

# **PARTE 6**

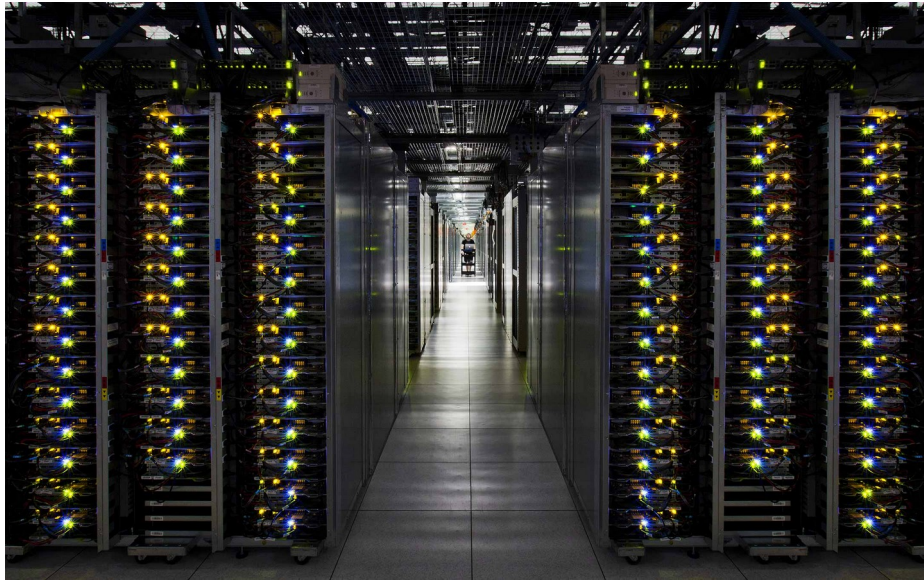
# **SOFTWARE DEFINED**

# **NETWORKS**

## **Modulo 1**

# **Overview di un data center Cloud**

# *Cosa c'è dietro?*



# ***Visione di un data center***

- **Data center molto complessi**
  - Tanti host
  - Ogni host ospita decine/centinaia di VM
  - Uso do container
- **Workload complesso**
  - Continui aggiornamenti
  - Nuove VM/Container
  - Sistemi scalabili on-dmand

# Non solo

## Scale Drives Architecture

**BUILD**  
& topology

**CONNECTIVITY**  
& routing

**SLA**  
& app visibility

**0.25 Million+**  
links per fabric\*\*

**10 Million +**  
routing rules per fabric\*\*

**3.5 Billion**  
searches per day

**10,000+**  
switches per fabric\*\*

**30,000 burst**  
rule updates within 1 min\*\*

**300 hours**  
of video uploaded every  
minute



**\*\*Typical numbers seen in large data center fabrics**



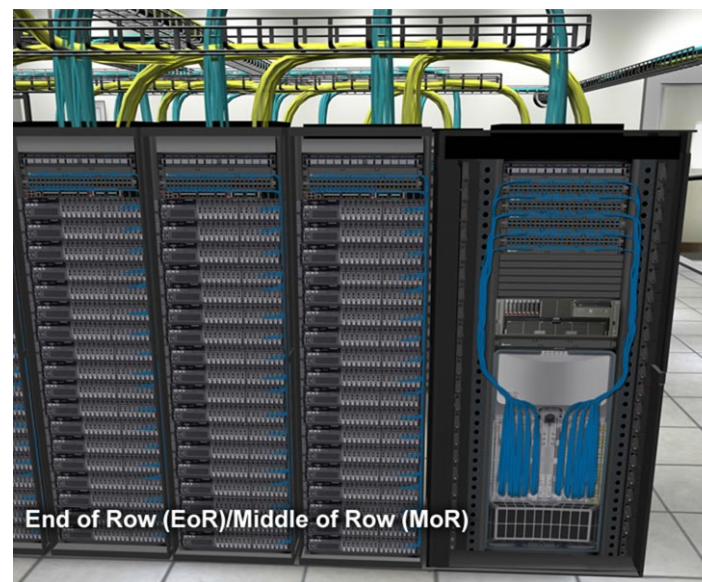
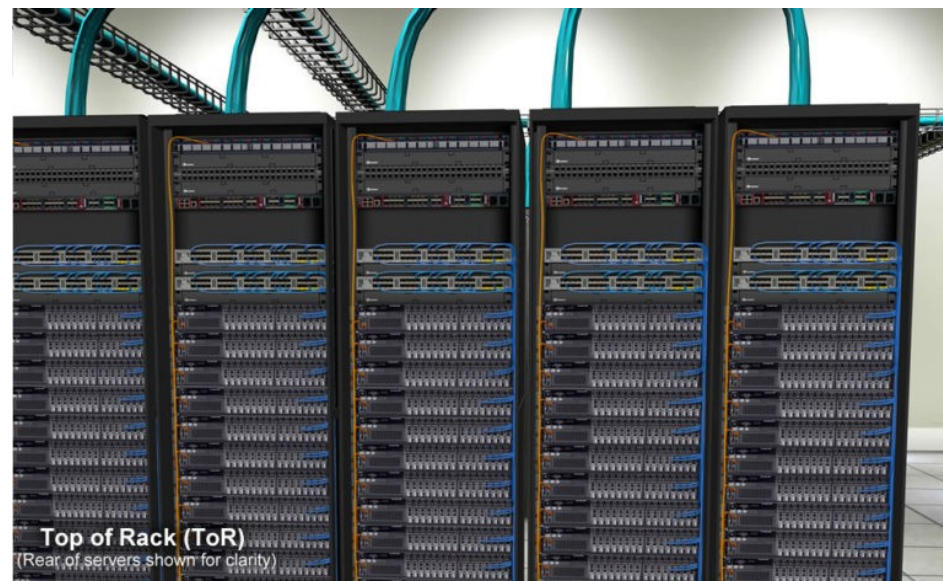
# *Struttura di un data center*

