Felhő alapú hálózatok (VITMMA02) Adatközpont hálózati topológiák, Ethernet kiegészítések

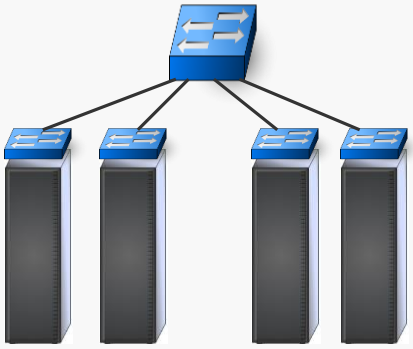
Dr. Maliosz Markosz

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

2020. tavasz

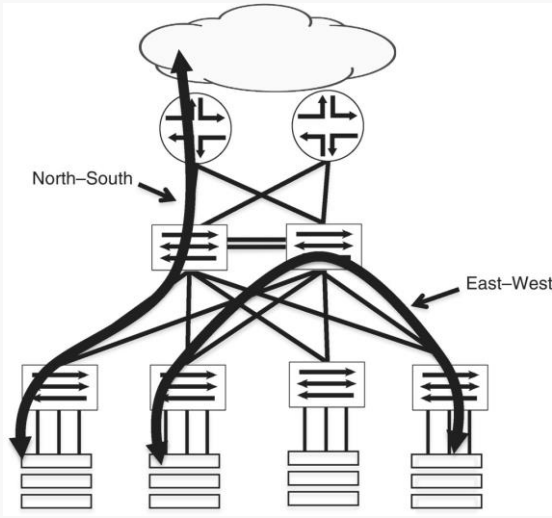
Hálózati topológiák

- 3 szintű hierarchia: ToR, aggregáló, központi kapcsoló
- lapos(abb) topológia, 2 szint: ToR és központi kapcsoló
  - egy nagy kp.-i kapcsoló: költséges, limitált portszám
    - pl. egy 128 portos GbE kapcsoló ára kb. 100-szorosa egy 48 portosénak



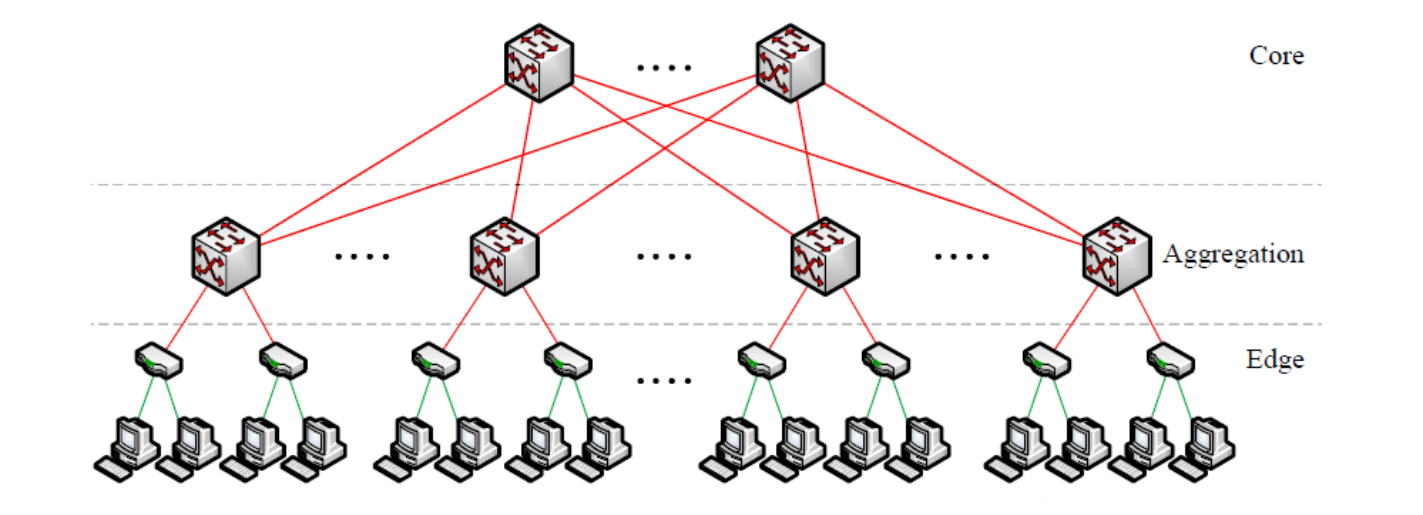
Adatközpont forgalmi minták

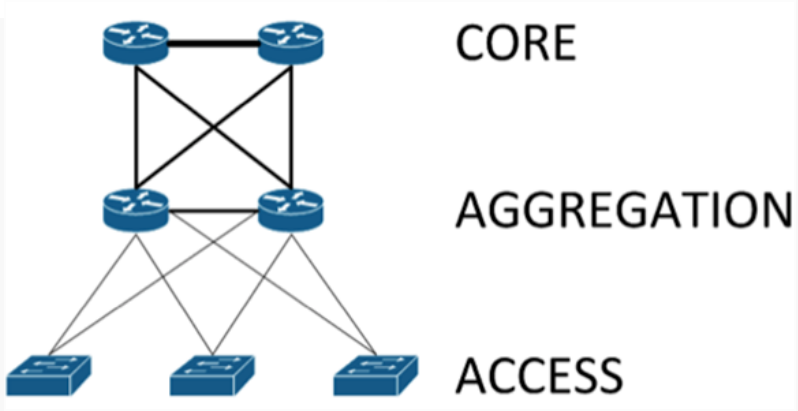
- Forgalom iránya
  - észak-dél: szerverek és központi kapcsoló között
  - kelet-nyugat: szerverek közötti
    - pl. VM migrálás, háttértár replikálás
- Kérés-válasz
  - régen: egy kliens kérésre egy szerver válaszol
  - ma: egy kérés szerverek egymás közötti interakciójával kapunk választ
    - pl. egy Google map keresés esetén
      - a címet egy helyi kereső kiszolgáló felé továbbítani
      - az eredmény alapján a térkép szervertől a térkép adatokat lekérni
      - közeli helyek keresése és megjelenítése
      - a kliens előzményeinek lekérdezése
      - ez utóbbi alapján célzott hirdetés megjelenítése



