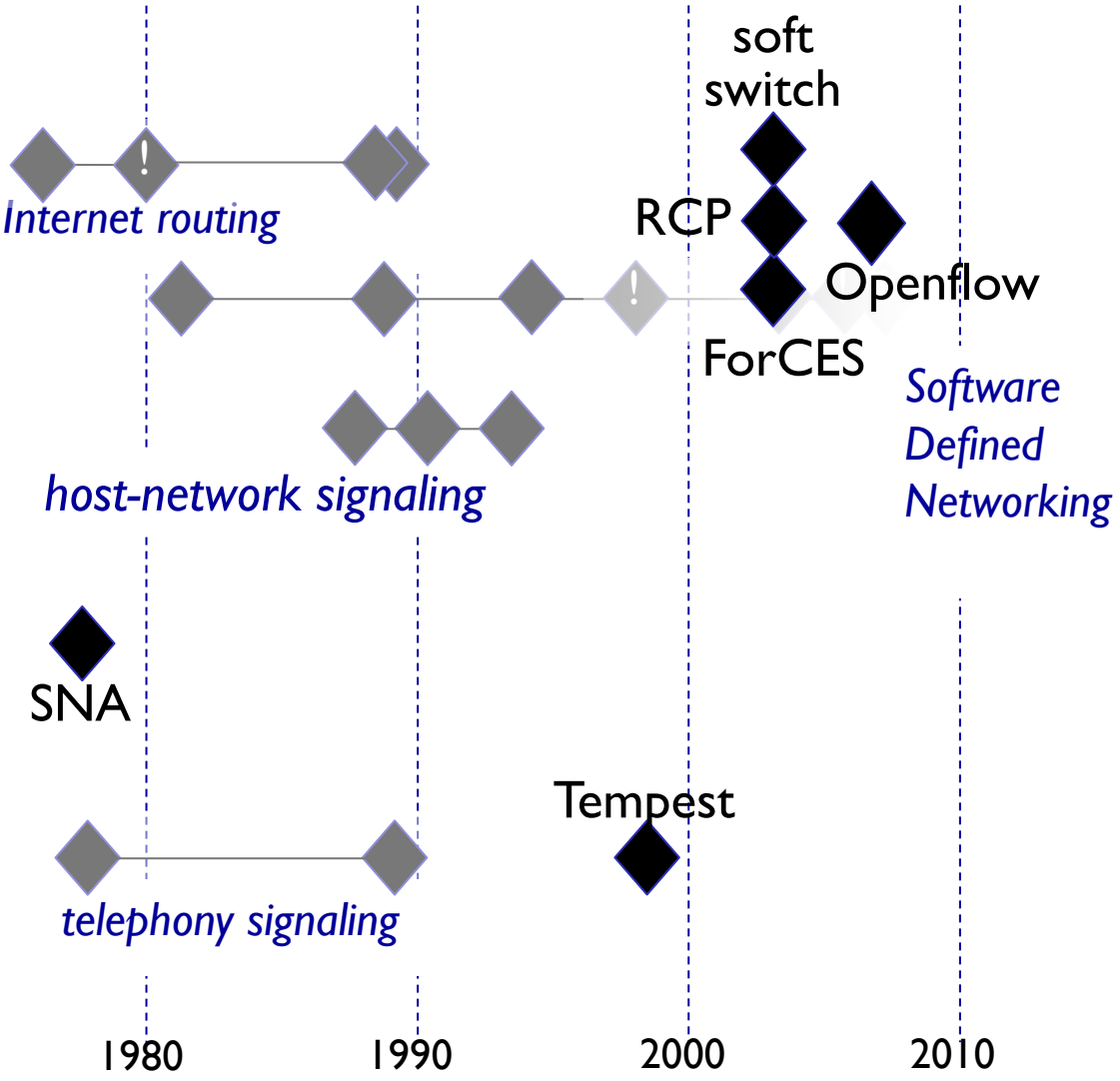
Software Defined Networking

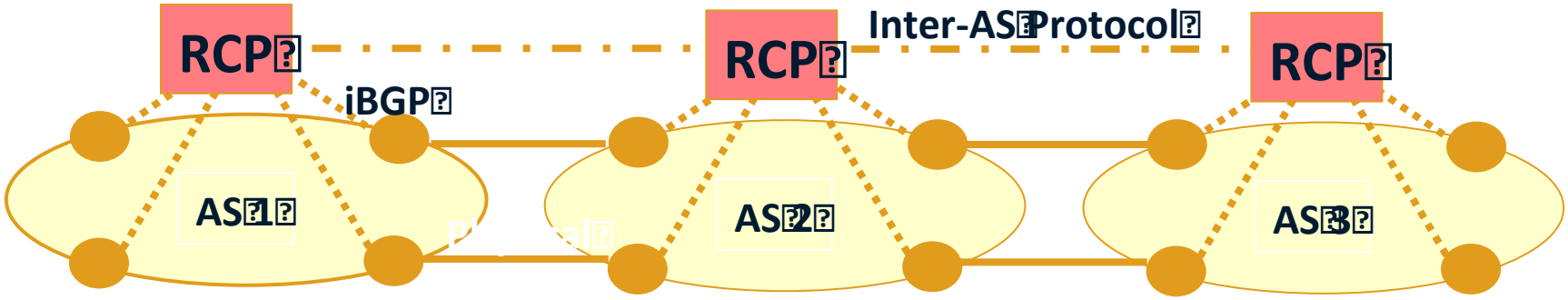
- SDN
  - motivazione, storia
  - visione ad alto livello
- Openflow
  - una (specifica) astrazione di switch
  - Il protocollo Openflow
- NOX/POX: uno (specifico) sistema operativo di rete
  - visione ad alto livello
  - sviluppi futuri

Piano di controllo di Internet

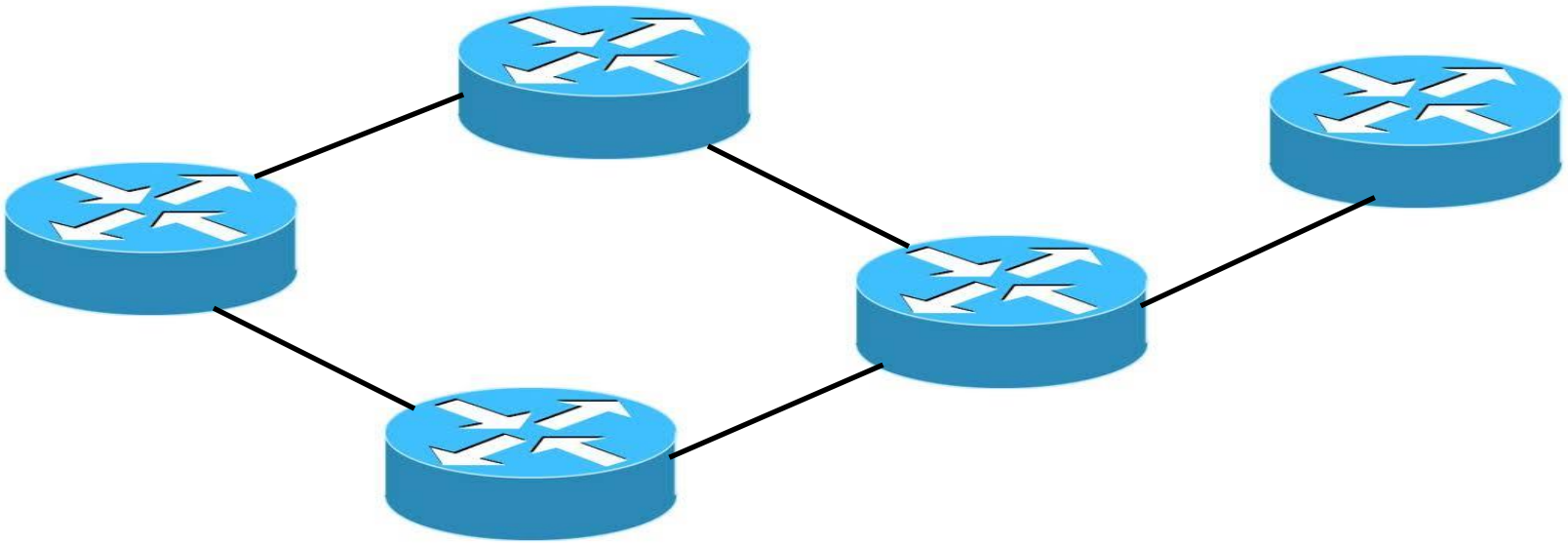


Preistoria di SDN

- RCP: router control platform, 2004
  - calcolo logicamente centralizzato dell'instradamento, in sostituzione del calcolo distribuito nei routers
  - usa protocolli esistenti (BGP) per comunicare ai router le informazioni di instradamento (routes)
  - IBM System Network Architecture (SNA): routing control server: 1970's

