

Spanning Tree Protocol

Jákó András

jako.andras@eik.bme.hu

BME EISzK

Tartalom

Bevezető

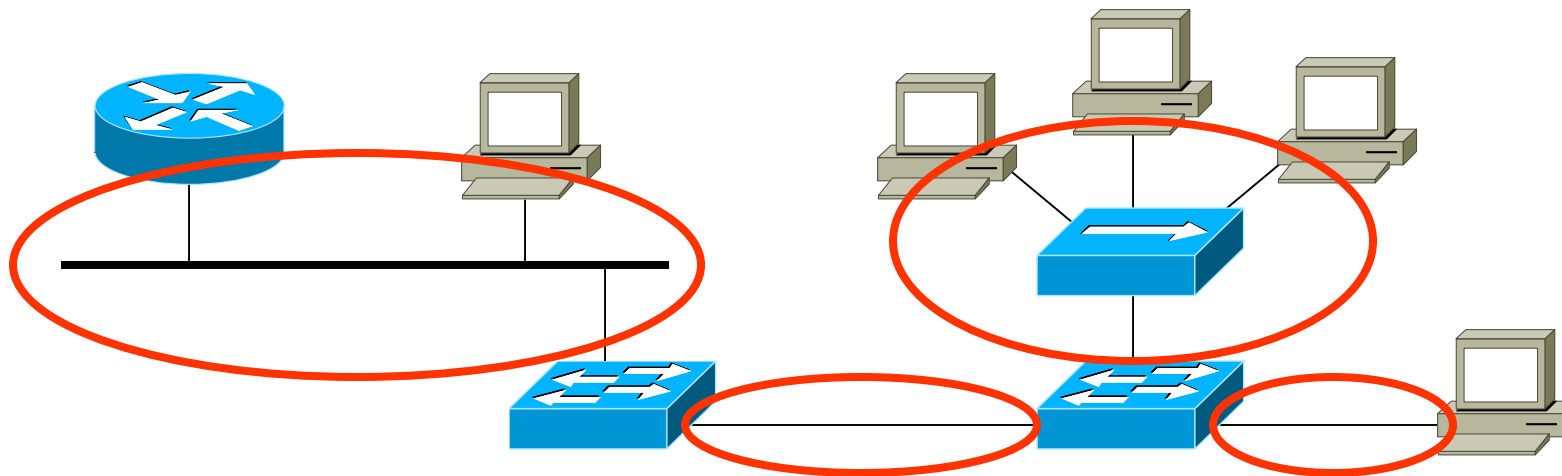
Spanning Tree Protocol

Rapid STP

Multiple STP

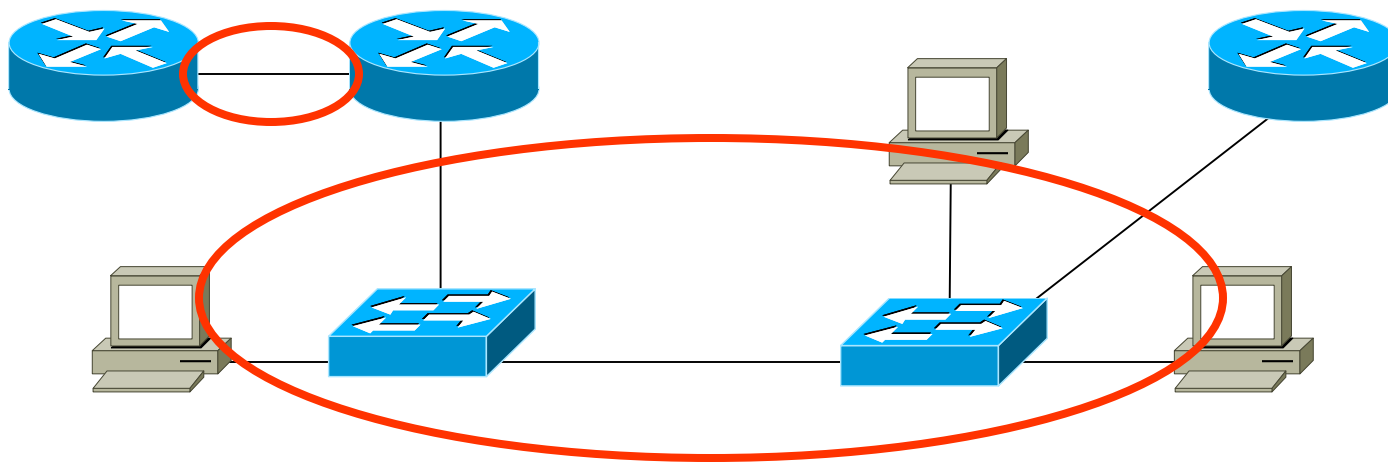
Collision domain

- az alábbi komponensekből álló **maximális összefüggő** része a hálózatnak:
 - kábel, csatlakozó (passzív alkatrészek)
 - transceiver, media converter
 - repeater, hub
- aki a collision domainben megszólal, azt mindenki hallja
- ha egyszerre beszélnek, akkor ütközés történik



Broadcast domain

- az alábbi komponensekből álló **maximális összefüggő** része a hálózatnak:
 - collision domain
 - bridge, switch
- MAC cím alapján lehet benne kereteket továbbítani
- a broadcast MAC címre (FF:FF:FF:FF:FF:FF) küldött kereteket mindenki megkapja a broadcast domainben

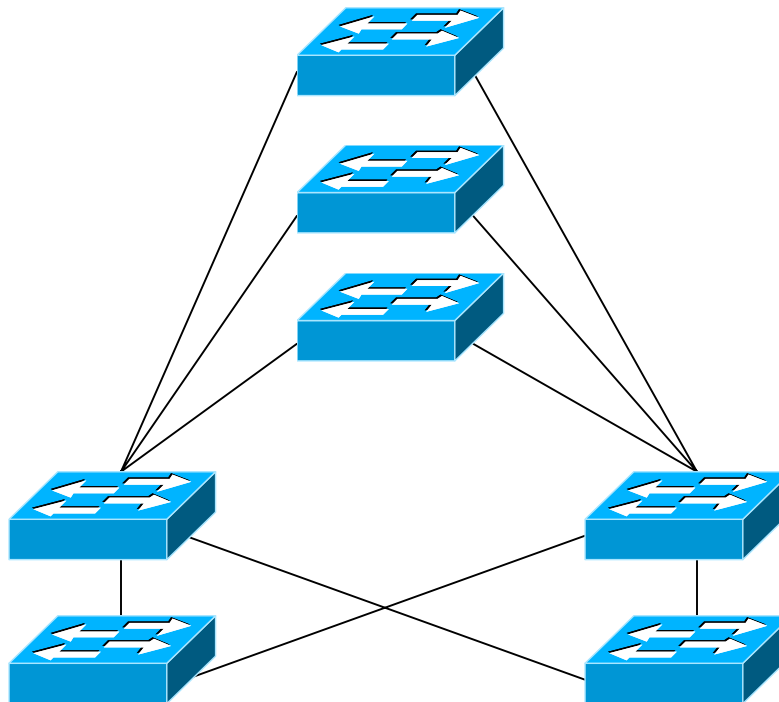


Keretek továbbítása

- minden switch/bridge nyilvántart egy **címtáblázatot**
 - melyik állomás (**cél MAC cím**) melyik **port** irányában van
- a beérkező kereteket a táblázat alapján továbbítja
 - ha a **cél MAC cím szerepel** a táblázatban, akkor a **táblázatban megadott portra** továbbítja a keretet
 - feltéve, hogy nem azon a porton érkezett a keret
 - ha a cél MAC cím **nem szerepel**, akkor **minden portra** továbbítja
 - kivéve a portot, amin érkezett
- a táblázat tartalma dinamikusan változik
 - a táblázat **kezdetben üres**
 - **tanulás**: a beérkező keretek **forrás címét** írja be a bridge a táblázatba a bejövő porttal
 - **felejtés**: a bejegyzések öregednek
 - ha régen nem jött keret az adott címről, akkor a bridge törli a bejegyzést

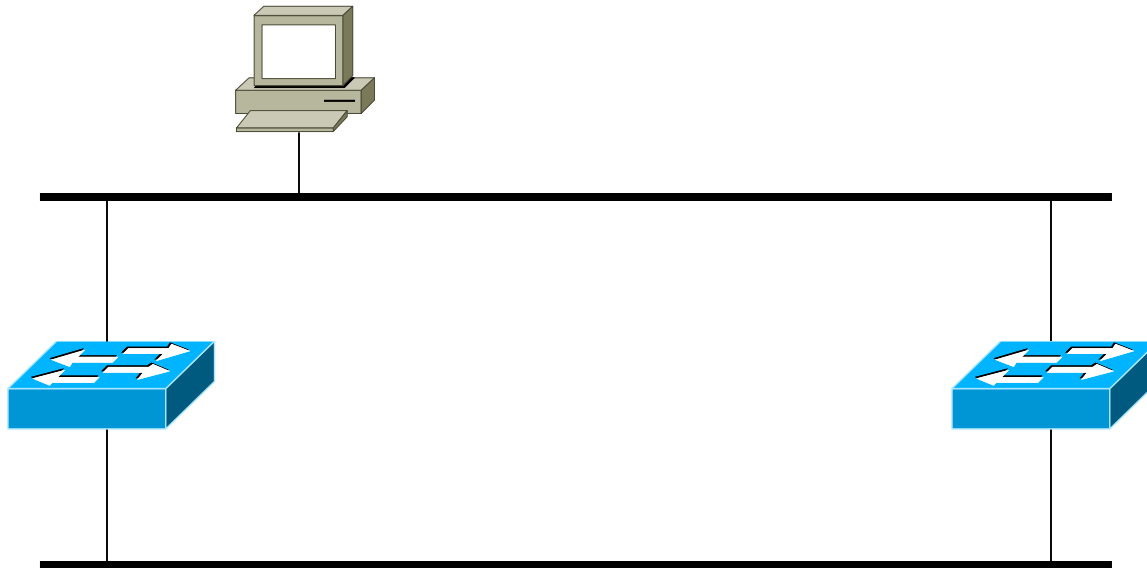
Körök, gyűrűk a hálózatban

- miért?
 - redundancia a **hibatűrés érdekében**
 - két pont között több út is van, tehát bizonyos meghibásodások esetén még mindig van összeköttetés
 - véletlenül



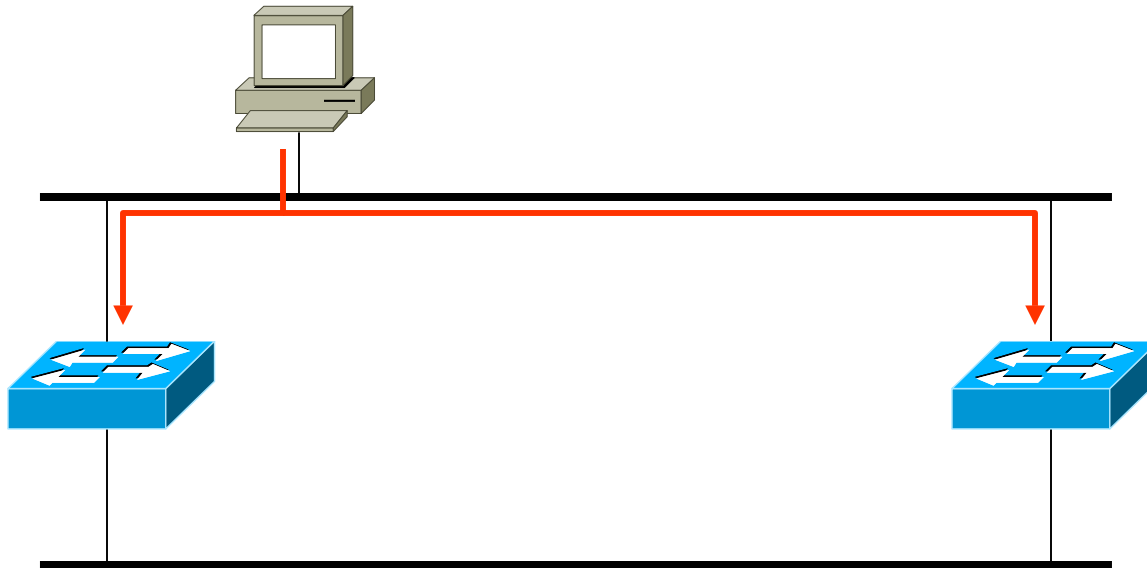
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



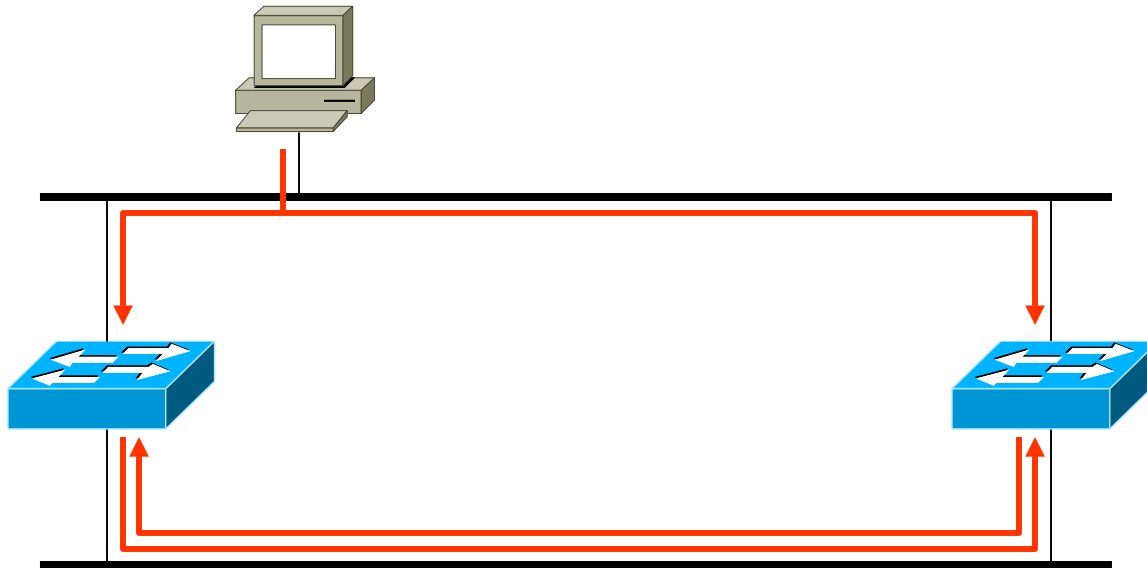
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



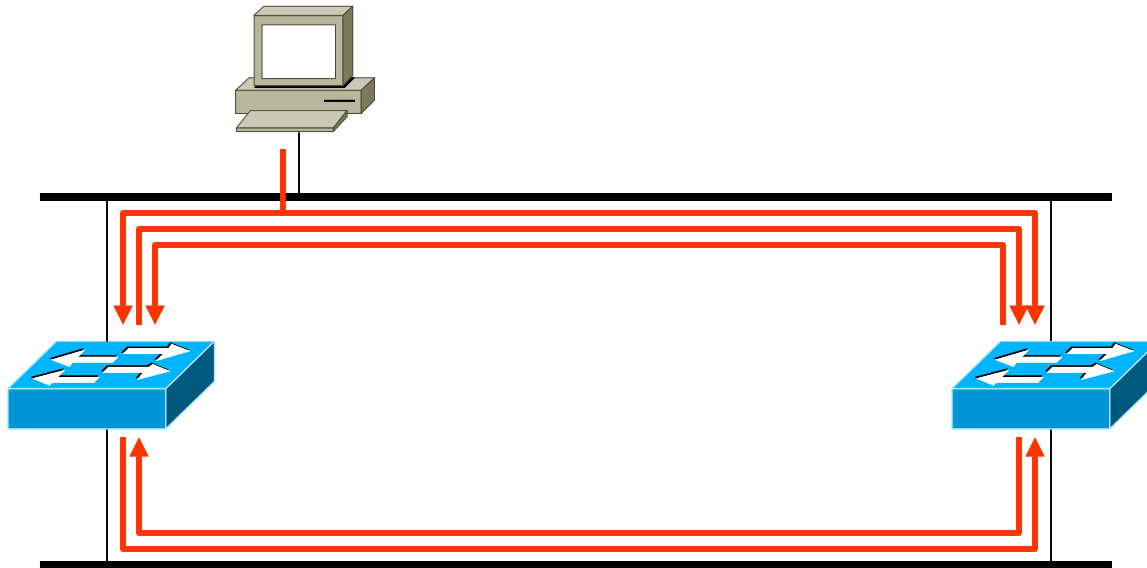
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



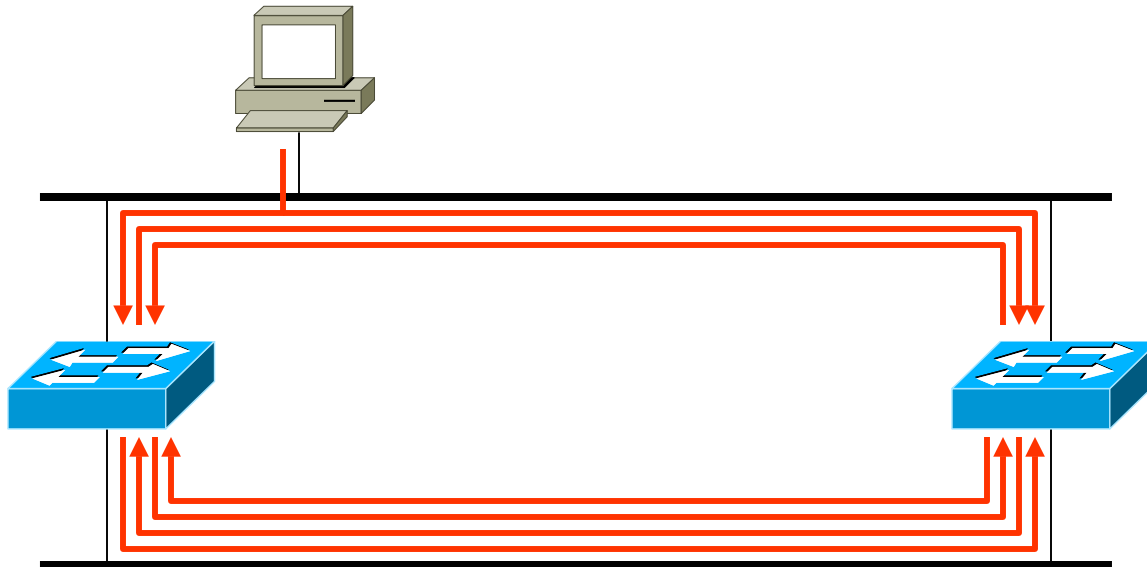
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



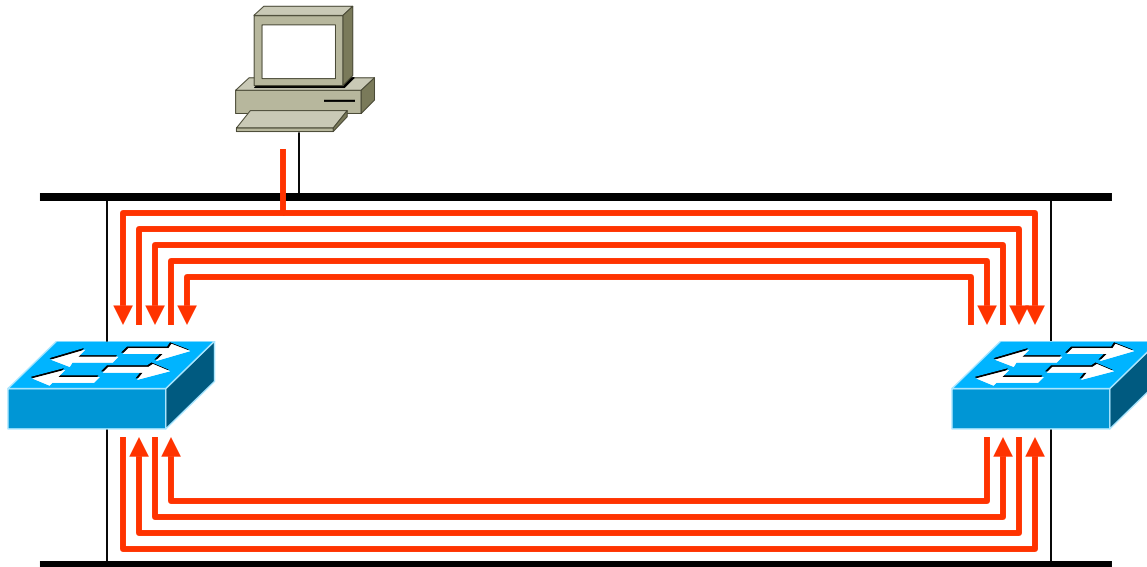
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



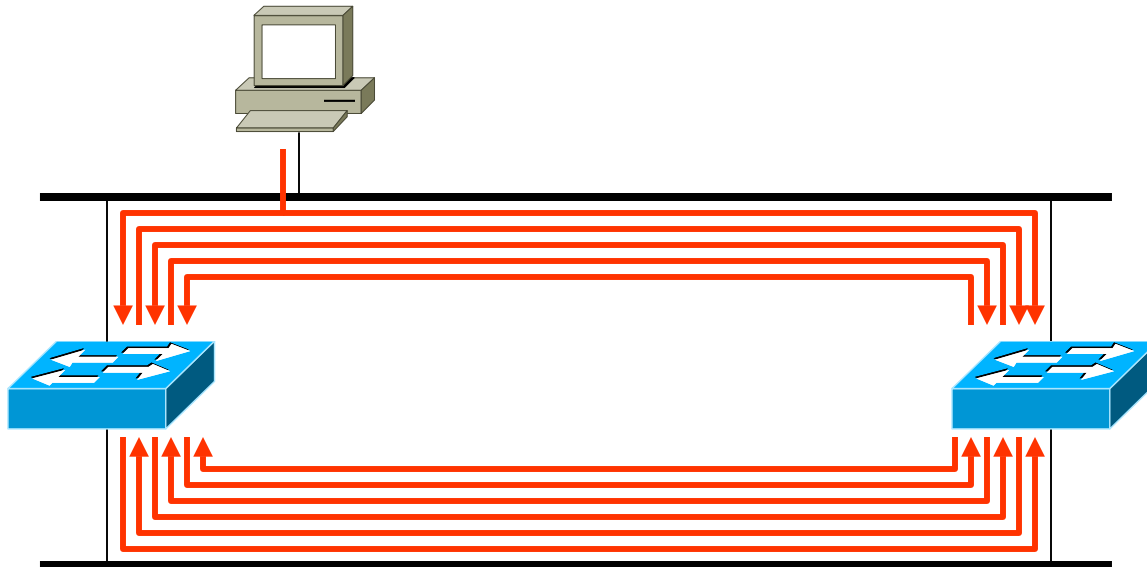
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



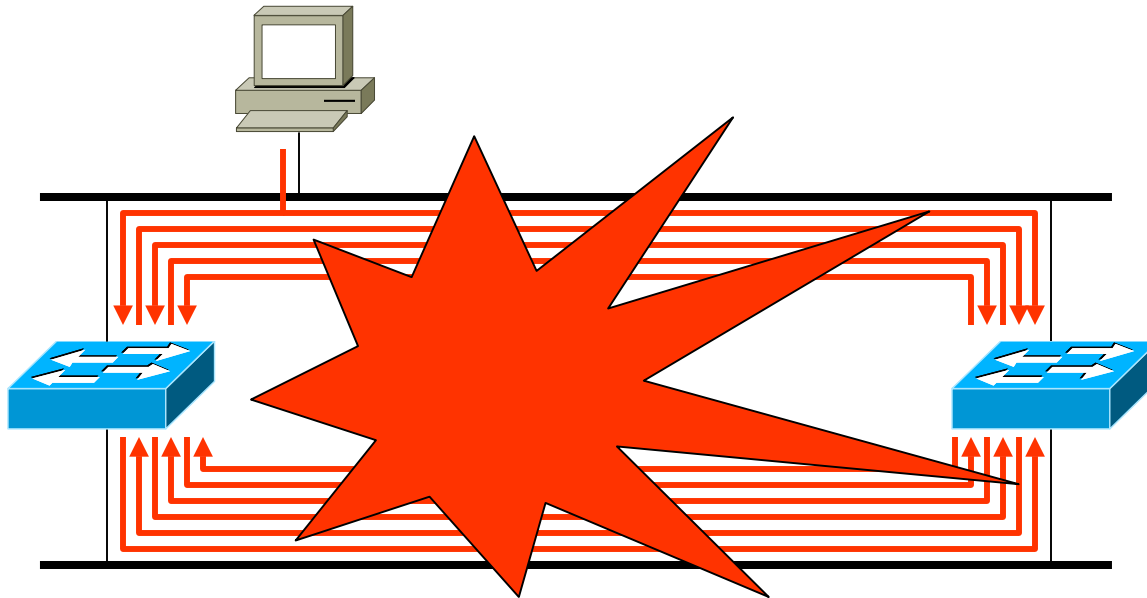
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



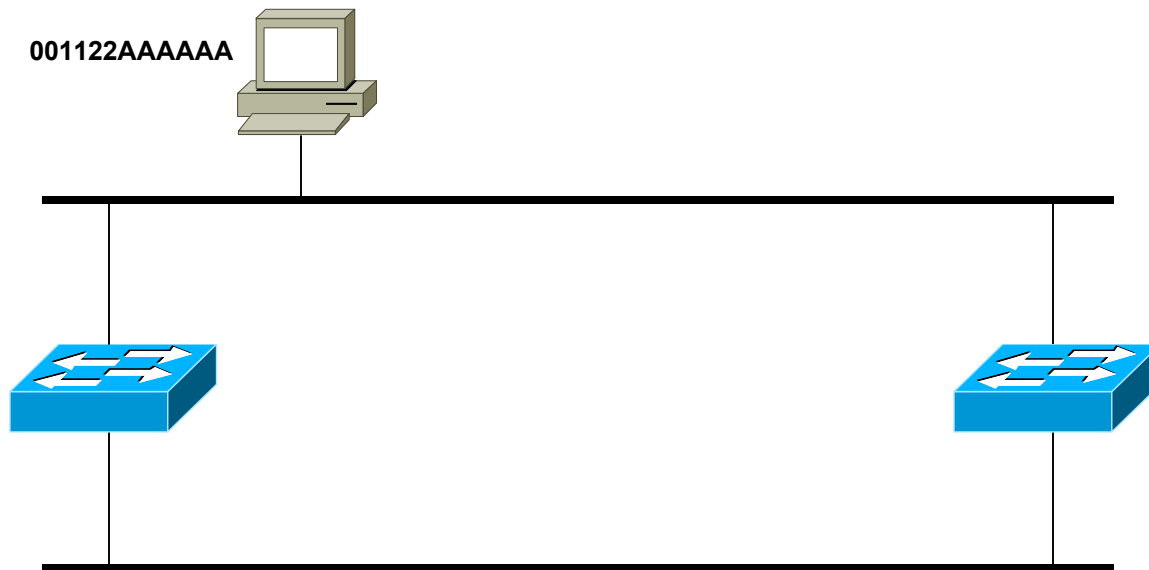
Broadcast storm

- minden broadcast vagy ismeretlen címre küldött keret
 - vég nélkül kering a hálózatban
 - több kör esetén többszöröződik is
- ez általában másodpercek alatt „leülteti” az egész switchelt hálózatot



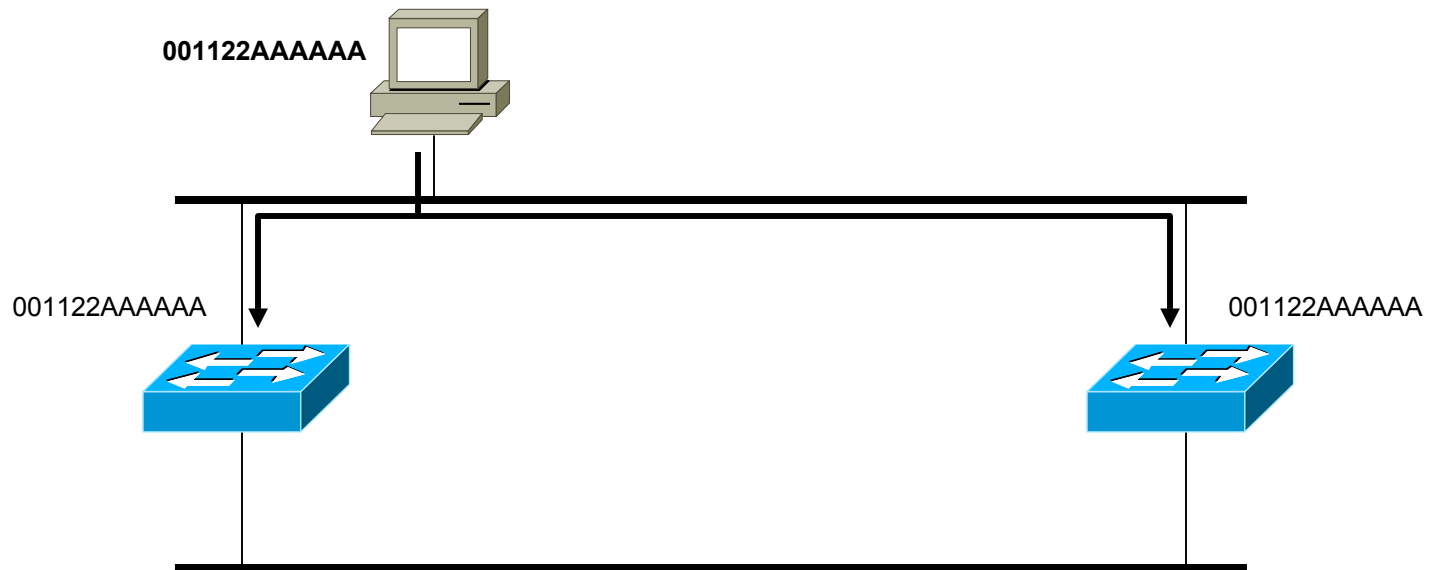
Címtáblázat korrupció

- a körben található switch címtáblázatában ide-oda ugrál a keret forrás MAC címe



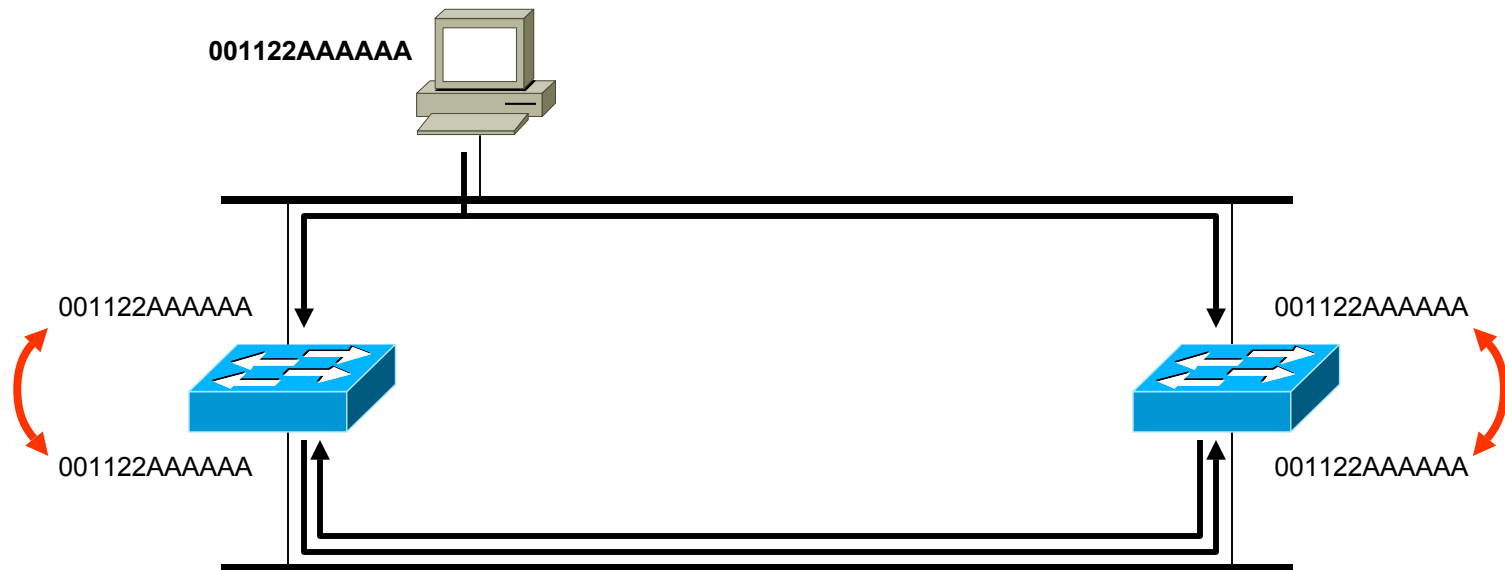
Címtáblázat korrupció

- a körben található switch címtáblázatában ide-oda ugrál a keret forrás MAC címe



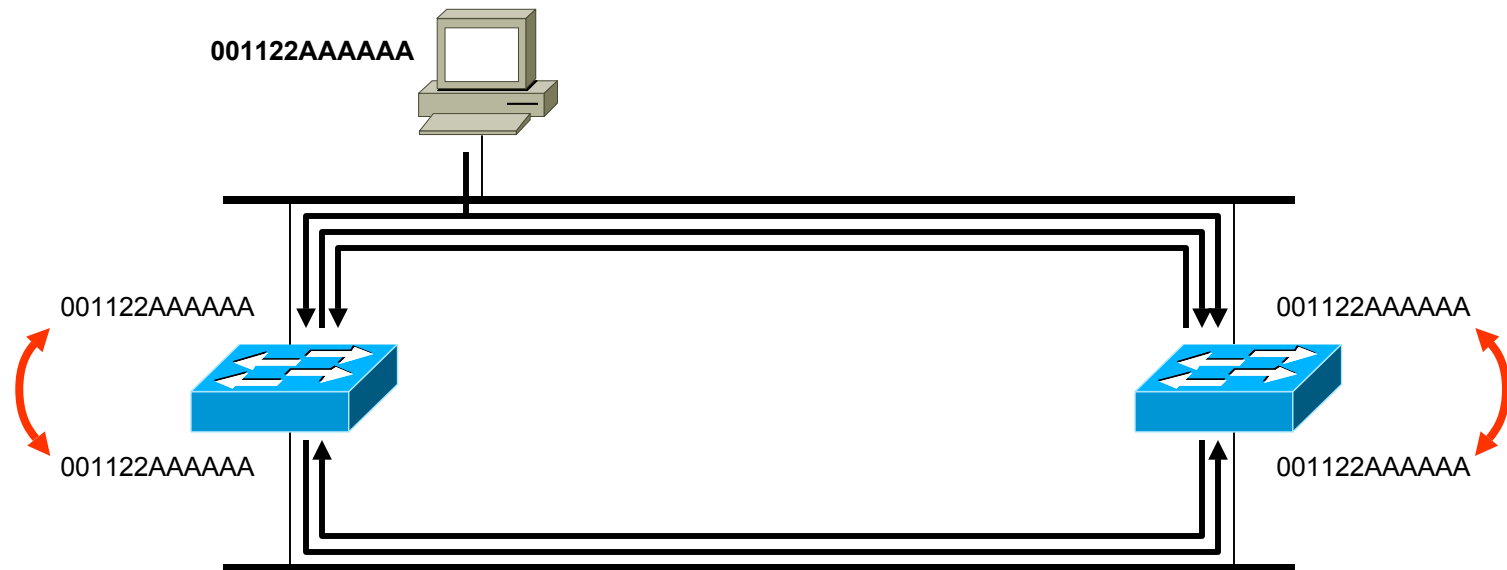
Címtáblázat korrupció

- a körben található switch címtáblázatában ide-oda ugrál a keret forrás MAC címe



Címtáblázat korrupció

- a körben található switch címtáblázatában ide-oda ugrál a keret forrás MAC címe



Spanning Tree Protocol

- 1983-ban az említett problémák elkerülésére találta fel [Radia Perlman](#) a Spanning Tree Protocolt
- az első széles körben elterjedt implementációt a DEC készítette
- a protokollt később az [IEEE szabványosította](#)
- az aktuális IEEE szabványok:
 - [Spanning Tree Algorithm and Protocol](#): 802.1D-1998, 802.1t-2001
 - [Rapid Spanning Tree Algorithm and Protocol](#): 802.1w-2001 (802.1D-2004?)
 - [Multiple Spanning Tree Protocol](#): 802.1s-2002, 802.1Q-2003



Kitekintés: Routing loop

- kör a L3 topológiában
- szintén probléma, mert az érintett csomagok nem jutnak el a célállomáshoz
- de nem keringenek a végtelenségig, mert van **TTL**
 - Time To Live mező minden csomag IP fejlécében
 - minden router legalább 1-el csökkenti a TTL-t továbbításakor
 - ha elfogyott, akkor eldobja
- így a hibás routing működés miatti hurok **általában nem rontja el a többi forgalmat**

Tartalom

Bevezető

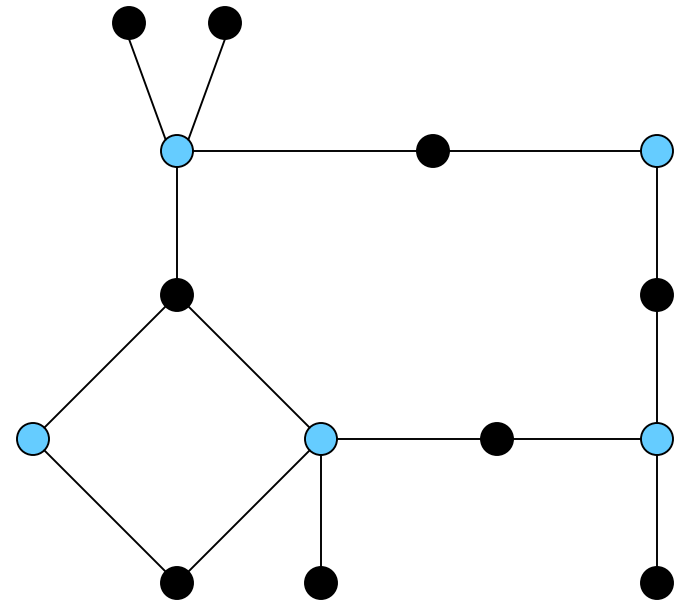
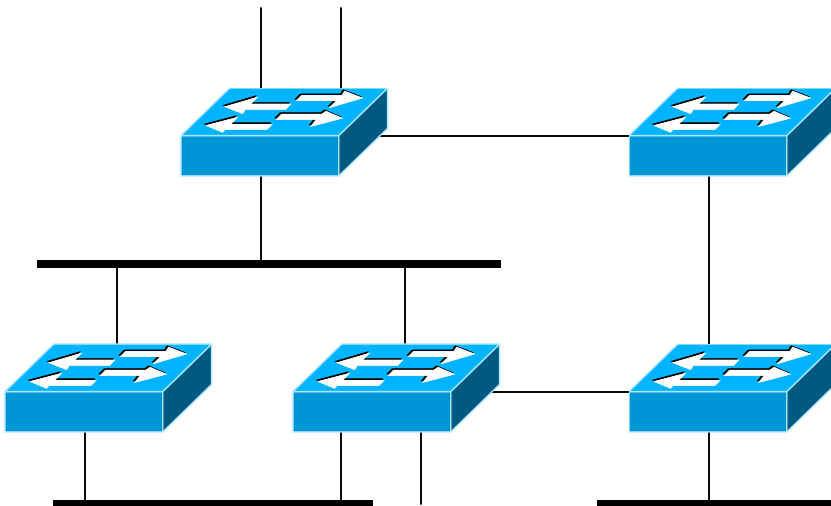
Spanning Tree Protocol

Rapid STP

Multiple STP

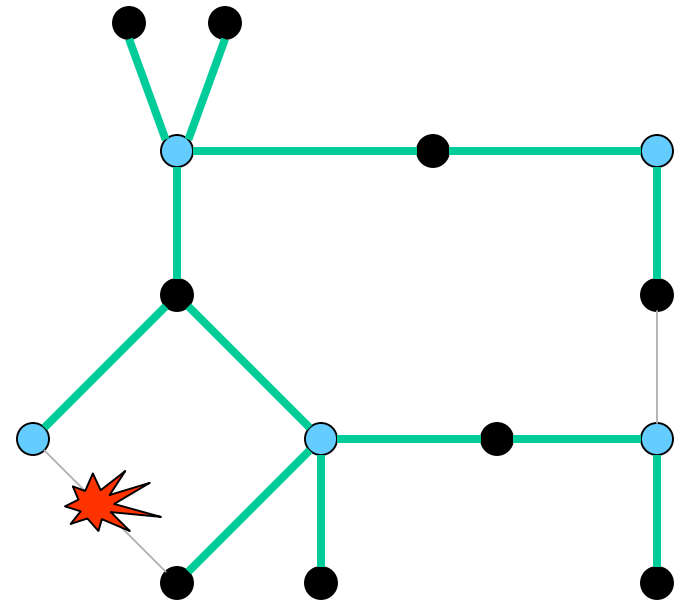
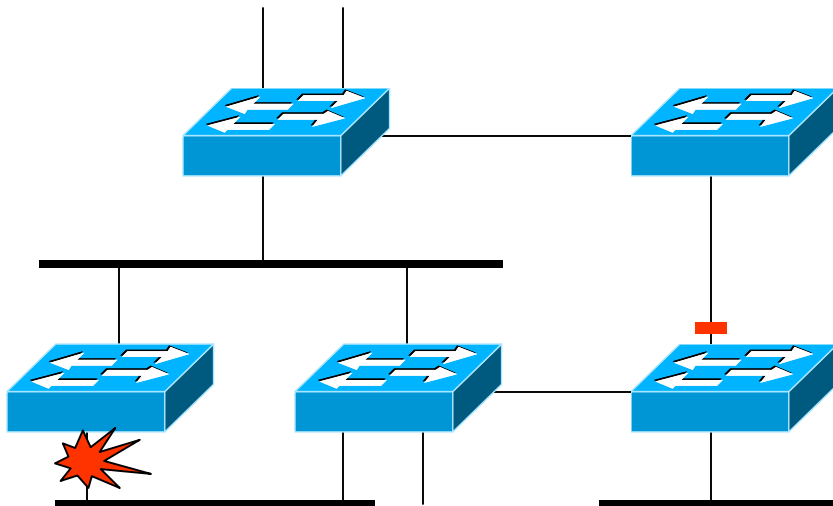
A hálózat mint gráf

- pontjai a **bridge**-ek és a **broadcast domaine**ek
- élei a **portok**
- a gráf páros, az alábbi ponthalmazokkal:
 - bridge-ek
 - collision domaine



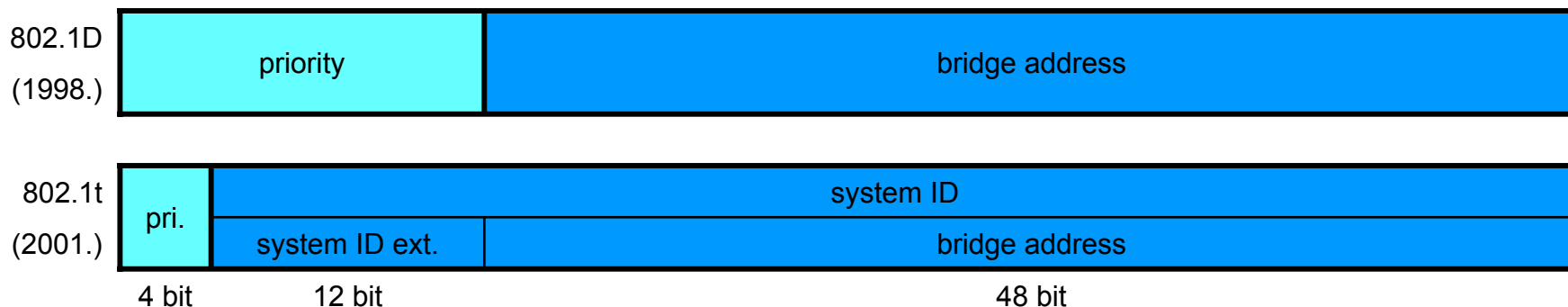
Spanning Tree Protocol

- **dinamikusan** működik: a fizikai topológia változásakor új logikai topológiát alakít ki, ha szükséges
 - új bridge vagy összeköttetés kerül a hálózatba
 - megszakad egy összeköttetés, ami a feszítőfa része volt
 - eltűnik egy bridge



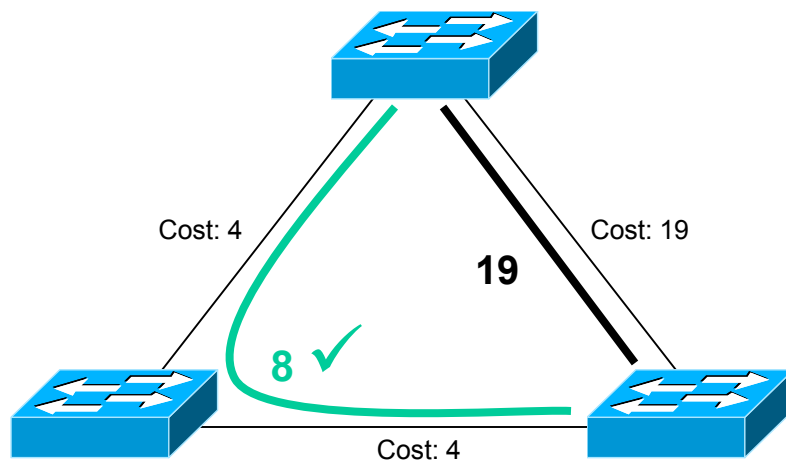
Bridge ID

- a bridge/switch **egyedi azonosítója**
- 64 bit
 - 16 vagy 4 bit **prioritás**
 - default: 32768 (0x8000)
 - konfigurálható
 - ha csak 4 bit, akkor is 0-65535 közötti érték, de 4096-tal oszthatónak kell lennie
 - 48 bit **bridge cím**
 - a switchet azonosító MAC address
 - valamelyik port címe vagy egy külön cím
 - system ID extension: MST példány száma (lásd később)
- a számszerűen **kisebb BID a jobb**



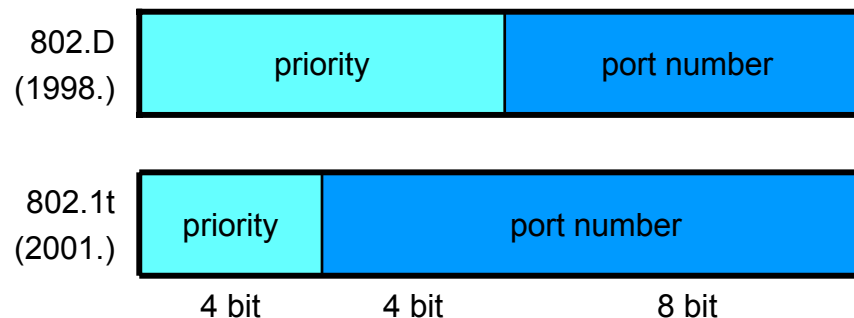
Path Cost

- két szomszédos bridge között definiálva van a távolság
 - az IEEE szabvány terminológiájában ez a „path cost”, kicsit megtévesztően
- út hossza = szomszédos bridge-ek távolságainak összege
- a rövidebb út a jobb



Port ID

- a **bridge-en belül** azonosítja a portokat
- 16 bit
 - 8 vagy **12 bit sorszám**
 - az IEEE 802.1D-1998 szabványban még 8 bit
 - az IEEE 802.1t-2001 ezt 12 bitre emeli
 - mert sok modern switchnek 256-nál több portja van
 - 8 vagy **4 bit prioritás**
 - mindét szabvány szerint 0-255 közötti érték
 - 802.1t szerint 16-tal osztható, hogy lényegében csak 4 bitet foglaljon
 - default: 128 (0x80)
- a **kisebb** Port ID a **jobb**



A feszítőfa kialakítása – 1.

