

## Curs 04

# Optimizarea rețelelor locale



# Obiective

- Rolul VLAN-urilor în rețele
- Stabilirea conectivității între VLAN-uri
- STP
- Etherchannel

# VLAN

Provocări în LAN

Porturi acces

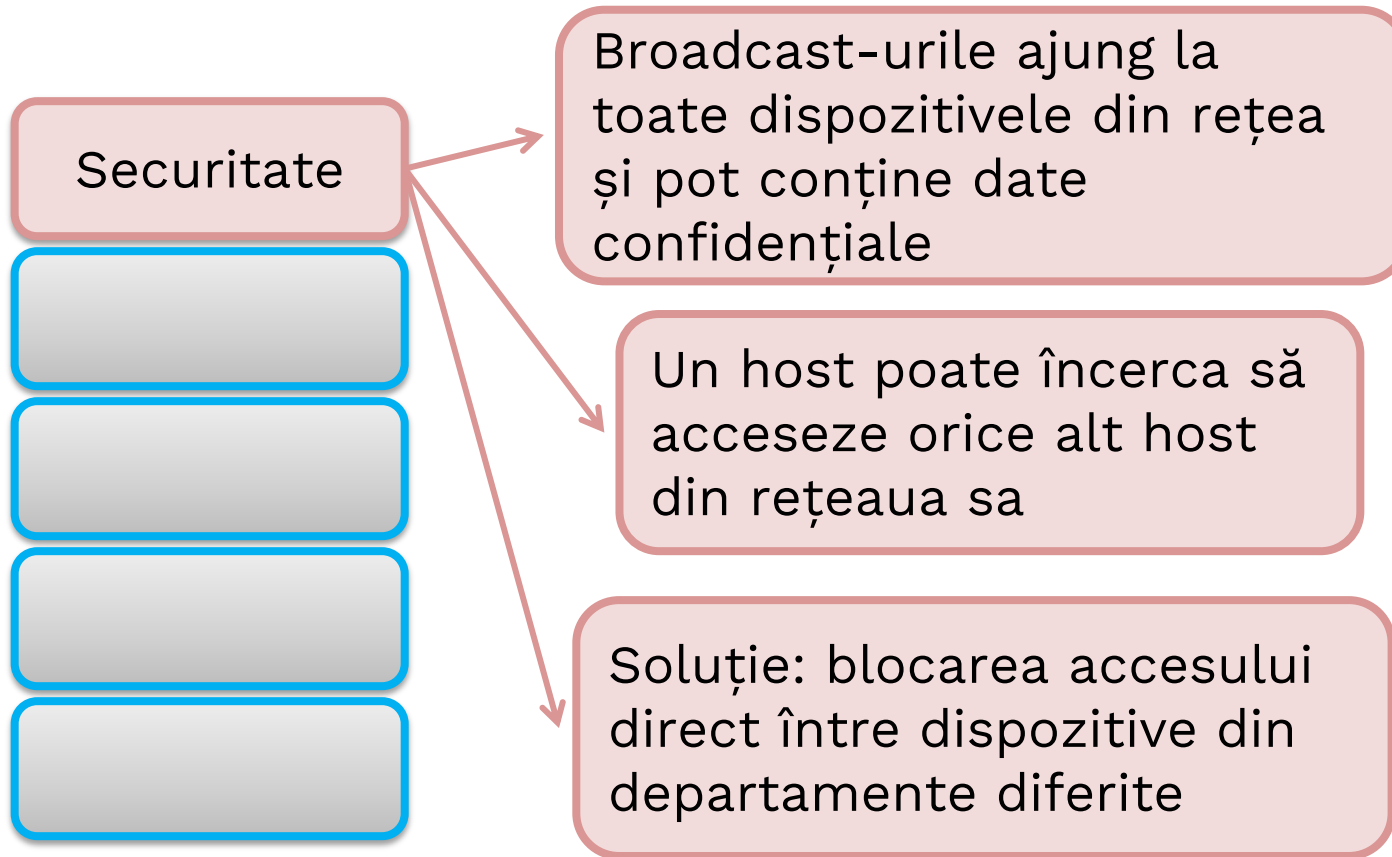
Porturi trunchi



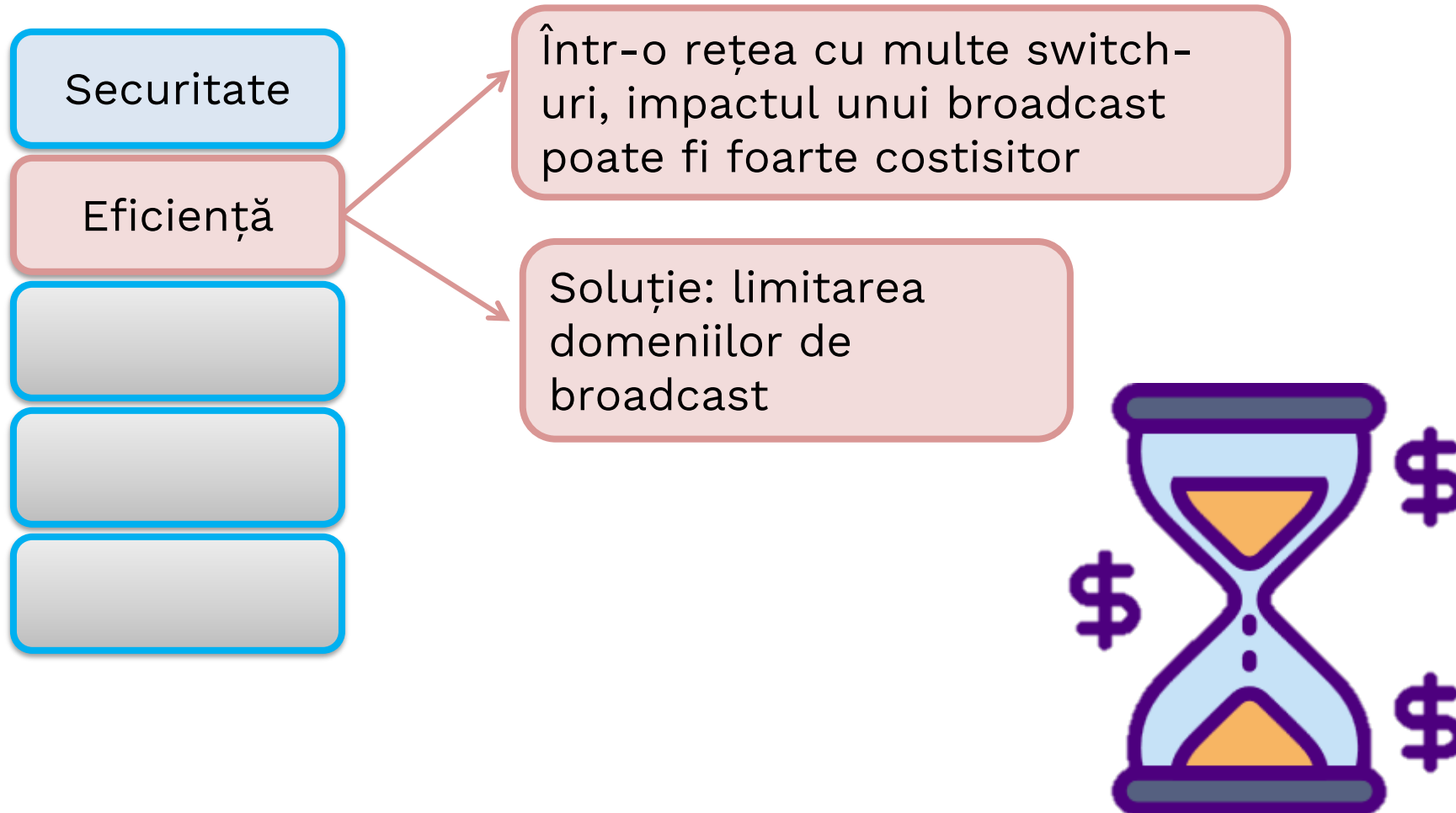
**RL**

crunch it connected

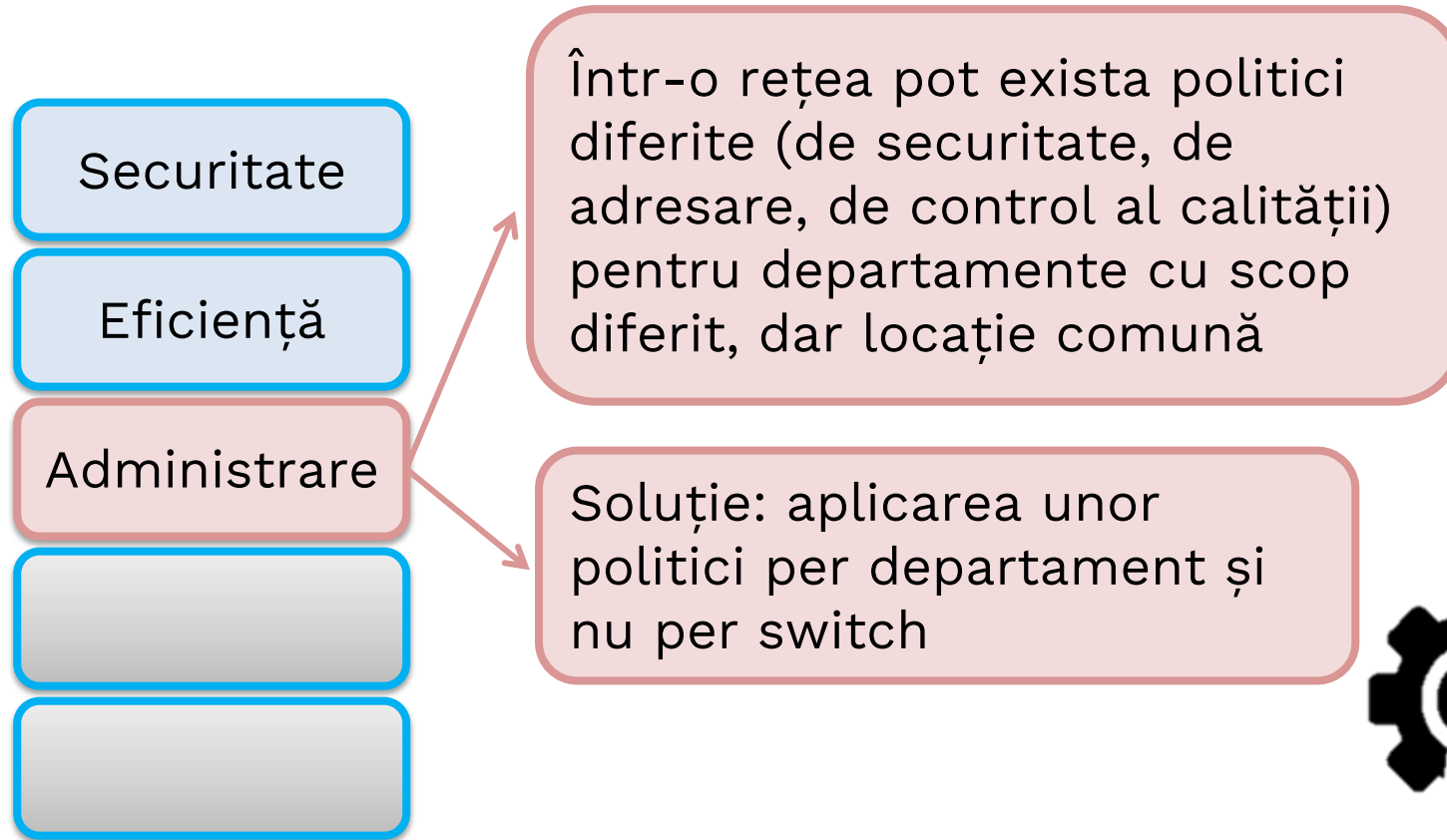
# Probleme în LAN-uri



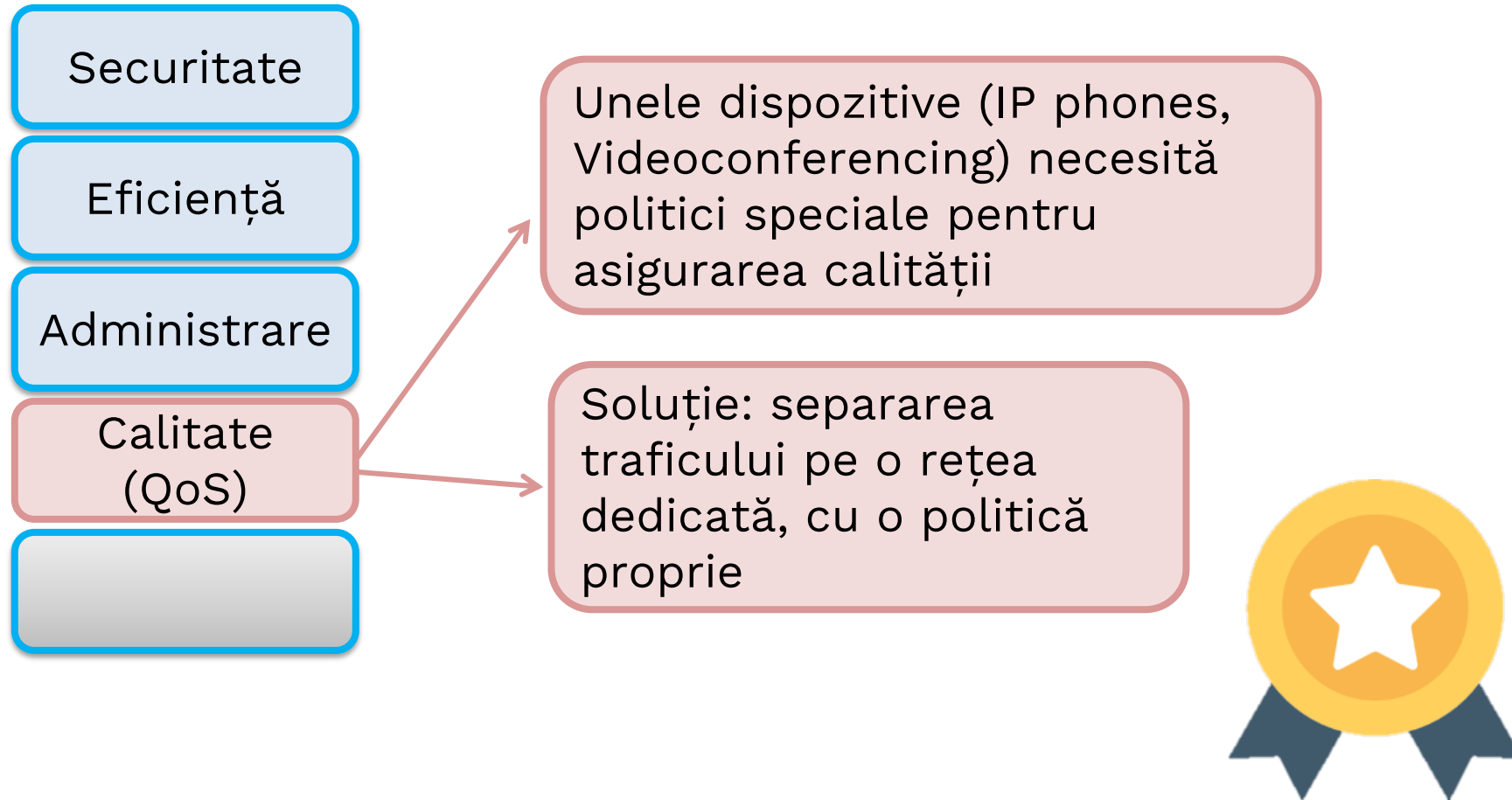
# Probleme în LAN-uri



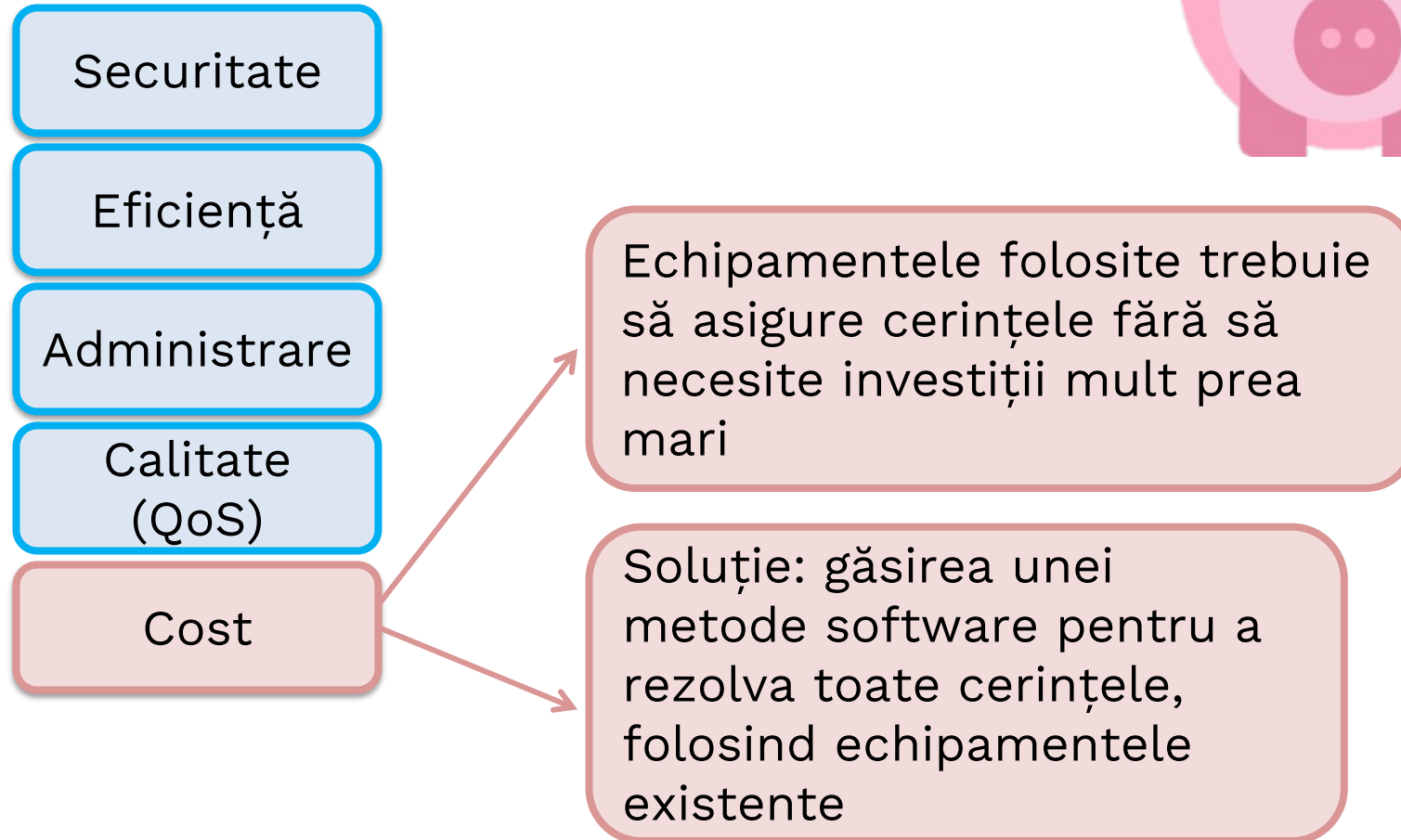