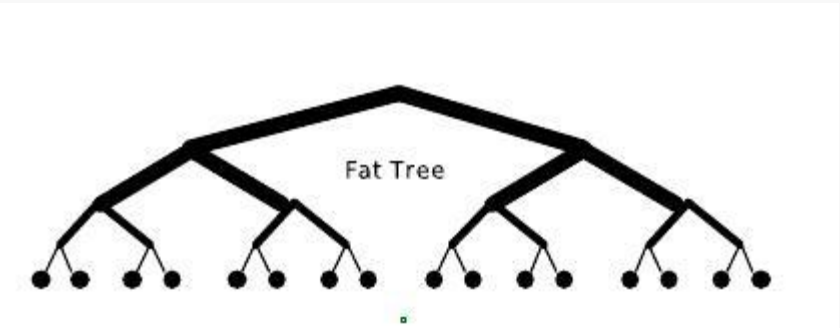
Hálózati topológiák

- Redundancia és/vagy terheléselosztás
  - kettős csillag

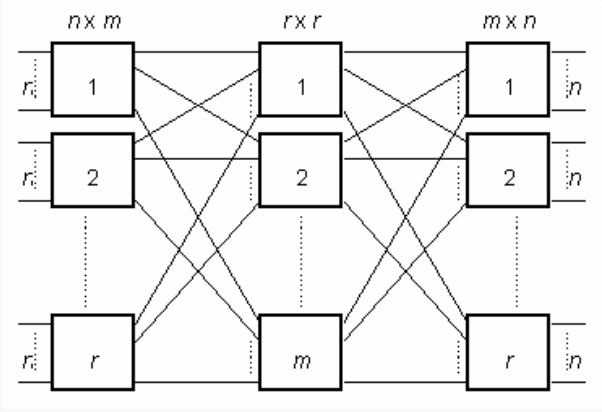




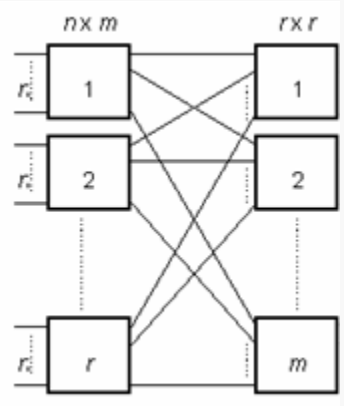
Fat-tree topológia



- Fat-tree
  - 1:1 oversubscription
  - felfelé egyre nagyobb sávszélesség
  - nem praktikus: változó portszám
- Többfokozatú kapcsolás
  - Charles Clos 1952, telefonhálózatra

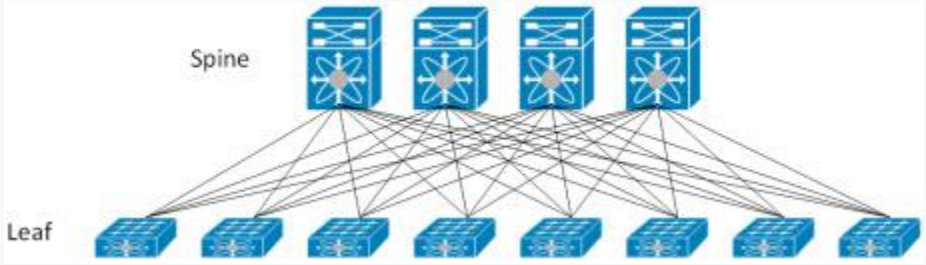


- Összehajtott többfokozatú kapcsolás
  - folded Clos
  - ki és bemenet összevonva
  - ezt is fat-tree-nek nevezik



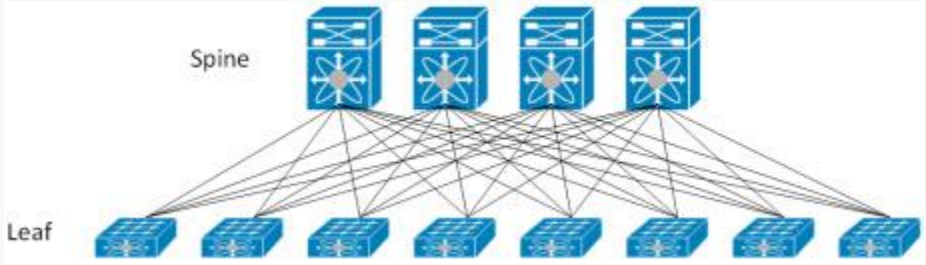
Fat-tree topológia az adatközpontban

- teljes szövevény: vezetékezés komplex
- levél és gerinc kapcsolók
- terheléskiegyenlítés a gerinc kapcsolókon, ECMP
- azonos eszközökből felépíthető
- ***k*** portos azonos kapcsolók
  - levél: ***k***/2 lefelé, ***k***/2 felfelé (max. ***k***/2 db gerinc kapcsoló) – 1:1 oversubscription
    - ezért nevezik fat-tree-nek
  - gerinc: ***k*** port ⇨ max. ***k*** db levél kacsoló
  - összesen max.
    - 1,5x ***k*** db kapcsoló
    - ***k*** x ***k***/2 db szerver a levél kapcsolókhoz csatlakoztatva



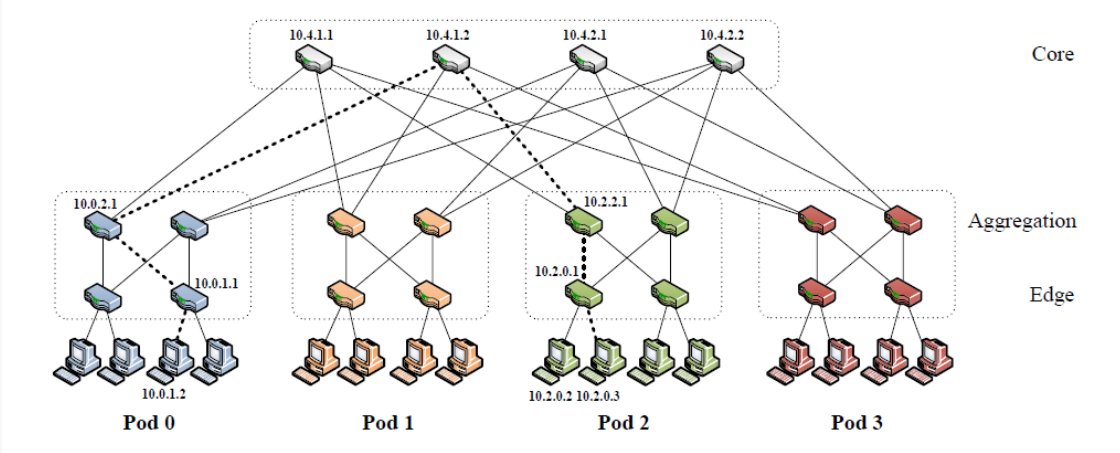
Fat-tree topológia az adatközpontban

- Terheléskiegyenlítés
  - ideálisan: forgalom egyenletes szétosztása a gerincen
  - valóság
    - folyam alapú terheléselosztás
      - round robin
      - hash
    - jumbo keretek (9kB)
    - levél kapcsolók nincsenek koordinálva
- Hibatűrés
  - gerinc kapcsoló meghibásodás
    - összes összeköttetés él csökkentett sávszélességgel
  - levél kapcsoló meghibásodás
    - a kapcsolódó szerverek elérhetetlenek
    - védekezés: két hálózati kártya, két különböző levél kapcsolóhoz



