Internet Control Message Protocol (ICMP)

- RFC 792 (https://tools.ietf.org/html/rfc792)

"The network connecting devices are called Gateways…
Occasionally a gateway or ***destination host will communicate with a source host****, for example,* ***to report an error in datagram processing****.*

For such purposes this protocol,
the Internet Control Message Protocol (ICMP), is used*.*

ICMP, uses the basic support of IP as if it were a higher level protocol, *however,* ***ICMP is actually an integral part of IP,*** *and must be implemented by every IP module.*

The purpose of these control messages is to ***provide feedback about problems in the communication environment****, not to make IP reliable."*

ICMP

- Protocollo di livello 3 di supporto a IP:
  - **i router (o gateway) della rete inviano dei messaggi ICMP per segnalare malfunzionamenti nella rete**
  - un pacchetto ICMP utilizza un header IP standard, come se fosse un protocollo di livello superiore, ma **è parte del protocollo IP stesso e quindi supportato da tutti i dispositivi di rete di livello 3**
  - **non rende IP affidabile**, ma cerca di fornire delle informazioni sullo stato della rete
    - non vengono mai inviati messaggi ICMP riguardanti pacchetti ICMP non inviati correttamente

Tool di rete di livello 3

- Oltre a fornire feedback sullo stato delle connessioni, ICMP viene utilizzato per controllare lo stato della rete:
  - alcune applicazioni di rete utilizzano attivamente messaggi ICMP per verificare la connettività della rete (e.g., ***echo request*** e ***echo reply*** utilizzati da ***ping***)
  - altre applicazioni utilizzano i messaggi ICMP per ottenere informazioni sulla rete in maniera indiretta (e.g., ***time exceeded*** utilizzato da ***traceroute***)

Schema di un pacchetto ICMP (1)

- Utilizza un header IP standard, in cui il campo *protocollo* è settato a **1**:
  - indirizzi IP sorgente e destinazione identificano la comunicazione
  - campo TTL per evitare che il pacchetto circoli all'infinito nella rete

- può includere informazioni riguardanti altri livello dello stack TCP/IP, non solo IP (e.g., ***port unreachable***)

- la prima informazione fornita nel payload del pacchetto ICMP è il campo ***type:***
  - identifica il tipo di informazione fornita dal pacchetto e la struttura e la logica del resto del pacchetto cambiano sulla base di esso
  - vediamo alcuni tipi di messaggi nelle prossime slide

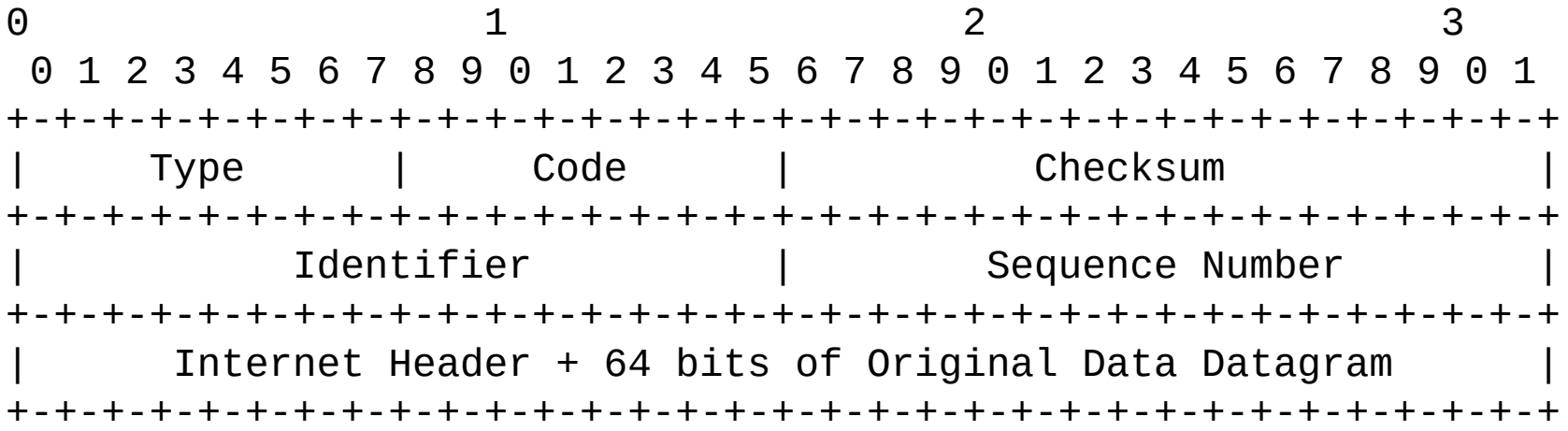
Alcuni tipi di messaggi ICMP

(type) Message

0	**Echo Reply**
3	**Destination Unreachable**
4	Source Quench
5	**Redirect**
8	**Echo**
11	**Time Exceeded**
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Echo or Echo Reply Message (type = 8, type = 0)

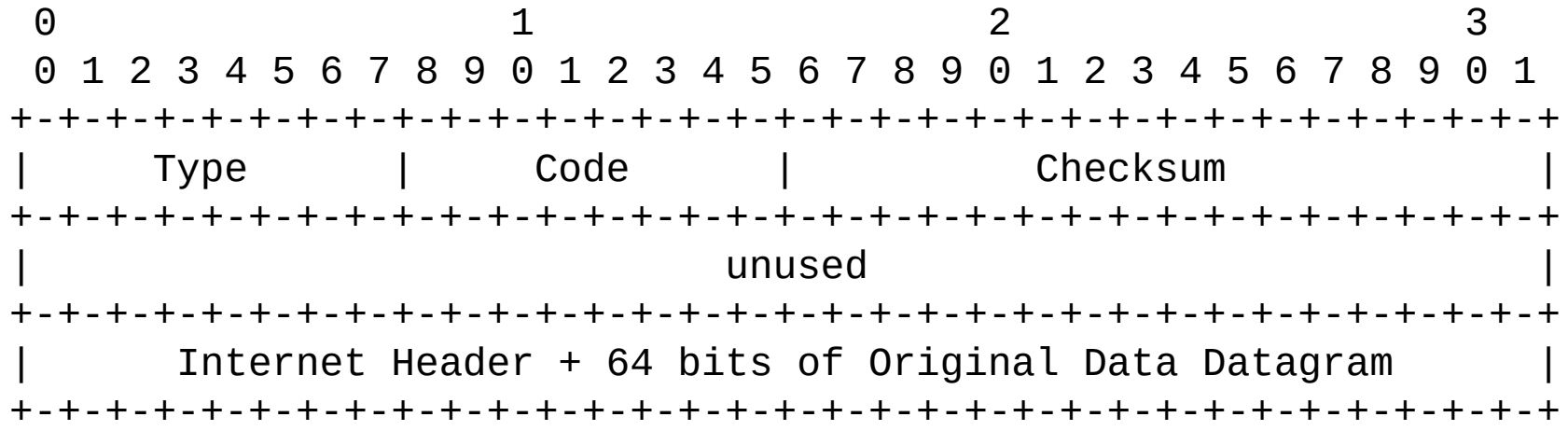
- *"The data received in the echo message must be returned in the echo reply message."* → *Se un host riceve un echo (request), deve rispondere con un echo reply*



