Adatközpontok belső hálózata

A táblára rajzolt otthoni hálózattól a Clos-topológiáig és a Fat-tree-ig —
miért nem elég a megszokott vállalati hálózati struktúra egy adatközpontban.

Honnan indulunk — és hova jutunk el

1. blokk — Egyetlen géptől a redundáns vállalati hálózatig: hogyan épül fel lépésről lépésre a router + switch + Core/Edge struktúra

2. blokk — Miért fullad ez adatközpontban zsákutcába: Észak-Dél vs. Kelet-Nyugat irányú forgalom

3. blokk — A Clos-hálózat elmélete: Spine, Leaf, blokkolásmentesség, fokozatok

4. blokk — Clos a gyakorlatban: BGP, ECMP, VXLAN, méretezés, és a Fat-tree

A cél, hogy ugyanazt az utat járjuk végig gondolatban, amit egy hálózat tervezője is bejárna — a legegyszerűbb megoldástól addig a pontig, ahol az egyszerű megoldás elfogy.

A legegyszerűbb eset

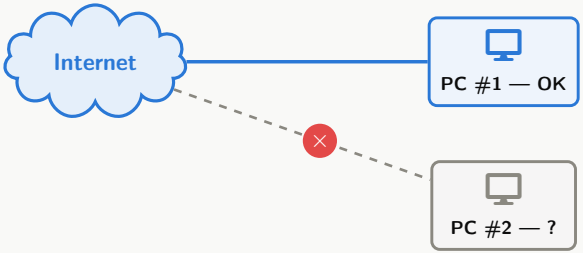
Kezdjük a legelején: az internetszolgáltató kábele befut a lakásba, és a másik végén ott a gépünk — semmi más nincs közte.



Ez tökéletesen működik — a gépünk közvetlenül, publikus IP-vel látszik az interneten. De vajon skálázódik ez, ha nem egy, hanem több gépünk van?

Az első probléma: csak egyetlen eszköz

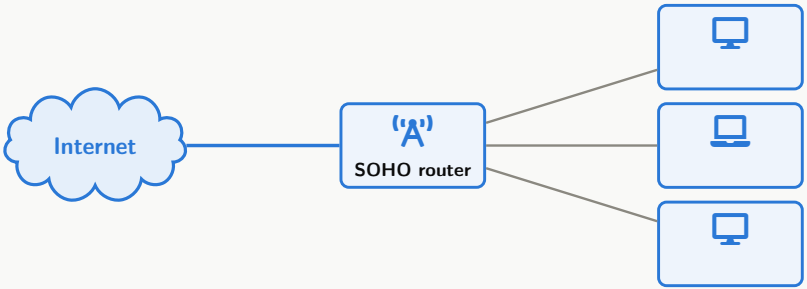
A bejövő vonal (pl. egy modem) fizikailag egyetlen eszközt tud kiszolgálni. Mi történik, ha egy második gépet is csatlakoztatnánk ugyanoda?



⚠ Kell valami, ami a bejövő egyetlen vonalat szét tudja osztani több eszköz felé — és ami mögé el tudjuk rejteni a saját, belső hálózatunkat.

A megoldás: egy doboz, ami mindent tud

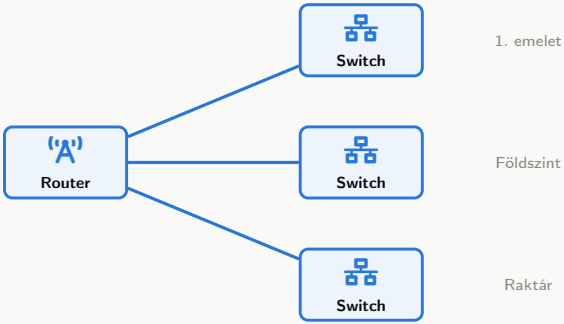
Az internetszolgáltatók örömünkre olyan eszközt adnak, amiben egyszerre van wifi *és* switch is (SOHO router).



Ezzel megszületik a **belső hálózat** fogalma: a router NAT-tal (Network Address Translation) egyetlen publikus IP mögé rejti a privát IP-ket, és tűzfalat, DHCP-t is ad hozzá.

