Taller de Capa de Red

Teoría de las Comunicaciones

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

16.05.2018

Agenda

1. ICMP: el protocolo de control de internet
   - ICMP desde Scapy
2. Traceroute: construyendo la ruta que siguen los datagramas
   - Implementaciones

El protocolo ICMP

- Protocolo de control que forma parte del núcleo de la arquitectura TCP/IP.
- La sigla: *Internet Control Message Protocol*.
- Objetivo: proveer mensajes de error y de control. No intercambia datos!
- Especificado en el RFC 792.

Cómo y dónde se usa

- Del RFC: ICMP **debe** ser implementado por cada módulo IP.
- Pueden ser enviados tanto por routers como por hosts arbitrarios.
- Son generados a causa de:
  - Errores en los datagramas IP.
  - Necesidad de comunicar información de diagnóstico.
  - Necesidad de comunicar información de ruteo.
- Siempre se envían a la dirección source del datagrama IP que motivó el mensaje.

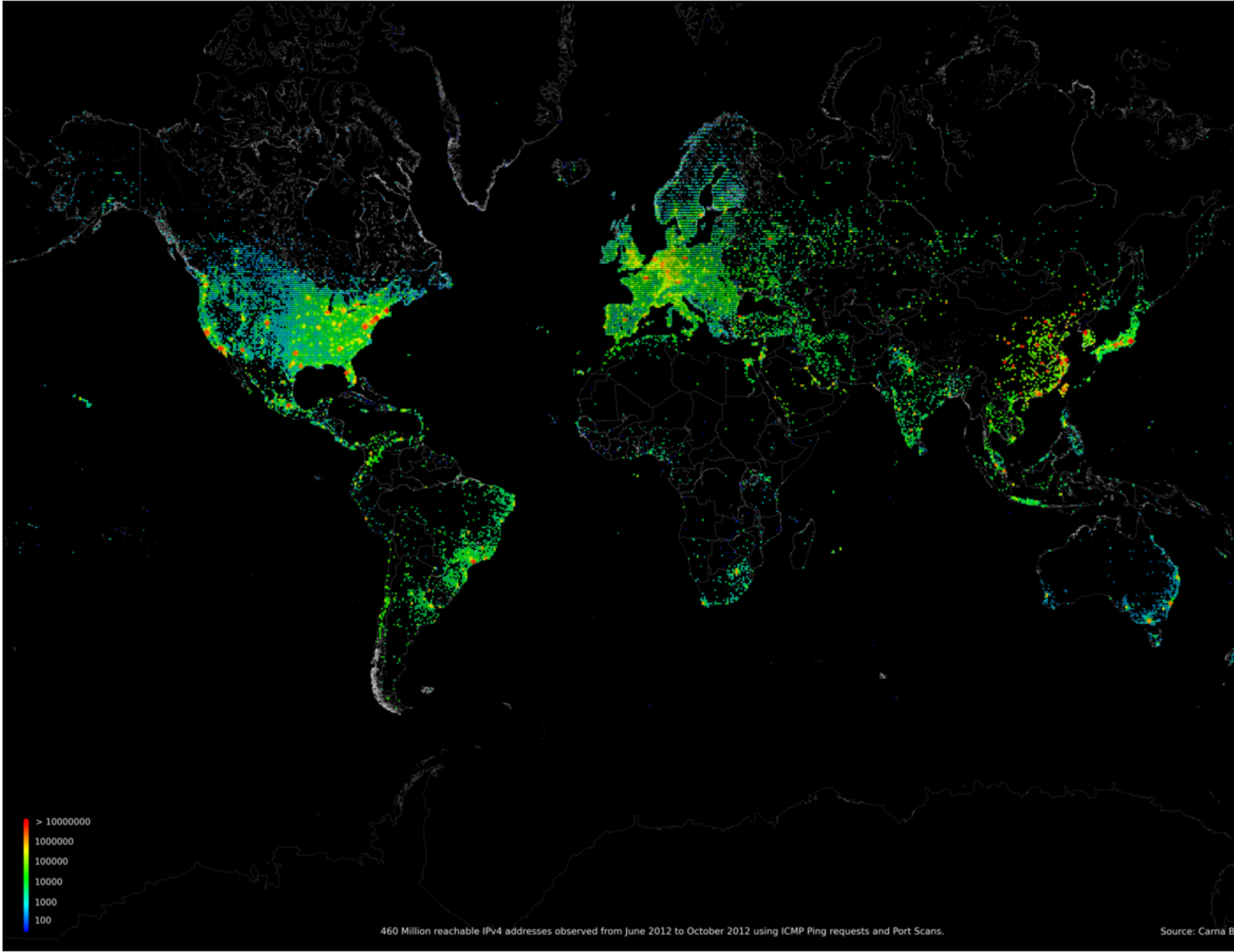
Formato de los paquetes

- Los paquetes constan de un header de 8 bytes y una sección de datos variable.
- **Header**:
  - `Type` (1 byte): indica el tipo del mensaje y define el formato de lo que sigue.
  - `Code` (1 byte): especifica el subtipo.
  - `Checksum` (2 bytes): usa el algoritmo de IP sobre el header más los datos del paquete ICMP.
  - Los restantes 4 bytes dependen del tipo.

