer

---



Il software utilizzato per la simulazione di una rete OSPF è *Packet Tracer*, sviluppato dalla Cisco. Questo software ha fini didattici in quanto è progettato per migliorare l'apprendimento del Networking.

Viene distribuito all'interno del programma Cisco Networking Academy e fu ideato da un team di programmatori ed ex studenti, sotto la guida di Dennis Frezzo di Cisco System.

Questo software migliora e finalizza l'apprendimento in quanto:

- Consente un'esperienza e una gestione panoramica della rete da parte del singolo utente.
- Simula e rende visibile in tempo reale qualsiasi modifica del funzionamento e della struttura logica di rete sottostante.

I protocolli supportati dall'applicativo, sono quelli presenti nella tabella sottostante:

Layer	Cisco Packet Tracer Supported Protocols
Application	• FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR VOIP, SSCP config and calls, ISR command support, Call Manager Express
Transport	• TCP and UDP, TCP Nagle Algorithm & IP Fragmentation, RTP
Network	• BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPSec, RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL, Zone-based policy firewall and Intrusion Protection System on the ISR, GRE VPN, IPSec VPN
Network Access/Interface	• Ethernet (802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP

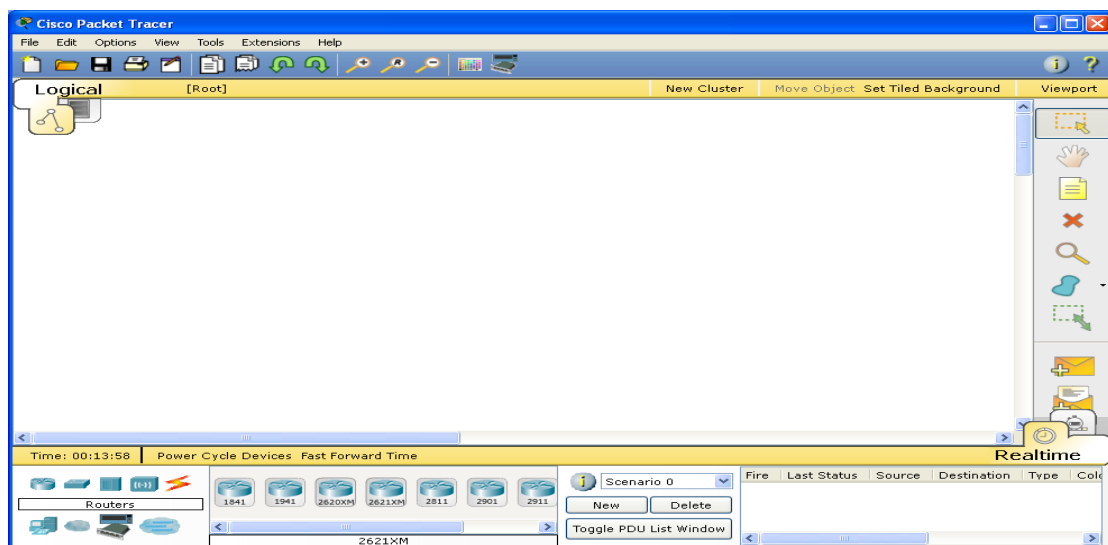


## 2.1 Presentazione





Dopo aver installato e avviato il programma, all'utente si presenterà un'interfaccia con un riquadro centrale principale dove sarà possibile costruire la propria rete; sul lato destro invece saranno presenti varie opzioni per poter modificare la rete e per poterla simulare.

In basso invece è possibile selezionare gli elementi da poter inserire all'interno del riquadro principale e avere le informazioni circa le simulazioni.

Di seguito la schermata appena descritta.



All'interno del riquadro principale verranno posizionati i vari elementi della rete; i principali sono:

-  **Router:** rappresenta un dispositivo elettronico che si occupa di instradare i pacchetti fra reti diverse.
-  **Switch:** rappresenta un dispositivo di rete che si occupa di commutazione a livello di collegamento del modello ISO/OSI.
-  **Hub:** rappresenta un dispositivo di rete che funge da nodo di smistamento-dati di una rete di comunicazione dati.
-  **End-System:** Indica un terminale collegato ad una rete informatica.

Di ciascuno dei precedenti elementi esistono vari sotto-modelli a seconda delle specifiche di simulazione.

Una volta posizionati i vari device, è possibile collegarli tra loro con diversi tipi di collegamenti come cavi coassiali, fibra ottica, fili in rame , eccetera.

Uno dei principali vantaggi di Packet Tracer, per i fini didattici, sta nel rappresentare in modo quasi realistico i vari device: infatti ne permette una visualizzazione pari a quella reale, sia a livello hardware che software.

## 2.2 Livello Hardware

A livello di rappresentazione di dispositivi hardware, prendendo ad esempio i routers che saranno i dispositivi maggiormente utilizzati, *Packet Tracer* mette a disposizione delle rappresentazioni con le varie slot, al cui interno sono inseriti più tipi di collegamenti con la possibilità di aggiungerne altri per una maggiore personalizzazione.

Di seguito un esempio:



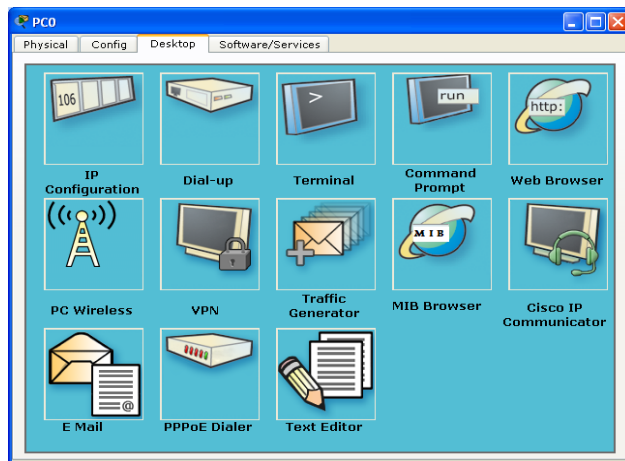
Come si può notare, sono disponibili diversi collegamenti per la simulazione, una slot per aggiungerne altri e un pulsante di ON-OFF per abilitare o meno il router.

## 2.3 Livello Software

A livello software, per i settaggi dei router, Packet Tracer mette a disposizione dell'utente una **CLI** (*Command Line Interface*) per simulare l'interfaccia *utente* a riga di comando del sistema operativo Cisco IOS, uguale a quella presente sui dispositivi Cisco. Con tale utility è possibile inserire i comandi per applicare un determinato protocollo o, più in generale, per fare altre operazioni di impostazioni sui router.