- **Organizzazione gerarchica dell'HW**
- **Blade system**
- **Blade racchiusi in chassis**
- **Chassis organizzati in rack**
- **Rack contiene anche elementi di rete**



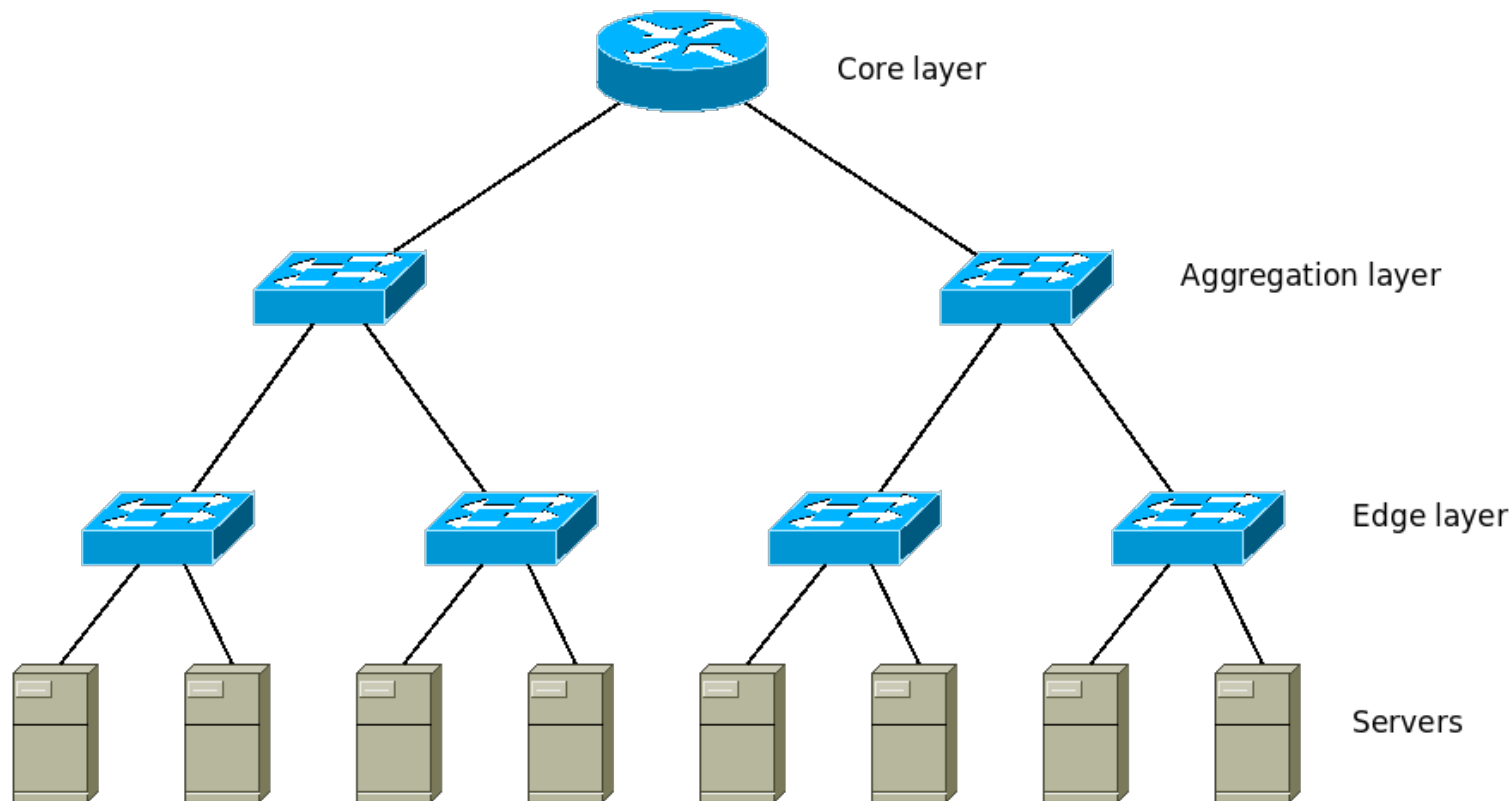
# Struttura di un data center

- **Elementi di switching**
- **Top-of-Rack (ToR)**
- **End of Row (EoR)**



# Struttura di rete

- **Organizzazione della rete**
  - Struttura gerarchica
  - Tipicamente 3 o 4 livelli
- **Topologia ad albero**





# Struttura di rete

- **Edge/Access layer**
  - Connette gli host
  - Tipicamente composto da switch ToR
- **Aggregation layer**
  - Connette le zone del data center
  - Funzioni avanzate (QoS, Traffic shaping, path ridonanti)
- **Core Layer**
  - Backbone del data center
  - Connette con l'esterno
  - Fondamentale per garantire la disponibilità dei servizi

# ***Nuove linee di evoluzione delle reti***

- **I data center adottano tecniche di virtualizzazione delle risorse di calcolo**
- **Tale approccio si sta estendendo anche alle reti**
- **NFV: Network Function Virtualization**
- **SDN: Software Defined Networking**

- **Origine:**
  - Soluzione nata nelle reti di grossi ISP
- **Obiettivo:**
  - Riduzione di costi OPEX e CAPEX dei sistemi di rete
- **Metodologia:**
  - Uso di sistemi COTS per le funzioni di reti più sofisticate al posto di hardware dedicato
- **Area di interesse:**
  - Switch e router

- **Origine:**
  - Soluzione nata in reti limitate e maturata nell'ambito di data center
- **Obiettivo:**
  - Supporto per SDDC (Software Defined Data Center)
  - Migliorare orchestrazione in sistemi Cloud