1. a Root Bridge megválasztása

- a legjobb BID-vel rendelkező bridge
- bekapcsoláskor minden bridge rootnak gondolja magát
 - erről aztán a többiek hamar lebeszélnek, ha nem az övé a legjobb BID

A feszítőfa kialakítása – 2.

1. a Root Bridge megválasztása
2. minden bridge-en a Root Path Cost és a Root Port meghatározása
 - **Root Path Cost**: a **roothoz vezető** legrövidebb **út hossza**
 - a Root Bridge esetében 0
 - **Root Port**: biztosítja a **bridge kapcsolatát a fa gyökere felé**
 - ezen a porton elindulva van a legközelebb a Root Bridge
 - ha több ilyen van, akkor a **Port ID** dönt
 - a Root Porton hallja a bridge a legjobb Config BPDU-t

A feszítőfa kialakítása – 3.

1. a Root Bridge megválasztása
2. minden bridge-en a Root Path Cost és a Root Port meghatározása
3. collision domainenként a Designated Bridge és a Designated Port megválasztása
 - **Designated Bridge**: biztosítja az adott **collision domain** kapcsolatát a fa gyökere felé
 - akié a **legkisebb Root Path Cost** a collision domainhez kapcsolódó bridge-ek közül
 - ha több ilyen van, akkor akié a **legjobb BID**
 - **Designated Port**: ezen a porton kapcsolódik a Designated Bridge az adott collision domainhez
 - ha több ilyen van, akkor a **Port ID** dönt
 - a Designated Bridge küldi a collision domainbe a legjobb BPDU-t a Designated Porton

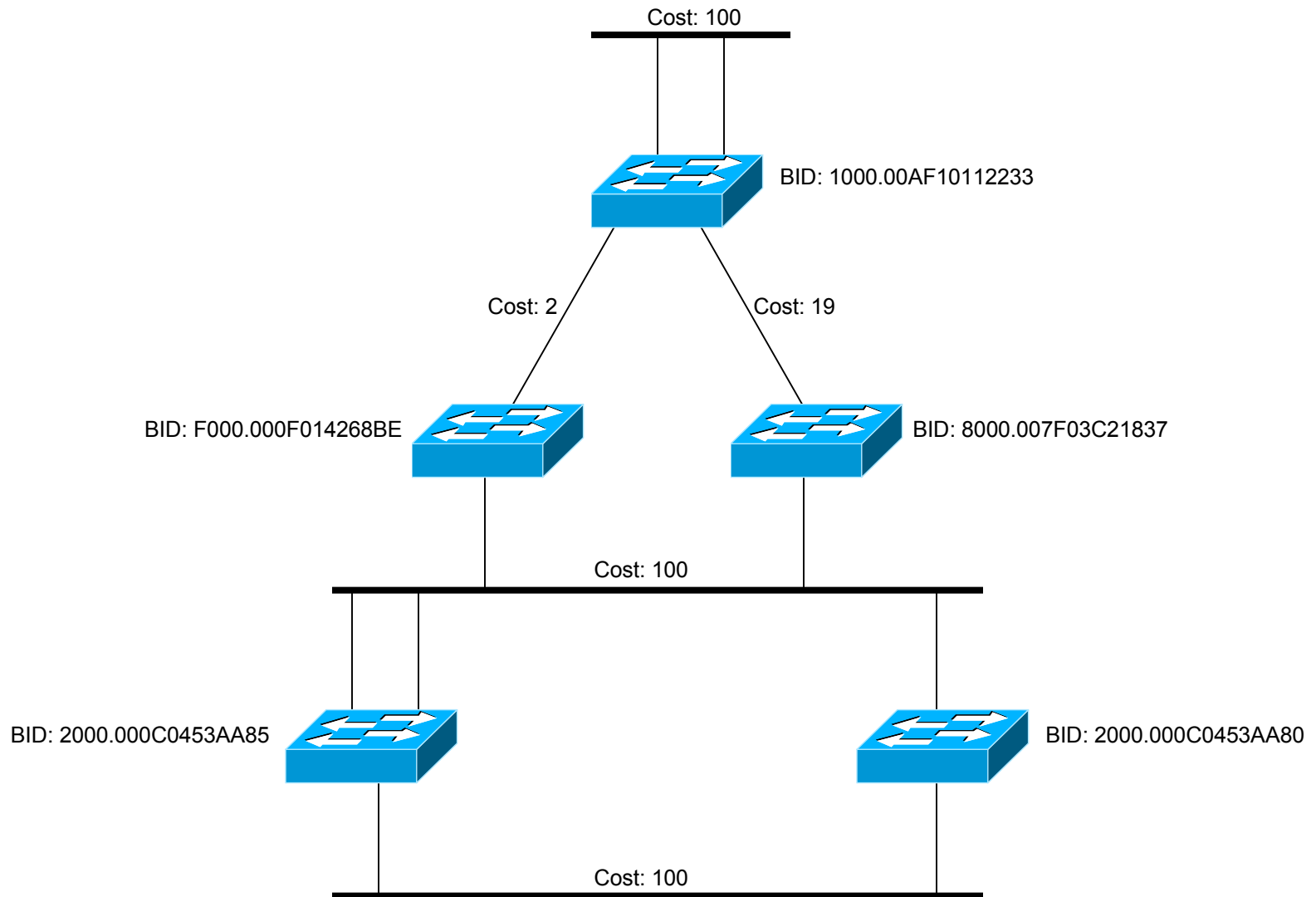
A feszítőfa kialakítása – 4.

1. a Root Bridge megválasztása
2. minden bridge-en a Root Path Cost és a Root Port meghatározása
3. collision domainenként a Designated Bridge és a Designated Port megválasztása
4. a **feszítőfában részt vevő portok** meghatározása
 - minden **Designated Port** és minden **Root Port** része a feszítőfának

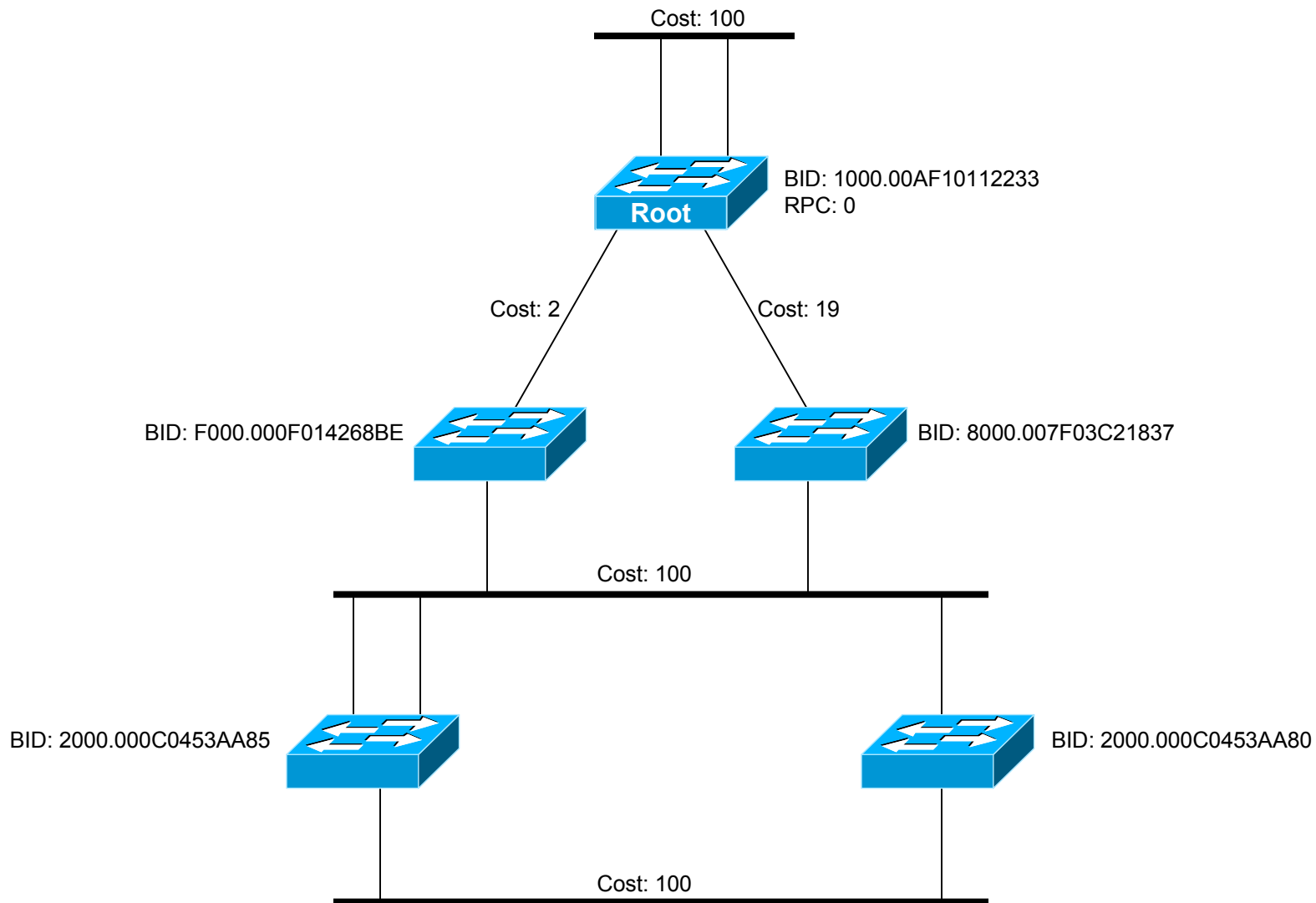
Port szerepek

- **Root Port:** a bridge kapcsolata a root felé
 - az a port, amelyiken át legközelebb van a Root Bridge
 - a Root Bridge-nek nincs
 - a többi bridge-nek 1 root portja van
- **Designated Port:** a collision domain kapcsolata a root felé
 - a Designated Bridge egyik portja
 - minden collision domainben 1 Designated Port van
 - a Root Bridge-nek általában minden portja Designated Port
 - más bridge-nek annyi Designated Portja van, ahány collision domainen ő a Designated Bridge
- **Alternate Port:** minden más működő port
- **Disabled Port:** nem működő port
 - ki van kapcsolva, vagy
 - MAC nem működik („nincs rajta link”), vagy
 - sikertelen 802.1X autentikáció