Per simulare l'uso dei vari end-system, viene rappresentata anche una schermata desktop semplificata nella quale è possibile selezionare alcune funzionalità tipiche di un end-system come la *scelta* e la *configurazione* degli indirizzi IP, l'*apertura* del Command Prompt, la possibilità di *inviare e-mail*, la facoltà di *navigare* su internet e altre funzionalità ancora.

Di seguito la schermata desktop:



## 2.4 Simulazioni

Per verificare il funzionamento di una rete è possibile selezionare, sulla barra destra, l'invio di un **PDU** (*Protocol Data Unit*), quindi un punto della rete, da cui far partire il messaggio, e un altro punto a cui si vuole far giungere il messaggio, infine si può attivare la simulazione.

Per avviare e quindi simulare il funzionamento di una rete, sono disponibili due modalità:

- **Real-time**
- **Simulation**

### 2.4.1 Real-Time

In questa modalità ciascun cambiamento, che viene effettuato sulla rete, ha effetto immediato e quindi è possibile analizzare in tempi molto brevi quali sono gli effetti delle modifiche apportate alla rete, a discapito di una non-completa visione dei processi che avvengono sulla rete stessa.

### 2.4.2 Simulation

In questa modalità è possibile analizzare e osservare quali sono gli effetti apportati da una modifica sulla rete, in quanto vengono rappresentati graficamente i pacchetti che viaggiano sulla rete per trasportare le informazioni. Questo tipo di simulazione è più lenta della precedente ma mostra in dettaglio il funzionamento.

Tramite l'uso di determinati filtri, è possibile osservare solo i pacchetti desiderati che viaggiano da un punto all'altro. E' possibile impostare l'esecuzione dei vari step della

simulazione in modalità *automatica* e *continuativa*, oppure in modalità *manuale guidata* dall'utente che decide quando passare al successivo step.

Nella modalità *simulation*, cliccando sui pacchetti che si vedranno viaggiare sulla rete, sarà possibile visionare le informazioni contenute e la loro strutturazione.

Per verificare il funzionamento della rete, da linea di comando, possiamo utilizzare due programmi che sono **Ping** e **Traceroute**.

## 2.5 PING

Il programma *PING* viene utilizzato per verificare la connettività verso un host e per misurare la latenza di trasmissione, in millisecondi: ciò viene fatto inviando uno o più messaggi ICMP (*Internet Control Message Protocol*).

PING invia un messaggio ICMP, di tipo 8 e codice 0 , verso l'host con cui vuole comunicare. L'host destinazione, vedendo la *echo request* , risponde con una *echo reply*, che è un messaggio ICMP di tipo 0 e codice 0.

Per eseguire questo comando, nel Command Prompt della schermata desktop basterà digitare il seguente comando:

```
ping 172.158.13.2
```

## 2.6 Traceroute

Il comando *Traceroute* invece viene utilizzato per rilevare il percorso tra due host, visualizzando tutti gli indirizzi IP degli host attraversati, fino a che il messaggio non giunge a destinazione.

Questo programma, per realizzare il proprio scopo, utilizza il campo **TTL** (*Time to live*) del pacchetto IP. Questo campo indica il numero di host che il pacchetto può attraversare prima di essere dichiarato scaduto, poiché, ad ogni attraversamento, il campo TTL viene decrementato di una unità. Quando tale campo diventa pari a 0, viene inviato un messaggio ICMP di errore al mittente ( nello specifico di tipo 11 e codice 0 ) , nel quale è indicato il router che lo ha generato.

Il Traceroute invia una serie di datagrammi IP verso la destinazione: il *primo* con TTL pari ad **uno**; quindi il primo router, constatando che il TTL è diventato 0, invierà un errore al mittente con il proprio indirizzo IP. A questo punto il mittente invierà nuovamente un altro datagramma IP, ma questa volta con TTL pari a **due**, e si ripeterà l'operazione precedente finché non si giungerà alla destinazione prefissata.

In questo modo sarà possibile conoscere gli indirizzi IP dei router attraversati. Per eseguire questo programma , basterà digitare nel Command Prompt il seguente comando:

```
tracert 172.158.13.2
```

## Capitolo 3: Simulazione

---

La simulazione, sarà suddivisa in *3 fasi* principali.

Una *prima fase* di strutturazione della rete e configurazione dei singoli dispositivi.

Una *seconda fase* di testing della rete nella normale operatività in cui tutti i collegamenti sono attivi.

La *terza* e ultima *fase* in cui si prevede un testing della rete nella condizione di link-fail tra 2 router.

### 3.1 Strutturazione della rete

Tramite il programma Cisco Packet Tracer, è stata disegnata la rete su cui effettuare la simulazione. La topologia della rete è formata da 4 endsystem (PC su cui girano i programmi), 4 router e 4 switch.

Successivamente al disegno della rete, i router sono stati rinominati utilizzando la Command Line Interface , nel seguente modo:

Router1:

```
Router>enable
Router#configure terminal
Router#hostname Router1
```

Router2:

```
Router>enable
Router#configure terminal
Router#hostname Router2
```

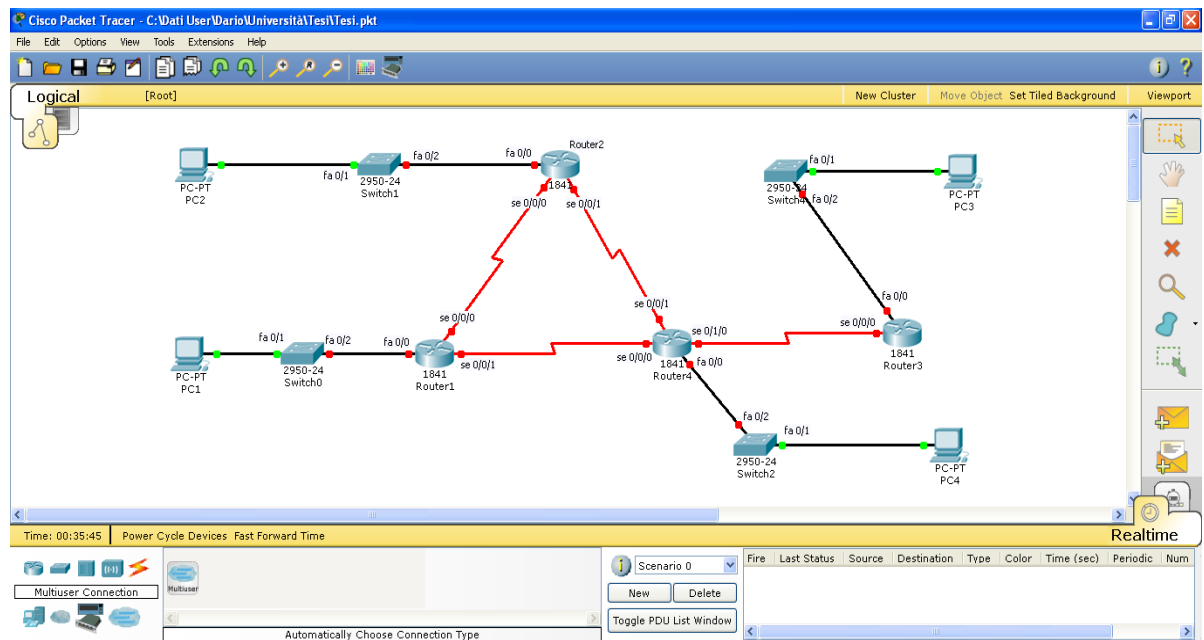
Router3:

```
Router>enable
Router#configure terminal
Router#hostname Router3
```

