Relazione 1

Advanced Ping

Mattia Chiarle 269969, Michele Ferrero 268542, Gabriele Ferro 268510

Gruppo 22 Anno 2021/2022

Configurazione della rete

La configurazione di rete generalmente utilizzata risulta essere la seguente. Eventuali modifiche vengono segnalate nei punti corrispondenti.

Indirizzo IP H1	172.16.22.1/26
Indirizzo IP H2	172.16.22.2/26
Indirizzo IP H3	172.16.22.3/26
Subnet mask	255.255.255.192
Network address	172.16.22.0

1 - Ping broadcast e al network address

1.1 - Ping broadcast

Il primo test consiste nell'utilizzare il comando ping in broadcast e, tramite l'utilizzo di wireshark, visualizzare i pacchetti scambiati tra i vari host.
Per effettuare il comando bisogna aggiungere il parametro '-b' in modo da permettere l'invio broadcast delle ICMP echo request.

```
ping -b 172.16.22.63          (Effettuato da H2)
```

L'host che effettua il comando (in questo caso H2) non invia una arp request ai vari host in quanto non è necessario: il MAC broadcast è infatti un indirizzo noto ovvero FF:FF:FF:FF:FF:FF.
Tutti gli altri host della subnet, prima di rispondere con la ICMP echo reply, effettuano la richiesta dell'indirizzo MAC della sorgente del ping tramite ARP. Infine ogni host risponde alla ICMP echo request con una ICMP echo reply indirizzata ad H2.

```
laboratorio@laboratorio:~/Desktop$ ping -b 172.16.22.63
WARNING: pinging broadcast address
PING 172.16.22.63 (172.16.22.63) 56(84) bytes of data.
64 bytes from 172.16.22.2: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 172.16.22.3: icmp_seq=1 ttl=64 time=0.366 ms (DUP!)
64 bytes from 172.16.22.1: icmp_seq=1 ttl=64 time=0.367 ms (DUP!)
64 bytes from 172.16.22.2: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 172.16.22.3: icmp_seq=2 ttl=64 time=0.360 ms (DUP!)
64 bytes from 172.16.22.1: icmp_seq=2 ttl=64 time=0.361 ms (DUP!)
64 bytes from 172.16.22.2: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 172.16.22.3: icmp_seq=3 ttl=64 time=0.355 ms (DUP!)
64 bytes from 172.16.22.1: icmp_seq=3 ttl=64 time=0.355 ms (DUP!)
^C
```

Nell'output del comando è possibile notare la presenza della stringa (DUP!) la quale indica che l'host si è accorto di aver ricevuto più echo reply relative alla stessa richiesta. La duplicazione della risposta è identificata tramite l'utilizzo dei campi icmp_seq e identifier del pacchetto ICMP. Essendo questi campi uguali, poiché i diversi host rispondono alla stessa echo request, l'host H2 li accetta e ne segnala la duplicazione.
Il tempo impiegato da ogni richiesta dipende da molti fattori, ma è possibile notare che H2 risponde sempre con un tempo inferiore. Questo è dovuto al fatto che la sua risposta parte non appena la richiesta raggiunge lo strato di rete, livello dove il protocollo IP si accorge della richiesta in broadcast e invia la risposta tramite l'interfaccia di loopback e non attraverso eth0.

1.2 - Ping al network address

Il secondo test richiede di effettuare un ping all'indirizzo della rete (Network address indicato in tabella).
L'output ottenuto non differisce da quello precedente per quanto riguarda i pacchetti inviati e ricevuti.

2 - Indirizzi duplicati

Indirizzo IP H1	172.16.22.1/26
Indirizzo IP H2	172.16.22.2/26
Indirizzo IP H3 (H1')	172.16.22.1/26

È stata configurata la rete nel seguente modo, imponendo a H3 lo stesso indirizzo IP di H1.