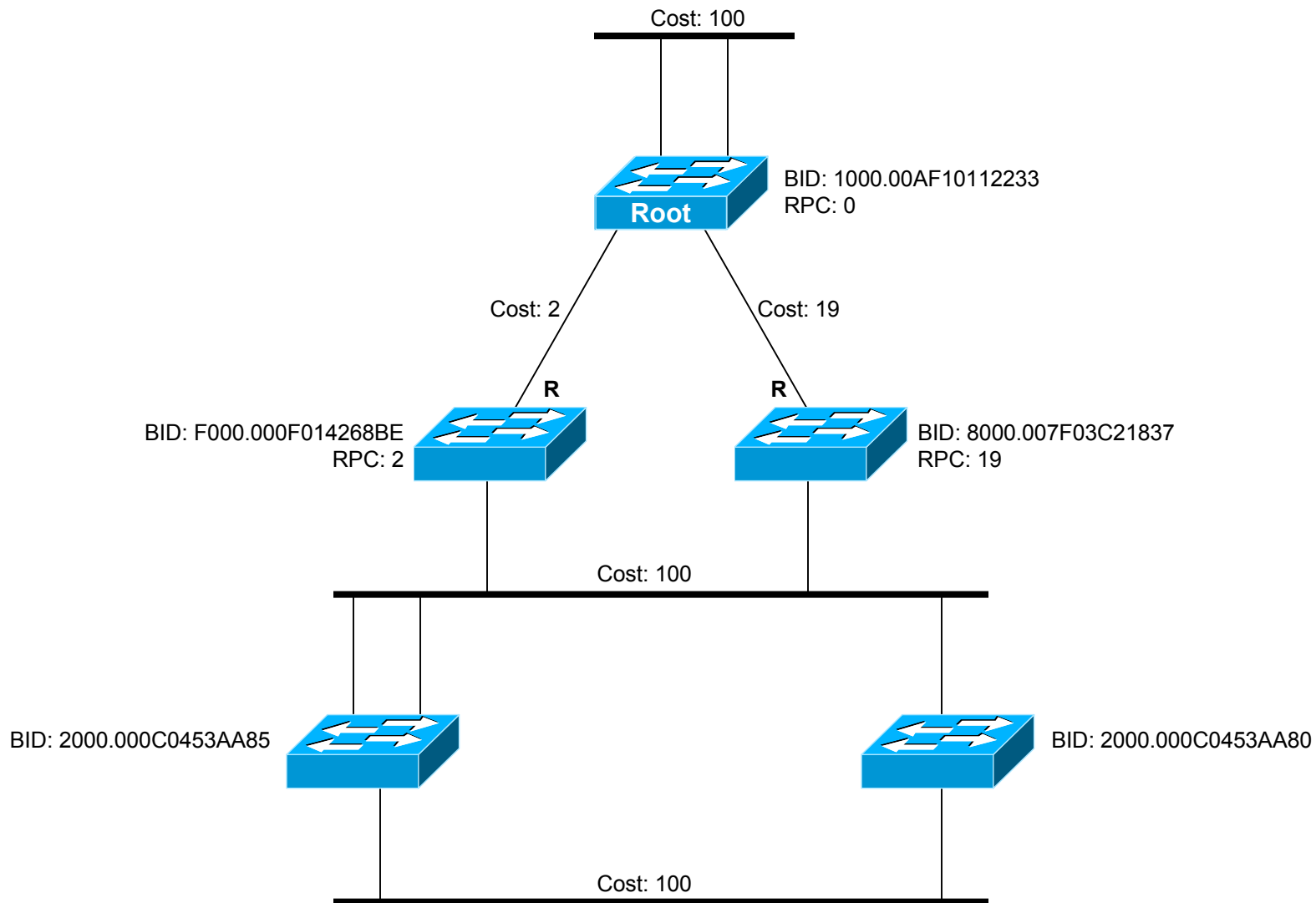
Példa: Feszítőfa



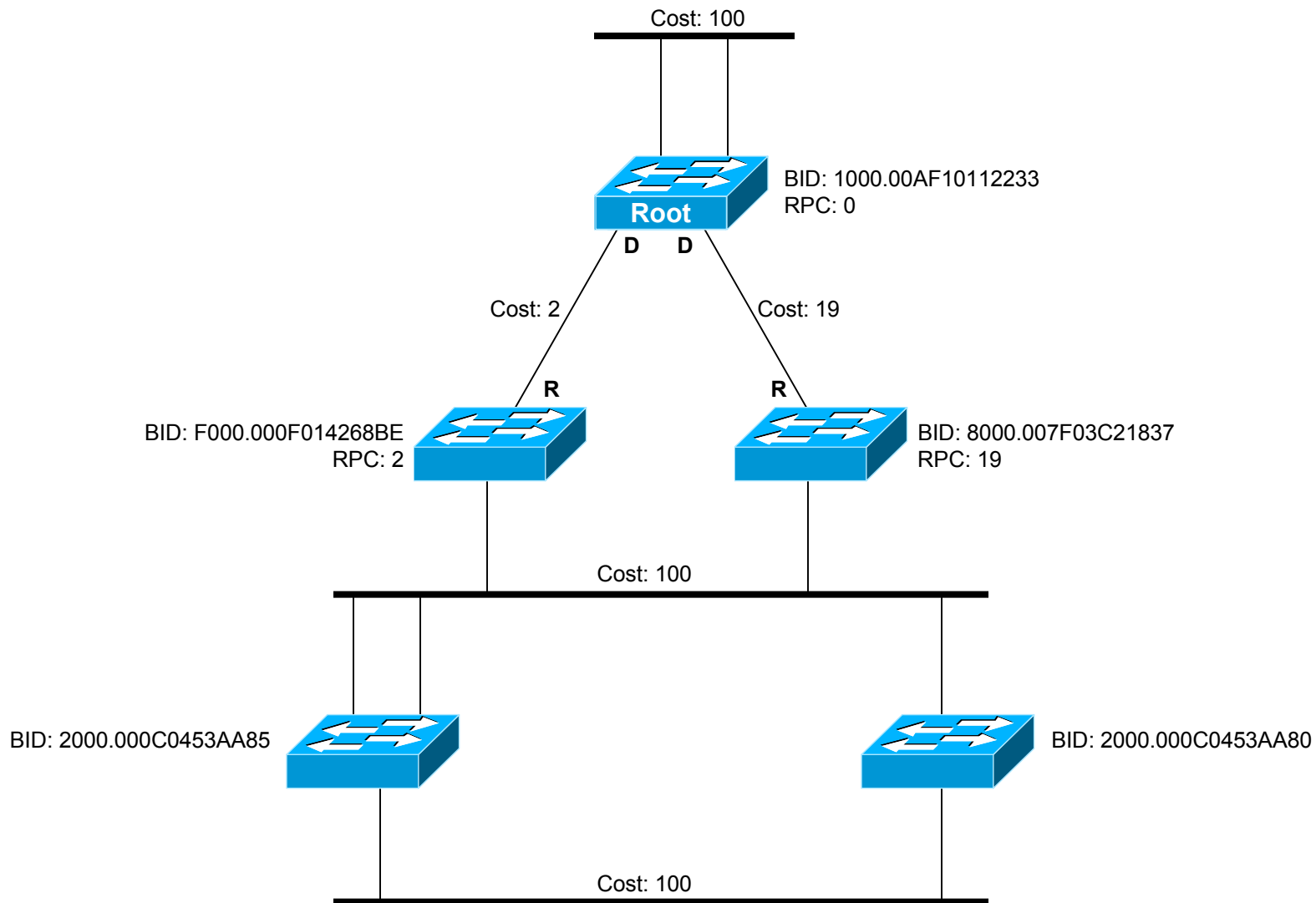
Példa: Feszítőfa



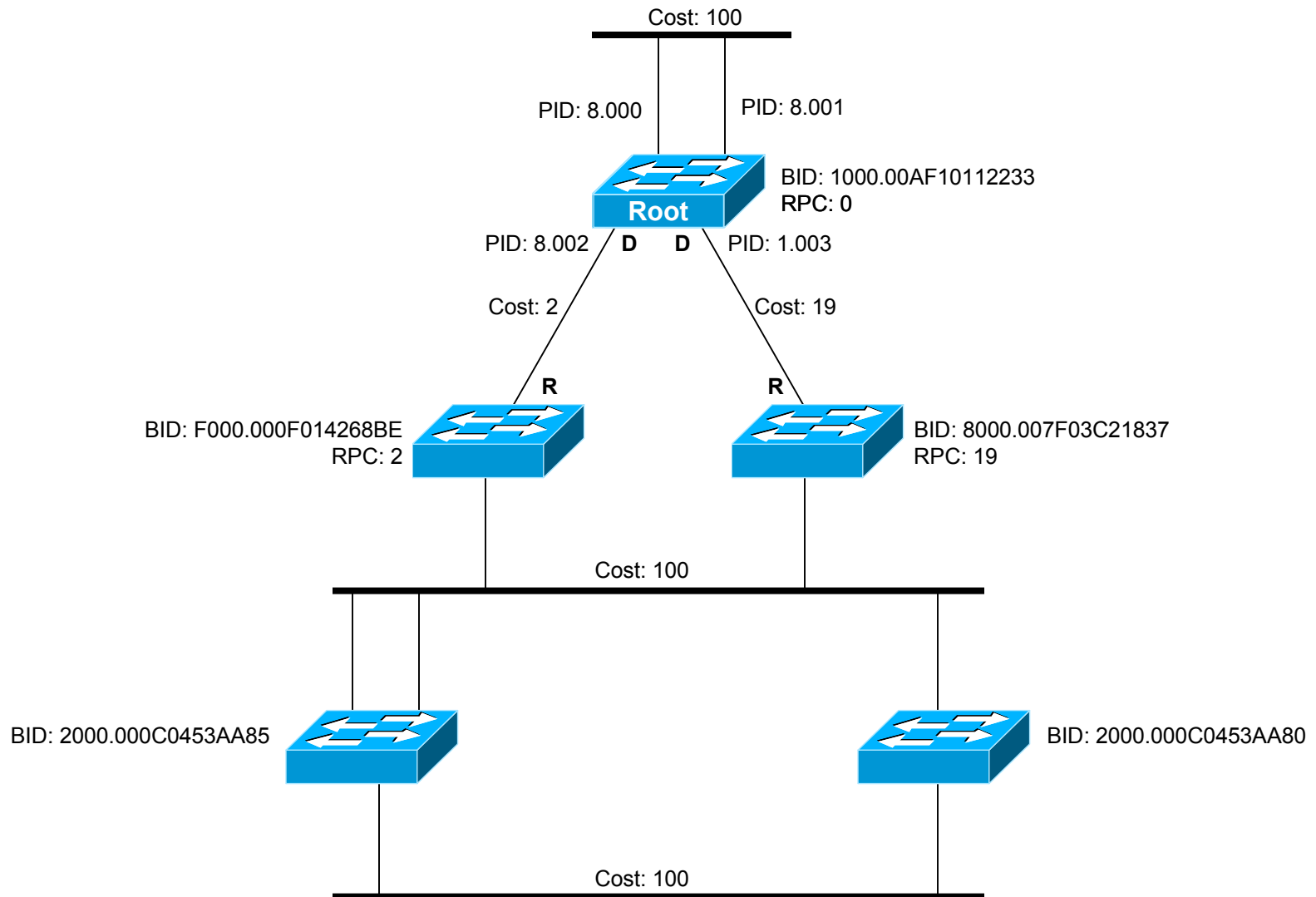
Példa: Feszítőfa



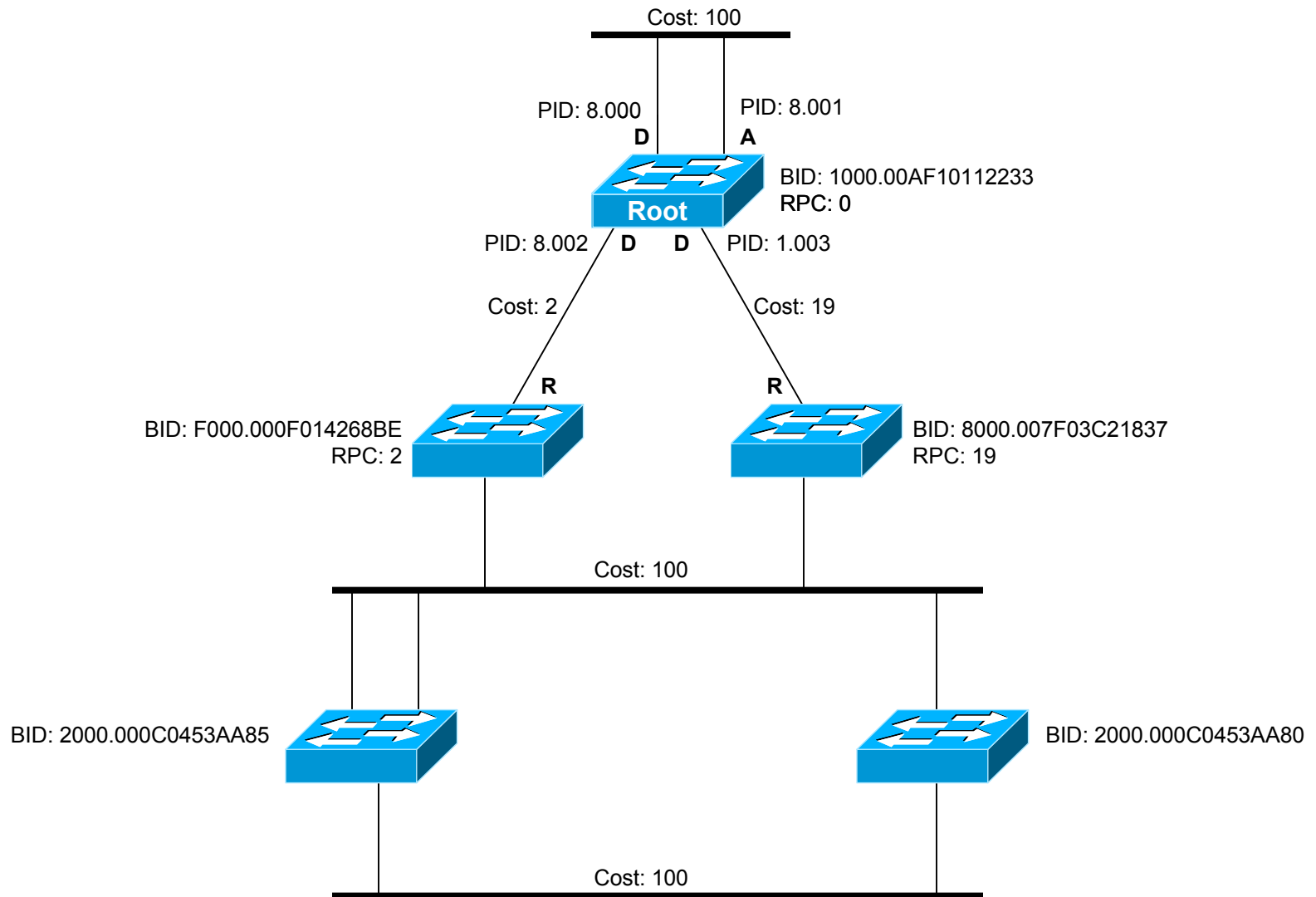
Példa: Feszítőfa



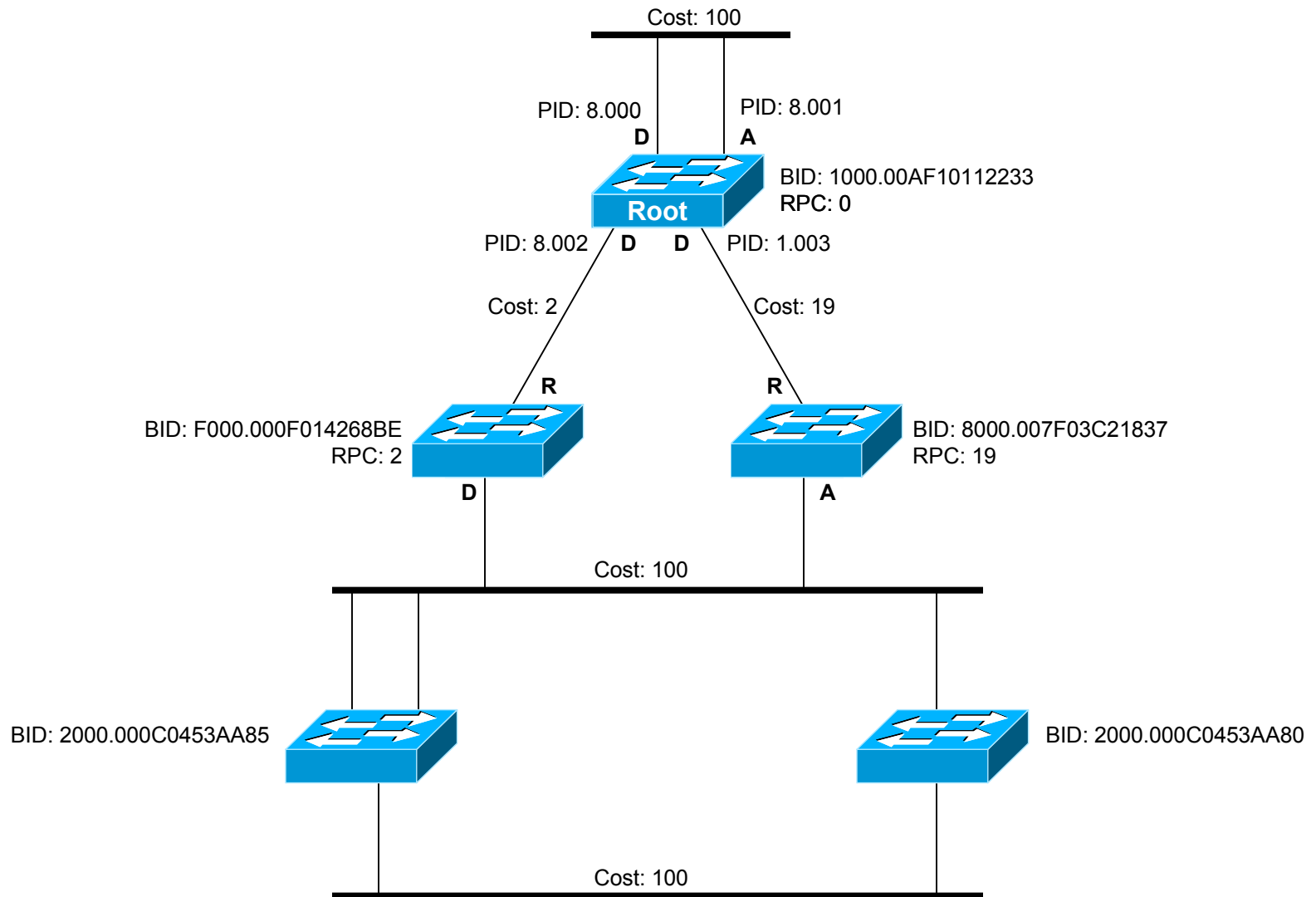
Példa: Feszítőfa



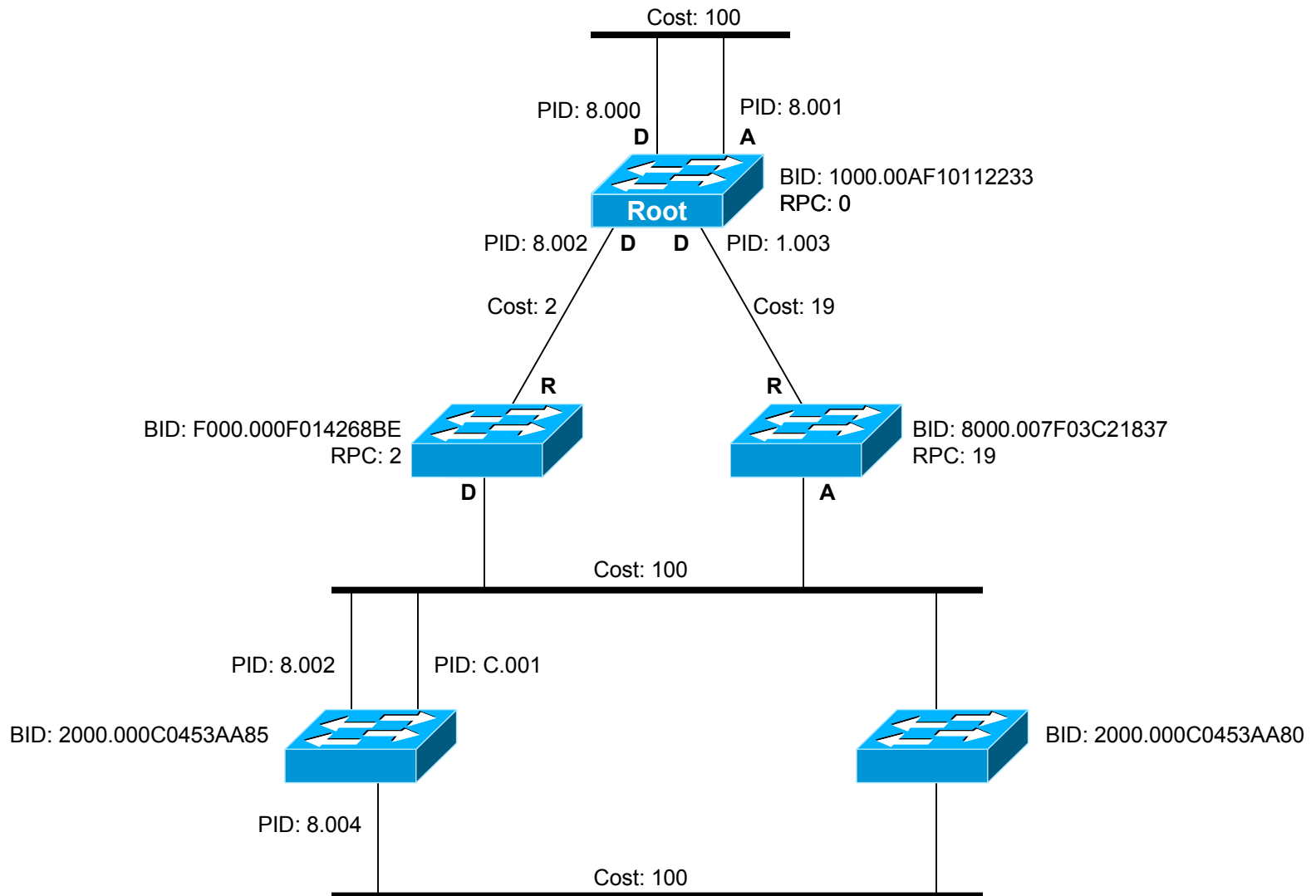
Példa: Feszítőfa



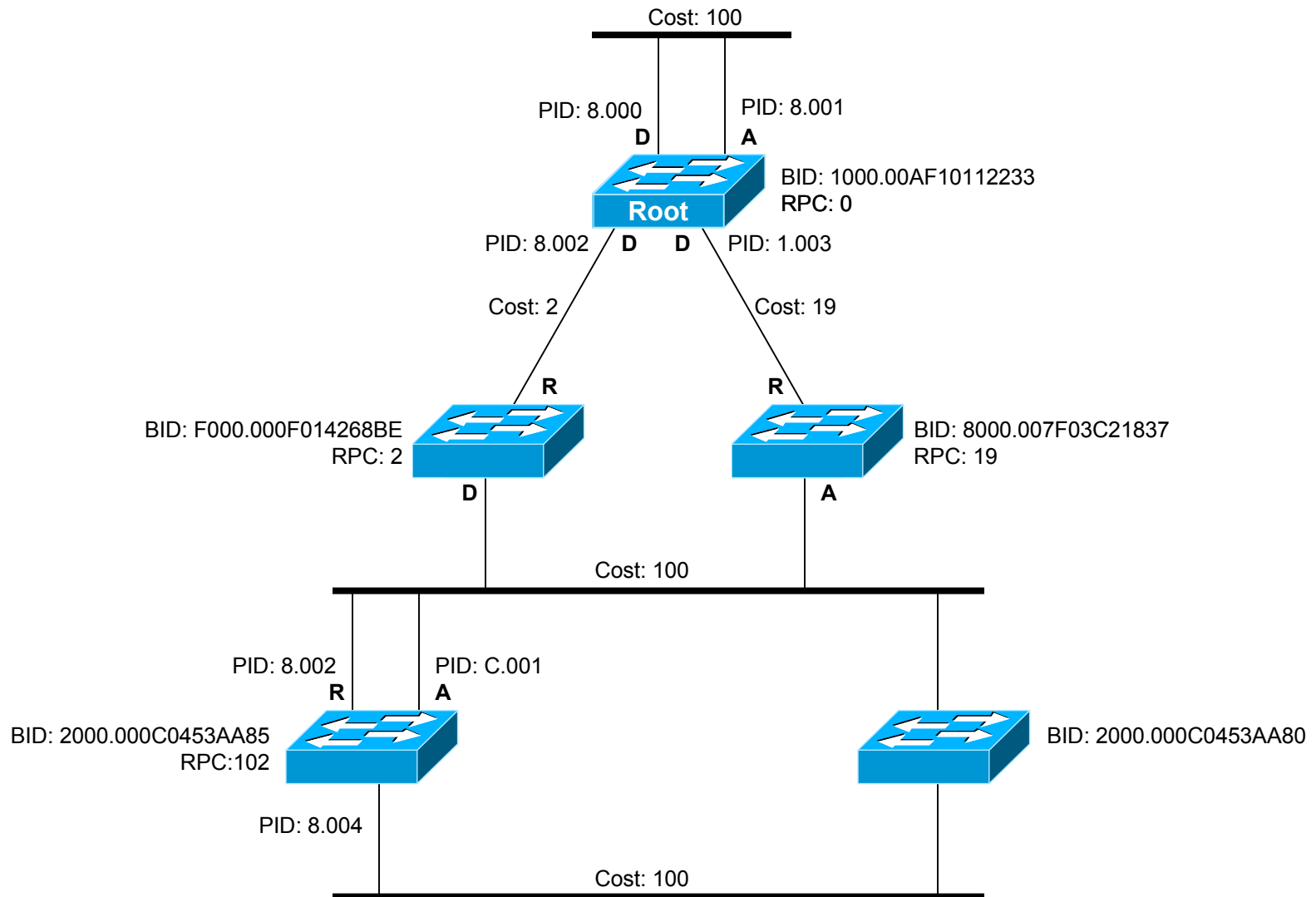
Példa: Feszítőfa



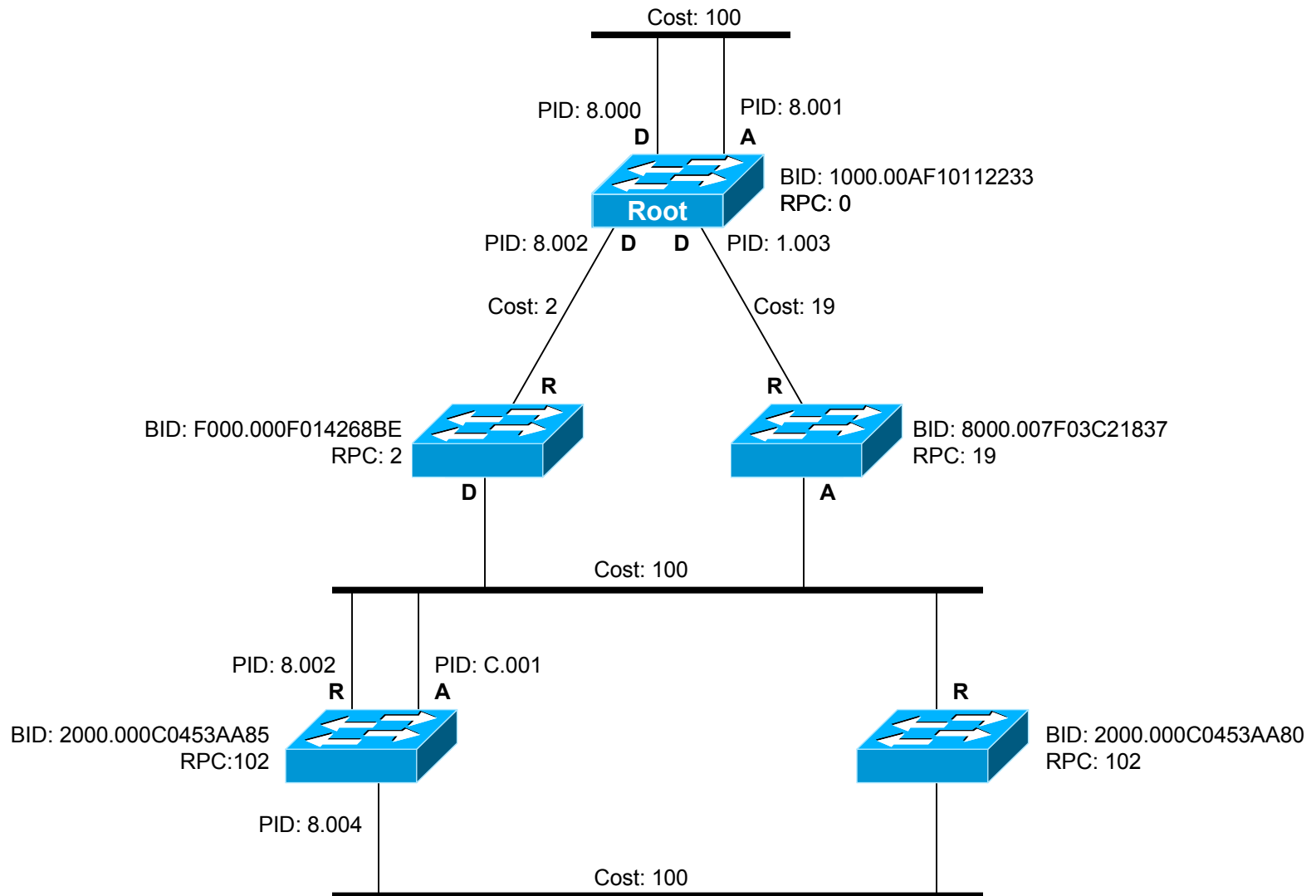
Példa: Feszítőfa



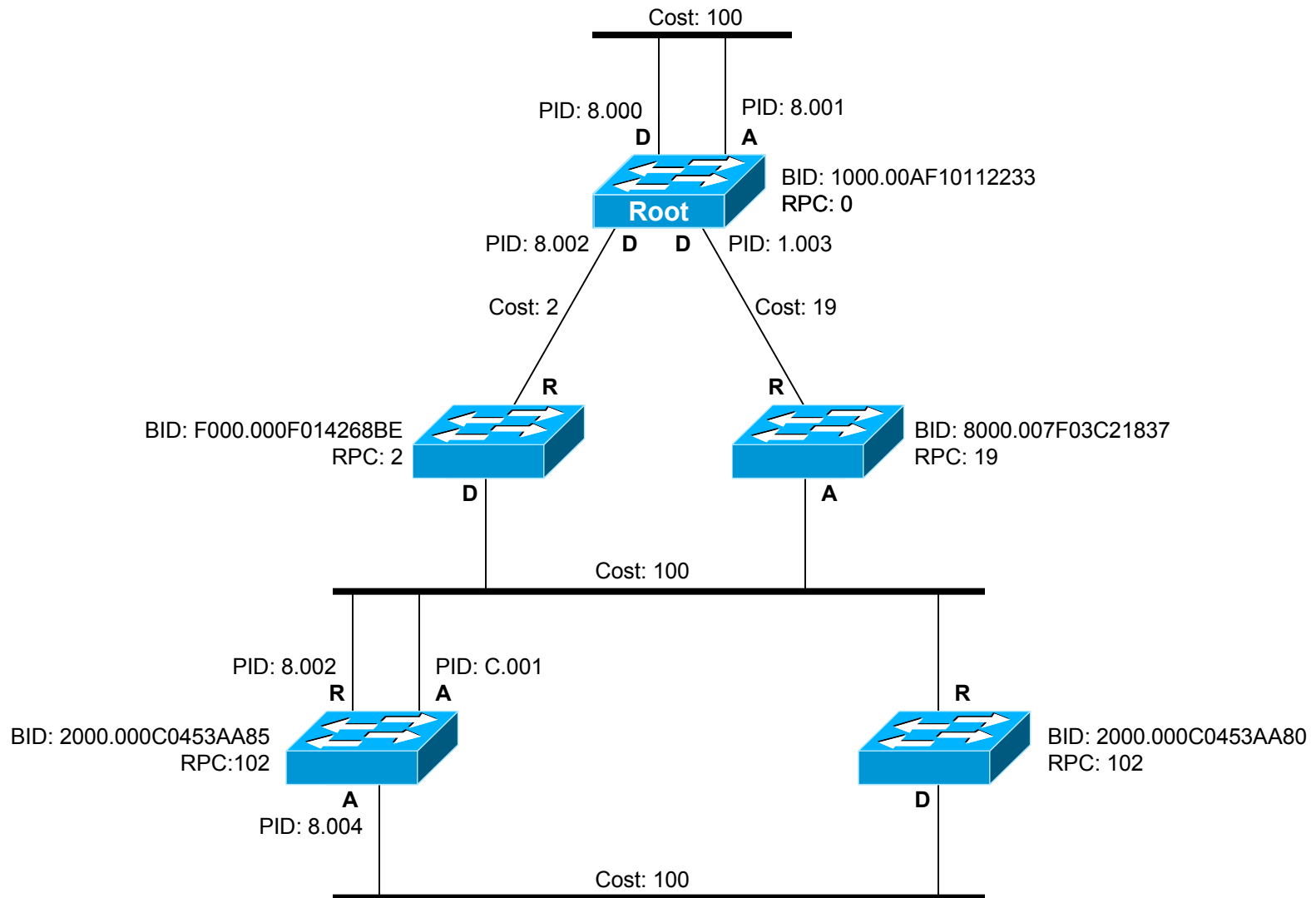
Példa: Feszítőfa



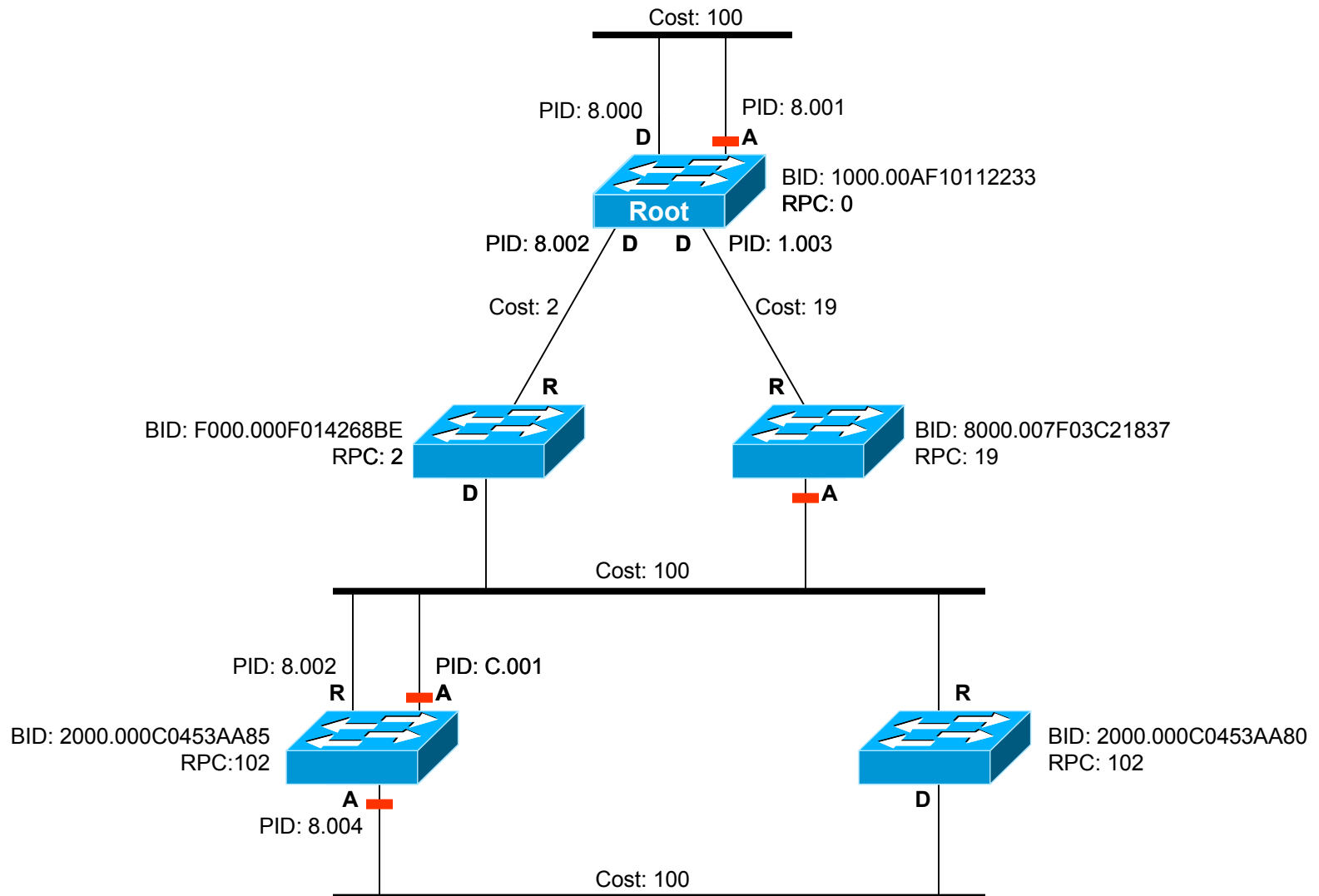
Példa: Feszítőfa



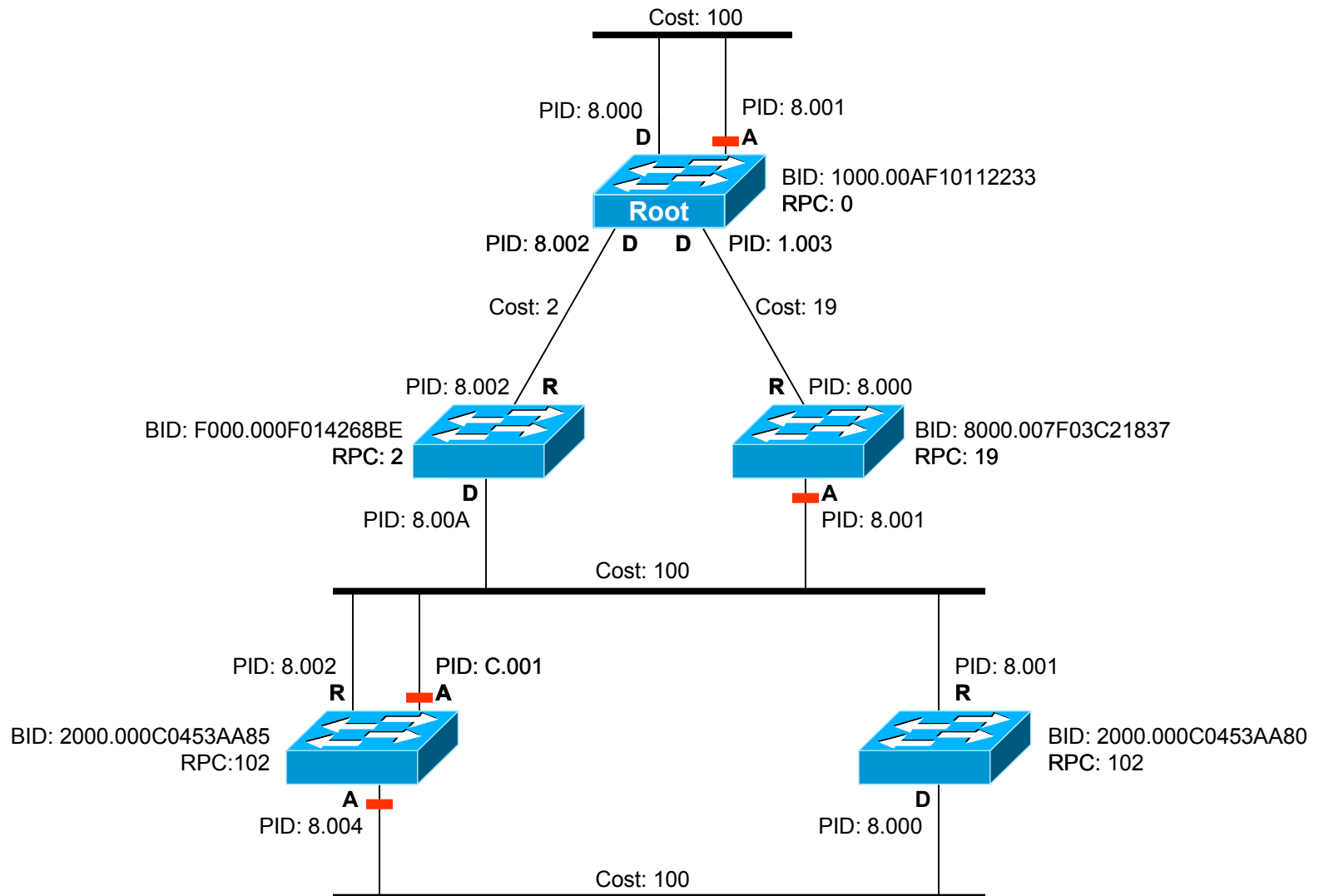
Példa: Feszítőfa



Példa: Feszítőfa

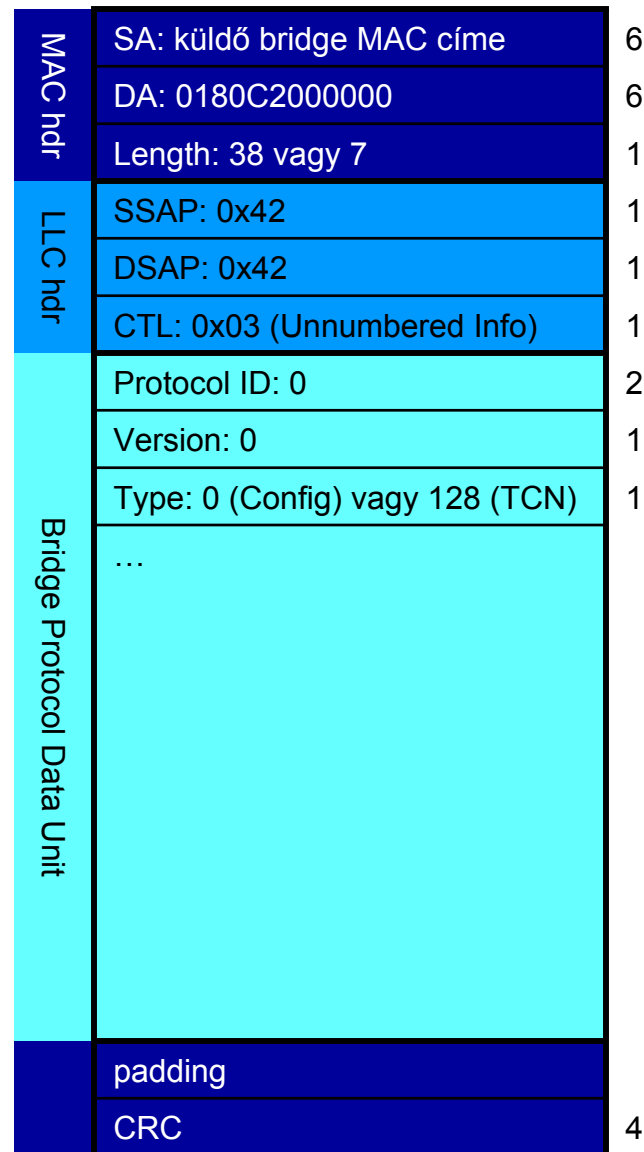


Példa: Feszítőfa



Bridge Protocol Data Unit

- a **Spanning Tree Protocol** üzenetei
- bridge-ek küldik **szomszédai**knak
- nem továbbítják a collision domainen kívülre
- cél MAC cím: 01:80:C2:00:00:00
 - „minden bridge”
 - multicast
- LLC
 - SSAP: 0x42
 - DSAP: 0x42
- két BPDU típus
 - **Configuration** BPDU
 - **Topology Change Notification** BPDU



Configuration BPDU – Root ID

- **Root ID:** root BID a küldő szerint
 - ha **jobbat hall** az addig feltételezett Root ID-nál, akkor **megtanulja** azt
 - és ezentúl azt hirdeti
 - stabil állapotban egyetértés van, mindenki azonos Root ID-t hirdet

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

Configuration BPDUs – Root Path Cost

- Root ID: root BID a küldő szerint
- **Root Path Cost**: a küldő távolsága a Root Bridge-től
 - hozzáadva a kettőnk közti távolságot megvan a roothoz a küldőn át vezető út hossza
 - ezek közül a legrövidebb a **saját Root Path Cost**
 - ez a **Bellman-Ford** algoritmus
 - ha úgy tetszik, a STP magja egy distance vector routing protocol, ahol a distance vector egyetlen eleme a fa gyökerétől való távolságot adja meg (azaz a fa gyökere az egyetlen útvonalválasztási cél)
 - Designated Bridge kiválasztásához

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

Configuration BPDU – Bridge ID

- Root ID: root BID a küldő szerint
- Root Path Cost: a küldő távolsága a Root Bridge-től
- **Bridge ID**: küldő BID
 - Designated Bridge kiválasztásához egyenlő Root Path Cost esetén

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

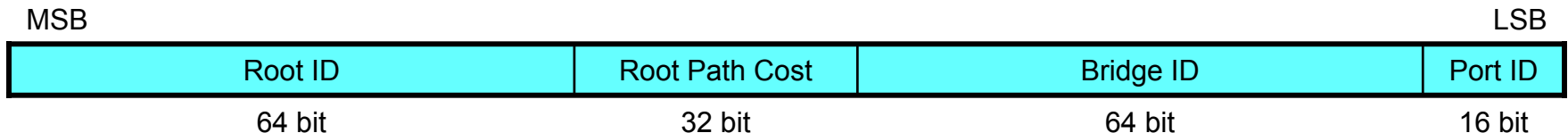
Configuration BPDU – Port ID

- Root ID: root BID a küldő szerint
- Root Path Cost: a küldő távolsága a Root Bridge-től
- Bridge ID: küldő BID
- **Port ID**: a küldő port azonosítója
 - Designated Port kiválasztásához azonos Bridge ID esetén
 - a bridge **máshonnan nem tudja**, hogy több portja azonos collision domainben van, csak a saját magától hallott Config BPDU-ból

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

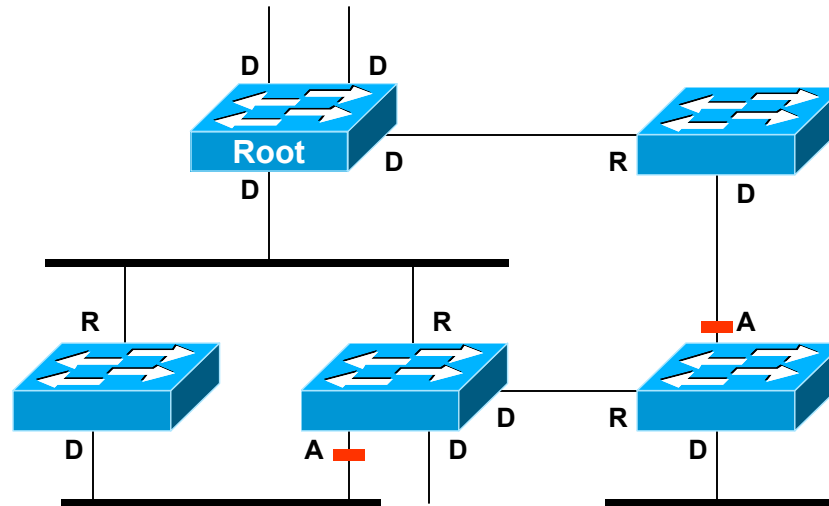
Config BPDU feldolgozás

- két Config BPDU egyszerűen összehasonlítható
 - (Root ID, Root Path Cost, Bridge ID, Port ID) egy **hosszú számként kezelhető**
 - hallott BPDU esetén összehasonlítás előtt a Root Path Cost-ot korigálni kell
- minden bridge **nyilvántartja portonként a legjobb BPDU-t**
 - beleszámítja a hallott BPDU-kat és a sajátját is, amit azon a porton küldene
 - ha jobbat hall a jelenleginél, akkor
 - azt teszi el
 - újraszámít minden szükséges paramétert
- az aktuális legjobb BPDU-kból adódik:
 - a Root BID
 - a saját Root Path Cost és saját Root Port
 - hogy a saját portok közül melyik Designated Port
 - amelyiken a saját BPDU a legjobb



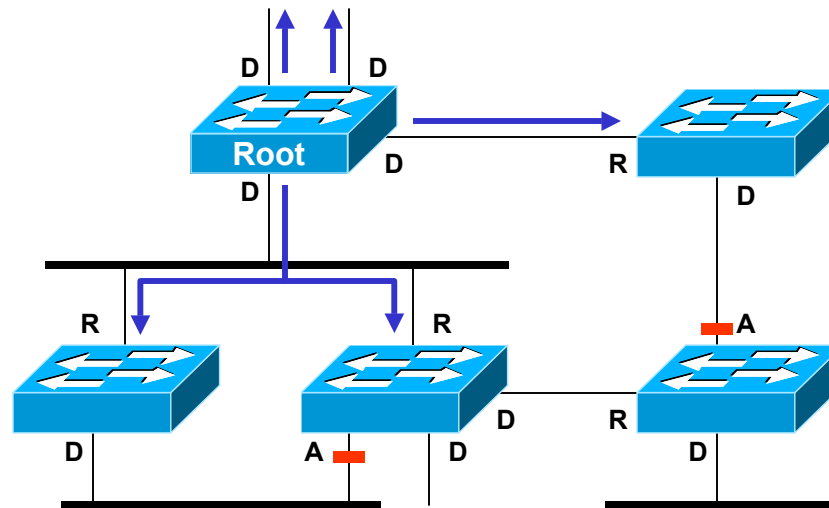
Config BPDU küldés

- a **Designated Portjain** küld Config BPDU-t minden bridge
- a **Root Bridge periodikusan** küld Config BPDU-kat
- a **többi bridge** akkor küld Config BPDU-kat, **amikor Config BPDU-t kap** a Root Porton
 - stabil állapotban periodikusan „szaladnak szét” Config BPDU-k a Root Bridge-től a feszítőfa leveleihez



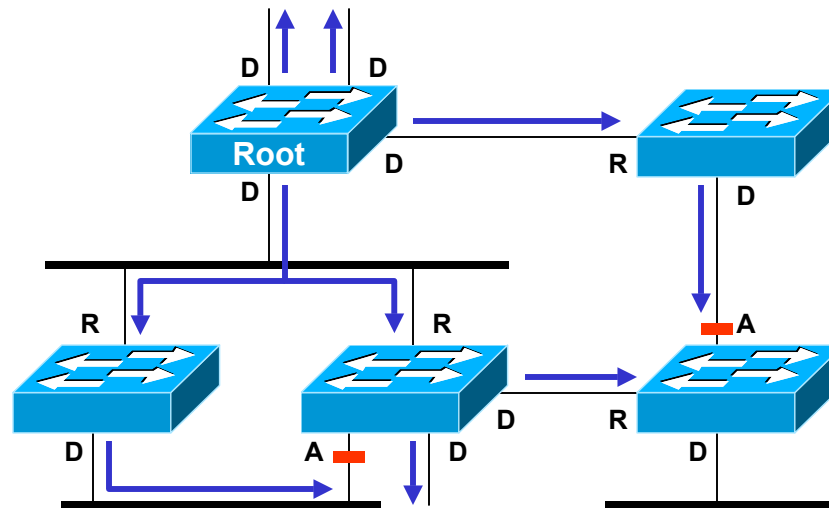
Config BPDU küldés

- a **Designated Portjain** küld Config BPDU-t minden bridge
- a **Root Bridge periodikusan** küld Config BPDU-kat
- a **többi bridge** akkor küld Config BPDU-kat, **amikor Config BPDU-t kap** a Root Porton
 - stabil állapotban periodikusan „szaladnak szét” Config BPDU-k a Root Bridge-től a feszítőfa leveleihez



Config BPDU küldés

- a **Designated Portjain** küld Config BPDU-t minden bridge
- a **Root Bridge periodikusan** küld Config BPDU-kat
- a **többi bridge** akkor küld Config BPDU-kat, **amikor Config BPDU-t kap** a Root Porton
 - stabil állapotban periodikusan „szaladnak szét” Config BPDU-k a Root Bridge-től a feszítőfa leveleihez

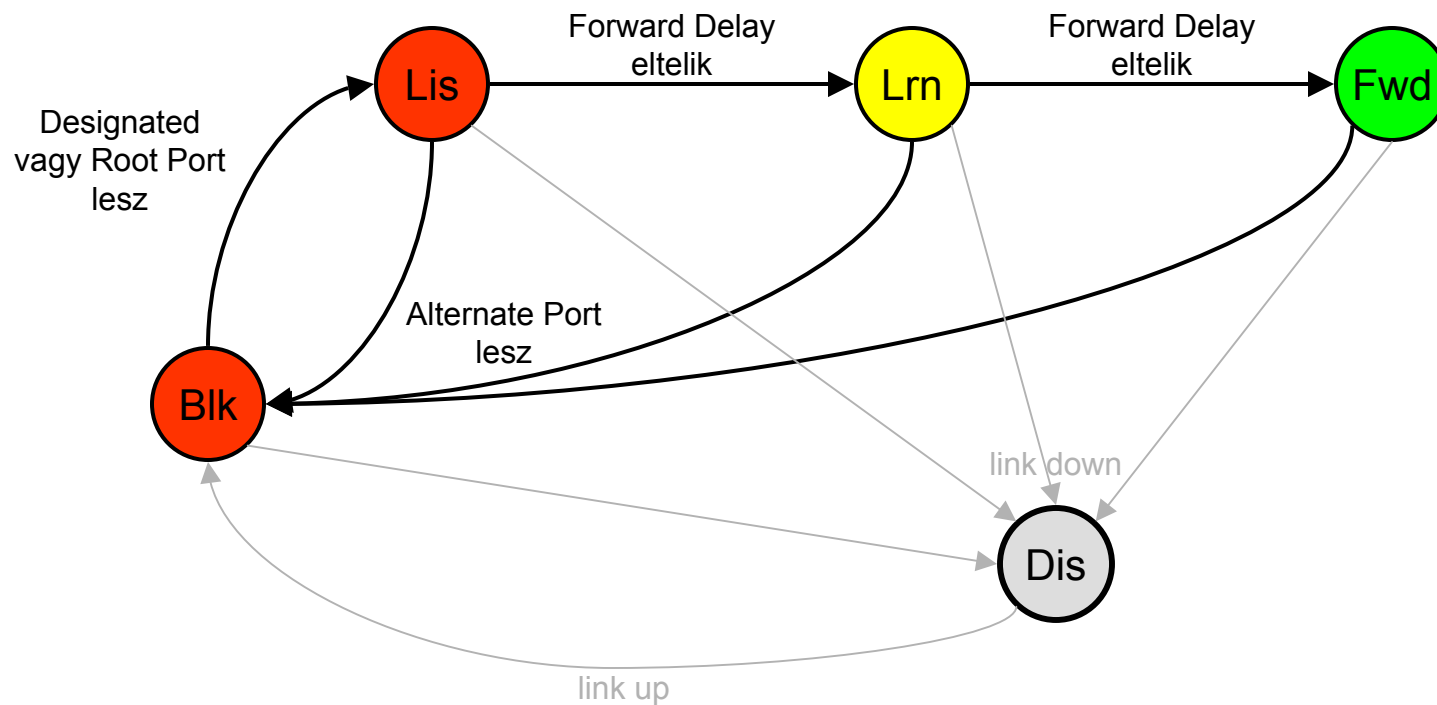


Port állapotok

- **Blocking**
 - csak a **BPDU-kat hallgatja**, adatforgalom nem megy rajta se ki se be
- **Listening, Learning**
 - most lett Designated vagy Root Port
 - **csak BPDU-kat küld és fogad**, adatkereteket nem, amíg kiderül, hogy tényleg a feszítőfa része marad-e
 - ha ilyenkor hall egy jobb BPDU-t, akkor visszamegy Blocking állapotba (Alternate Port szerepbe)
 - hurkot rövid időre sem szabad megengedni!!!
 - Learning állapotban már tanulja a hallott adatkeretek forráscímét
- **Forwarding**
 - „rendesen” **megy** rajta **minden forgalom**

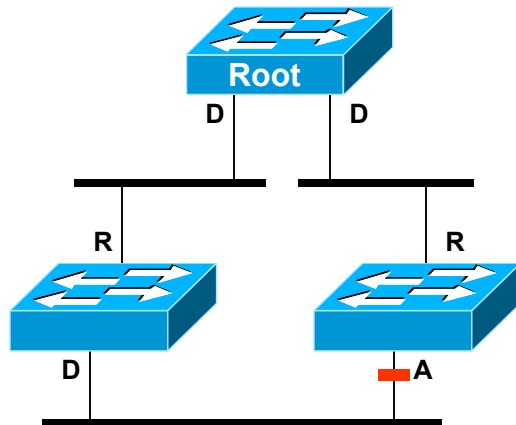
port szerep	port állapot
Root Designated	Forwarding
	Learning
	Listening
Alternate	Blocking
Disabled	Disabled

Port állapotgép



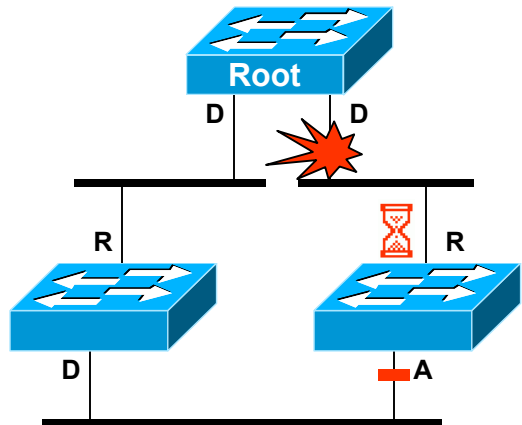
Bridge kiesése, link megszakadása

- a nyilvántartott legjobb **Config BPDUs** öregednek
 - ha nem jön új BPDUs a root felől
 - **Max Age** elérésekor törli a bridge a BPDUs-t
 - elkezdődik a feszítőfa (egy részének) újraszámítása



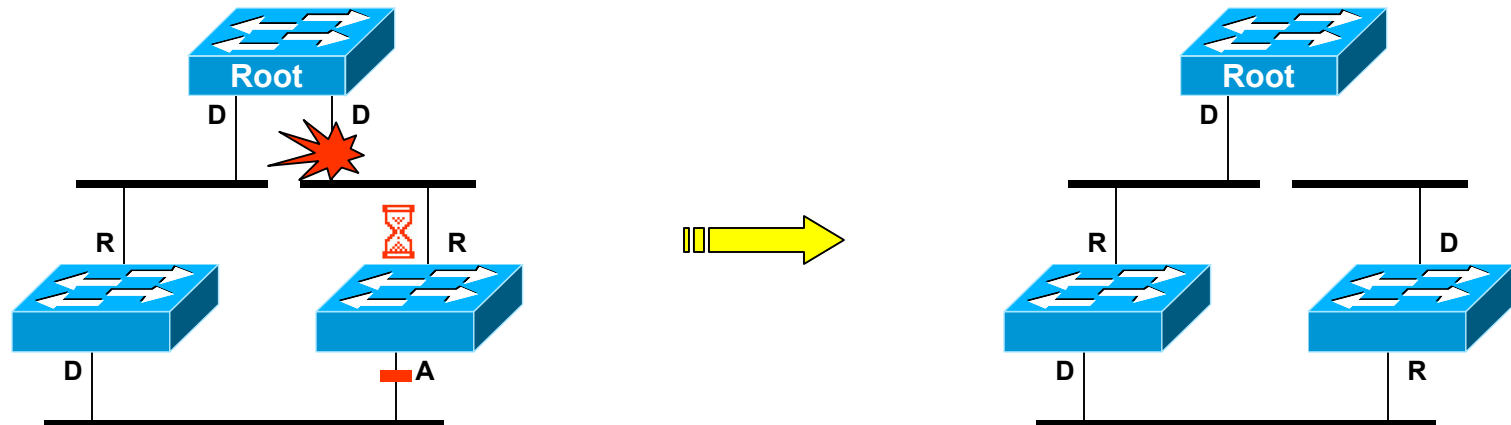
Bridge kiesése, link megszakadása

- a nyilvántartott legjobb **Config BPDUs** öregednek
 - ha nem jön új BPDUs a root felől
 - **Max Age** elérésekor törli a bridge a BPDUs-t
 - elkezdődik a feszítőfa (egy részének) újraszámítása



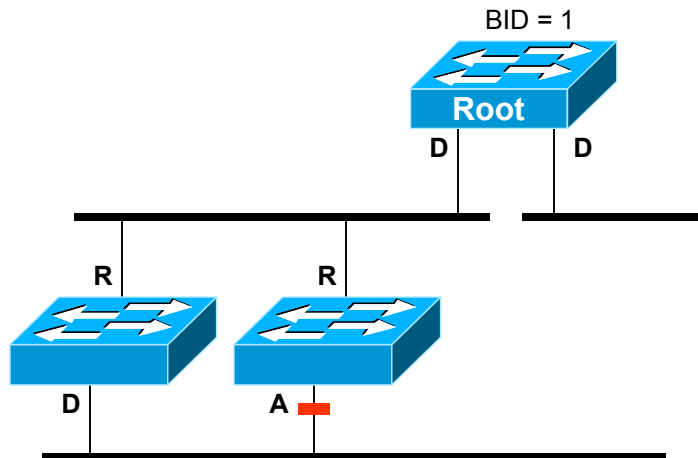
Bridge kiesése, link megszakadása

- a nyilvántartott legjobb **Config BPDUs** öregednek
 - ha nem jön új BPDUs a root felől
 - **Max Age** elérésekor törli a bridge a BPDUs-t
 - elkezdődik a feszítőfa (egy részének) újraszámítása



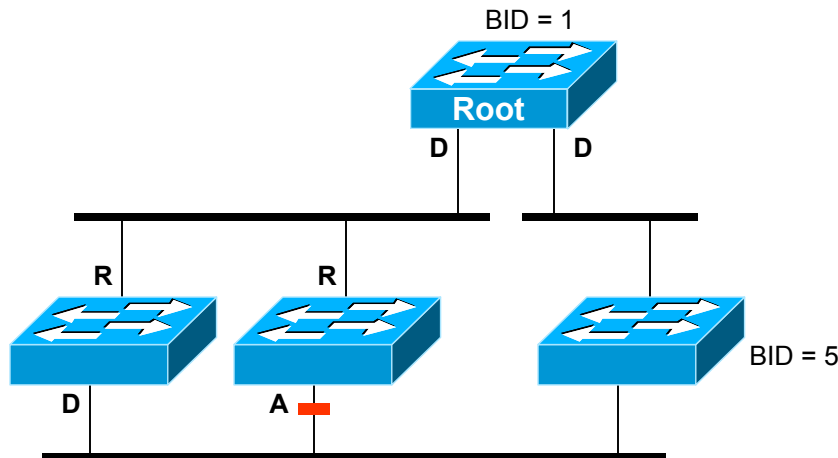
Bridge, link feléléedése

- a feléléedő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - feléléedő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



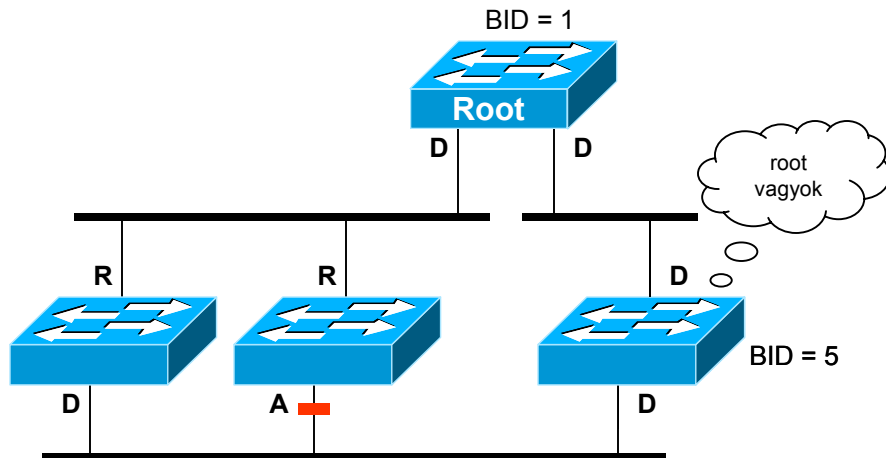
Bridge, link felélédése

- a felélédő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - felélédő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



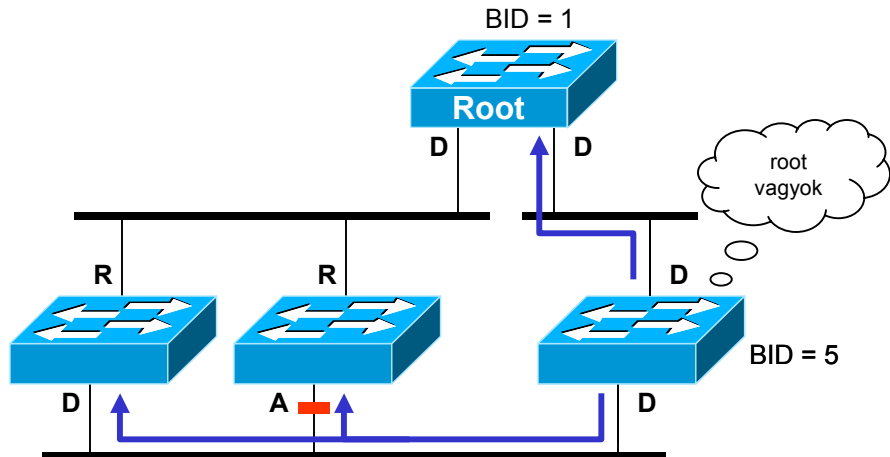
Bridge, link feléléedése

- a felélédő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - felélédő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



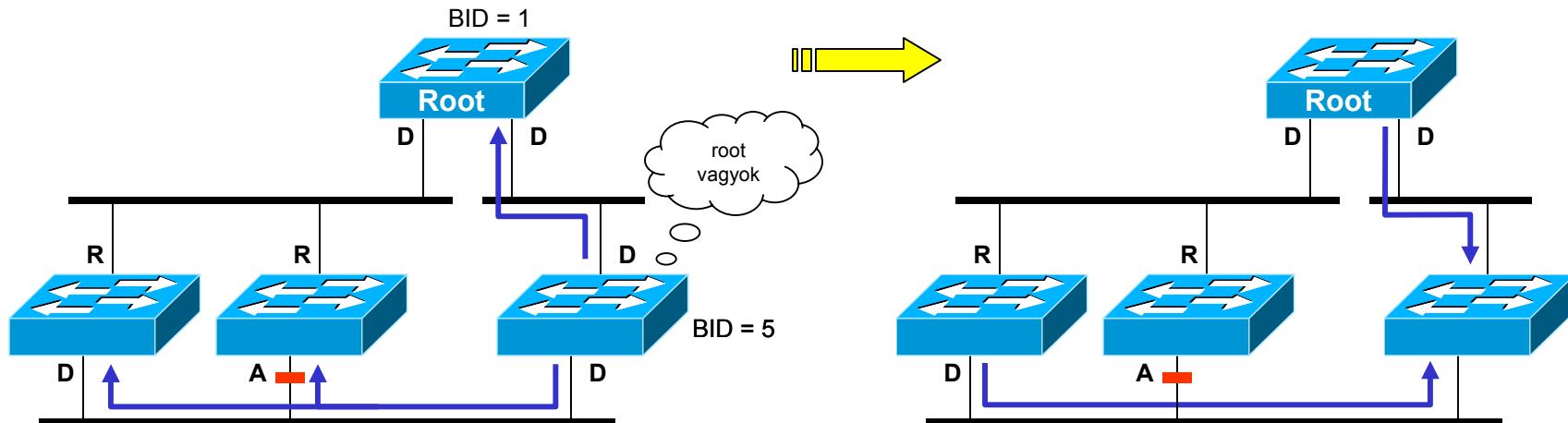
Bridge, link felélédése

- a feléledő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - feléledő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



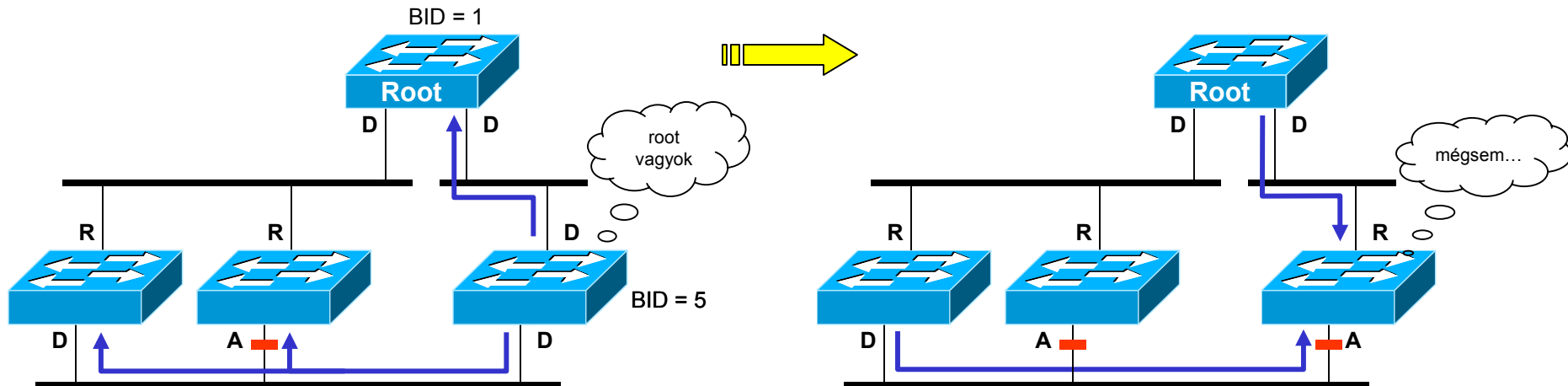
Bridge, link felélédése

- a felélédő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - felélédő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



Bridge, link felélédése

- a felélédő bridge/port mindig **sokat feltételez magáról**
 - induló bridge: ő a Root Bridge
 - felélédő port: Designated Port
- a túl nagyratörő feltételezésekről a tényleges **Designated Bridge lebeszéli** őket
 - ha a **Designated Porton** az aktuálisnál **rosszabb Config BPDU érkezik**, akkor a Designated Bridge elküldi az érvényes jobb BPDU-t
 - azonnal, nem várja meg, amíg Config BPDU-t kap a Root Portján



Configuration BPDU – Message Age

- **Message Age**: becsült eltelt idő, amióta a Root Bridge a jelen BPDU alapjául szolgáló BPDU-t generálta
 - 1/256 sec-ben megadva
 - ha rég jött BPDU a root felől, amikor egy új portot/bridge-et kell lebeszélni, akkor öreg BPDU-t kap az új eszköz
 - nála sem fog tovább élni a régi BPDU, mint másutt

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

STP időzítési paraméterek

- **Hello Time**: a Root Bridge ilyen időközönként küld Config BPDU-t
 - konfigurálható (default: 2 sec)
- **Forward Delay**: a Listening és a Learning állapotban eltöltött maximális idő
 - konfigurálható (default: 15 sec)
 - amikor egy port Designated vagy Root Port lesz, akkor 30 másodpercig (2 * Forward Delay) csak BPDU-kat forgalmaz, hogy az esetlegesen inkonzisztens állapotban se keletkezzen hurok
- **Max Age**: a Config BPDU maximális életkora
 - konfigurálható (default: 20 sec)
- **Hold Time**: két Config BPDU küldése közti minimális idő
 - eredetileg (IEEE 802.1D-1998): 1 sec
 - később (IEEE 802.1t-2001): maximum 3 BPDU 2 másodperc alatt

Miért pont ennyi?

