



Felhő alapú hálózatok (VITMMA02)

SDN a felhőben

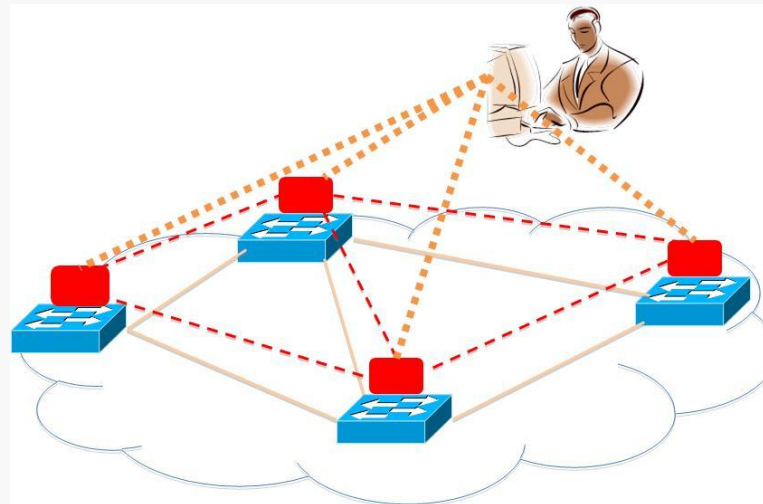
Dr. Maliosz Markosz

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

2019. tavasz

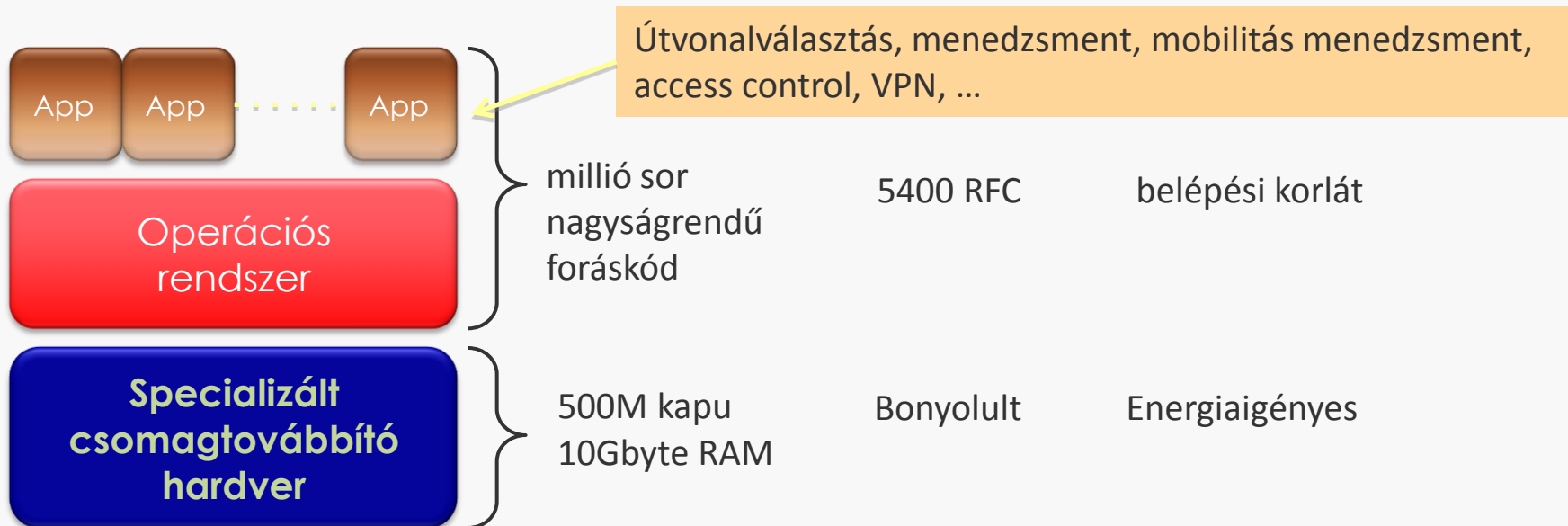
Hagyományos hálózatok

- » **Adat sík (Data plane):** linksebesség időskálán működik (gyors)
 - » csomagok kezelése: továbbítás, szűrés, pufferezés, jelölés, ütemezés, számlálók
- » **Vezérlő sík (Control plane):** lassabb időskála (vezérlő üzenetek kezelése)
 - » elosztott algoritmusok
 - » topológia változások követése, útvonalak számítása, továbbítási szabályok beállítása
- » **Menedzsment sík (Management plane):** emberi időskála
 - » központosított
 - » mérések összegyűjtése és eszközök konfigurációja

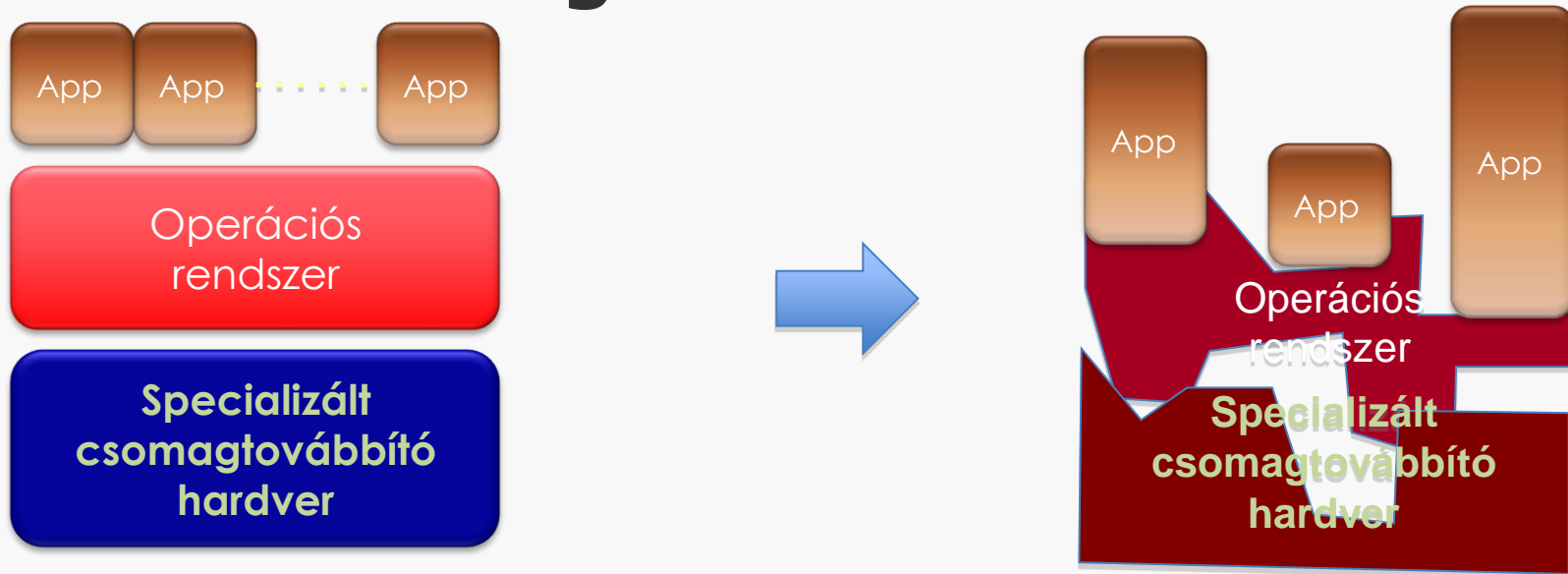


Hálózat és hálózati eszköz architektúra

- » Régen egyszerű volt: Ethernet, IP, TCP...
- » Az új **vezérlési** igények nagyon bonyolulttá tették
 - » Izoláció ⇨ VLAN, ACL
 - » Traffic engineering ⇨ MPLS, ECMP, súlyok
 - » Csomagfeldolgozás ⇨ Tűzfalak, NAT, middleboxes
 - » Csomagtartalom elemzés ⇨ Deep packet inspection (DPI)
 - » ...
- » Sok komplex funkció az infrastruktúra része lett
 - » OSPF, BGP, multicast, differentiated services, Traffic Engineering, NAT, firewalls, MPLS, ...
- » „mainframe” mentalitás – monolitikus architektúra