# Probleme în LAN-uri



# Probleme în LAN-uri

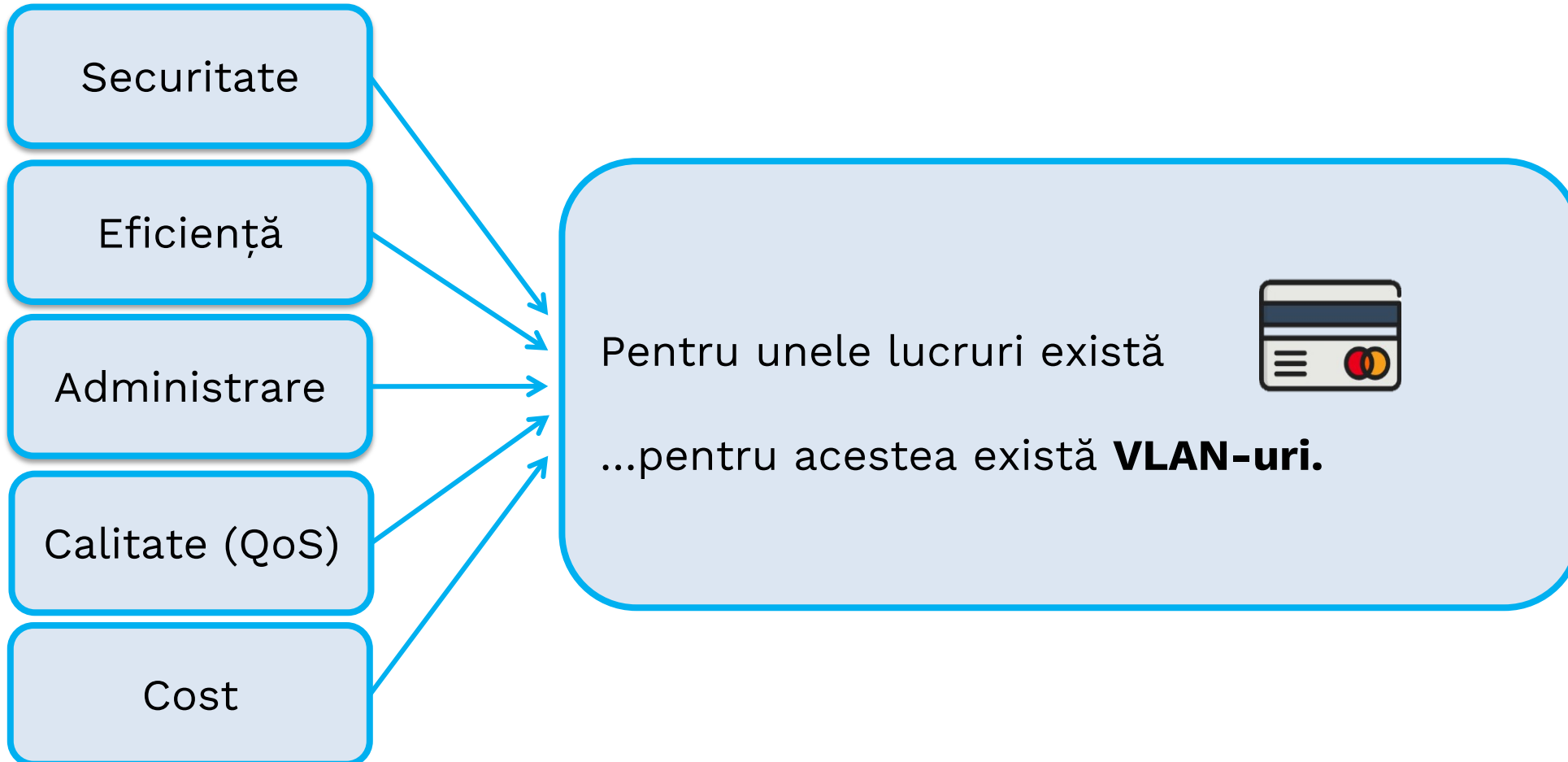


# Probleme în LAN-uri





# Soluția

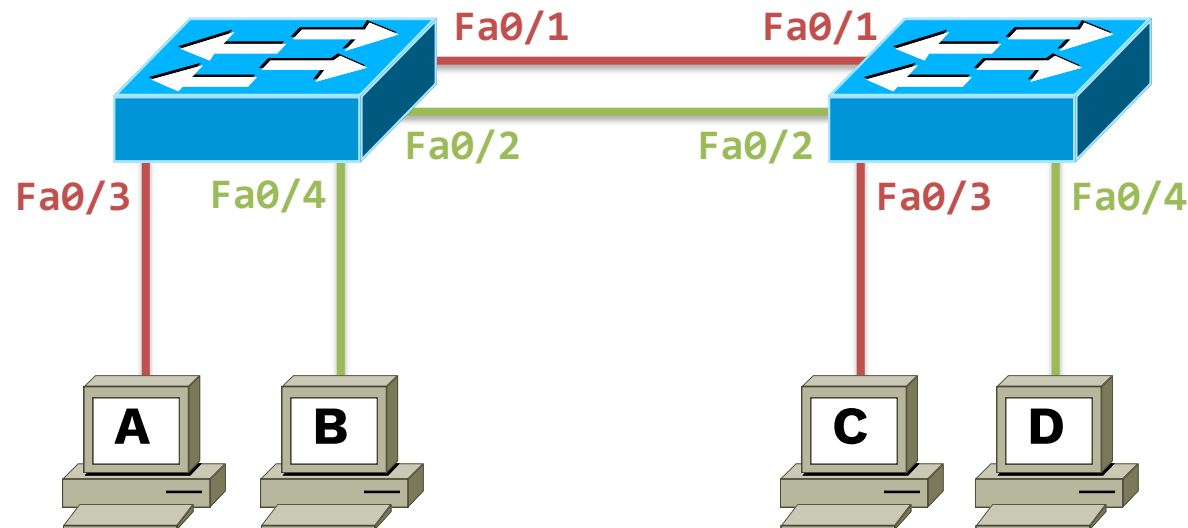


# De ce nu un ruter?

- Uneori dispozitive de la departamente diferite pot fi situate în aceeași locație fizică
- Ruterile sunt mai scumpe
- Ruterile fac operații mai costisitoare deci impun o latență mai mare
- Segmentează domeniile de broadcast și vrem ca stațiile unui departament să fie în același domeniu

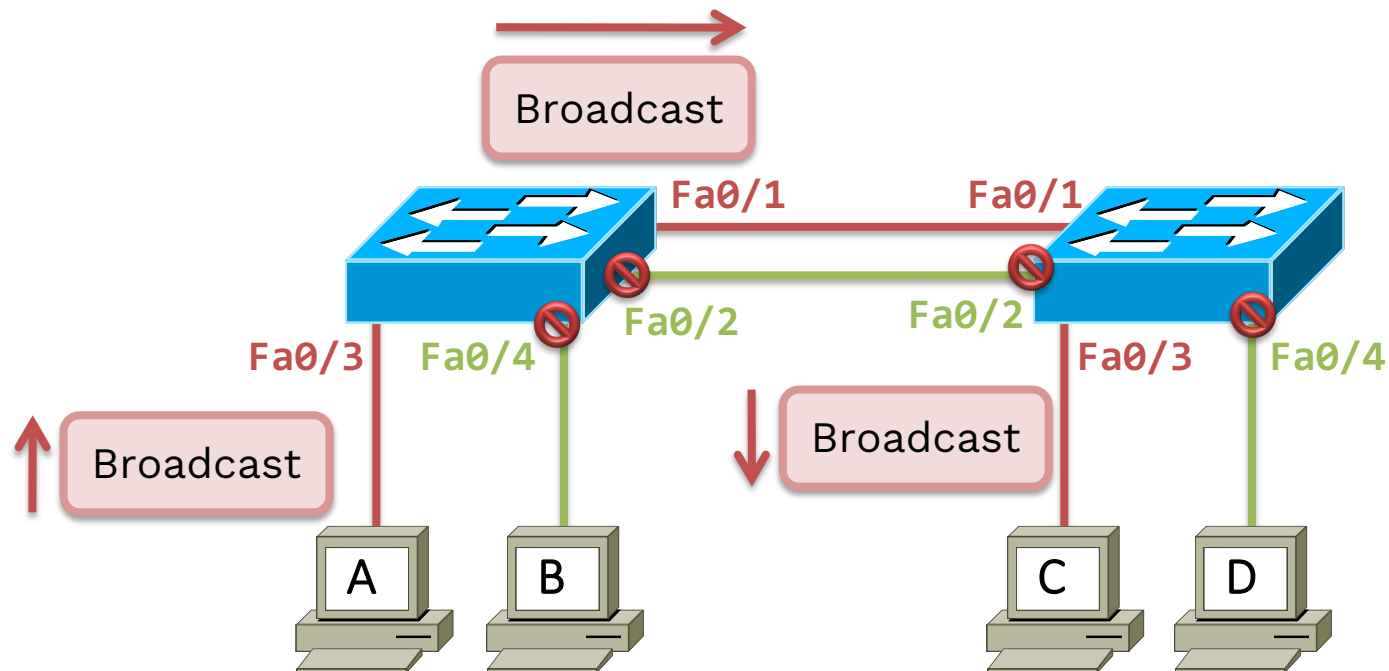