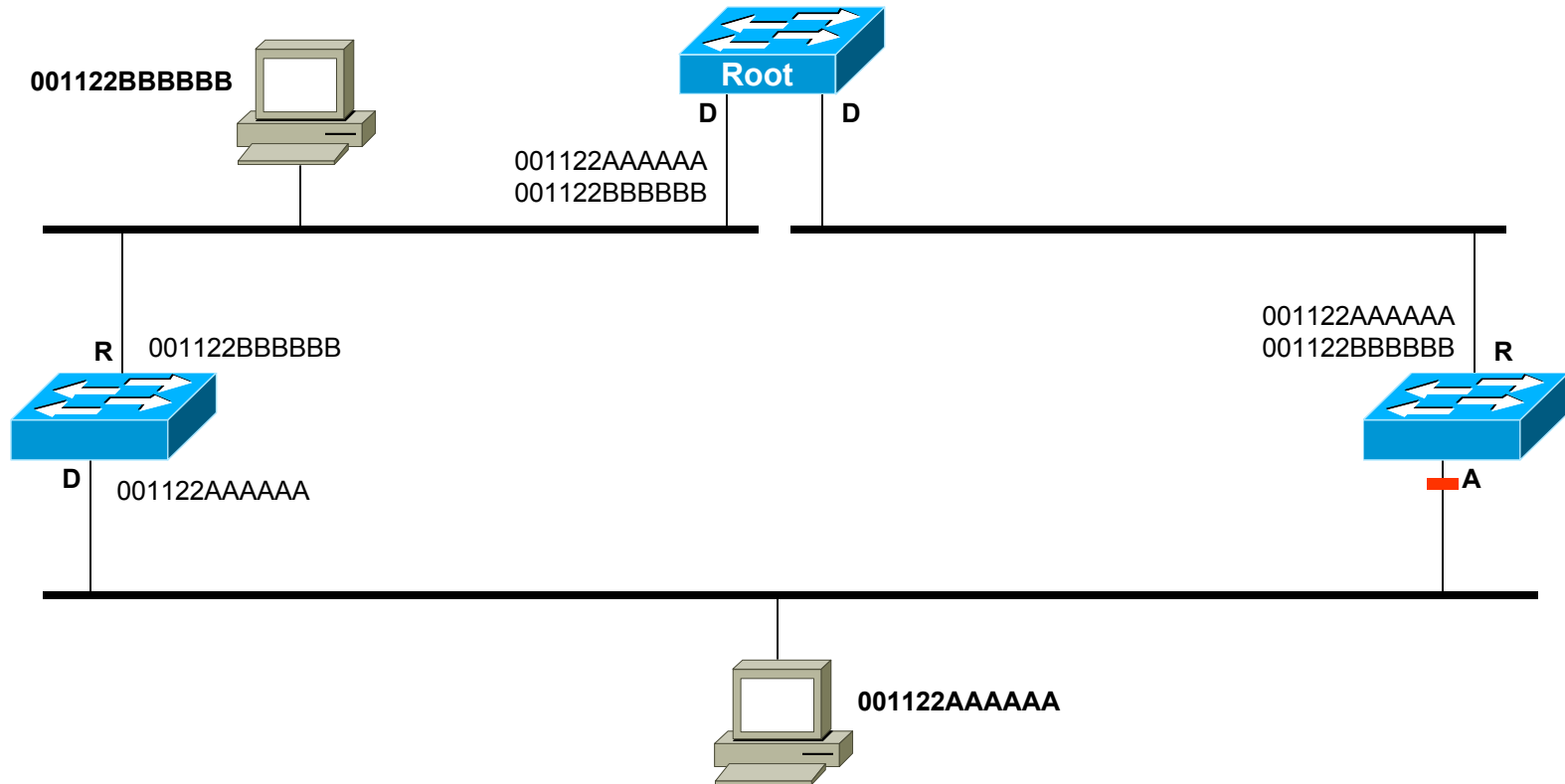
- pl.: Max Age
 - ennyi időnként mindig el kell jutnia legalább egy Config BPDU-nak a hálózat széléig, hogy ne legyen timeout
 - tényleges terjedési idő
 - kiindulási paraméterek:
 - 3 egymást követő Config BPDU elvész: *lost_msgs=3*
 - 2 sec Hello Time: *hello_t=2s*
 - 7 bridge a hálózat átmérője: *dia=7*
 - 1 sec PDU továbbítási késleltetés bridge-enként: *pdu_d=1s*
 - nem fut szinkronban a bridge-ek órája: még egyszer *hello_t*
 - $(lost_msgs+1)*hello_t + pdu_d*(dia-1) = 14s$
 - idő túlbecsülése a Message Age növelésekor
 - kb. *pdu_d*-vel növeli a Message Age-et minden bridge (túlbecsülve a tényleges továbbítás idejét)
 - $pdu_d*(dia-1) = 6s$
 - Max Age = 14s + 6s = 20s
- részletesebben: [IEEE 802.1D Annex B](#)

Configuration BPDU – Timers

- Message Age: becsült eltelt idő, amióta a Root Bridge a jelen BPDU alapjául szolgáló BPDU-t generálta
- **Max Age, Hello Time, Forward Delay**
 - 1/256 sec-ben megadva
 - konfigurálhatók, de **minden bridge-en azonosnak** kell lenniük
 - az aktuális Root Bridge-en beállított értékeket használja mindenki

Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

„Hibás” címtáblázatok



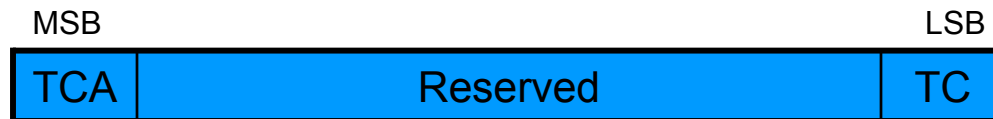
Topology Change Notification BPDU

- a TCN BPDU célja **tájékoztatni a Root Bridge-et** a topológia megváltozásáról
- a **változást észlelő bridge** TCN BPDU-t küld a Root Bridge felé
 - változásnak számít:
 - {Learning, Forwarding} → Blocking
 - → Forwarding, és van Designated Portja
 - a TCN BPDU-t mindig a **Root Porton küldi** minden bridge
- a következő bridge **nyugtázza** ezt, majd **továbbküldi** a TCN BPDU-t a root felé
 - ha nem érkezik nyugta, akkor Hello Time időközönként újra küldi a TCN BPDU-t

Protocol ID: 0	2
Version: 0	1
Type: 128 (TCN)	1

Configuration BPDU – Flags

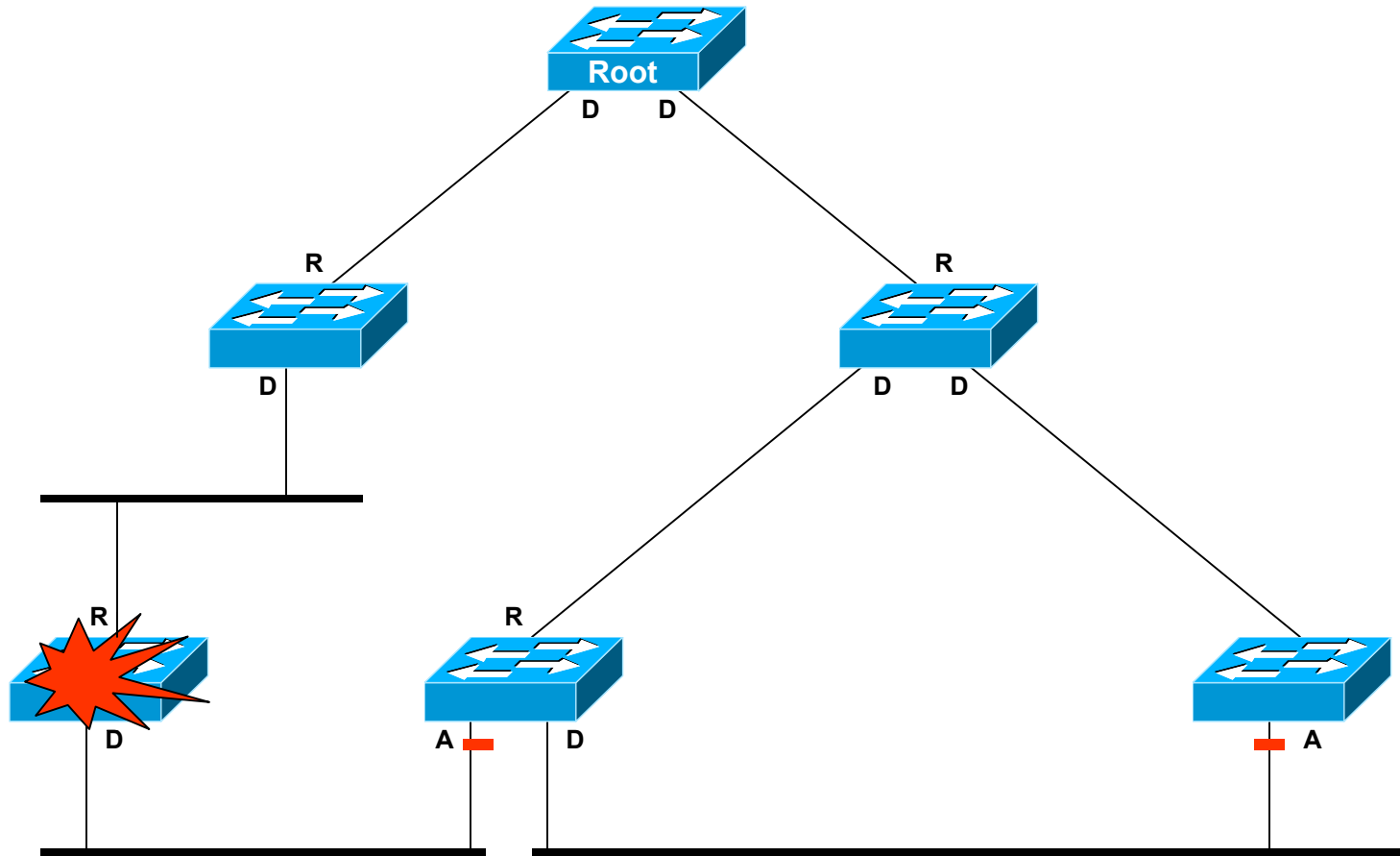
- **Topology Change Ack: TCN BPDU** nyugtázása



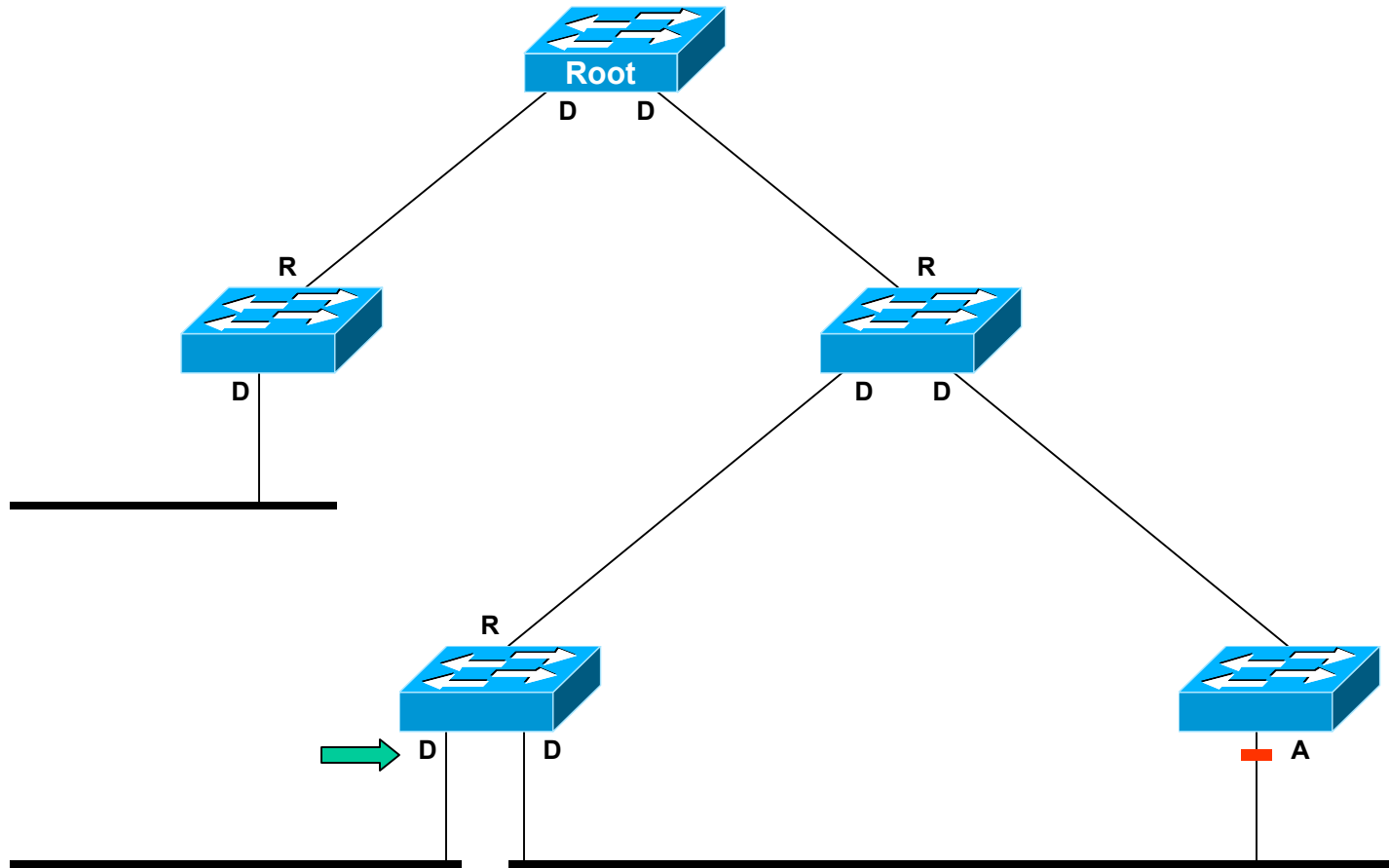
Protocol ID: 0	2
Version: 0	1
Type: 0 (Config)	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

- **Topology Change:** a Root Bridge jelzi minden bridge-nek, hogy változás történt
 - miután TCN BPDU-t kapott
 - Max Age + Forward Delay ideig minden Config BPDU-ban
 - hatására minden bridge **ideiglenesen csökkenti a címbejegyzések maximális életkorát** a Forward Delay értékére

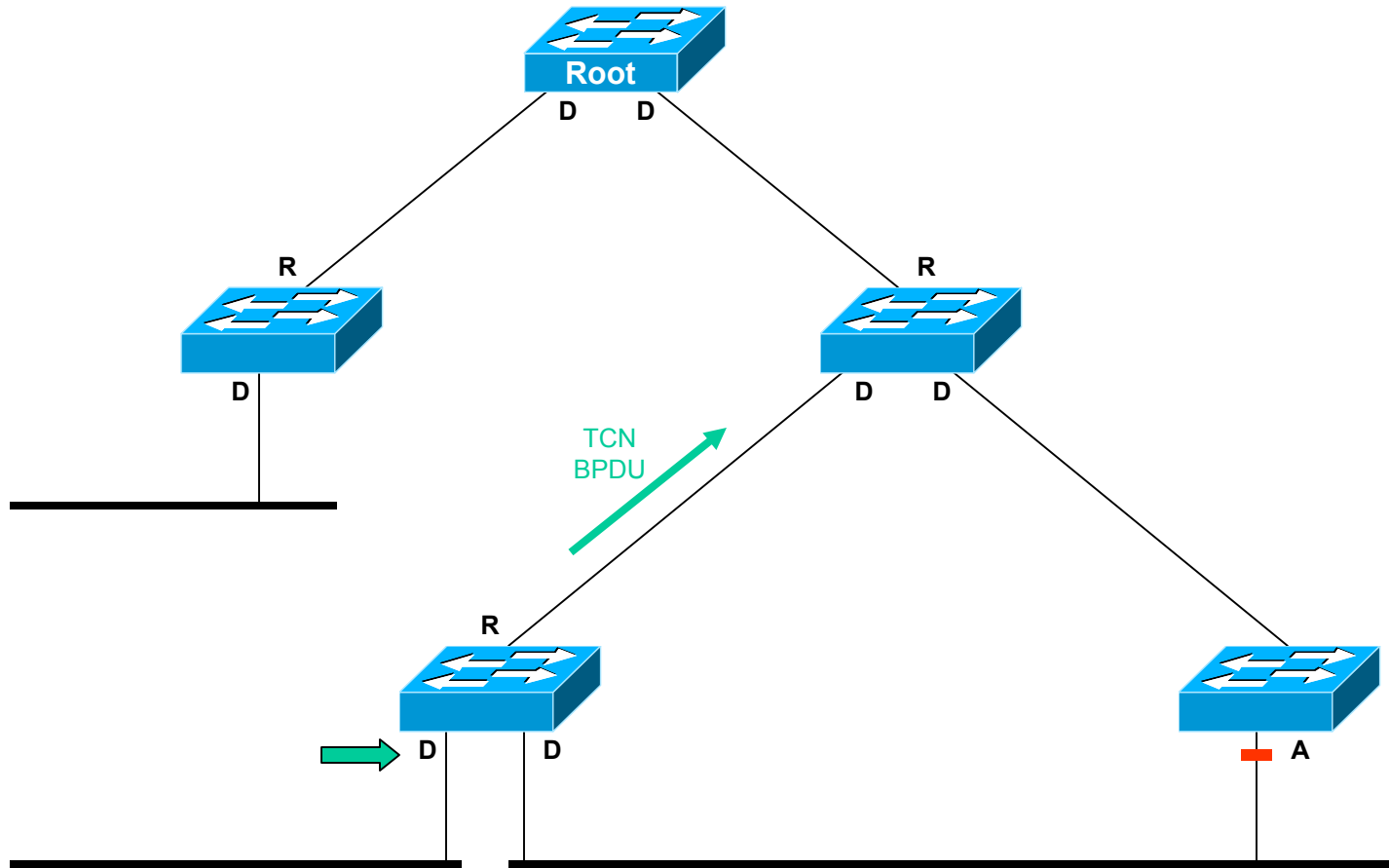
Topology change notification



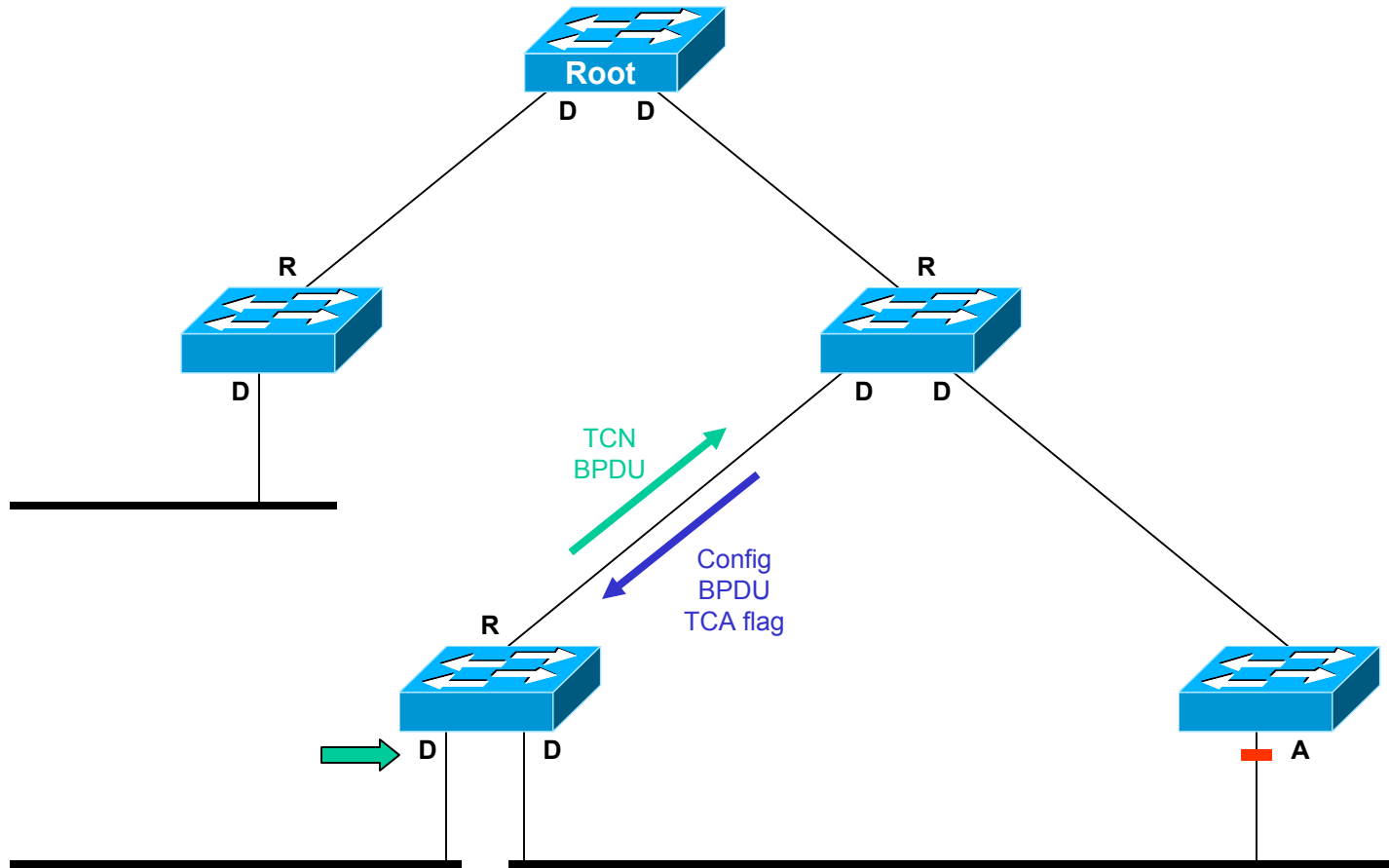
Topology change notification



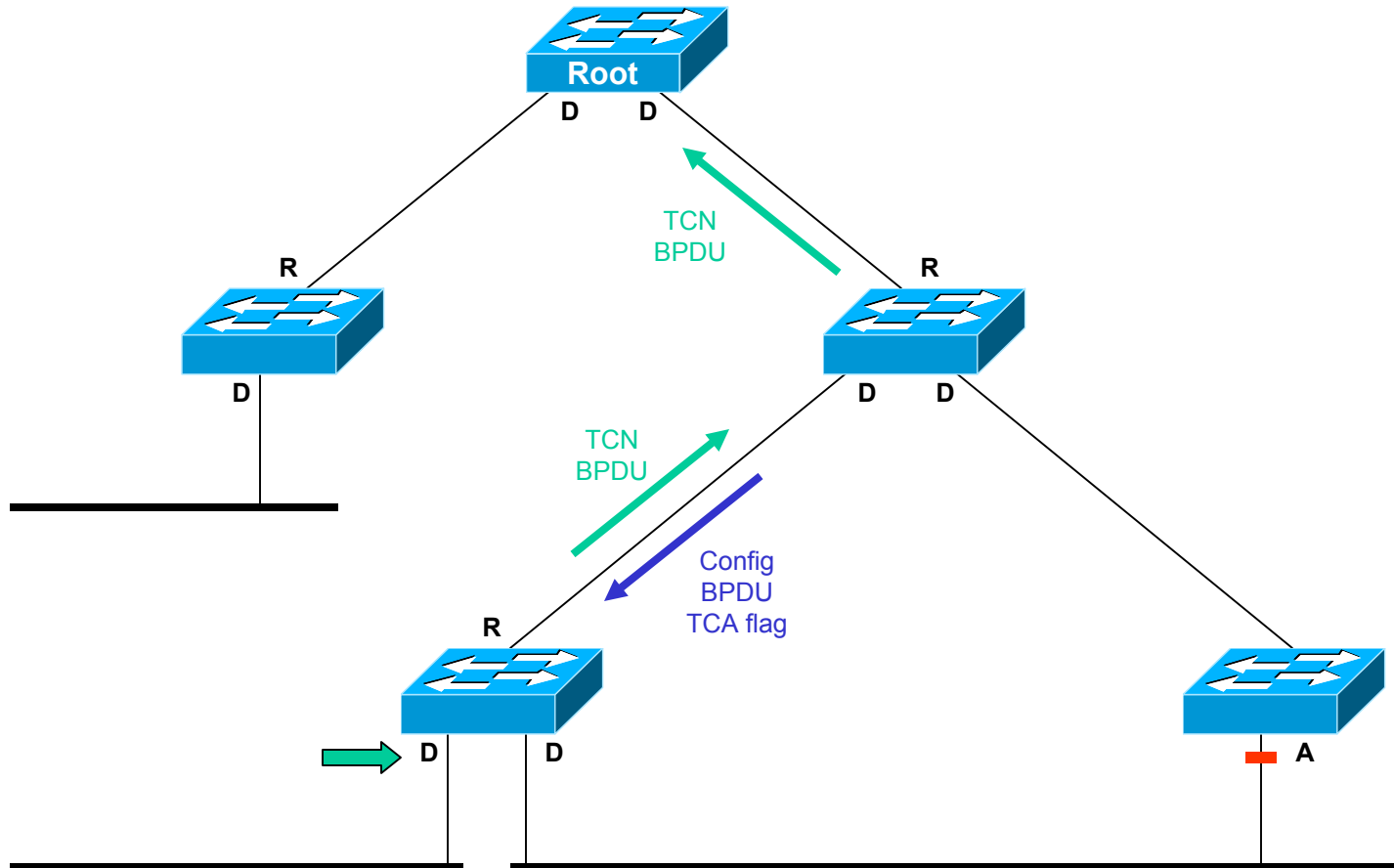
Topology change notification



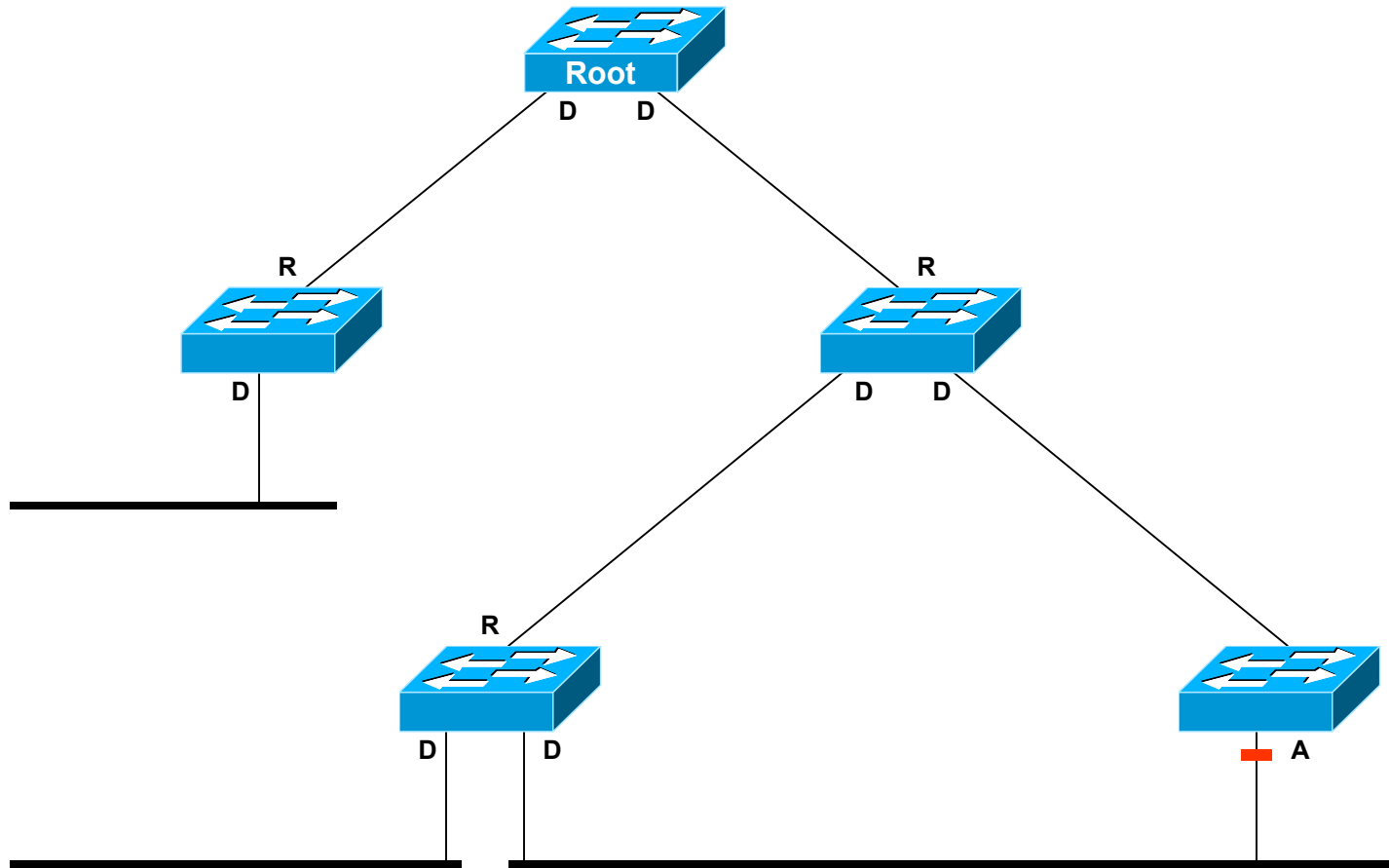
Topology change notification



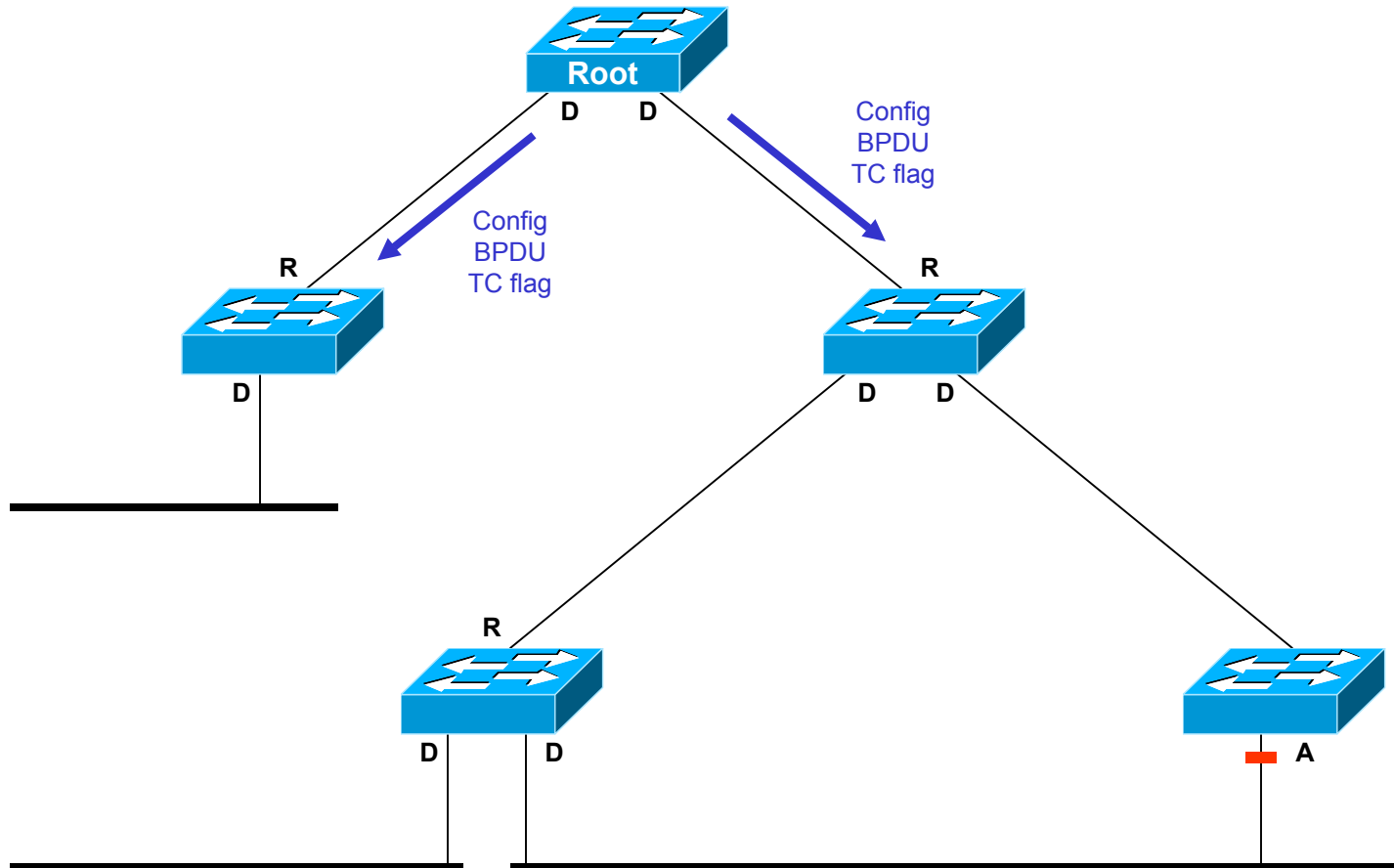
Topology change notification



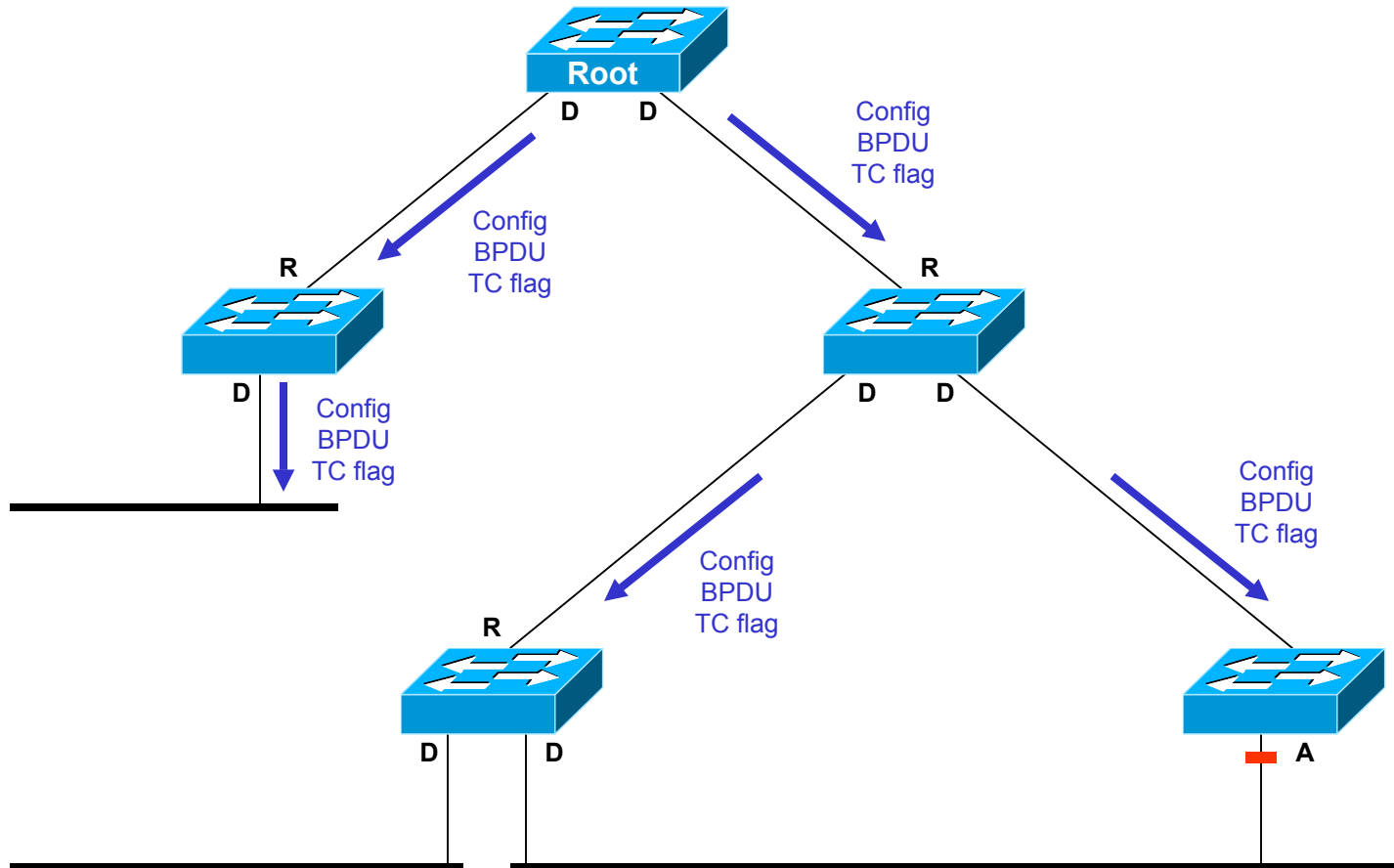
Topology change notification



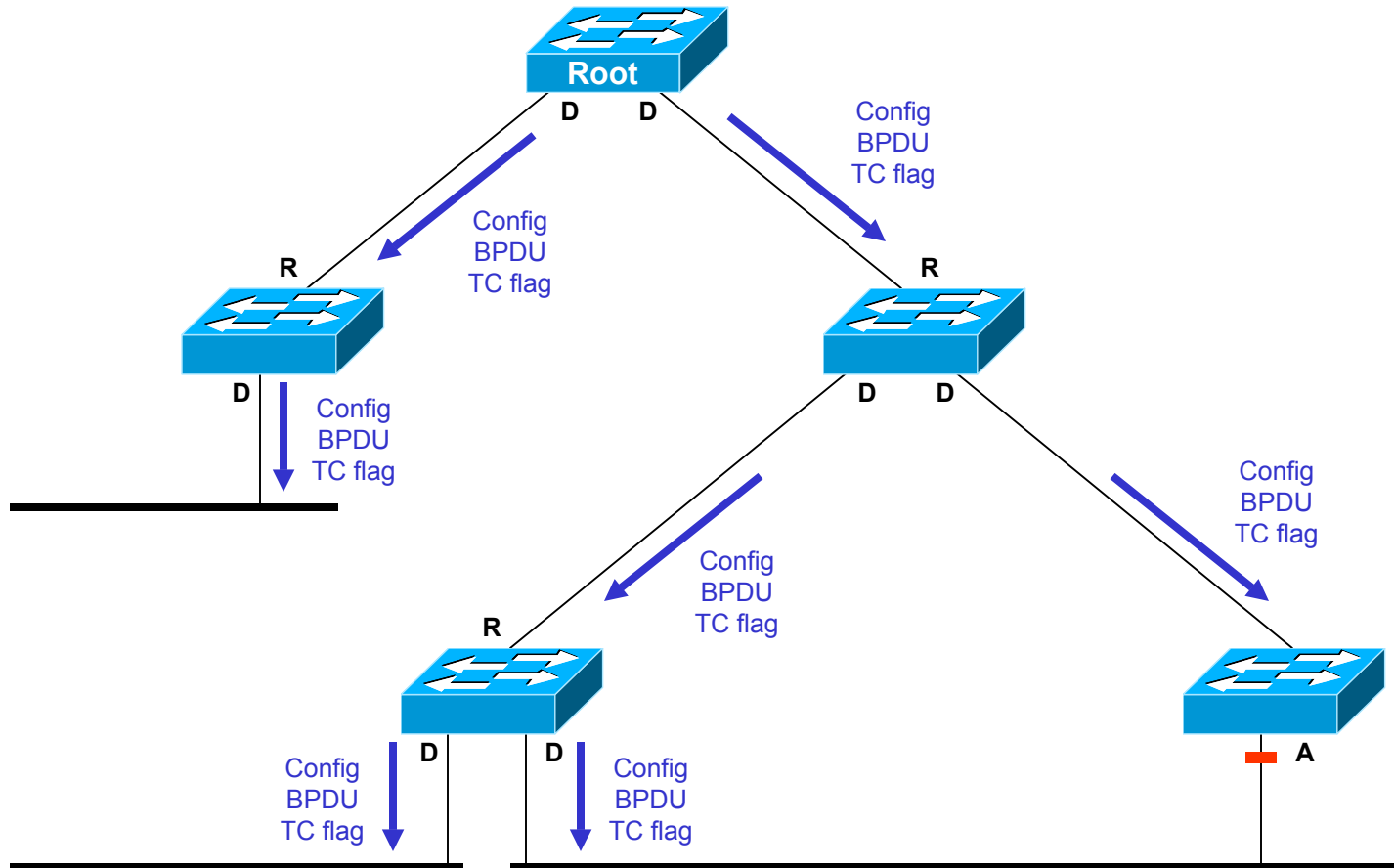
Topology change notification



Topology change notification



Topology change notification



Ajánlott cost értékek

	IEEE 802.1D-1990	IEEE 802.1D-1998	IEEE 802.1t-2001
<= 100 kbps	-	-	200 000 000
1 Mbps	1000	-	20 000 000
4 Mbps	250	250	-
10 Mbps	100	100	2 000 000
16 Mbps	62	62	-
100 Mbps	10	19	200 000
1 Gbps	1	4	20 000
10 Gbps	-	2	2 000
100 Gbps	-	-	200
1 Tbps	-	-	20
10 Tbps	-	-	2
általában	1 Gbps / bw	1 Gbps / bw módosítva	20 Tbps / bw