Router4:

```
Router>enable
Router#configure terminal
Router#hostname Router4
```

Terminata questa operazione, la rete si presenterà configurata come segue:



## 3.2 Configurazione indirizzi IP

Per ciascuna interfaccia coinvolta nella simulazione dei router e dei PC sono stati assegnati degli indirizzi IP secondo la seguente gerarchia: *rete/area/subnet/host*. Poiché gli elementi del sistema appartengono tutti alla stessa rete e area, i primi valori degli indirizzi IP coincidono tra loro.

L'assegnazione degli IP è suddivisa in 2 gruppi: una per i *Router* tramite la Command Line Interface, e l'altra, quella dei *PC*, tramite schermata desktop fornita dal simulatore.

Per configurare l'IP di ciascuna interfaccia, occorre attivare la modalità privilegiata tramite "enable", successivamente utilizzare "configure terminal" per specificare l'interfaccia su cui si vuole lavorare e infine assegnare l'indirizzo.

Una volta eseguite queste operazioni, tramite "no shutdown", quella determinata porta è stata resa attiva.

#### Router 1:

```
Router1>enable
Router1#configure terminal
Router1(config)#interface Serial0/0/0
Router1(config-if)#ip address 192.168.21.2 255.255.255.0
Router1(config-if)#no shutdown

Router1(config)#interface Serial0/0/1
Router1(config-if)#ip address 192.168.23.1 255.255.255.0
Router1(config-if)#no shutdown

Router1(config)#interface FastEthernet0/0
Router1(config-if)#ip address 192.168.11.1 255.255.255.0
Router1(config-if)#no shutdown
```

#### Router 2:

```
Router2>enable
Router2#configure terminal
Router2(config)#interface Serial0/0/0
Router2(config-if)#ip address 192.168.21.1 255.255.255.0
Router2(config-if)#no shutdown

Router2(config)#interface Serial0/0/1
Router2(config-if)#ip address 192.168.22.1 255.255.255.0
Router2 (config-if)#no shutdown

Router2(config)#interface FastEthernet0/0
Router2(config-if)#ip address 192.168.12.1 255.255.255.0
Router2(config-if)#no shutdown
```

#### Router3:

```
Router3>enable
Router3#configure terminal
Router3(config)#interface Serial0/0/0
Router3(config-if)#ip address 192.168.24.2 255.255.255.0
Router3(config-if)#no shutdown

Router3(config)#interface FastEthernet0/0
Router3(config-if)#ip address 192.168.13.1 255.255.255.0
```



```
Router3(config-if)#no shutdown
```

## Router4

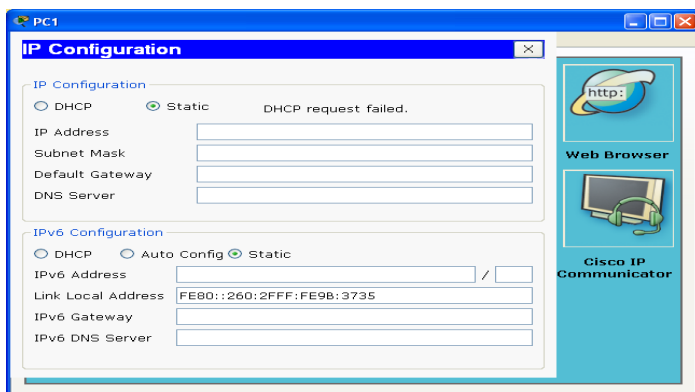
```
Router4>enable
Router4#configure terminal
Router4(config)#interface Serial0/0/0
Router4(config-if)#ip address 192.168.23.2 255.255.255.0
Router4(config-if)#no shutdown

Router4(config)#interface Serial0/0/1
Router4(config-if)#ip address 192.168.22.2 255.255.255.0
Router4(config-if)#no shutdown

Router4(config)#interface Serial0/1/0
Router4(config-if)#ip address 192.168.24.1 255.255.255.0
Router4(config-if)#no shutdown

Router4(config)#interface FastEthernet0/0
Router4(config-if)#ip address 192.168.14.1 255.255.255.0
Router4(config-if)#no shutdown
```

Per i PC la schermata desktop, fornita da Cisco Packet Tracer per la configurazione degli indirizzi IP, è la seguente.



Gli indirizzi IP, le Subnet Mask e i Default Gateway assegnati, sono i seguenti:

### PC1

```
IP Address: 192.168.11.2
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.11.1
```

## PC2

IP Address: 192.168.12.2  
Subnet Mask: 255.255.255.0  
Default Gateway: 192.168.12.1