- **Metodologia:**
  - Separazione funzioni di rete di basso livello da funzioni più sofisticate
  - Routing protocol vs. Packet forwarding



- **Area di interesse:**
  - Switch e router
- **Standard:**
  - ForCES
  - OpenFlow

→ **Focus di questa lezione**

## **Modulo 2**

# **Motivazioni**

# ***Necessità di condividere risorse***

- **Difficoltà nella gestione della rete a fronte di carichi di lavoro variabili**
  - Riconfigurazione: processo complesso
  - Scambio di messaggi tra dispositivi autonomi (es. router)
- **Infrastrutture di rete dimensionate su esigenze di picco**
  - Costi elevati
  - Risorse sotto-utilizzate (Google B4)



- **Serve un meccanismo flessibile per implementare policy complesse**

# *Performance isolation*

- **Flussi di traffico non dovrebbero interferire tra di loro**
- **Scenario critico: data center multi-tenant**
- **Soluzioni oggi disponibili**
  - VLAN + Switch managed
  - Traffic shaping
- **Difficile interoperabilità tra sistemi diversi**
  - Problema con policy complesse
  - Scenario critico: data center con migrazione di VM e riconfigurazione della rete
- **Serve un meccanismo flessibile per implementare policy complesse**



# ***Supporto per la ridondanza***

- **Ridondanza necessaria per fault tolerance**
- **Algoritmi per spanning tree**
- **Scarso supporto per altri algoritmi (e.g. load balancing)**
- **Scarsa interoperabilità con sistemi volti a gestire altre esigenze**
  - Resource sharing
  - Performance isolation
- **Serve un meccanismo flessibile per implementare policy complesse**