2.1 - Ping all'indirizzo duplicato

H2 prova a eseguire un ping di H1. Si può notare, tramite Wireshark, la presenza di una race condition sulla ARP request. H2 infatti non possiede il MAC associato all'indirizzo IP di H1 (in quanto l'ARP table è vuota prima di eseguire il comando ping) e, per poter inviare i pacchetti ICMP Request, provvede ad effettuare una ARP request broadcast. Tuttavia, sia H1 sia H1' pensano che il pacchetto sia rivolto a loro, in quanto il loro indirizzo IP coincide con quello presente nel payload del pacchetto ARP, e rispondono con una ARP reply. H2 riceverà quindi due risposte ad una sola domanda, accettando la prima (in quanto effettivamente si aspetta una sola ARP reply in risposta alla sua ARP request) e scartando la seconda, poiché è una risposta a una domanda mai fatta e quindi potenzialmente derivante da un errore. Il ping procede poi normalmente e non ci sono più possibili indeterminazioni, in quanto le successive ARP request e reply saranno in unicast (sono infatti volte solo a verificare che il MAC address ottenuto precedentemente sia ancora valido) e verranno di conseguenza ricevute solo dall'host coinvolto nel ping e non dall'altro. Le ARP table di H1 e H1' conterranno l'indirizzo MAC di H2 mentre l'ARP table di H2 conterrà solo l'indirizzo MAC del primo host ad aver mandato la ARP reply.

2.2 - Ping dagli indirizzi duplicati

Quando H1 e H1' provano a effettuare contemporaneamente un ping verso H2 si verifica un comportamento molto particolare. Infatti, inizialmente sia H1 sia H1' effettuano una ARP request verso H2; per questi due host l'operazione verrà eseguita correttamente (H2 risponderà a entrambi), mentre H2 avrà soltanto una entry nella ARP table, in quanto le due ARP request a lui pervenute risultano associate allo stesso IP. Di conseguenza, ogni volta che deve rispondere ad una ARP request si salverà il MAC address del mittente. Alla fine di queste operazioni preliminari l'ARP table di H2 conterrà il MAC dell'host che ha consegnato per secondo la sua ARP request.
Sia H1 sia H1' partiranno con l'invio dei pacchetti ICMP echo request; tuttavia, H2 sarà in grado di rispondere solo a uno dei due (in quanto nella ARP table è presente solo uno dei due MAC address). Questo porta a ricevere entrambe le ICMP echo reply sullo stesso host. Di conseguenza un host manda una richiesta e non riceve risposta, mentre l'altro ne manda una e ne riceve due.
È stato inoltre notato un altro comportamento particolare: se i campi identifier e icmp_seq dei pacchetti ICMP echo request inviati da H1 e H1' sono diversi, l'host che riceve due pacchetti visualizza solo la risposta alla propria richiesta sul terminale, in quanto probabilmente ICMP scarta i pacchetti basandosi su questi campi (capisce che non sono relativi al proprio ping e sono derivati da un errore). Tuttavia, se quei campi nei pacchetti dei due host risultano uguali, ICMP non riesce più a scartare i pacchetti dell'altro host e visualizza quindi sul terminale due risposte per ogni richiesta inviata (segnalando che per lui uno dei due pacchetti è duplicato inserendo DUP! dopo il secondo). Inoltre, se gli orologi interni dei due host non sono perfettamente allineati c'è anche il rischio che il calcolo del RTT relativo ai pacchetti dell'altro host dia un risultato negativo, il che è ovviamente impossibile. Per evitare di inserire un RTT negativo, che sarebbe ovviamente assurdo, ICMP lo imposta a 0 ms, segnalando però ogni volta l'errore.

Bene

```
64 bytes from 172.16.22.2: icmp_seq=13 ttl=64 time=0.000 ms
64 bytes from 172.16.22.2: icmp_seq=13 ttl=64 time=0.519 ms (DUP!)
ping: Warning: time of day goes back (-91581us), taking countermeasures
64 bytes from 172.16.22.2: icmp_seq=14 ttl=64 time=0.000 ms
64 bytes from 172.16.22.2: icmp_seq=14 ttl=64 time=0.494 ms (DUP!)
ping: Warning: time of day goes back (-91606us), taking countermeasures
```

Quella appena descritta è la situazione presente inizialmente. Tuttavia, come è stato notato anche nei precedenti laboratori, durante l'esecuzione del ping gli host tendono a effettuare delle ARP request in unicast per verificare che il