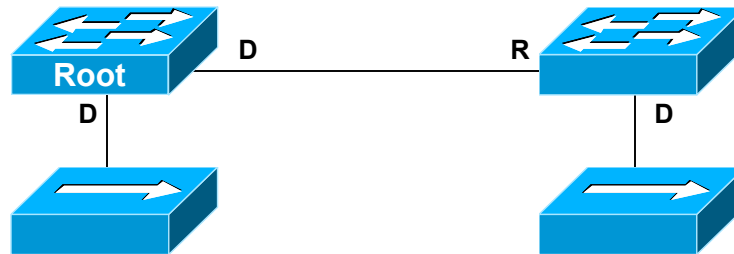
- a 802.1D 1990-es és 1998-as kiadása csak 1 – 65,535 közti értékeket enged meg
- a 802.1t már 1 – 200,000,000 köztieket
 - egy 20 tagból álló út hossza még így is elfér 4 byte-on a Config BPDU-ban

A fa topológiájának befolyásolása

- a bridge prioritás beállítható
 - érdemes is beállítani, ha redundáns a fizikai topológia
- a costok portonként beállíthatók
- a port prioritások beállíthatók

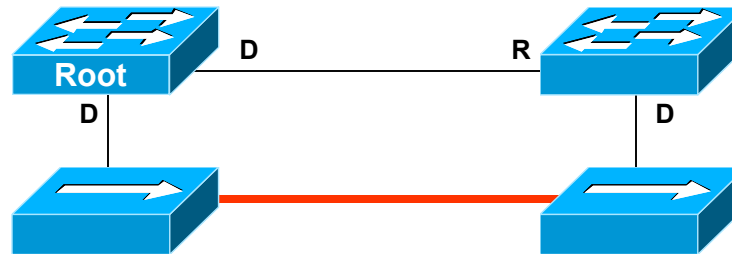
Amikor mégis hurok keletkezik

- kettő, már Forwarding állapotban levő port
- egyszer csak egybeolvasztjuk a két collision domaint
 - pl. hub vagy repeater segítségével
- szerencsés esetben valahogy átjut a jobbik Config BPDU



Amikor mégis hurok keletkezik

- kettő, már Forwarding állapotban levő port
- egyszer csak egybeolvasztjuk a két collision domaint
 - pl. hub vagy repeater segítségével
- szerencsés esetben valahogy átjut a jobbik Config BPDU



Tartalom

Bevezető

Spanning Tree Protocol

Rapid STP

Multiple STP

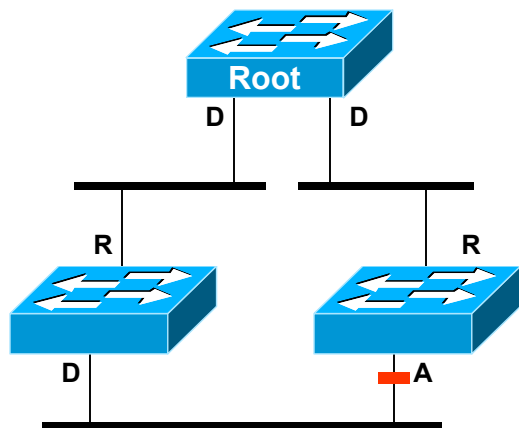
Rapid Reconfiguration

- Rapid Spanning Tree Algorithm and Protocol
- célja a topológia változásakor fellépő 30-50 másodperces kiesések rövidítése
- a kialakított feszítőfa topológia változatlan
- IEEE 802.1w-2001: Rapid Reconfiguration
 - az IEEE 802.1D következő kiadásának része lesz

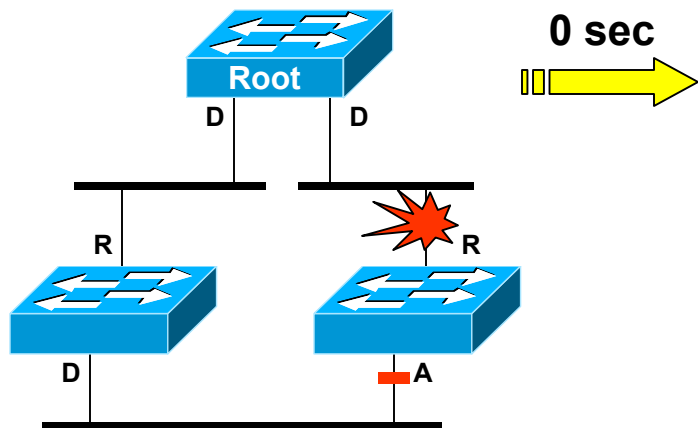
Miért lassú a STP?

- a **rossz hírek** (bridge vagy link kiesése) **lassan terjednek**
 - a rossz hírt általában csak a tárolt legjobb Config BPDU előregedése jelzi
- sokszor hamar kiderül, hogy „be kell kapcsolni” egy portot (→Listening), de
- a **portok bekapcsolása** (Listening→Forwarding) **lassú**
 - mert a teljes hálózat konzisztens állapotának elérését (worst case) időzítések biztosítják

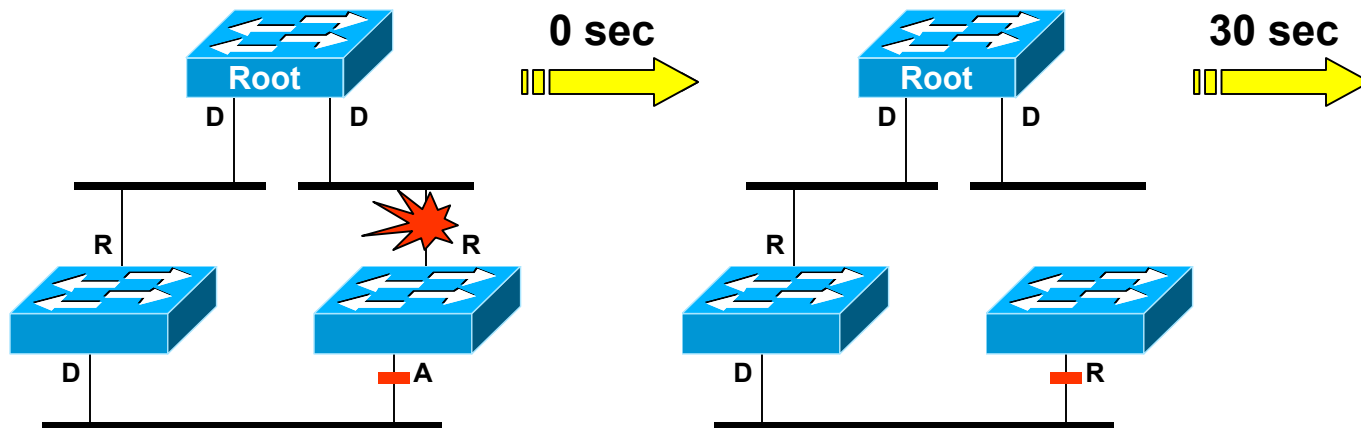
Lassú STP konvergencia – 1.



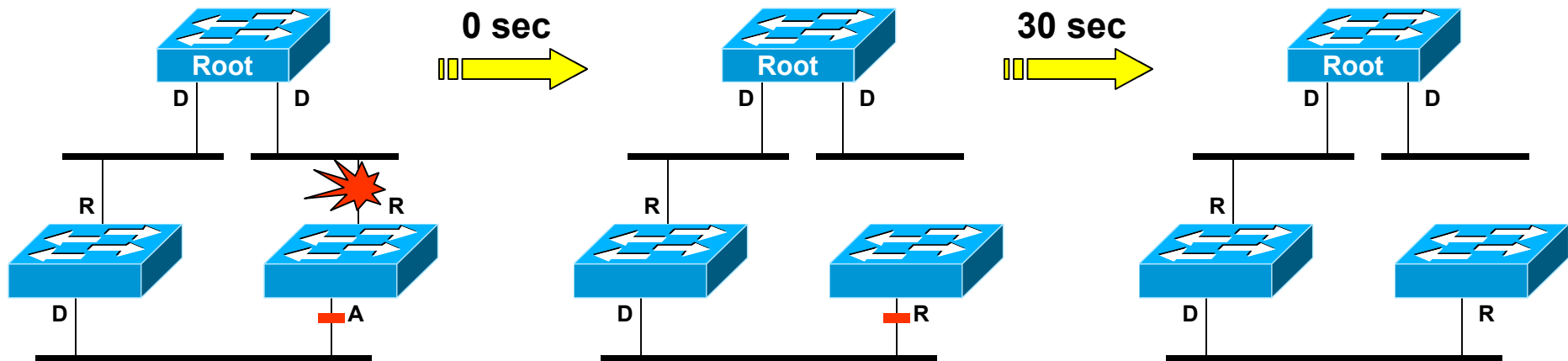
Lassú STP konvergencia – 1.



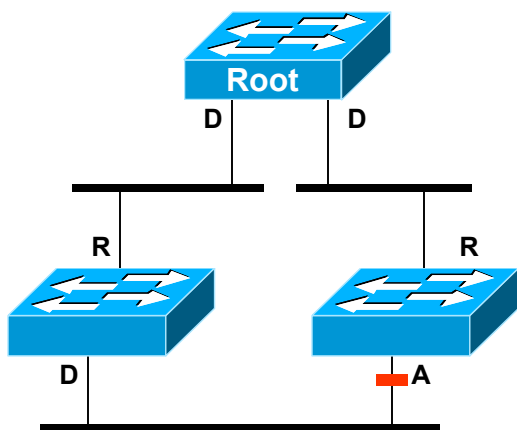
Lassú STP konvergencia – 1.



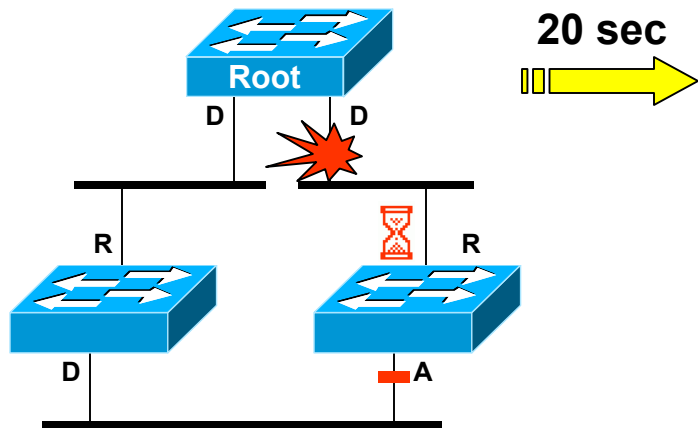
Lassú STP konvergencia – 1.



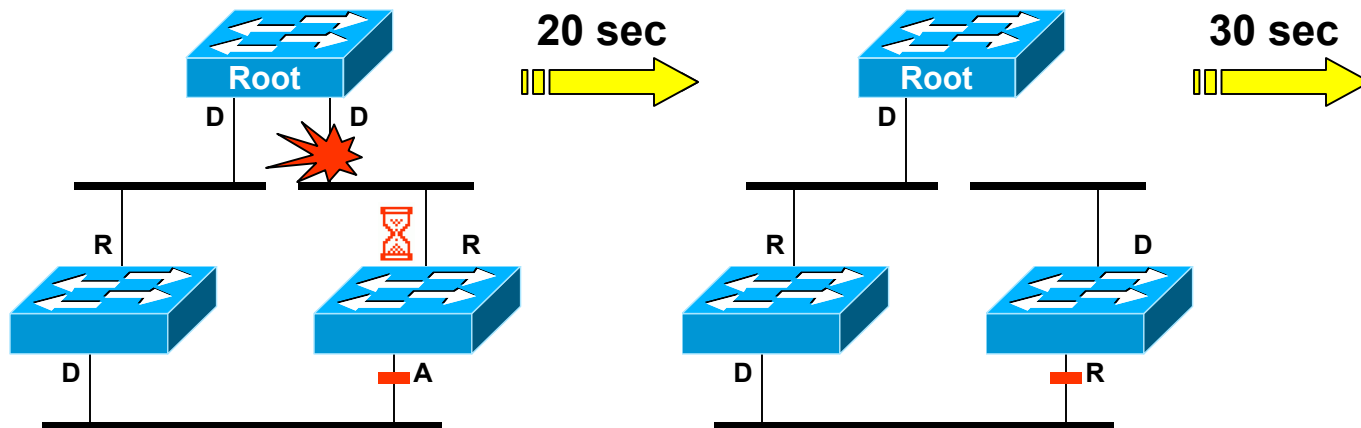
Lassú STP konvergencia – 2.



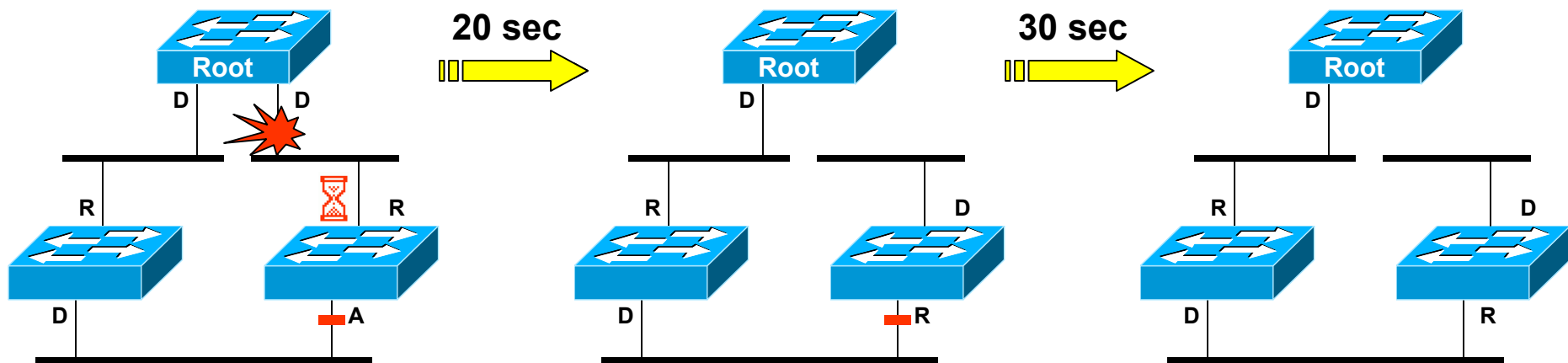
Lassú STP konvergencia – 2.



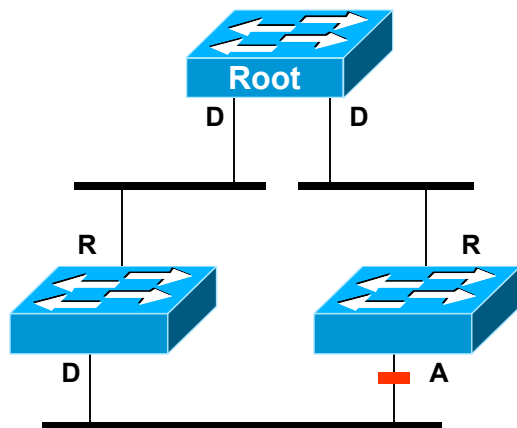
Lassú STP konvergencia – 2.



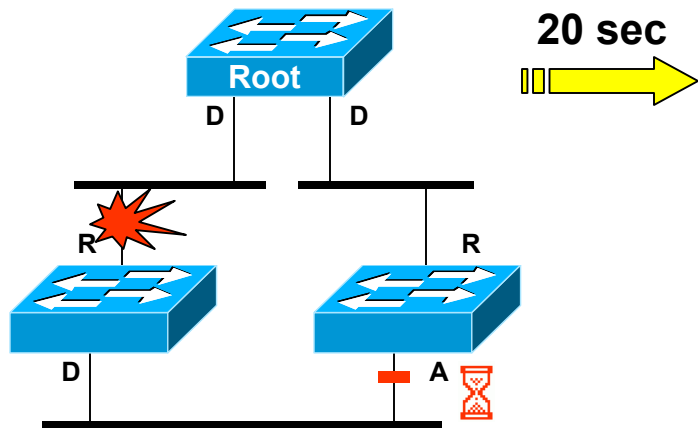
Lassú STP konvergencia – 2.



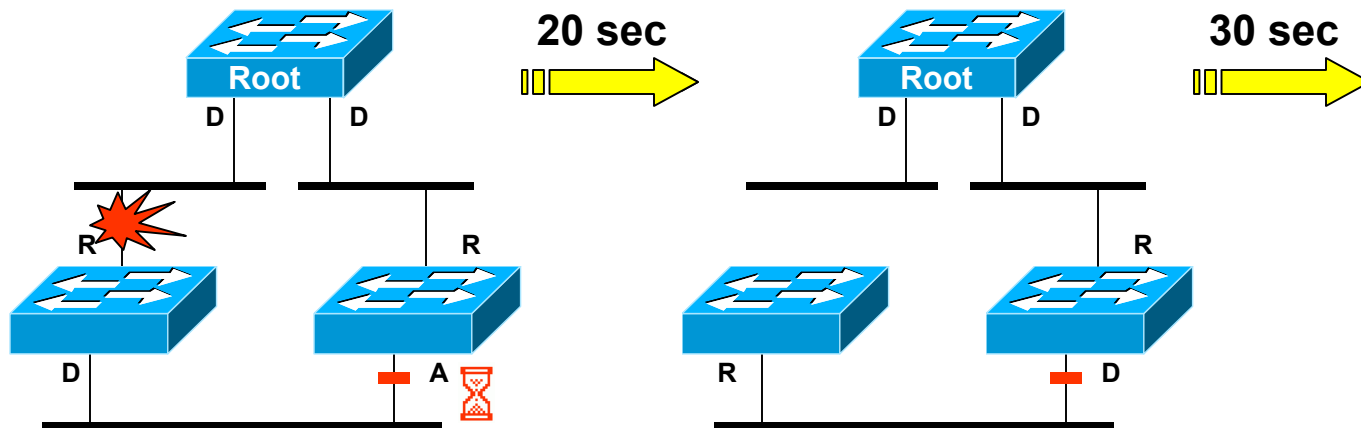
Lassú STP konvergencia – 3.



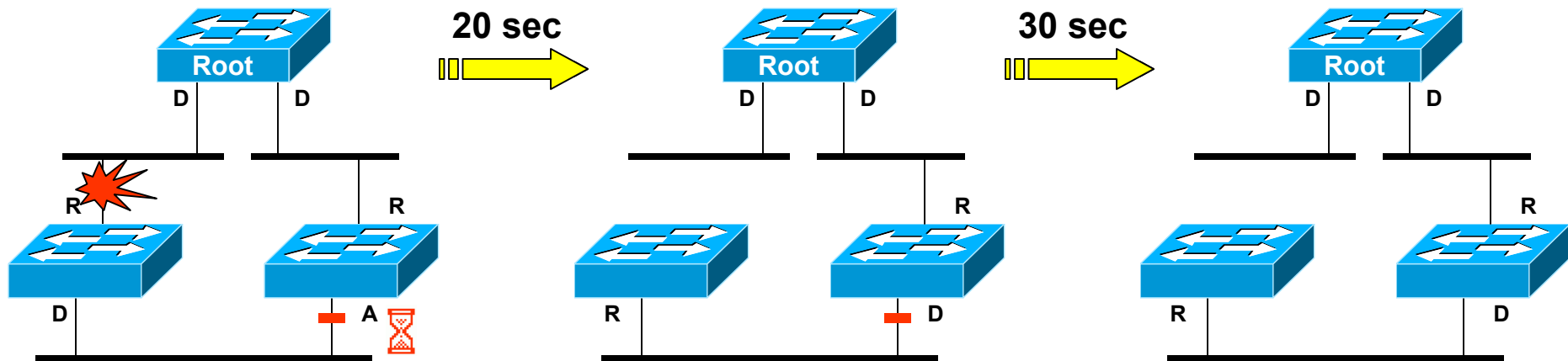
Lassú STP konvergencia – 3.



Lassú STP konvergencia – 3.



Lassú STP konvergencia – 3.



RSTP újdonságok

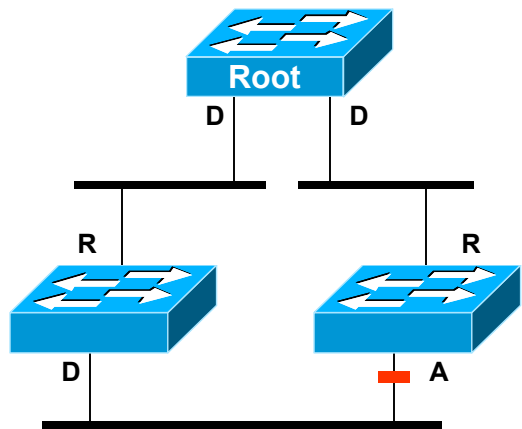
- módosított/kiegészített BPDU
- módosított BPDU kezelés
- gyors állapotváltás (Forwarding állapotba)
- új módszer a topológia változás kezelésére
- módosított port állapot és port szerep elnevezések
- a Message Age paraméter hop count szerepet kap

Config BPDU generálás

- minden Designated Bridge küld Config BPDU-t a Designated Portjain
 - Hello Time időközönként
 - nem csak akkor, ha a Root Bridge felől Config BPDU érkezik
 - és amikor változás történt a Root Porttól tanult információkban
- 3 egymás utáni Config BPDU kimaradása esetén a Designated Port elérhetetlennek minősül
 - így ugyanis már nem a teljes hálózatra kell méretezni a timeout hosszát

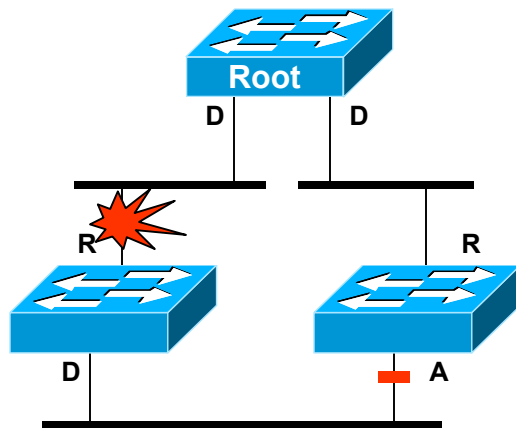
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



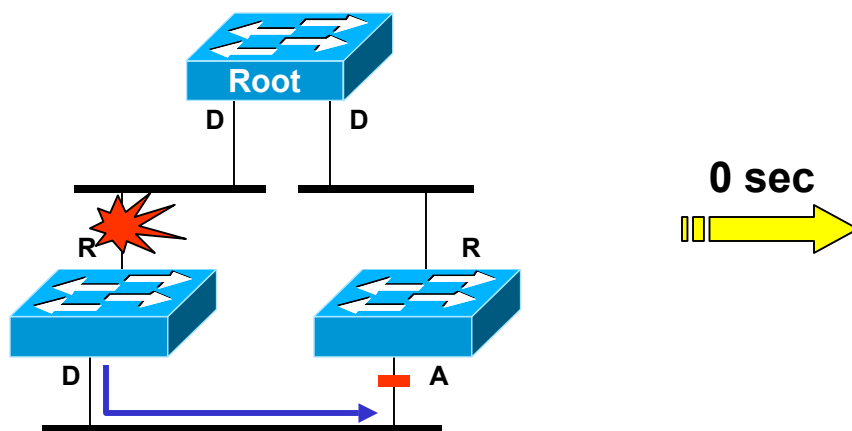
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



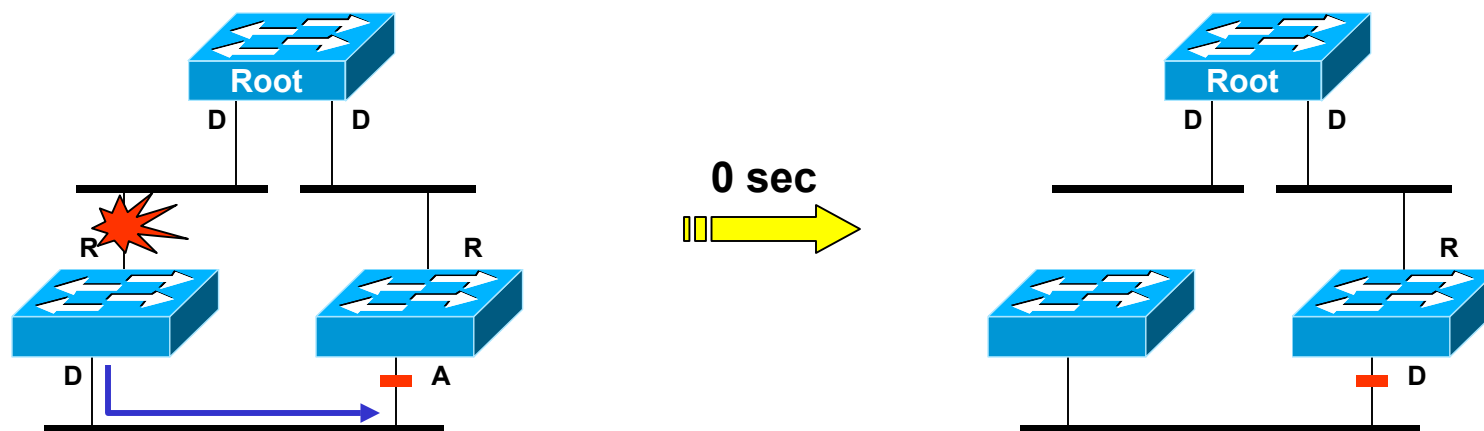
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



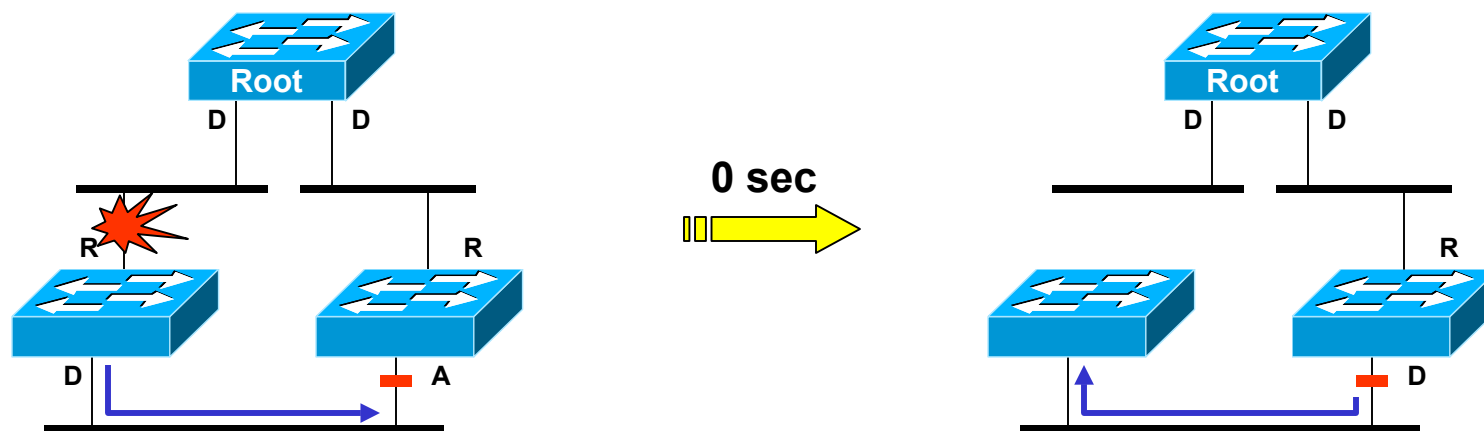
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



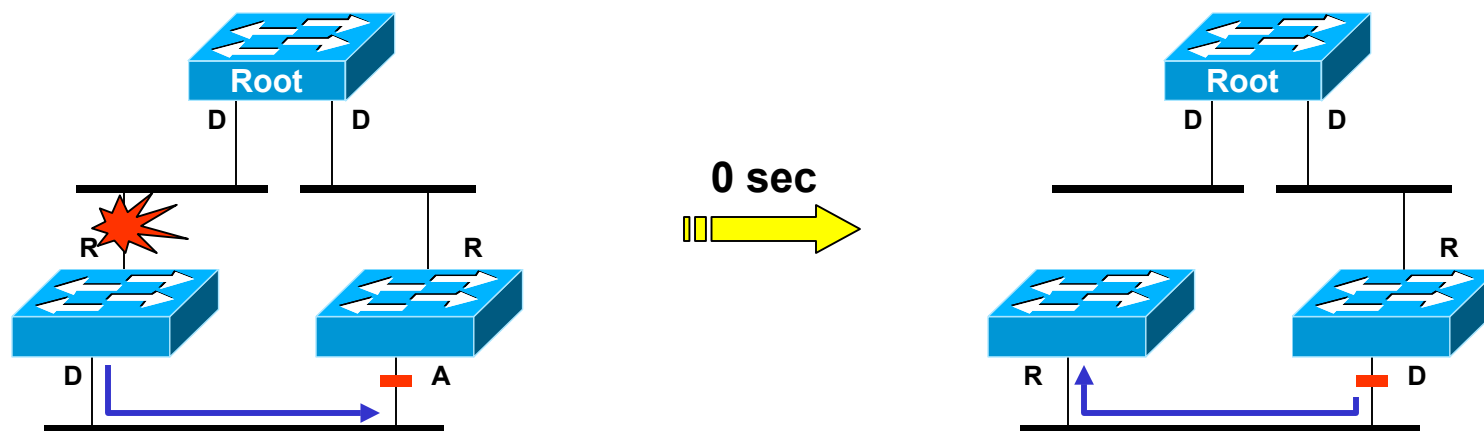
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



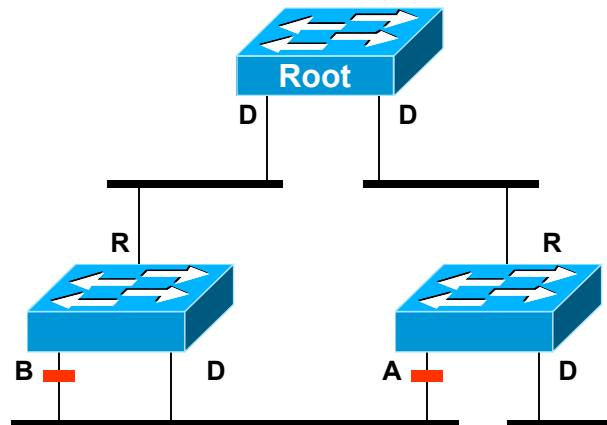
Config BPDU feldolgozás

- **Designated Porttól** a korábbinál **rosszabb információt is** elfogad a bridge
 - ez segít kibillenteni az Alternate vagy Root szerepből a szemközti portokat
 - nem kell Max Age ideig várni



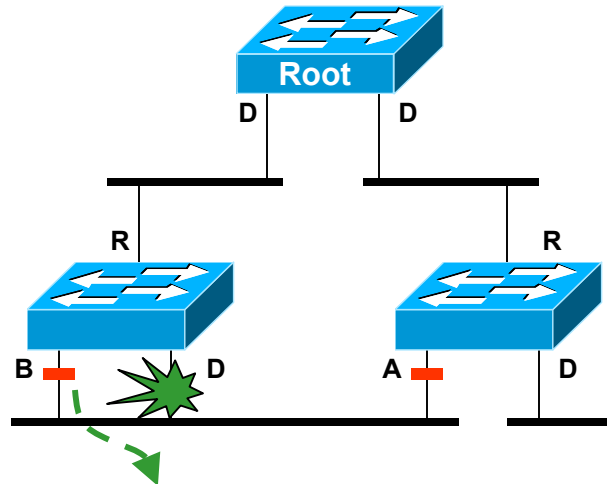
Port szerepek

- Root Port, Designated Port, Disabled Port változatlan
- **Alternate helyett**
 - **Backup Port:** tartalék irány a fa levelei felé
 - a Designated Porttal „párhuzamos” port, amikor a collision domainhez a Designated Bridge több porttal is kapcsolódik
 - a legjobb BPDU-t ez a bridge küldi, egy másik portján
 - **Alternate Port:** alternatív irány a Root Bridge felé
 - a legjobb BPDU-t egy másik bridge küldi
 - egyik sem része a feszítőfának



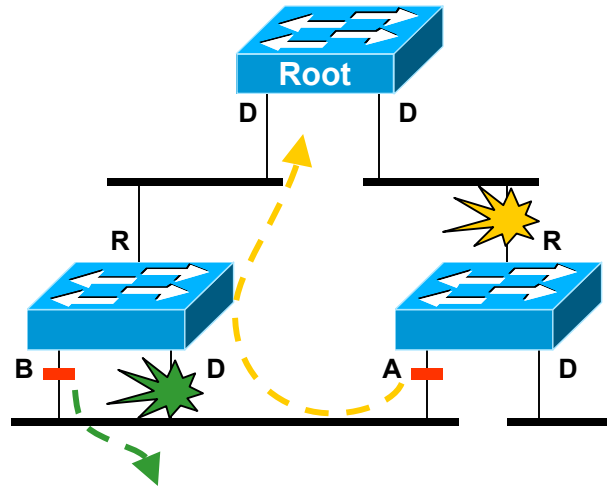
Port szerepek

- Root Port, Designated Port, Disabled Port változatlan
- **Alternate helyett**
 - **Backup Port:** tartalék irány a fa levelei felé
 - a Designated Porttal „párhuzamos” port, amikor a collision domainhez a Designated Bridge több porttal is kapcsolódik
 - a legjobb BPDU-t ez a bridge küldi, egy másik portján
 - **Alternate Port:** alternatív irány a Root Bridge felé
 - a legjobb BPDU-t egy másik bridge küldi
 - egyik sem része a feszítőfának



Port szerepek

- Root Port, Designated Port, Disabled Port változatlan
- **Alternate helyett**
 - **Backup Port:** tartalék irány a fa levelei felé
 - a Designated Porttal „párhuzamos” port, amikor a collision domainhez a Designated Bridge több porttal is kapcsolódik
 - a legjobb BPDU-t ez a bridge küldi, egy másik portján
 - **Alternate Port:** alternatív irány a Root Bridge felé
 - a legjobb BPDU-t egy másik bridge küldi
 - egyik sem része a feszítőfának



Port állapotok

- Forwarding és Learning változatlan
- Listening, Blocking és Disabled helyett Discarding

port szerep	port állapot
Root Designated	Forwarding
	Learning
	Listening
Alternate	Blocking
Disabled	Disabled

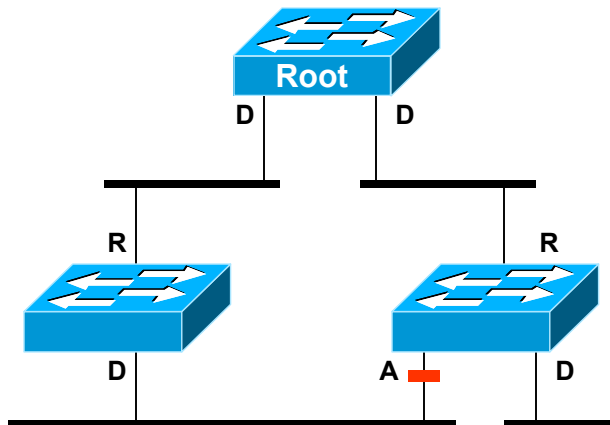
STP

port szerep	port állapot
Root Designated	Forwarding
	Learning
Alternate Backup	Discarding
Disabled	

RSTP

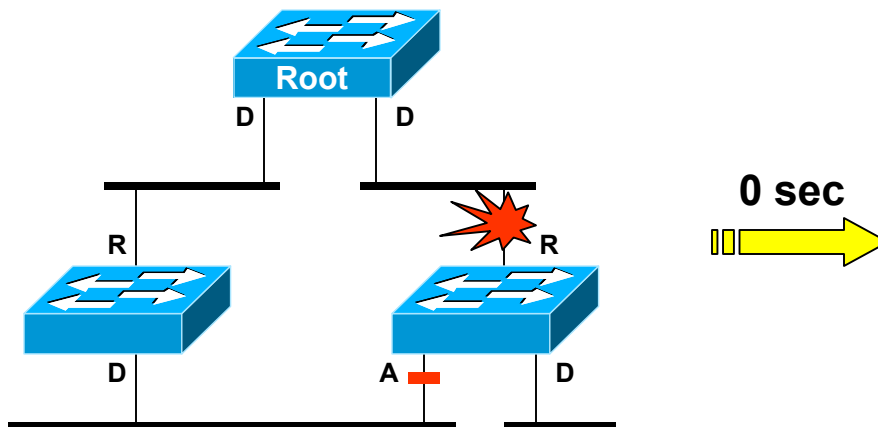
Gyors állapotváltás (→Forwarding) – 1.

- új Root Port
 - régi Root Port nem működik
 - egy Alternate Port lesz helyette az új Root Port
 - a réginél jobb Root Port keletkezett
 - jobb Config BPDU érkezett valamelyik porton, prioritások megváltoztak, stb.
 - eddig nem volt Root Port
- az új Root Port azonnal Forwarding állapotba léphet, ha a korábbi Root Port már Discarding állapotban van



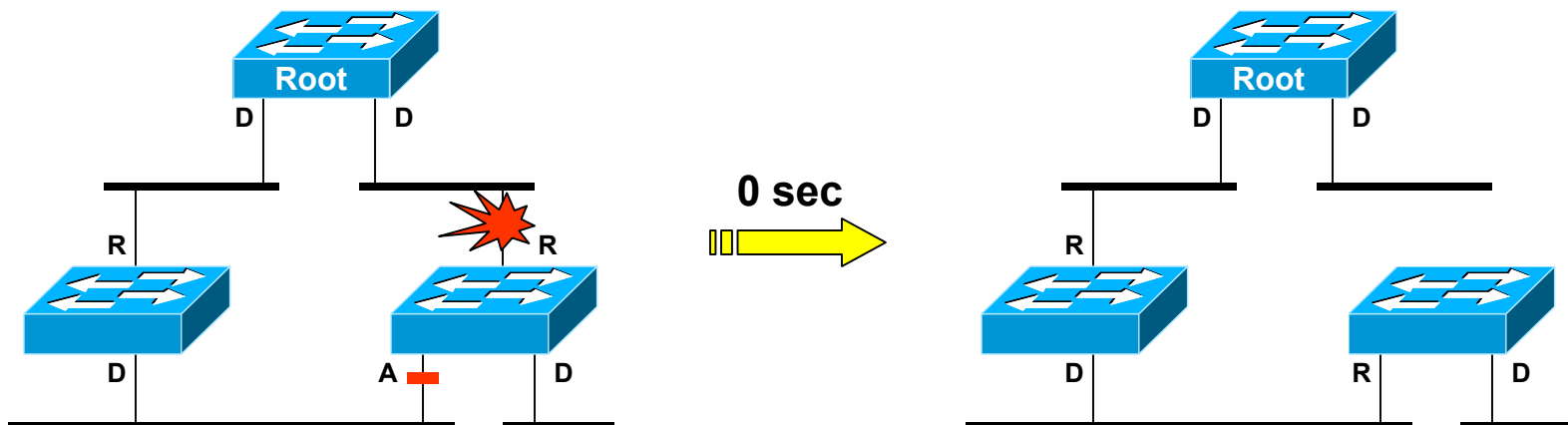
Gyors állapotváltás (→Forwarding) – 1.

- új Root Port
 - régi Root Port nem működik
 - egy Alternate Port lesz helyette az új Root Port
 - a réginél jobb Root Port keletkezett
 - jobb Config BPDU érkezett valamelyik porton, prioritások megváltoztak, stb.
 - eddig nem volt Root Port
- az új Root Port azonnal Forwarding állapotba léphet, ha a korábbi Root Port már Discarding állapotban van



Gyors állapotváltás (→Forwarding) – 1.

