

Si sta estendendo, recentemente, alle reti un approccio di virtualizzazione delle risorse simile a quello che avviene nei data center ([laC](#)). **Quindi se abbiamo un problema al server 55 non deve per forza arrivare un omarino con una matassa di cavi a braccio.**

Vi sono al momento due approcci principali:

- **NFV: Network Function Virtualization**
  - Ha origine nelle reti di grossi ISP;
  - Ha l'obiettivo di ridurre i costi [OPEX e CAPEX](#);
  - Fa uso di sistemi [COTS](#) per la riduzione dei costi, invece che di hw specifico;
- **SDN: Software Defined Networking**
  - Reti limitate e data center;
  - Scalabilità e flessibilità aumentate, possiamo espandere o comprimere le risorse di rete in funzione delle esigenze;
  - Infrastruttura più semplice da utilizzare;
  - Come l'NFV fa utilizzo di hw standard e non specifico.

**Ma perché sono nati questi sistemi di virtualizzazione?**

- Carichi di lavoro variabili,
- Le infrastrutture di rete fisiche sono dimensionate sulle esigenze di picco, dunque anche in casi in cui questo picco non venga mai raggiunto pago gli stessi costi;
- Importante è anche la performance isolation: in uno scenario data center multi-tenant i flussi di traffico, per quanto numerosi NON devono interferire tra loro;
- Supporto per la ridondanza, necessaria per la fault tolerance;
- Supporto per funzioni aggiuntive, come Firewall, NAT, traffic shaping..., i quali hanno API o protocolli specifici, non interoperabili;

[SDN è la soluzione](#), diversamente da come si potrebbe pensare, è un meccanismo flessibile per implementare policy complesse

## **SDN**

Funzioni di basso livello:

- Data plane → non si occupa d'altro se non di gestire i dati ("arriva il pacchetto e faccio cose);
  - Distribuito;
  - Parallelo;
  - Veloce.

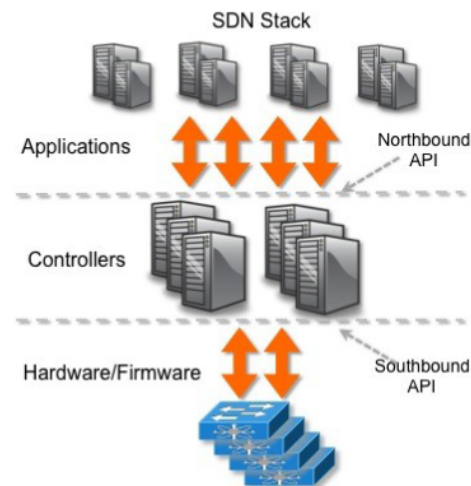
Funzioni di alto livello:

- Control plane
  - Centralizzato;
  - Flessibile;
  - Supporta policy complesse.

## Interfacce

Tra **control plane** e **applications** vi è una northbound interface che fa uso di REST API-

Tra **control plane** e **data plane** vi è una southbound interface che fa uso di **OpenFlow** come standard de facto.



## Data Plane

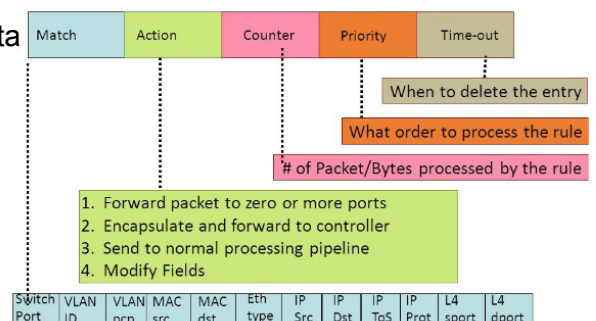
In un router tradizionale il data plane ha un inoltro basato su tabelle di routing, match su NetID e next hop.

Nel caso di SDN questa tabella viene estesa e generalizzata → **OpenFlow**

In **OpenFlow** ogni riga della tabella ha:

- Predicato di match, **non solo IP!**;
- Azione, che si attiva solo se avviene il match;
- Contatori, quante bytes sono processati da questa regola;
- Priorità;
- Time-Out, se la regola non viene usata per un quanto di tempo deciso preventivamente, può essere eliminata.

### OpenFlow: Anatomy of a Flow Table Entry



## Predicato di match

- Supporto per deep packet inspection
- Grande Flessibilità
- Ma...
- Possibili problemi di prestazioni dovuti all'enorme dimensione delle tabelle
- L'emulazione software potrebbe danneggiare le prestazioni

## Azioni

- Inoltro → azione di default del router;
- Drop → azione di default di un firewall;
- Modifica → gestione VLAN, NAT, redirectione verso altri nodi...;
- Invio al controller → usato per sollevare eccezioni, consente la modifica delle tabelle openflow;
- Metering & Shaping → Aggiornamento counter, riordino pacchetti per non eccedere il TR o per garantire una certa QoS.

## Counter

- Numero attivazioni;
- Numero di pacchetti con match;
- Volume di dati associati alla regola.

## Southbound Interface (CP & DP)

Control Plane e Data Plane possono scambiarsi diversi tipi di messaggio:

- Messaggi da controller a switch;
- Messaggi asincroni (switch → controller);
- Messaggi simmetrici;
- Messaggi legati allo stato e alla configurazione dello switch;
- Messaggi di invio pacchetti;
- Messaggi ping per il controllo connettività

## Control Plane

Sono disponibili vari controller, i principali sono ONOS e OpenDaylight.

Hanno caratteristiche architetturali comuni:

- Livello comunicazione (int. southbound):
  - Gestione dell'int. southbound;
  - Supporto protocolli per data plane, tipo OpenFlow;
- Interfaccia applicazioni (int. northbound):
  - Gestione dell'int.northbound;
  - Fa uso di API REST dunque è basato su HTTP;
  - Espone strutture dati che mostrano stato della rete e statistiche;
  - Codifica lo spazio degli URL;
  - Consente la manipolazione delle strutture per inviare comandi all'infrastruttura;
  - Consente di integrare la gestione della rete nelle altre logiche di gestione di un DC;
  - Supporto per interazione con altri controller;
- Gestione rete:
  - Repository dello stato della rete SDN;
  - Visione globale della rete, non limitata al singolo dispositivo.