1 Lezione del 31-10-25

1.0.1 Limitazioni di IPv4

Abbiamo visto come gli indirizzi IP (versione 4) formano uno spazio di 32 bit (dimensione 2^32, quindi circa 4 miliardi). L'**ICANN** (*Internet Corporation for Assigned Numbers Autorithy*) gestisce tale spazio, attraverso 5 **RR** (*Registri Nazionali*), che a loro volta allocano i loro registri privati. Questi registri allocano poi blocchi di indirizzi IP agli ISP, che a loro volta li forniscono ad organizzazioni, abitazioni, ecc... (secondo il **CIDR**, *Classless InterDomain Routing*).

Il problema è che lo spazio di indirizzi IPv4 (quelli a 32 bit) è stato esaurito nel 2011. Meccanismi come il *NAT* aiutano a risolvere questo problema, ma ciò non toglie che uno spazio a 32 bit per gli indirizzi Internet è effettivamente poco. Per questo motivo è stata introdotta la *versione 6* del protocollo IP (**IPv6**), che ha uno spazio di indirizzi a 128 bit.

La transizione verso IPv6 è attualmente in corso, e il protocollo ad oggi non è sempre supportato.

1.1 NAT

Il **NAT** (*Network Address Translation*) è un meccanismo secondo il quale più dispositivi su una rete locale condividono lo stesso indirizzo IPv4 per il resto del mondo.

Ricordiamo che alcuni indirizzi IP sono riservati:

- L'indirizzo *tutto a 0* (`0.0.0.0`) viene usato dagli host che ancora non hanno un loro indirizzo (ad esempio come sorgente nel DHCP);
- L'indirizzo *tutto a 1* (`255.255.255.255`) è l'indirizzo di *broadcast*. Esiste anche un'indirizzo di broadcast composto dalla maschera di rete, più il resto dei bit a 1;
- Alcuni spazi di indirizzamento sono **privati**: questi sono `10.0.0.0`, `127.0.0.0`, `192.168.0.0`, ecc... Gli indirizzi privati non possono essere usati per collegarsi alla rete Internet pubblica (non sono indirizzi *pubblici*), ma si limitano alla rete locale dove vengono usati.

Il modo in cui possono tornare utili gli indirizzi privati è effettuando un'apposita traduzione dalla rete locale alla rete Internet pubblica, effettuata appunto dal *NAT*, quando i datagrammi IP che vengono inviati sulla rete privata sono destinati ad un host furoi dalla rete privata.

In questo caso si sostituisce all'indirizo privato un indirizzo condiviso fra tutti gli host sulla rete privata (sostanzialmente quello del router). Dall'altra parte della rete, l'unica cosa che distinguerà datagrammi diversi dalla stessa rete NAT sarà il numero di porta.

Dal punto di vista implementativo, quindi, un router che supporta NAT deve:

- Per i datagrammi **in uscita**, rimpiazzare l'indirizzo privato col suo IP, e assegnare al datagramma una nuova porta;
- **Ricordare** in una tabella, detta *NAT Translation Table*, ogni coppia sorgente (indirizzo private all'interno della rete, porta originale) destinazione (indirizzo pubblico, nuova porta);

- Per i datagrammi **in entrata**, effettuare la ricerca al contrario, rimpiazzando l'indirizzo pubblico con quello privato, e assegnando al datagramma la sua porta originale.

Un problema che potremmo avere è come indirizzare server su NAT. Nel nostro esempio attuale, dietro il NAT si trovava un client: questo fa le richieste, il router le traduce, il server inoltra risposte e il router sa come inoltrarle al client.

Un server, di contro, non potrebbe apparentemente essere indirizzato se si trovasse dietro un NAT. Una soluzione è quella di prevedere entrate nella NAT Translation Table, impostate manualmente, che associano ad esempio a ogni richiesta sulla porta 80 un singolo server all'interno del NAT. In questo modo ogni client che vuole accedere al web server all'indirizzo NAT (porta 80) potrà farlo, e il router tradurrà l'indirizzo all'indirizzo privato corrispondente al web server (porta 80).

1.2 IPv6

Abbiamo introdotto IPv6 IN 18.0.1, motivandone l'introduzione per:

1. Superare l'esaurimento degli indirizzi ormai associato ad IPv4;
2. Limitare l'overhead di 40 byte di header di IPv4.

Il protocollo IPv6, come vedremo, rimuove molte funzionalità presenti in IPv4 per essere più veloce ed efficiente.

1.2.1 Datagramma IPv6

Il **datagramma IP** versione 6 ha la seguente struttura:

```
16 bit            16 bit
<ver> <pri> <flow_label>                  % header
<payload len>     <next hdr>  <hop limit>
<source address (IPv6)>
(128 bit)
<destination address (IPv6)>
(128 bit)
<payload> (lunghezza variabile)           % payload
```

- Il campo **version** è su 4 bit, come per IPv4. Seguono campi di **priorità** è identificatori di **flusso** (il concetto di flusso non è ben definito);
- Al contrario di IPv4, allochiamo la **lunghezza del payload** (anziché dell'intero datagramma). Questo anche per il fatto che IPv6 non prevede *opzioni*, che in IPv4 potevano variare la lunghezza dell'header,
- Il **checksum** non viene introdotto, in quanto andava ricalcolato ad ogni hop (si associa un **limite di hop**, quello che in IPv4 chiamavamo *time to live*, che va aggiornato ad ogni hop);
- **Next header** rappresenta il tipo dell'header di livello 4 incapsulato;
- Come sempre, segue il **payload** vero e proprio.

Notiamo che, oltre al checksumming e le opzioni, IPv6 rimuove anche il supporto alla *frammentazione*. Deleghiamo infatti tale compito al livello superiore, e ci limitiamo a scartare i datagrammi troppo grandi per il livello datalink.

Notiamo che anche il protocollo *ICMP* viene di conseguenza aggiornato, per fornire messaggi di notifica su quali datagrammi vengono scartati per dimensioni troppo grandi.