- Pensato appositamente per testare attivamente la connettività di rete
- Protocolli utilizzati da ***ping***

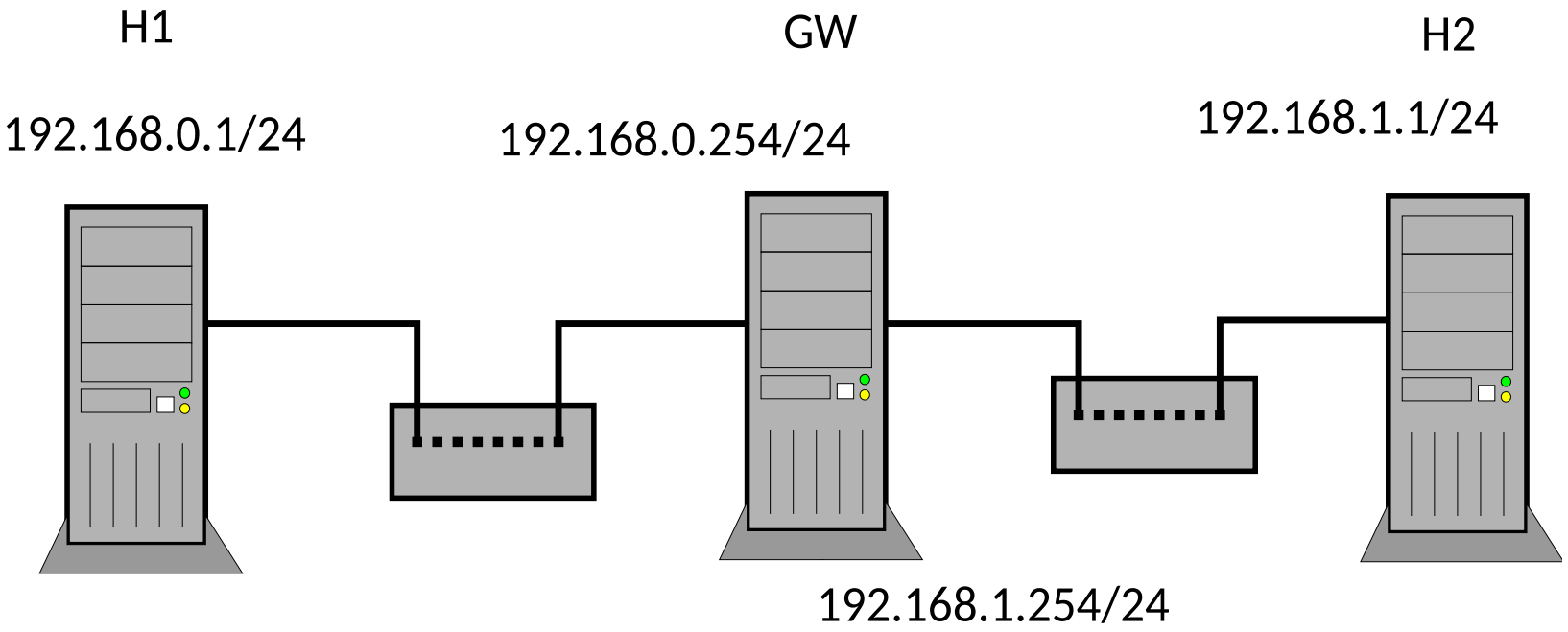
Destination Unreachable Message (type = 3)

- *Pacchetto utilizzato per informare che è impossibile raggiungere una certa destinazione (non solo IP)*



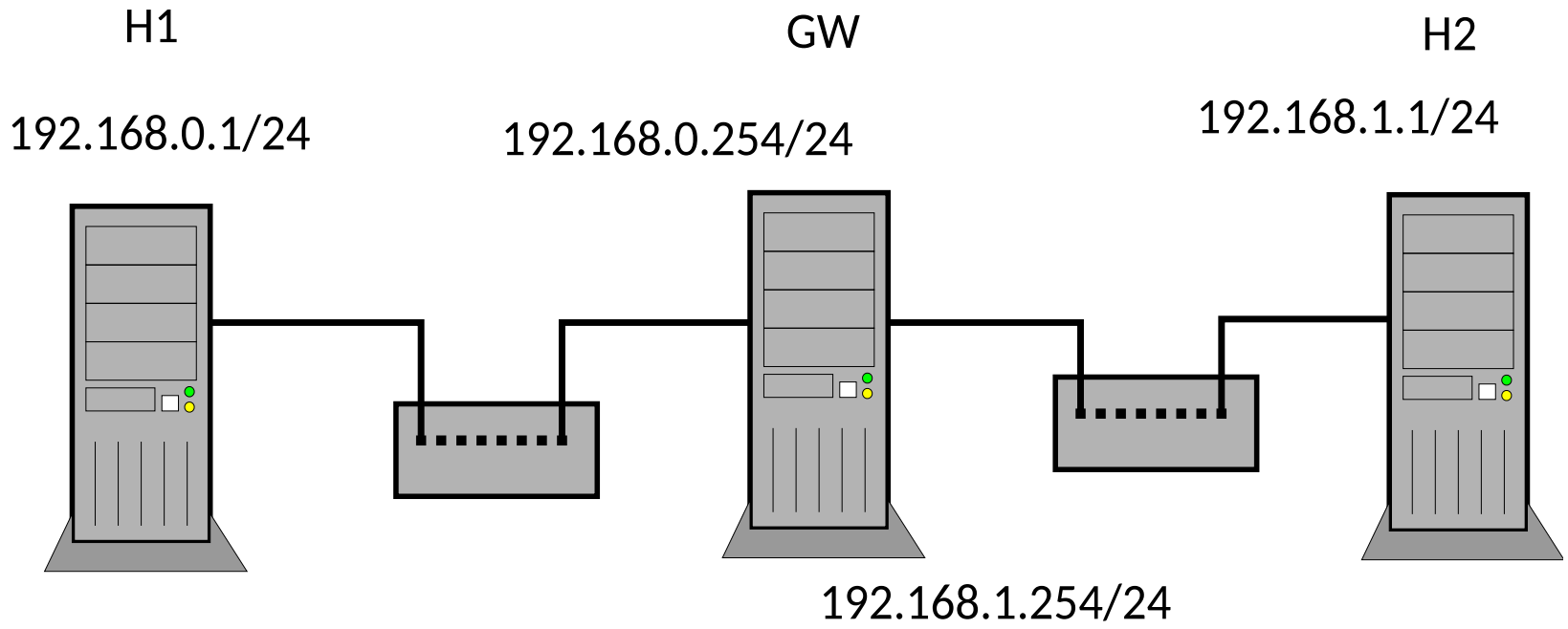
- L'indirizzo IP di destinazione utilizzato sarà l'indirizzo IP sorgente del pacchetto interessato dall'errore; l'indirizzo IP sorgente è l'indirizzo del router/gateway/host che invia il paccheto ICMP
- Il campo ***code*** definisce il messaggio specifico, ad *esempio*:
  - code = 0 → network unreachable *(e.g., next-hop sconosciuto)*
  - code = 1 → host unreachable *(e.g., host irraggiungibile a livello H2N)*
  - code = 3 → port unreachable *(e.g., impossibile utilizzare la porta indicata, messaggio inviato sempre da un host)*