Kisvállalati lépték: egy router már nem elég

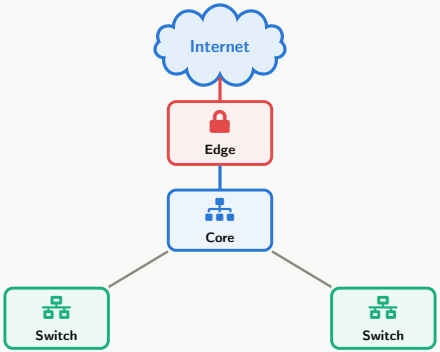
Egy iroda több helyiséget, emeletet szolgál ki — a router pár wifi-portja és pár kábele ehhez kevés. Külön **switch** kell minden szárnyhoz.



Ezzel megjelenik a **router + több switch** technológia — a router marad az egyetlen internetkijárat, a switchek pedig szétosztják a belső forgalmat.

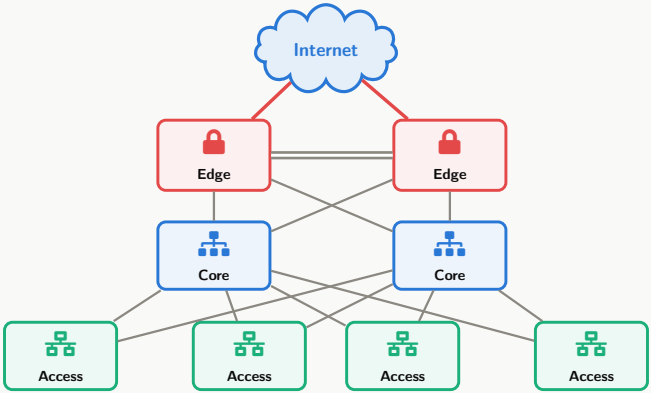
A belső és a külső forgalom szétválasztása

Ha komolyan vesszük a biztonságot és a teljesítményt, érdemes külön eszközre bízni az internet felé menő forgalmat és a belső hálózat forgalmát.



Edge: internet felé néz, támadások (DoS/DDoS) kezelése · **Core**: a belső hálózati forgalom irányítása

Redundáns Edge és Core: nincs egypontos hiba



A párba állított Edge és Core minden eszköze **mindkét** szomszédos réteg eszközéhez kapcsolódik — innen az „X” mintázat.

Ahol ez elakad

Ez a struktúra kiváló egy irodaházhoz — de van két rejtett feltételezése, ami adatközpontban nem igaz:

Észak-Dél irányra optimalizált

Azt feltételezi, hogy a forgalom fő iránya kliens ↔ internet — a szerver és a tárhely (storage) között csak egyetlen switch van.

Drágán bővül széltében

Ha új szerver-szekrény kell, a Core-nak és a Distribution rétegnek is növekednie kell — ez az irány nincs olcsón, korlátlanul skálázva megtervezve.

⚠ Egy adatközpontban a tárhely **bárhol lehet** — akár több szekrénnyel arrébb —, és a szerverek zöme **egymással** (Kelet-Nyugat irányban) kommunikál, nem csak kifelé. Kell egy másik struktúra.

2–3. BLOKK

A Clos-hálózat

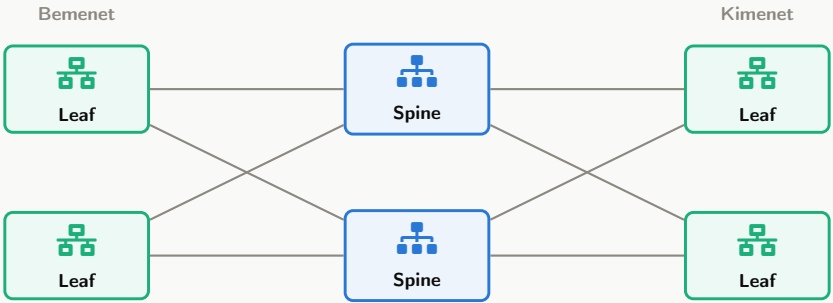
Egy 70 éves ötlet a telefonközpontokból, ami ma minden nagy adatközpont hálózatának gerincét adja.