# Supporto per funzioni aggiuntive

- **Funzioni aggiuntive**

- Firewall
- NIDS
- NAT
- Traffic shaping



- **Implementate con middleboxes**
- **Necessità di integrare tali funzioni con funzioni base di rete**
  - Routing
  - VLAN/Spanning tree
- **Funzioni non integrate**
- **Serve un meccanismo flessibile per implementare policy complesse**

# ***Il problema del management***

- **Necessità:**
  - Adattarsi a scenari fortemente dinamici
  - Agire in modo rapido e automatico
- **Problema: Mancanza di integrazione**
  - Ogni dispositivo ha API/protocolli specifici di solito non interoperabili
- **Problema: Mancanza di stabilità**
  - Algoritmi distribuiti possono convergere a soluzioni di routing differenti a seguito di crash [Google B4]
- **Serve un meccanismo flessibile per implementare policy complesse**

***Introducing...***





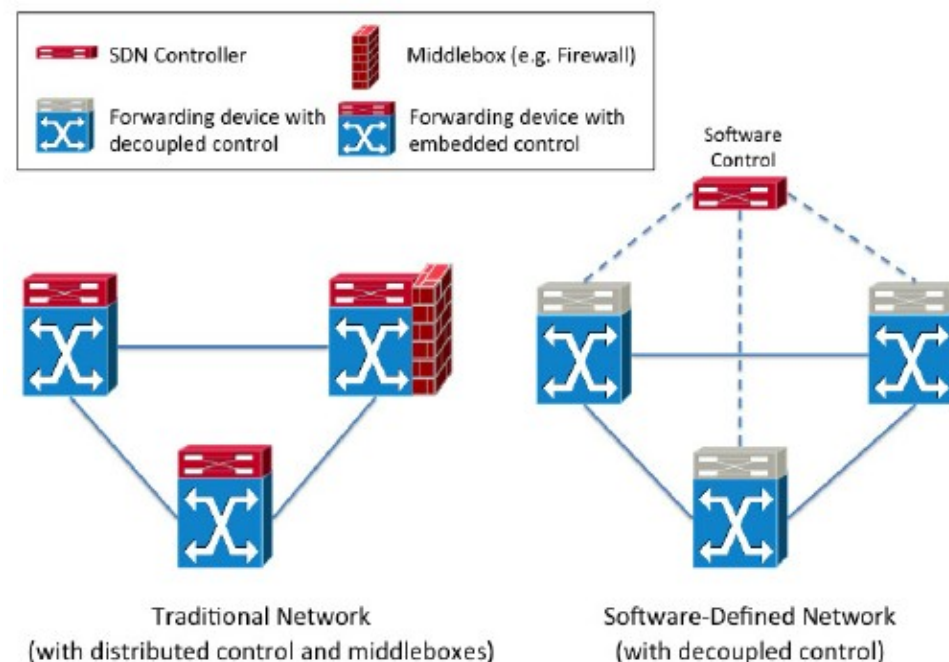
# SDN: Funzioni

- **Funzioni di basso livello**

- Data plane
- Veloce
- Distribuito

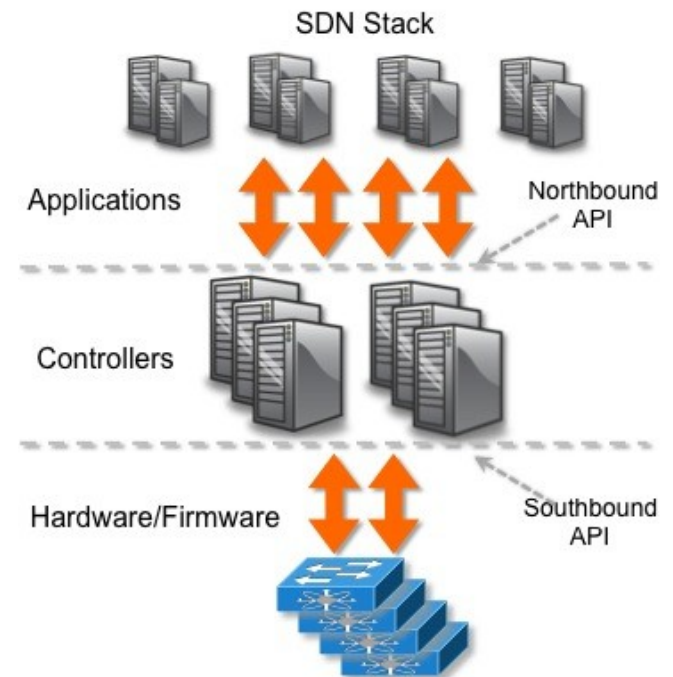
- **Funzioni di alto livello**

- Control plane
- Centralizzato
- Flessibile
- Supporta policy complesse



# ***SDN: Definizione di interfacce***

- **Tra control plane e applicazioni**
  - Northbound interface
  - REST API
  - OSPF/BGP/...
- **Tra data plane e control plane**
  - Southbound interface
  - OpenFlow  
(standard de facto)
  - ForCES




