



**SDN / NFV** → soluzione per ambienti dinamici

↓  
Software  
Defined  
Network

↳ si ha la necessità di adattarsi a nuovi protocolli e standard

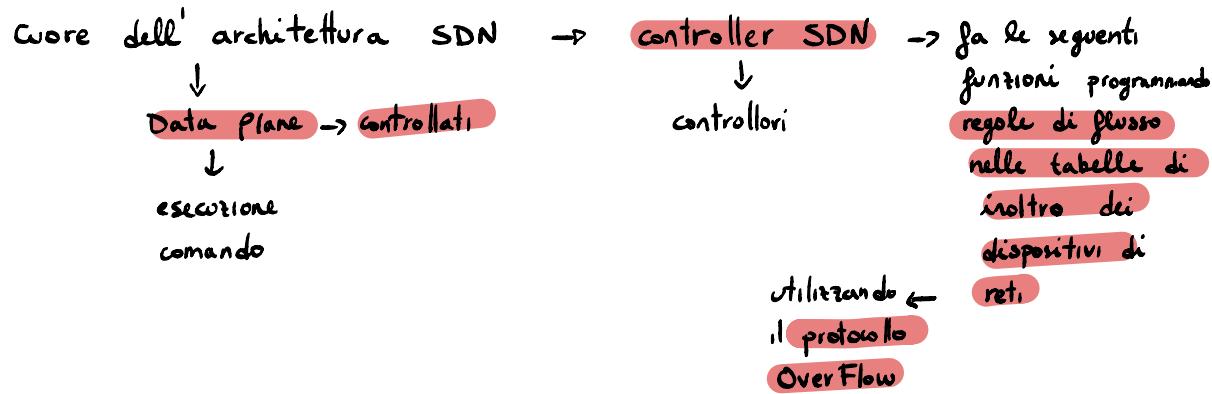
→ interfaccia programmabile

→ Protocollo OverFlow

- SDN è un'architettura di rete che separa piano controllo (CP) da piano dati DP, consentendo una gestione centralizzata della rete tramite software.

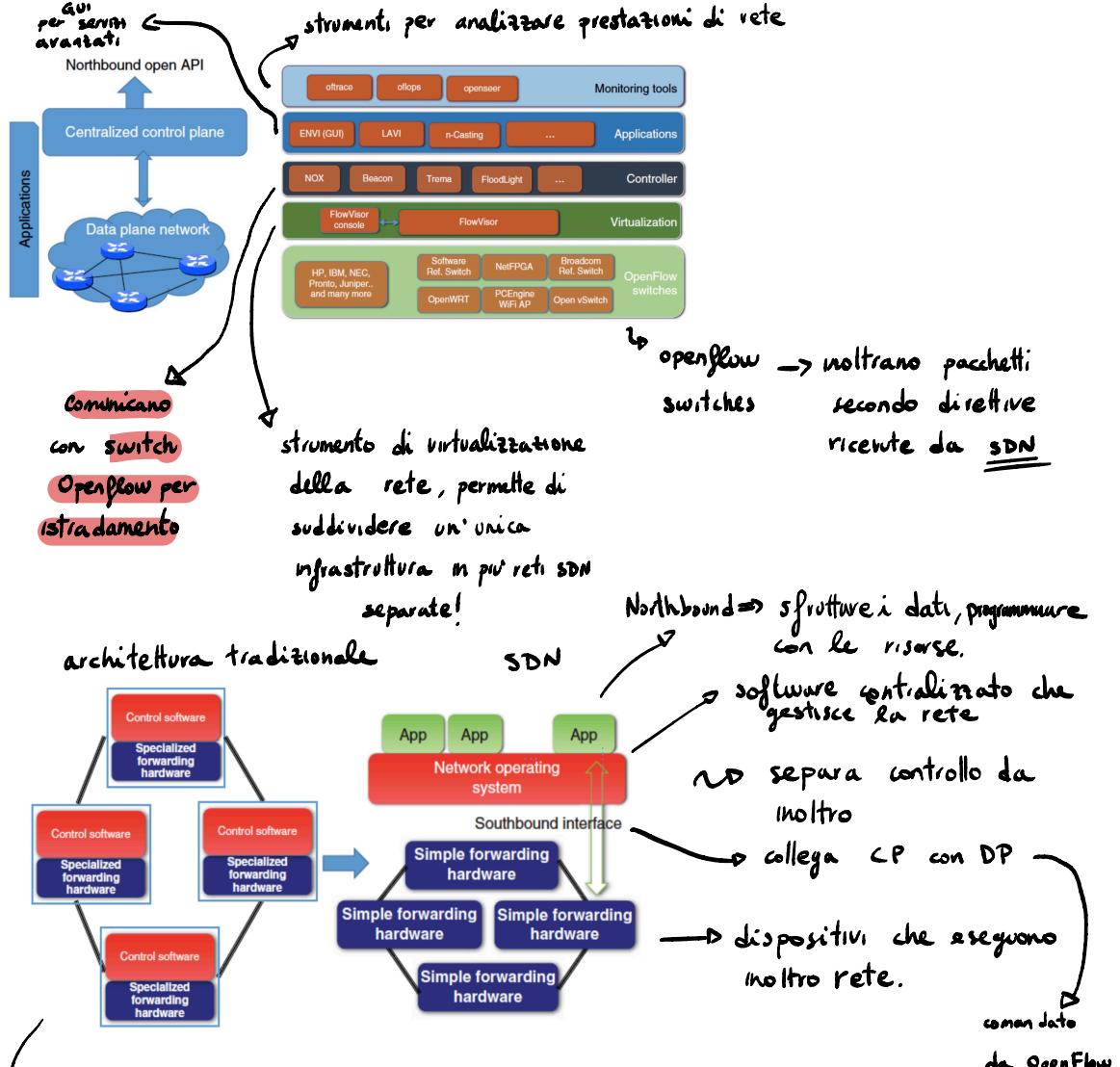
- Viene utilizzato per alcune funzioni di rete:

- ~ bilanciamento del carico
- ~ QoS
- ~ recupero rapido da guasti



SDN → migliorano velocità servizio.

↳ innovazione. (data dalla programmabilità della rete)

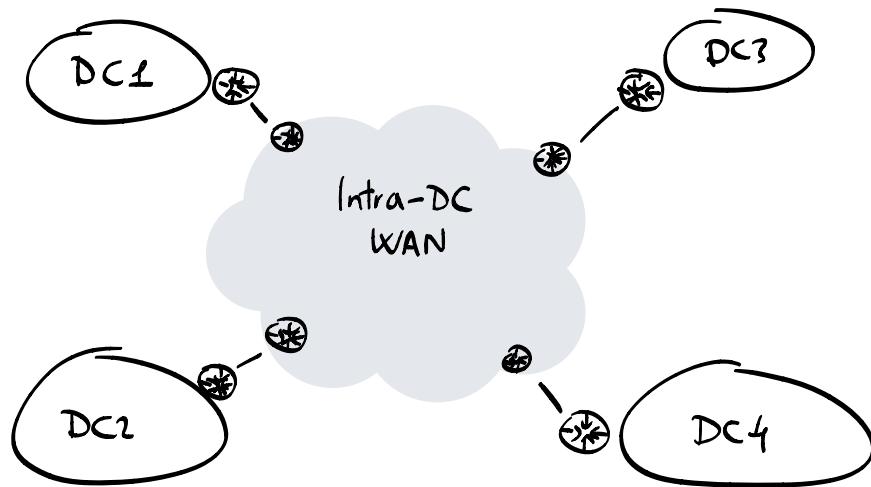


→ architettura tradizionale  
dei sistemi distribuiti  
↓  
il software è distribuito  
sui dispositivi della rete.