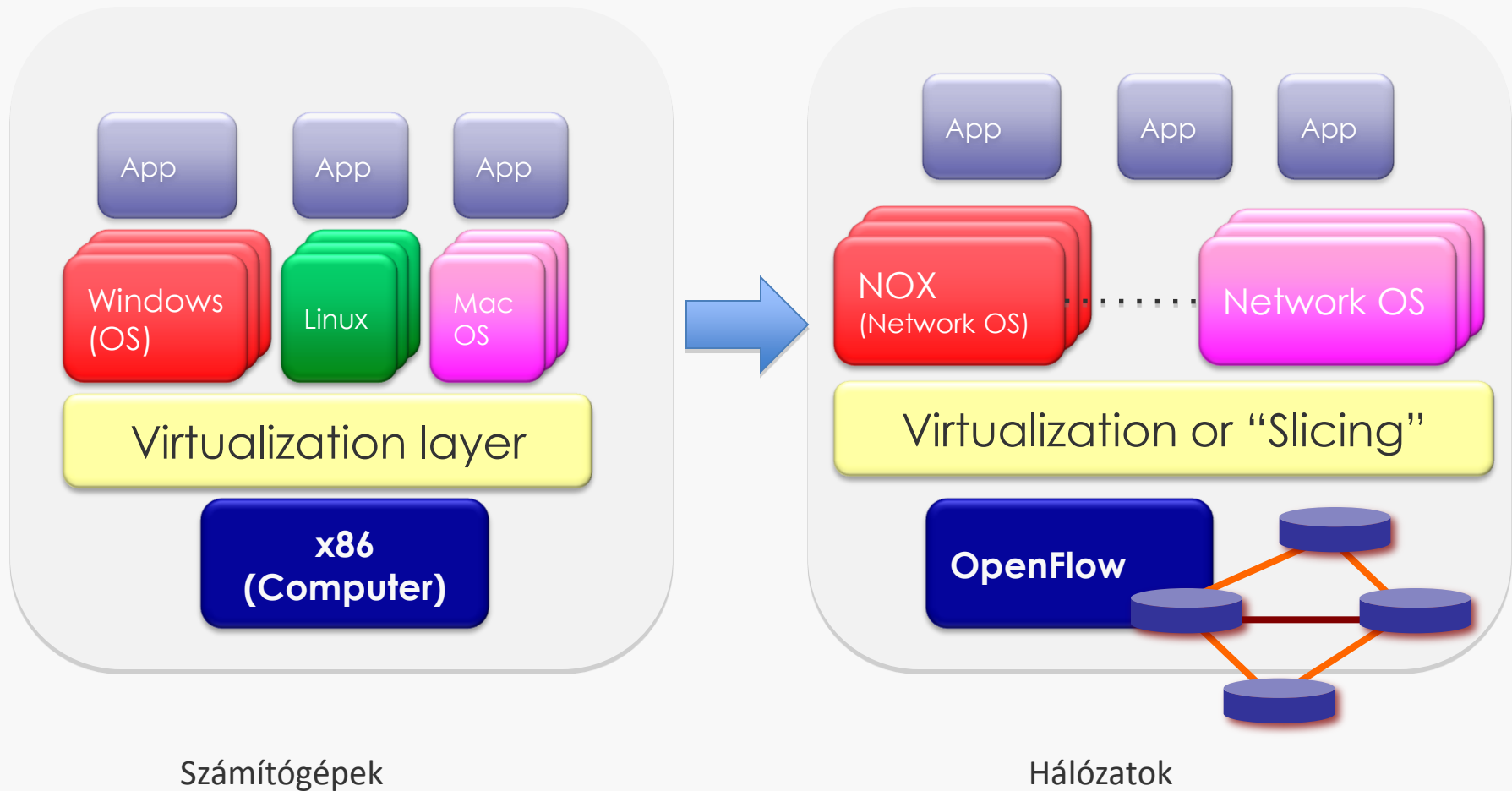


Ideális és megvalósult architektúra



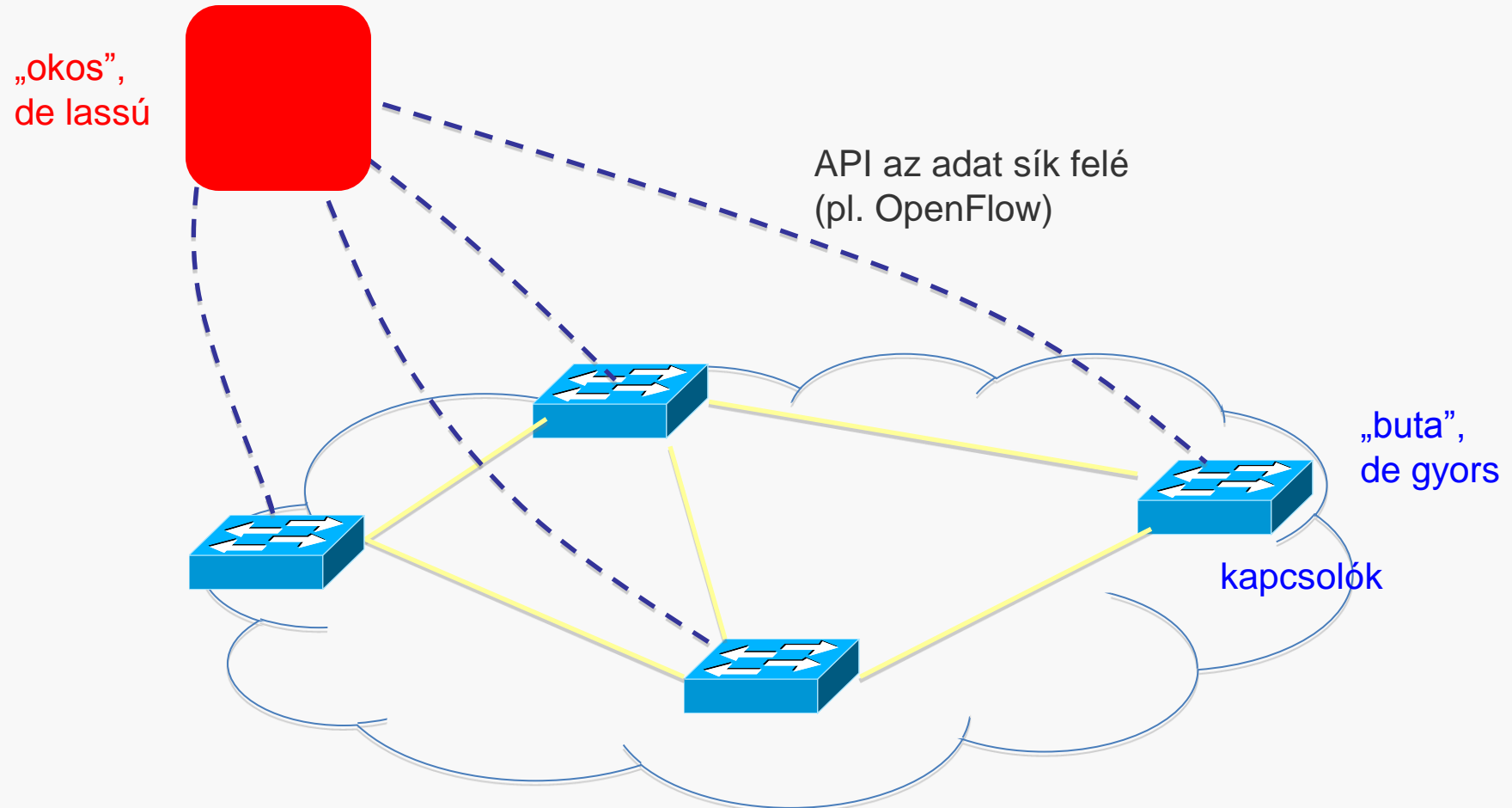
- » Miért fontos a rétegek elválasztása?
 - » szétválasztott szolgáltatások komponensekkel megvalósítva
 - » független, de kompatibilis innováció lehetséges az egyes rétegekben
- » Zárt architektúra
 - » elmosódott határok, zárt interfészek
 - » szoftver és hardver összekötve
 - » függőlegesen integrált, komplex, zárt, egyedi, gyártó specifikus interfészek

Hasonlóság a számítógép architektúrák



Szoftver Definiált Hálózatok (SDN)