# Ce este un VLAN?

- Virtual LAN
- Reprezintă un domeniu de broadcast compus doar din anumite porturi ale unor switch-uri
- Un VLAN este definit prin porturile ce îi aparțin



# Ce este un VLAN?

- Dispozitive din două VLAN-uri diferite nu pot comunica între ele în absența unui dispozitiv de nivel 3 care să facă rutarea
- Un broadcast se va propaga doar în VLAN-ul respectiv:



# Ce este un VLAN?

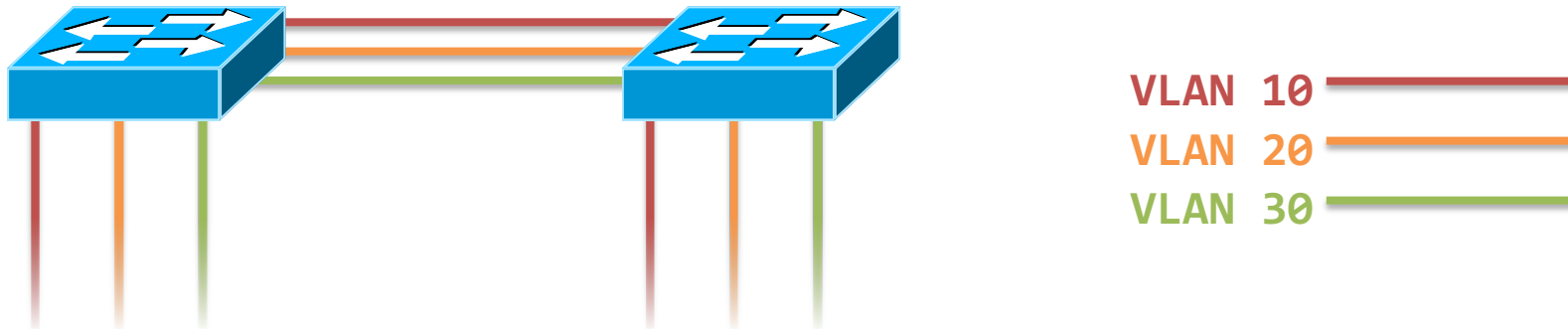
- VLAN-urile sunt identificate prin numere numite **VLAN ID**
- Un VLAN ID este reprezentat pe 12 biți (1 – 4096)
- Intern, fiecare switch asociază unui port un VLAN ID
- Pe switch-urile Cisco, toate porturile aparțin inițial VLAN-ului 1
- Un port ce aparține unui singur VLAN poartă numele de **Access Port**
- Pentru stațiile conectate la un Access Port, faptul că aparțin unui VLAN este transparent

# Configurarea VLAN-urilor

- Un VLAN trebuie creat pe un switch înainte să îi fie asociate porturi
- Pentru a comuta trafic aparținând VLAN-ului **<X>** un switch trebuie să aibă configurat VLAN-ul **<X>**