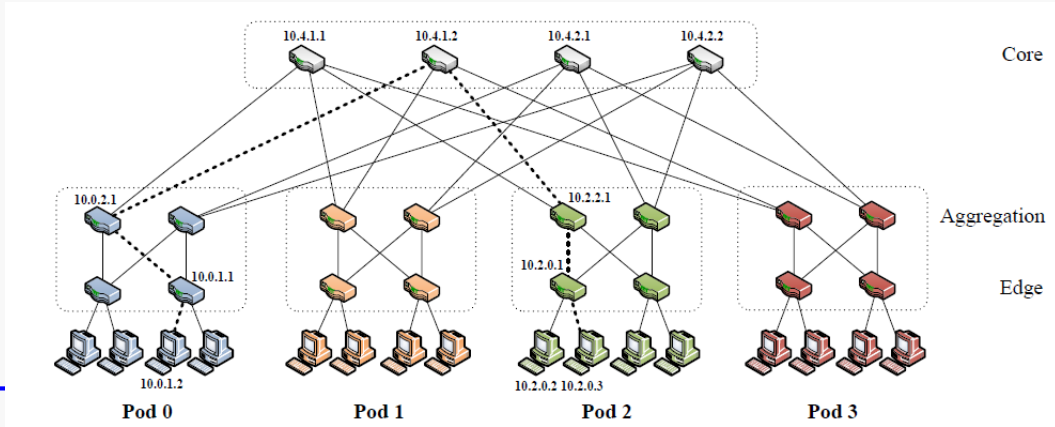
Fat-tree topológia az adatközpontban

- Egy séma
  - k portos kapcsolók
  - k csoport (pod)
  - k/2 edge és aggr. kapcsoló / csoport
  - core kapcsoló mindegyik csoporthoz csatlakoztatva
    - k/2-es egységenként az aggr. kapcsolókhoz



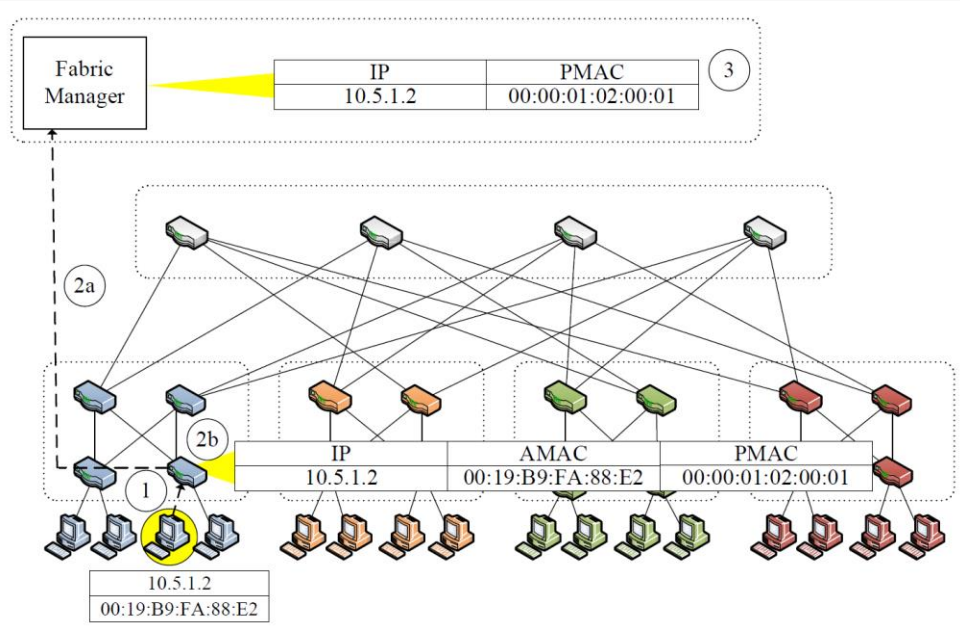
Fat-tree topológia az adatközpontban

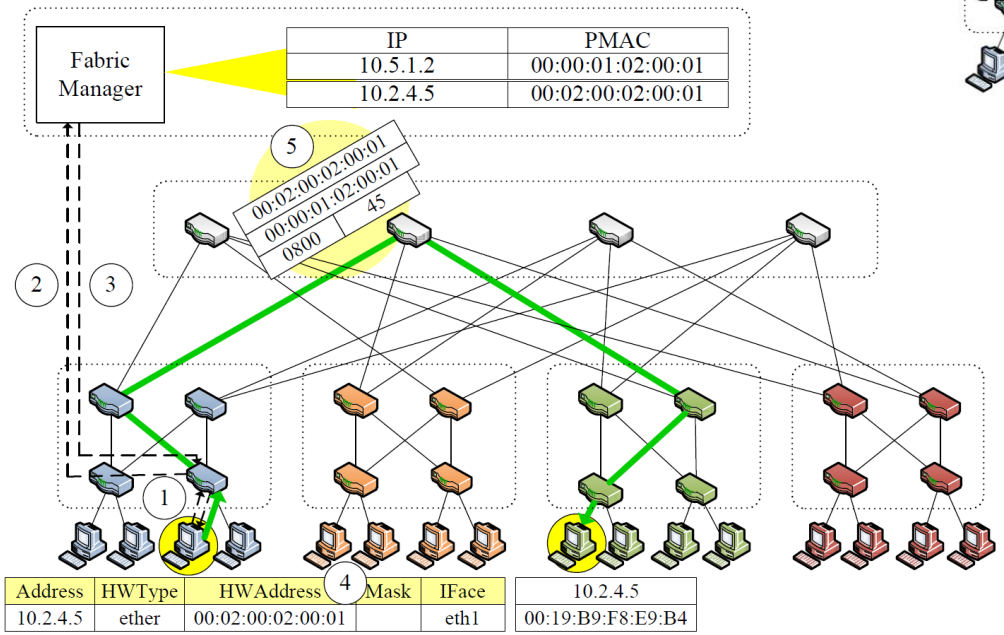
- #szerver:
  - #edge sw * #edge ports = (k * k/2) * (k/2) = $k^3/4$ szerver
- #kapcsoló: #POD sw + #core sw
  - #POD sw: k*k
  - #core sw : uplink from aggr= k * k/2 * k/2 = k * x -> x = $(k/2)^2$
  - Összesen: k*k+ $(k/2)^2$ = 5/4 k^2 kapcsoló
- #ECMP utak:
  - egy core sw-től egy adott szerverig: 1
  - Egy adott szervertől egy tetszőleges core sw-ig: $(k/2)^2$
  - Összesen: $(k/2)^2$ ECMP út
- #összeköttetés:
  - Core – aggr: k * k/2 * k/2 = $k^3/4$
  - PODokon belül: k * k/2 * k/2= $k^3/4$
  - Összesen: = $k^3/2$
- STP esetén hány összeköttetés van használatban:
  - #kapcsoló – 1 = 5/4 k^2 – 1
- ábra: k=4
- k=48
  - 27 648 szerver
  - 2 880 kapcsoló
  - 576 ECMP út