## Modulo 3 Data Plane




- **Il data plane in un router tradizionale**
  - Inoltro basato su tabelle di routing
  - Match su NetID e selezione next hop

Command Output

 Route Display...

**IPv4 Routing Table**

Destination	Gateway	Genmask	Metric	Ref	Use	Interface	Type	Flags
127.0.0.1	127.0.0.1	255.255.255.255	1	0	0	lo	Static	UP,Gateway,Host
192.168.2.0	0.0.0.0	255.255.255.0	0	0	0	bdg34	Dynamic	UP
192.168.1.0	0.0.0.0	255.255.255.0	0	0	0	bdg1	Dynamic	UP
192.168.1.0	192.168.1.1	255.255.255.0	1	0	0	bdg1	Dynamic	UP,Gateway
172.16.15.0	0.0.0.0	255.255.255.0	0	0	0	eth1	Dynamic	UP
10.10.0.0	0.0.0.0	255.255.0.0	0	0	0	eth1	Static	UP
10.10.0.0	10.10.0.200	255.255.0.0	1	0	0	eth1	Static	UP,Gateway
127.0.0.0	0.0.0.0	255.0.0.0	0	0	0	lo	Dynamic	UP
0.0.0.0	10.10.1.1	0.0.0.0	0	0	0	eth1	Dynamic	UP,Gateway

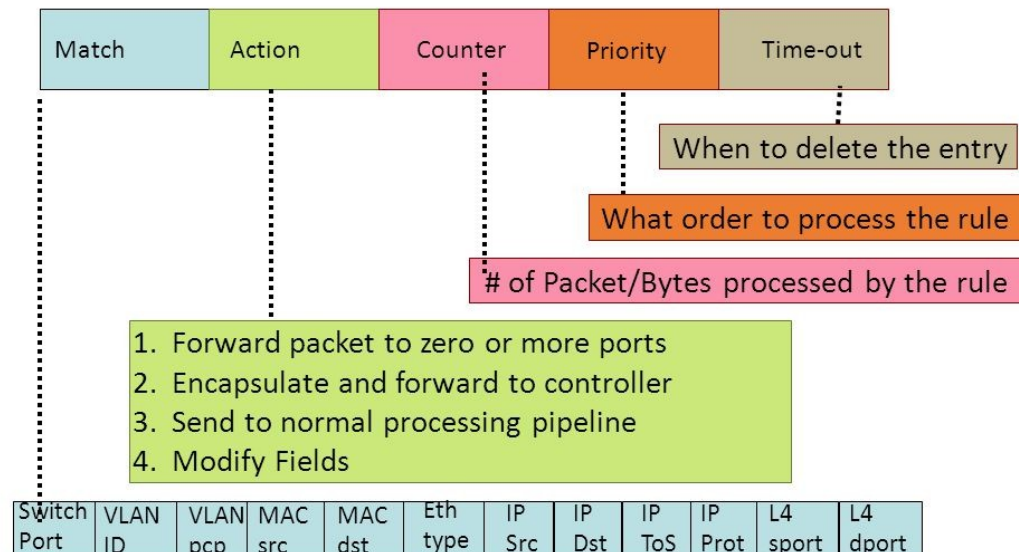




# Applicazione al caso SDN

- **Estendo e generalizzo la tabella di routing**
- **Il caso OpenFlow**
- **In OpenFlow ogni riga della tabella ha**
  - Predicato di match
  - Azione
  - Contatori
  - Priorità
  - Time-out

OpenFlow: Anatomy of a Flow Table Entry



# ***Predicato di match***

- **Opera su numerosi campi (12 - 41 campi a seconda delle versioni)**
  - Porta di ingresso del device
  - VLAN ID + priority (PCP)
  - MAC src + dst
  - Type Eth
  - IP src + dst
  - IP proto
  - IP ToS
  - TCP/UDP src + dst
  - ...