Casi d'uso:  
 ↗ Reti di centri dati  
 ↗ Domini di accesso/aggregazione delle reti pubbliche di telecomunicazione  
 ↗ Cloud Bursting: aggrega Data Center pubblici e privati

↳ controllo di WAN

# WAN inter - DATA Center

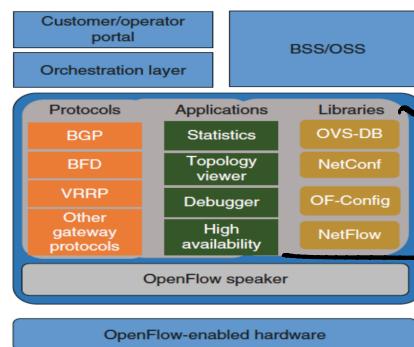


Controller SDN → capacità di osservare e controllare una rete

- **north bound** ⇒ fornisce alle applicazioni la base informativa di rete (NIB - Network Information Base)
  - ↓
    - applicazioni utilizzano queste per prendere decisioni di gestione
- software controller viene eseguito su un server
- **south bound** ⇒ controllore installa istruzione sullo switch indipendentemente dal hardware

come è formato?

si occupa dei protocolli di rete tradizionali



→ supporto interface verso sud

→ per utilizzare informazioni della rete

il pacchetto va ad uno switch → header viene confrontato con criteri definiti dal controller

matchato?  
si  
aumento  
counter o  
azioni corrispondenti

no

viene inoltrato  
al controller

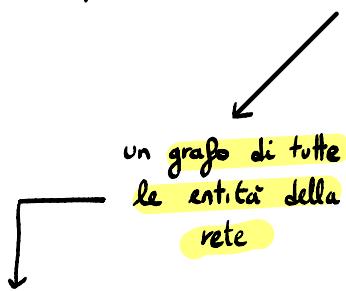
ci penserà  
lui a capire  
che fare

↳ flow entries

a)  
stanno dentro il  
flow table nello  
switch

Facendo così, il controller crea una visione della rete, affinché riesca ad inoltrare il traffico.

Importante componente → SDN NIB → raccoglie, organizza e mantiene una visione globale della rete.



Network Information Based

ogni entità della NIB è una coppia chiave valore

Modelli di Distribuzione:

back up ↑

1. hot - standby: il controller master è protetto dal modello hot standby

↓  
si fanno due istante, una master e una slave

↓ quando c'è guasto → il controllo della rete viene passato alla istanza di standby

## 2. Distribuito NIB:

il controller della rete  $\Rightarrow$  un cluster di controller  $\rightarrow$  info della rete  
 $\downarrow$   
ognuno controlla  
una parte di rete  
vengono replicate  
su più controller

## 3. ibrido:

combinazione dei due modelli:

- ~ le info vengono replicate per avere disponibilità
- ~ i controller sono raggruppati in cluster e ogni cluster ha un master e un istanza di standby.

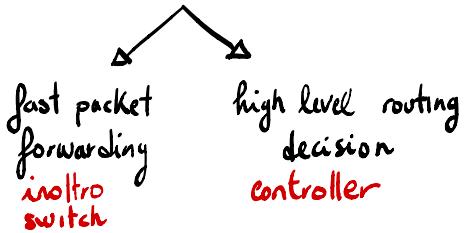
Negli switch  $\rightarrow$  Problemi nella scrittura delle regole?

Openflow  $\leftarrow$  si fanno conflitti nella scrittura  
dice:  
"il primo regna"  
 $\downarrow$   
di più applicazioni SDN.  
  
 $\hookrightarrow$  ogni applicazione  
ha il proprio obiettivo  
 $\downarrow$   
si ha conflitto  
tra app  
conflitti per banda o  
spegnere/accendere dispositivi  
di rete

**Openflow**  $\rightarrow$  serve a disaccoppiare intelligenza nel routing (SW)  
e l'inoltro (HW)

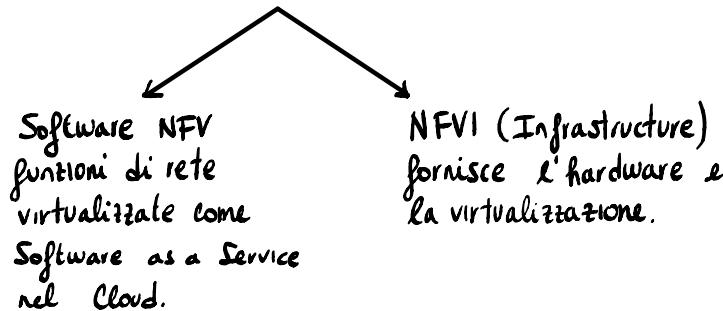
$\rightarrow$  nella visione classica  $\rightarrow$  router classico ha nello stesso dispositivo  
 $\rightarrow$  mentre in Openflow