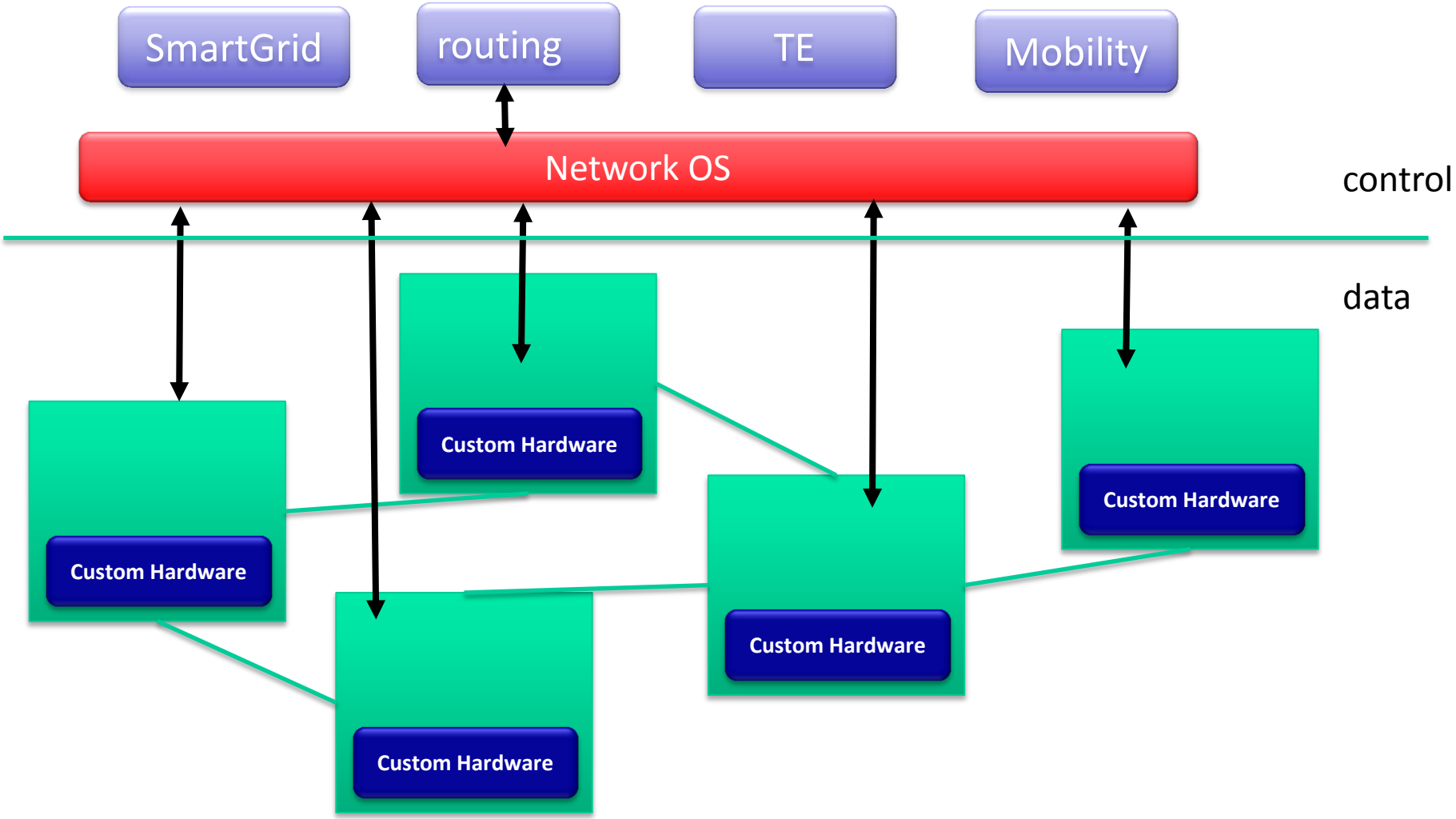


SDN: piano di controllo di rete aperto



Ack.: N. McKeown, G. Parulkar

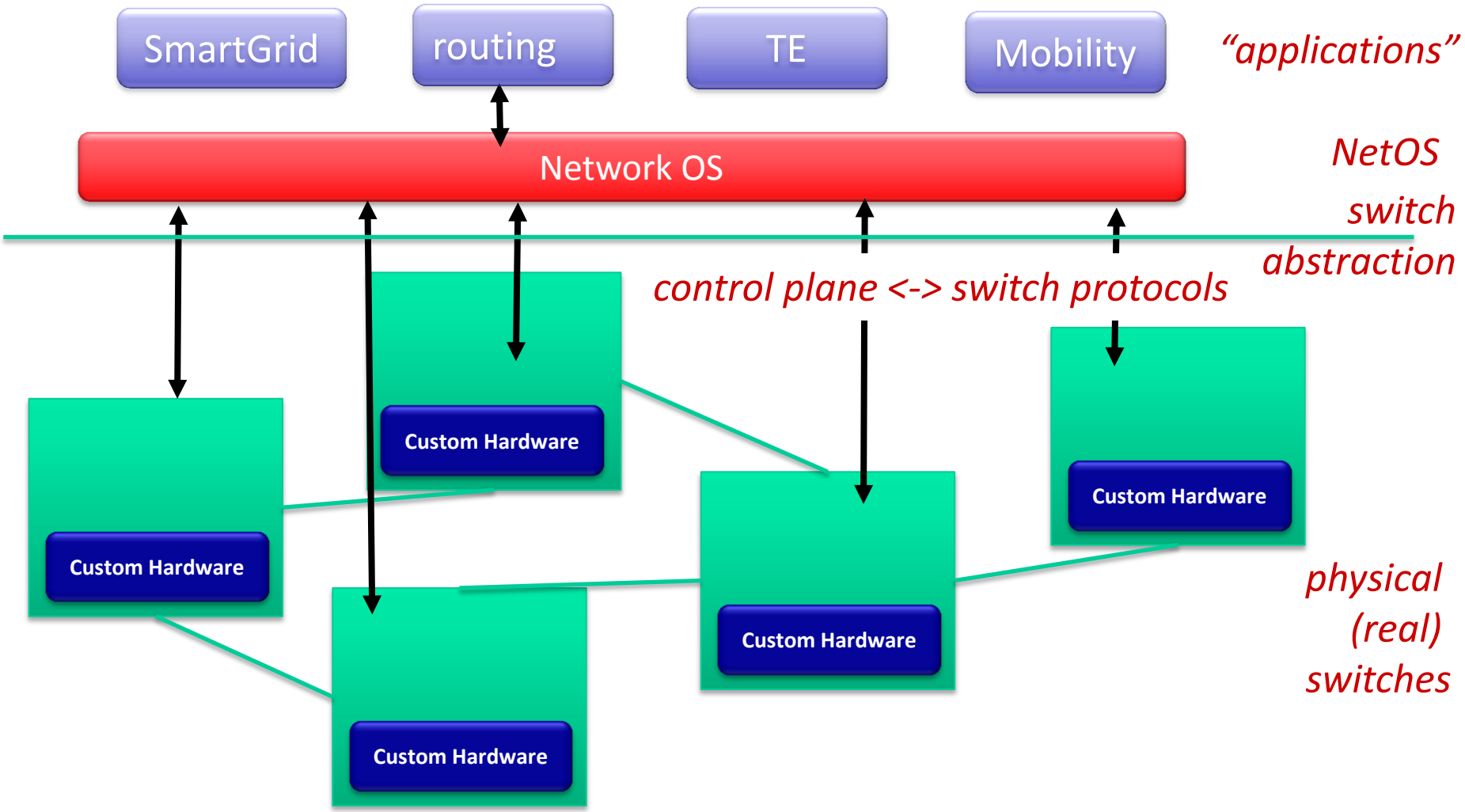
SDN: separation of control, data



Piano di controllo di Internet: SDN

- ripensamento della "divisione dei compiti" nella rete
- chiara e netta separazione tra piano dati e piano di controllo
  - piano di controllo *logicamente centralizzato*
  - sfruttamento di tecniche note per la gestione dello stato in sistemi distribuiti e cloud-based (e.g., sistemi transazionali, basi di dati tolleranti ai guasti, etc)
- interfaccia aperta verso il piano dati
  - permette programmabilità nel piano di controllo, personalizzazioni, innovazione
  - enfasi su innovazione per amministratori di rete (piuttosto che per utenti, ricercatori)
- controllo generale di "middleboxes" (non solo routing):
  - NAT, firewalls