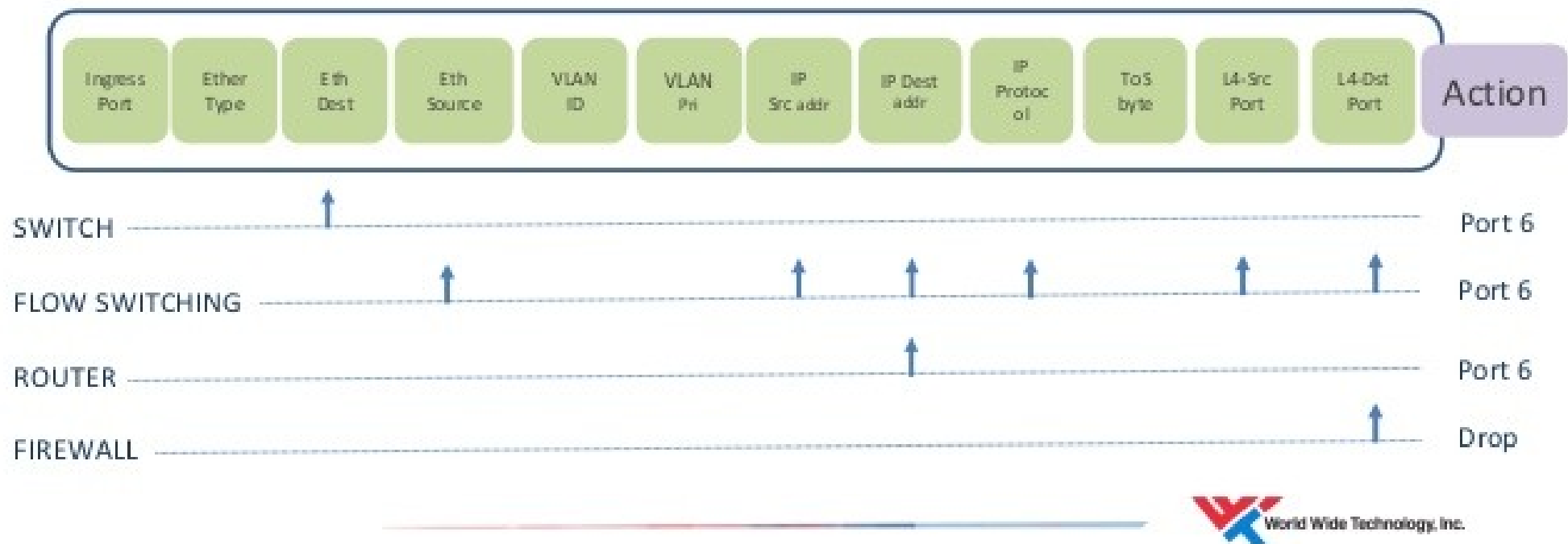
# ***Predicato di match***

- **Supporto per deep packet inspection**
  - Considera campi di livelli differenti dello stack
- **Grande flessibilità**
  - Match esatti ma anche di tipo range/prefix
  - Il numero di campi supportati cresce ad ogni versione dello standard
- **Possibili problemi di prestazioni**
  - Dimensione delle tabelle
  - Problemi in alcune operazioni di match
- **Emulazione software danneggia le prestazioni**

# A cosa serve il matching

## Flexibility in Matching

- The ability to match on Layer-1 through Layer-4 allows the switch to behave like a variety of network devices



- **Azioni supportate**
  - Inoltro
  - Drop
  - Modifica di pacchetto
  - Invio al controller
  - Metering & shaping
  - Altro...
- **Inoltro**
  - Come azione di default di router
- **Drop**
  - Azione di default di un firewall

- **Modifica di pacchetto**
  - Gestione VLAN
  - Implementazione di NAT
  - Redirezione verso altri nodi
  - load balancing in cluster
  - transparent proxy
- **Invio al controller**
  - Usato per sollevare situazioni non previste dalle tabelle locali
  - Consente al controller di modificare le tabelle OpenFlow