- új Root Port
 - régi Root Port nem működik
 - egy Alternate Port lesz helyette az új Root Port
 - a réginél jobb Root Port keletkezett
 - jobb Config BPDU érkezett valamelyik porton, prioritások megváltoztak, stb.
 - eddig nem volt Root Port
- az új Root Port azonnal Forwarding állapotba léphet, ha a korábbi Root Port már Discarding állapotban van

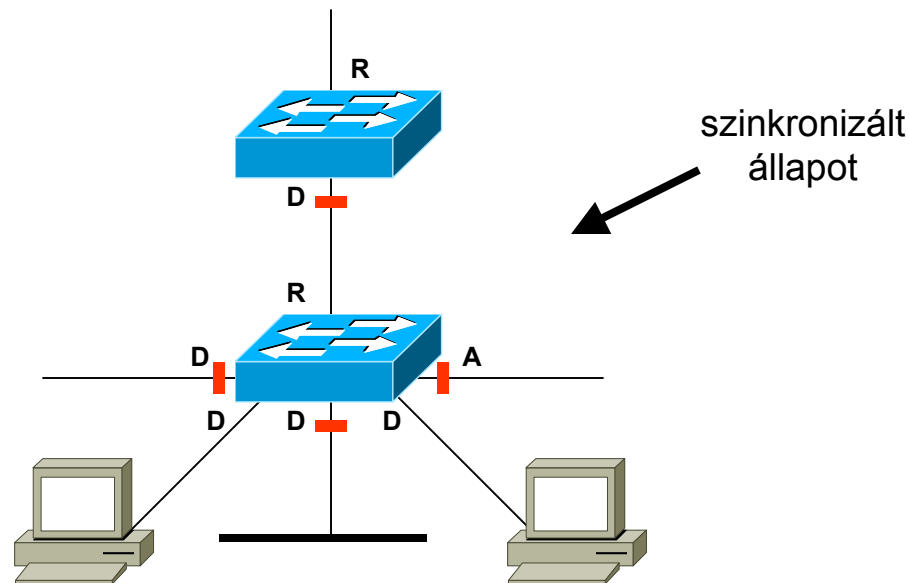


Gyors állapotváltás (→Forwarding) – 2.

- **Edge Port**: egy végberendezés kapcsolódik hozzá
 - vagy legalábbis **másik bridge** biztosan **nem kapcsolódik hozzá**
 - konfigurációs paraméter
- az Edge Port szerepe Designated lesz
- **felesleges** $2 * \text{Forward Delay}$ ideig **várni** a Forwarding állapot előtt
 - mert az Edge Porton keresztül nem képződhet hurok
- ha mégis **BPDU érkezik** egy Edge Porton, akkor a bridge **felülbírálja a konfigurált beállítást**
 - ilyenkor átmegy „normális” RSTP port működésbe
 - link-up után újra elhiszi a konfigurációs beállítást
- Edge Port funkcionalitást (vagy ahhoz hasonlót) sok gyártó implementált már a RSTP előtt is

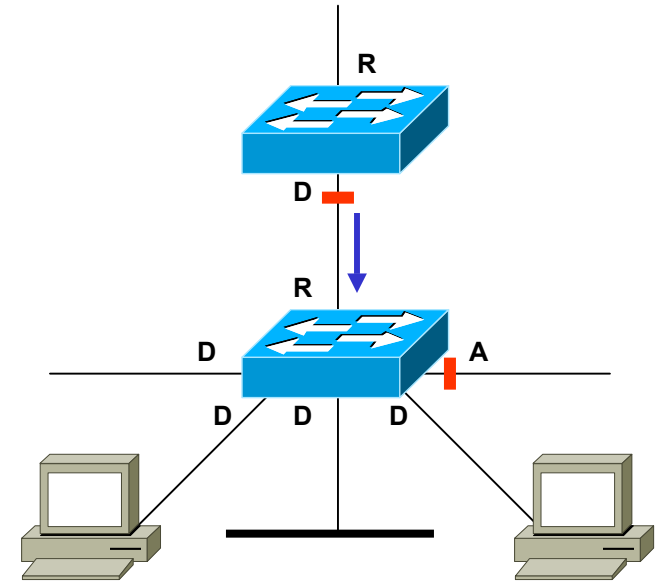
Gyors állapotváltás (→Forwarding) – 3.

- nem Edge **Designated Port** esetén **explicit handshake** után
 - csak **point-to-point** linken
 - kettőnél több bridge esetén bonyolult lenne az egyeztetés
 - és csak ha **szemben Root Port** van
 - és csak ha **a szomszéd portjai szinkronizálva** vannak
 - azaz a többi nem edge Designated Portja Discarding állapotban van
 - ilyenkor nem keletkezhet hurok



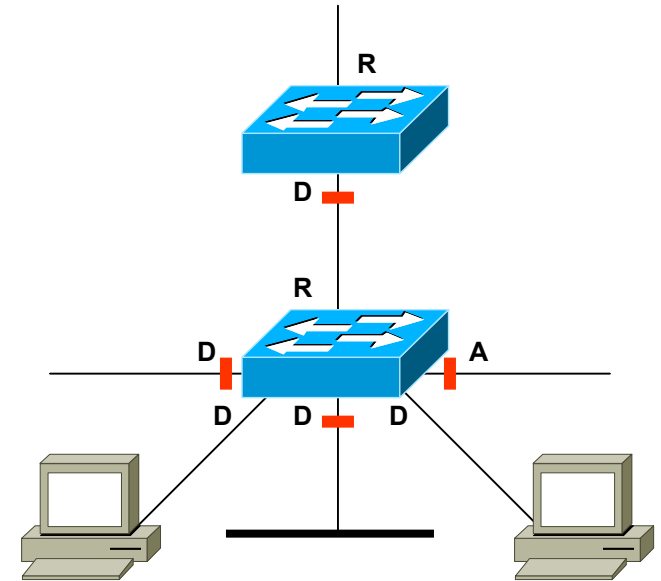
Handshake

1. a Designated Port jelzi a szomszédjának, hogy szeretne gyorsan Forwarding állapotra váltani



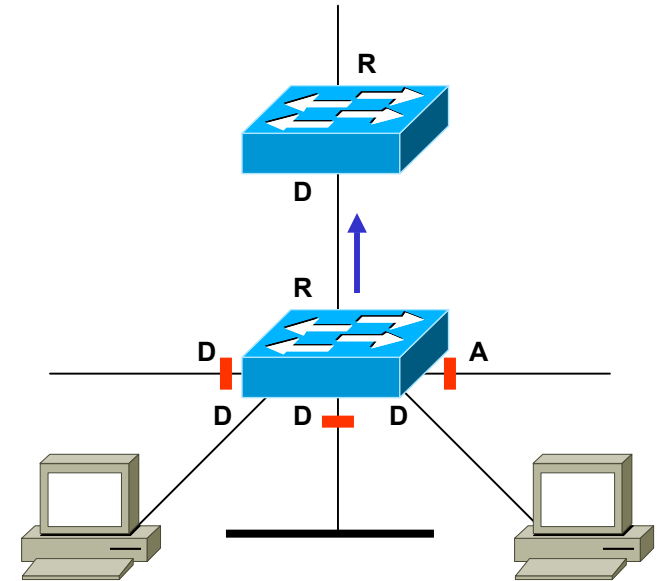
Handshake

1. a Designated Port jelzi a szomszédjának, hogy szeretne gyorsan Forwarding állapotra váltani
2. a szomszéd (ha egyetért) szinkronizálja portjait
 - az összes Designated Portját az edge portok kivételével Discarding állapotba állítja
 - ezzel elvágja a lehetséges hurkot, egy lépéssel távolabb



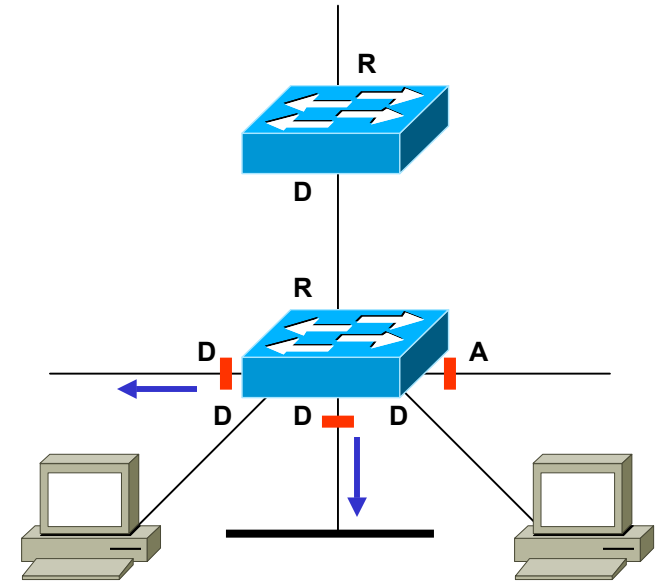
Handshake

1. a Designated Port jelzi a szomszédjának, hogy szeretne gyorsan Forwarding állapotra váltani
2. a szomszéd (ha egyetért) szinkronizálja portjait
 - az összes Designated Portját az edge portok kivételével Discarding állapotba állítja
 - ezzel elvágja a lehetséges hurkot, egy lépéssel távolabb
3. válaszol: mehet Forwarding állapotba a Designated Port

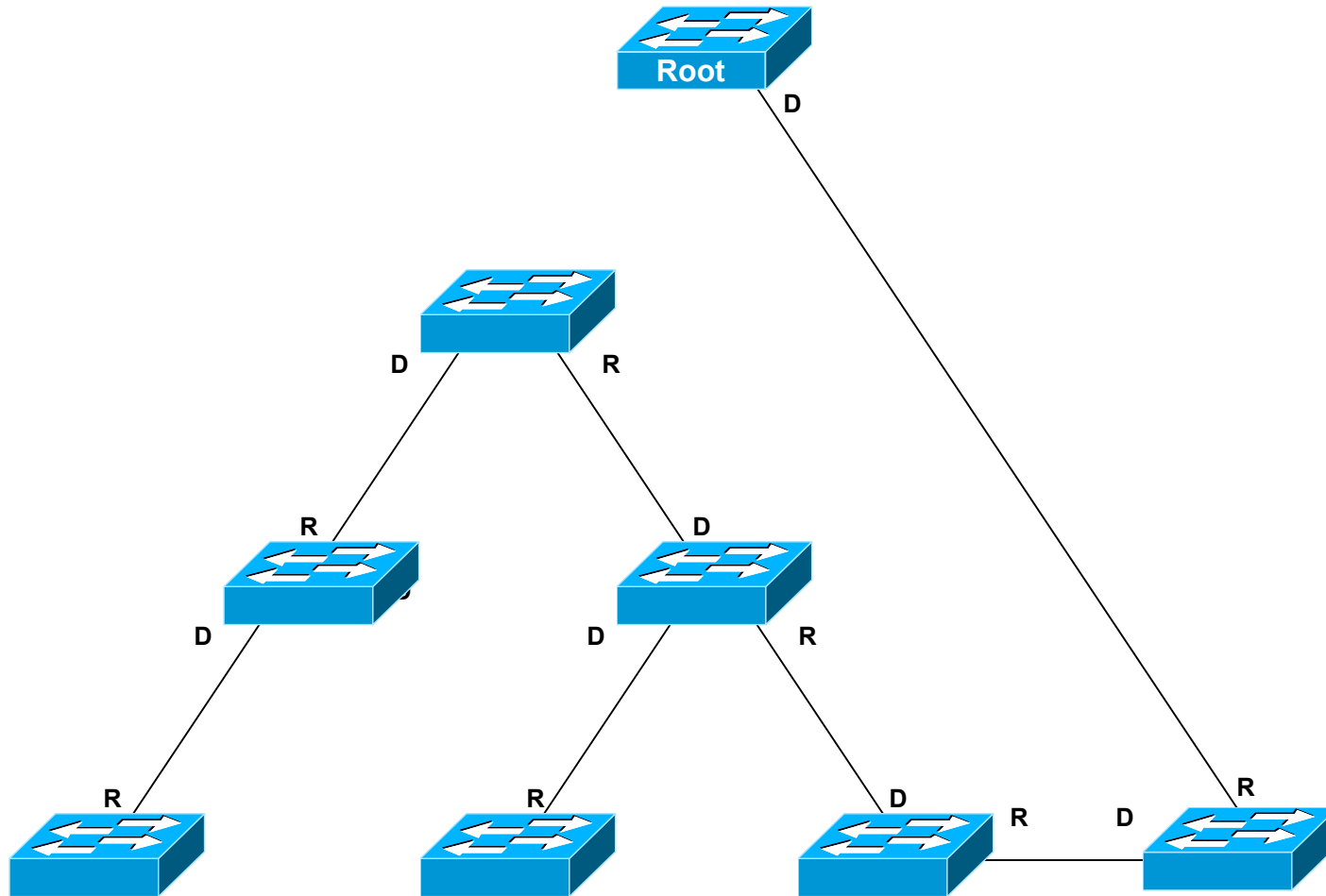


Handshake

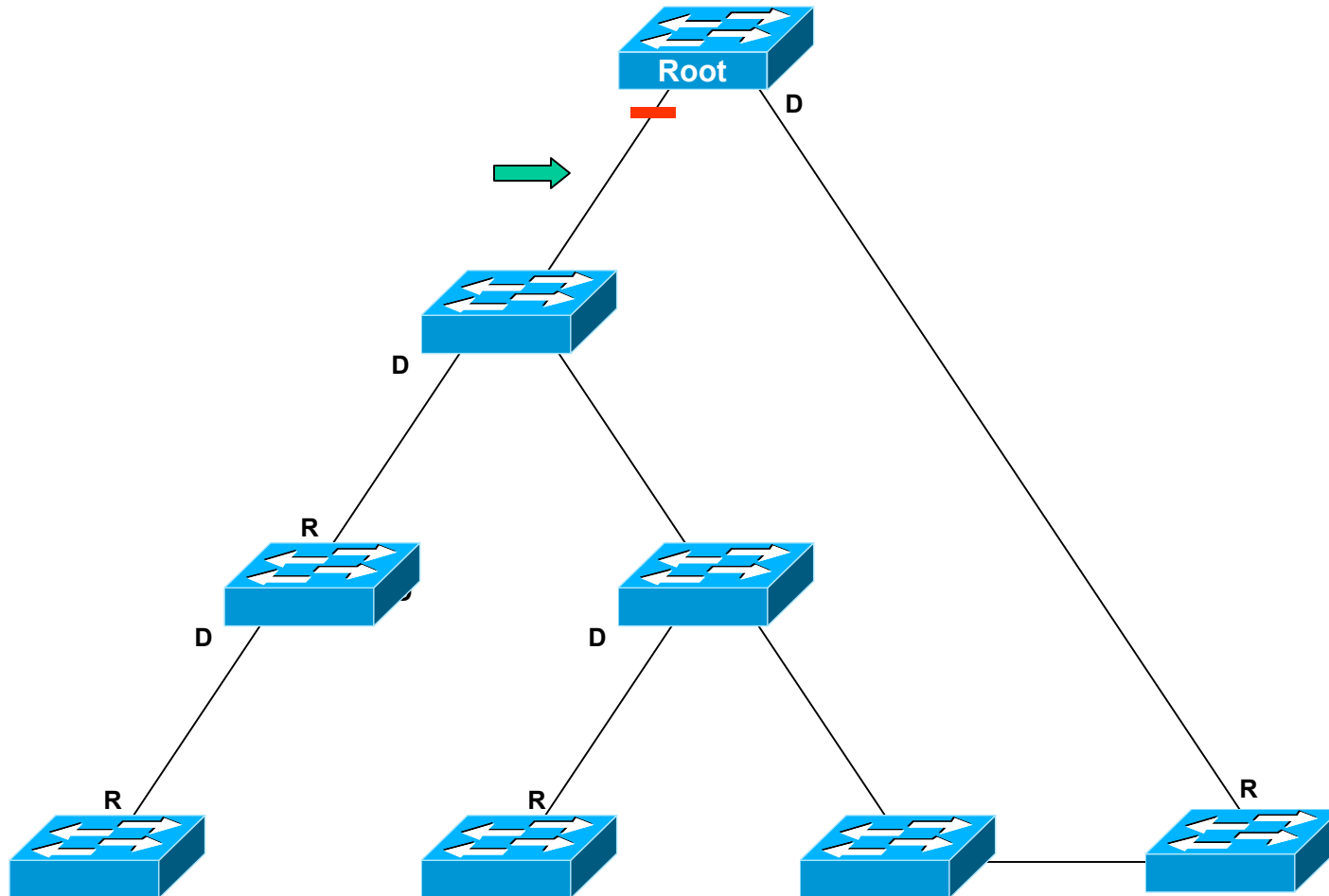
1. a Designated Port jelzi a szomszédjának, hogy szeretne gyorsan Forwarding állapotra váltani
2. a szomszéd (ha egyetért) szinkronizálja portjait
 - az összes Designated Portját az edge portok kivételével Discarding állapotba állítja
 - ezzel elvágja a lehetséges hurkot, egy lépéssel távolabb
3. válaszol: mehet Forwarding állapotba a Designated Port
 - a szomszéd hasonlóan jár el a saját lekapcsolt Designated Portjain
 - így **tolódik a gráf vágata** a végleges helyéig ill. a hálózat széléig



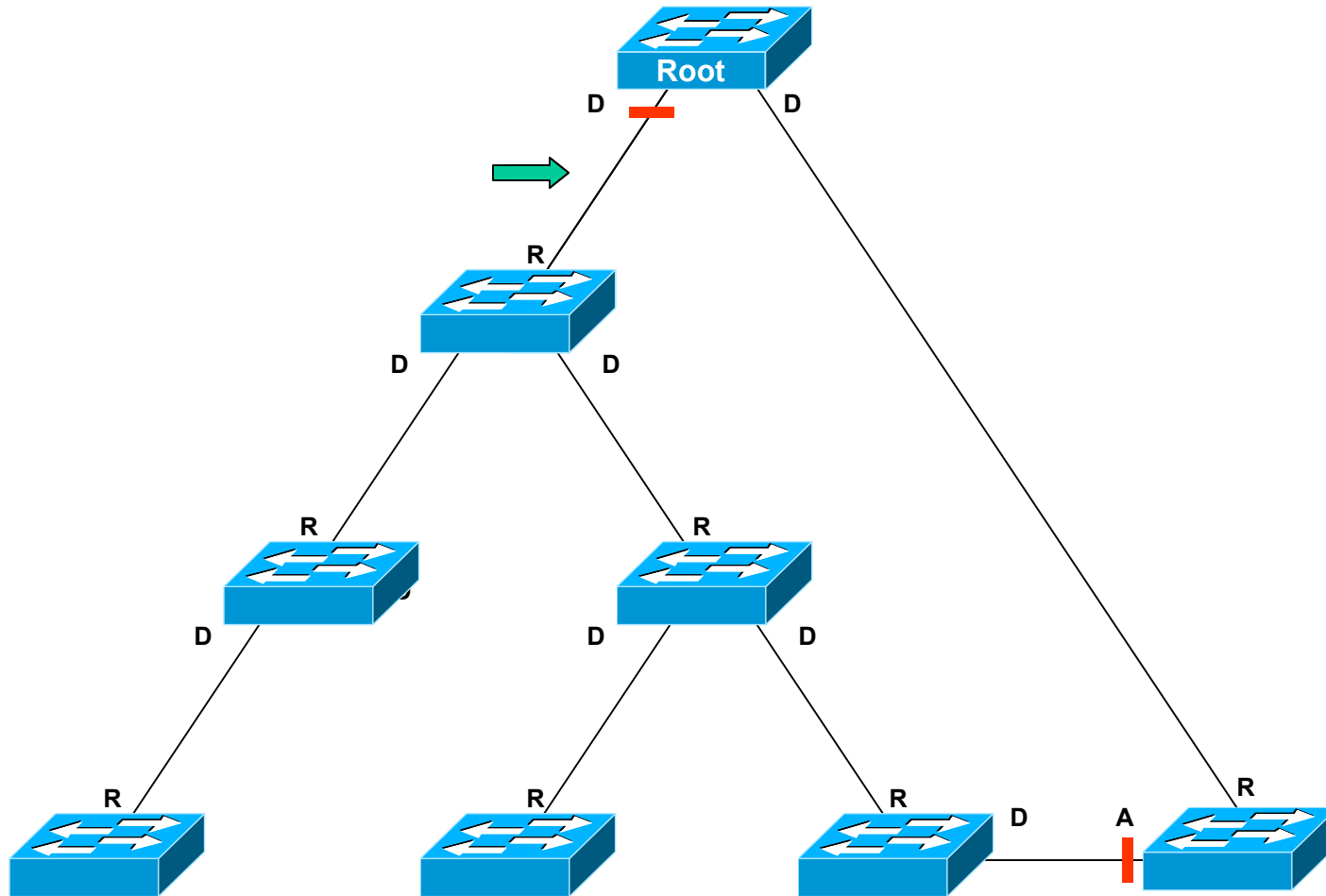
Szinkronizálási hullám



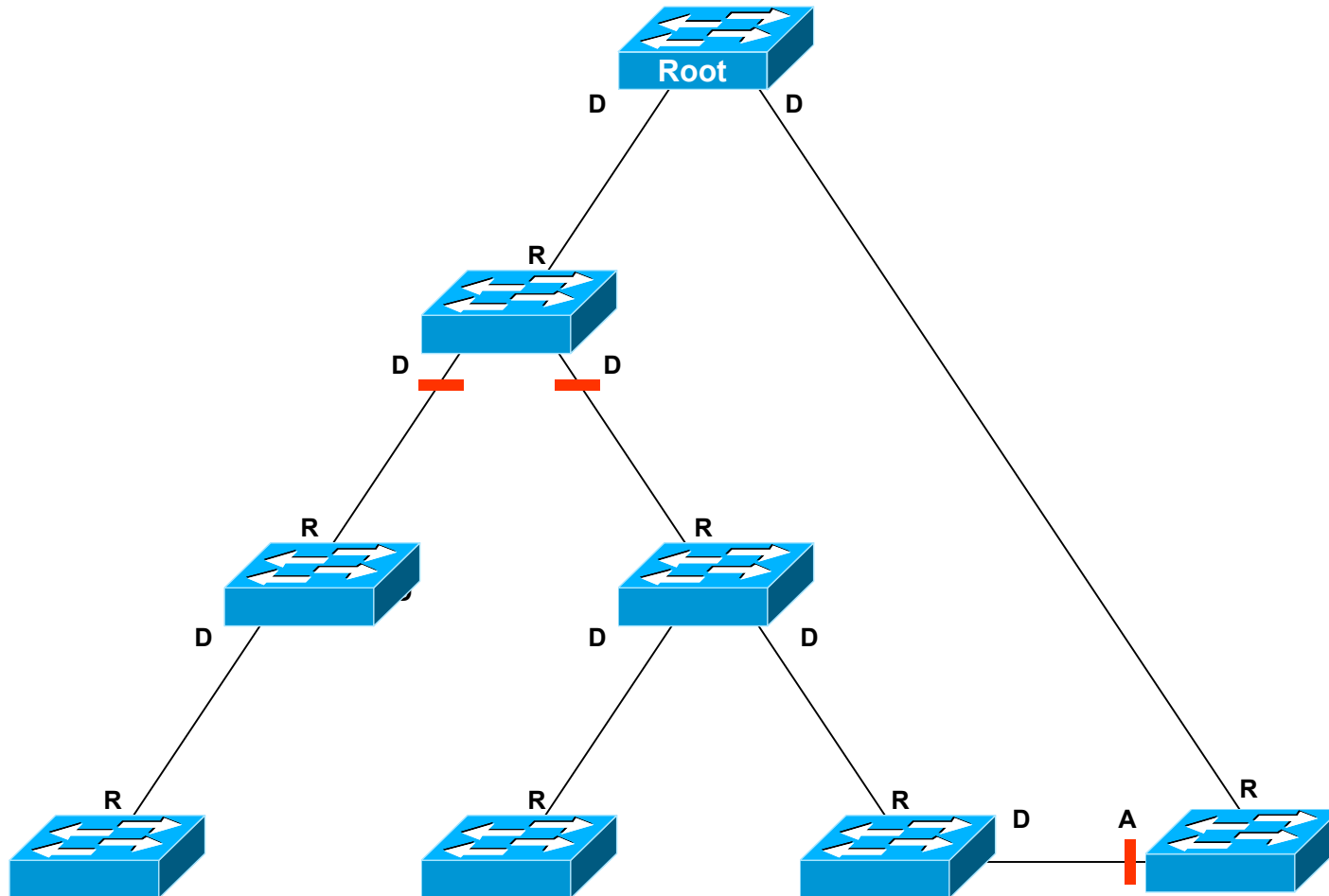
Szinkronizálási hullám



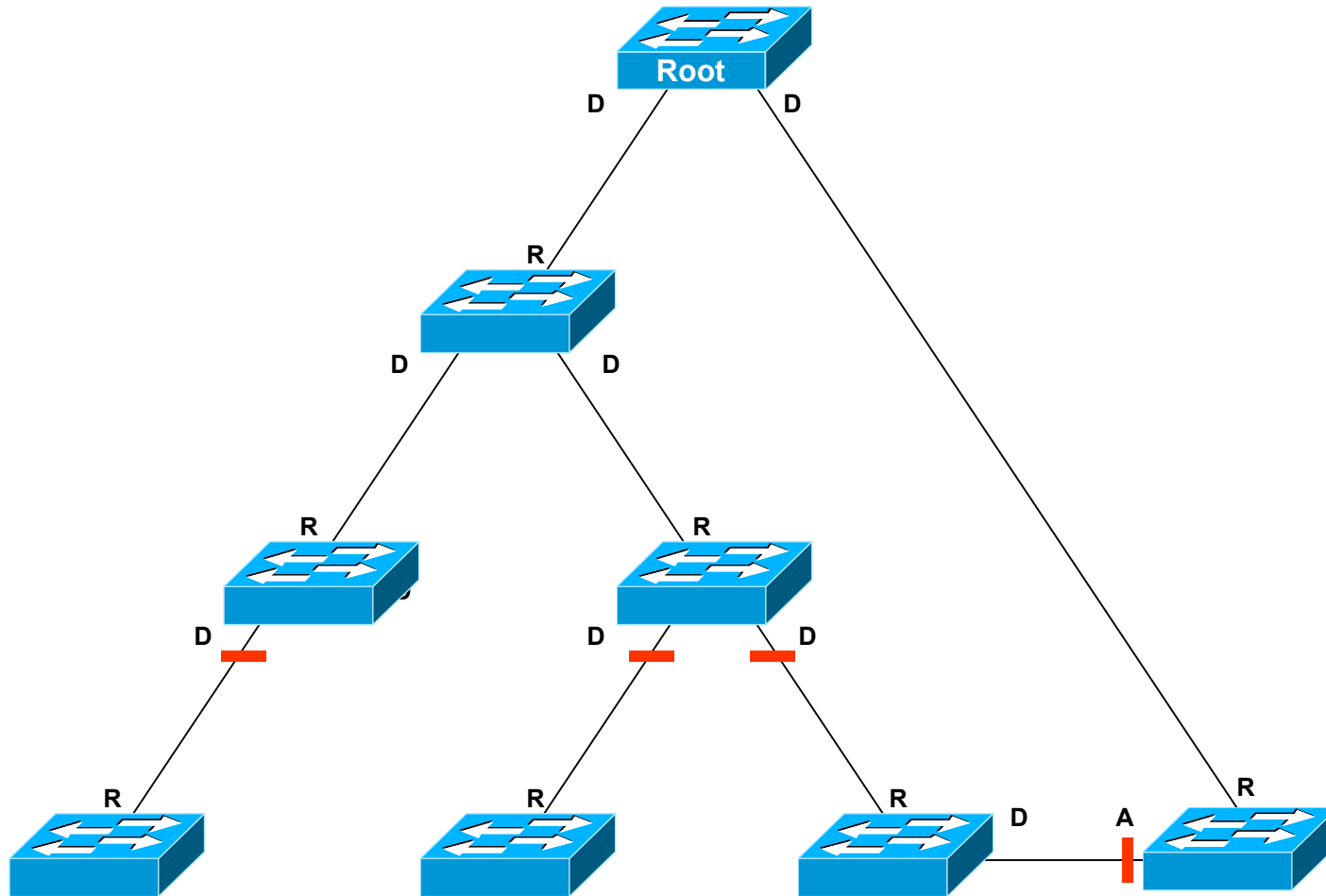
Szinkronizálási hullám



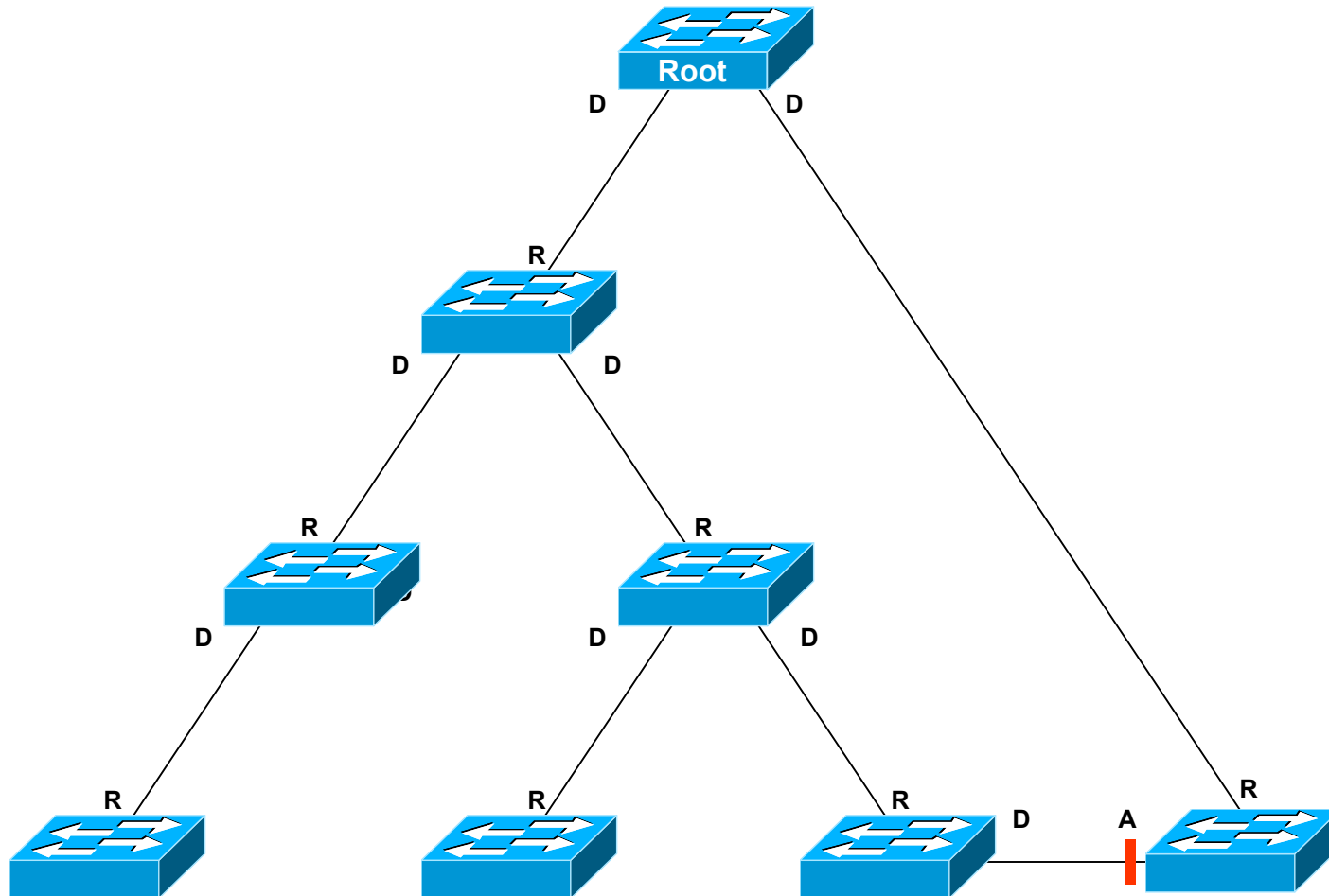
Szinkronizálási hullám



Szinkronizálási hullám



Szinkronizálási hullám



RST BPDU

- **Version 1 Length:** 1-es protokollból
 - Extended STP, IEEE 802.1G



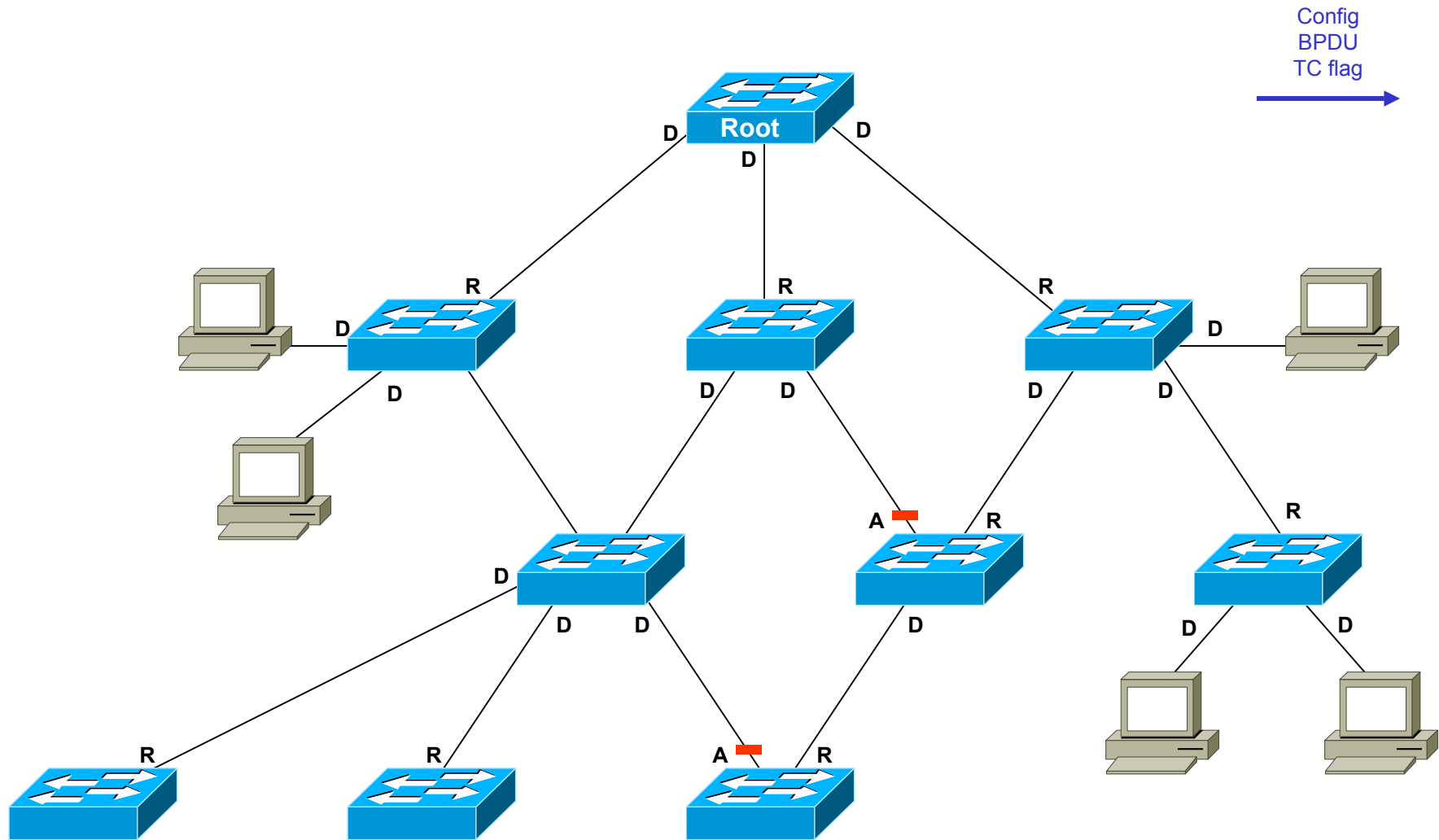
Protocol ID: 0	2
Version: 2 (RST) ←	1
Type: 2 ←	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2
Version 1 Length: 0	1

- **Proposal:** Designated Port állítja be, ha szeretne Forwarding állapotba lépni
- **Agreement:** Root Port állítja be válaszul a Proposal flagre, amikor szinkronizálta a többi portját
- **Port Role:** a küldő port aktuális szerepe
- Learning/Forwarding: ???