Ki volt Charles Clos?

1953, *„A Study of Non-Blocking Switching Networks”*, Bell System Technical Journal — a cél az volt, hogy egy vezetékes telefonközpontban **bármely két végpont mindig tudjon egymással beszélni**, függetlenül attól, hányan telefonálnak éppen egyszerre.

A telefonközpontok problémája és egy mai adatközpont hálózatának problémája meglepően hasonló: mindkettő nagyszámú végpont közötti, kiszámítható, **blokkolásmentes** összeköttetést akar garantálni.

A Clos-modell alapfelépítése



A **Spine** (gerinc) szabadon maradt portjára *soha* nem köthetünk végponti eszközt — az kizárólag a **Leaf** (a fa levelei) szintre csatlakozhat.

A blokkolásmentesség szabálya

Egy Clos-hálózat **blokkolásmentes**, ha a Spine–Leaf kapcsolatok száma nagyobb vagy egyenlő, mint a Leaf-ekre kötött végponti eszközök kapcsolatainak száma.

Az előző ábrán egy **3 fokozatú** (3-stage) Clos-hálózat látható. Ez a szám tetszőlegesen növelhető: minden magasabb fokozat elérhető úgy, hogy a középső (Spine) réteget lecseréljük egy önálló, 3 fokozatú Clos-hálózatra.

A fokozatok száma mindig megegyezik a forrás és a cél közti **ugrások** (hop-ok) számával — a fenti példában ez pontosan 3: bemeneti Leaf → Spine → kimeneti Leaf.

3. BLOKK

A Clos-hálózat a gyakorlatban

Hogyan lesz egy 1953-as telefonközpont-elméletből a mai adatközpontok Spine–Leaf hálózata.

Miért nem elég a vállalati struktúra?

Vállalati hálózat (Észak-Dél)

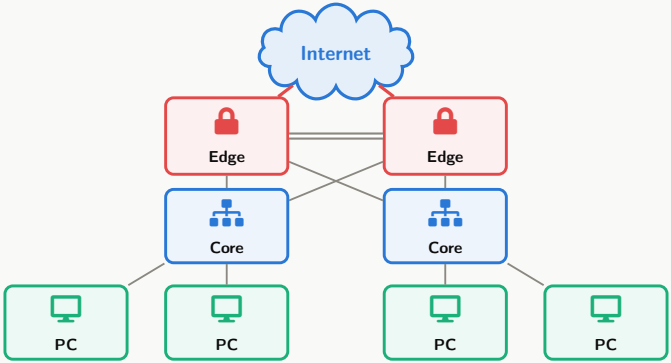
Kliens ↔ internet forgalomra tervezve. Szerver és tárhely között 1 switch. Bővítés (Core-nál) drága és a szélesség nem elsődleges szempont.

Adatközponti igény (Kelet-Nyugat)

A tárhely bárhol lehet a teremben. Szerver–szerver forgalom dominál. Olcsón, gyakorlatilag korlátlanul bővíthető *széltében*.

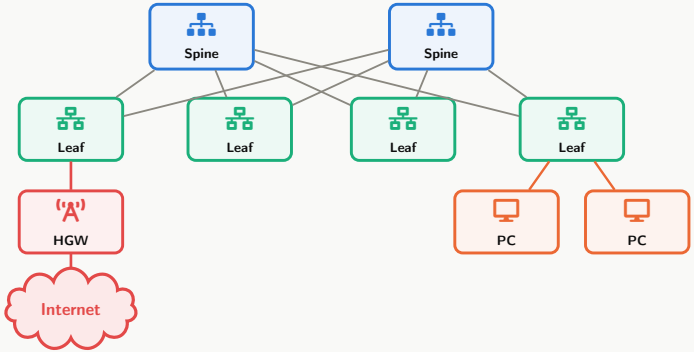
Erre a két elvárásra pontosan illik a Clos-modell — csak a felső fokozatát kell **tükrözni** a Spine rétegre.