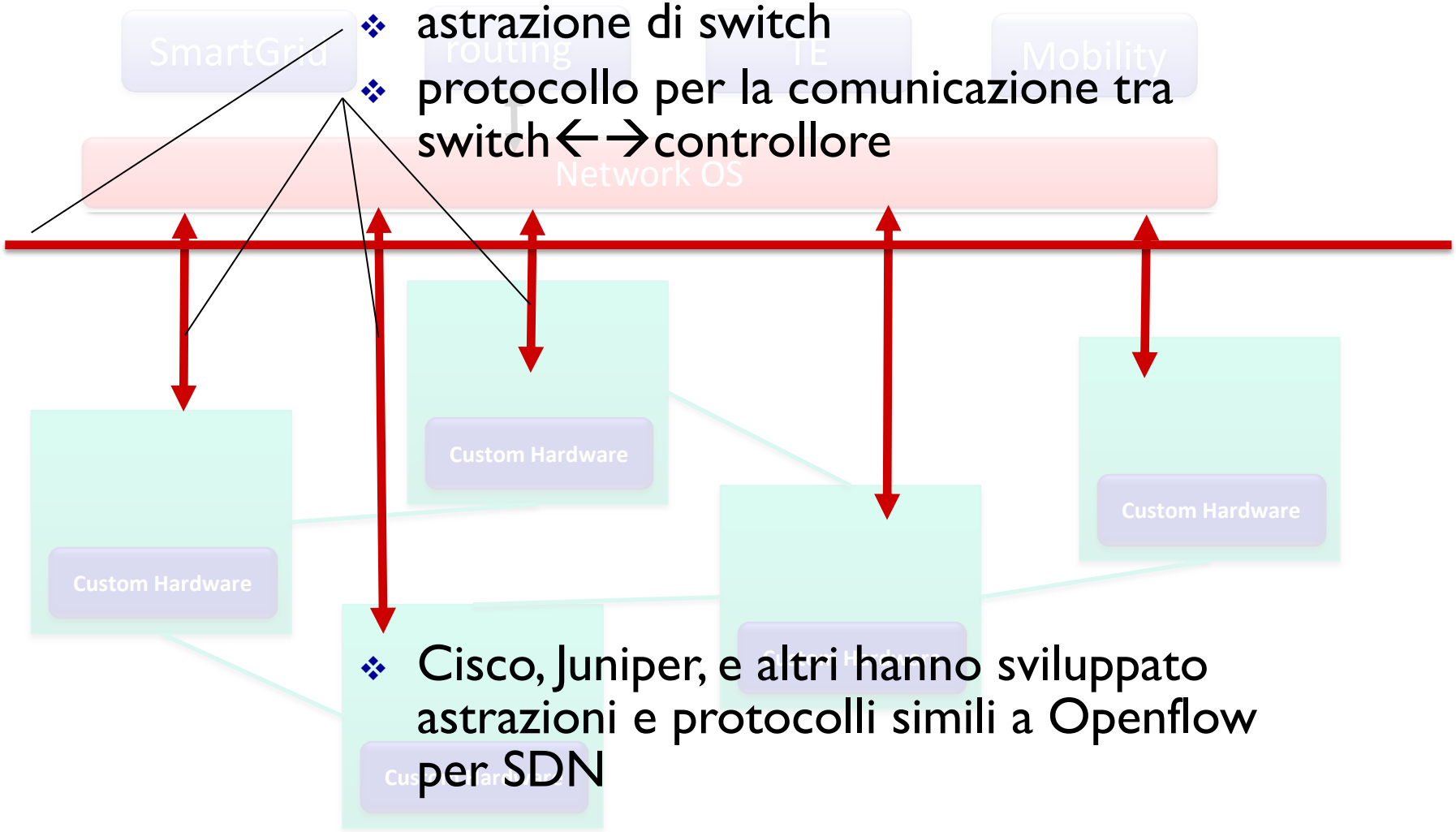
Componenti di SDN



SDN: Openflow

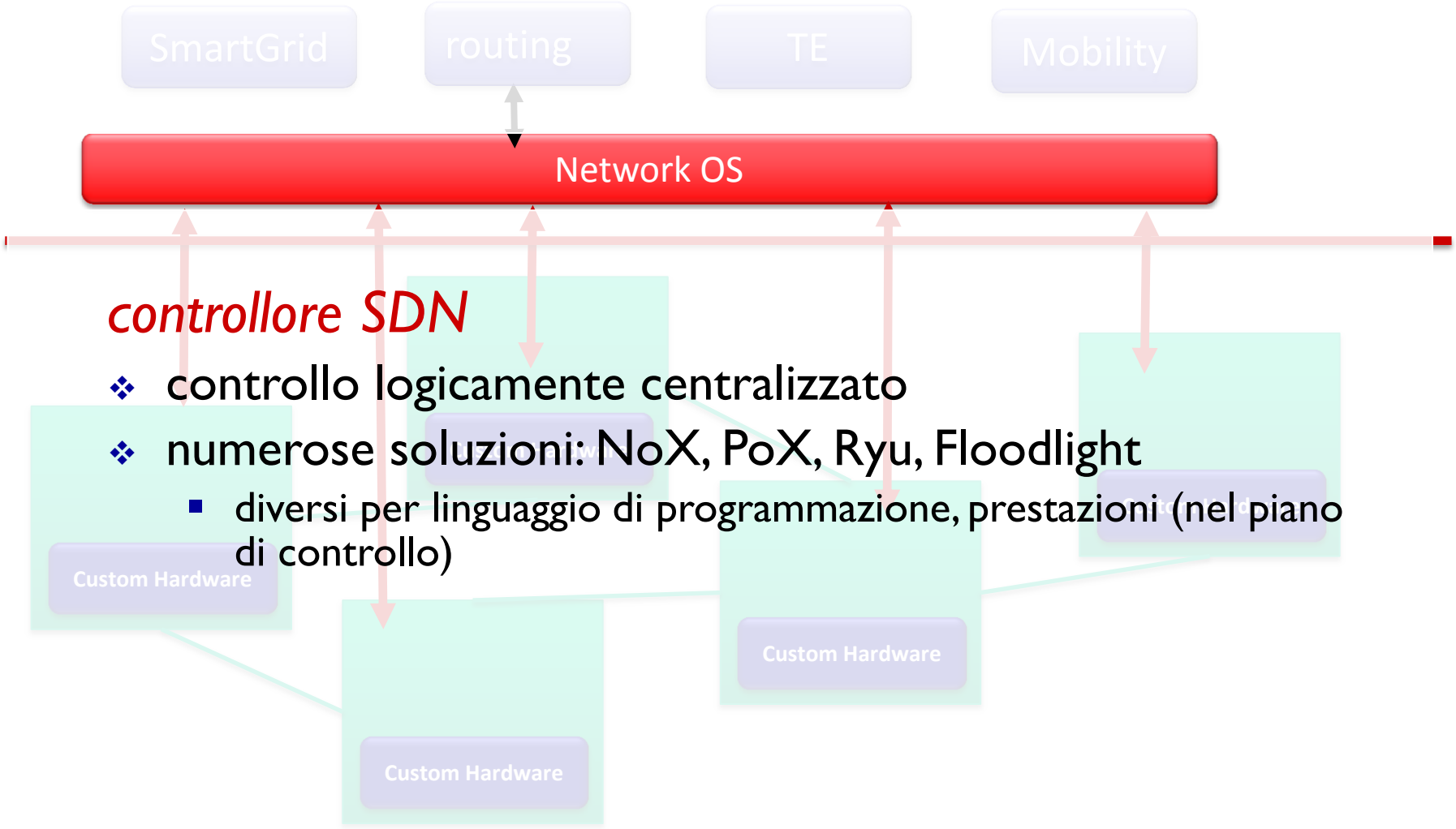
Openflow

- astrazione di switch
- protocollo per la comunicazione tra switch←→controllore



- Cisco, Juniper, e altri hanno sviluppato astrazioni e protocolli simili a Openflow per SDN

SDN: separazione piano dati/controllo

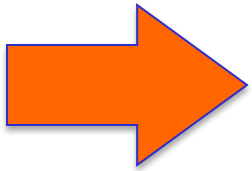


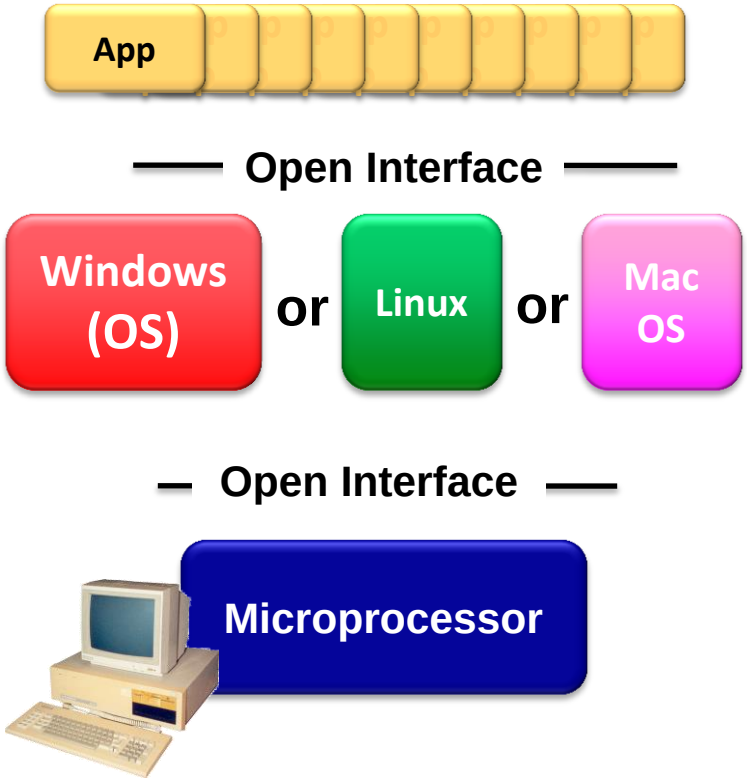
controllore SDN

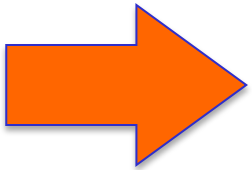
- controllo logicamente centralizzato
- numerose soluzioni: NoX, PoX, Ryu, Floodlight
  - diversi per linguaggio di programmazione, prestazioni (nel piano di controllo)