Logikailag központosított vezérlés

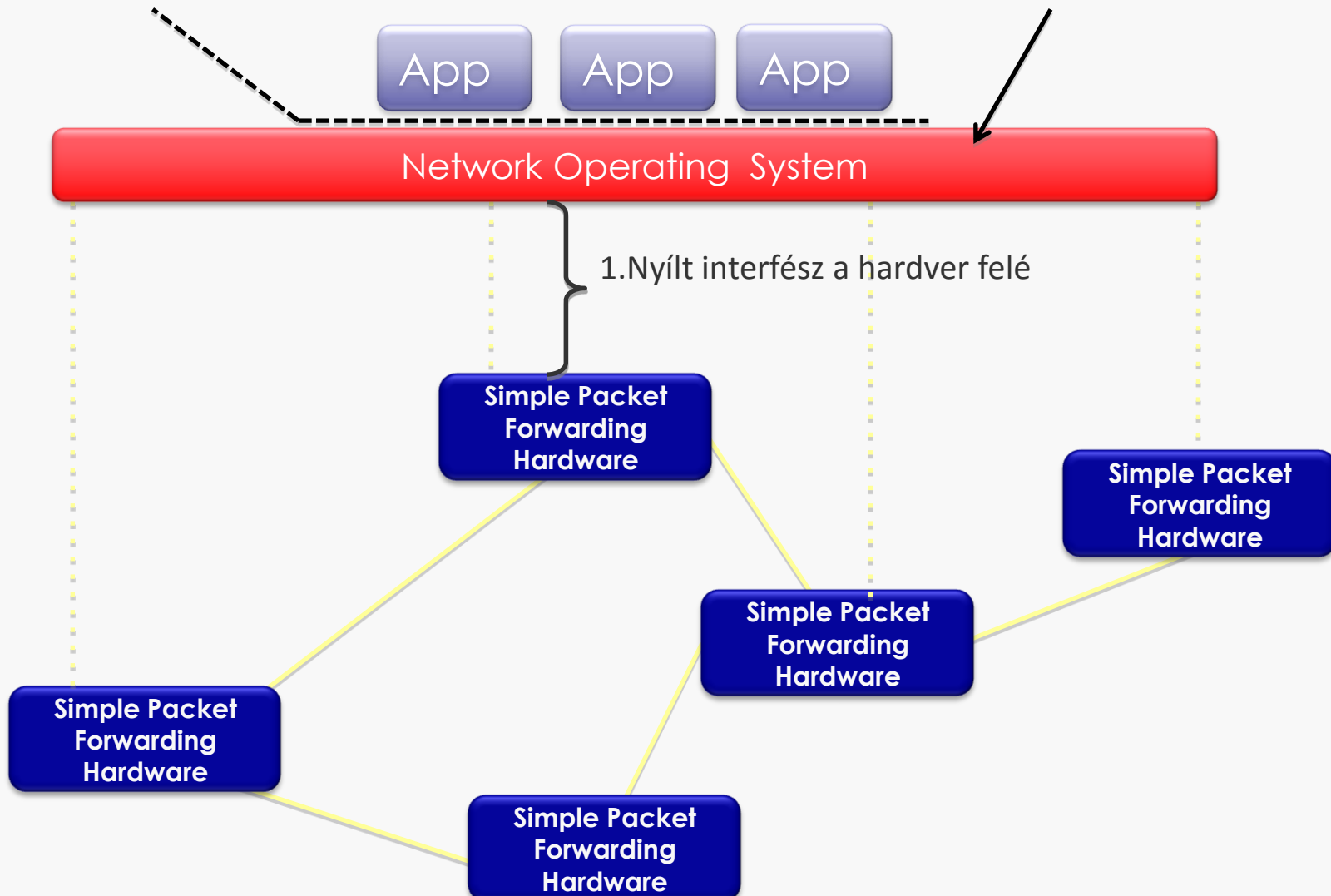




SDN komponensek

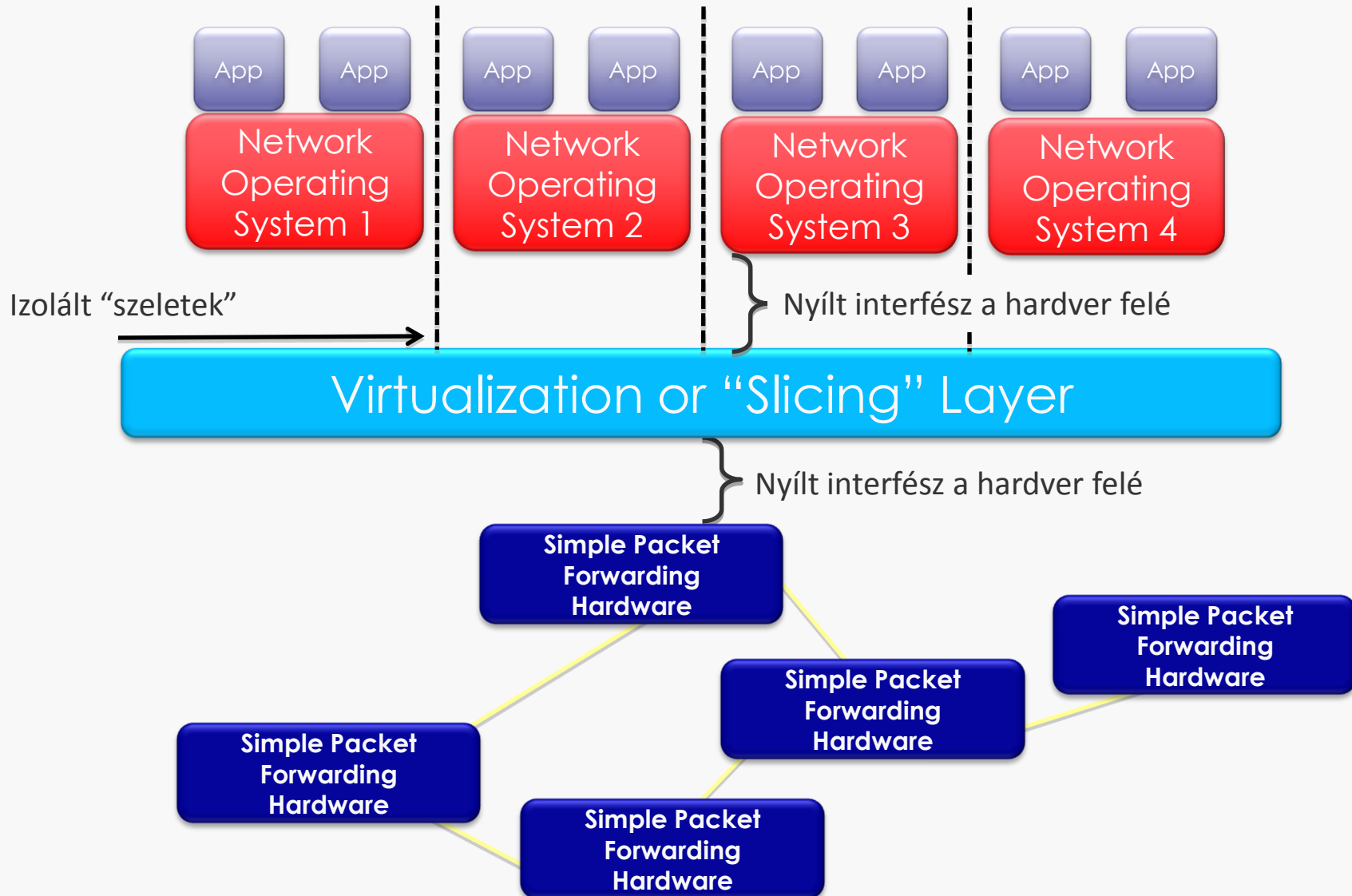
3. Jól definiált nyílt API

2. Operációs rendszer
Bővíthető, lehetőleg nyílt forráskódú



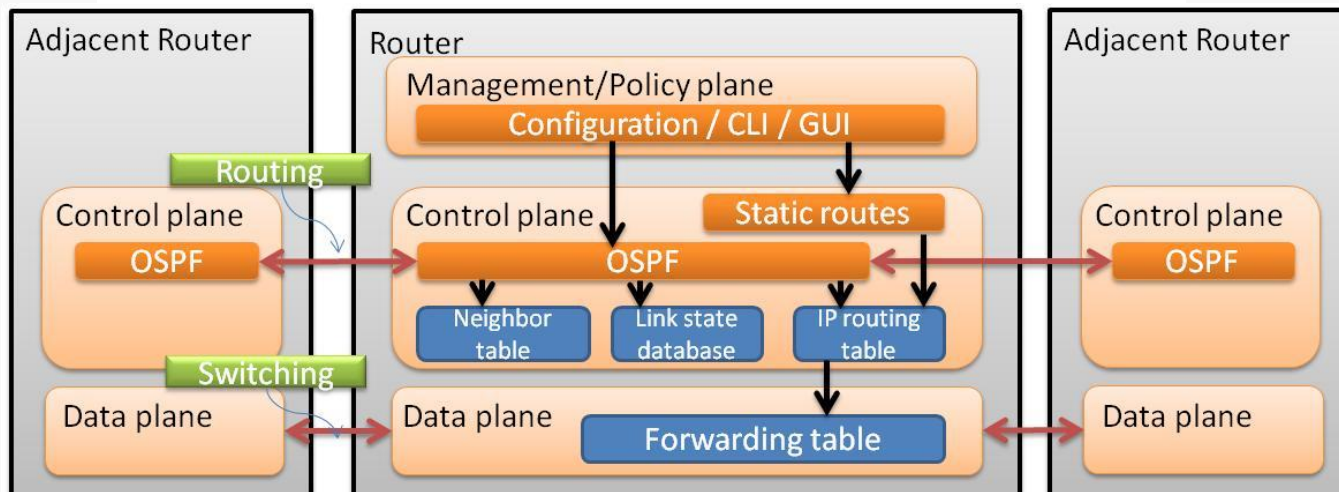
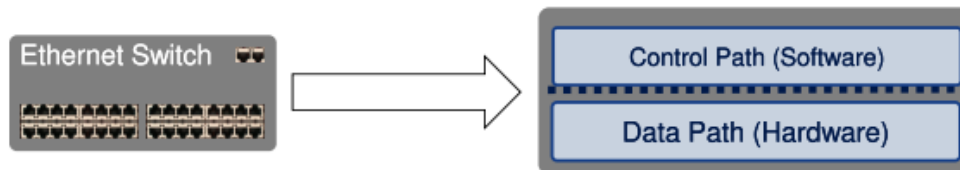


SDN virtualizáció



Hagyományos kapcsoló/útválasztó

- » Működés szétbontható síkokra
 - » Menedzsment sík / konfiguráció
 - » Vezérlő sík / döntések
 - » Adat sík / csomag továbbítás





SDN koncepció

- » A vezérlő- és adatsík elemek szétválasztása
 - » a hálózati intelligencia és állapot logikailag központosított
 - » a vezérelt hálózati infrastruktúra *absztrakt* formában jelenik meg az alkalmazások számára
- » A vezérlősík szoftver általános hardveren fut
 - » leválasztás a speciális hálózati hardverről
 - » általános szerverek alkalmazása
- » Az adatsík programozható
 - » Az adatsík felügyelete, vezérlése és programozás egy központi helyről
- » Nem csak a hálózati eszközök, hanem az egész hálózat vezérelhető



Vezérlő szoftver program

A vezérlő program az általa észlelt hálózati képen végez műveleteket

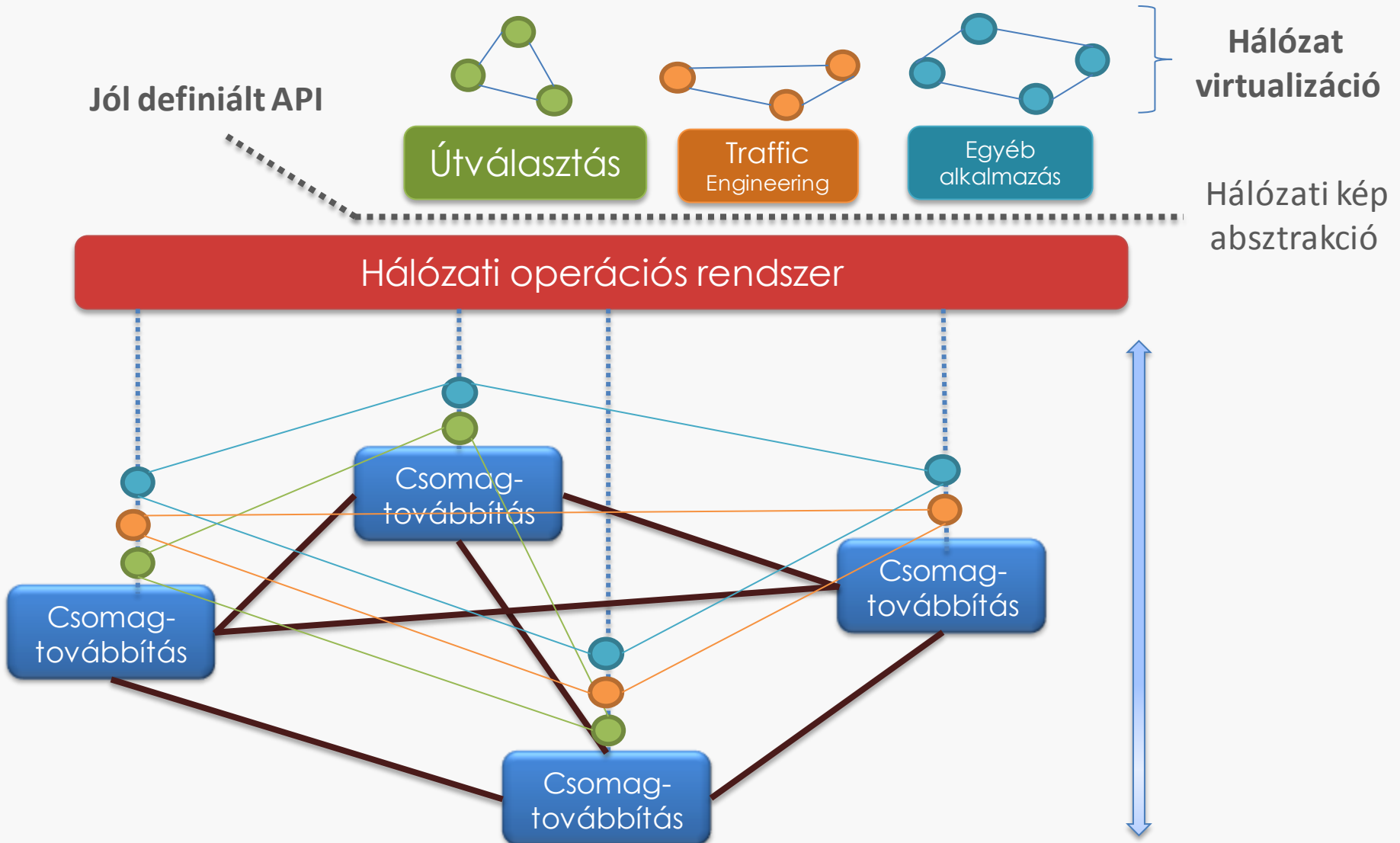
- » **Bemenet:** globális hálózati kép (gráf/adatbázis)
 - » API-n keresztül elérhető információkkal ellátott hálózati gráf
 - » a kapcsolóktól érkező „események”
 - » topológia változás
 - » forgalmi statisztikák
 - » érkező csomagok