Emlékeztető: a megszokott (Észak-Dél) hálózat



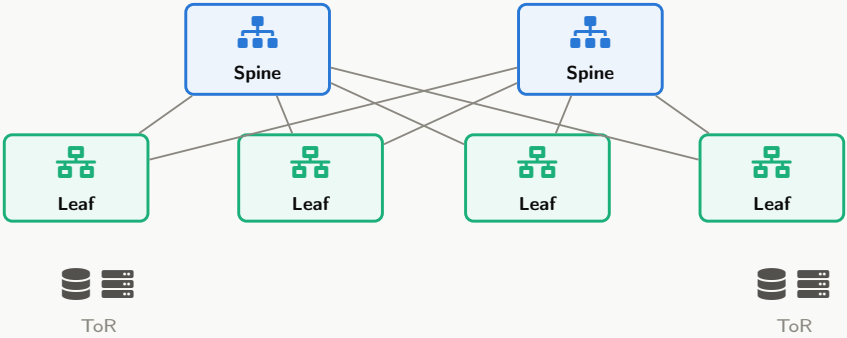
Ugyanaz a redundáns Edge/Core struktúra, amit az 1. blokkban felépítettünk — **klienseket (PC-ket)** szolgál ki. A következő dián ezt hasonlítjuk össze az adatközponti Spine–Leaf hálózattal, ami **szerver-rackeket** szolgál ki.

Ugyanez Spine–Leaf-en: PC és HGW



Az internetet adó router (HGW) **csak egy újabb Leaf-re kötött eszköz** — nincs külön, a Spine fölé emelt Edge réteg. A PC-k is egy Leaf-re csatlakoznak, mint egy szerver-rack.

A Clos-hálózat adatközponti verziója: Spine–Leaf



Spine: L3 switch, jellemzően 100 Gbps · **Leaf**: L3 switch, 40 Gbps, gyakran egybeépítve a rack tetején lévő (ToR) switch-csel

Miért routolunk L3-on egészen a rackig?

Ha Spine és Leaf között L2-t (switchelést) használnánk, a hurkok elkerülése végett a Spanning Tree Protocol (STP) letiltaná a redundáns portok egy részét:



A blokkolt link **kihasználatlan** marad — pedig fizettünk érte. Emellett egy nagy L2 hálózaton sok kéretlen broadcast/flooding forgalom is kering. **Megoldás:** L3 routing + ECMP + BGP — így minden link egyszerre, aktívan használható.

Ami ezt életre kelti: BGP, ECMP, VXLAN

BGP

Point-to-point IP-kkel dolgozunk a Spine–Leaf–ToR hálózaton. A BGP amúgy is kell a publikus címek hirdetéséhez, így ezt használjuk a belső címekhez is.

ECMP

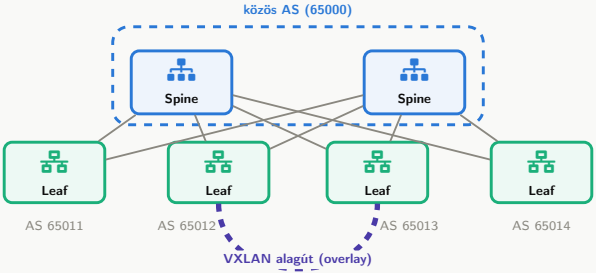
Equal-Cost Multipath — a BGP ezzel a Spine és Leaf közti *összes* kapcsolatot egyszerre kihasználja, kvázi összeadva a sávszélességüket.

VXLAN

A Leaf-ekre VXLAN-nal kapcsolódunk — így egy virtuális gép áttelepítése más rackbe nem jár az IP-tartomány mozgatásának gondjával.

⚠ Ára van: a BGP konfigurációs igénye az eszközök számával **exponenciálisan** nő — ezt önálló AS-ek kialakításával szokás mérsékelni.