- $\rightarrow$  high level routing decision affidata al controller
- $\rightarrow$  percorso dati affidato a switch



## NFV (Network Function Virtualization)

- trasformazione delle reti di telecomunicazione, riducendo i costi e aumentando flessibilità.
  - ↳ permette di virtualizzare funzioni di rete anziché eseguirle su hardware dedicato.
    - ↳ NAT, FIREWALL, IDS
- si basa sul principio del **cloud computing**
- questi servizi di rete **vengono eseguiti** su hardware generico e virtualizzato.



- componenti:
  - ~ NFV Orchestrator: gestisce il ciclo di vita delle VNF e le sue risorse
  - ~ VNF Manager: gestisce la configurazione, monitoraggio e la scalabilità delle VNF
  - ~ VIM: controlla infrastruttura fisica e virtuale

La VNF **riduce i costi hardware**, maggiore flessibilità nello sviluppo e nell'implementazione di nuovi servizi, Scalabilità e adattabilità alle esigenze di rete.

**NFV MANO**  $\Rightarrow$  e' una componente chiave dell' architettura NFV, il suo scopo e' gestire e orchestrare le risorse virtualizzate e le VNF all'interno dell'infrastruttura.



formato da 3 componenti fondamentali:

$\rightarrow$  **Orchestrator NFV MANO:**

- gestione del ciclo di vita dei servizi di rete virtualizzati.
- coordina allocazione risorse tra funzioni di rete.
- Registra e gestisce i servizi nella rete virtualizzata
- può concatenare più VNF per servire più servizi.

$\rightarrow$  **VNF Manager:**

- controlla ciclo di ogni singola VNF
- scalare/ridurre VNF

$\rightarrow$  **VIM:** controlla e gestisce i componenti NFVI per ospitare vari servizi di rete.

Controlla dispositivi di rete (switch, router e connessioni)

Controlla le risorse di computing (CPU, Memoria, storage)

**VNF Placement Problem:**

posizionamento ottimale delle funzioni di rete virtuali all'interno dell'infrastruttura garantendo efficiente, prestazioni e costi ridotti.

un flusso di traffico

che attraversa una catena

di funzioni virtuali di rete

$\rightarrow$  devono passare attraverso una catena di funzioni di rete virtuale

## Problema dello zaino



il problema tratta proprio del dover decidere:

**Scalabilità:**  
Capacità del sistema di adattarsi all'aumento della domanda di traffico o della crescita di rete.

Io

- ~ dove posizionare le VNF all'interno dei server disponibili.
- ~ quale percorso far eseguire ai pacchetti tra le VNF.
- ~ come bilanciare il carico tra le istanze delle NVF.
- ~ come minimizzare i costi di utilizzo delle risorse di rete.

## SDN + NFV in IoT:

- SDN e NFV permettono insieme di realizzare reti IoT flessibili e scalabili.
  - ↳ supporti a reti eterogenee
  - ↳ supporti nella mobilità dei dispositivi IoT
- virtualizzazione della rete d'accesso e core (principale) a cui dispositivi IoT si collegano
- creazione di piattaforme cloud/fog/edge per gestire sensori e attuatori.
- standardizzazione API per il modello Sensing/Actuating as a Service (SAaaS)
  - ↳ non importa hardware del sensore

# Architettura SDN per IoT

- Device Layer:
  - sensori raccolgono dati
  - attuatori eseguono azione
- Communication Layer
  - switch e gateway abilitati a SDN per inoltro controllato dei dati
- Computing Layer
  - controller SDN distribuiti per impostare tabelle d'inoltro
  - nodi e archiviazione per gestire la rete e la Sicurezza
- Service Layer:
  - accesso agli sviluppatori di app IoT tramite Northbound