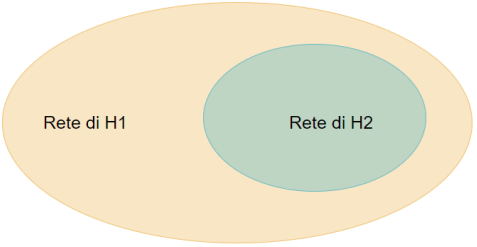
MAC address che stanno usando sia ancora valido. Se questa viene effettuata dall'host che attualmente sta ricevendo le ICMP echo reply o da H2 non si nota alcuna differenza (la ARP request conferma semplicemente quanto presente nella ARP table di H2 e dell'host attivo); se invece questa viene effettuata dall'host che sta attualmente inviando pacchetti senza ricevere alcuna risposta, alla ricezione della ARP request unicast H2 dovrà modificare la sua ARP table in modo da poter inviare correttamente la ARP reply. Tuttavia, la modifica della ARP table implica che, a partire da ora, H2 saprà come raggiungere solo quest'ultimo host.
In generale quindi H2 invierà le ICMP echo reply relative a entrambi i ping soltanto all'ultimo host che ha effettuato una ARP request. Ogni host avrà quindi alcuni momenti in cui riceverà due pacchetti per ogni richiesta e altri in cui invece non ne riceverà nessuno.

276	194.652575234	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=11/2816, ttl=64 (no respo…
277	195.676558307	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=12/3072, ttl=64 (no respo…
278	196.700558772	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=13/3328, ttl=64 (no respo…
279	197.724506331	f4:39:09:15:79:81	80:fa:5b:76:92:64	ARP	42 Who has 172.16.22.2? Tell 172.16.22.1
280	197.724579084	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=14/3584, ttl=64 (reply in…
281	197.724888582	80:fa:5b:76:92:64	f4:39:09:15:79:81	ARP	60 172.16.22.2 is at 80:fa:5b:76:92:64
282	197.724888802	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=14/3584, ttl=64 (request …
283	197.853619293	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000c, seq=14/3584, ttl=64
284	198.748580102	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=15/3840, ttl=64 (reply in…
285	198.748938309	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=15/3840, ttl=64 (request …
286	198.877611973	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000c, seq=15/3840, ttl=64
287	199.772560863	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=16/4096, ttl=64 (reply in…
288	199.772972596	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=16/4096, ttl=64 (request …
289	199.901625774	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000c, seq=16/4096, ttl=64
290	200.796563456	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=17/4352, ttl=64 (reply in…
291	200.796940812	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=17/4352, ttl=64 (request …
292	200.925619687	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000c, seq=17/4352, ttl=64
293	201.820557677	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=18/4608, ttl=64 (reply in…
294	201.820897785	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=18/4608, ttl=64 (request …
295	201.949642077	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000c, seq=18/4608, ttl=64
296	202.842219894	80:fa:5b:76:92:64	f4:39:09:15:79:81	ARP	60 Who has 172.16.22.1? Tell 172.16.22.2
297	202.842238318	f4:39:09:15:79:81	80:fa:5b:76:92:64	ARP	42 172.16.22.1 is at f4:39:09:15:79:81
298	202.844560050	172.16.22.1	172.16.22.2	ICMP	98 Echo (ping) request id=0x000d, seq=19/4864, ttl=64 (reply in…
299	202.844907773	172.16.22.2	172.16.22.1	ICMP	98 Echo (ping) reply id=0x000d, seq=19/4864, ttl=64 (request …

La foto riporta la situazione appena descritta: inizialmente l'host manda in continuazione ICMP echo request senza ricevere risposta. Non appena però invia una ARP request in unicast l'host inizia a ricevere le due ICMP echo reply, in quanto la ARP table di H2 è cambiata. Si nota inoltre come la ARP request in unicast inviata da H2 non causi nessuna modifica, in quanto l'host continua a ricevere le ICMP echo reply.
Nonostante non si riceva mai risposta, sul terminale non viene visualizzato nessun errore in quanto effettivamente (a parte l'errata configurazione della rete) non c'è nessun errore: ARP infatti non fallisce.