AS-ek és VXLAN a gyakorlatban



AS-kiosztás: mindkét Spine **közös** AS-t kap (bekeretezve) — minden Leaf saját, **egyedi** AS-t. Minden Leaf–Spine kapcsolat külön eBGP session, a Spine-ok között nincs szükség iBGP-re. A **VXLAN alagút** két Leaf között, az útvonalválasztástól **függetlenül** létezik — az alatta futó BGP/ECMP semmit sem tud a benne utazó virtuális hálózatokról.

A teljes blokkolásmentesség ára

A gyakorlatban ritkán építenek teljesen blokkolásmentes hálózatot — több ezer porthoz ez irreálisan drága. Tipikusan a szükséges sávszélesség **harmadát** építik ki Spine és Leaf között.

Egy rackben 10 szerver van, fejenként 4×10 Gbps kapcsolattal a Leaf (ToR) felé $\Rightarrow$
rack-sávszélesség: $10 \times 4 \times 10 = 400$ Gbps.
Spine–Leaf célsávszélesség: $400/3 \approx 134$ Gbps.
100 Gbps-es linkekkel számolva ehhez **2 Spine** switch elegendő — kevesebb pedig már nem is lehetne, mert az megszüntetné a redundanciát.

Mivel a Spine–Leaf sávszélesség (100 Gbps) jellemzően nagyobb, mint a végpont–Leaf sávszélesség (40 Gbps), kevesebb Spine eszköz is elég — így több port marad szabadon a Leaf oldalon.

A hálózat bővítése

Új Leaf hozzáadása

Több **port** = több csatlakoztatható eszköz. Az új Leaf-et rá kell kötni *az összes* meglévő Spine switch-re.

Új Spine hozzáadása

Jelentősen nő a rendszer **sávszélessége** — minden Leaf egy-egy új linket kap a Spine réteg felé.

⚠ A bővítés felső korlátja a Leaf switchek **portszámának fele** — ha ennél több Spine-t telepítünk, garantáltan kihasználatlan linkek maradnak még maximális terhelés mellett is.

4. BLOKK

A Fat-tree topológia

Amikor a Spine réteg maga is egy Clos-hálózattá alakul — és a hálózat elméletileg korlátlanul bővül.

A Fat-tree: speciális Clos-hálózat

A Clos-hálózat Spine fokozata lecserélhető egy önálló, 3 fokozatú Clos-hálózatra — ezzel a hálózat fokszáma 3-ról 5-re nő.



Ez pontosan az a bővítési trükk, ami korlátlan portszám-megsokszorozást tesz lehetővé — a kapcsolás blokkolásmentes marad, mintha egyetlen, sokkal nagyobb switchet használnánk.

POD-ok és a rétegek átnevezése

N Spine switch-et **csoportokba** (plane-ekbe) rendezünk, és szerver-csoportokat — úgynevezett **POD**-okat — kötünk hozzájuk úgy, hogy minden POD minden switch-csoporttal kapcsolatban álljon.

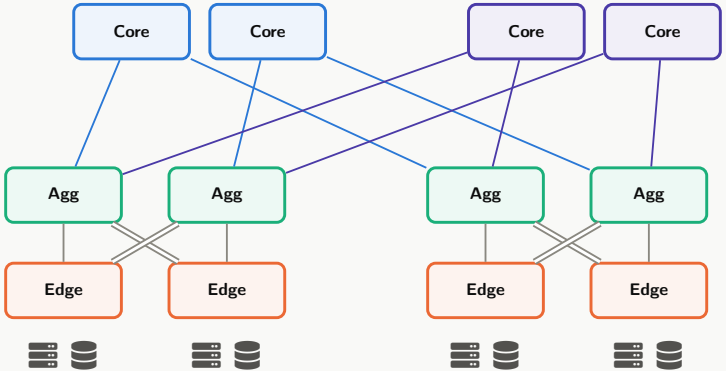
Spine → **Core** réteg

Leaf → **Aggregation** réteg

Leaf alá kötött új switchek → **Edge** réteg

Egy új **plane** telepítéséhez minden POD Aggregation (és Edge) rétegébe kell egy új switch — ezután az összes új switch csatlakozik az új plane eszközéhez.