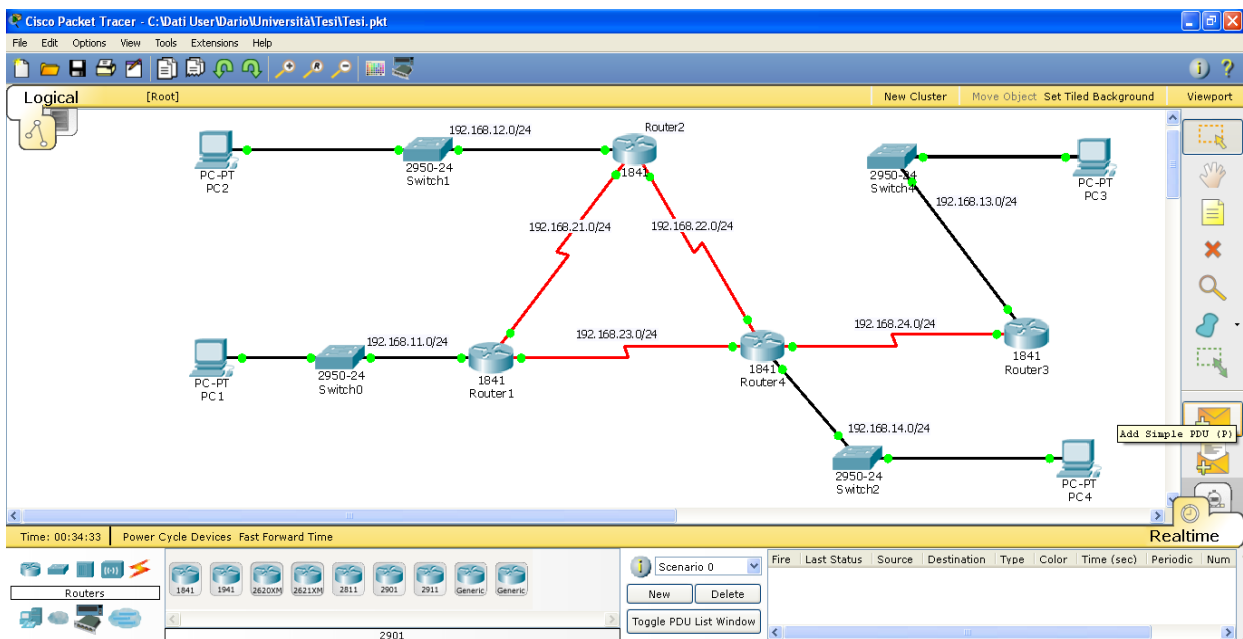
## PC3

IP Address: 192.168.13.2  
Subnet Mask: 255.255.255.0  
Default Gateway: 192.168.13.1

## PC4

IP Address: 192.168.14.2  
Subnet Mask: 255.255.255.0  
Default Gateway: 192.168.14.1

Al termine di queste configurazioni, la nostra rete con gli indirizzi IP si presenta come mostrato in figura:



## 3.3 Analisi Tabelle di Routing

Prima di eseguire i comandi per configurare OSPF, è interessante analizzare le tabelle di routing.

Eseguendo il comando “show ip route”, all’interno del Command Line Interface, avremo per ciascun router:

#### R1

```
Router1>show ip route
C    192.168.11.0/24 is directly connected, FastEthernet0/0
C    192.168.21.0/24 is directly connected, Serial0/0/0
C    192.168.23.0/24 is directly connected, Serial0/0/1
```

#### R2

```
Router2>show ip route
C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    192.168.21.0/24 is directly connected, Serial0/0/0
C    192.168.22.0/24 is directly connected, Serial0/0/1
```

#### R3

```
Router3>show ip route
C    192.168.13.0/24 is directly connected, FastEthernet0/0
C    192.168.24.0/24 is directly connected, Serial0/0/0
```

#### R4

```
Router3>show ip route
C    192.168.14.0/24 is directly connected, FastEthernet0/0
C    192.168.22.0/24 is directly connected, Serial0/0/1
C    192.168.23.0/24 is directly connected, Serial0/0/0
C    192.168.24.0/24 is directly connected, Serial0/1/0
```

Come si evince dalle tabelle di Routing, sono presenti solamente gli indirizzi IP delle sottoreti direttamente connesse al router: infatti, associato a ciascun indirizzo IP, abbiamo la lettera **C** che indica gli indirizzi degli host direttamente connessi.

In questa situazione, se volessimo inviare un ping dal PC1 al PC3 non sarebbe possibile in quanto, una volta che il messaggio di ping è giunto al Router 1, quest’ultimo non saprebbe dove inoltrarlo ulteriormente per farlo giungere al PC3. Per questo motivo risulta necessaria una gestione dell’instradamento di tipo *statico* o *dinamico*. Nel nostro caso sarà predisposto un instradamento dinamico tramite il protocollo OSPF.

## 3.4 Configurazione OSPF

Per ciascun router si procede alla configurazione del protocollo OSPF.

Si vanno a definire gli indirizzi IP e le aree di appartenenza degli host vicini.

Per queste operazioni, oltre ai soliti comandi “enable” e “configure terminal”, si utilizzeranno altri due comandi che sono:

- “router ospf ID”, che permette la configurazione con il protocollo OSPF (dove con ID si identifica il processo di routing all’interno del router).
- “network area” che definisce la sottorete su cui opera il protocollo OSPF con la relativa area di appartenenza.

#### Router 1:

```
Router1>enable
Router1#configure terminal
Router1(config)#router ospf 1
Router1(config-router)#network 192.168.11.2 255.255.255.0 area 0
Router1(config-router)#network 192.168.21.1 255.255.255.0 area 0
Router1(config-router)#network 192.168.23.2 255.255.255.0 area 0
Router1(config-router)#exit
Router1(config)#exit
```

#### Router 2:

```
Router2>enable
Router2#configure terminal
Router2(config)#router ospf 1
Router2(config-router)#network 192.168.12.2 255.255.255.0 area 0
Router2(config-router)#network 192.168.21.2 255.255.255.0 area 0
Router2(config-router)#network 192.168.22.2 255.255.255.0 area 0
Router2(config-router)#exit
Router2(config)#exit
```

#### Router 3:

```
Router3>enable
Router3#configure terminal
Router3(config)#router ospf 1
Router3(config-router)#network 192.168.13.2 255.255.255.0 area 0
Router3(config-router)#network 192.168.24.1 255.255.255.0 area 0
Router3(config-router)#exit
Router3(config)#exit
```

#### Router 4:

```
Router4>enable
Router4#configure terminal
Router4(config)#router ospf 1
Router4(config-router)#network 192.168.24.2 255.255.255.0 area 0
```

```

Router4(config-router)#network 192.168.14.2 255.255.255.0 area 0
Router4(config-router)#network 192.168.22.1 255.255.255.0 area 0
Router4(config-router)#network 192.168.23.1 255.255.255.0 area 0
Router4(config-router)#exit
Router4(config)#exit

```

### 3.4.1 Analisi tabelle di Routing

Una ulteriore analisi delle tabelle di routing ci consentirà di vedere come sono cambiate, dopo l'esecuzione del protocollo OSPF.

