

Laboratorio: Design OSPF Scalabile (Single-Area vs. Multi-Area)

1. Obiettivo di Design

Questo laboratorio ha lo scopo di analizzare, testare e confrontare due diversi approcci di design per un routing domain OSPF:

- **Design Piatto (Single-Area):** Veloce da implementare, ma con notevoli problemi di scalabilità.
- **Design Gerarchico (Multi-Area):** Richiede pianificazione (design), ma garantisce stabilità, isolamento dei guasti e prestazioni ottimali.

L'obiettivo non è configurare OSPF, ma **capire l'impatto architetturale** delle Aree OSPF sulla Link-State Database (LSDB), sulla CPU dei router e sulla propagazione dei guasti.

2. Prerequisiti

- **Piattaforma:** PNetLab
- **Immagine:** Cisco IOL (es. i86bi_linux-l3-adventerprisek9-ms.bin)
- **Router:** 9

3. Piano di Cablaggio (IOL)

Dispositivo A	Interfaccia A	Dispositivo B	Interfaccia B	Area / Scopo
Backbone				Area 0
R1	Eth0/1	R2	Eth0/1	R1 <-> R2
R1	Eth0/2	R3	Eth0/1	R1 <-> R3
R2	Eth0/2	R3	Eth0/2	R2 <-> R3
Area 1				Area 1
R2	Eth0/3	R4	Eth0/1	ABR <-> Area 1
R4	Eth0/2	R5	Eth0/1	R4 <-> R5
R4	Eth0/3	R6	Eth0/1	R4 <-> R6
R5	Eth0/2	R6	Eth0/2	R5 <-> R6
Area 2				Area 2
R2	Eth0/0	R7	Eth0/1	ABR <-> Area 2
R7	Eth0/2	R8	Eth0/1	R7 <-> R8
R7	Eth0/3	R9	Eth0/1	R7 <-> R9
R8	Eth0/2	R9	Eth0/2	R8 <-> R9

4. Tabella Indirizzamento IP (IOL)

Dispositivo	Interfaccia	Indirizzo IP	Subnet Mask	Note / Link a
R1 (Area 0)	Loopback0	192.168.0.1	255.255.255.255	
	Eth0/1	10.0.12.1	255.255.255.0	R2
	Eth0/2	10.0.13.1	255.255.255.0	R3
---	---	---	---	---
R2 (ABR)	Loopback0	192.168.0.2	255.255.255.255	
	Eth0/0	10.2.27.2	255.255.255.0	R7 (Area 2)
	Eth0/1	10.0.12.2	255.255.255.0	R1 (Area 0)
	Eth0/2	10.0.23.2	255.255.255.0	R3 (Area 0)
	Eth0/3	10.1.24.2	255.255.255.0	R4 (Area 1)
---	---	---	---	---
R3 (Area 0)	Loopback0	192.168.0.3	255.255.255.255	
	Eth0/1	10.0.13.3	255.255.255.0	R1
	Eth0/2	10.0.23.3	255.255.255.0	R2
---	---	---	---	---
R4 (Area 1)	Loopback0	192.168.0.4	255.255.255.255	

			5	
	Eth0/1	10.1.24.4	255.255.255.0	R2
	Eth0/2	10.1.45.4	255.255.255.0	R5
	Eth0/3	10.1.46.4	255.255.255.0	R6
---	---	---	---	---
R5 (Area 1)	Loopback0	192.168.0.5	255.255.255.255	
	Eth0/1	10.1.45.5	255.255.255.0	R4
	Eth0/2	10.1.56.5	255.255.255.0	R6
---	---	---	---	---
R6 (Area 1)	Loopback0	192.168.0.6	255.255.255.255	
	Eth0/1	10.1.46.6	255.255.255.0	R4
	Eth0/2	10.1.56.6	255.255.255.0	R5
---	---	---	---	---
R7 (Area 2)	Loopback0	192.168.0.7	255.255.255.255	
	Eth0/1	10.2.27.7	255.255.255.0	R2
	Eth0/2	10.2.78.7	255.255.255.0	R8
	Eth0/3	10.2.79.7	255.255.255.0	R9

---	---	---	---	---
R8 (Area 2)	Loopback0	192.168.0.8	255.255.255.255	
	Eth0/1	10.2.78.8	255.255.255.0	R7
	Eth0/2	10.2.89.8	255.255.255.0	R9
---	---	---	---	---
R9 (Area 2)	Loopback0	192.168.0.9	255.255.255.255	
	Eth0/1	10.2.79.9	255.255.255.0	R7
	Eth0/2	10.2.89.9	255.255.255.0	R8