Host unreachable, esempio




```
(h1) $ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
From 192.168.0.254 icmp_seq=1 Destination Host Unreachable
From 192.168.0.254 icmp_seq=2 Destination Host Unreachable
From 192.168.0.254 icmp_seq=3 Destination Host Unreachable
```

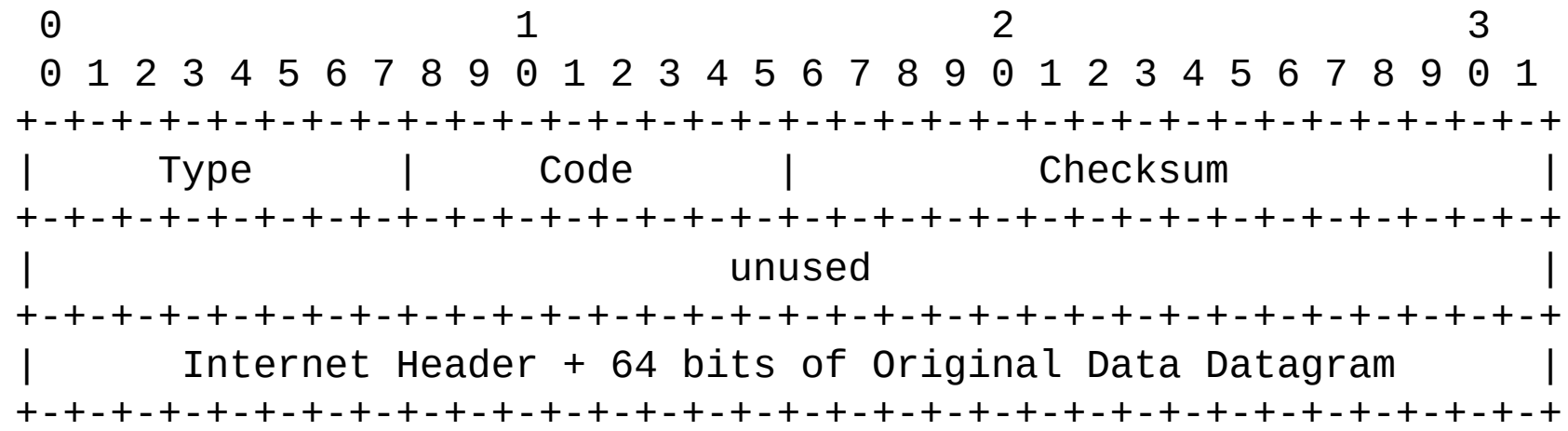
Network unreachable, esempio




```
(h1) # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
From 192.168.0.254 icmp_seq=1 Destination Net Unreachable
From 192.168.0.254 icmp_seq=2 Destination Net Unreachable
From 192.168.0.254 icmp_seq=3 Destination Net Unreachable
```

Time Exceeded Message (type = 11)

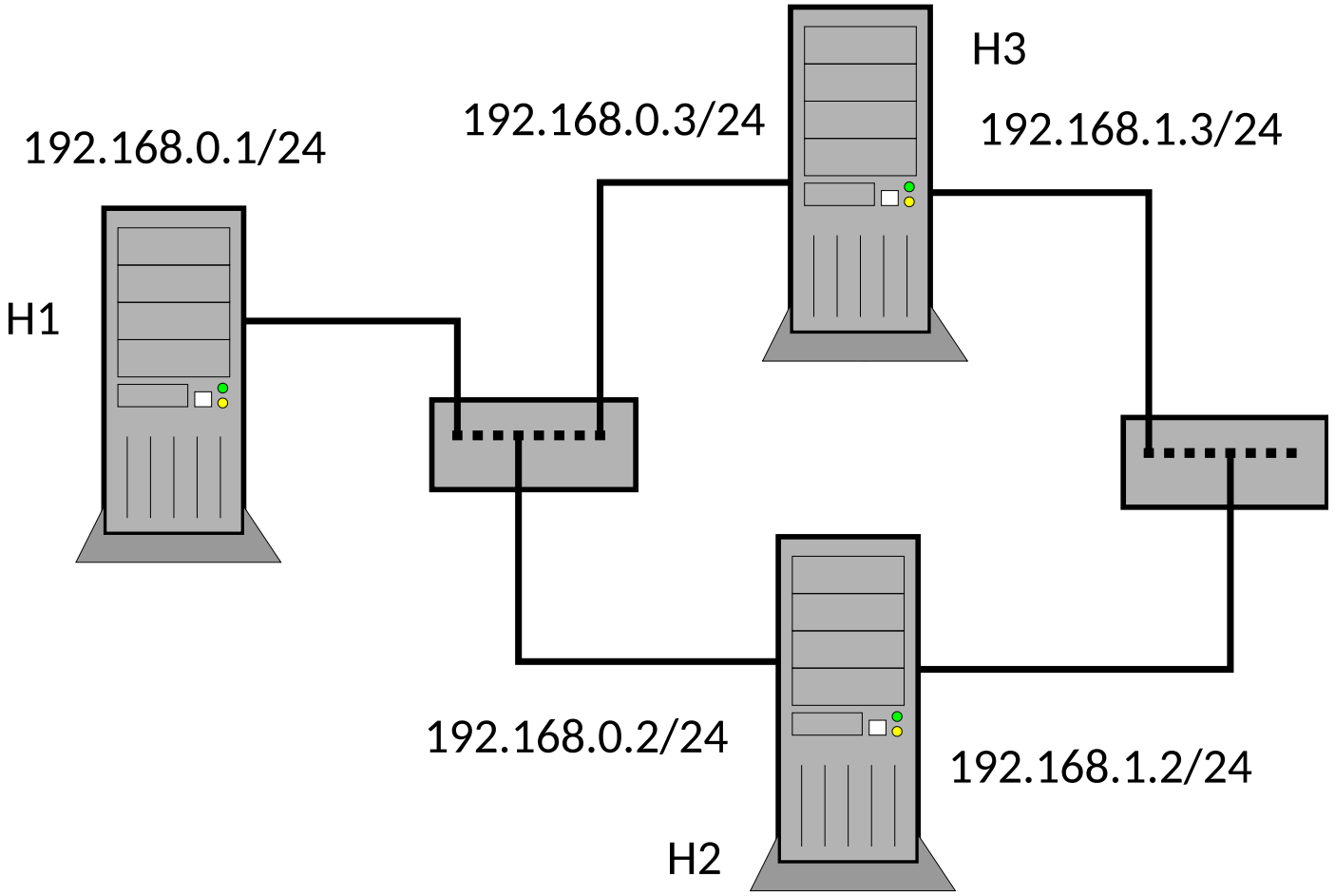
- *Il tempo di vita del pacchetto è esaurito*