- **Metering & shaping**
  - Aggiornamento contatori
  - Pacchetti riordinati per non eccedere traffic rate o per garantire QoS
- **Altre azioni**
  - Gestione gerarchica delle tabelle
  - Azione: inoltro a tabella secondaria
  - Azioni meno probabili sono inserite in tabelle secondarie
  - Fatto per motivi prestazionali
  - Alcune tabelle secondarie sono implementate in SW

- **Per ogni regola sono considerati**
  - Numero di attivazioni
  - → Numero di pacchetti con match
  - Volume di dati associati alla regola
- **Le azioni possono essere usate per azzerare i contatori periodicamente**
  - Calcolo di frequenza di attivazioni

# ***Southbound interface***

- **Messaggi controller → switch**
- **Messaggi asincroni (switch → controller)**
- **Messaggi simmetrici**
- **Messaggi legati a operazioni di:**
  - Richiesta descrizione switch
  - Lettura/scrittura stato switch
  - Lettura/scrittura configurazione switch
  - Invio di pacchetti
  - Controllo connettività (ping)

# ***Messaggi Controller → Switch***

<b>Message</b>	<b>Description</b>
Features	Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities.
Configuration	Set and query configuration parameters. Switch responds with parameter settings.
Modify-State	Add, delete, and modify flow/group entries and set switch port properties.
Read-State	Collect information from switch, such as current configuration, statistics, and capabilities.
Packet-out	Direct packet to a specified port on the switch.
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.
Role-Request	Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous-Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers.

# ***Messaggi Asincroni e Simmetrici***

<b>Message</b>	<b>Description</b>
Packet-in	Transfer packet to controller.
Flow-Removed	Inform the controller about the removal of a flow entry from a flow table.
Port-Status	Inform the controller of a change on a port.
Error	Notify controller of error or problem condition.

<b>Message</b>	<b>Description</b>
Hello	Exchanged between the switch and controller upon connection startup.
Echo	Echo request/reply messages can be sent from either the switch or the controller, and they must return an echo reply.
Experimenter	For additional functions.