- » **Kimenet:** minden egyes hálózati eszköz beállítása
 - » A vezérlő mechanizmus egy program, ami pl. egy gráf algoritmust valósít meg
 - » Üzenetek küld a kapcsolóknak
 - » szabályok beállítása, törlése
 - » statisztikák lekérdezése
 - » csomagok küldése

A vezérlő program **nem elosztott** rendszer

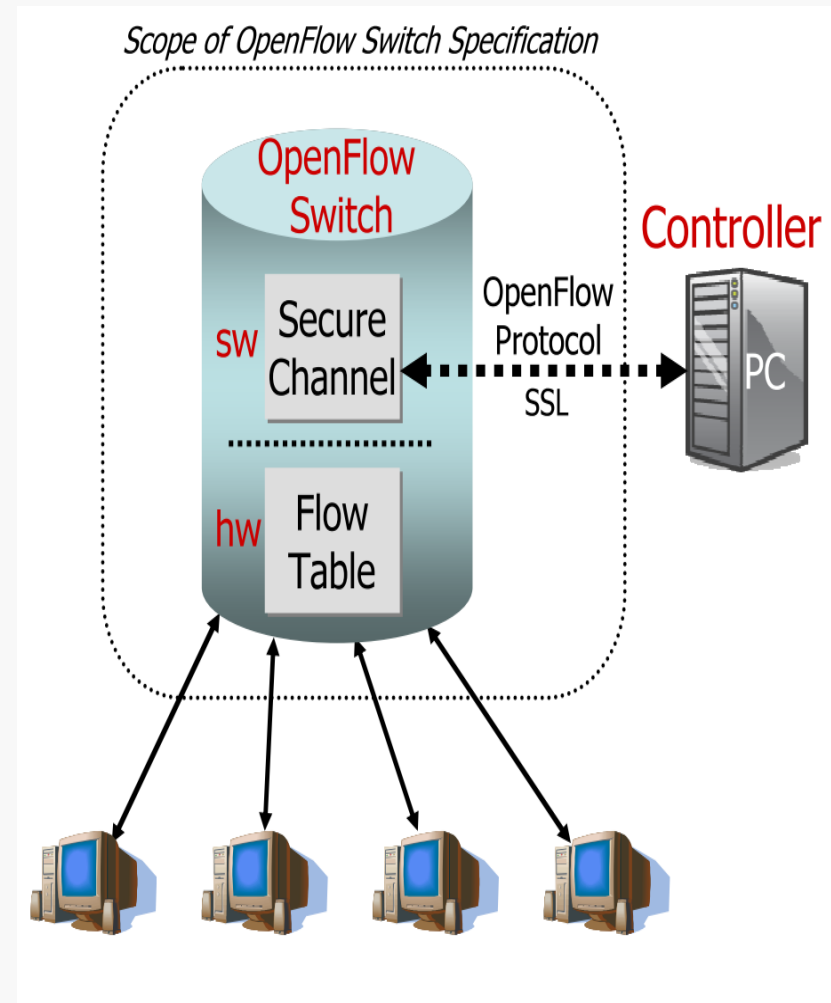
- » az absztrakció elrejti az elosztott állapot részleteit

SDN absztrakció a vezérlő síkon



OpenFlow

- » Nyílt interfész „fekete dobozként” kezelhető hálózati eszközökhöz (útválasztó, L2/L3 kapcsoló)
- » Szeparált vezérlő és adat sík
 - » Egy OpenFlow kapcsoló adat síkja tartalmazza a folyam táblázatot (Flow Table), és egy műveletet (action) minden bejegyzéshez
 - » A vezérlő síkban: vezérlő (controller), amely a folyam bejegyzéseket felprogramozza
- » Az OpenFlow tkp. egy standard interfész, amin keresztül hozzáadhatók és törölhetők bejegyzések egy Ethernet kapcsoló belső folyam táblázatához





OpenFlow eszközök

Vezérlő/NOS

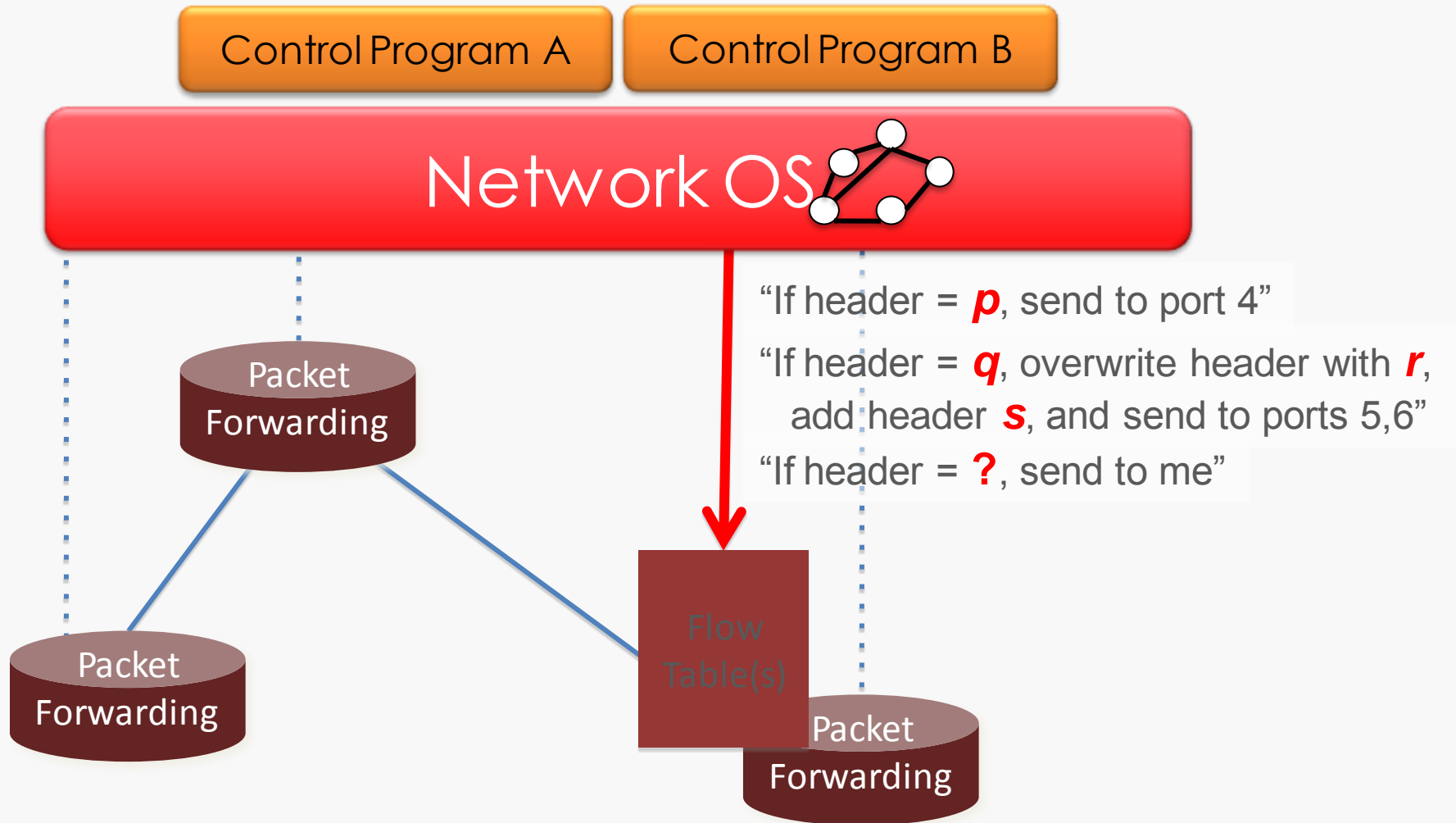
- » POX (Python)
 - » általános SDN vezérlő
- » NOX (C++)
 - » az első OpenFlow controller
- » Floodlight (Java)
 - » ipari szintű megoldás
- » OpenDaylight (Java)
 - » NFV
- » Ryu (Python)
 - » nyílt forrású Network Operating System (NOS)
- » ovs-controller (C)
 - » referencia vezérlő az Open vSwitch-hez
- » ...