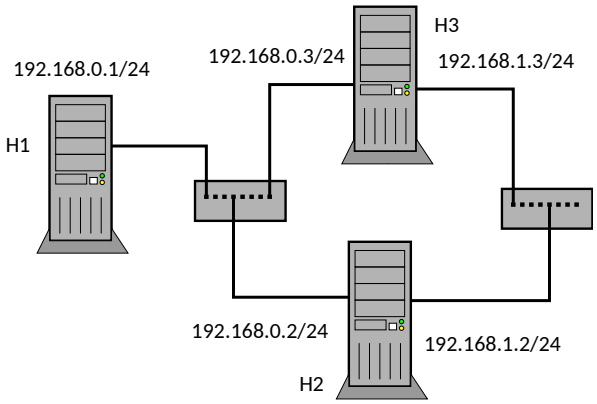
- **Code = 0 → utilizzato per segnalare che il pacchetto è stato scartato a causa del TTL esaurito**
  - Ogni volta che un router inoltra un pacchetto IP, decrementa di 1 il campo TTL nell'header IP del pacchetto
  - Prima di inviare il pacchetto (ovvero, dopo aver decremenetato di uno il TTL), il sistema controlla se il campo del TTL
  - Se TTL = 0, il pacchetto non viene inviato, e viene invece scartato. Il router manda il pacchetto ICMP con *type=11* e *code=0* per segnalare questa informazione

Time Exceeded, esempio (1)

Ipotizziamo una rete configurata male, in cui i gateway si inoltrano i pacchetti a vicenda creando un ciclo



Time Exceeded, esempio (2)



(h1) # route -n

```
Kernel IP routing table
Destination  Gateway       Genmask         Iface
0.0.0.0      192.168.0.3   0.0.0.0         eth1
192.168.0.0  0.0.0.0       255.255.255.0   eth0
```

(h2) # route -n

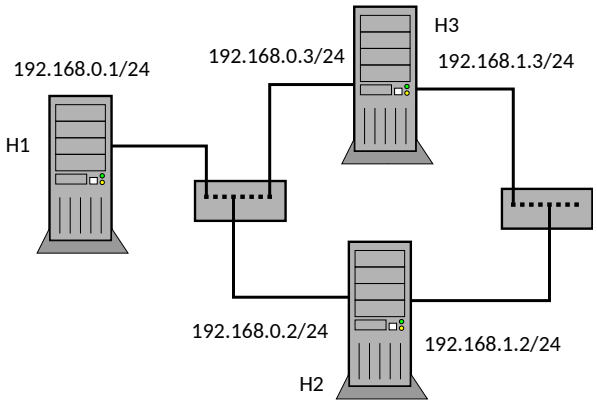
```
Kernel IP routing table
Destination     Gateway         Genmask         Iface
0.0.0.0         192.168.0.3     0.0.0.0         eth0
192.168.0.0     0.0.0.0         255.255.255.0   eth0
192.168.1.0     0.0.0.0         255.255.255.0   eth1
```

(h3) # route -n

```
Kernel IP routing table
Destination     Gateway         Genmask         Iface
0.0.0.0         192.168.1.2     0.0.0.0         eth1
192.168.0.0     0.0.0.0         255.255.255.0   eth0
192.168.1.0     0.0.0.0         255.255.255.0   eth1
```

Time Exceeded, esempio (3)

Se h1 invia un pacchetto destinato ad un host che non fa parte delle due sottoreti, h2 e h3 si inoltrano il pacchetto a vicenda.




```
(h1) # ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
From 192.168.0.2 icmp_seq=1 Time to live exceeded
From 192.168.0.2 icmp_seq=2 Time to live exceeded
From 192.168.0.2 icmp_seq=3 Time to live exceeded
```

Dopo un certo numero di “hop”, il campo TTL raggiunge il valore 0 e viene scartato scartato dal router. In questo sappiamo che il router che ha rilavato il TTL=0 e ha scartato il pacchetto è H2

`traceroute`

Il comando **`traceroute`** permette di capire per quali router passano i pacchetti IP quando sono diretti ad una data destinazione.

`traceroute` è compreso nella dotazione standard di tutte i sistemi Unix.

Nei sistemi Windows il comando ha il nome **`tracert`** e sintassi leggermente differente.

`traceroute` trova utilizzo anche nell'ambito dell'amministrazione della rete per isolare guasti o incoerenze nelle tabelle di routing. Dal punto di vista didattico la sua utilità è di mostrare il percorso dei pacchetti **svelando un po' della topologia nascosta di Internet**.