L2 topológia szerinti címzés

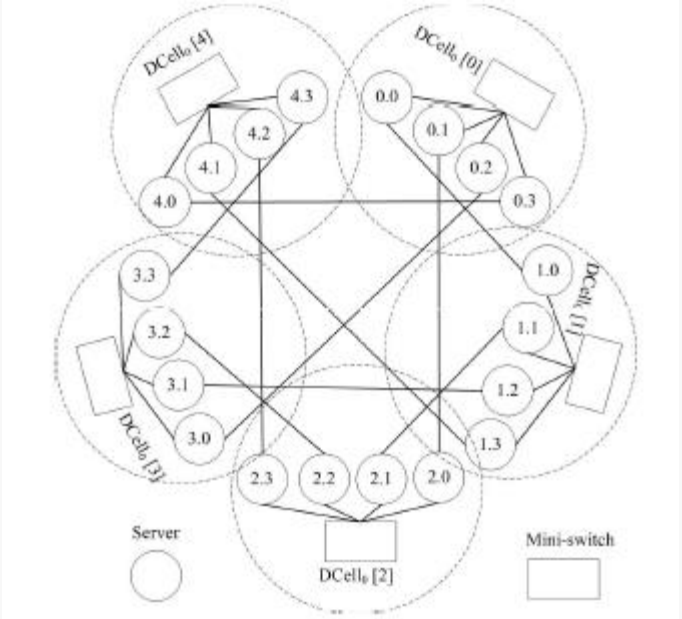
- Portland
- Pseudo MAC (PMAC)
  - topológia alapján:
    - pod:pozíció:port:vmid
- Fabric manager
  - ARP kérések kezelése
- Location Discovery Protocol





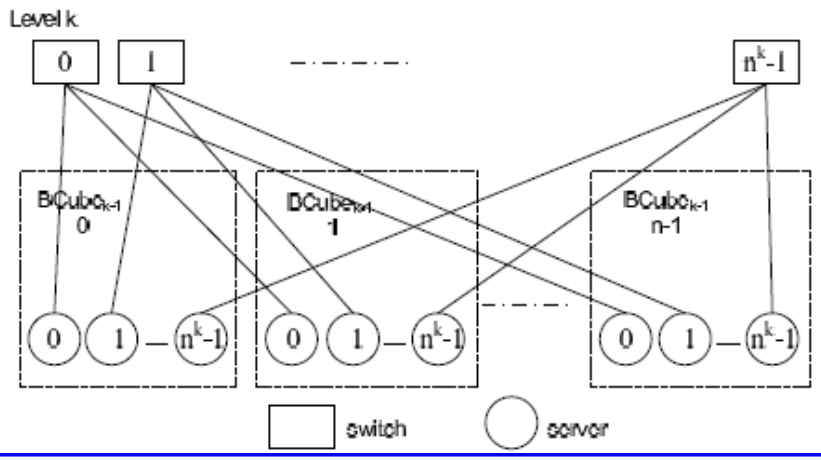
Hibrid hálózat: szerver és kapcsolók

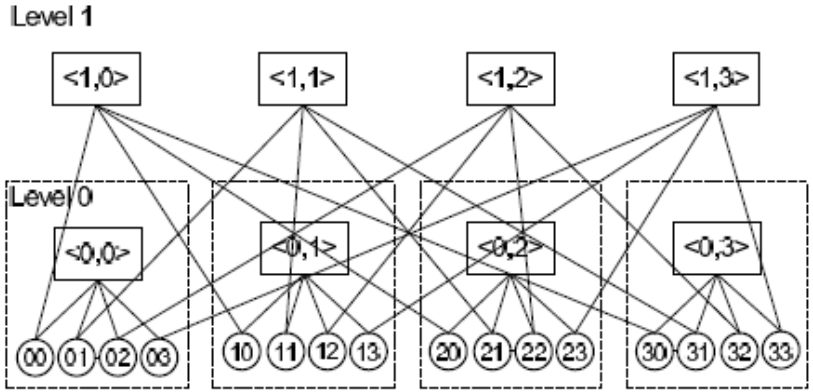
- Rekurzív topológia modell: DCell
- Inkrementális bővítés
- Szintek
  - 0. szint: n szerver és 1 kapcsoló
  - k+1. szint: k. szint szervereinek száma +1 db k szintű cellák összekötve teljes szövevényben
- Hibrid megoldás
  - cellán belül a kapcsolón keresztül
  - cellán kívül a szervereken, mint útválasztókon keresztül
    - először a kezdő és végpontot tartalmazó két azonos szintű cella közötti, majd a cellákon belüli út meghatározása
    - nem min hop
- Robusztus
  - sok alternatív út
- Teljesítmény
  - átviteli sávszélesség függ a hálózat méretétől
  - több köztes pont