Analogia: evoluzione da mainframe a PC

Integrazione verticale
Chiuso, proprietario
Innovazione lenta
Pochi attori



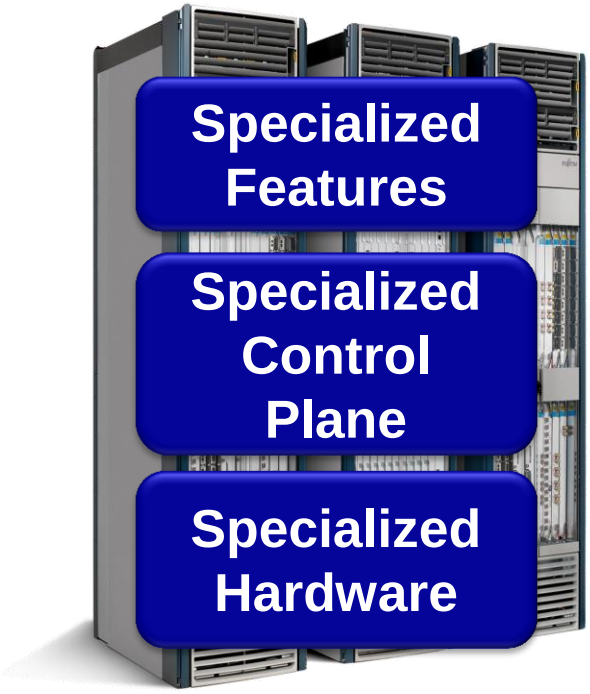


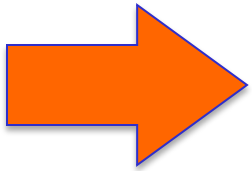


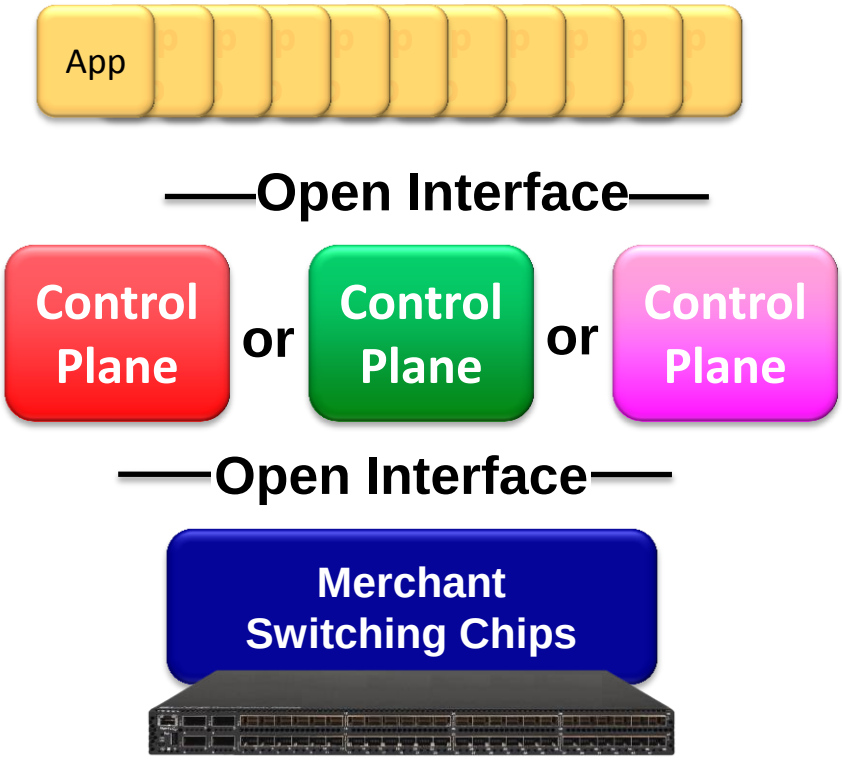
Separazione verticale
Interfacce aperte
Permette rapida innovazione
Molti attori

(ACK: N. McKeown)

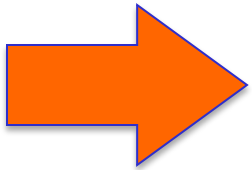
Routers/Switches







Integrazione verticale
Chiuso, proprietario
Innovazione lenta



Interfacce aperte
Innovazione rapida

Software Defined Networking

- SDN
  - motivazione, storia
  - visione ad alto livello
- Openflow
  - una (specifica) astrazione di switch
  - Il protocollo Openflow

- NOX/POX: uno (specifico) sistema operativo di rete
  - visione ad alto livello
  - sviluppi futuri

OpenFlow: Panoramica ad alto livello

- SDN != OpenFlow
  - OpenFlow è solo una di tante possibili implementazioni di SDN
  - Cisco Open Network Environment (ONE)
  - Juniper Contrail
  - Huwei SOFTcom
- astrazione fondamentale di OpenFlow, granularità del controllo: *flusso*
  - *azioni prese sulla base di ~10 combinazioni di valori di headers*

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport

+ mask

  - insieme comune di funzioni per: switch, router, NAT, firewall
  - nota: no *per-packet processing*, *deep packet inspection*

Uno switch OpenFlow consiste di:

Tre componenti:

- *tabella dei flussi (flow table)* con specificata l'azione che lo switch deve compiere sul flusso
  - esempi: inoltro verso la porta x, scarto
  - varie regole permettono allo switch di operare come un Ethernet layer 2 switch, layer 3 IP router, firewall, NAT box, load balancer
- *un canale di comunicazione sicuro* verso il controllore (per l'installazione di entries nella flow table entries, segnalazione di eventi, etc.)
- *Il protocollo OpenFlow*: per lo scambio di messaggi tra il processo del controllore e lo switch