In questo caso l'analisi si concentrerà principalmente sulle tabella di routing dei Router 1 e Router 4 che sono quelli maggiormente sottoposti alla simulazione.

#### Router 1:

```

Router1>show ip router
C    192.168.11.0/24 is directly connected, FastEthernet0/0
C    192.168.23.0/24 is directly connected, Serial0/0/1
C    192.168.21.0/24 is directly connected, Serial0/0/0
O    192.168.12.0/24 [110/65] via 192.168.21.1, 00:19:36, Serial0/0/0
O    192.168.13.0/24 [110/129] via 192.168.23.2, 00:16:23, Serial0/0/1
O    192.168.14.0/24 [110/65] via 192.168.23.2, 00:16:23, Serial0/0/1
O    192.168.22.0/24 [110/128] via 192.168.21.1, 00:16:23, Serial0/0/0
O    192.168.22.0/24 [110/128] via 192.168.23.2, 00:16:23, Serial0/0/1
O    192.168.24.0/24 [110/128] via 192.168.23.2, 00:16:23, Serial0/0/1

```

#### Router 4:

```

Router4>show ip route
C    192.168.14.0/24 is directly connected, FastEthernet0/0
C    192.168.22.0/24 is directly connected, Serial0/0/1
C    192.168.23.0/24 is directly connected, Serial0/0/0
C    192.168.24.0/24 is directly connected, Serial0/1/0
O    192.168.11.0/24 [110/65] via 192.168.23.1, 00:21:53, Serial0/0/0
O    192.168.12.0/24 [110/65] via 192.168.22.1, 00:29:33, Serial0/0/1
O    192.168.13.0/24 [110/65] via 192.168.24.2, 00:29:33, Serial0/1/0
O    192.168.21.0/24 [110/128] via 192.168.22.1, 00:21:53, Serial0/0/1
O    192.168.21.0/24 [110/128] via 192.168.23.1, 00:21:53, Serial0/0/0

```

Confrontando le tabelle di routing con quelle viste precedentemente, si nota subito che, oltre ai collegamenti diretti (identificati con la **C**), abbiamo gli indirizzi IP delle sottoreti non direttamente connesse (identificati con la **O**), le quali sono state aggiunte tramite l'utilizzo del protocollo OSPF.

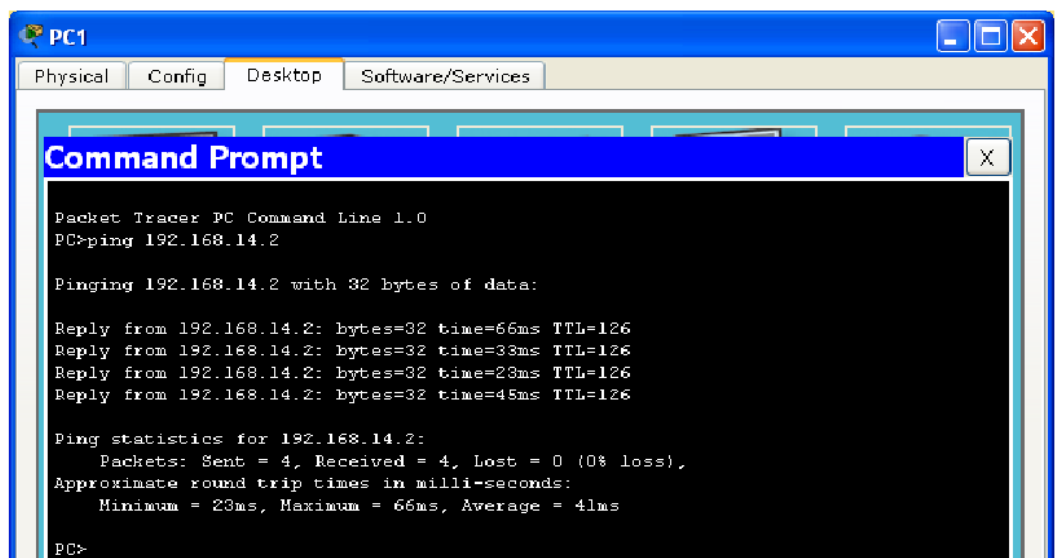
Per ciascuna riga identificata con la O, oltre all'indirizzo della sottorete da raggiungere, vi sono informazioni sulla *distanza* (date dai valori presenti tra parentesi quadra) e sull'*indirizzo IP del next hop*, necessario per poter giungere alla sottorete desiderata.

Da osservare che, sia per le sottoreti 192.168.22.0/24 del Router 1 che per quelle del 192.168.21.0/24 del Router 4, vi sono 2 percorsi equivalenti tra di loro in termini di distanza da percorrere e che, per questo motivo, ci vengono forniti entrambi.

### 3.5 Testing

Una volta configurata la rete, vogliamo testarne il funzionamento e analizzare i risultati che la simulazione ci fornisce. Per fare ciò eseguiamo un ping tra il PC1 e il PC4. A questo scopo avviamo il Command Prompt tramite la schermata desktop del PC1, al cui interno inseriremo il comando ping con l'indirizzo IP del PC con cui vogliamo verificare se la connessione è attiva.

In questo caso testiamo la connettività verso il PC4 che ha indirizzo IP 192.168.14.2. Di seguito viene mostrata la simulazione con gli annessi risultati.



```
PC1
Physical Config Desktop Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.14.2

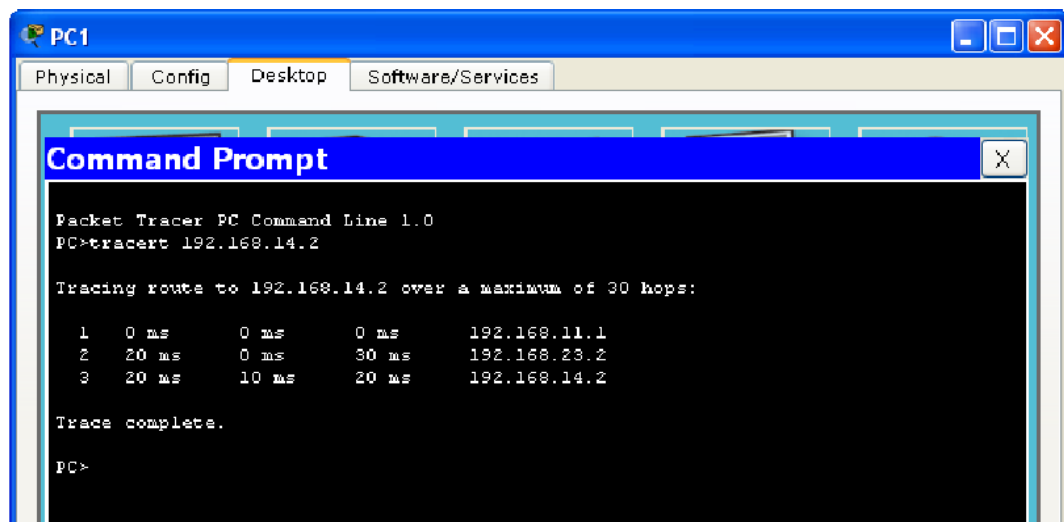
Pinging 192.168.14.2 with 32 bytes of data:

Reply from 192.168.14.2: bytes=32 time=66ms TTL=126
Reply from 192.168.14.2: bytes=32 time=33ms TTL=126
Reply from 192.168.14.2: bytes=32 time=23ms TTL=126
Reply from 192.168.14.2: bytes=32 time=45ms TTL=126

Ping statistics for 192.168.14.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 23ms, Maximum = 66ms, Average = 41ms
PC>
```