Kapcsolók

- » Szoftveres kapcsolók
 - » Stanford Reference Implementation v1.0
 - » Open vSwitch
 - » Linux-based Software Switch (Kernel Space implementáció)
 - » Nem csak OF kapcsoló, hanem hypervisorokban is alkalmazott
- » Szoftver → Hardver
 - » Általános hardveren
 - » OpenWRT-t futtatva
 - » szoftveres kapcsolók portolhatók
 - » CPU-n futtatva
 - » user space implementáció
 - » NetFPGA-alapú implementáció
- » Hardver kapcsoló gyártók
 - » HP, Cisco, Juniper, IBM, Arista, NEC, Netgear, Pronto, ...



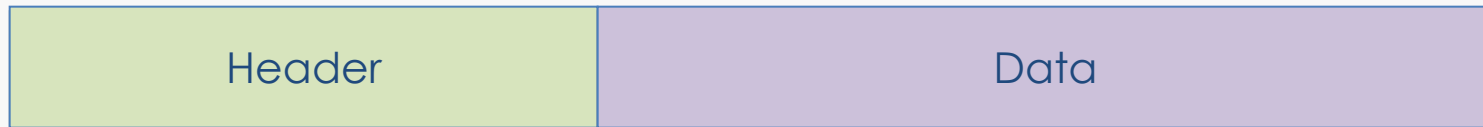
OpenFlow





OpenFlow szabályok, műveletek

- » Egyszerű csomagkezelési szabályok
 - » <Header match, Action>
 - » <Fejléc illeszkedési minta, művelet>
 - » tetszőleges bitminta megadható:



Match: 1000x01xx0101001x

- » Művelet (action)
 - » Továbbítás megadott port(ok)ra, eldobás, továbbküldés a vezérlőnek
 - » Fejléc felülírás, hozzáadás (push), levétel (pop)
 - » Továbbítás megadott bitsebességgel



Folyam táblázat

- » Lehet több is, át lehet irányítani egyikből a másikba a feldolgozást

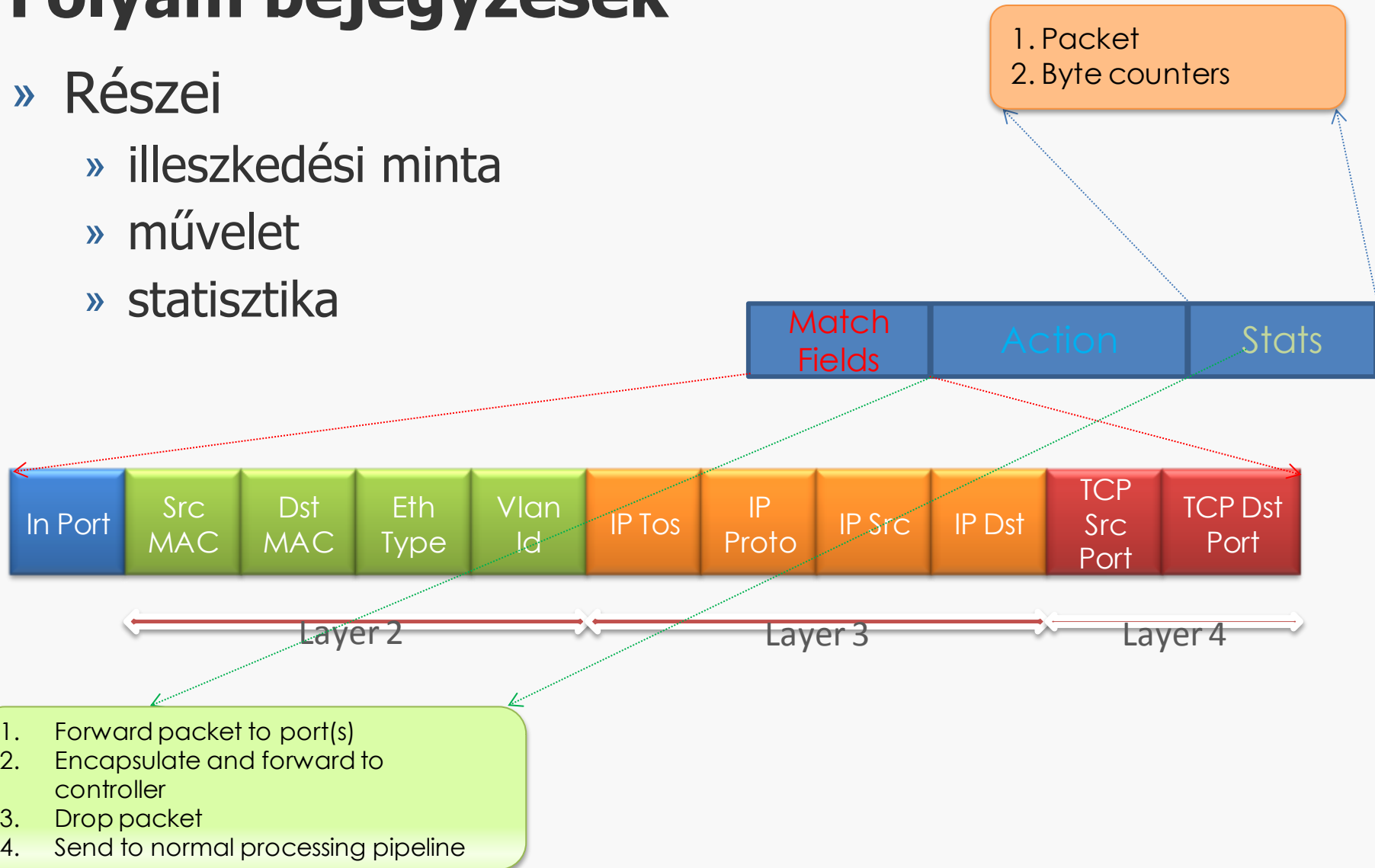
| | | | |
|---------|----------------------------|----------------|------------|
| Flow 1. | Rule (exact & wildcard) | Action | Statistics |
| Flow 2. | Rule (exact & wildcard) | Action | Statistics |
| Flow 3. | Rule (exact & wildcard) | Action | Statistics |
| | | | |
| Flow N. | Rule (exact & wildcard) | Default Action | Statistics |



Folyam bejegyzések

» Részei

- » illeszkedési minta
- » művelet
- » statisztika





Példák 1.

Ethernet kapcsolás

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

Folyam szintű kapcsolás

| 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 |

Tűzfal

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



Példák 2.

Útvonalválasztás

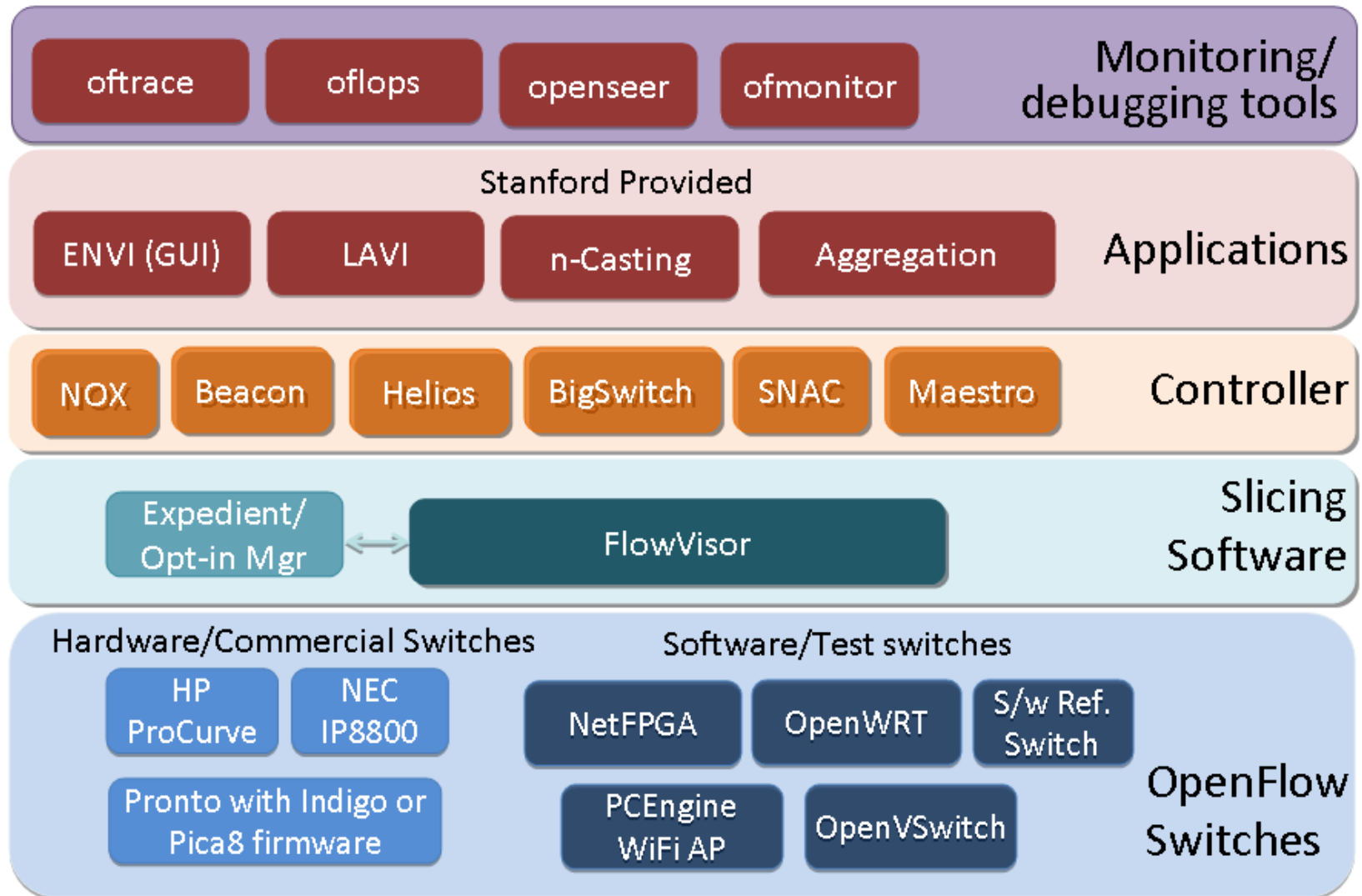
| Switch Port | MAC src | MAC dst | Eth type | VLAN ID | IP Src | IP Dst | IP Prot | TCP sport | TCP dport | Action |
|-------------|---------|---------|----------|---------|--------|---------|---------|-----------|-----------|--------|
| * | * | * | * | * | * | 5.6.7.8 | * | * | * | port6 |