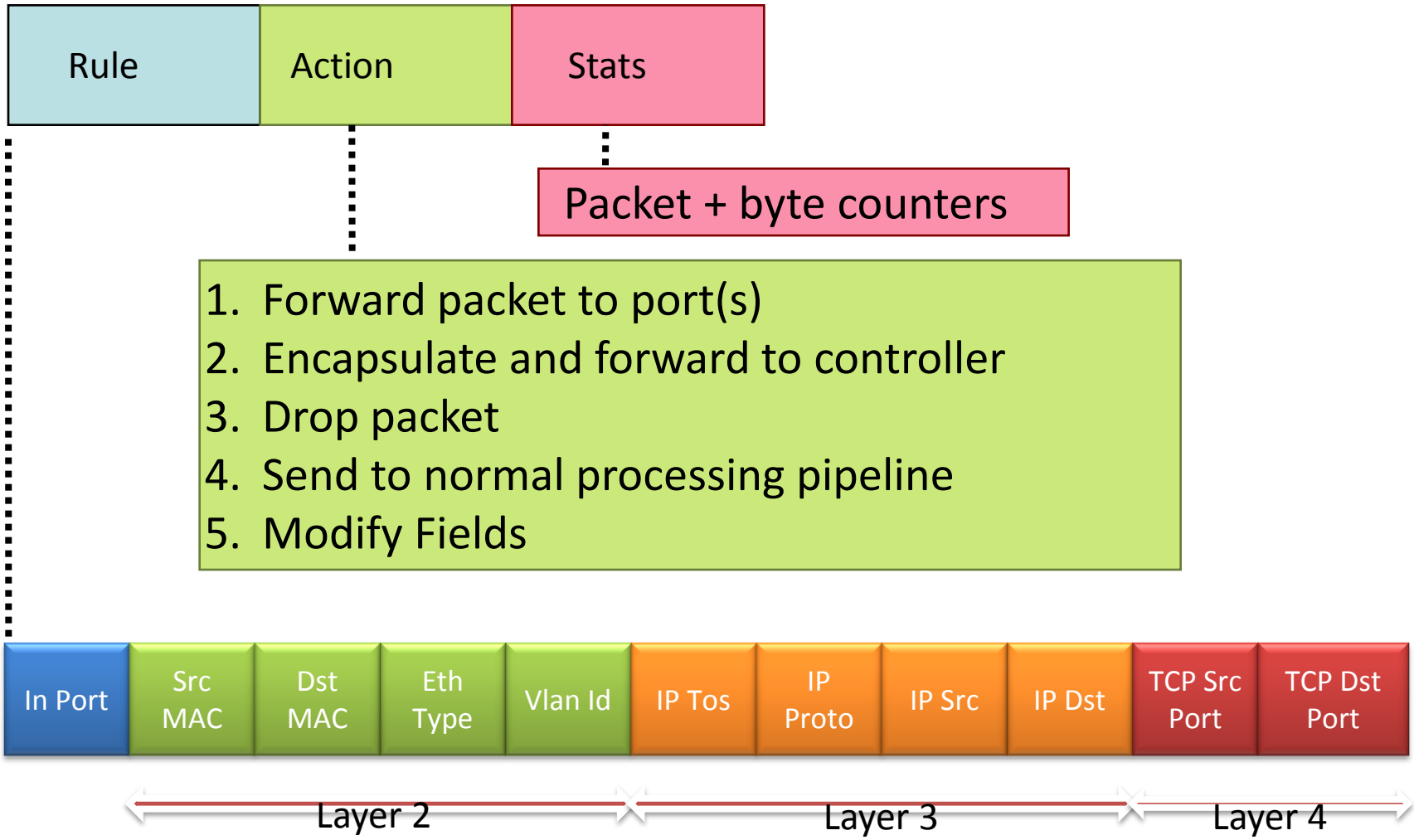
OpenFlow: astrazione del piano dati

- *flusso*: definito da insieme di headers
- semplici regole di gestione dei pacchetti
  - *Pattern:* bit-matching sul valore degli header
  - *Azioni:* scarto, inoltro, modifica, invio al controllore
  - *Priorità*: per disambiguare pattern sovrapposti
  - *Contatori:* #bytes e #pacchetti

* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow: Entries della Flow Table



Esempi

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Switching alivello di flusso (notare la granularità più fine rispetto a esempio precedente)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

Exampi

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

OpenFlow: potenza della astrazione

- match+azione: unifica diversi tipi di middleboxes

- Router
  - *match:* longest prefix matching sull'indirizzo IP di destinazione
  - *azione:* invio su un link di uscita
- Switch
  - *match:* indirizzo MAC di destinazione
  - *azione:* inolltro su porta specifica o flooding
- Firewall
  - *match*: indirizzi IP e porte TCP/UDP
  - *azione:* permesso o blocco
- NAT
  - *match:* indirizzo IP e numero di porta
  - *azione:* riscrittura di indirizzo/porta

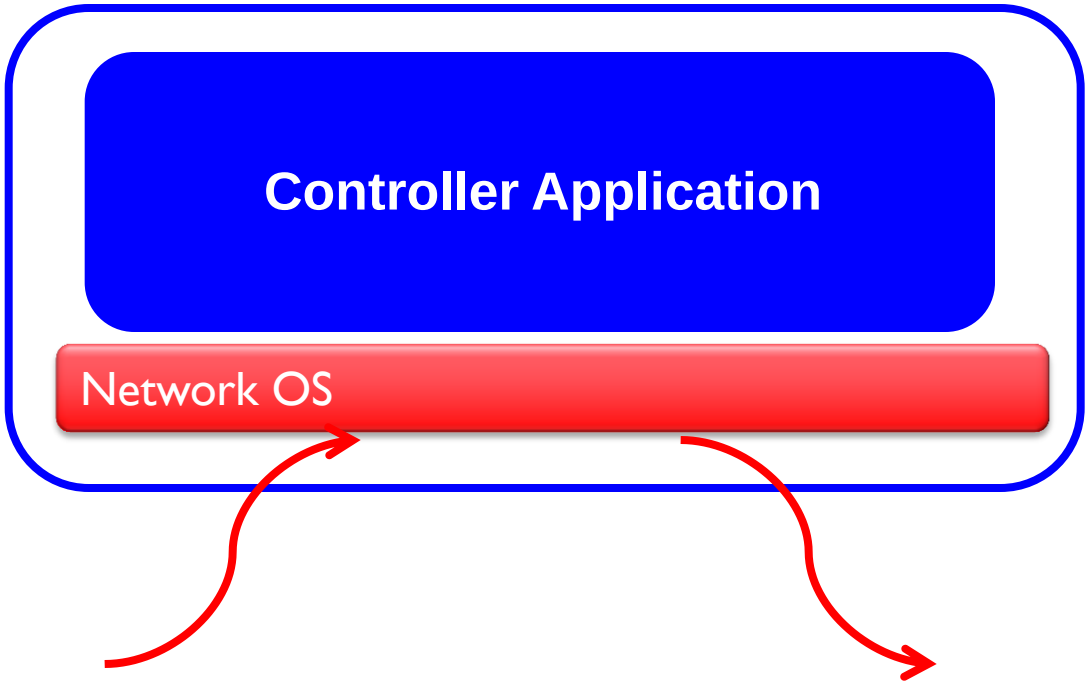
Azioni OpenFlow: Inoltro/Scarto

- *Inoltro:*
  - ALL: Invio su tutte le interfacce di uscita, esclusa quella di arrivo
  - CONTROLLER: Reincapsula e invia al controllore
  - LOCAL: Invio al networking stack locale dello switch
  - TABLE: Esegue azione specificata nella flow table (solo per pacchetti in uscita dallo switch)
  - IN PORT: Invio sulla porta di arrivo
- *Scarto:*
  - se l'azione non viene specificata, di default tutti i pacchetti che fanno match vengono scartati

Azioni OpenFlow: Modifica

- *Modifica*: viene modificato il valore di alcuni campi degli header, ad esempio:
  - VLAN ID, priorità, etc.
  - indirizzo IP di destinazione
  - numero di porta

Controllore: Programmabilità



Events from switches

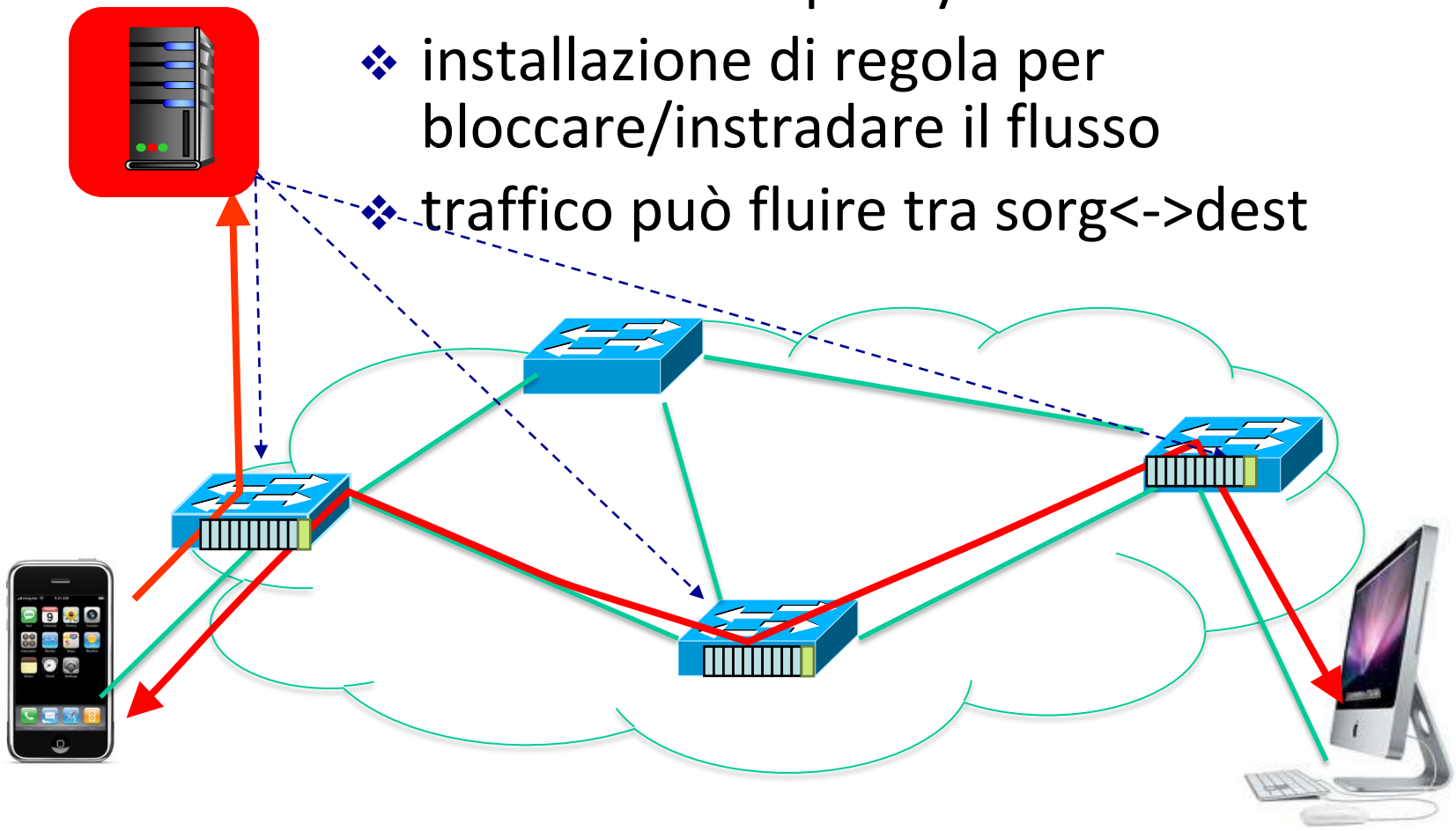
- **Topology changes,**
- **Traffic statistics,**
- **Arriving packets**