3 - Subnet diverse

Indirizzo IP H1	172.16.22.5/26
Indirizzo IP H2	172.16.22.2/30



3.1 - Ping a subnet interna

Quando l'host H1 effettua un ping all'host H2 si nota che H1, guardando la sua netmask, riconosce H2 come parte della sua rete locale e quindi invia una ARP request in broadcast. H2 riceve la ARP request da parte di H1 ma non risponde con una ARP reply, questo perché il protocollo ARP, oltre ad avere funzionalità di livello 2, ha anche funzionalità di livello 3 tra cui riconoscere il proprio IP e riconoscere gli host appartenenti alla propria rete. Questo permette ad ARP di scartare richieste di IP provenienti da altre subnet poiché violano il protocollo.
La ARP table di H1 viene aggiornata, infatti al suo interno si trova una entry incompleta con l'IP di H2, mentre la ARP table di H2 risulta vuota.

```
laboratorio@laboratorio:~/Desktop$ ping 172.16.22.2
PING 172.16.22.2 (172.16.22.2) 56(84) bytes of data.
From 172.16.22.5 icmp_seq=1 Destination Host Unreachable
From 172.16.22.5 icmp_seq=2 Destination Host Unreachable
From 172.16.22.5 icmp_seq=3 Destination Host Unreachable
```

3.2 - Ping a subnet esterna

Quando l'host H2 effettua un ping all'host H1 il risultato è differente, infatti H2 non vede H1 come parte della sua subnet e questo implica che a livello 3 IP riconosca che non vi è una entry nella routing table a cui inviare il pacchetto. La command line restituisce quindi "Network unreachable".

```
laboratorio@laboratorio:~/Desktop$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.22.0     0.0.0.0         255.255.255.252 U     0      0        0 eth0
```

```
laboratorio@laboratorio:~/Desktop$ ping 172.16.22.5
ping: connect: Network is unreachable
laboratorio@laboratorio:~/Desktop$
```

Le ARP table rimangono invariate a seguito dell'esecuzione del comando ping in quanto non si supera il livello 3 e, sempre per lo stesso motivo, sia H1 che H2 non trasmettono pacchetti in rete.

3.2 bis - Ping ad intersezioni di subnet

Se si modificasse H1 con un indirizzo appartenente all'intersezione delle reti di H1 e H2 allora non si avrebbero problemi di ARP request/reply (entrambi gli IP rientrerebbero in entrambe le netmask) e quindi il ping di H1 otterrebbe risposta.

4 - Conflitto tra indirizzo broadcast e sottorete

Indirizzo IP H1	172.16.0.127/24
Indirizzo IP H2	172.16.0.1/25

4.1 - Ping da broadcast della sottorete

Quando H1 effettua un ping verso H2 manda una ARP request al MAC broadcast (FF:FF:FF:FF:FF:FF) per sapere a quale indirizzo MAC corrisponde l'IP di H2. H2 però non riesce a rispondere in quanto l'indirizzo IP del mittente della ARP request sarebbe il suo IP broadcast e rispondere a questa richiesta comporterebbe dunque associare ad un indirizzo IP broadcast un MAC nella ARP table di H2 violando il protocollo. H1 continua quindi a mandare ARP request che non avranno mai risposta. La ARP table di H1 risulterà quindi incompleta (non ha mai ricevuto il MAC associato a quell'indirizzo IP), mentre quella di H2 risulterà vuota in quanto ARP non prova neanche a riempirla con una entry che sarebbe insensata.

4.2 - Ping al broadcast della sottorete

Se H2 prova a effettuare un ping a H1 di fatto effettua un ping broadcast. Gli unici due host presenti nella rete di H2 saranno quindi H1 e H2; quest'ultimo risponderà attraverso l'interfaccia di loopback, come analizzato nel punto 1.1. H1 invece lo vedrà come un ping diretto (in quanto per lui 172.16.0.127 è il suo indirizzo e non quello broadcast); vorrebbe quindi rispondere, ma non conosce il MAC di H2. Per ottenerlo prova quindi a effettuare una ARP request, ottenendo la stessa situazione descritta nel punto 4.1. Nuovamente quindi la ARP table di H1 sarà incompleta, mentre quella di H2 vuota.

```
165 147.083160967 172.16.0.1          172.16.0.127        ICMP      98 Echo (ping) request  id=0x0012, seq=6/1536, ttl=64 (no respon
166 148.095927105 f4:39:09:15:79:81   ff:ff:ff:ff:ff:ff   ARP       42 Who has 172.16.0.1? Tell 172.16.0.127
```

```
laboratorio@laboratorio:~/Desktop$ ping 172.16.0.127 -b
WARNING: pinging broadcast address
PING 172.16.0.127 (172.16.0.127) 56(84) bytes of data.
64 bytes from 172.16.0.1: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 172.16.0.1: icmp_seq=2 ttl=64 time=0.017 ms
```

Molto bene 11/10