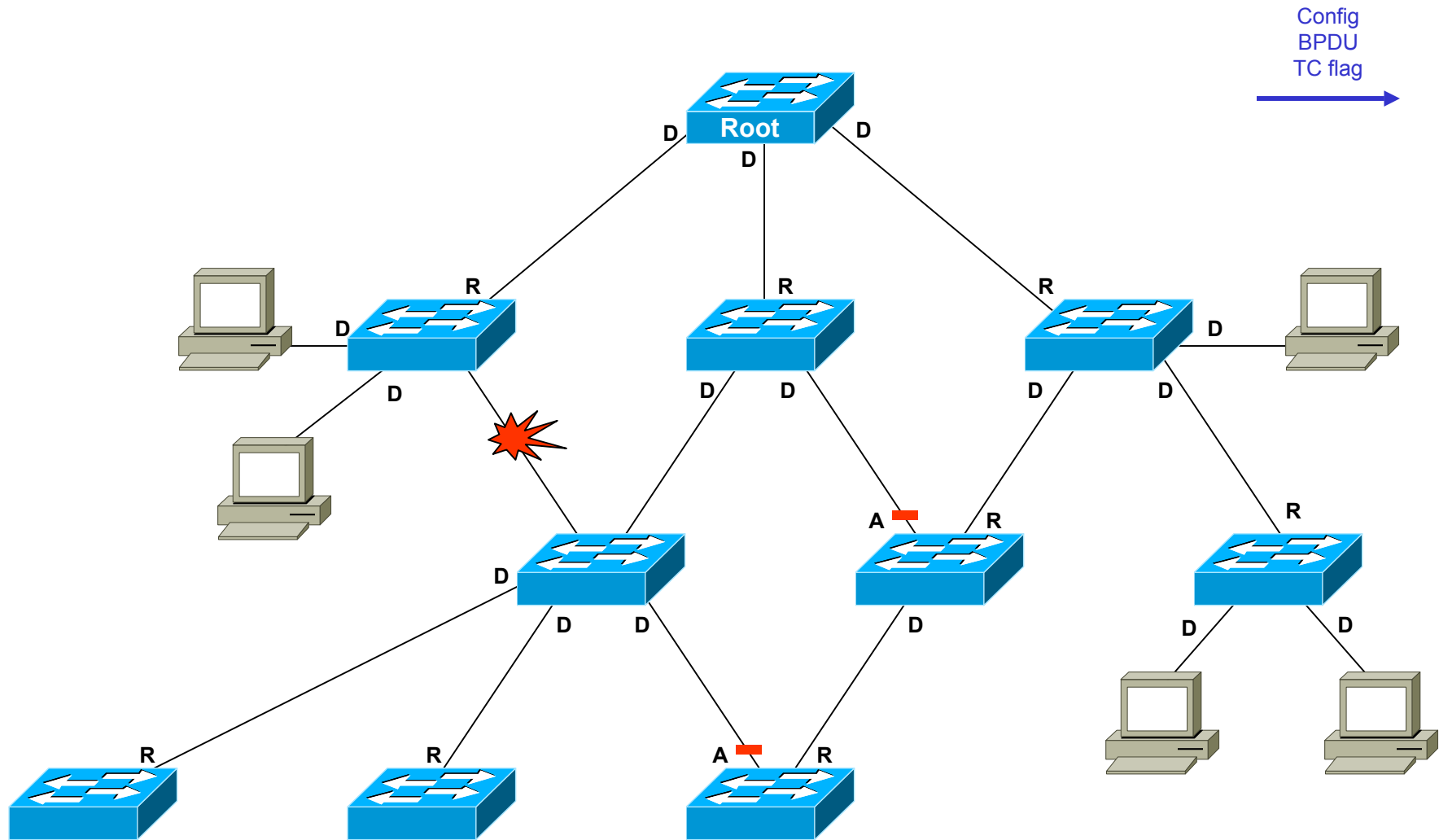
Topológia változás kezelése

- ha egy port **Forwarding állapotba kerül**, akkor a bridge
 - címbejegyzéseit **törli a többi porton**
 - ezen a porton **értesíti a szomszédait**
 - BPDU-t küld beállított TC bittel
 - aki ilyen BPDU-t kap, az
 - törli a címbejegyzéseit a többi portján
 - továbbküldi a TC jelzést az összes többi Forwarding állapotú portján
- **Edge Port** címbejegyzései **változatlanok** maradhatnak
- a **feszítőfából kikerülő** port címbejegyzéseit is **törölni** kell
- a TC bit használata megváltozott
 - a Forwarding állapotba kerülő porttól a gyökéren keresztül a fa leveleiig viszi el a változás hírért
 - a bridge „alatti” részébe nem megy jelzés
 - mivel ott helyesek maradnak a címbejegyzések

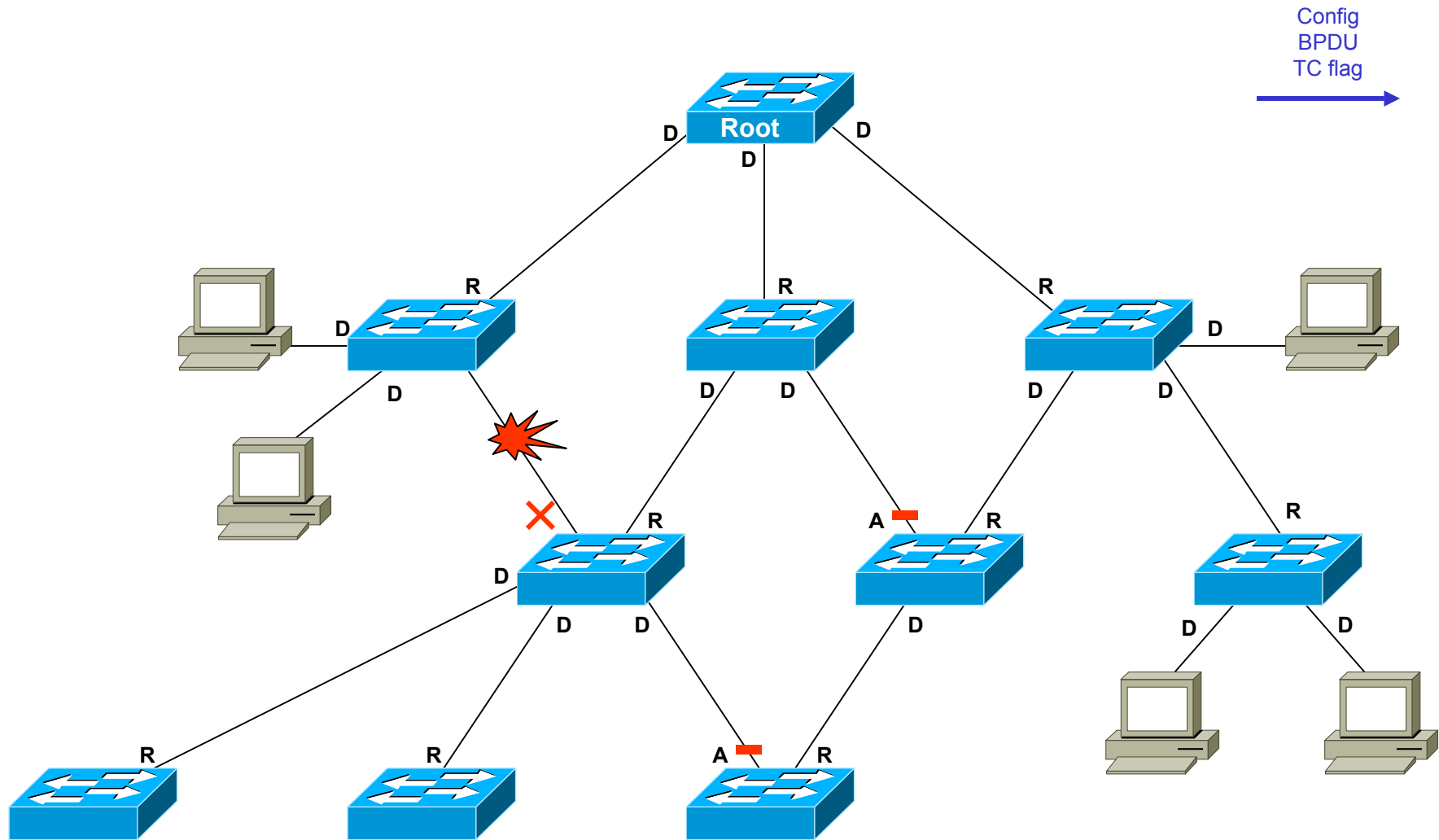
Topológia változás példa



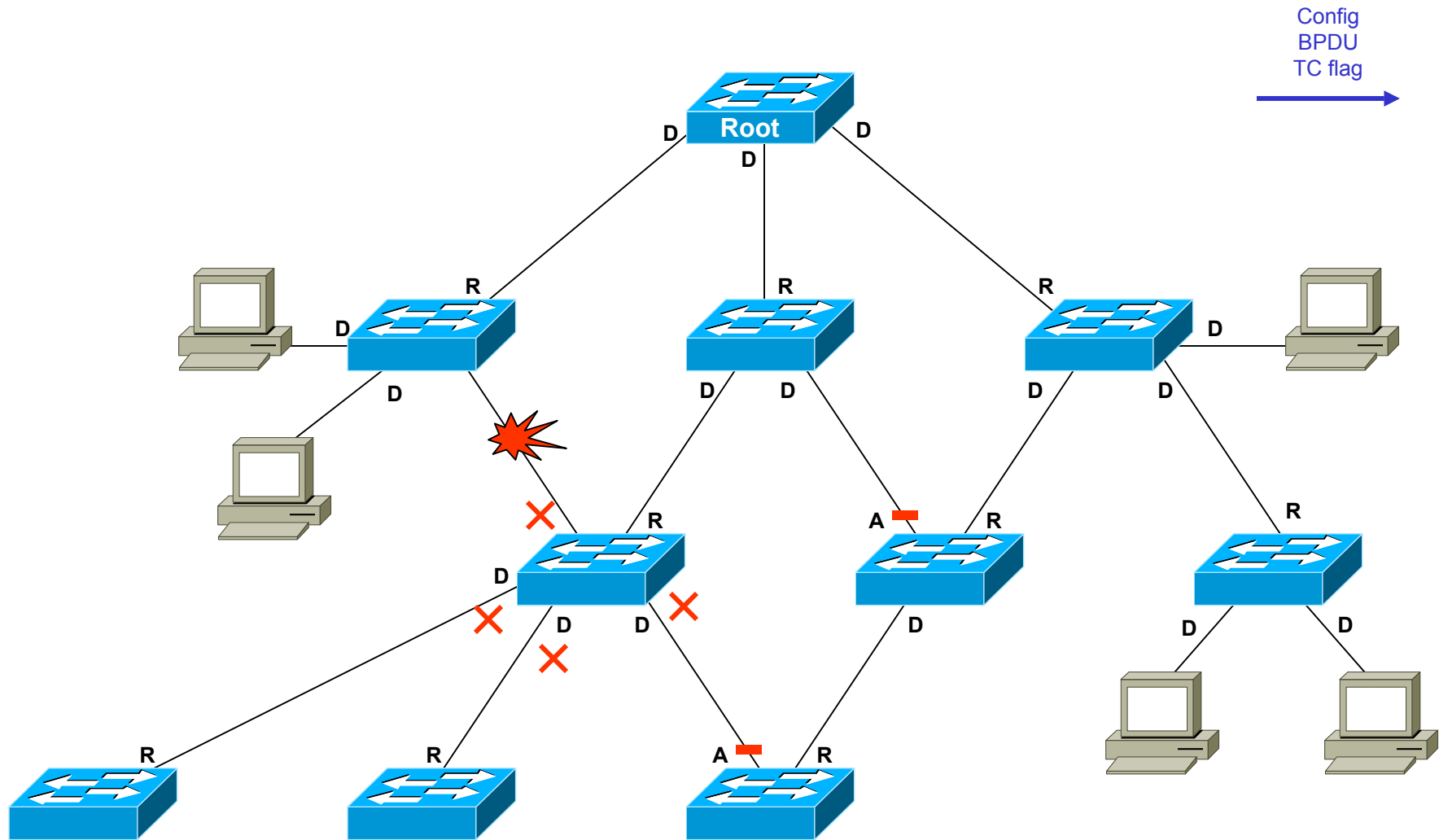
Topológia változás példa



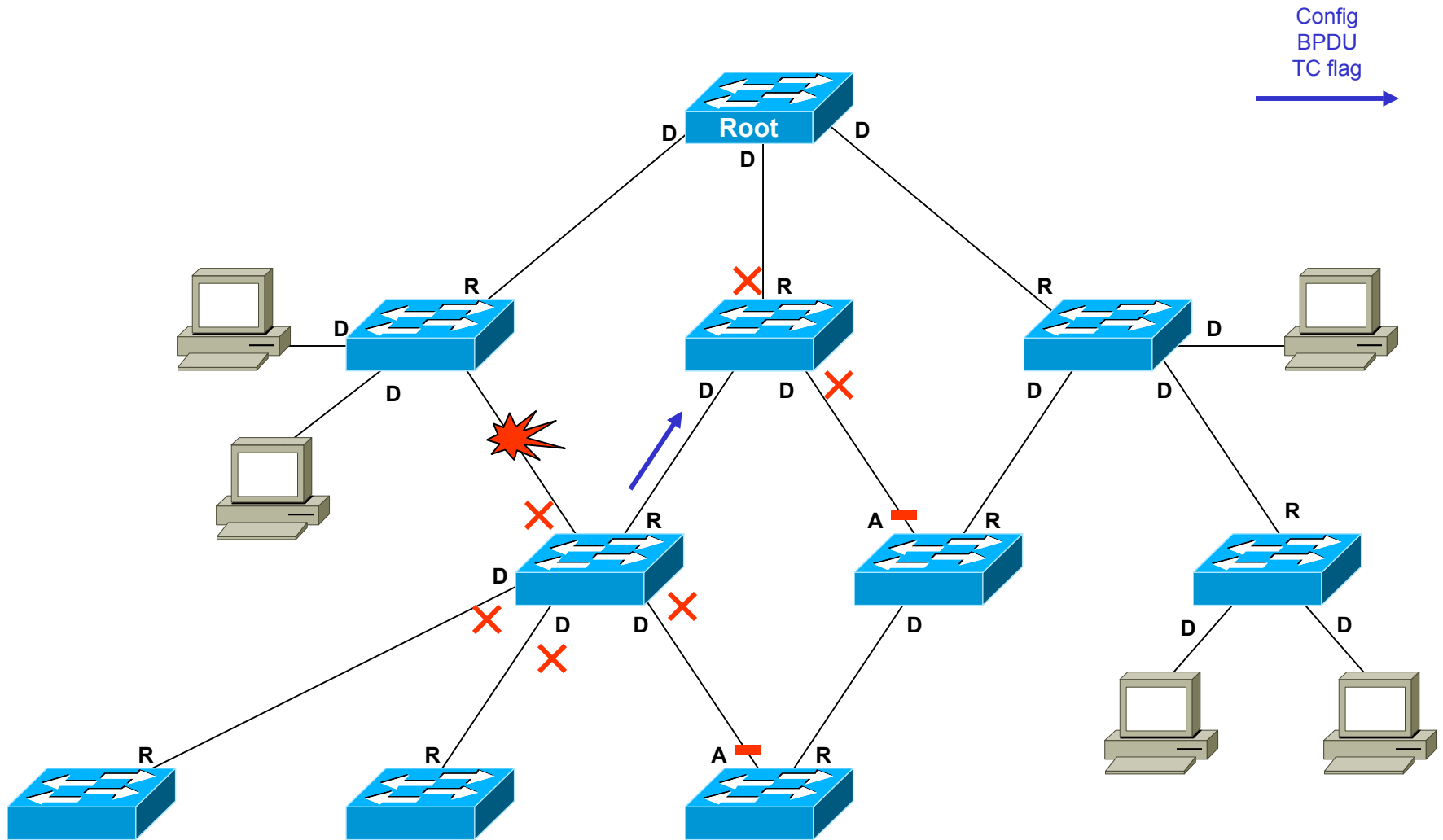
Topológia változás példa



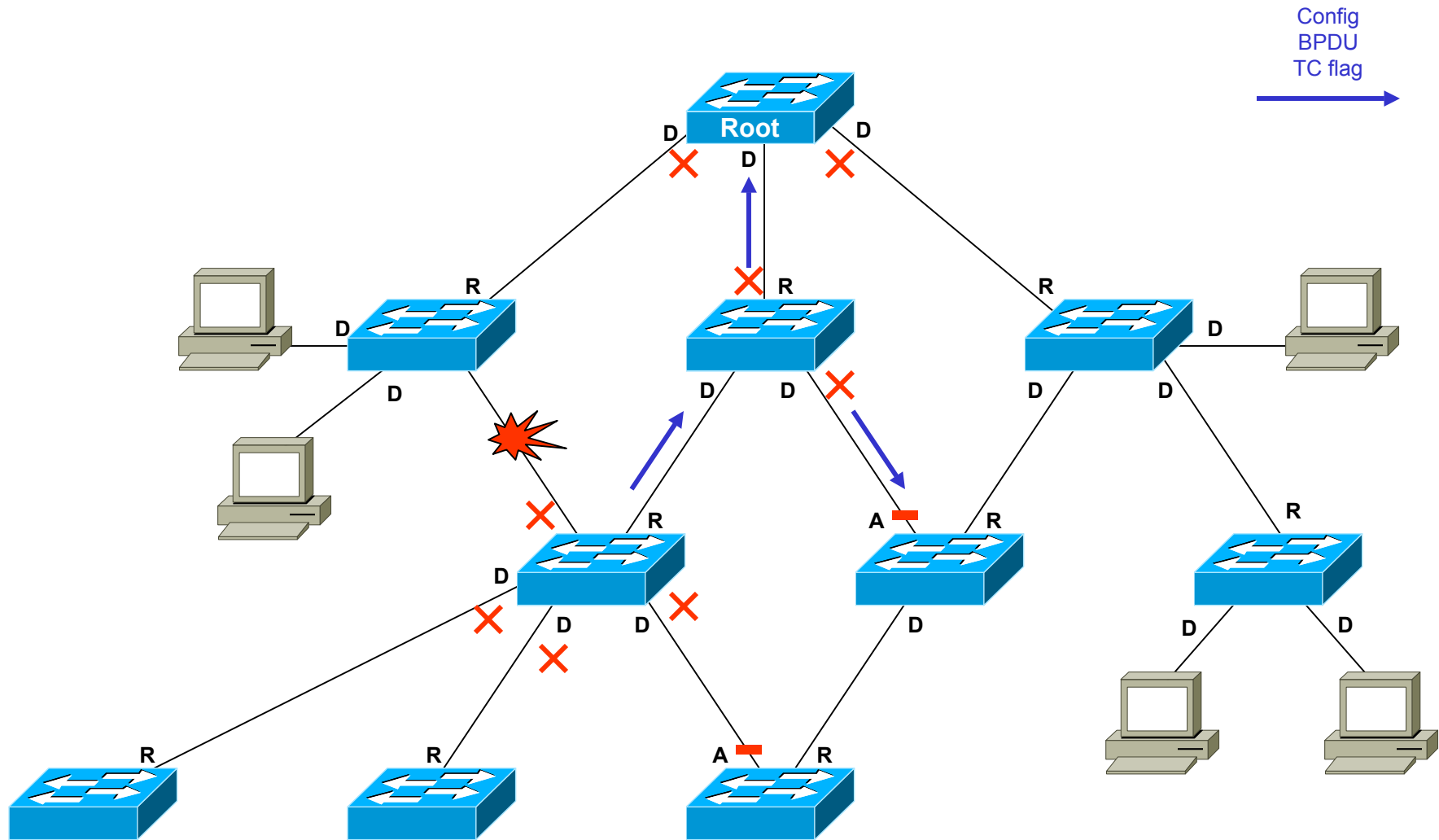
Topológia változás példa



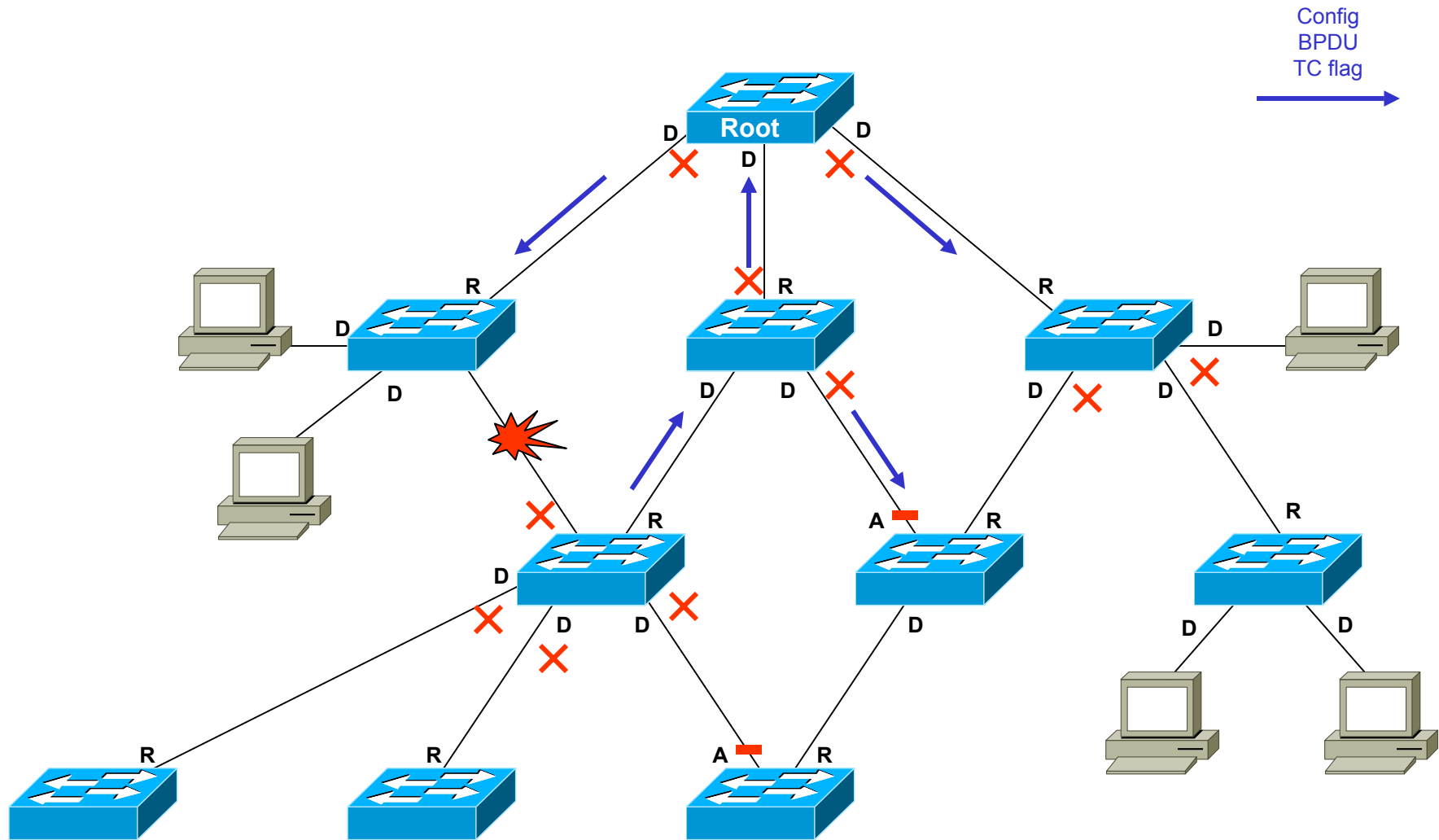
Topológia változás példa



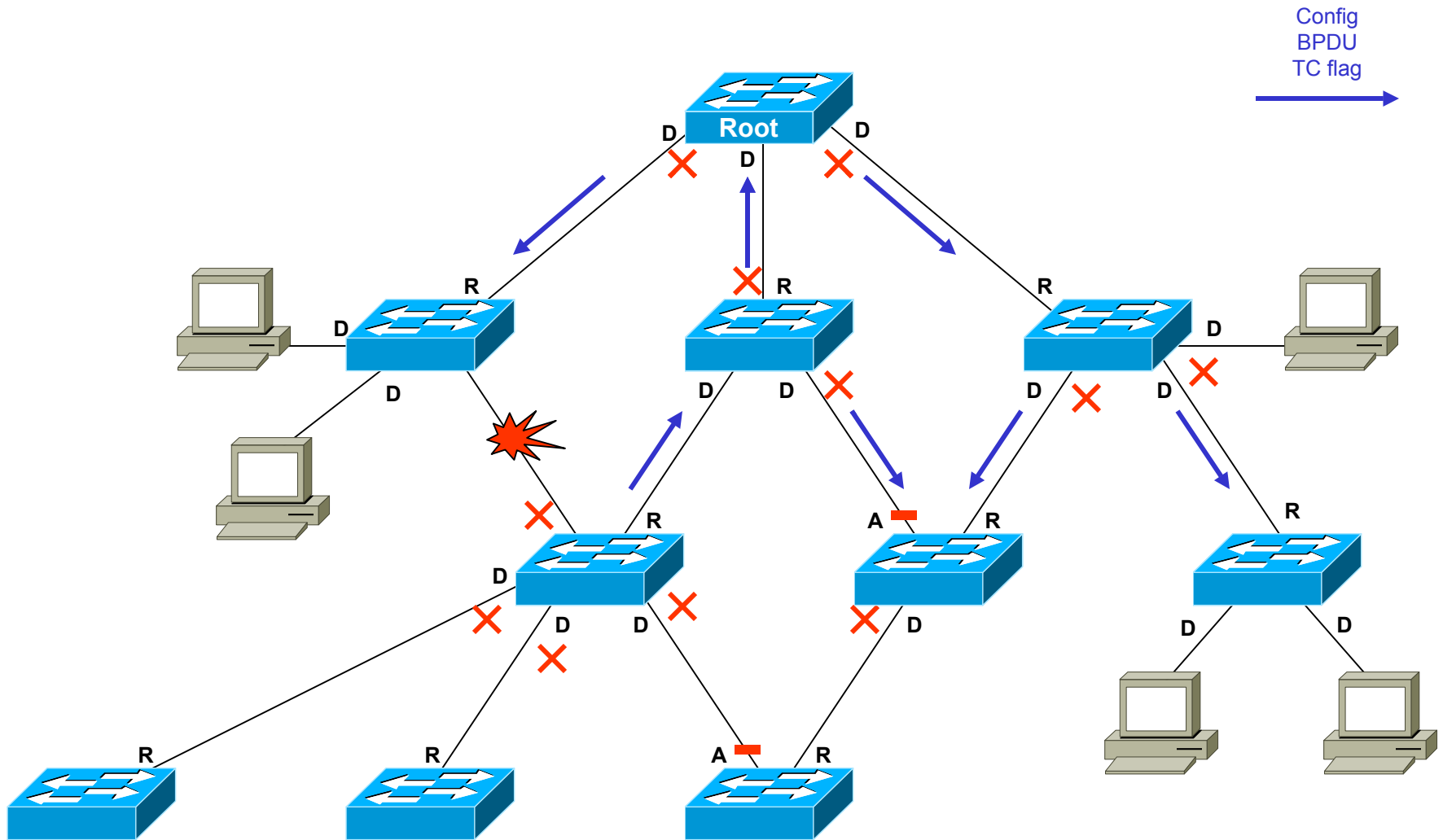
Topológia változás példa



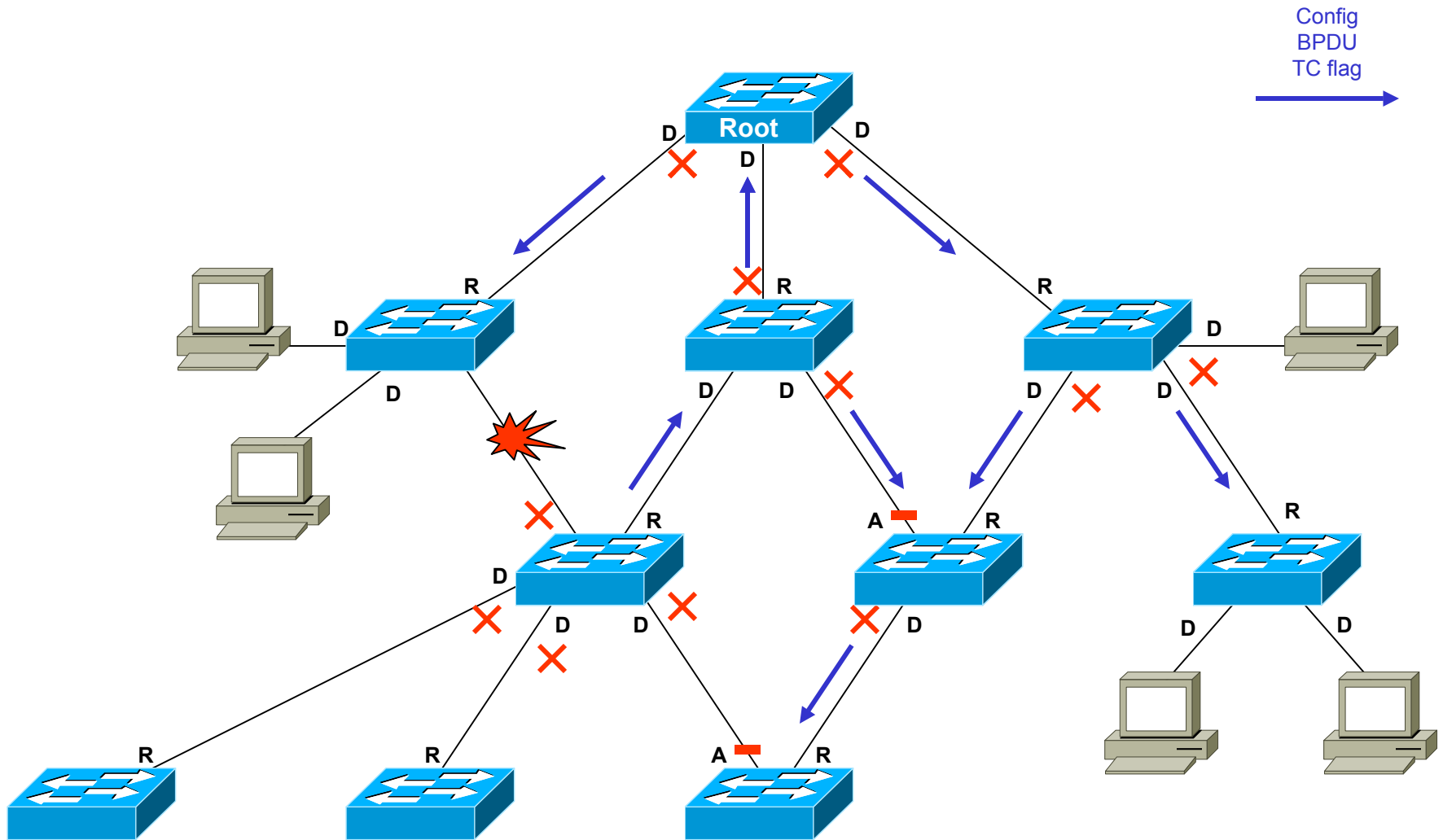
Topológia változás példa



Topológia változás példa



Topológia változás példa



RSTP időzítések

- **Max Age** és **Forward Delay** normális esetben szükségtelen
 - **elveszett handshake** üzenetek esetén érvényesül
 - **nem pont-pont** összeköttetések esetén szükséges
 - **kompatibilitáshoz** (STP bridge-ekkel) kell
- a **Message Age** értékét a Max Age 1/16-odával (de legalább 1 másodperccel) növeli minden bridge
 - mert különben előfordulhatna, hogy egy időközben kiesett Root Bridge-től származó Config BPDU körbe-körbe járna Max Age ideig (és közben folyamatosan nőne a Root Path Costja)
 - Designated Porttól elfogadott rosszabb BPDU okozhat ilyen szituációt
 - azaz a Message Age **hop count**ként funkcionál

RSTP-STP kompatibilitás

- **collision domainenként** lehetséges különböző működési mód
 - ha egy collision domainben van **legalább egy régi (STP) bridge**, akkor ott
 - STP (version 0) BPDU-kat használ minden bridge
 - nincsenek gyors Designated Port állapotváltások
 - STP timerek működnek
- az RSTP bridge a **hallott BPDU-k alapján** alkalmazkodik
 - új (RSTP) módban indul minden portján
 - 3 másodpercig (Migration Delay) feltétlenül az új/régi módban marad
 - függetlenül attól, hogy mit hall
 - 3 sec után hajlandó módot váltani
 - másik verziójú BPDU billenti át
- az STP bridge-ek eltűnése esetén beragad a régi működési mód
 - mivel ezt a régi módban működő RSTP bridge-ek nem tudják detektálni
 - ebből adminisztratív beavatkozással lehet kibillenteni a bridge-eket

A Rapid STP használata

- legjobban akkor jön ki a RSTP előnye, ha a **bridge-ek között point-to-point összeköttetések** vannak
 - full-duplex Ethernet esetén a bridge konfigurálás nélkül is point-to-pointnak tekinti a linket
- ideális eset, ha minden bridge a gyors változatot futtatja
 - ha csak a hálózat magjában tudnak RSTP-t a switchek, általában akkor is érdemes használni
- érdemes beállítani az Edge Portokat
 - bár a rákötött gép bekapcsolásakor általában az OS a lassabb

Tartalom

Bevezető

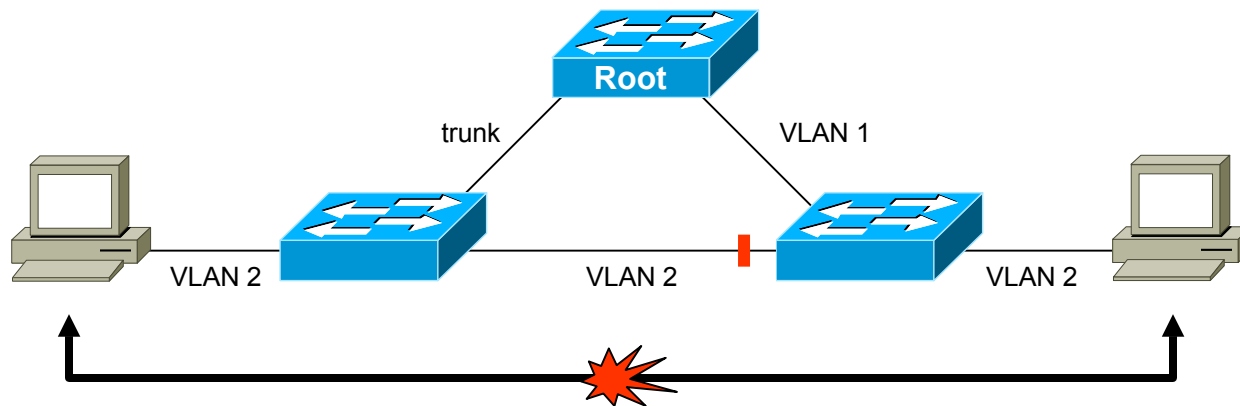
Spanning Tree Protocol

Rapid STP

Multiple STP

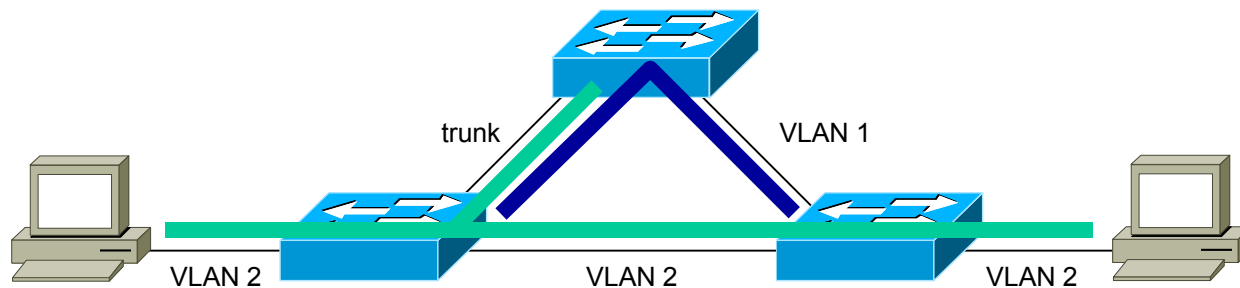
Single Spanning Tree

- az IEEE 802.1Q-1998 szabvány **egyetlen közös feszítőfa** használatát írja elő
- **kis CPU** (és sávszélesség) **igény**
- az adott VLAN tagja kell, hogy legyen minden olyan port, ami a feszítőfa része lehet a VLAN két pontja közt
 - különben a VLAN-t kettészakíthatja a STP
 - hibalehetőség!!!
 - VLAN-ok szerinti **load balancing nem lehetséges**



Per-VLAN Spanning Tree

- külön feszítőfa példány VLAN-onként
- sok VLAN esetén nagy erőforrásigény
- load balancing lehetséges, a forgalom VLAN-ok szerinti elosztásával
 - eltérő feszítőfa topológia
- PVST funkcionalitást sok gyártó implementált



Multiple Spanning Trees

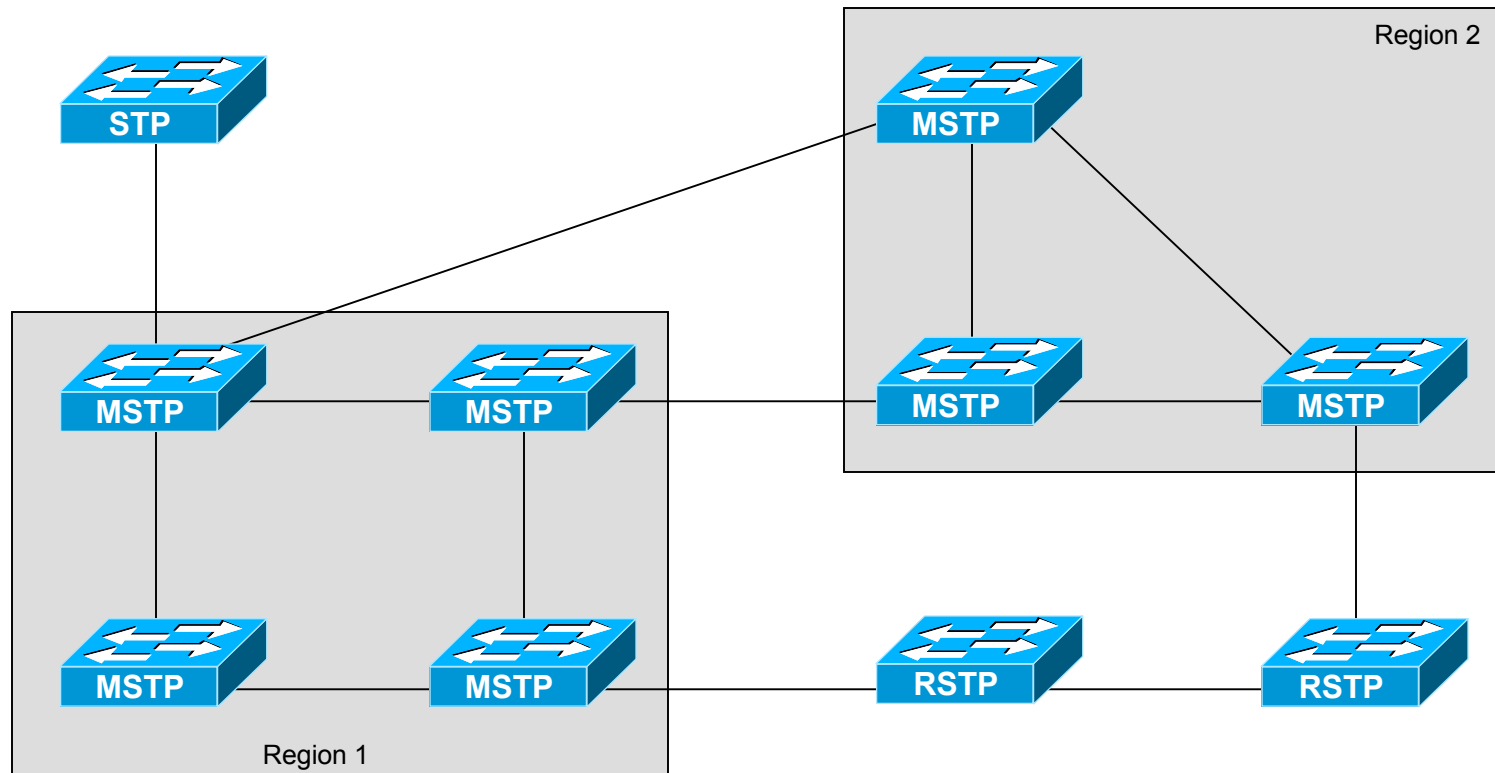
- Multiple Spanning Tree Protocol
- célja a Single Spanning Tree és a Per-VLAN Spanning Tree jó tulajdonságainak ötvözése
- IEEE 802.1s-2002: Multiple Spanning Trees
 - IEEE 802.1Q-2003 része lett

Régiók

- MST bridge-ek összefüggő csoportja, egységes MST konfigurációval
- Mire jó ez?
 - kompatibilitás
 - több Spanning Tree csak a régión belül működik
 - így a régiókon kívül lehet a hálózatban RSTP és STP bridge is
 - további flexibilitás

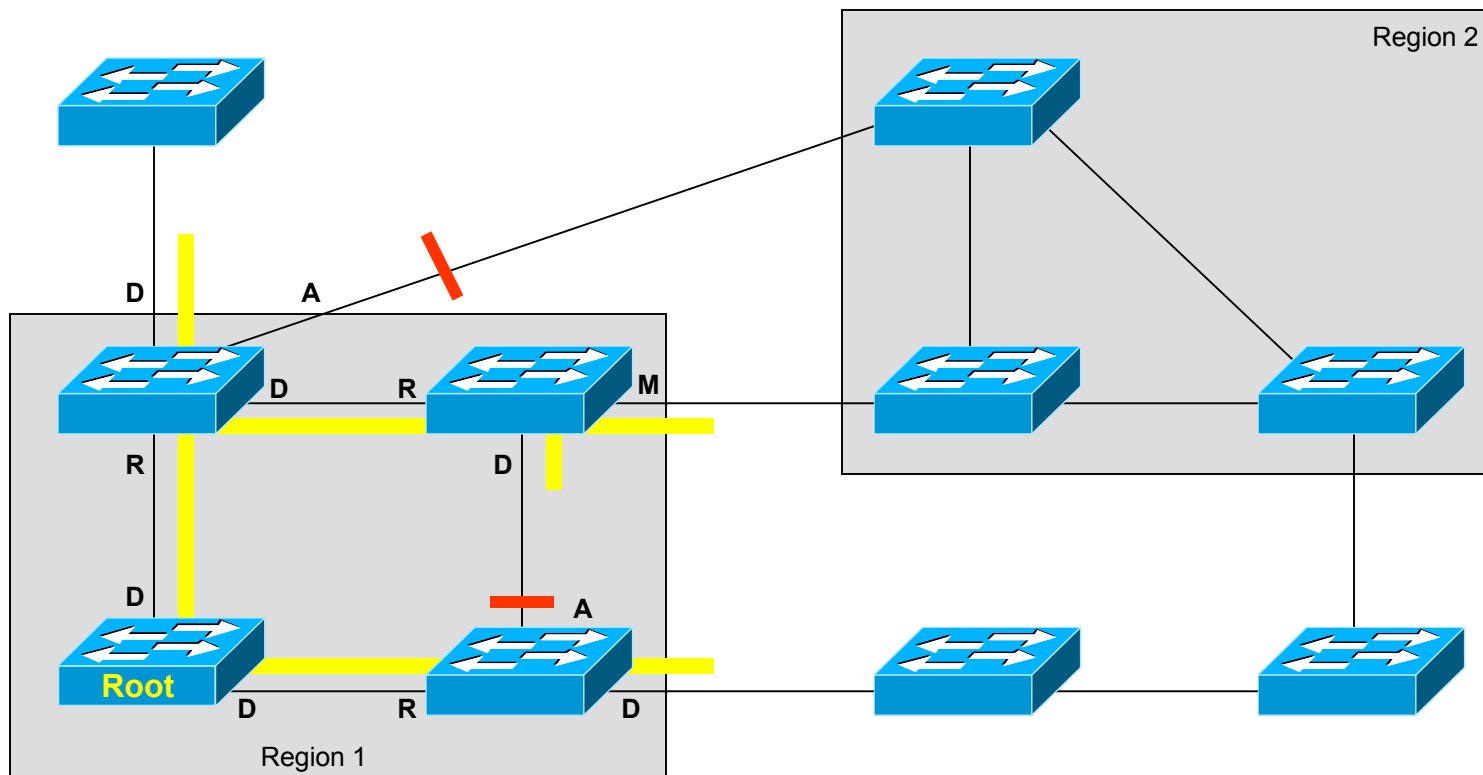
Régiók

- több Spanning Tree a **régió**n belül
- régiókon kívül csak **egy** Spanning Tree
 - 802.1Q-1998 Single Spanning Tree



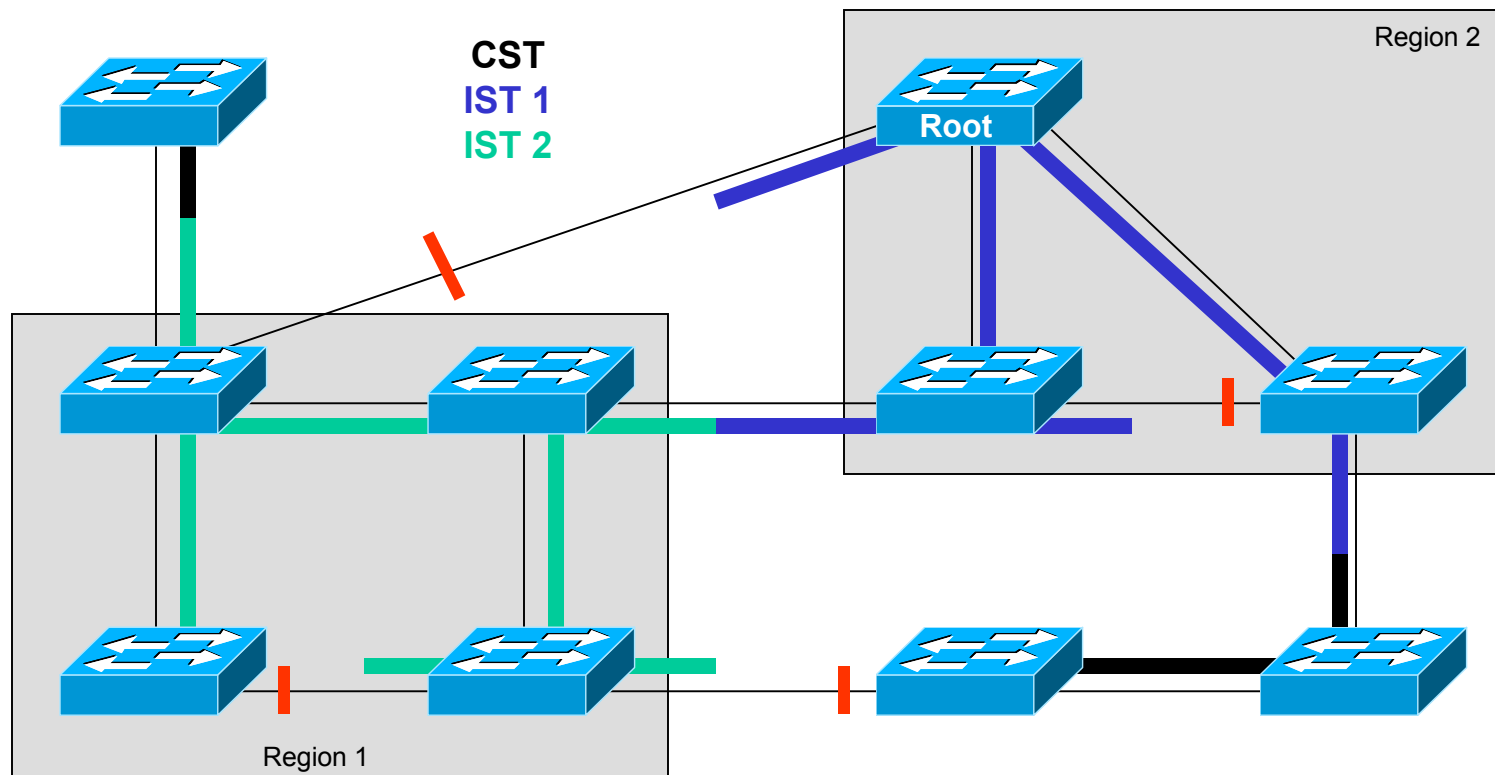
MST Instance

- feszítőfa példány egy régióban
 - teljesen a **régión belül** van
 - bármely két pontja közti út a régión belül halad
 - az adott régión kívül nem is tudnak a létezéséről