Da un'analisi dei valori forniti dal ping, possiamo vedere che, su 4 pacchetti inviati verso il PC4, ne sono stati ricevuti 4, (quindi la percentuale di pacchetti persi è pari a 0%), e che il tempo medio di trasmissione tra le 2 parti è di 41 millisecondi.

A questo punto vogliamo conoscere il percorso effettuato dai pacchetti per poter constatare se il protocollo OSPF ha fornito correttamente il percorso più breve per far comunicare i 2 end-system. A tale scopo, usiamo il comando “tracert” con l’IP del PC4 nel Command Prompt



The screenshot shows a Packet Tracer PC window for PC1. The 'Desktop' tab is active, displaying a 'Command Prompt' window. The command prompt shows the execution of the 'tracert' command to reach the destination IP 192.168.14.2. The output indicates a successful path through three hops: the source PC (192.168.11.1), a router (192.168.23.2), and the destination PC (192.168.14.2).

```
Packet Tracer PC Command Line 1.0
PC>tracert 192.168.14.2

Tracing route to 192.168.14.2 over a maximum of 30 hops:

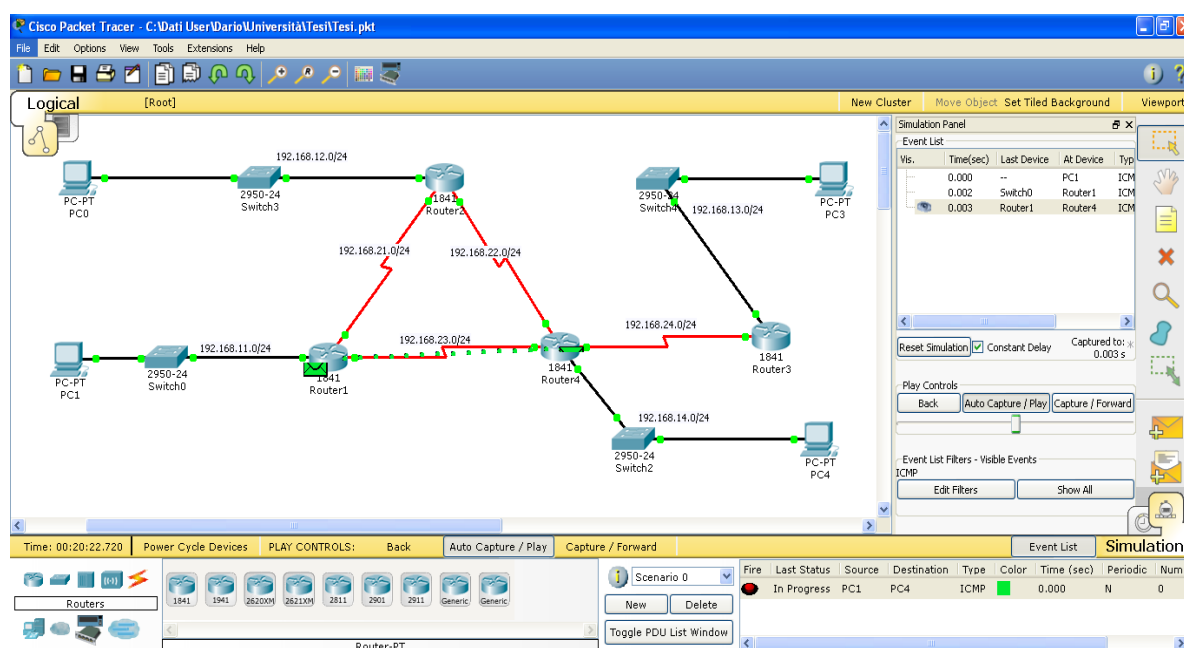
  0  0 ms    0 ms    0 ms    192.168.11.1
  1  20 ms   0 ms   30 ms   192.168.23.2
  2  20 ms   10 ms  20 ms   192.168.14.2

Trace complete.

PC>
```

Quindi possiamo vedere, tramite gli indirizzi IP, che il nostro pacchetto è passato prima per il Router 1, poi per il Router 4 ed infine è giunto al PC4: quindi ha seguito correttamente il percorso più breve.

Per constatare graficamente ciò, abbiamo usato la modalità Simulation in Packet Tracer e il risultato è il seguente.



Come da figura, il pacchetto sta passando per la sottorete 192.168.23.0/24, che rappresenta il percorso più breve.