Hibrid hálózat: szerver és kapcsolók

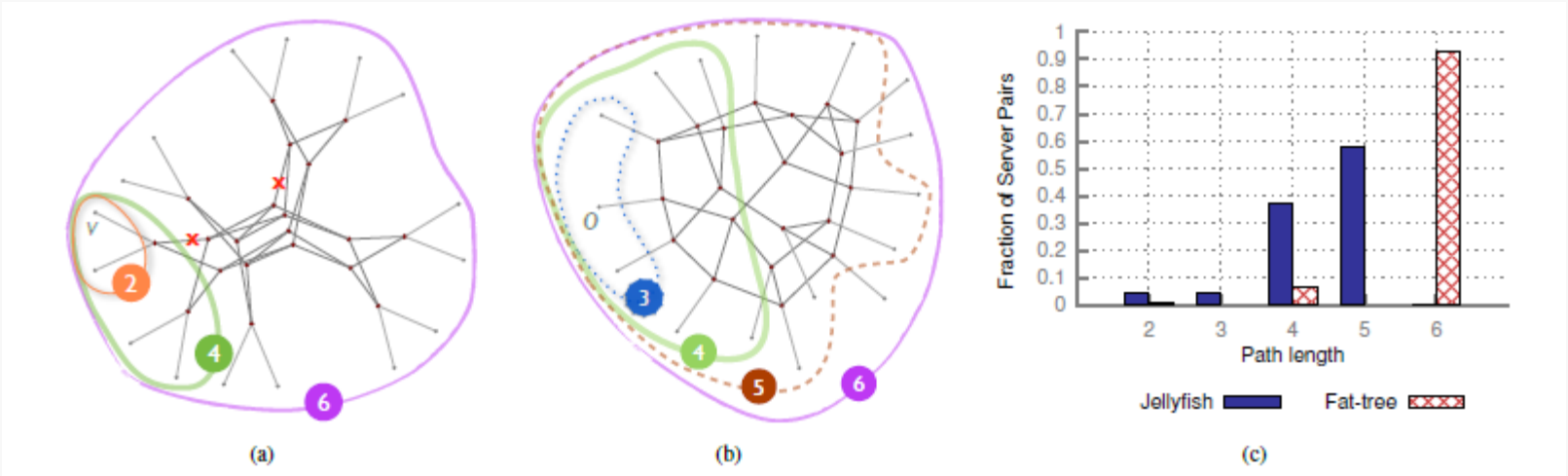
- BCube: konténer alapú moduláris adatközpont egységek számára
  - 1000-es nagyságrendű szerver
- Jellemzők
  - fokozatos teljesítménycsökkenés hiba esetén
  - kis átmérőjű hálózat
  - sok párhuzamos kapcsolat a szerverek között
  - forrásból számított utak
    - többutas
    - hálózati próbák
- Rekurzív topológia modell
  - Szintek
    - 0.: n szerver egy n portos kapcsolóval összekötve
    - k.: n db k-1. szintű BCube és n^k n portos kapcsoló
  - k. szinten
    - n^{k+1} szerver
    - szerverek: k+1 port
    - k+1 szint kapcsolókból, minden szinten n^k n portos kapcsoló





Jellyfish topológia

- Véletlen gráffal köti össze a ToR kapcsolókat
- Inkrementális bővítés
- Különböző portszámú kapcsolók
- Előnyök
  - az átlagos úthossz kisebb
  - ugyanannyi kapcsolóval több szervert összeköt, mint a fat-tree

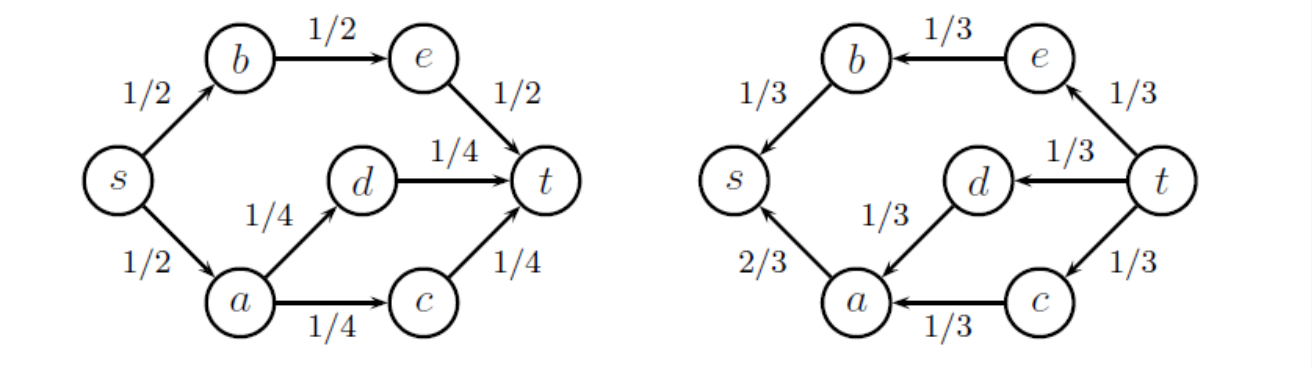


Sávszélesség kihasználás

- Ethernet Spanning Tree Protocol
  - feszítőfa: kihasználatlan kapacitások
  - Rapid STP (RSTP)
  - Multiple STP (MSTP)
  - tetszőleges és változó topológiára ideális
- Adatközpontokban ez nem ideális
  - strukturált és nem gyakran változó hálózat
  - megoldások
    - Equal Cost MultiPath (ECMP) útvonalválasztás
    - Shortest Path Bridging (SPB)
    - Transparent Interconnection of Lots of Links (TRILL)

ECMP

- Equal Cost MultiPath

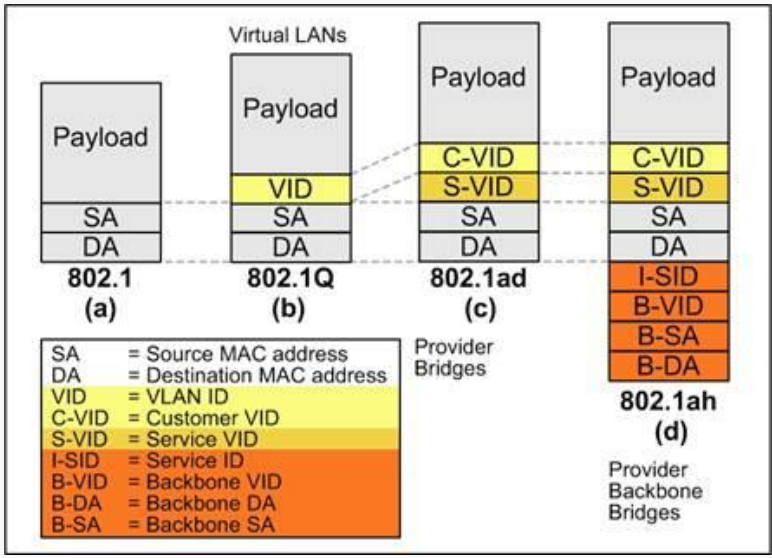


- nem igazán alkalmazzák általánosan
  - ha az utak találkoznak a célnál, akkor csak a komplexitás nő, a sávszélesség kihasználás nem
  - virtuális hálózat ⇔ fizikai hálózat

