

1. Si dispone di un'insieme di indirizzi con netmask a 22 bit, dato che un blocco di indirizzi di classe C ha netmask a 24 bit abbiamo impiegato questi due bit di differenza (bit di supernetting) per individuare i 4 blocchi di indirizzi di classe C corrispondenti [198.10.96.X/24, 198.10.97.X/24, 198.10.98.X/24, 198.10.99.X/24]. Ottenuti rispettivamente impostando i penultimi due bit del 3° ottetto con le configurazioni [00, 01, 10, 11].

Con l'idea di effettuare SuperNetting su questi blocchi, se necessario, abbiamo realizzato l'assegnazione dei prefissi alle varie LAN applicando la tecnica di subnetting.

LAN	Prefix	NetMask	BroadCast Address	#Indirizzi Non utilizzati
LAN A	198.10.99.0	255.255.255.0	198.10.99.255	
LAN B	198.10.98.64	255.255.255.192	198.10.98.128	
LAN C	198.10.98.128	255.255.255.128	198.10.98.255	
LAN D	198.10.97.0	255.255.255.0	198.10.96.255	
LAN E	198.10.96.192	255.255.255.224	198.10.97.224	
LAN F	198.10.96.224	255.255.255.240	198.10.97.240	
LAN G	198.10.98.32	255.255.255.224	198.10.98.64	
LAN H	198.10.98.0	255.255.255.252	198.10.98.3	
LAN I	198.10.96.248	255.255.255.252	198.10.97.252	
LAN L	198.10.96.240	255.255.255.252	198.10.97.244	
LAN M (Emula Internet)	198.10.100.0	255.255.255.252	198.10.100.4f	

Per raggiungere l'obiettivo di minimizzare il numero di regole di instradamento nel router r1 abbiamo impiegato i blocchi .99 e .98 sul branch di sinistra (da r2 a seguire) in quanto condividono il 23° bit a 1 e ciò consente di ridurre di uno il numero di regole di instradamento su r1, scrivendo una sola regola con netmask di destinazione a 23 bit. La stessa logica è stata applicata sul branch di destra (da r3 a seguire) in quanto i blocchi .96 e .97 condividono il 23° bit a 0.

Inoltre abbiamo effettuato un'assegnazione contigua di indirizzi alle varie LAN in modo da poter (nei limiti del possibile) riutilizzare o rendere disponibili gli indirizzi al variare della configurazione proposta lasciando meno "buchi" possibili.

[TODO: ADD BLOCKS OF ADDRESSES AND TREE DIVISION OF SUBNETS]

2. Le tabelle sono l'output del comando qui indicato, eseguito nei rispettivi router.

```
$ route -n
```

Tabella di routing di r0:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
198.10.96.0	198.10.96.242	255.255.254.0	UG	0	0	0	eth0
198.10.96.240	0.0.0.0	255.255.255.252	U	0	0	0	eth0
198.10.98.0	198.10.96.242	255.255.254.0	UG	0	0	0	eth0
198.10.100.0	0.0.0.0	255.255.255.252	U	0	0	0	eth1

Tabella di routing di r1:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	198.10.96.241	0.0.0.0	UG	0	0	0	eth2
198.10.96.0	198.10.96.250	255.255.254.0	UG	0	0	0	eth1
198.10.96.240	0.0.0.0	255.255.255.252	U	0	0	0	eth2
198.10.96.248	0.0.0.0	255.255.255.252	U	0	0	0	eth1
198.10.98.0	0.0.0.0	255.255.255.252	U	0	0	0	eth0
198.10.98.0	198.10.98.2	255.255.254.0	UG	0	0	0	eth0

Tabella di routing di r2:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	198.10.98.1	0.0.0.0	UG	0	0	0	eth2
198.10.98.0	0.0.0.0	255.255.255.252	U	0	0	0	eth2
198.10.98.32	0.0.0.0	255.255.255.224	U	0	0	0	eth1
198.10.98.64	198.10.98.35	255.255.255.192	UG	0	0	0	eth1
198.10.98.128	198.10.98.35	255.255.255.128	UG	0	0	0	eth1
198.10.99.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

Tabella di routing di r3:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	198.10.96.249	0.0.0.0	UG	0	0	0	eth2
198.10.96.192	0.0.0.0	255.255.255.224	U	0	0	0	eth1
198.10.96.224	0.0.0.0	255.255.255.240	U	0	0	0	eth0
198.10.96.248	0.0.0.0	255.255.255.252	U	0	0	0	eth2
198.10.97.0	198.10.96.227	255.255.255.0	UG	0	0	0	eth0

Tabella di routing di r4:

```
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          198.10.98.34   0.0.0.0         UG      0      0      0 eth2
198.10.98.32     0.0.0.0        255.255.255.224 U        0      0      0 eth2
198.10.98.64     0.0.0.0        255.255.255.192 U        0      0      0 eth0
198.10.98.128    0.0.0.0        255.255.255.128 U        0      0      0 eth1
```

Tabella di routing di r5:

```
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          198.10.96.194  0.0.0.0         UG      0      0      0 eth2
198.10.96.192    0.0.0.0        255.255.255.224 U        0      0      0 eth2
198.10.96.224    0.0.0.0        255.255.255.240 U        0      0      0 eth1
198.10.97.0      0.0.0.0        255.255.255.0   U        0      0      0 eth0
```