5. Diagramma di Rete (Codice Mermaid)

Snippet di codice Mermaid

```
graph TD
    subgraph "Area 0 (Backbone)"
        R1(R1) --- |"Eth0/1 - Eth0/1"| R2(R2 - ABR)
        R1(R1) --- |"Eth0/2 - Eth0/1"| R3(R3)
        R3(R3) --- |"Eth0/2 - Eth0/2"| R2(R2 - ABR)
    end

    subgraph "Area 1 (Uffici)"
        R2(R2 - ABR) --- |"Eth0/3 - Eth0/1"| R4(R4)
        R4(R4) --- |"Eth0/2 - Eth0/1"| R5(R5)
        R4(R4) --- |"Eth0/3 - Eth0/1"| R6(R6)
        R5(R5) --- |"Eth0/2 - Eth0/2"| R6(R6)
    end

    subgraph "Area 2 (Data Center)"
        R2(R2 - ABR) --- |"Eth0/0 - Eth0/1"| R7(R7)
        R7(R7) --- |"Eth0/2 - Eth0/1"| R8(R8)
        R7(R7) --- |"Eth0/3 - Eth0/1"| R9(R9)
        R8(R8) --- |"Eth0/2 - Eth0/2"| R9(R9)
    end

    %% Stile per evidenziare l'ABR
    style R2 fill:#f9d,stroke:#333,stroke-width:2px
```

6. Fase 1: Il Design "Piatto" (Tutto in Area 0)

In questa fase, configuriamo tutti i router per appartenere alla stessa Area 0, simulando una rete cresciuta "organicamente" senza un design gerarchico.

6.1. Configurazione (Template)

Applica una configurazione simile a questa su *tutti i 9 router*, adattando gli IP e le interfacce.

Template per R1:

Cisco CLI

```
hostname R1
!
interface Loopback0
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet0/1
 description LINK-TO-R2
 ip address 10.0.12.1 255.255.255.0
 no shutdown
!
interface Ethernet0/2
 description LINK-TO-R3
 ip address 10.0.13.1 255.255.255.0
 no shutdown
!
router ospf 1
 ! Abilita OSPF su tutte le interfacce configurate
 network 0.0.0.0 255.255.255.255 area 0
!
```

Template per R2 (Router Centrale):

Cisco CLI

```
hostname R2
!
interface Loopback0
 ip address 192.168.0.2 255.255.255.255
!
interface Ethernet0/0
 description LINK-TO-R7-AREA2
 ip address 10.2.27.2 255.255.255.0
 no shutdown
!
interface Ethernet0/1
 description LINK-TO-R1
 ip address 10.0.12.2 255.255.255.0
 no shutdown
!
interface Ethernet0/2
 description LINK-TO-R3
 ip address 10.0.23.2 255.255.255.0
 no shutdown
!
interface Ethernet0/3
 description LINK-TO-R4-AREA1
 ip address 10.1.24.2 255.255.255.0
 no shutdown
!
router ospf 1
 network 0.0.0.0 255.255.255.255 area 0
!
```


6.2. Analisi del Design (Piatto)

Una volta che tutti i router sono configurati e le adiacenze sono FULL, esegui queste verifiche.

1. Analisi Tabella di Routing (Su R5, Area 1):

```
R5# show ip route ospf
```

- **Osservazione:** Noterai che *tutte* le rotte OSPF (per i loopback e i link) sono marcate come "O" (Intra-Area). Non c'è distinzione gerarchica.

2. Analisi della LSDB (Su R5 e R8):

```
R5# show ip ospf database
```

```
R8# show ip ospf database
```

- **Osservazione:** Le due LSDB sono **assolutamente identiche**. Ogni router nella rete (R1, R5, R9...) ha una mappa topologica completa e dettagliata (LSA Type 1 e 2) dell'intera infrastruttura.
- **Domanda di Design:** È efficiente che R5 (Area 1) conosca i dettagli di ogni singolo link del Data Center (Area 2)?

6.3. Test di Impatto (Fault Isolation)

Questo è il test cruciale. Simuleremo un guasto in Area 2 e vedremo l'impatto sulla CPU di un router in Area 1.

1. Su R5 (Area 1), attiva il debug dell'SPF:

```
R5# debug ip ospf spf
```

2. Su R8 (Area 2), spegni un link:

```
R8(config)# interface Ethernet0/2  
R8(config-if)# description LINK-TO-R9  
R8(config-if)# shutdown
```

3. Osserva l'output su R5:

Vedrai immediatamente un log simile a questo:

```
*Nov  4 10:30:00.123: OSPF: Rcv LSUPD from 192.168.0.7 on Eth0/x,  
length 84  
*Nov  4 10:30:00.123: OSPF: SFP RUN: OLD STATE: 1, NEW STATE: 1  
*Nov  4 10:30:00.123: OSPF: SFP: Schedule SPF calculation...
```

- **Conclusione di Design (Fase 1):** In un design "piatto", qualsiasi instabilità (flap di link) ovunque nella rete forza *ogni singolo router* a ricalcolare il proprio albero SPF. Questo è un enorme spreco di CPU e rende la rete instabile e lenta a convergere su larga scala.