# Trunking

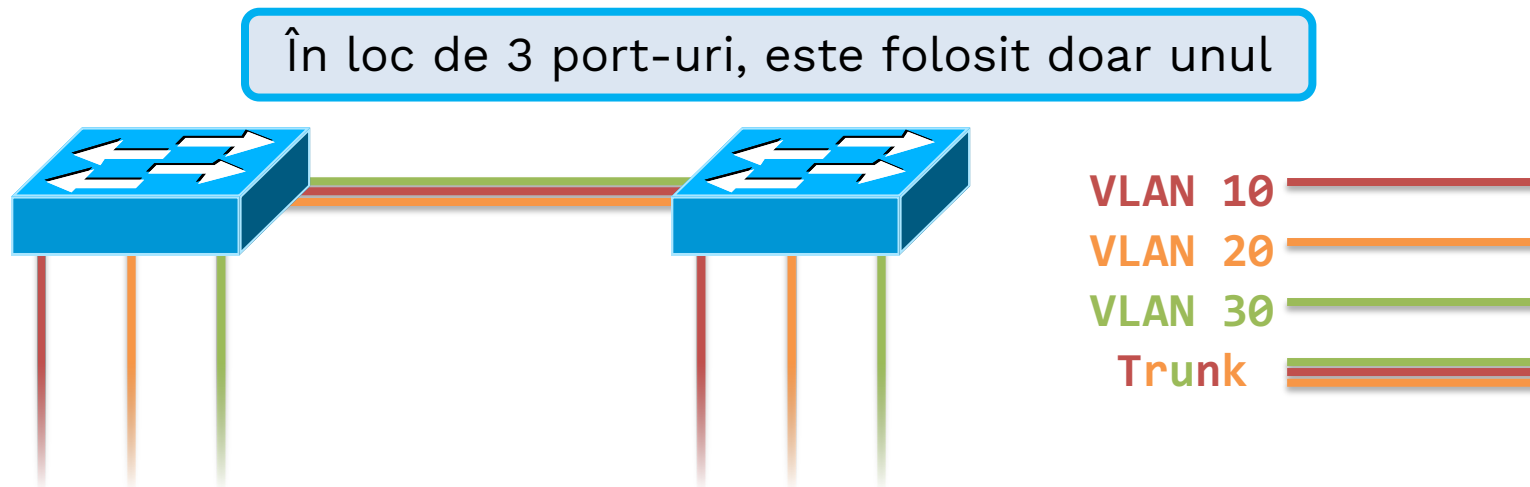
- Ce se întâmplă când două switch-uri trebuie să transporte date aparținând mai multor VLAN-uri între ele?



- Prea multe porturi folosite pentru a transporta toate VLAN-urile
- Soluția: trunking

# Trunking

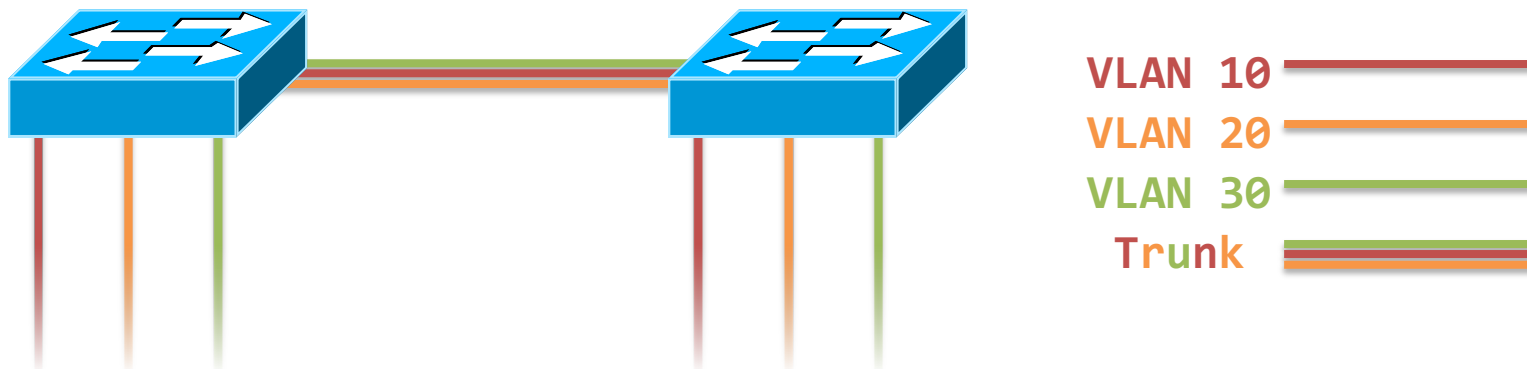
- Porturile nu pot funcționa doar ca Access Ports, ci și ca Trunk Ports
- Acestea au proprietatea că pot trimite trafic aparținând mai multor VLAN-uri pe același port
- O linie trunk trebuie să aibă la ambele capete port-uri configurate ca Trunk Ports





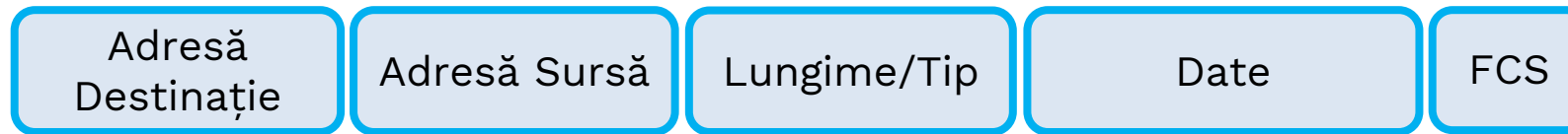
# Trunking

- Setul de VLAN-uri ce pot fi trimise pe o linie trunk este configurabil și trebuie stabilit de administrator
- Implicit, setul va include toate VLAN-urile
- Problemă: dacă switch-ul 1 trimite un cadru aparținând VLAN-ului 10, cum își dă seama switch-ul 2 în ce VLAN să-l plaseze?



# Formatul 802.1q

- Soluția: **802.1q**
- Recapitulare – formatul Ethernet:



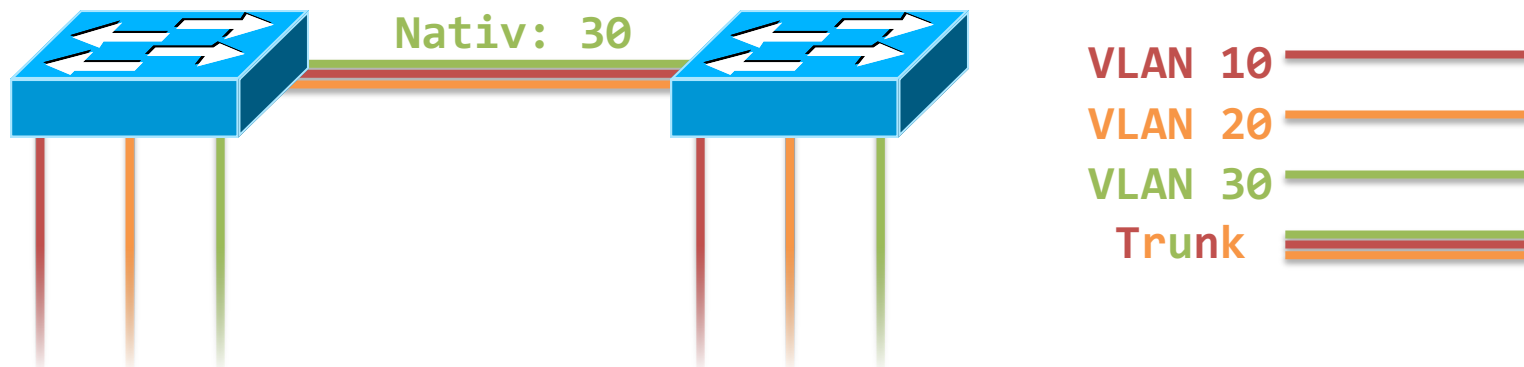
- Pentru a reține informația de VLAN, se introduce un câmp nou format din 4 octeți: **802.1q tag**



