

# Gestione di Reti

Federico Matteoni



# Indice

1	lezione 2	5
---	-----------	---

1.1	Ethernet . . . . .	5
-----	--------------------	---

2	lezione 3	7
---	-----------	---

## Introduzione

...

# Capitolo 1

## lezione 2

Livello 2 consente di identificare un device sulla rete. In tutte le reti c'è la necessità di identificare la porta di rete. Ogni dispositivo ha almeno un'interfaccia di rete: loopback, che consente di far comunicare processi di rete sulla stessa macchina. 127.0.0.1 consente di parlare su stessa macchina senza trasmettere sul filo, fondamentalmente un cortocircuito.

`ifconfig` consente di vedere le interfacce di rete disponibili su unix.

Se si vuole gestire una rete è fondamentale la standardizzazione.

Output `ifconfig`. Parte degli indirizzi, no indirizzo hw su loopback perché il traffico non esce mai (loopback sulla pila OSI è nel livello 3 Network, il MAC address è sul livello 2 Data Link, che non viene toccato da loopback). Indirizzo MAC 6 byte divisi in blocchi dai due punti. I primi 3 identificano il costruttore della scheda di rete. I successivi tre identificano la scheda di rete per il costruttore, che lo setta univocamente. Ciò garantisce univocità. Per primo blocco di tre ho 16M di dispositivi possibili. I MAC address quindi **non sono univoci**, lo sono *probabilmente*. L'univocità è fondamentale sulla stessa rete. Quindi indirizzo hw identifica univocamente device sulla rete locale. divisi in due blocchi, il primo identifica costruttore della scheda di rete.

Qualsiasi dispositivo ha indirizzo hw diverso per ciascuna porta.

### 1.1 Ethernet

Ethernet è un cavo seriale, trasmissione e ricezione. Mezzo seriale. Un filo.

Quando si mandano dati non posso tutti insieme ma man mano. Non c'è collisione perché ricezione e trasmissione sono su due fili separati.

Pacchetti inviati nel tempo sul filo. Vengono distinti tra loro dal **preamble**. Pacchetti inviati in una direzione: preambolo, destinazione, sorgente, tipo dei dati, dati effettivi, padding (per rendere pacchetto di 64 se pacchetto è troppo corto), CRC.

Quindi per spedire pacchetto necessito di indirizzi (chi voglio e chi sono) e cosa mandare. chi sono lo so, è scritto nella scheda. Voglio conoscere indirizzo di chi voglio.

Alla connessione del cavo, se DHCP manda fuori pacchetto per richiesta quindi switch lo impara, se IP statico manda pacchetto ARP quindi switch lo impara.

MAC address randomizzato per privacy, spesso e volentieri sui dispositivi mobili.

Possibile più di un utente sulla stessa rete con soliti indirizzi, apparati avanzati se ne accorgono.



# Capitolo 2

## lezione 3

un pacchetto è interamente creato dal computer, quindi "non ci si può fidare"

Bisogna andare a livello fisico e autenticare, un po' come chiedere la carta d'identità. Metter in atto meccanismi che impediscano di inibire riconoscimento della sorgente.

802.1x permette di entrare in rete. Se configurato, il device prima di entrare in rete espone delle credenziali (utente, password, protocollo autenticazione...)

Da quel momento in poi **allegato** al pacchetto c'è il mio nome, ma le informazioni di autenticazione non fanno parte del pacchetto: pacchetti creati quando non c'era preoccupazione e interesse in fattori di sicurezza delle trasmissioni.

L'informazione non è parte del pacchetto ma lo riconosce in qualche altro modo il device e **rimane nel device** (Access Point). Ciò non serve per autenticazione fisica sul cavo: so che sei tu su questo cavo. Ma è necessaria per autenticazione su mezzi condivisi (wifi).

Su router MAC cambia ad ogni hop (ethernet comunicazione punto-punto), IP cambia solo se c'è NAT. Le parti da lv 3 in su non cambiano (a meno di frammentazioni...)

Robustezza delle reti si fa tramite la ridondanza. Tipico mettere più strade per spedire il traffico: load balancing.

Vale sia per corrente elettrica che per traffico di rete.

### 2.0.1 Aree funzionali

FCAPS per gestire qualsiasi sistema, da giochi a sistemi di rete

Fault Management: error detection, isolation and repair

Se qualcuno rileva malfunzionamento (riempito disco, ram, sovraccarico CPU...) lo deve notificare

Configuration management: devo sapere com'è configurato il sistema. leggere config importante così che le app si basano sulle API comuni e funzionerà. Fondamentale capire configurazione perché permette di definire l'amministrazione, servizi..., possibile riconoscere anche le adiacenze e "questo filo qui va su questa porta qua". che impatto ho se stacco questo cavo, o si rovina? Informazioni sufficienti per amministrare la rete

Account management: rilevare il consumo di risorse

Performance management: efficienze e statistiche, performance di sistema sia lato utente sia lato fornitore. Per l'utente è riuscire ad usare la rete, per l'operatore è il giusto compromesso tra investimento sul mezzo e contentezza utente.

Security management: assicurarsi che ciò che uno fa è effettivamente possibile farlo, autoproteggendosi perché con le reti odierne posso intasare rete (volente o no) e quindi intasare internet, provocando danni

### 2.0.2 Interagire con management object

primitive

Quando faccio richiesta protocollo: richiesta – risposta

Contenuto richieste varia durante il transito aggiungendo determinate informazioni. Es. SMS durante il transito aggiunge numero mittente per poter comunicare a destinatario chi inviava.

### 2.0.3 Servizi

Confermati: faccio richiesta mi aspetto risposta. Telegram/Whatsapp

Non confermati: faccio richiesta e fine. SMS