VLAN kapcsolás

| 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 |

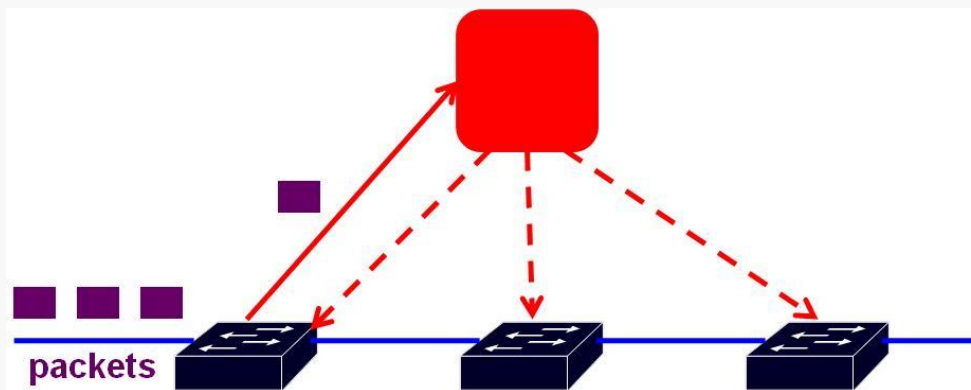


OpenFlow építőelemek



Folyam szintű csomagkezelés: reaktív

- » folyam első csomagja a vezérlőhöz továbbítva
- » a vezérlő felprogramozza a folyamnak megfelelő szabályokat az adatsíkon
 - » rendszerint egy szabály, de lehet több is (lista)
- » a vezérlő visszaküldi az első csomagot a hálózati eszköznek
- » a folyam további csomagjai a felprogramozott szabályok alapján továbbítódnak



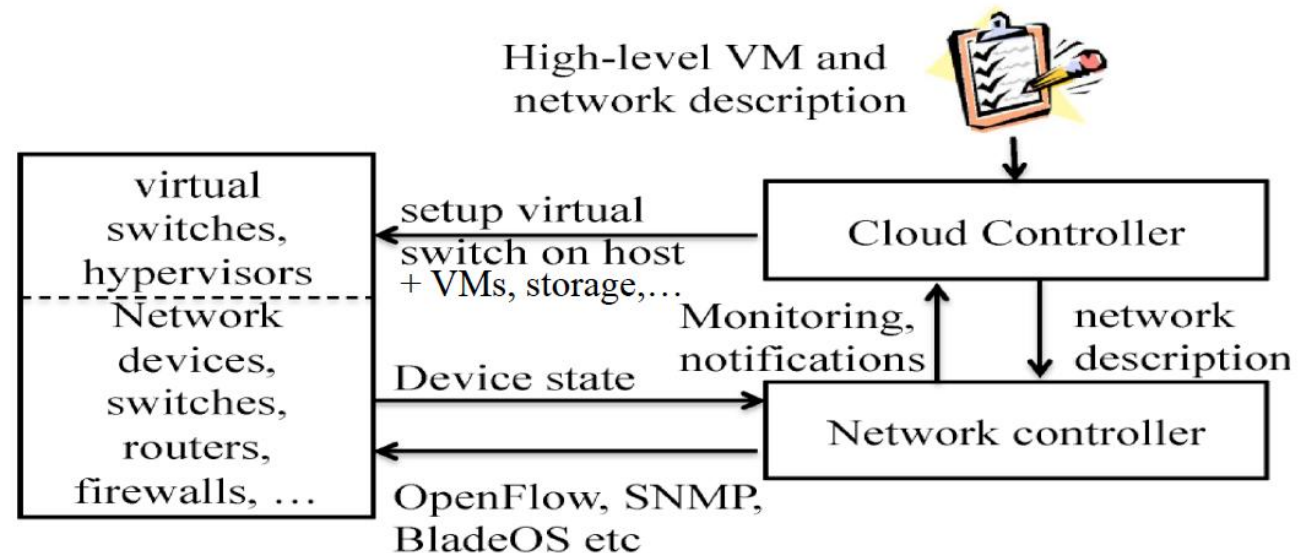
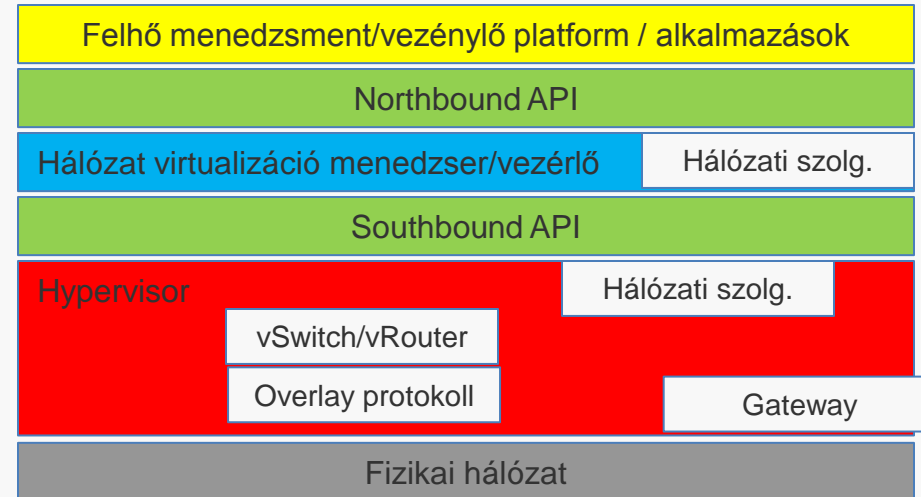


SDN a felhőben

- » Reaktív végpont-végpont hálózatok helyett...
 - » első csomag a vezérlőhöz \Rightarrow késleltetés
 - » végpont-végpont: sok bejegyzés, skálázhatósági probléma
 - » előfizetők/VM-ek változása minden kapcsolót érint
- » ...proaktív fedőhálózat (overlay)
 - » a fizikai hálózat L2/L3 összeköttetést biztosít
 - » a vezérlő előre felprogramozza az eszközöket \Rightarrow kis késleltetés
 - » alagutak: előfizető állapot csak a végpontokban (hypervisor virt. kapcs. / útválasztó), skálázható
 - » kevesebb bejegyzés a továbbítási táblázatokban
 - » nem a VM-ek, hanem csak fizikai szerverek közötti kapcsolatok
 - » előfizetők változása a fizikai hálózatot nem érinti

Felhő menedzsment és SDN

- » Orchestration (vezénylés): OpenStack biztosítja
 - » magasabb szintű absztrakció
 - » a virtuális erőforrásokat látja
 - » nem csak a hálózat, hanem egy teljes alkalmazás rendszer
 - » VM-ek, háttértárak, stb. + hálózat
 - » CLI vagy horizon dashboard
 - » automatizált: Heat
 - » sablonok
- » SDN
 - » a fentiek alacsonyabb szintű hálózati megvalósítása



OpenStack

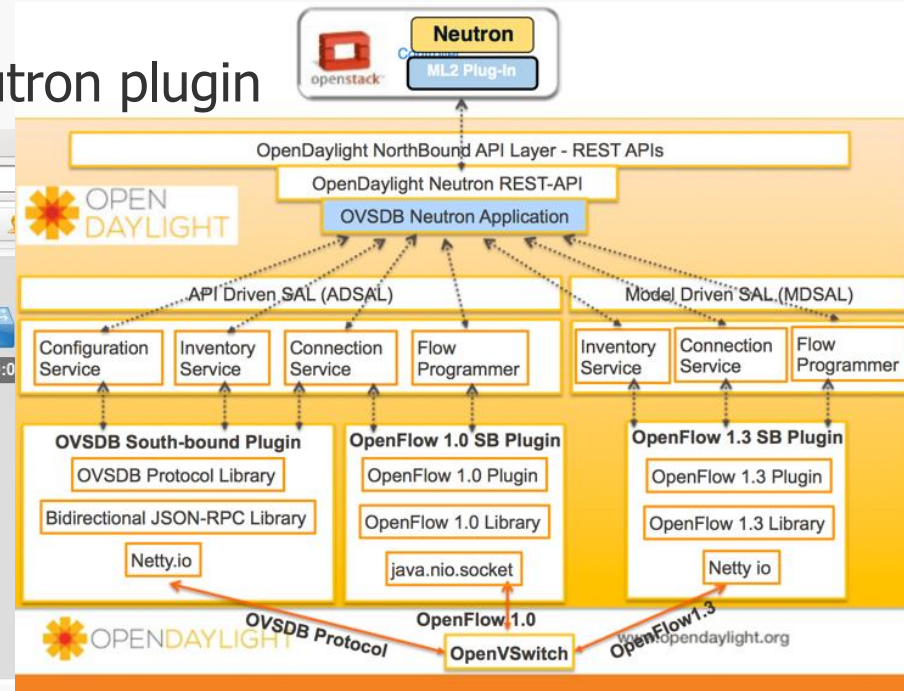
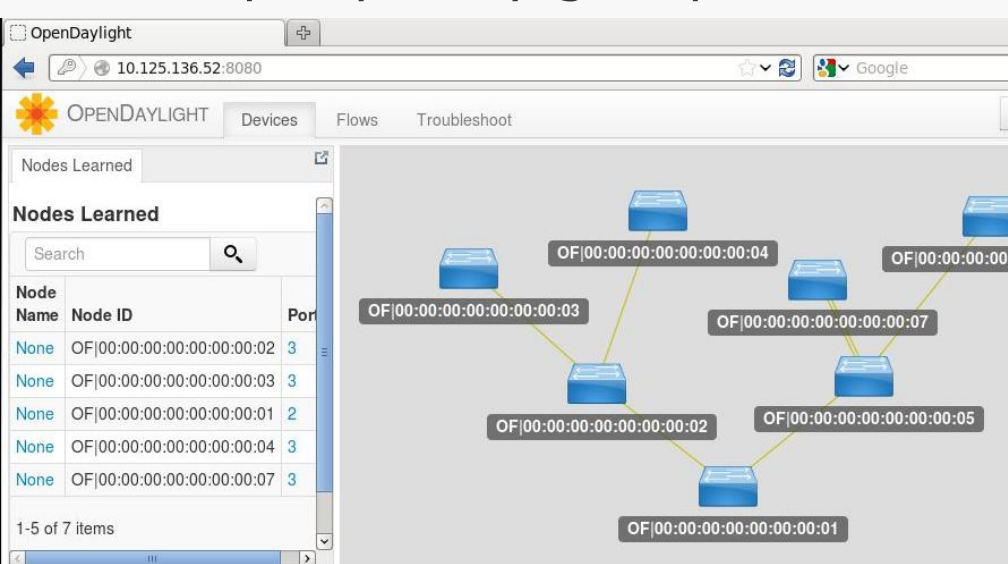
» OVS Neutron plugin