Commands to switches

- **(Un)install rules,**
- **Query statistics,**
- **Send packets**

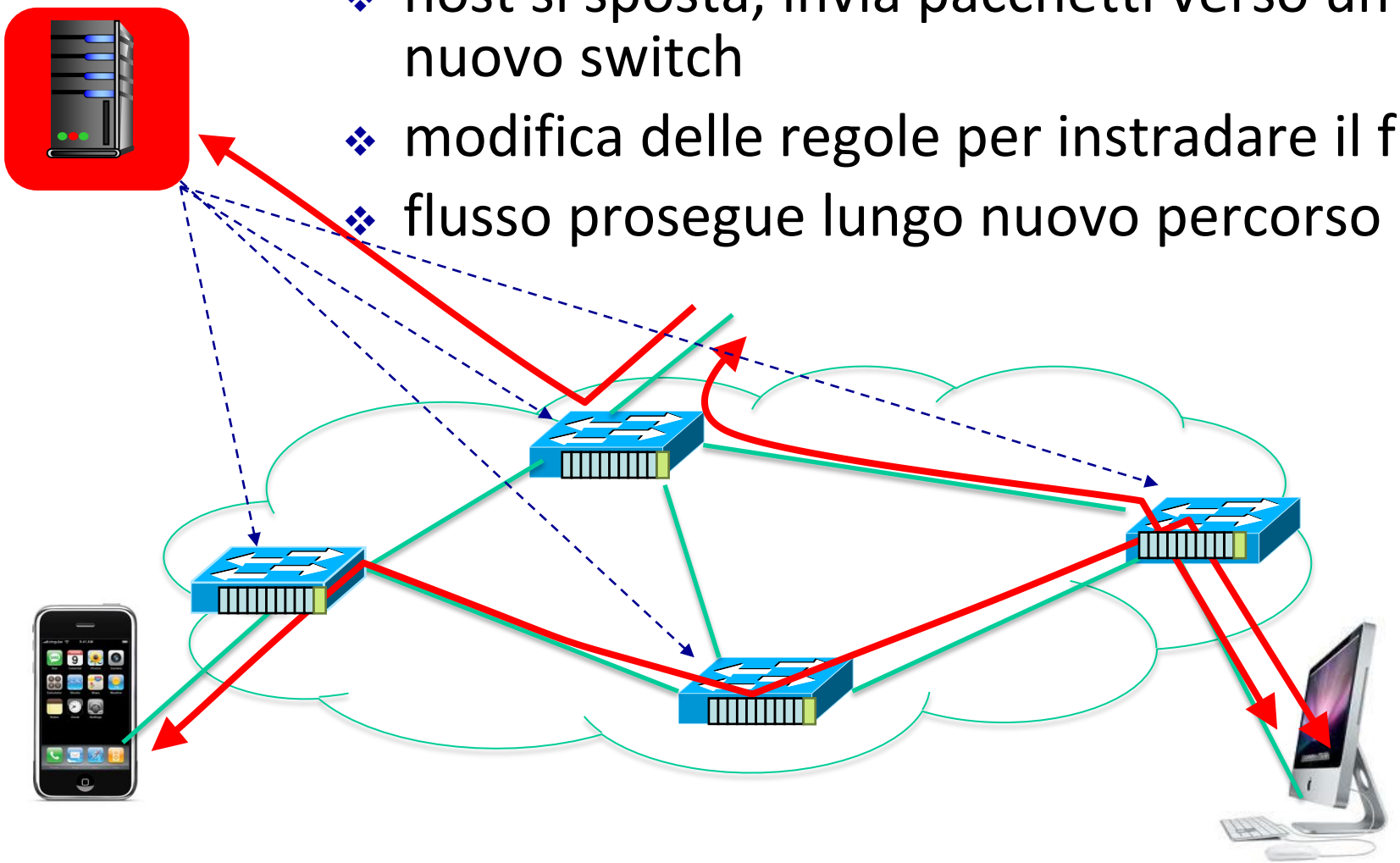
Esempio: Controllo di accesso dinamico

- inspezione del primo pacchetto del flusso
- consultazione policy nel controllore
- installazione di regola per bloccare/instradare il flusso
- traffico può fluire tra sorg<->dest



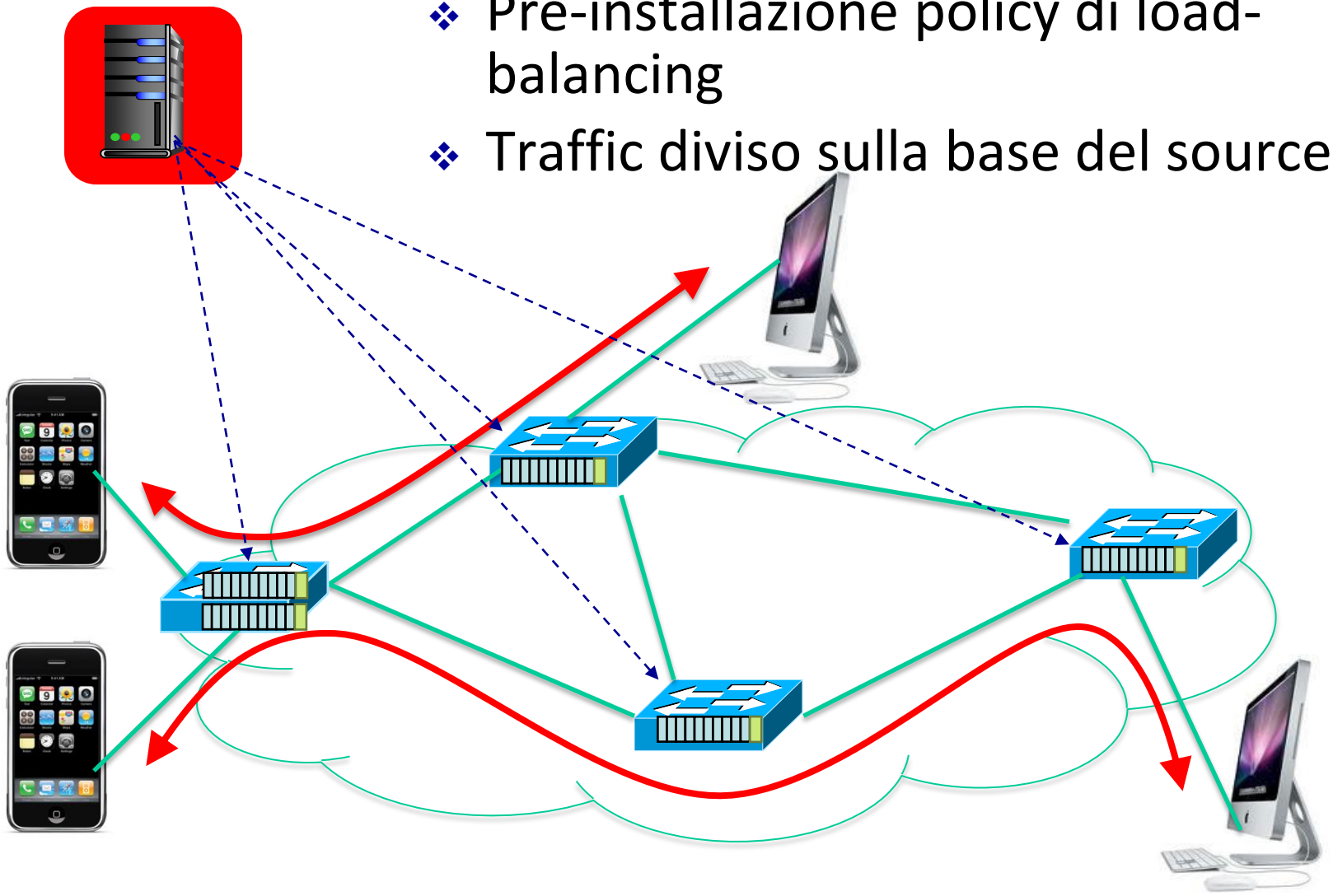
Esempio: mobilità trasparente

- flusso in corso
- host si sposta, invia pacchetti verso un nuovo switch
- modifica delle regole per instradare il flusso
- flusso prosegue lungo nuovo percorso