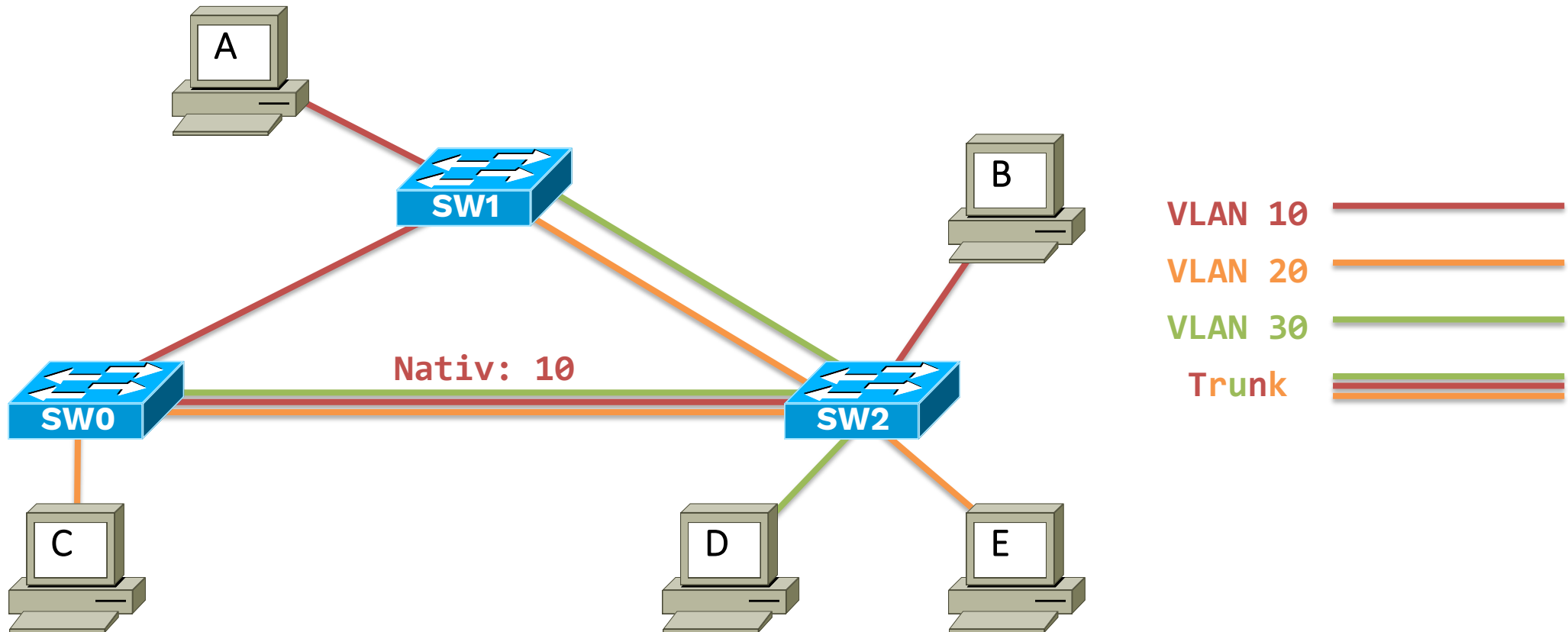
- Noul format al cadrului poartă numele de formatul 802.1q și e folosit pe legăturile trunk

# VLAN nativ

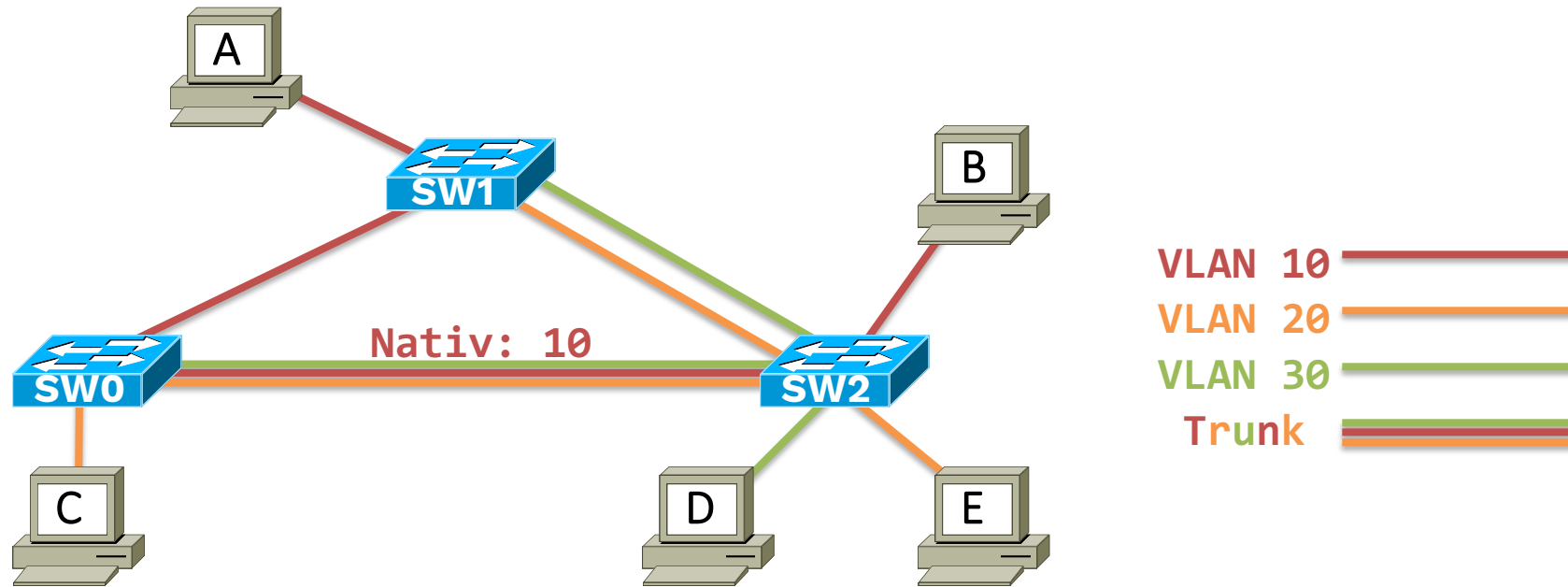
- O legătură trunk are un VLAN special numit VLAN nativ
- Cadrele aparținând VLAN-ului nativ circulă pe trunk în format Ethernet standard (nu 802.1q)
- Porturile de la capătul legăturii trebuie să aibă configurat același VLAN nativ