traceroute - esempio

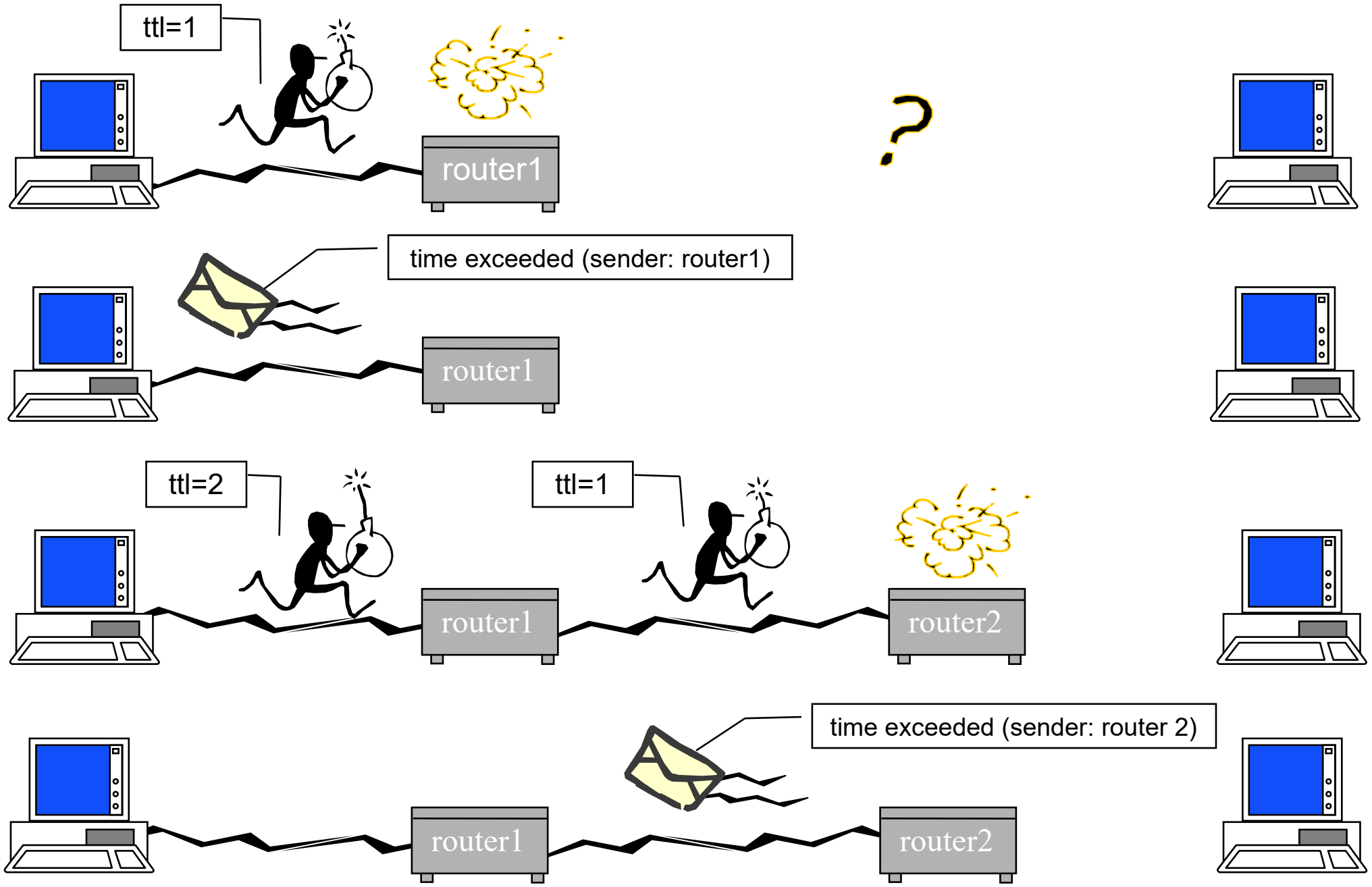
```
<utente@pascal ~>traceroute wilma.cs.brown.edu
traceroute to wilma.cs.brown.edu (128.148.19.15), 30 hops max, 40
byte packets
1  gw1.fis.uniroma3.it (193.204.160.1)  3 ms  3 ms  2 ms
2  141.108.132.1 (141.108.132.1)  832 ms  967 ms  402 ms
3  mp4rm1.roma1.infn.it (141.108.127.6)  267 ms  106 ms  417 ms
4  atm-garrten-rm.infn.it (192.135.31.5)  100 ms  939 ms  839 ms
5  cnafint-ten34.infn.it (192.135.34.21)  1100 ms *  1056 ms
6  mix-serial3-4.Washington.mci.net (204.189.152.161)  618 ms * *
7  * core1-fddi-0.Washington.mci.net (204.70.2.1)  1249 ms *
8  * * *
9  wtn-bbn-nap.Washington.mci.net (206.157.77.218)  766 ms * *
10  * * chicago1-br1.bbnplanet.net (4.0.1.5)  857 ms
11  * * *
12  * boston1-br1.bbnplanet.net (4.0.2.245)  846 ms *
13  boston1-br2.bbnplanet.net (4.0.2.250)  680 ms * *
14  * * boston1-mr4.bbnplanet.net (4.0.44.19)  648 ms
15  providence-cr1.bbnplanet.net (4.0.45.106)  416 ms providence-
cr1.bbnplanet.net (4.0.45.102)  1298 ms *
16  brown.bbnplanet.net (131.192.32.2)  1444 ms  615 ms  802 ms
17  * * *
18  * ftp.cs.brown.edu (128.148.19.15)  834 ms  435 ms
<utente@pascal ~>
```

traceroute (2)

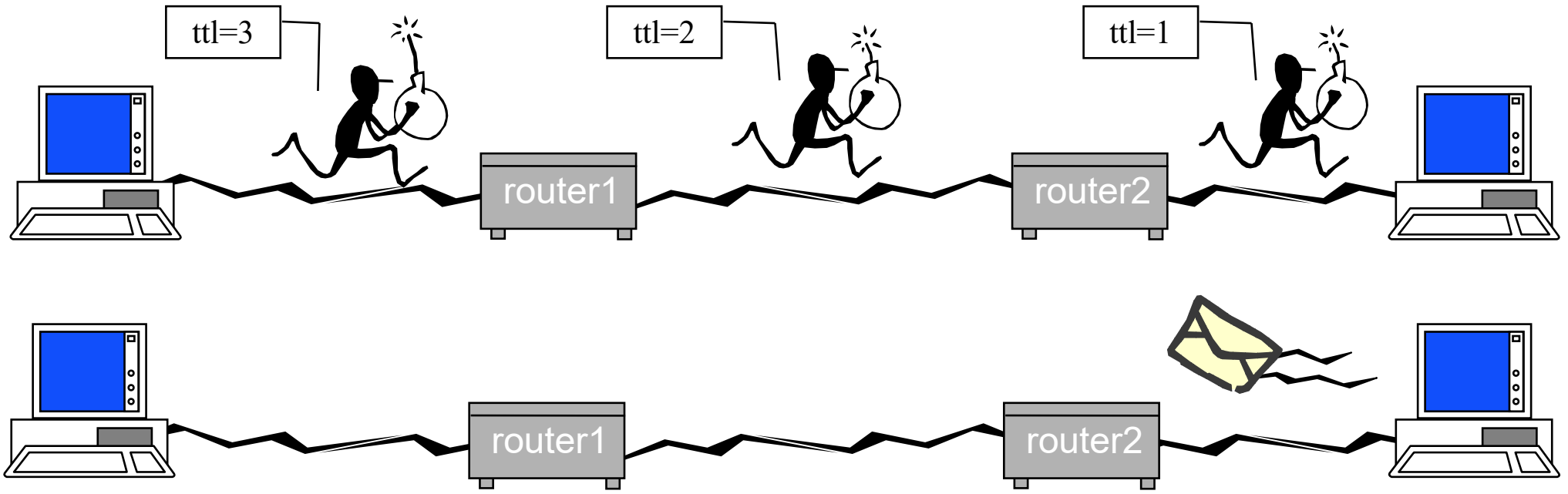
- Si ipotizzi di inviare un pacchetto, ad esempio un **ECHO_REQUEST** con TTL=*n* “basso”. Se il destinatario è troppo lontano l’ultimo router risponderà con un **TIME_EXCEEDED**, se il destinatario risponde con **ECHO_REPLY** allora è raggiungibile con *n* hop
- In realtà alcuni **traceroute** non utilizzano gli **ECHO_REQUEST** se non gli viene chiesto esplicitamente ma altri protocolli (**UDP**).
  - Per forzare l'utilizzo di ICMP usare il parametro opzionale **--icmp**:

    traceroute --icmp <host>
- Inoltre manda 3 pacchetti per ogni valore di *n*. Se non si osserva un **TIME_EXCEEDED** entro un tempo prestabilito **traceroute** visualizza un asterisco “*”
- Il manuale in linea di **traceroute** è estremamente interessante e mostra anche esempi di strani comportamenti da parte dei router

Esempio: `traceroute` (2)



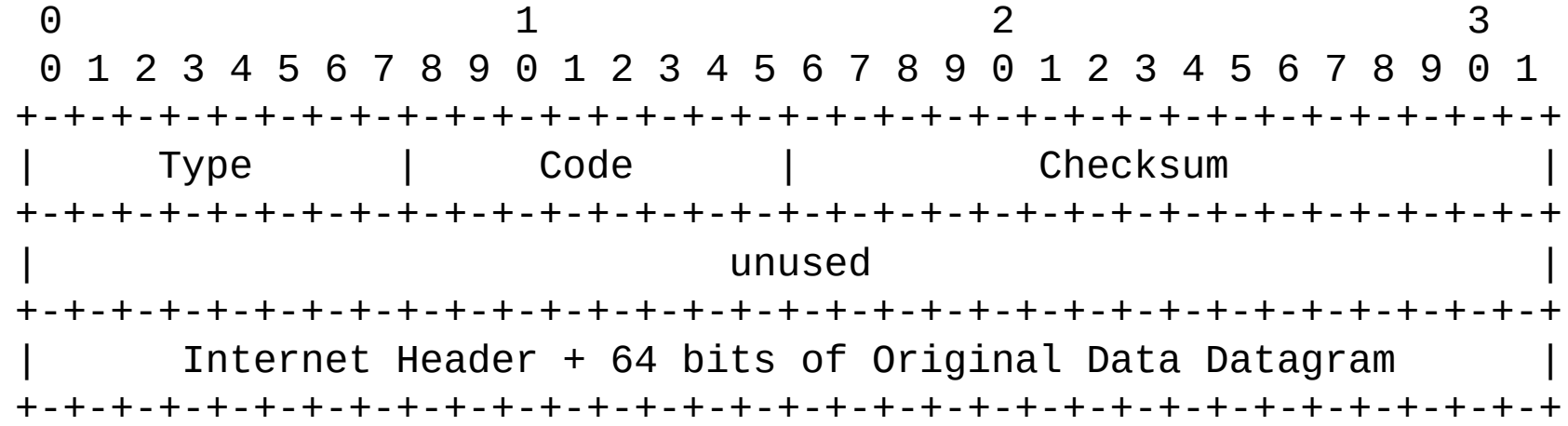
Esempio: `traceroute` (2)



Il messaggio di risposta dipende dal messaggio inviato

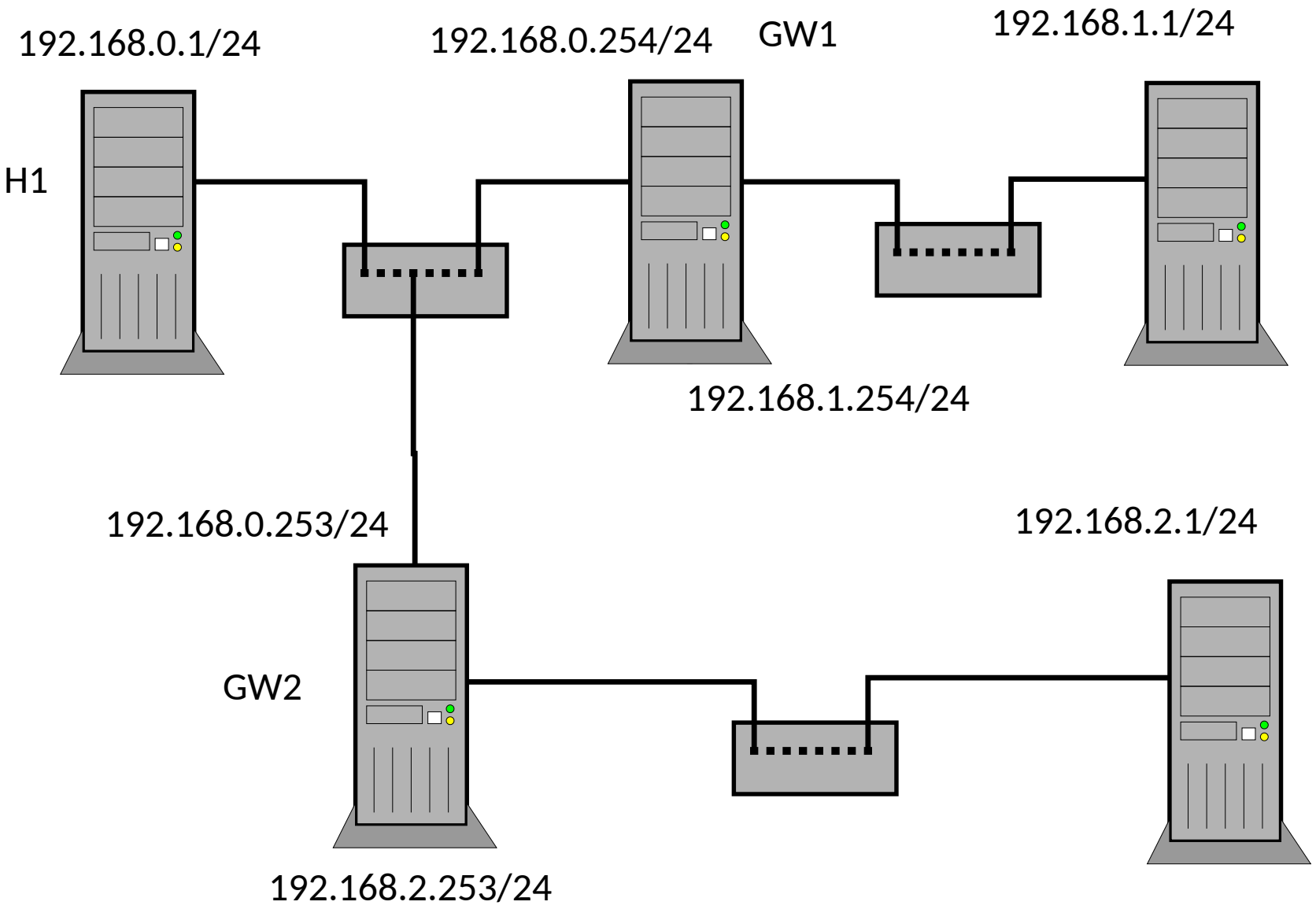
Redirect Message (type = 5)