» OpenFlow a virtuális kapcsoló táblázatok felprogramozására

- » VM MAC címe és a szerver hypervisor transzport IP címe közötti leképezés – ezt a vezénylés (orchestration) számára ismert
- » proaktív
- » északi interfész (northbound): Neutron
- » déli interfész (southbound): OpenFlow

» Lehet más SDN vezérlő plugin

- » pl. OpenDaylight OpenStack Neutron plugin



SND a felhőben

- » Nem csak a virtuális kapcsolók/útválasztók beállítására, hanem a fizikai hálózati eszközökre is

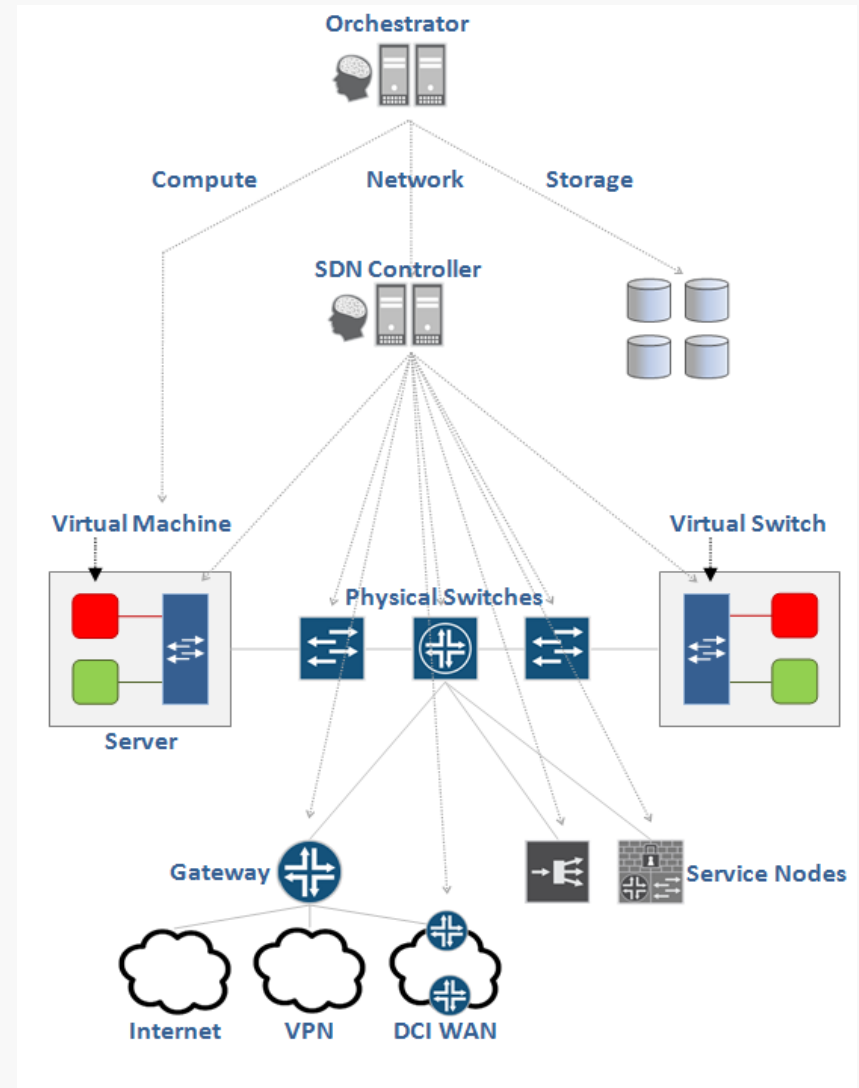


Figure 22: The Role of Orchestration in the Data Center

Forrás: <http://www.opencontrail.org/opencontrail-architecture-documentation/>



Adatközpont hálózati követelmények

- » Kapcsolók konfigurációjának és állapotának minimalizálása
 - » automatizálás, amennyire lehetséges
- » Hatékony forgalom továbbítás, nagy teljesítmény
 - » ne legyen hurok
 - » alkalmazkodás a forgalmi változásokhoz
 - » ügyfél SLA betartása
- » VM migráció gyorsan és könnyen
 - » transzparens migrálás
- » Gyors, hatékony hiba felderítés/elhárítás
 - » elég gyakori a nagy méretből adódóan
 - » a hálózatnak is igazodnia kell a hibaelhárításhoz



Tradicionális megoldások

- » Layer 3
 - + hierarchikus címzés \Rightarrow kis továbbítási táblázatok
 - + OSPF gyors hibakezelés
 - + IP TTL: hurkok kivédése
 - magas az adminisztrációs teher (alhálózatok konfigurálása, DHCP, stb.)
- » Layer2
 - + Flat MAC címzés (helyfüggetlen)
 - + hurkok kivédése: STP
 - + kevesebb az adminisztrációs teher
 - broadcast forgalom (nem jól skálázható)
 - STP nem tudja kihasználni a teljes topológiát
- » VLAN
 - » skálázhatóság (max. 4K)
 - » statikus konfigurációból származó hátrányok



SDN megoldás

- » a vezérlő teljes hálózati képet kap
 - » eszközök felderítése
 - » MAC, IP címek, kapcsolatok
- » a vezénylés által adott feladat alacsonyabb szintű hálózati megvalósítása
- » gyors és dinamikus hálózat kialakítás
 - » rugalmas: ügyfelek által megadott módon
 - » automatizált hálózati erőforrás kiosztás/kezelés
 - » forgalmi terhelés optimalizálása, akár adatközpontok között
- » skálázhatóság
- » NFV

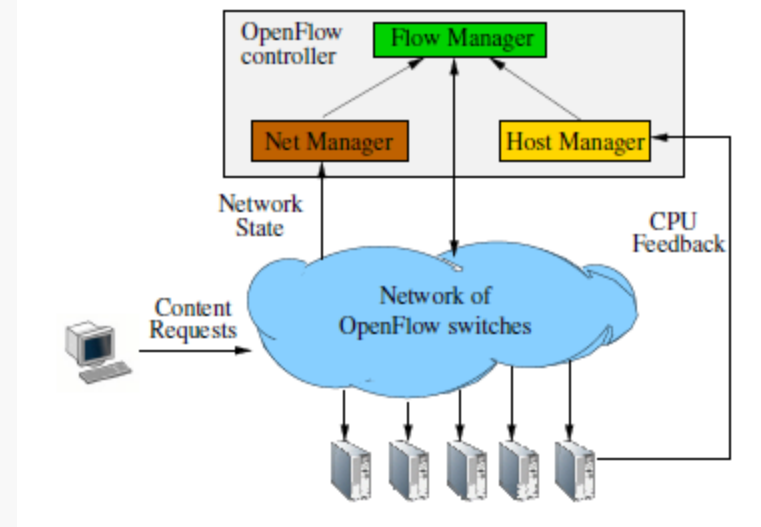
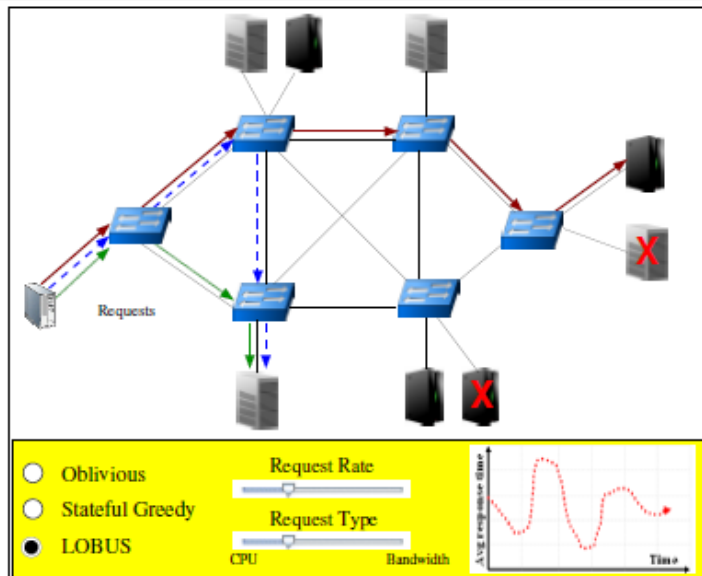


Felhő specifikus feladatok

- » terheléskiegyenlítés (Load Balancing – LB)
- » adatközpontok közötti alagút
- » VM migrálás
- » skálázható csomagtovábbítás