# Topologia exemplu

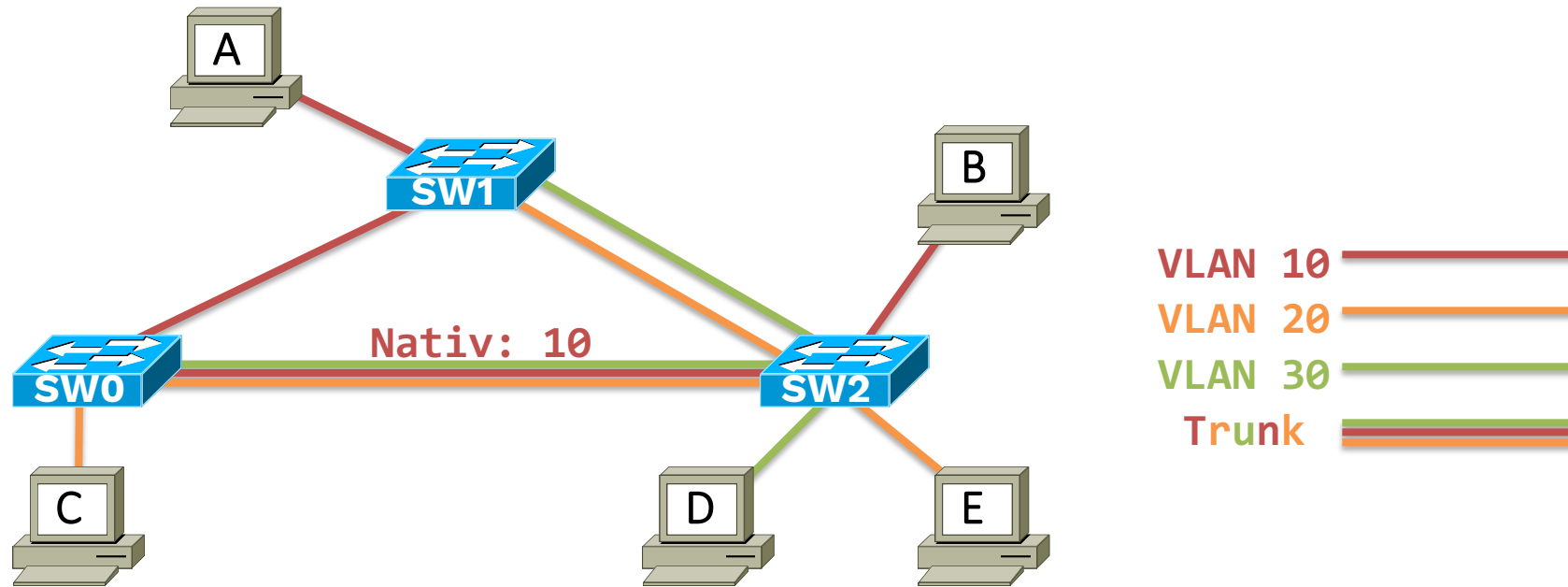


# Exercițiul 1: Broadcast A



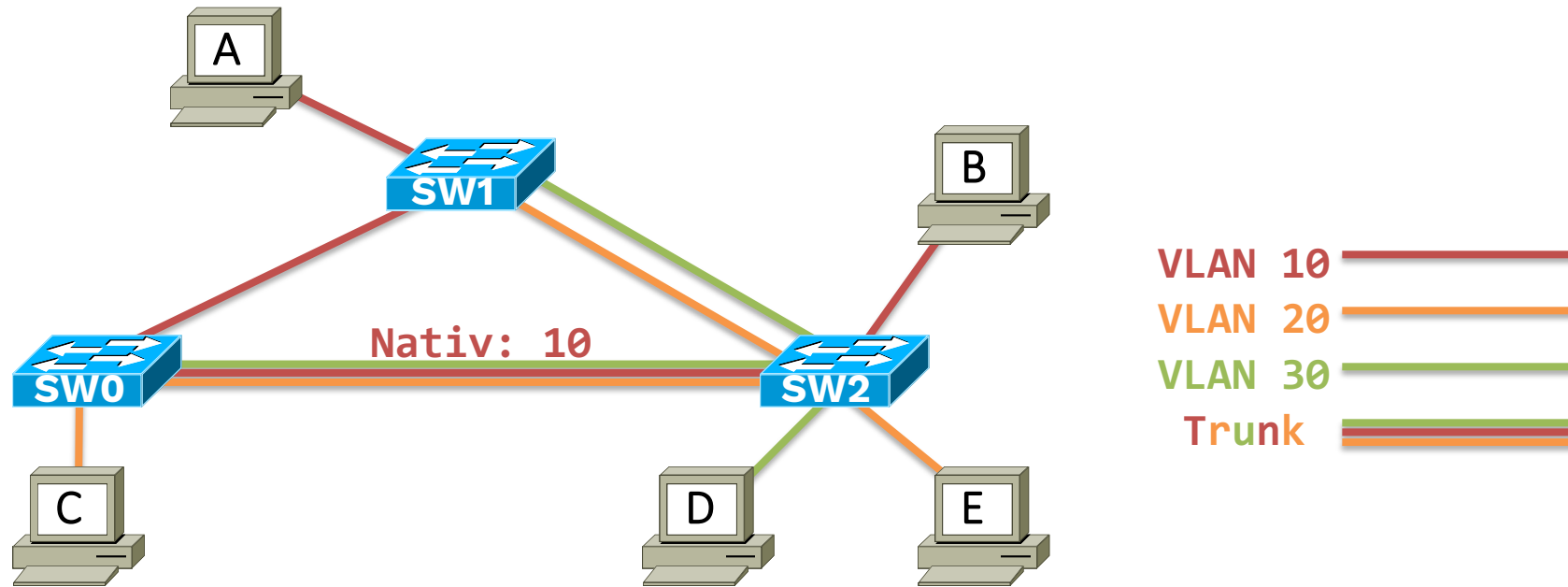
- A trimite un broadcast; la ce stații va ajunge respectivul broadcast?
- Pe ce cale ajunge la fiecare destinație?

# Exercițiul 1: Broadcast A



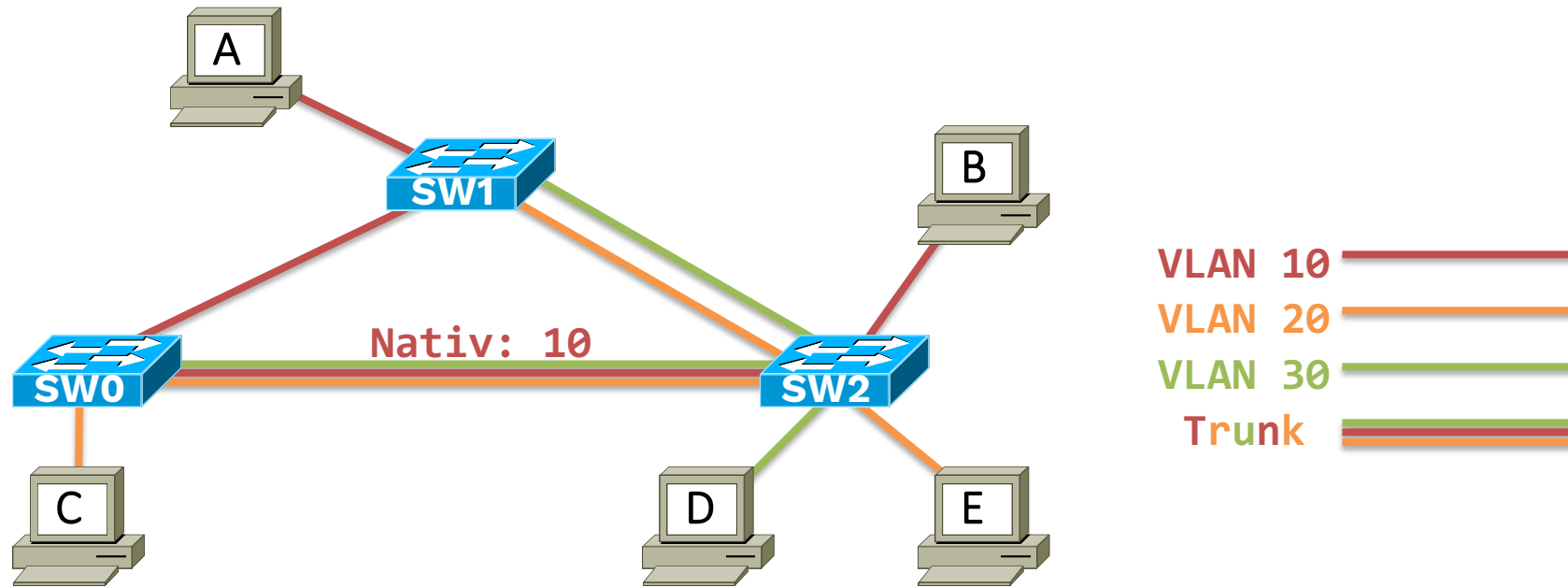
- A trimite un broadcast; la ce stații va ajunge respectivul broadcast?
  - R: B
- Pe ce cale ajunge la fiecare destinație?
  - R: A → SW1 → SW0 → SW2 → B

# Exercițiul 1: Broadcast A



- Ce format va avea broadcastul anterior pe legătura **SW0 - SW1**?
- Ce format va avea broadcastul anterior pe legătura **SW0 - SW2**?