Shortest Path Bridging

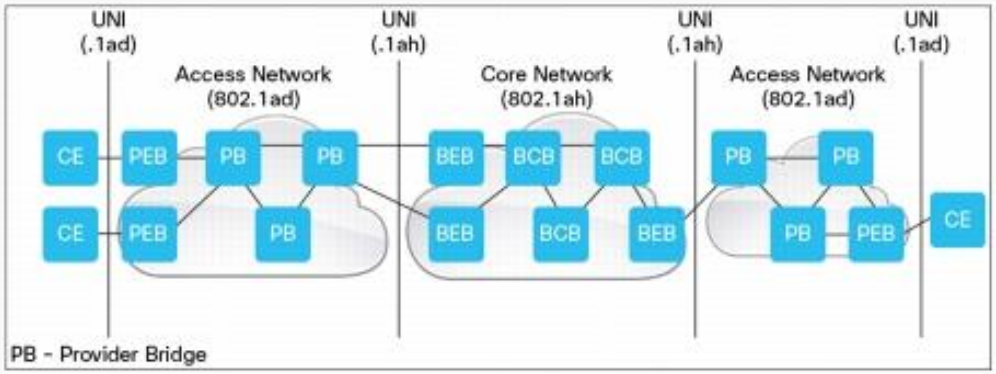
- Előzmények: Carrier Ethernet
  - Provider Bridging (PB) 802.1ad
  - Provider Backbobe Bridging (PBB) 802.1ah
- Shortest Path Bridging (SPB) 802.1aq

Carrier Ethernet



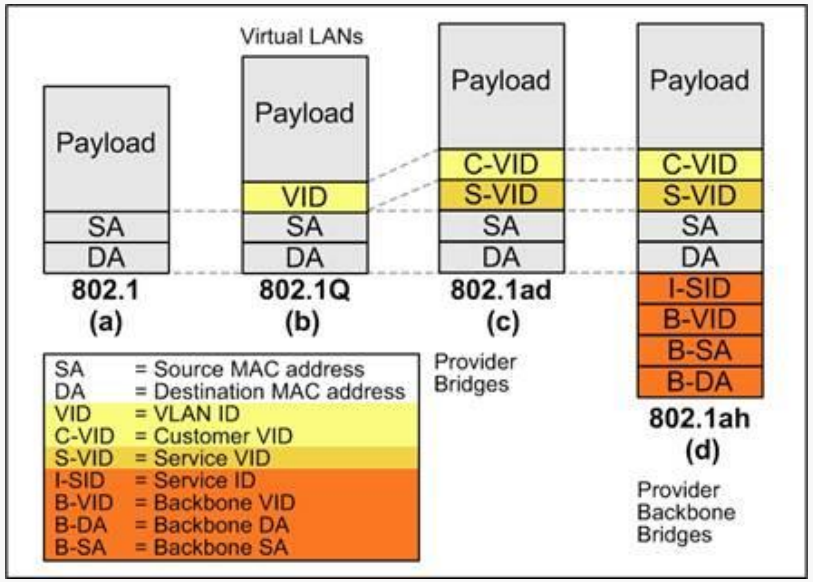
Ethernet a szolgáltatói hálózatban (MAN, WAN)

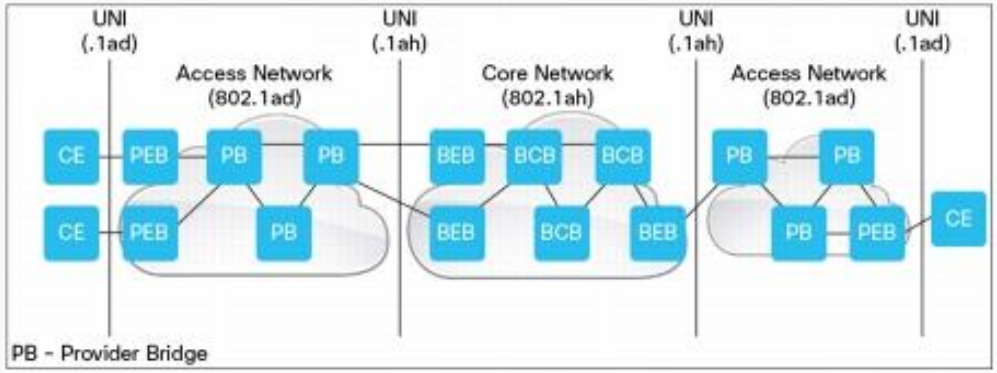
- Ethernet szolgáltatás több előfizetőnek
  - előfizetők szeparálása
- hierarchiaszintek bevezetése
  - előfizetői VLAN információ megtartása
  - szolgáltatás példányok (előfizetők) szeparálása (PB)
    - Q-in-Q: Customer tag, Service tag
    - két VLAN ID (VID)
    - 4096 szolgáltatás példány (felső korlát)
  - szolgáltatói és előfizetői címtér szeparálás (PBB)
    - MAC-in-MAC: külön címtartomány
      - a szolgáltató kapcsolóinak nem kell az előfizetői címeket kezelnie
    - service tag: 24 bites I-SID (service identifier) 16M szolg. példány
    - a szolgáltatás és transzport réteg szétválasztva: I-SID és B-VID



Carrier Ethernet

- Virtuális hálózatok leképezése a határon
  - C-VID ⇨ S-VID ⇨ I-SID ⇨ B-VID
  - Edge Brigdes
- A maghálózatban VLAN ID és cél MAC alapján
  - Core Bridges