- Output del traceroute eseguito dal router r5 ad un host della LAN M (che emula Internet), ottenuto dall'esecuzione del seguente comando:

```
$ traceroute 198.10.100.2
```

```
traceroute to 198.10.100.2 (198.10.100.2), 64 hops max
 1  198.10.96.194  0.041ms  0.040ms  0.040ms
 2  198.10.96.249  0.069ms  0.034ms  0.041ms
 3  198.10.96.241  0.034ms  0.062ms  0.044ms
 4  198.10.100.2  0.035ms  0.036ms  0.039ms
```

La regola di instradamento inserita nel router r5 è una regola di instradamento di default che vincola il traffico diretto verso la nostra LAN M a passare per l'interfaccia "eth1" del router r3.

```
route add default gw 198.10.96.194 dev eth2 #for external ip's and internet
```

- Consideriamo la seguente configurazione di rete di h1 e r2 ottenuta tramite il comando:

```
$ ifconfig
```

Per h1 (omessa Loopback interface):

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 198.10.99.1 netmask 255.255.255.0 broadcast 198.10.99.255
    ether ba:58:9d:46:10:0a txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 1436 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

E per r2 (omessa interfaccia eth2 e lo) :

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 198.10.99.2 netmask 255.255.255.0 broadcast 198.10.99.255
    ether 1a:10:8e:a7:04:2d txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 1436 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 198.10.98.34 netmask 255.255.255.224 broadcast 198.10.98.63
    ether 66:5b:55:7f:06:99 txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 1436 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Effettuiamo poi un ping tra h1 e h2 attraverso il comando (eseguito su h1). :

```
$ ping 198.10.98.65
```

Osservando che le tabelle arp di h1 e r2 sono inizialmente vuote, catturiamo poi il traffico in r2 attraverso il comando:

```
$ tcpdump -i eth0 -w r2Cap.pcap
```

Riportiamo quindi i pacchetti catturati da Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::20ef:62ff:fea9:1644	ff02::2	ICMPv6	70	Router Solicitation from 22:ef:62:a9:16:44
2	20.457439	fe80::e842:6fff:fe54:abfa	ff02::2	ICMPv6	70	Router Solicitation from 0e:de:db:f7:d7:ac
3	25.682213	ba:58:9d:46:10:0a	Broadcast	ARP	42	Who has 198.10.99.2? Tell 198.10.99.1
4	25.682229	1a:10:8e:a7:04:2d	ba:58:9d:46:10:0a	ARP	42	198.10.99.2 is at 1a:10:8e:a7:04:2d
5	25.682266	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=1/256, ttl=6
6	25.682562	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=1/256, ttl=6
7	26.730056	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=2/512, ttl=6
8	26.730188	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=2/512, ttl=6
9	27.753569	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=3/768, ttl=6
10	27.753698	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=3/768, ttl=6
11	28.777375	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=4/1024, ttl=6
12	28.777678	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=4/1024, ttl=6
13	29.802031	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=5/1280, ttl=6
14	29.802160	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=5/1280, ttl=6
15	30.698641	1a:10:8e:a7:04:2d	ba:58:9d:46:10:0a	ARP	42	Who has 198.10.99.1? Tell 198.10.99.2
16	30.698830	ba:58:9d:46:10:0a	1a:10:8e:a7:04:2d	ARP	42	198.10.99.1 is at ba:58:9d:46:10:0a
17	30.826085	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=6/1536, ttl=6
18	30.826220	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=6/1536, ttl=6
19	31.849418	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=7/1792, ttl=6
20	31.849715	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=7/1792, ttl=6
21	32.875382	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=8/2048, ttl=6
22	32.875689	198.10.98.65	198.10.99.1	ICMP	98	Echo (ping) reply id=0x0035, seq=8/2048, ttl=6
23	33.898809	198.10.99.1	198.10.98.65	ICMP	98	Echo (ping) request id=0x0035, seq=9/2304, ttl=6

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
 > Ethernet II, Src: 22:ef:62:a9:16:44 (22:ef:62:a9:16:44), Dst: IPv6mcast_02 (33:33:00:00:00:02)

In seguito alla comunicazione le ARP table di h1 e r2 risultano così configurate:

ARP table h1:

Address	HWtype	HWaddress	Flags Mask	Iface
198.10.99.2	ether	1a:10:8e:a7:04:2d	C	eth0

ARP table r2:

Address	HWtype	HWaddress	Flags Mask	Iface
198.10.99.1	ether	ba:58:9d:46:10:0a	C	eth0
198.10.98.35	ether	c2:2d:3b:3b:da:2a	C	eth1

Come si evince dal pacchetto numero 3, l'host h1 (con indirizzo MAC aa:bb:cc:dd:ee:0a) invia in broadcast ,a livello 2, una ARP request per venire a conoscenza dell'indirizzo MAC del suo gateway (r2) che risponde in unicast con una ARP reply (pacchetto numero 4 sulla cattura di Wireshark) contenente il suo indirizzo MAC (aa:bb:cc:dd:ee:2d). Una volta terminata la procedura, la ARP table di h1 conterrà un nuovo record con IP di r2 sull'interfaccia eth0 e indirzzo MAC di r2. In questo modo r2 potrà indirizzare correttamente l'Echo request, secondo le sue regole di instradamento ,e consegnare il pacchetto a h2 che risponderà (dopo aver terminato se necessario la sua procedura ARP) con una Echo reply.

5.

[TODO: add graph on OSI stack]

6.