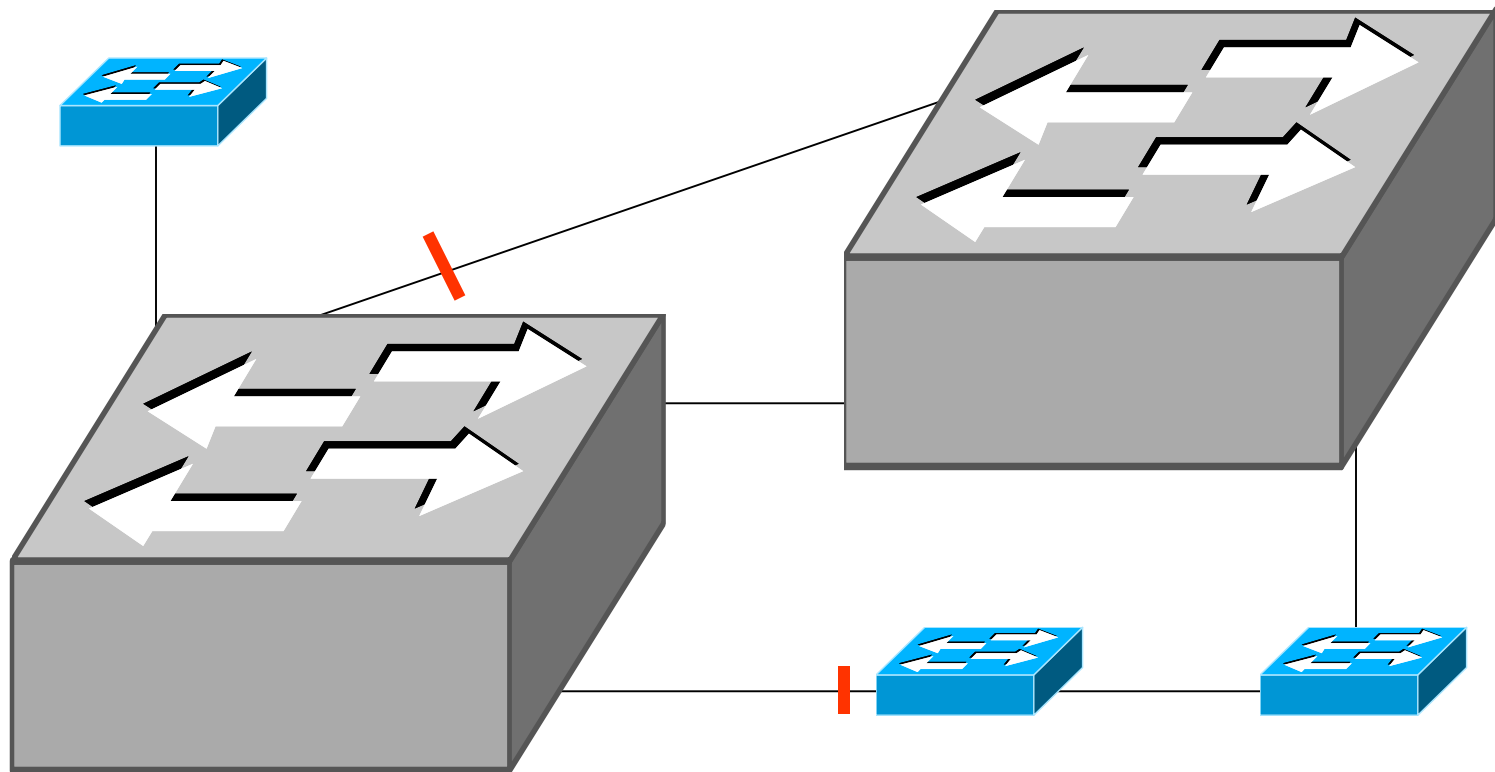
Common and Internal Spanning Tree

- **CIST**: feszítőfa a teljes hálózatra (802.1Q-1998 Single ST)
 - része régióként egy-egy Internal Spanning Tree (IST)
 - olyasmi, mint egy MST példány
 - része a régiók között a Common Spanning Tree (CST)



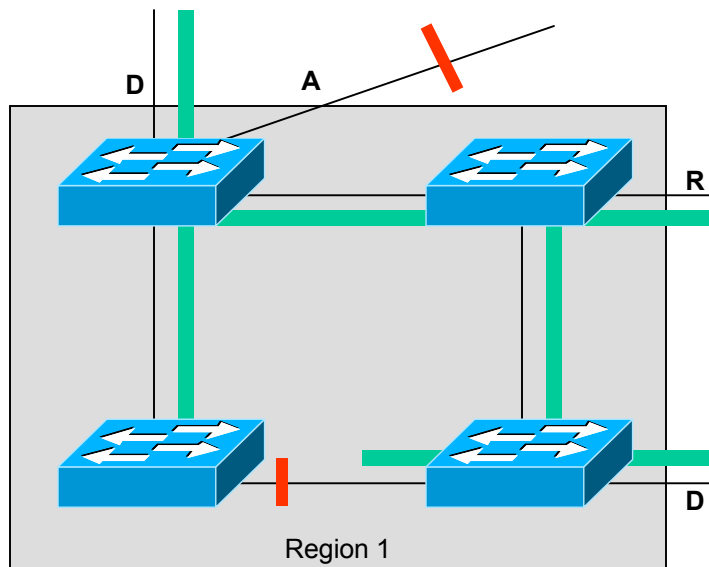
A régió mint bridge

- egy régió a CST szemszögéből egy „nagy RSTP bridge”

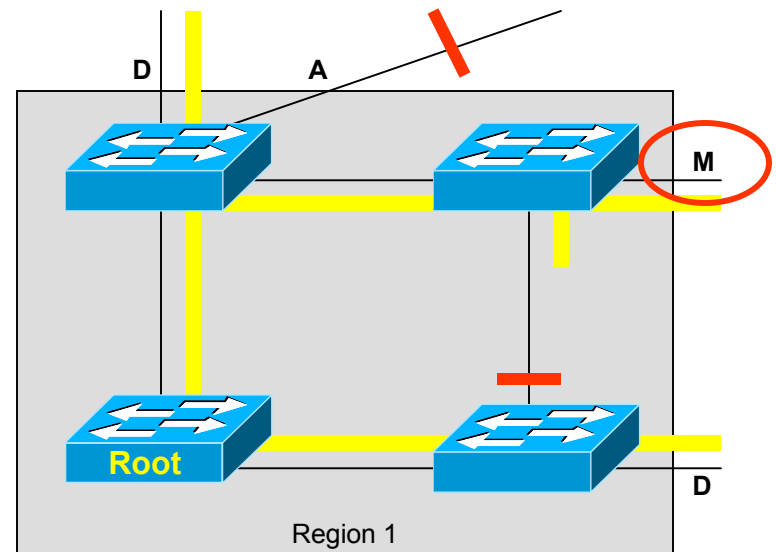


A régió pereme

- az MST példányok portjainak állapota a régiók peremén a CIST portok állapotát követi
- minden régiónak egyetlen CIST Root Portja van
 - vagy egy sincs, ha a CIST Root abban a régióban van
 - új port szerep: ez a port az MST példányok Master Portja



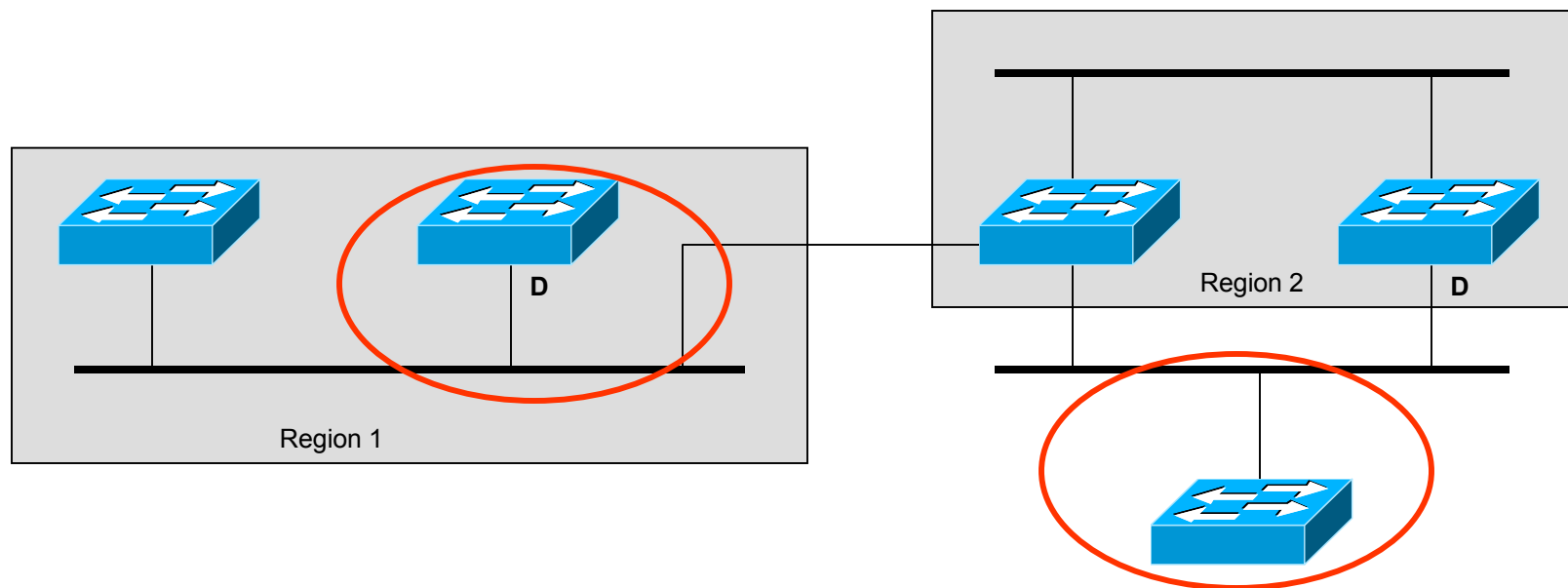
CIST



MSTI x

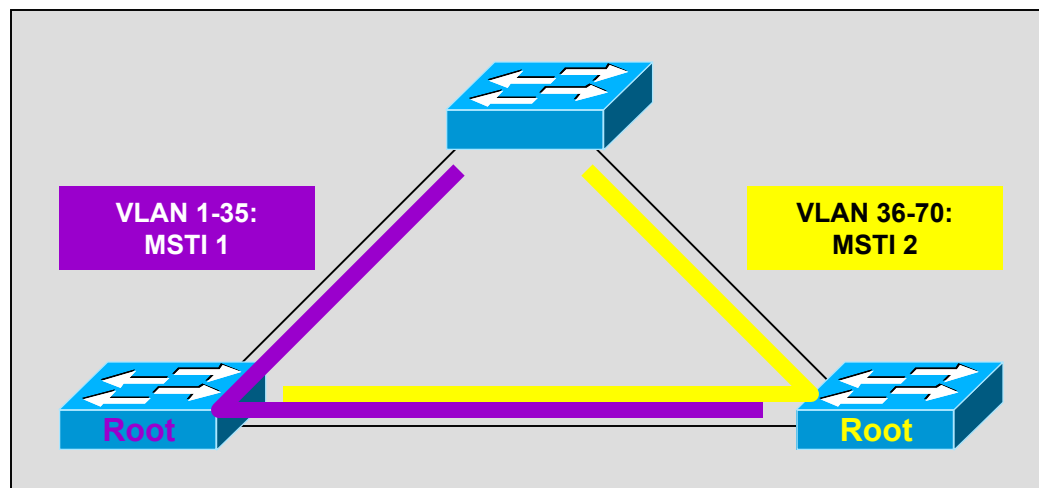
Collision domainek hovatartozása

- a collision domain
 - nem tartozik régióba, ha kapcsolódik hozzá régiókon kívüli (nem MST) bridge
 - különben abba a régióba tartozik, amelyikbe a domain CIST Designated Bridge-e is
- a régión kívülre kerülő collision domain kettészakíthatja a régiót annak ellenére, hogy az MST konfiguráció azonos



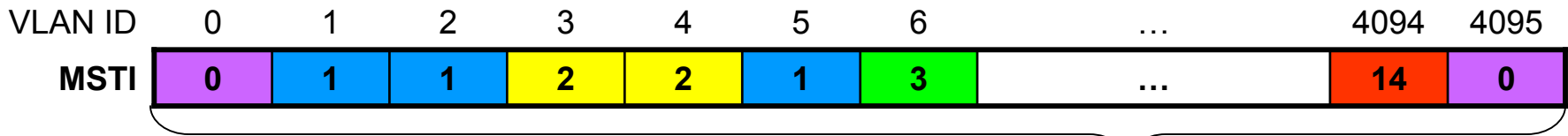
Feszítőfák használata régióon belül

- régióon belül a VLAN-ok hozzá vannak rendelve a feszítőfa példányokhoz (vagy az IST-hez)
 - az összerendelés csak az adott régióban érvényes
- több VLAN is használhat egy MST példányt
 - csak annyi példány, ahány eltérő topológiára szükség van
- az MST példány azokon a porton működik, ami tagja a példányhoz tartozó VLAN-ok valamelyikének
 - load balancing példányok között lehetséges

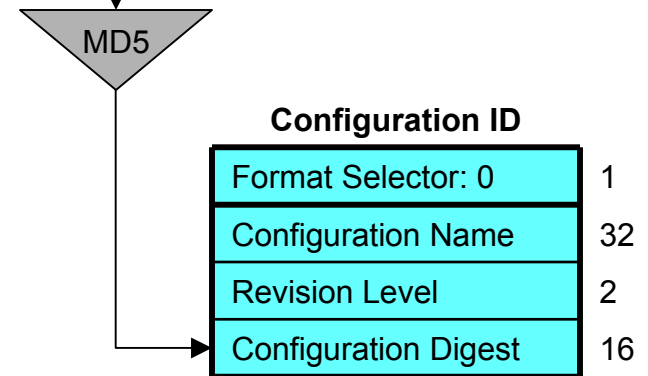


MST Configuration

- Configuration Table: **VLAN** → **MSTI** összerendelés
 - minden lehetséges VLAN-ra megadja, hogy melyik MST példányt használja

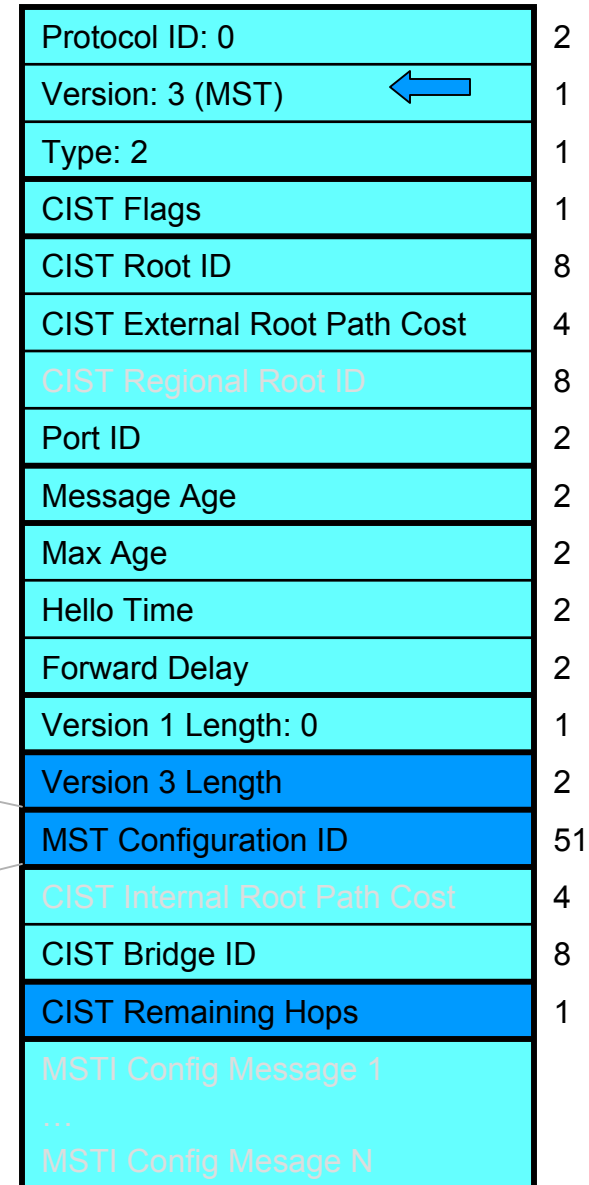
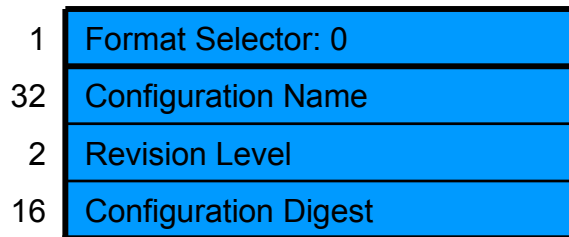


- Configuration ID:
 - **Configuration Name**: régió neve
 - **Revision Level**: konfiguráció sorozatszama
 - **Configuration Digest**: Config Table lenyomata (HMAC-MD5) 16 byte-on
- a konfigurációnak a **régióban** **egységesnek** kell lennie
 - ellenőrizhető, mert a Configuration ID szerepel a BPDU-ban



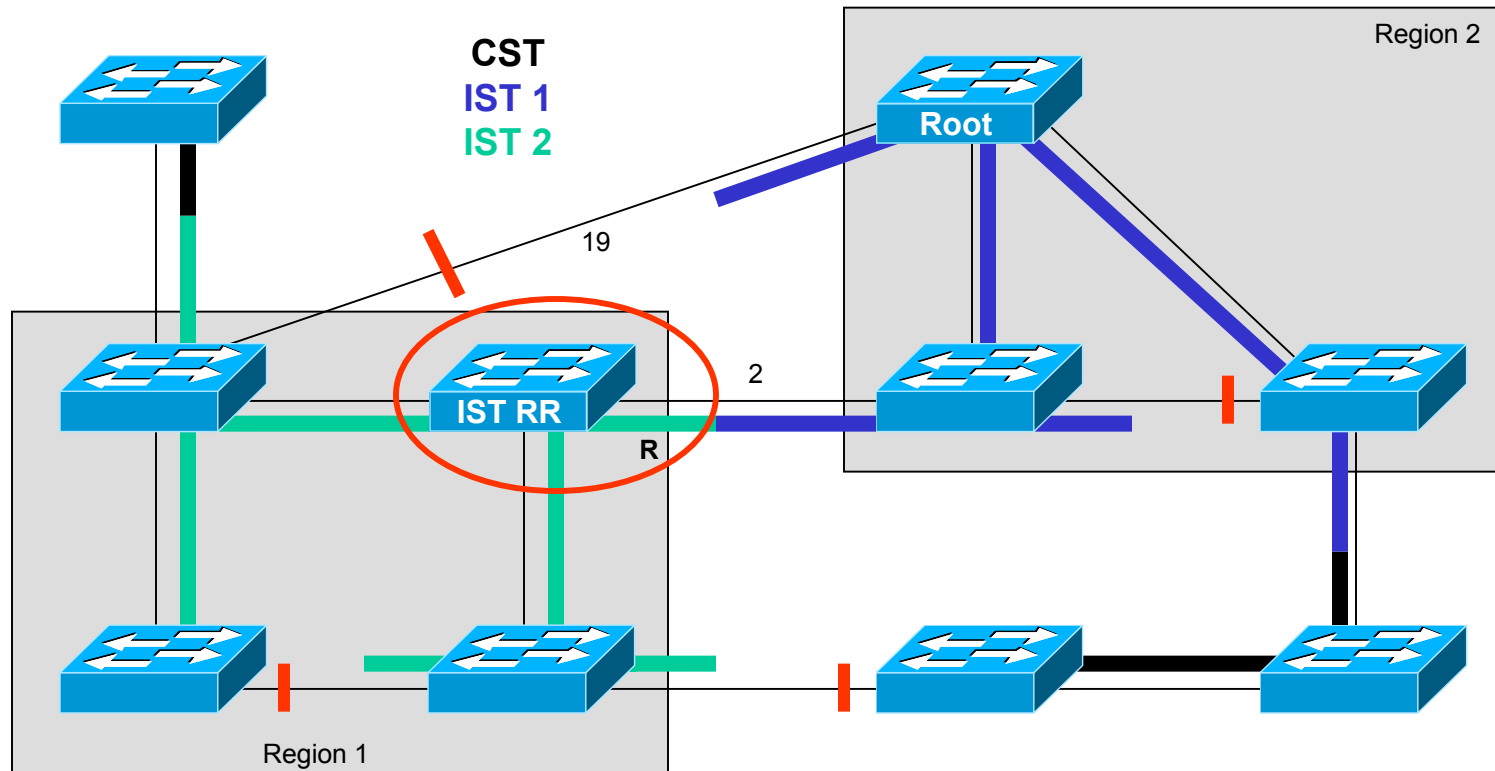
MST BPDU

- **Version 3 Length:** további byte-ok száma
 - v2 Length nem kell, mert az a verzió (RSTP) nem bővítette az előző változat BPDU-ját
- **Remaining Hops**
 - az RSTP által hop count funkcióra átalakított Message Age helyett



IST Regional Root Bridge

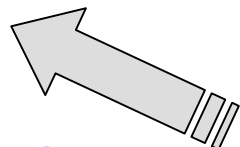
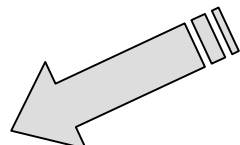
- övé a **legjobb CIST Root Path Cost** (External + Internal)
 - ha több ilyen van a régióban, akkor a jobb BID dönt
- reprezentálja a régiót kifelé
 - az ő Root Portja a „nagy bridge” Root Portja
 - a „nagy bridge” BID-ja az övé



MST BPDUs – priority vector

- a Bridge ID elé beékelődik a régióon belüli Root ID és Root Path Cost
- CIST External Root Path Cost
 - az IST Regional Root külső (CST) RPC-ja
 - értéke a régióon belül állandó

Root ID
External Root Path Cost
Regional Root ID
Internal Root Path Cost
Bridge ID
Port ID

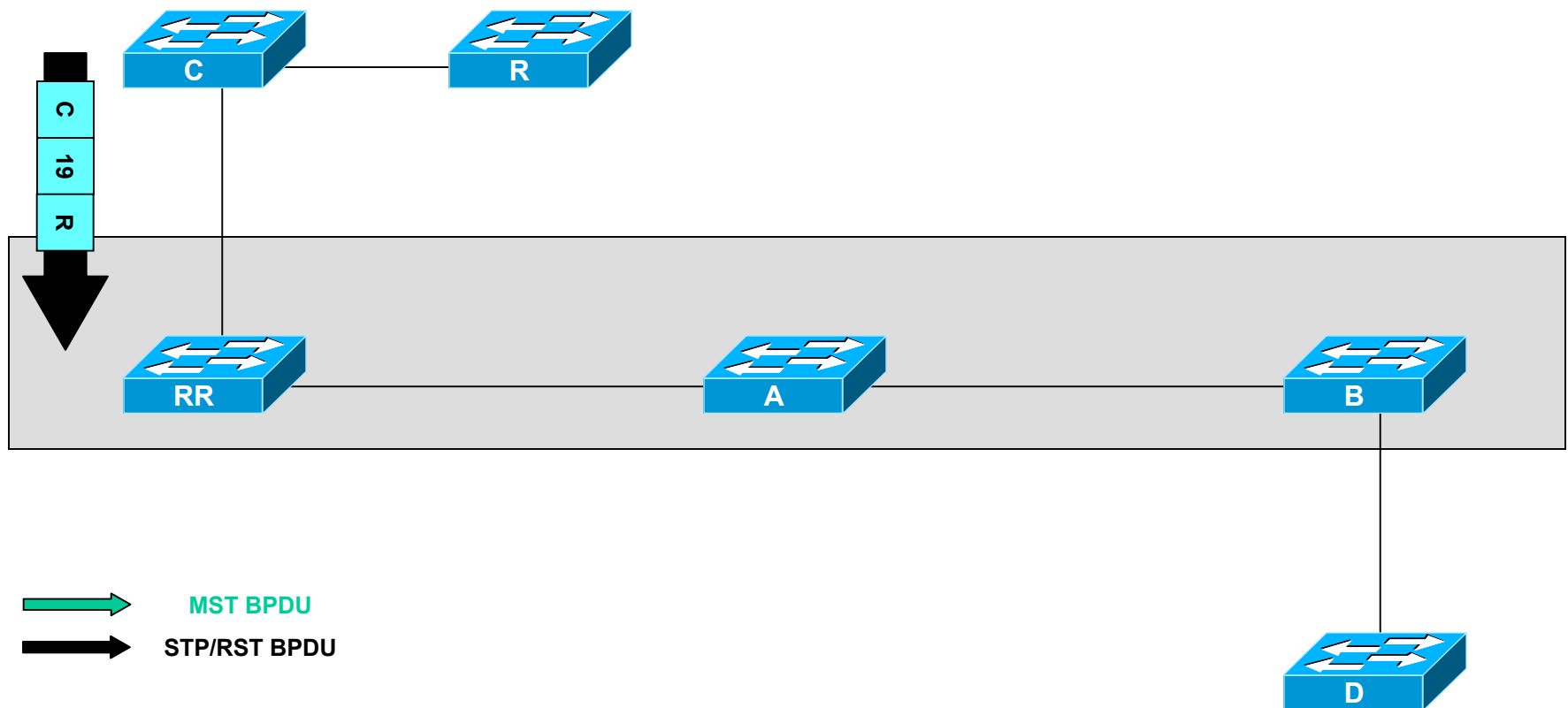


- így mindig a régióon belüli utak a preferáltak a régiót elhagyókkal szemben

Protocol ID: 0	2
Version: 3 (MST)	1
Type: 2	1
CIST Flags	1
1. CIST Root ID	8
2. CIST External Root Path Cost	4
3. CIST Regional Root ID	8
6. Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2
Version 1 Length: 0	1
Version 3 Length	2
MST Configuration ID	51
4. CIST Internal Root Path Cost	4
5. CIST Bridge ID	8
CIST Remaining Hops	1
MSTI Config Message 1	
...	
MSTI Config Message N	

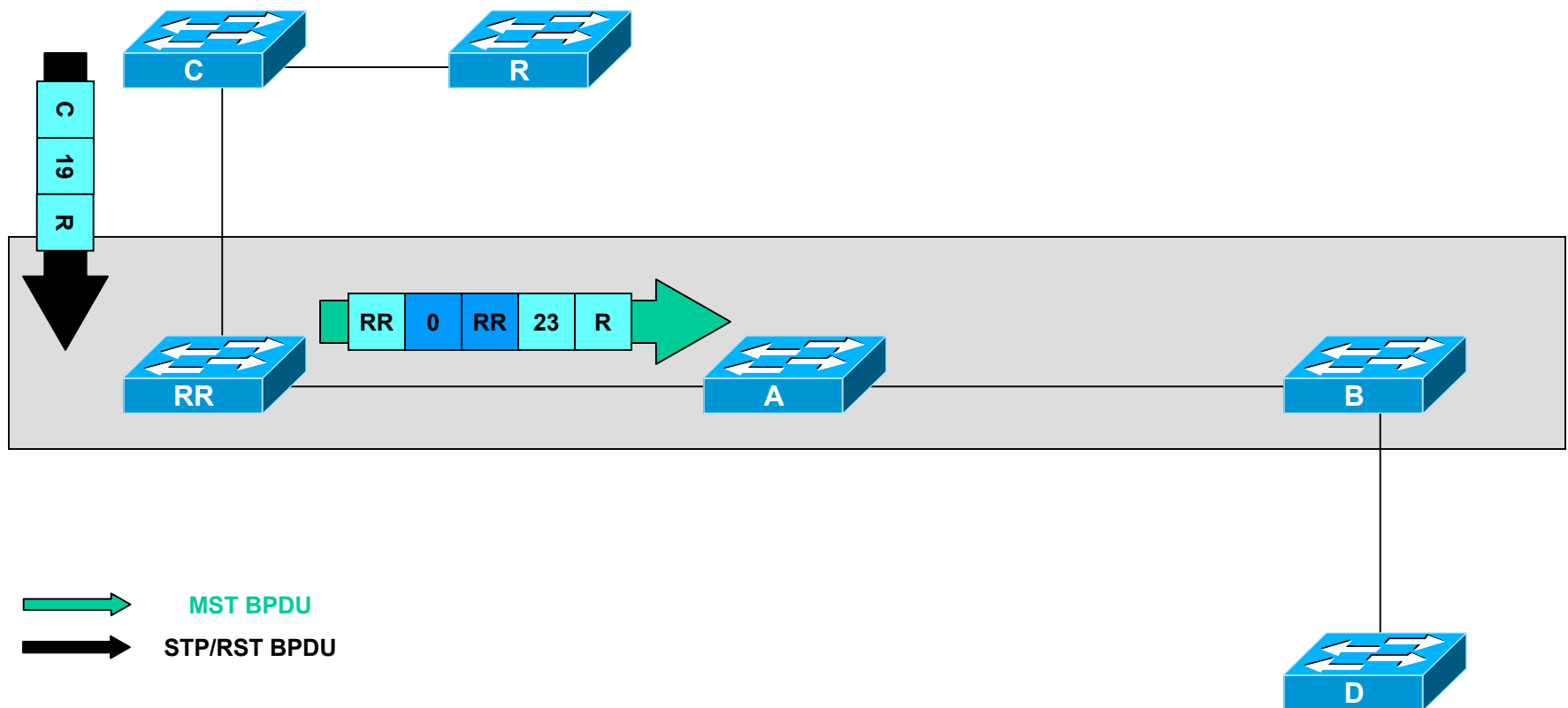
IST topológia

- a régióba bejövő BPDU a régió szélén **kiegészül** a régióon belül használatos paraméterekkel
- a külső Root Path Cost régióon belül változatlan
- a régiót elhagyó BPDU-ból **kikerülnek** a belső paraméterek



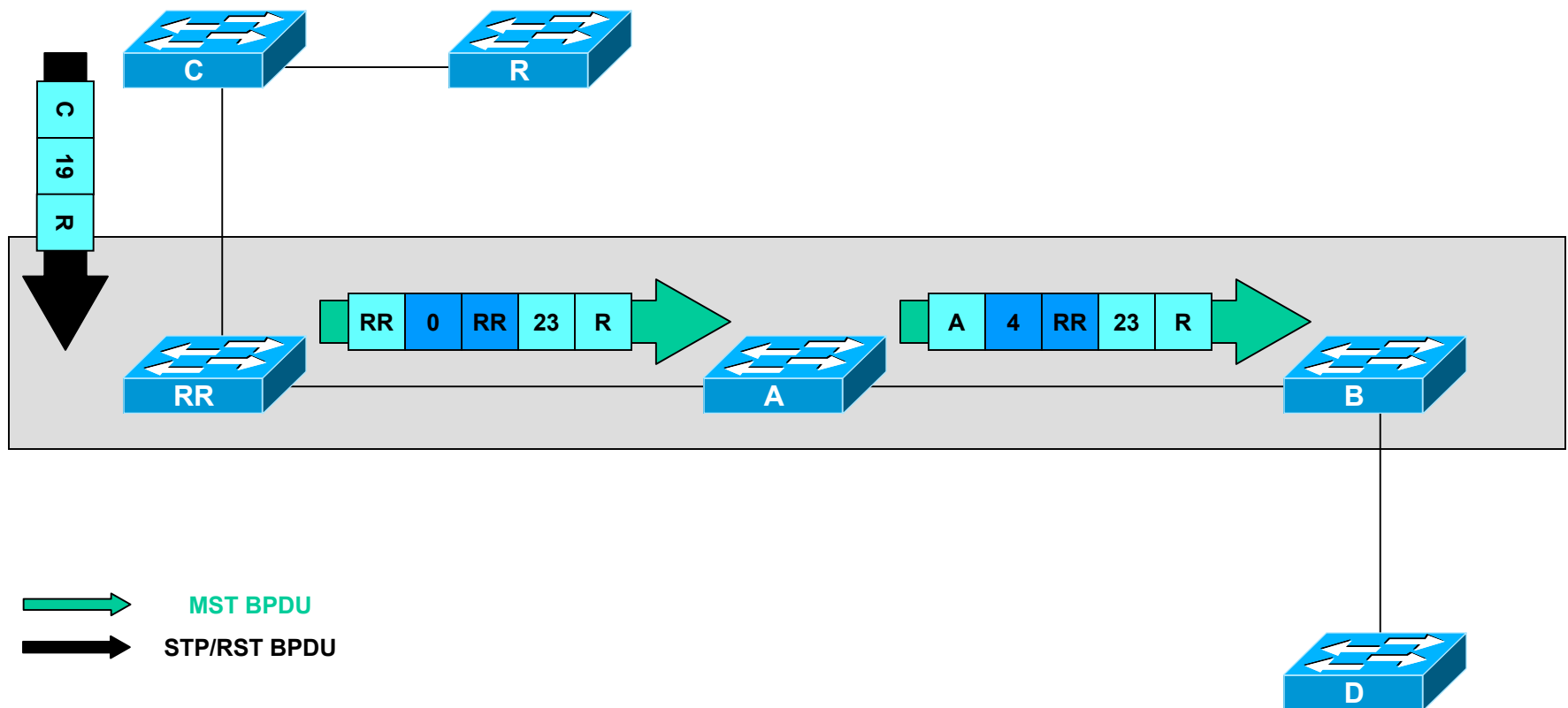
IST topológia

- a régióba bejövő BPDU a régió szélén **kiegészül** a régióon belül használatos paraméterekkel
- a külső Root Path Cost régióon belül változatlan
- a régiót elhagyó BPDU-ból **kikerülnek** a belső paraméterek



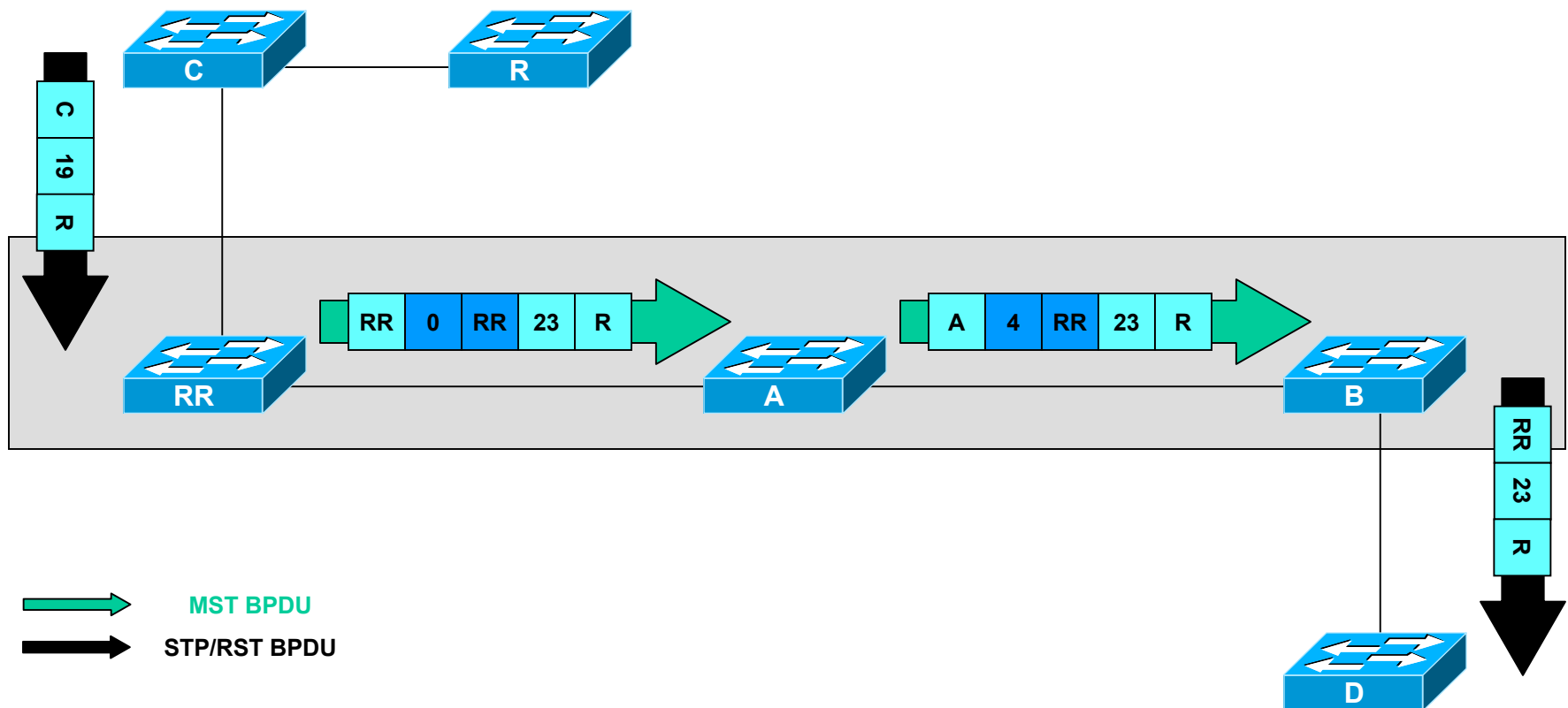
IST topológia

- a régióba bejövő BPDU a régió szélén **kiegészül** a régióon belül használatos paraméterekkel
- a külső Root Path Cost régióon belül változatlan
- a régiót elhagyó BPDU-ból **kikerülnek** a belső paraméterek



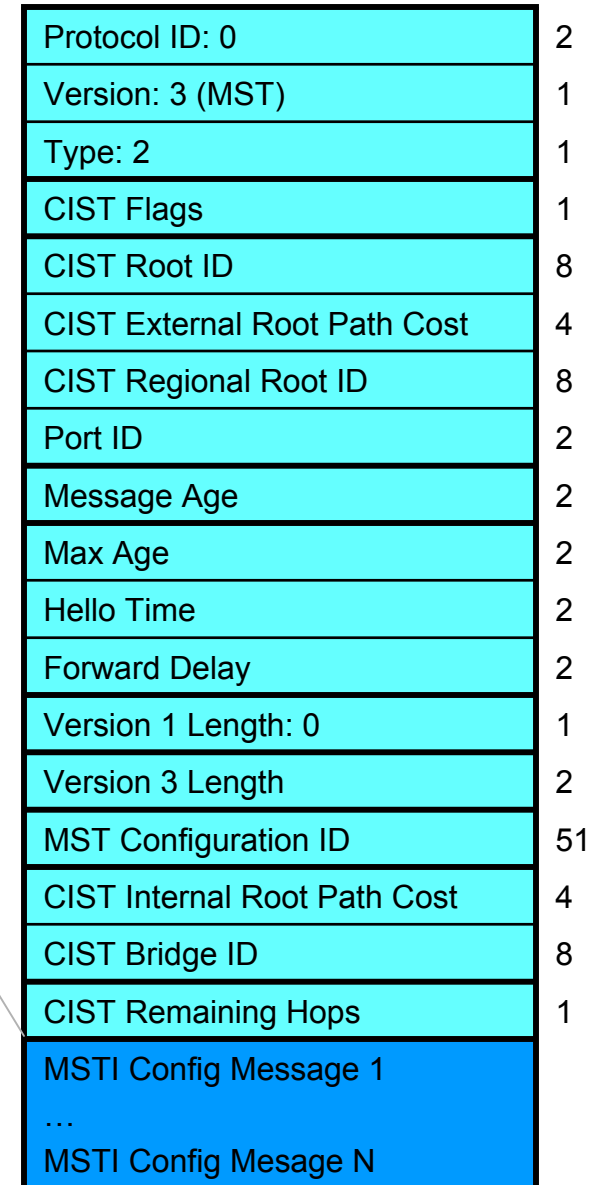
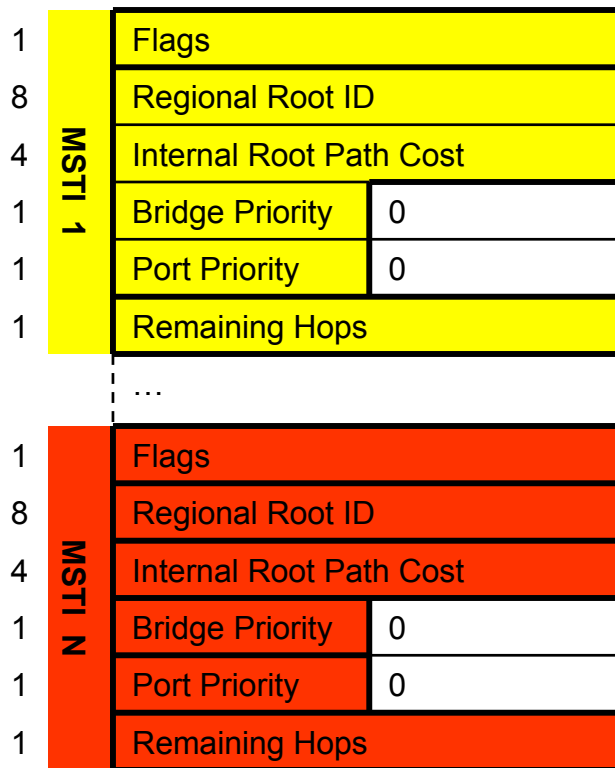
IST topológia

- a régióba bejövő BPDU a régió szélén **kiegészül** a régióon belül használatos paraméterekkel
- a külső Root Path Cost régióon belül változatlan
- a régiót elhagyó BPDU-ból **kikerülnek** a belső paraméterek



MSTI Configuration Messages

- MST példány száma a Reg. Root ID-ben
- Bridge/Port ID-ből elég a 4 bit prioritás
 - a többi azonos az IST-ével
- max. 64 MSTI (a keretméret miatt)



MST példányok topológiája

- a szokásos topológia +
 - a régió szélén levő portok állapota a CIST port állapotát veszi fel
 - az IST Regional Root Bridge Root Portja a MST példány Master Portja
- egy MST BPDUs-ban minden a porton működő VLAN által használt példányhoz tartozó MSTI Config Message szerepel