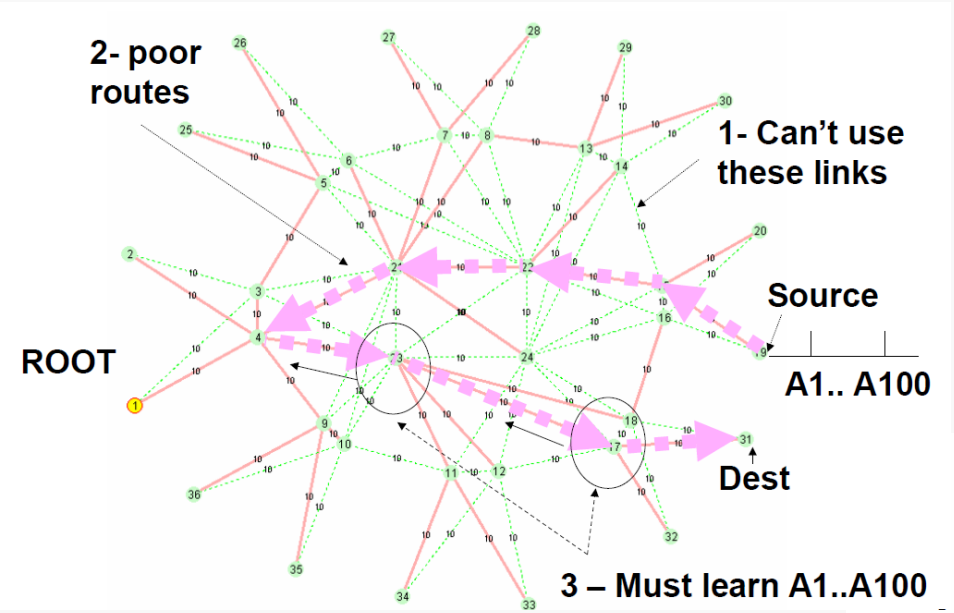


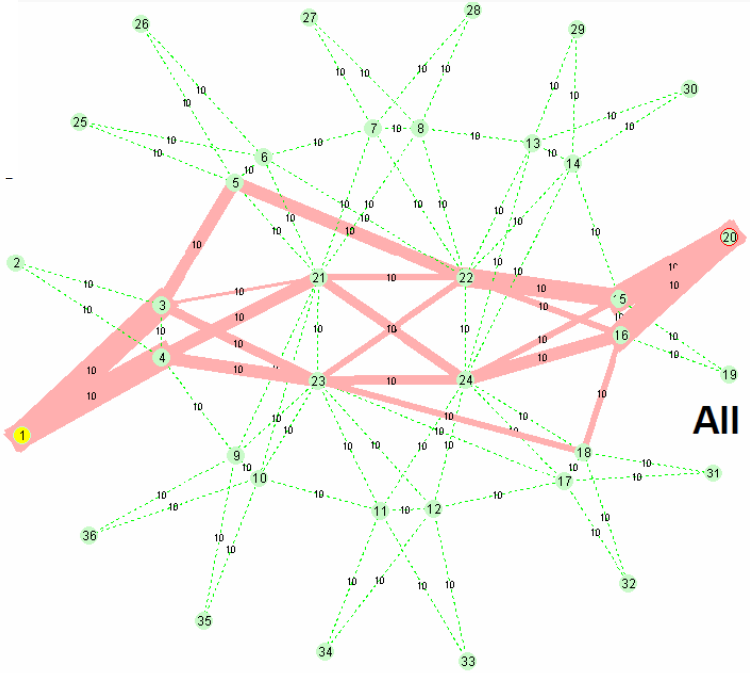


Shortest Path Bridging

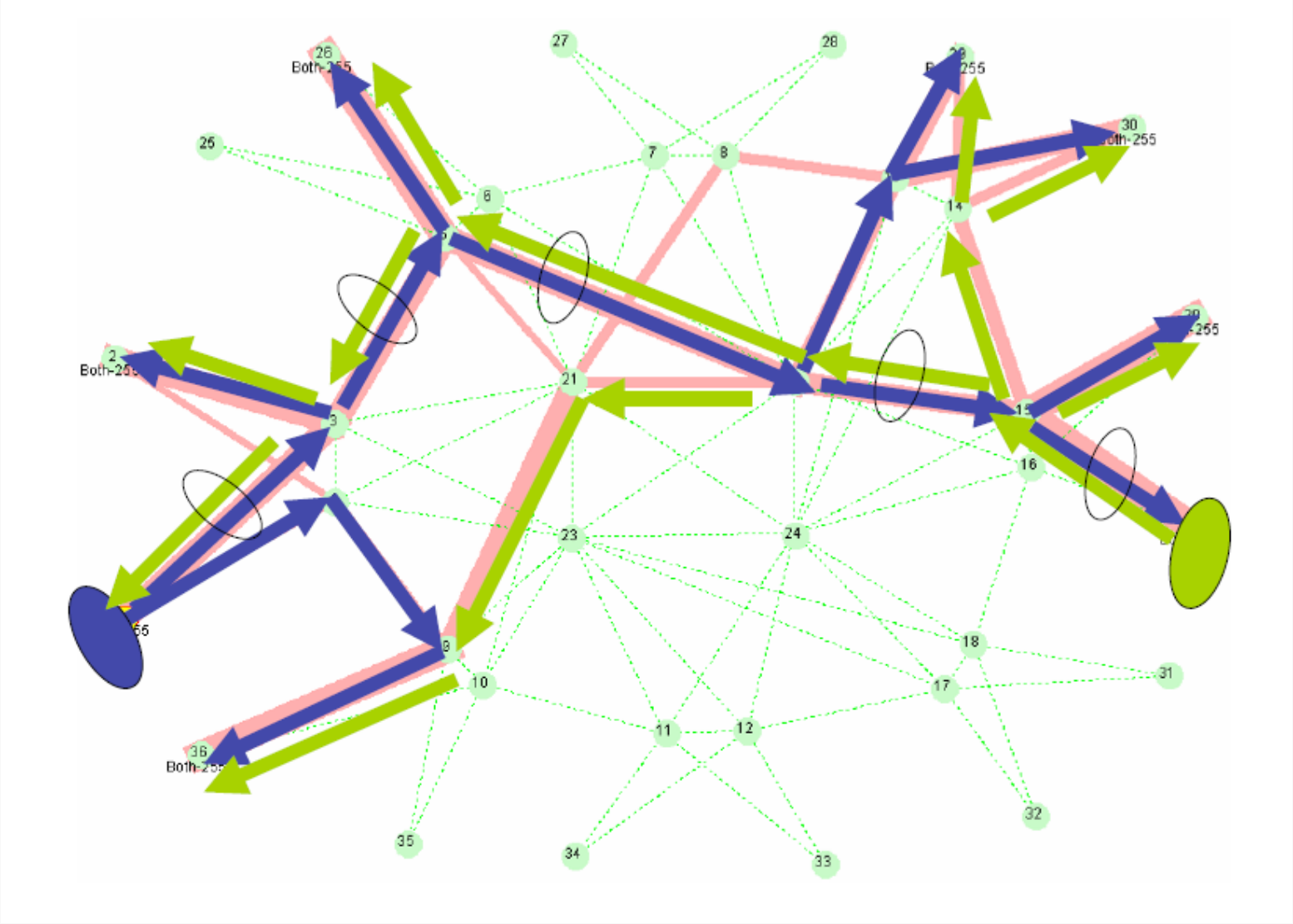
- STP helyettesítése új vezérlő síkkal
  - link state protokoll hirdeti a topológiát és a logikai hálózat tagságot
    - Intermediate System to Intermediate System (IS-IS) kiegészítésekkel
    - közvetlenül L2 rétegben fut, nem kell IP cím, mint az OSPF-ben
    - automatikus link állapot felderítés
    - nincsenek blokkolt portok, linkek
  - equal cost *multiple* shortest paths kihasználása
  - minden forrás legrövidebb utakból álló fát számít
    - szimmetrikus oda-vissza utak
- PB ⇨ SPB Vlan ID (SPBV)
- PBB ⇨ SPB MAC (SPBM)
- gyártók: Avaya, Huawei, Alcatel-Lucent

STP vs. SPB


2- poor routes
1- Can't use these links
Source
A1.. A100
ROOT
Dest
3 – Must learn A1..A100