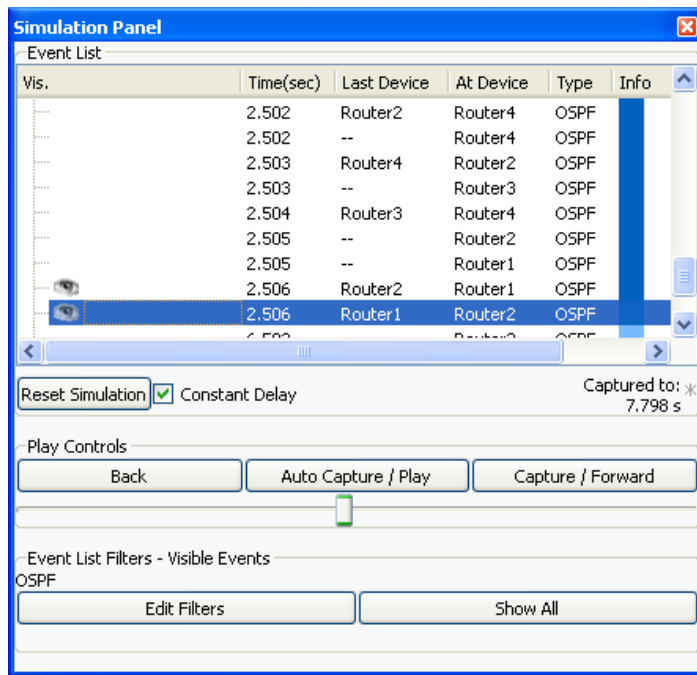
### 3.6 Link Fail

Supponiamo ora che, per un malfunzionamento oppure per una disconnessione intenzionale, il collegamento tra il Router 1 e il Router 4 non sia più attivo. Vogliamo quindi analizzare come si comporta in questo caso il protocollo OSPF per gestire i nuovi percorsi di routing. A tal proposito, tramite la Comand Line Interface del Router 4, eseguiamo il comando “shutdown” sull’interfaccia di interesse, come di seguito.

```
Router4>enable
Router4#configure terminal
Router4(config)#interface Serial0/0/0
Router4(config-if)#shutdown
Router4(config-if)#exit
```

Tramite l’uso della modalità *Simulation*, possiamo conoscere i tempi che sono stati necessari ai pacchetti OSPF per aggiornare le tabelle di Routing:





Dalla figura si evince che dopo 2.506 secondi il Router 1 ha ricevuto le informazioni necessarie per aggiornare le proprie tabelle di routing.

Vediamo ora come sono variate le tabelle di routing a seguito di questo cambiamento nella rete. Come prima usiamo il comando “show ip route”, sui Router 1 e Router 4 :

**Router1:**

```
Router1>show ip route
C    192.168.11.0/24 is directly connected, FastEthernet0/0
C    192.168.21.0/24 is directly connected, Serial0/0/0
O    192.168.12.0/24 [110/65] via 192.168.21.1, 00:03:59, Serial0/0/0
O    192.168.13.0/24 [110/193] via 192.168.21.1, 00:03:59, Serial0/0/0
O    192.168.14.0/24 [110/129] via 192.168.21.1, 00:03:59, Serial0/0/0
O    192.168.22.0/24 [110/128] via 192.168.21.1, 00:03:59, Serial0/0/0
O    192.168.23.0/24 [110/192] via 192.168.21.1, 00:03:59, Serial0/0/0
O    192.168.24.0/24 [110/192] via 192.168.21.1, 00:03:59, Serial0/0/0
```

**Router4:**

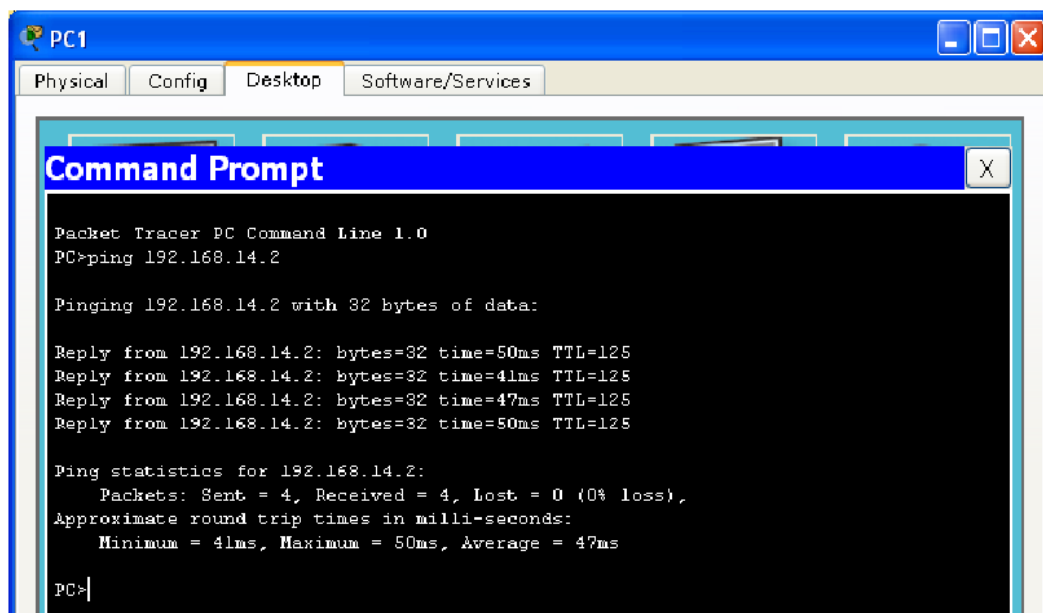
```
Router4>show ip route
C    192.168.14.0/24 is directly connected, FastEthernet0/0
C    192.168.22.0/24 is directly connected, Serial0/0/1
C    192.168.24.0/24 is directly connected, Serial0/1/0
O    192.168.11.0/24 [110/129] via 192.168.22.1, 00:05:51, Serial0/0/1
O    192.168.12.0/24 [110/65] via 192.168.22.1, 00:13:11, Serial0/0/1
O    192.168.13.0/24 [110/65] via 192.168.24.2, 00:13:11, Serial0/1/0
O    192.168.21.0/24 [110/128] via 192.168.22.1, 00:05:51, Serial0/0/1
```

Confrontando le tabelle di routing con le precedenti, si può vedere che i collegamenti che prima passavano per la sottorete 192.168.23.0/24, non sono più presenti, in quanto la linea non è più attiva. Quindi per giungere a quelle destinazioni, che prevedevano il passaggio per la linea ora disconnessa, sono stati creati percorsi alternativi per comunicare con la sottorete 192.168.14.0/24 tramite il Router 1.

Il nuovo percorso prevede come next hop 192.168.21.1, per cui le tabelle di routing sono state aggiornate.

Eseguiamo come prima, dei test per verificare la connettività e i nuovi percorsi che seguiranno i pacchetti.