7. Fase 2: Il Design Gerarchico (Multi-Area)

Ora modifichiamo il design per introdurre la gerarchia, senza cambiare un singolo cavo. Trasformeremo R2 in un **Area Border Router (ABR)**.

7.1. Modifica Configurazione (Refactoring)

1. Su R2 (Il nuovo ABR):

Dobbiamo rimuovere il comando "catch-all" e specificare le aree per interfaccia.

```
hostname R2
!
router ospf 1
! Rimuove la vecchia regola "catch-all"
no network 0.0.0.0 255.255.255.255 area 0
!
! Interfacce assegnate all'Area 0 (Backbone)
network 192.168.0.2 0.0.0.0 area 0
network 10.0.12.2 0.0.0.0 area 0
network 10.0.23.2 0.0.0.0 area 0
!
! Interfaccia assegnata all'Area 1
network 10.1.24.2 0.0.0.0 area 1
!
! Interfaccia assegnata all'Area 2
network 10.2.27.2 0.0.0.0 area 2
!
```

2. Su R4, R5, R6 (Router Area 1):

Sposta tutto in Area 1.

```
hostname R4 (o R5, o R6)
!
router ospf 1
! Rimuove la vecchia regola "catch-all"
no network 0.0.0.0 255.255.255.255 area 0
```

```
!  
! Abilita OSPF su tutte le interfacce in Area 1  
network 0.0.0.0 255.255.255.255 area 1  
!
```

3. Su R7, R8, R9 (Router Area 2):
Sposta tutto in Area 2.

```
hostname R7 (o R8, o R9)  
!  
router ospf 1  
! Rimuove la vecchia regola "catch-all"  
no network 0.0.0.0 255.255.255.255 area 0  
!  
! Abilita OSPF su tutte le interfacce in Area 2  
network 0.0.0.0 255.255.255.255 area 2  
!
```

7.2. Analisi del Design (Gerarchico)

Dopo che le adiacenze tornano FULL:

1. Verifica ABR (Su R2):

```
R2# show ip ospf
```

- **Osservazione:** L'output ora indicherà: It is an area border router.

2. Analisi Tabella di Routing (Su R5, Area 1):

```
R5# show ip route ospf
```

- **Osservazione:** La tabella è cambiata!
 - Le rotte interne all'Area 1 (es. 192.168.0.4, 192.168.0.6) sono ancora "O".
 - Tutte le rotte per l'Area 0 e l'Area 2 (es. 192.168.0.8, 10.0.12.0/24) sono ora "O IA" (Inter-Area).
- **Domanda di Design:** Qual è il vantaggio di vedere le rotte esterne come "Inter-Area"? (Risposta: Sono riepiloghi. R5 ora sa solo "per andare in Area 2, vai a R2". Non conosce la topologia *interna* dell'Area 2).

3. Analisi della LSDB (Su R5):

```
R5# show ip ospf database
```

- **Osservazione:** La LSDB di R5 è ora **significativamente più piccola**.
 - Contiene LSA Type 1 e 2 (Router e Network) *solo* per l'Area 1.
 - *Non contiene* i LSA Type 1 e 2 dell'Area 0 e 2.
 - Contiene invece LSA Type 3 (Summary) generati da R2, che riepilogano le reti delle altre aree.

7.3. Test di Impatto (Fault Isolation)

Ripetiamo lo stesso test di prima.

1. **Su R5 (Area 1), assicurati che il debug sia attivo:**

```
R5# debug ip ospf spf
```

2. **Su R8 (Area 2), riattiva e spegni di nuovo il link:**

```
R8(config)# interface Ethernet0/2
R8(config-if)# no shutdown
R8(config-if)# shutdown
```

3. Osserva l'output su R5:

- **Osservazione: NON VEDRAI NESSUN RICALCOLO SPF!**
 - R5 riceverà un LSA Type 3 (Summary) aggiornato da R2, ma poiché la sua topologia interna (LSA 1/2) non è cambiata, e il suo next-hop (R2) è ancora valido, non ha bisogno di rieseguire l'algoritmo SPF.
-

8. Conclusioni di Design

- **Single-Area (Piatto):** Facile da implementare, ma **non scala**. Ogni modifica topologica impatta la CPU di ogni router, creando una rete "nervosa" e instabile. La LSDB cresce esponenzialmente.
- **Multi-Area (Gerarchico):** Richiede pianificazione per definire i confini delle aree (ABR). **Isola i guasti**, riduce la dimensione della LSDB e della tabella di routing, e limita l'utilizzo della CPU solo ai router nell'area interessata dal guasto. Questo è un design **stabile e scalabile**.

Network Diagram (with Mermaid)