All links usable

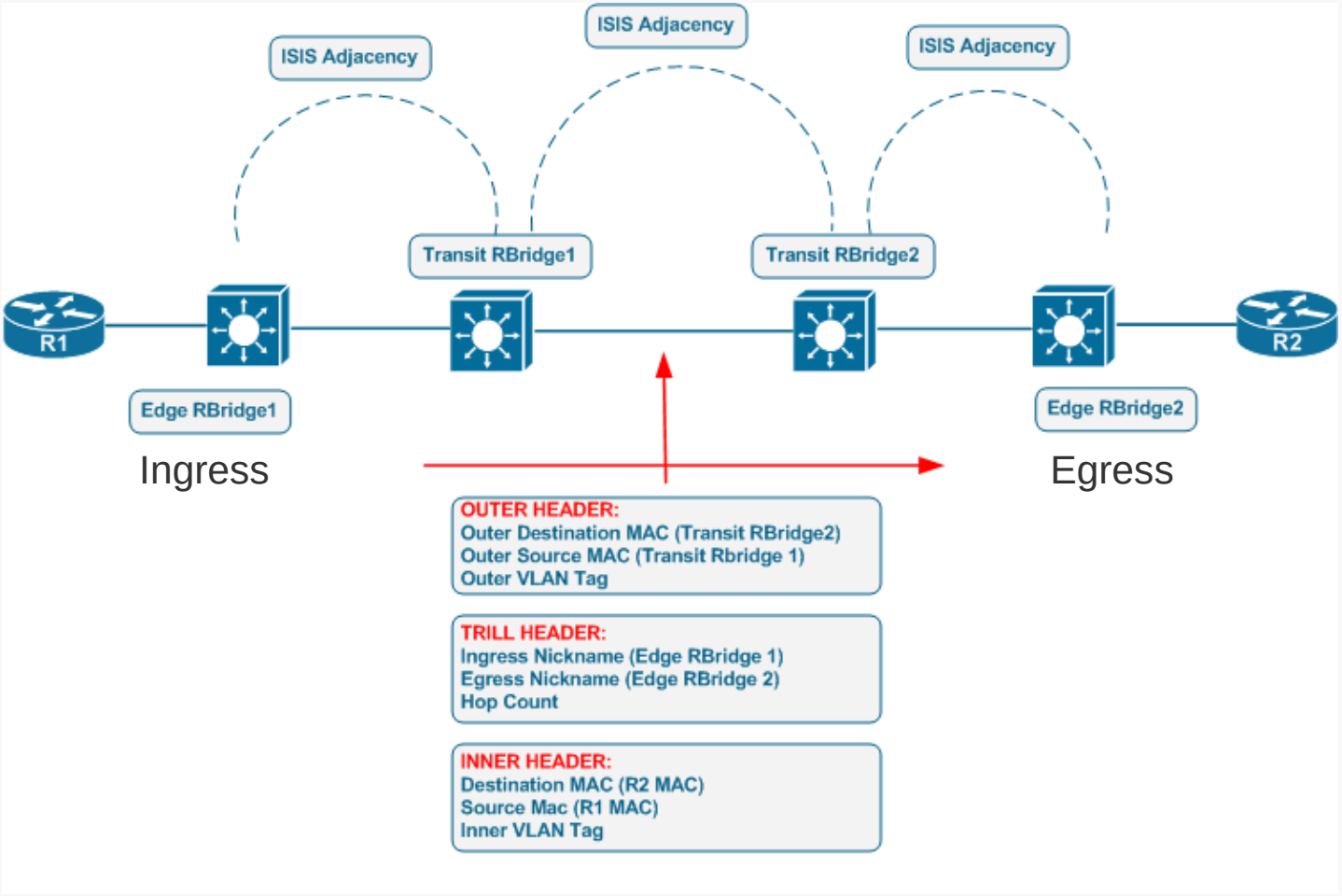
SPB



TRILL

- Transparent interconnection of lots of links
  - RBridge: routing bridge
    - *többutas* (ECMP) alagutak L2 tartomány felett
  - link state protokoll segítségével: IS-IS
    - TLV segítségével új adattípusok
  - plusz fejlécek
    - TRILL
      - hop count
      - RBridge ingress, egress nickname
    - külső Ethernet fejléc
      - RBridge source, destination MAC
      - VLAN tag
  - a tranzit/relay RBridge-ek a külső Ethernet fejlécet cserélik (swap) a next hop RBridge MAC címére
    - standard Ethernet kapcsolók a külső MAC cím alapján továbbítanak
- gyártók: Cisco, Brocade

TRILL


ISIS Adjacency
ISIS Adjacency
ISIS Adjacency
Transit RBridge1
Transit RBridge2
R1
R2
Edge RBridge1
Edge RBridge2
Ingress
Egress
OUTER HEADER:
Outer Destination MAC (Transit RBridge2)
Outer Source MAC (Transit Rbridge 1)
Outer VLAN Tag
TRILL HEADER:
Ingress Nickname (Edge RBridge 1)
Egress Nickname (Edge RBridge 2)
Hop Count
INNER HEADER:
Destination MAC (R2 MAC)
Source Mac (R1 MAC)
Inner VLAN Tag

SPB vs. TRILL

	SPB	TRILL
Szabványosító testület	IEEE	IETF
Adattovábbítás	Ethernet kapcsolás nincs MAC cím csere	RBridge nickname alapján MAC címek cseréje hop-by-hop
Virtuális hálózatok	SPBM: 16 millió	4096, opcionális fejléccel 16 millió
Megvalósítás	Meglévő, alacsony költségű Ethernet ASIC	Új hardver
Hurok kezelése	Reverse Path Forward Checking (RPFC)	RPFC + hop count
ECMP	Igen, 16 ág	Igen, 16 ág

Reverse Path Forward Checking

- a forrás cím ellenőrzése, hogy beérkezett interfészen visszairányba van-e bejegyzés az útvonalválasztási táblában, vagyis a legrövidebb úton érkezett-e
  - ha igen: továbbít
  - ha nem: eldob
- feltételek
  - útvonalválaszatási tábla korrekt és konvergált állapotban
  - szimmetrikus oda-vissza utak
- unicast és multicast

Források

- Shortest Path Bridging, IEEE 802.1aq, Tutorial and Demo, NANOG 50 Oct 2010, Peter Ashwood-Smith, Huawei
- Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. PortLand: a scalable fault-tolerant layer 2 data center network fabric. *SIGCOMM Comput. Commun. Rev.* 39, 4 (August 2009)
- Ankit Singla, Chi-Yao Hong, Lucian Popa, and P. Brighten Godfrey. 2012. Jellyfish: networking data centers randomly. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation* (NSDI'12). USENIX Association, Berkeley, CA, USA.