Teljes Fat-tree topológia (2 POD, 2 plane)



A két POD **nem különálló rendszer**: minden Agg switch mindkét plane-hez (kék *és* lila) kapcsolódik — akár a másik POD felett átívelve is. Nagyobb telepítésben ugyanez minden POD-ra és plane-re igaz.

Összefüggések

L = rétegek száma, k = a használt switchek portszáma

Tulajdonság	Általános	3 réteg
Spine (Core) eszközök max. száma	$(k/2)^{L-1}$	$(k/2)^2$
Maximális végpontok száma	$2(k/2)^L$	$2(k/2)^3$
Összes felhasznált switch	$(2L-1)(k/2)^{L-1}$	$5(k/2)^2$
Edge rétegbeli switchek száma	$2(k/2)^{L-1}$	$2(k/2)^2$
Maximális POD-szám	$2(k/2)^{L-2}$	k

Ezekkel a képletekkel $F(k, L)$ jelöléssel szokás hivatkozni egy adott Fat-tree konfigurációra — pl. $F(16, 3)$ egy 16 portos switchekből épített, 3 rétegű hálózat.

Méretezési példa: F(16, 3)

Feladat: 12 szerver, szerver–Edge kapcsolat 10 Gbps, Leaf–Spine kapcsolat 25 Gbps, 16 portos switchek.

$F(16,3)$: Core switchek max.: $(16/2)^2 = 64$ · Végpontok max.: $2(16/2)^3 = 1024$ · POD-ok max.: 16

⚠ A teljes, blokkolásmentes kiépítéshez **320 switch** kellene (64 Core + 128 Aggregation + 128 Edge, az összefüggés-táblázat $5(k/2)^2$ képlete szerint) — ez a jelenlegi 12 szerveres terheléshez sem indokolt, sem megfizethető. El kell döntenünk, mekkora hányadát építjük ki ténylegesen a kapacitásnak.

Részleges (tudatosan oversubscribed) méretezés

Tegyük fel, hogy a teljes kapacitás **1/4-ét** garantáljuk — ez *nem* lesz blokkolásmentes hálózat, de a terheléssel együtt fejleszthető.

12 szerver $\times$ 10 Gbps $=$ 120 Gbps teljes igény.
Cél sávszélesség (1/4 kapacitás): $120/4 = 30$ Gbps.
Spine switchek/csoport: $30/25 \approx 1{,}2 \Rightarrow$ kerekítve **2 Spine** (a redundancia miatt is minimum ennyi kell) $\Rightarrow 2 \times 25 = 50$ Gbps.
2 Spine switch-hez 2 Aggregation és 2 Edge switch tartozik POD-onként.
A 64 Core switch 8 plane-be rendezhető — jelenleg csoportonként csak 2 eszköz szükséges.

Ha a switchek kapacitása nő (pl. 25 helyett 50 vagy 100 Gbps a gerincen), a konfiguráció egyszerűen megsokszorozható: $F(16,3) \to F(32,3) \to F(64,3)$.

ÖSSZEFOGLALÁS

Miért pont ez a struktúra?

- Egyetlen géptől **router + switch(ek)** → **redundáns Edge/Core** hálózatig jutottunk — ez remek egy irodának, Észak-Dél irányú forgalomra optimalizálva
- Adatközpontban a forgalom zöme **Kelet-Nyugat** irányú, és a tárhely bárhol lehet — ezért kell más struktúra
- A **Clos-hálózat** (1953, telefonközpontok) blokkolásmentes, kiszámítható útvonalhosszú kapcsolatot ad tetszőleges fokozatszámmal
- A gyakorlatban **Spine–Leaf** formában, L3 routing + BGP + ECMP + VXLAN kombinációjával valósul meg
- A **Fat-tree** ugyanennek egy POD-okba szervezett, gyakorlatilag korlátlanul bővíthető változata

KÖVETKEZŐ ÓRA: ERŐFORRÁS VIRTUALIZÁLÁS

Kérdések?

Egy táblára rajzolt otthoni hálózattól jutottunk el a modern adatközpontok Spine–Leaf és Fat-tree hálózatáig.