- *Esiste un percorso migliore per inviare il pacchetto alla destinazione*



- In base alle proprie regole di routing, un gateway può informare il mittente del pacchetto IP che esiste un percorso migliore per inviare pacchetti alla destinazione desiderata:
  - Code = 1 → Host redirect:
    se un gateway deve inoltrare un pacchetto a un gateway presente nella stessa subnet dell'IP sorgente del pacchetto, può inviare un messaggio ICMP 'Host redirect' per comunicare all'host mittente l'indirizzo di questo gateway, che rappresenta un percorso migliore. L'host mittente può inviare I pacchetti successivi direttamente al gateway indicato (per questioni di sicurezza, l'host potrebbe anche decidere di ignorare il pacchetto)

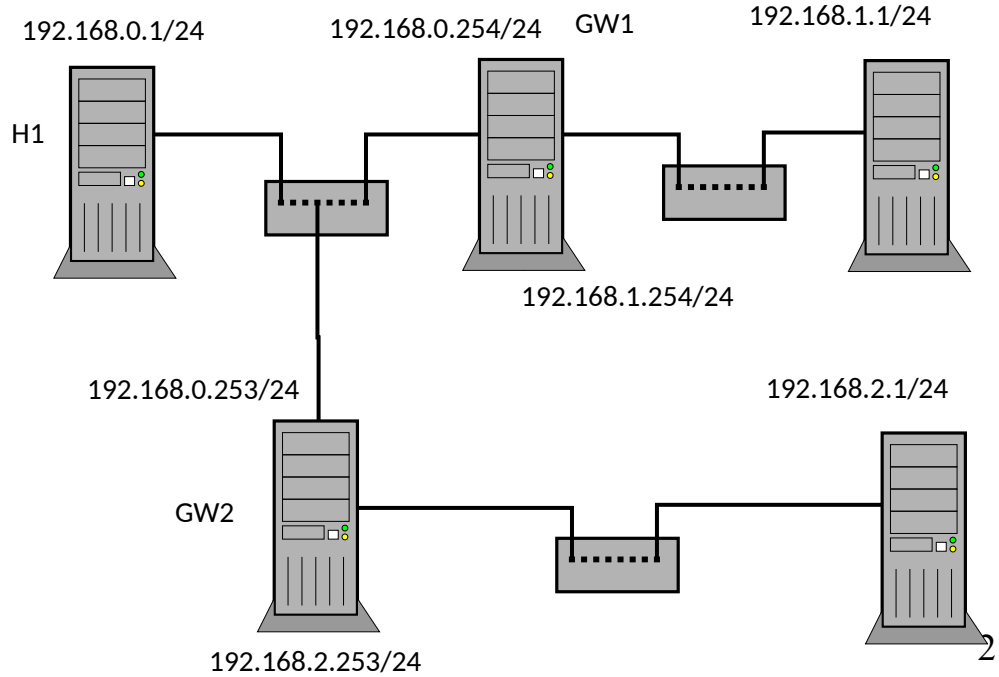
Host Redirect, esempio (1)



Host Redirect, esempio (2)

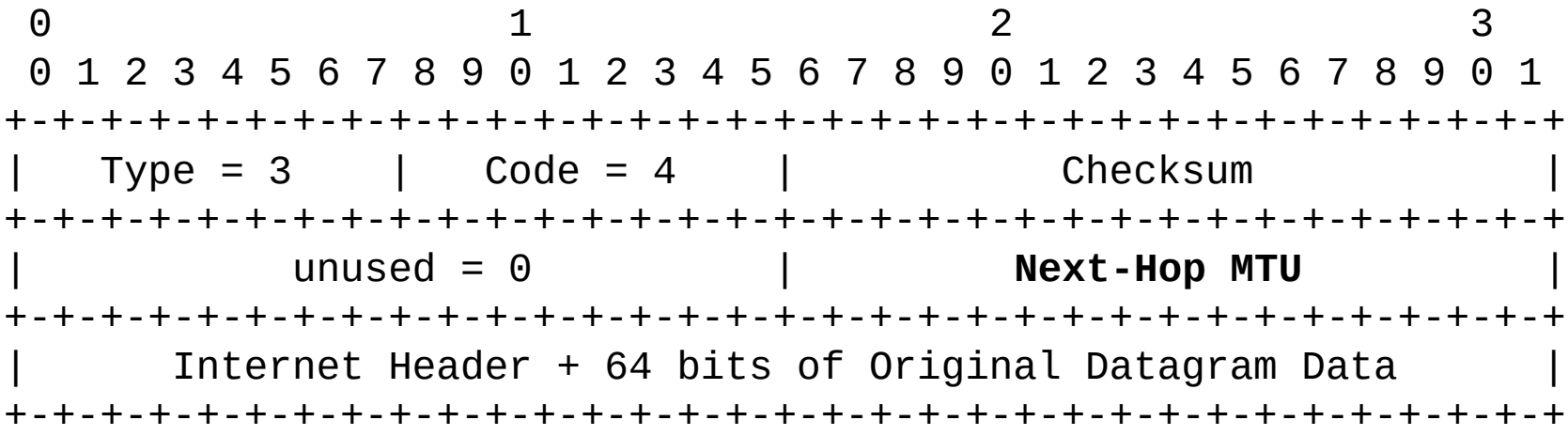
```
(h1) # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Iface
0.0.0.0         192.168.0.254   0.0.0.0         UG      eth0
192.168.0.0     0.0.0.0         255.255.255.0   U       eth0

(h1) # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=63 time=2.70 ms
From 192.168.0.254: icmp_seq=2 Redirect Host(New nexthop: 192.168.0.253)
64 bytes from 192.168.2.1: icmp_seq=2 ttl=63 time=2.91 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=63 time=2.49 ms
```



ICMP Packet too big

Un tipo particolare della famiglia *unreachable*, inviato dai router se il pacchetto IP non può essere inoltrato a causa di MTU troppo piccolo