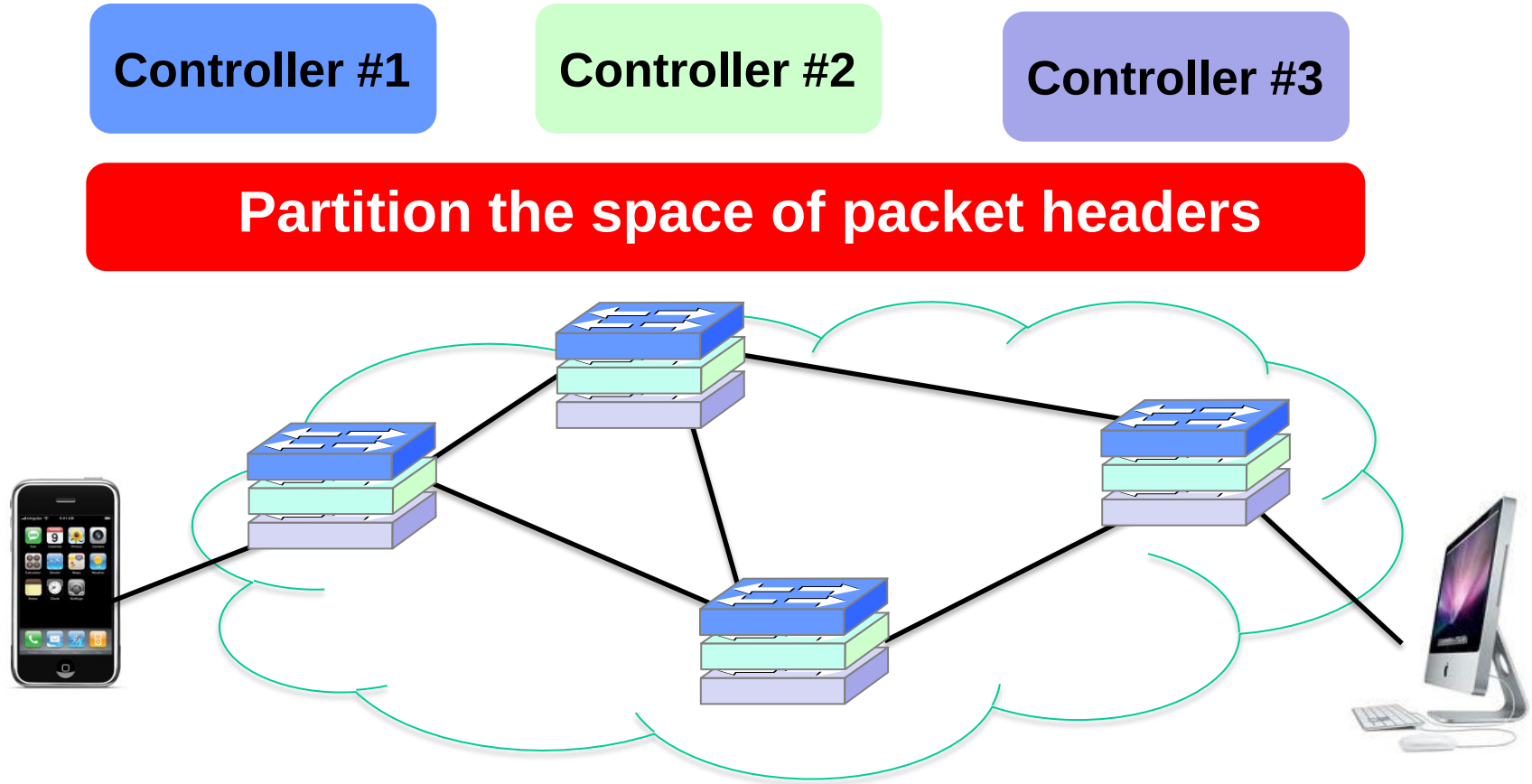
Esempio: Bilanciamento del carico

- Pre-installazione policy di load-balancing
- Traffic diviso sulla base del source IP

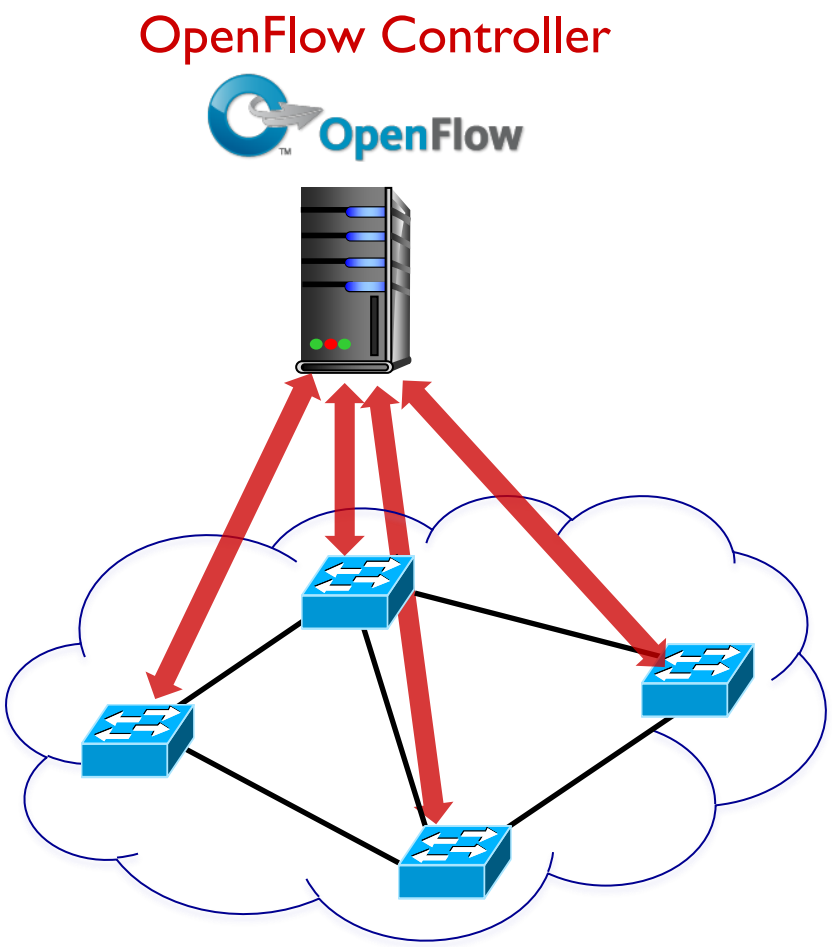


Esempio: Virtualizzazione della rete

- si divide lo spazio degli indirizzi in tre partizioni
- ogni partizione gestita da un diverso controllore



Il protocollo OpenFlow

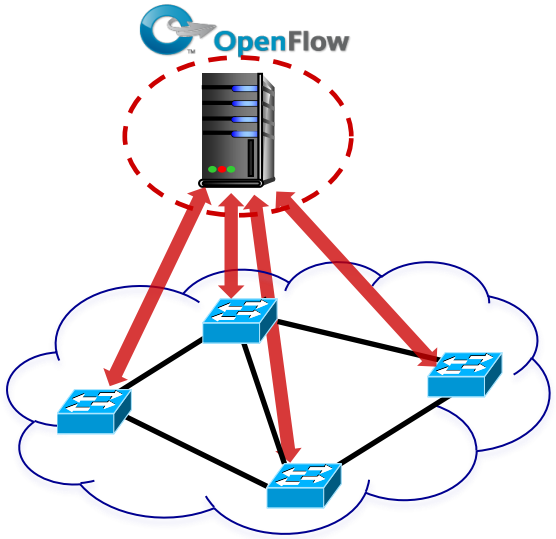


- opera tra il controllore e gli switch
- TCP usato per scambio dei messaggi
  - opzionalmente cifrati
- tre classi di messaggi OpenFlow:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc)

OpenFlow: messaggi controller-to-switch

Funzioni di base

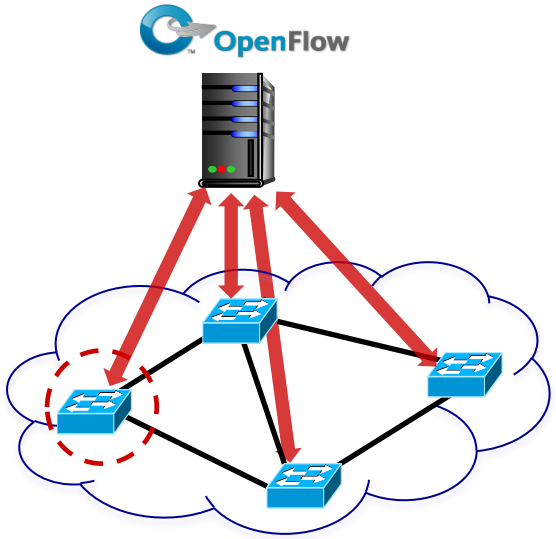
- *features:* controllore interroga lo switch in merito a una certa features, lo switch risponde
- *configure:* controllore interroga/setta un parametro di configurazione
- *modify-state:* aggiunta, cancellazione, modifica di entries nella Flow Table
- *packet-out:* controllore ordina di inviare pacchetto fuori da una specifica porta di uscita dello switch



OpenFlow: messaggi switch-to-controller

Funzioni di base

- *packet-in:* trasferisce il pacchetto (e il suo controllo) al controllore. (cfr. messaggio packet-out precedente)
- *flow-removed:* rimozione di una entry dalla flow table dello switch
- *port status:* informa il controllore del cambiamento di stato di una interfaccia.