Type	Name
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)

Ejemplo: Echo Request (PING)

- La herramienta de diagnóstico `ping` usa estos mensajes (y el respectivo *Echo Reply* - tipo 0).
- En este caso, los 2 bytes restantes del header indican:
  - `Identifier` (1 byte): permite asociar solicitudes con respuestas.
  - `Sequence Number` (1 byte): ídem anterior.
- Y la sección de datos puede contener información arbitraria que debe ser devuelta en la respuesta.


> 10000000
1000000
100000
10000
1000
100
460 Million reachable IPv4 addresses observed from June 2012 to October 2012 using ICMP Ping requests and Port Scans.
Source: Carna B

Ejemplo: Destination Unreachable

- El tipo 3, por otro lado, es el de *Destination Unreachable*.
- Tiene varios subtipos. Algunos ejemplos:
  - `Destination network unreachable` (código 0):

Ejemplo: Destination Unreachable

- El tipo 3, por otro lado, es el de *Destination Unreachable*.
- Tiene varios subtipos. Algunos ejemplos:
  - `Destination network unreachable` (código 0): si el router no sabe cómo pasar el paquete (i.e., no tiene una ruta programada para la red destino).
  - `Destination host unreachable` (código 1):

Ejemplo: Destination Unreachable

- El tipo 3, por otro lado, es el de *Destination Unreachable*.
- Tiene varios subtipos. Algunos ejemplos:
  - `Destination network unreachable` (código 0): si el router no sabe cómo pasar el paquete (i.e., no tiene una ruta programada para la red destino).
  - `Destination host unreachable` (código 1): si el host destino está en la red del router pero éste determinó que no puede llegar al host.
  - `Destination port unreachable` (código 3):

Ejemplo: Destination Unreachable

- El tipo 3, por otro lado, es el de *Destination Unreachable*.
- Tiene varios subtipos. Algunos ejemplos:
  - `Destination network unreachable` (código 0): si el router no sabe cómo pasar el paquete (i.e., no tiene una ruta programada para la red destino).
  - `Destination host unreachable` (código 1): si el host destino está en la red del router pero éste determinó que no puede llegar al host.
  - `Destination port unreachable` (código 3): el mensaje llegó al destino pero el puerto no tiene un proceso asociado. Lo envía el host - no el router como los anteriores.
- Header: los 2 bytes restantes quedan `unused`.
- Datos: Se copia el header IP del datagrama original más los primeros 8 bytes de los datos respectivos.

Implementación de ping

Armando y enviando un Echo Request

```
>>> packet = IP(dst='www.google.com') / ICMP()
>>> sr(pkt)
>>> res[0][ICMP].display()
0000 IP / ICMP 192.168.0.105 > 173.194.42.211
     echo-request 0 ==> IP / ICMP 173.194.42.211 > 192.168.0.105
                                    echo-reply 0
```

Jugando con el TTL

Armando un paquete con TTL bajo

```
>>> sr(IP(dst='www.dc.uba.ar', ttl=1))
>>> res[0][ICMP].display()
    0000 192.168.0.105 > 157.92.27.21 ip ==> IP / ICMP 192.168.0.1 >
            192.168.0.105 time-exceeded ttl-zero-during-transit / I
```

¿Qué es `traceroute`?

- Es una herramienta de diagnóstico para averiguar las rutas que atraviesan los paquetes en Internet.
- La mayoría de los sistemas operativos actuales proveen alguna implementación. Ejemplos:
  - `tracert` en Windows.
  - `traceroute` en *nix.
- Al correr la herramienta, se debe indicar hacia qué host destino se desea trazar la ruta.
- La salida obtenida suele mostrar las direcciones IP de los hops sucesivos y el respectivo tiempo de respuesta esperado.

`traceroute` sobre ICMP

- Implementa (esencialmente) el siguiente algoritmo:
  1. Sea h la IP del host destino y sea `ttl` $= 1$.
  2. Repetir los siguientes pasos hasta obtener una respuesta ICMP de tipo *Echo Reply* por parte de h:
  3. Enviar un paquete ICMP de tipo *Echo Request* al host h cuyo campo `TTL` en el header IP valga `ttl`.
  4. Si se recibe una respuesta ICMP de tipo *Time Exceeded*, anotar la IP origen de dicho paquete. En otro caso, marcar como desconocido (*) el hop.
  5. Incrementar `ttl`.

`traceroute` sobre ICMP: observaciones

- Usualmente suele enviarse una serie de paquetes por cada valor de `ttl` (por lo general tres).
- A través de esto, puede estimarse el tiempo medio de respuesta.
- El host origen define un timeout para esperar por cada respuesta. Pasado este intervalo, el hop actual se asume desconocido.
- Observar que las rutas no necesariamente serán siempre iguales!