Per prima cosa eseguiamo un ping tra il PC1 e il PC4 come fatto precedentemente.



```
PC1
Physical Config Desktop Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.14.2

Pinging 192.168.14.2 with 32 bytes of data:

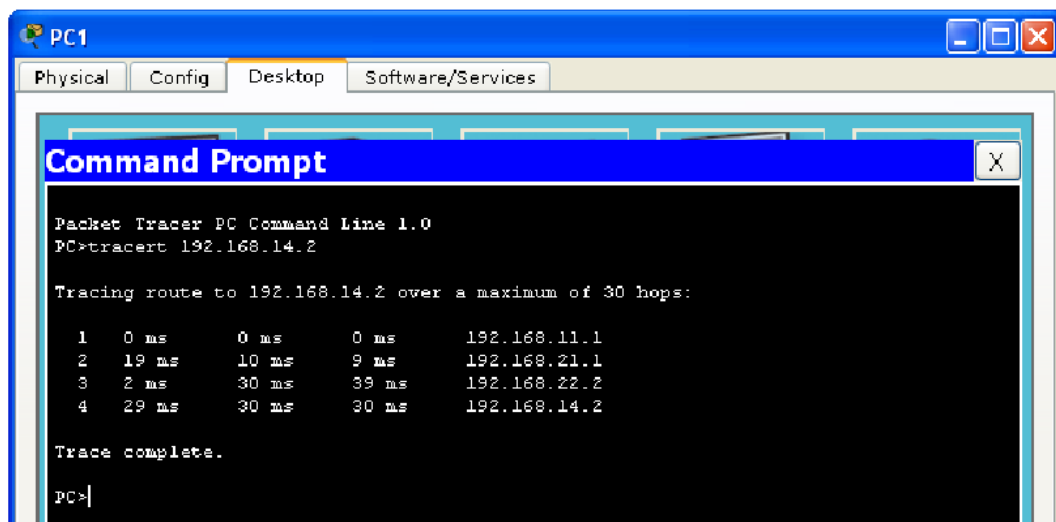
Reply from 192.168.14.2: bytes=32 time=50ms TTL=125
Reply from 192.168.14.2: bytes=32 time=41ms TTL=125
Reply from 192.168.14.2: bytes=32 time=47ms TTL=125
Reply from 192.168.14.2: bytes=32 time=50ms TTL=125

Ping statistics for 192.168.14.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 41ms, Maximum = 50ms, Average = 47ms

PC>
```

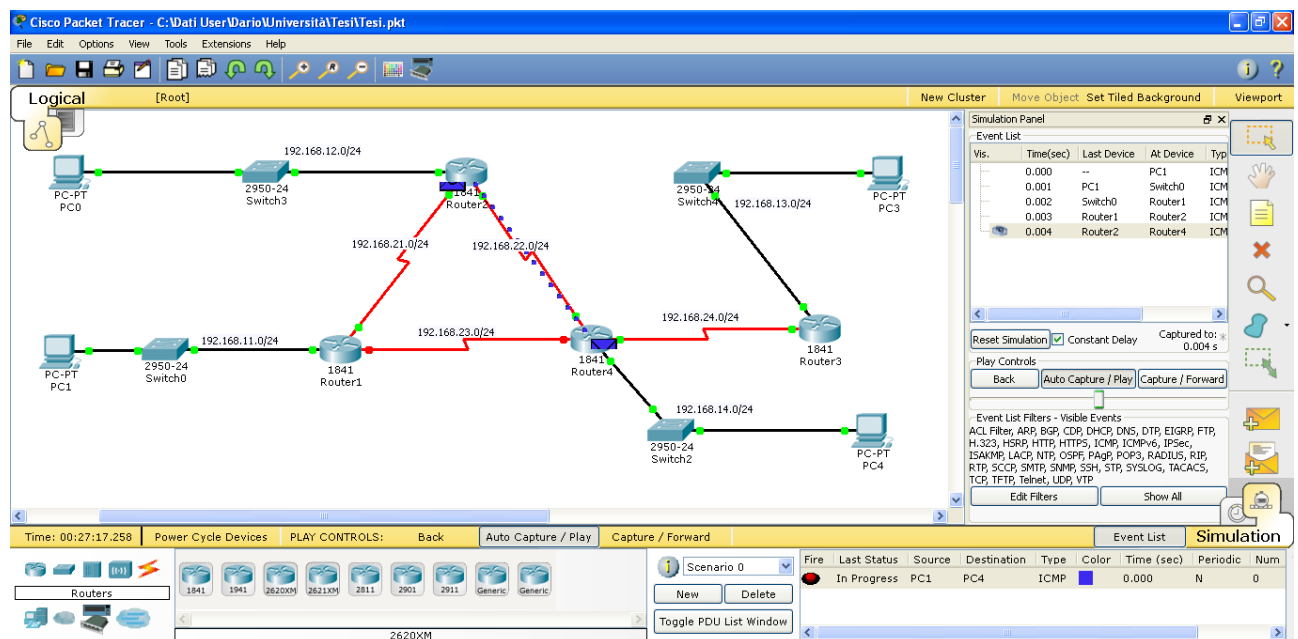
Il ping ha dato esito positivo in quanto, su 4 pacchetti che sono stati inviati, 4 ne sono stati ricevuti. Quindi un nuovo percorso tra i 2 PC è stato reso attivo.

Vediamo ora tramite il comando “tracert” quale è il nuovo percorso seguito dai pacchetti.



Il nuovo percorso prevede il passaggio prima per il Router 1, poi per il Router 2, successivamente per il Router 4, per poi arrivare al PC4.

Tramite l'opzione *Simulation* possiamo vedere, come prima, graficamente il percorso seguito dal pacchetto che è raffigurato di sotto.



Dalla figura si può vedere che il pacchetto ha seguito un nuovo percorso per superare il tratto dove si è avuto il Link-Fail.

## Cocclusioni

---

Tramite questa semplice simulazione si è riusciti ad osservare il funzionamento dell'protocollo OSPF.

Infatti tramite l'uso di un Link-Fail si è visto che il protocollo è riuscito a garantire un servizio di instradamento ai vari device della rete.

Abbiamo inoltre visto che, per giungere in un determinato punto della rete, vi sono vari percorsi ma l'OSPF è riuscito a fornire sempre quello più breve. Infatti, nel caso in cui tutti i collegamenti erano funzionanti, per la comunicazione tra router 1 e 4 si è fornito un percorso che utilizzava 2 router, mentre nel caso di link-fail tre.

Da ciò si deduce che il protocollo OSPF si è comportato correttamente durante l'esecuzione.

L'uso del software Cisco Packet Tracer in questo elaborato è stato molto utile per capire meglio e approfondire determinati aspetti della rete che, senza un approccio pratico, sono difficilmente assimilabili.

## Bibliografia

---

- [1] James F. Kurose, Keith W. Ross, “Reti di calcolatori e Internet, Un approccio top-down” , Mondadori 2008.
- [2] Cisco Packet Tracer – Scenario 1, [https://www.youtube.com/watch?v=0-0u4\\_apHjM](https://www.youtube.com/watch?v=0-0u4_apHjM).
- [3] RFC 2328 – OSPF Version 2 , Aprile 1998