Fortunatamente, gli operatori di rete non devono occuparsi direttamente della construzione/invio dei messaggi. Si usano invece linguaggi di più alto livello messi a disposizione dal sistema operativo di rete (es. POX, Floodlight, NOX)

Software Defined Networking

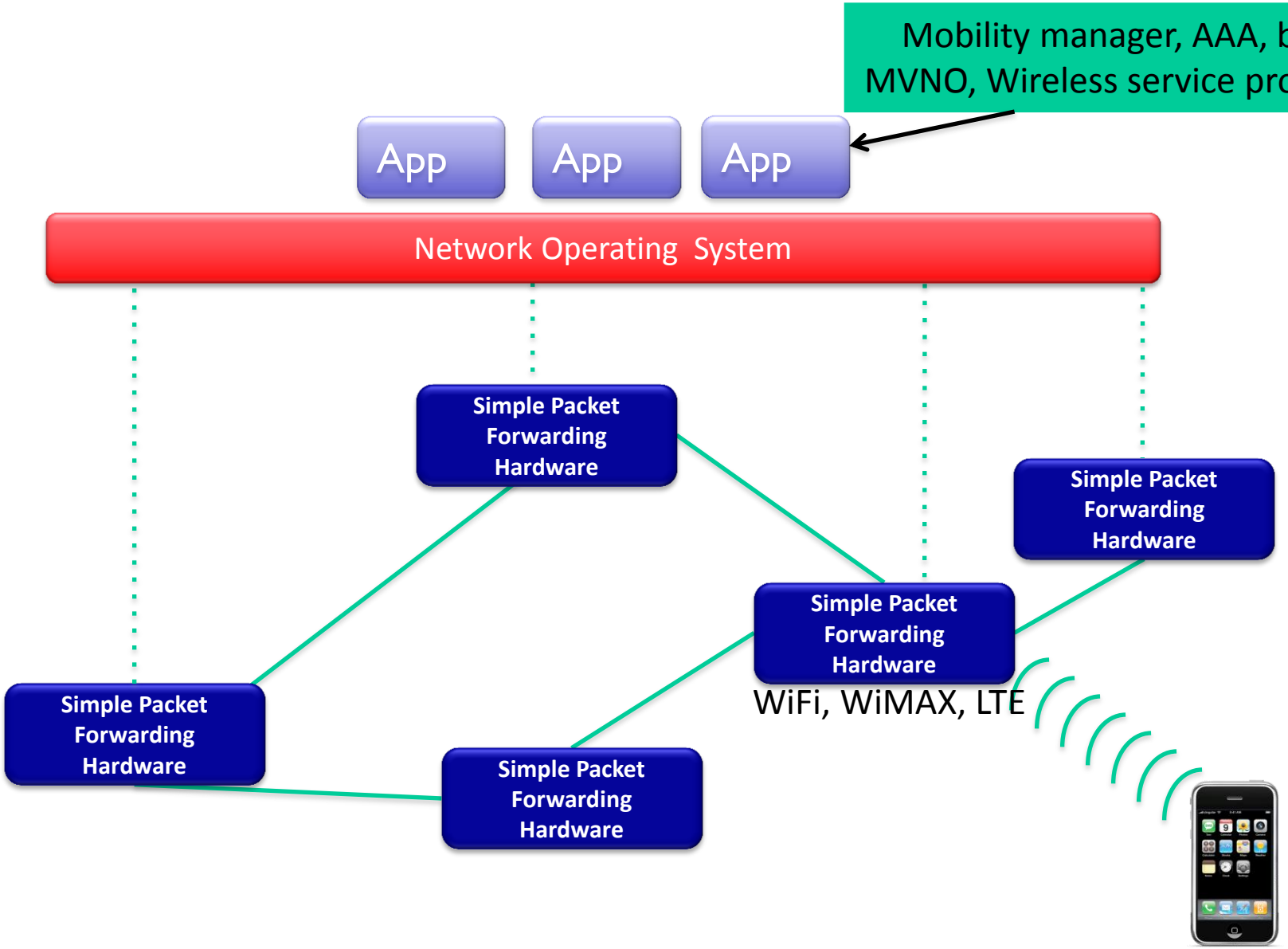
- SDN
  - motivazione, storia
  - visione ad alto livello
- Openflow
  - una (specifica) astrazione di switch
  - Il protocollo Openflow
- NOX/POX: uno (specifico) sistema operativo di rete
  - visione ad alto livello
  - sviluppi futuri

Il controllore SDN

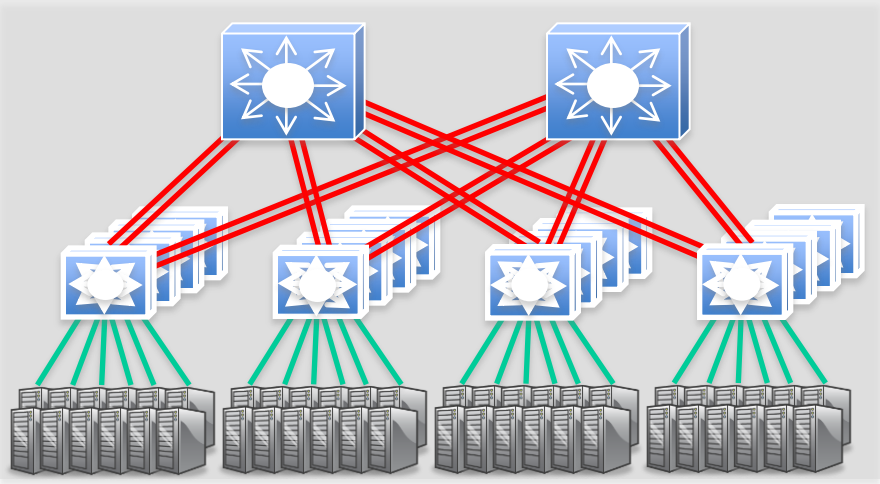
- è il "cervello" della rete
  - setta le entries della flow table negli switch
  - risponde a eventi inviati dagli switch
  - comunica con gli switch implementando (ad esempio) il protocollo OpenFlow
- Sono stati proposti vari "ambienti" per realizzare controllori che fanno uso di OpenFlow:
  - NoX, PoX, Beacon, Floodlight, Maestro, Ryu, Frenetic
- *paradigma di programmazione event-based:* controllore programmato per rispondere a eventi, ad esempio:
  - uno switch si aggiunge alla rete (procedura di inizializzazione)
  - cambiamento dello stato di un link riportato dalla rete (ricalcolo delle routes, modifica delle flow tables)

Software-defined Wireless Networks

SDN si applica anche alle reti wireless



Applicazione a Data Center



Cost

200,000 servers

Fanout of 20 ⇨ 10,000 switches

$5k commercial switch ⇨ $50M

$1k custom-built switch ⇨ $10M

Savings = $40M

Control

1. Optimize for features needed
2. Customize for services & apps
3. Quickly improve and innovate

Large data center operators are moving towards defining their own network in software.

Conseguenze per gli standard

Gli standard definiranno (solo) le interfacce

Il ruolo degli standard cambierà:

- Gli amministratori della rete definiranno il comportamento della rete
- Nuove features saranno adottate senza standard

Paradigma di programmazione

- Il buon software è adottato, non standardizzato

Sviluppi del networking con SDN

Le reti diventeranno

- Più programmatiche
- Definite da amministratori e operatori, non aziende costruttrici
- Più rapide a cambiare, per andare incontro ai bisogni degli operatori
- I costi della insfrattura, i costi operativi, il consumo di energia diminuiranno

Le astrazioni

- Salvano i programmatori dalla complessità
- Rendono i sistemi più prevedibili
- Ci porteranno in luoghi che ancora non immaginiamo