## **Modulo 4**

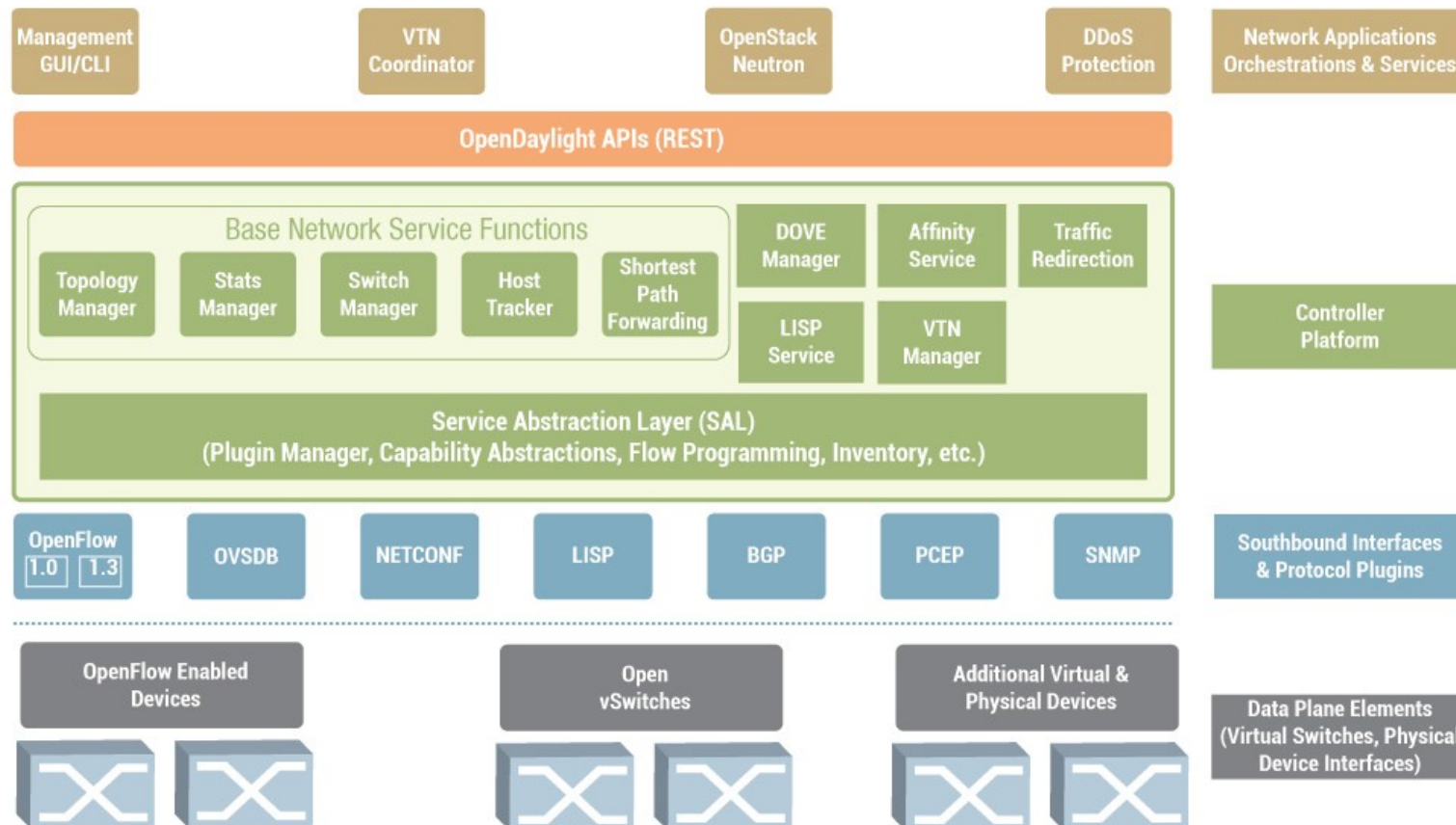
# **Control Plane**

# Visione della rete



First Code  
Release  
“Hydrogen”

VTN: Virtual Tenant Network  
DOVE: Distributed Overlay Virtual Ethernet  
DDoS: Distributed Denial Of Service  
LISP: Locator/Identifier Separation Protocol  
OVSDB: Open vSwitch DataBase protocol  
BGP: Border Gateway Protocol  
PCEP: Path Computation Element Communication Protocol  
SNMP: Simple Network Management Protocol





# Controller

- **Sono disponibili diversi controller SDN**

- Onos
- Daylight
- ...

- **Caratteristiche architetture comuni**

- Livello comunicazione (southbound)
- Interfaccia applicazioni (northbound)
- Gestione rete



# ***Livello comunicazione***

- **Gestione dell'interfaccia southbound**
- **Supporto protocolli per data plane**
  - Openflow
  - SNMP
  - ...

# ***Interfaccia con applicazioni***

- **Gestione dell'interfaccia Northbound**
- **Tipicamente si usano API REST (Representational State Transfer)**
  - Basata su HTTP
  - Espone strutture dati che mostrano lo stato della rete e statistiche
  - Codifica dello spazio degli URL
  - Consente di manipolare tali strutture per inviare comandi all'infrastruttura

# ***Interfaccia con applicazioni***

- **Consente di integrare la gestione della rete nelle altre logiche di gestione di un data center**
  - Supporto per gestire la migrazione VM
  - Integrazione con OpenStack, vSphere
- **Possibile supporto per interazione con altri controller**
  - Federazione di enti diversi
  - Supporto per reti non-SDN
  - Gestione protocolli come OSPF, BGP, ...

# ***Gestione dello stato globale della rete***

- **Repository dello stato della rete SDN**
  - Informazioni di stato su Switch, Host, ...
  - Informazioni su flussi gestiti
- **Visione globale della rete, non limitata ai singoli dispositivi**
  - Supporto per approccio centralizzato
  - es, uso algoritmo di Dijkstra per routing
  - Tali decisioni possono essere esternalizzate mediante interfaccia northbound

# **Modulo 5**

## **Conclusioni e Sfide Aperte**

- **Problemi prestazionali nelle interazioni con il controller**
  - Ogni pacchetto inoltrato al controller è soggetto ad alta latenza
  - Problema in presenza di numerosi flussi di piccole dimensioni
- **Problemi prestazionali nel data plane**
  - Emulazione software in caso di predicati di matching particolari
  - Emulazione software in caso di overflow nelle tabelle
  - Prestazioni: Switch 10 Gb/s → 20 Mb/s



- **Complessità del controller**
  - Grande capacità di implementare policy molto complesse
  - Difficoltà nel garantire prestazioni elevate
  - Controller potenziale Single Point of Failure
- **Controller ridondato/parallelizzato**
  - Fault tolerance
  - Scalabilità