Terheléskiegyenlítés

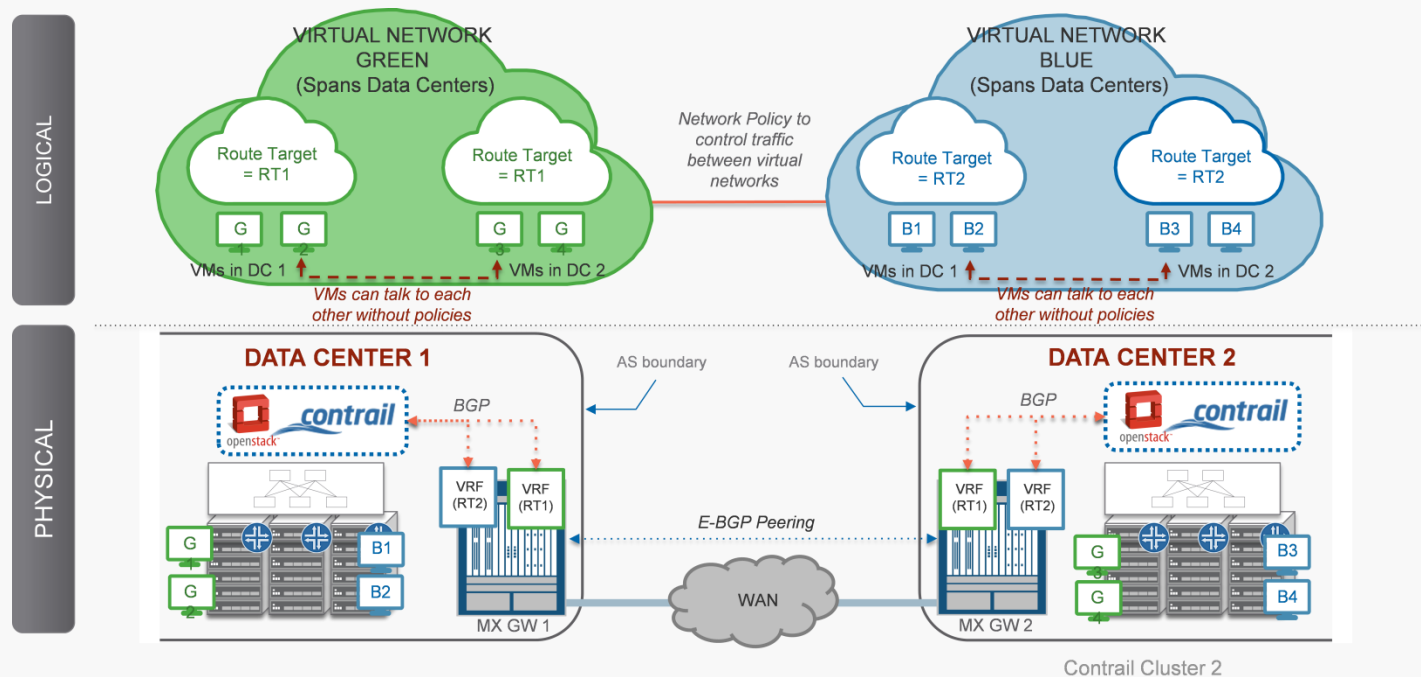
- » Dinamizmus
 - » az OpenFlow bejegyzésekhez időzítő tartozik
- » Terheléskiegyenlítéshez szükséges műveletek
 - » a publikus IP cím átírása a kiszolgáló IP címére
 - » a kiszolgálóhoz tartozó kimeneti portra továbbítás
 - » az ellenkező irányba fordítottan ugyanez
- » Megoldandó
 - » hash alapú útválasztás
 - » TCP flag vizsgálat az új folyamatok megkülönböztetésére
- » Plug-n-Serve: Load-Balancing Web Traffic using OpenFlow
 - » terheléskiegyenlítés a hálózat és a kiszolgálók terhelése alapján, elosztott módon



Forrás: <http://conferences.sigcomm.org/sigcomm/2009/demos/sigcomm-pd-2009-final26.pdf>

SDN adatközpontok közötti forgalomra

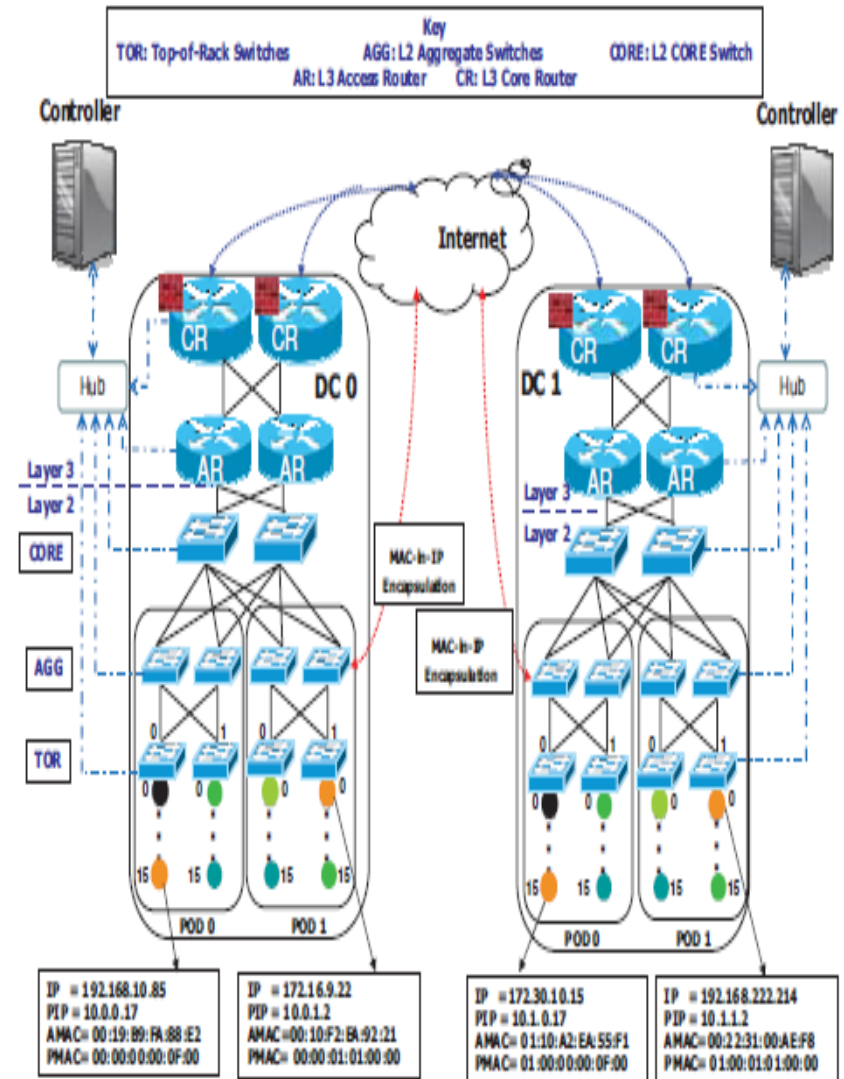
- » Forgalom
 - » cloud bursting
 - » földrajzi szempontok a terheléskegyenlítésben
- » Alagutak kiépítése reaktív módszerrel
 - » multipath
 - » útvonalak változtatása = fejlécek átprogramozása menet közben



Forrás: <http://www.opencontrail.org/how-to-setup-opencontrail-gateway-juniper-mx-cisco-asr-and-software-gw/>

VM migrálás

- » Okai
 - » karbantartás, terhelés kiegyenlítés
 - » VM-ek összerendezése (energiatakarékosság)
 - » katasztrófa elhárítás: teljes alkalmazás rendszer áttelepítés
- » Másik alhálózatba migrálás nehézségei
 - » hierarchikus IP címzés
 - » kézi átkonfigurálás nem életképes megoldás
 - » az élő TCP kapcsolatok ne szakadjanak meg
- » CrossRoads
 - » helyfüggetlenség: pseudo MAC (PMAC) és IP címek (PIP)
 - » SDN vezérlő kezeli az összerendeléseket



Forrás: Mann, V.; Vishnoi, A; Kannan, K.; Kalyanaraman, S., "CrossRoads: Seamless VM mobility across data centers through software defined networking," *Network Operations and Management Symposium (NOMS), 2012 IEEE*, vol., no., pp.88,96, 16-20 April 2012



SDN skálázhatóság

- » Kihívás a vezérlő sík számára
 - » VM-ek száma, ügyfél szabályok, SLA-k, folyamatok száma, stb.
- » multi domain környezetben vezérlők szövetsége (federation)
 - » információcsere
 - » állapotok megosztása
 - » könnyen bővíthető
- » NEC 2014. tesztek
 - » Trema OpenFlow vezérlő
 - » Layer 2 hálózatok VXLAN technológiával
 - » vezérlő terhelés kiegyenlítésével: több kiszolgáló
 - » egy kiszolgáló 410 kapcsolót kezel, lineáris skálázódás
 - » 16 000 virtuális hálózatot kezel
 - » 1024 kapcsoló, mindegyiken 128 VM
 - » konstans 4 mp egy virtuális hálózat kialakítása



Alkalmazások

- » Amazon, Google, Facebook, Microsoft Azure
 - » saját egyedi SDN megoldások
- » Google inter-datacenter WAN: SDN + OpenFlow
 - » központosított forgalom szervezés (traffic engineering)
 - » hálózati költségek csökkentése
- » NEC által telepített adatközpontokban
 - » költségek csökkentése
- » VMware
 - » Nicira (SDN, hálózat virt.)
 - » Network Virtualization Platform (NVP): overlay hálózati technológia ⇒ VMware NSX



Források

- » Nick McKeown (Stanford University), "Software-defined Networking", Infocom Keynote Talk, April 2009, Rio de Janeiro, Brazil
- » Srini Seetharaman, OpenFlow/SDN tutorial, Nov 2011
- » Jennifer Rexford (Princeton University), Computer Science 461: Computer Networks, Software Defined Networking
- » Matt Davy (Indiana University), Software Defined Networking & OpenFlow, GENI Workshop, July 7th, 2011
- » Open Networking Foundation,
<https://www.opennetworking.org/>
- » <http://www.openflow.org/>
- » CHIBA Yasunobu, SUGYOU Kazushi, „ OpenFlow Controller Architecture for Large-Scale SDN Networks”, NEC Technical Journal/Vol.8 No.2/Special Issue on SDN and Its Impact on Advanced ICT System, 2014