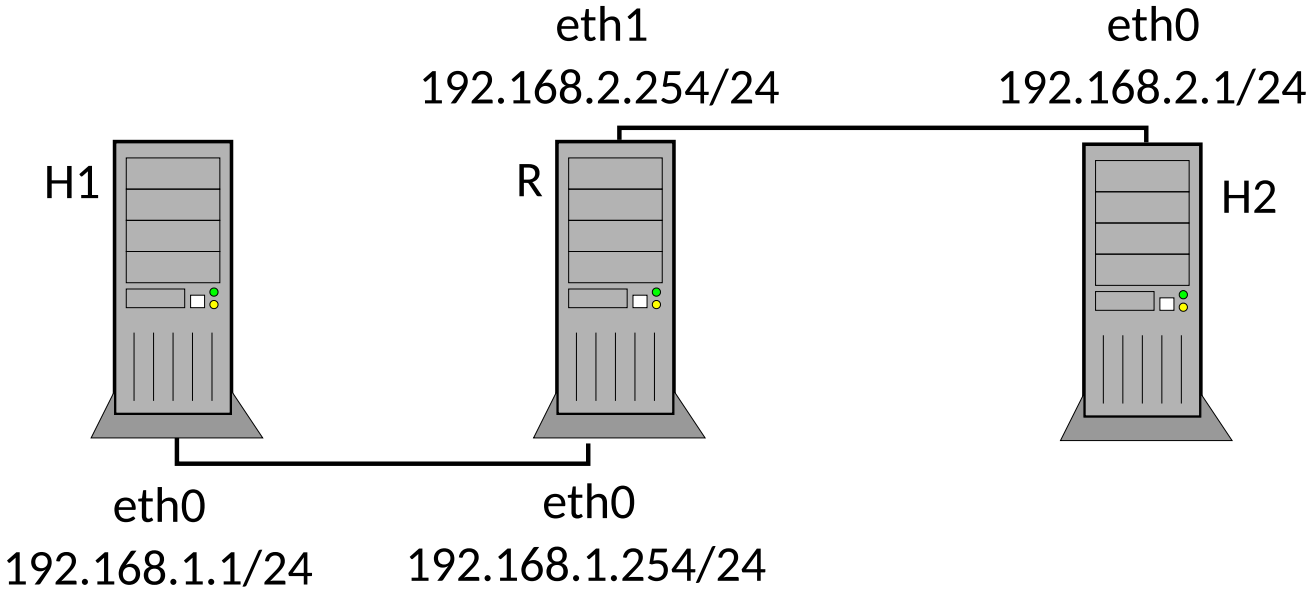


Messaggio ICMP potenzialmente inviato in caso di impossibilità di inoltro per pacchetto troppo grande (IP packet size > MTU interfaccia di inoltro)

- In caso di frammentazione non supportata dal router
- In caso di flag ***do not fragment*** settato dal mittente nell'header IP

Ha lo scopo di consentire l'individuazione del *Path MTU (PMTU)*, la dimensione del pacchetto IP supportata da tutti i link H2N nel percorso end-to-end

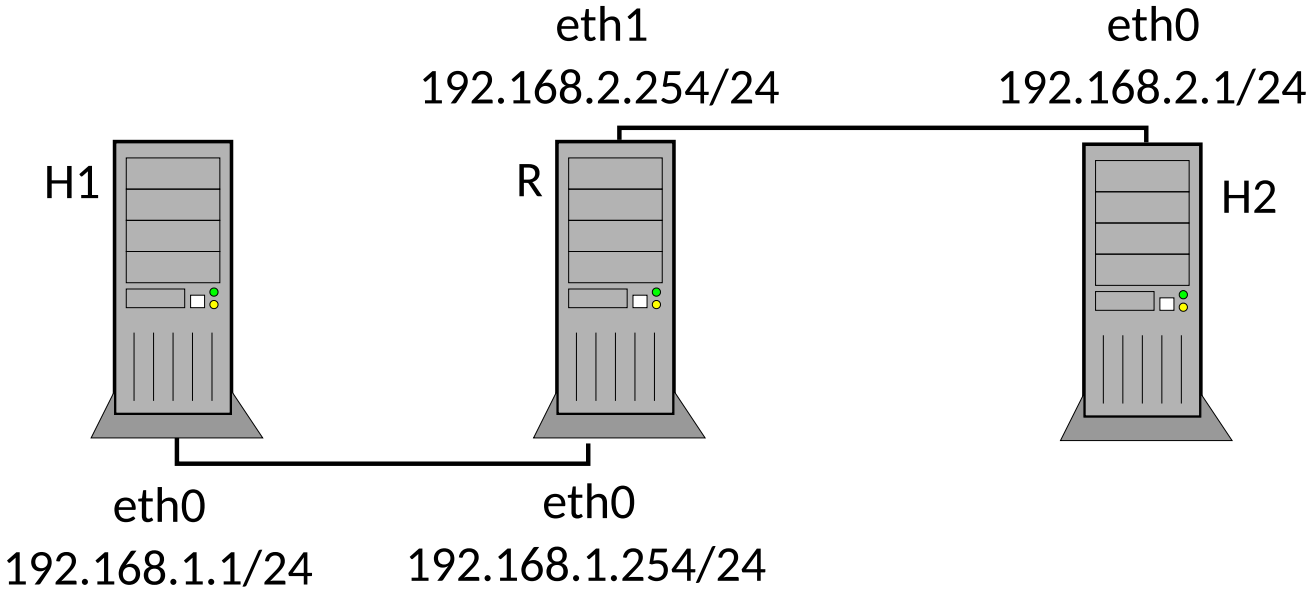
Esempio ICMP Packet too big [1] Configurazione




```
h1# ifconfig eth0 192.168.1.1/24 mtu 1500
h1# route add -net 192.168.2.0/24 gw 192.168.1.254
r1# sysctl -w net.ipv4.ip_forward=1
r1# ifconfig eth0 192.168.1.254/24 mtu 1500
r1# ifconfig eth1 192.168.2.254/24 mtu 600
h2# ifconfig eth0 192.168.2.1/24 mtu 600
h2# route add -net 192.168.1.0/24 gw 192.168.1.254
```

Nota: per limiti di Marionnet non possiamo usare MTU > 1500 (e.g., jumbo frames MTU 9000)

Esempio ICMP Packet too big [2]
Test




```
h1# ping -M do -s 1000 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 1000(1028) bytes of data.
From 192.168.1.254 icmp_seq=1 Frag needed and DF set (mtu = 600)
ping: local error: Message too long, mtu=600
ping: local error: Message too long, mtu=600

h1# ip route get 192.168.2.1
192.168.2.1 via 192.168.1.254 dev eth0  src 192.168.1.1
    cache  expires 572sec mtu 600
```