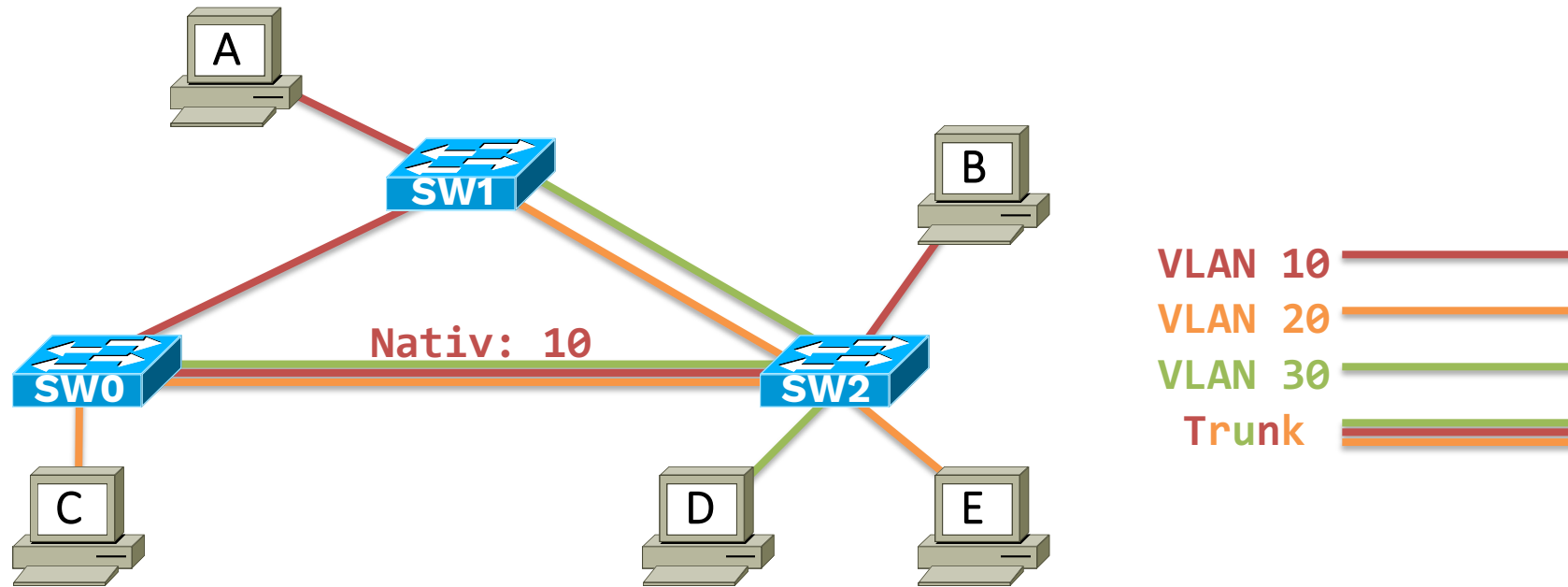
# Exercițiul 1: Broadcast A



- Ce format va avea broadcastul anterior pe legătura **SW0 - SW1**?
  - R: **Ethernet**
- Ce format va avea broadcastul anterior pe legătura **SW0 - SW2**?
  - R: **Ethernet**

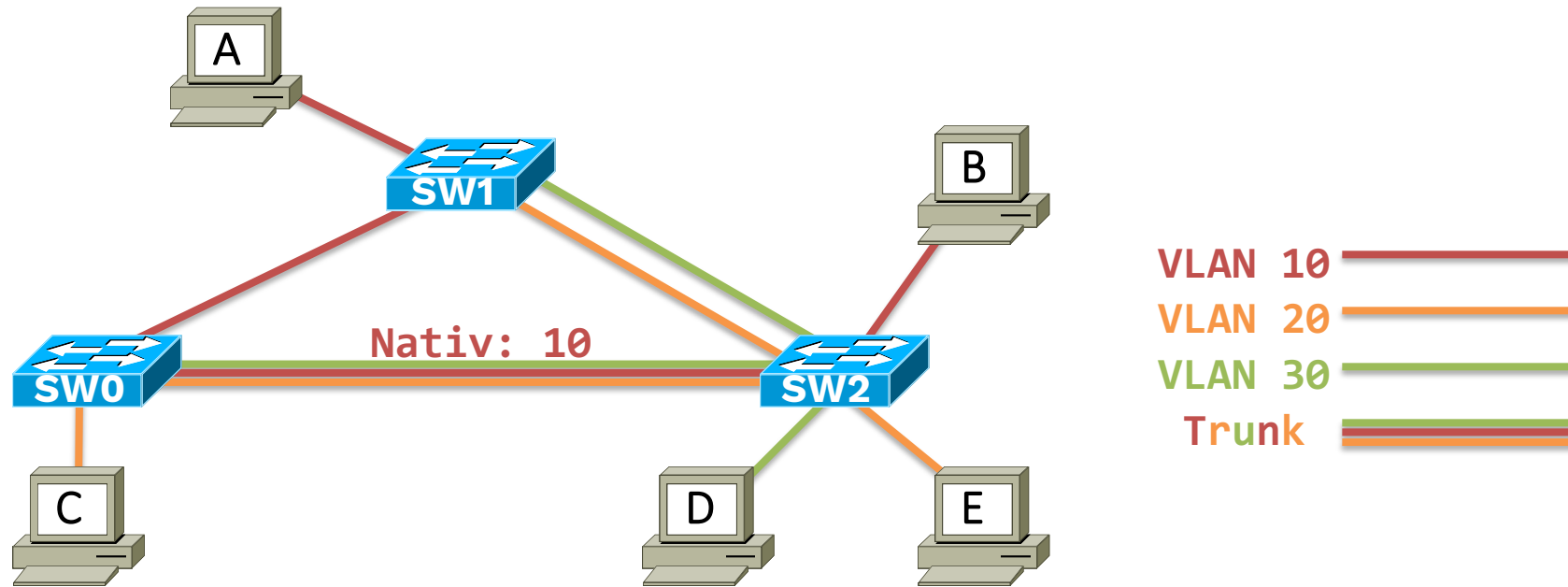


# Exercițiul 2: Unicast E → C



- Stația E trimite un unicast către stația C; toate switch-urile au tabela CAM vidă; la ce dispozitive de rețea va ajunge unicast-ul?
- Ce format va avea cadrul pe legătura **SW2 - SW1**? Dar pe legătura **SW0 - SW2**

# Exercițiul 2: Unicast E → C



- Stația E trimite un unicast către stația C; toate switch-urile au tabela CAM vidă; la ce dispozitive de rețea va ajunge unicast-ul?
  - R: **SW0, SW1, SW2, C** (switch-urile fac flood)
- Ce format va avea cadrul pe legătura **SW2 - SW1**? Dar pe legătura **SW0 - SW2**
  - R: **Ethernet** pe **SW2 - SW1**, **802.1q** pe **SW0 - SW2**

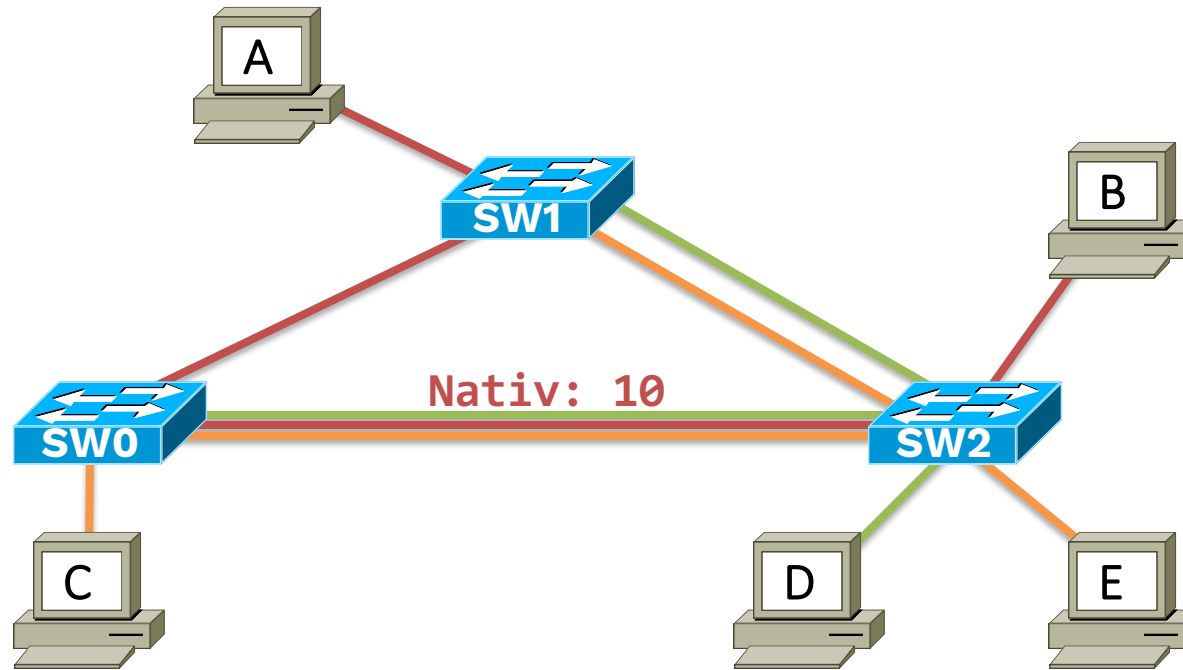
# Rutarea Inter-VLAN

Rutare clasică

Router-on-a-stick



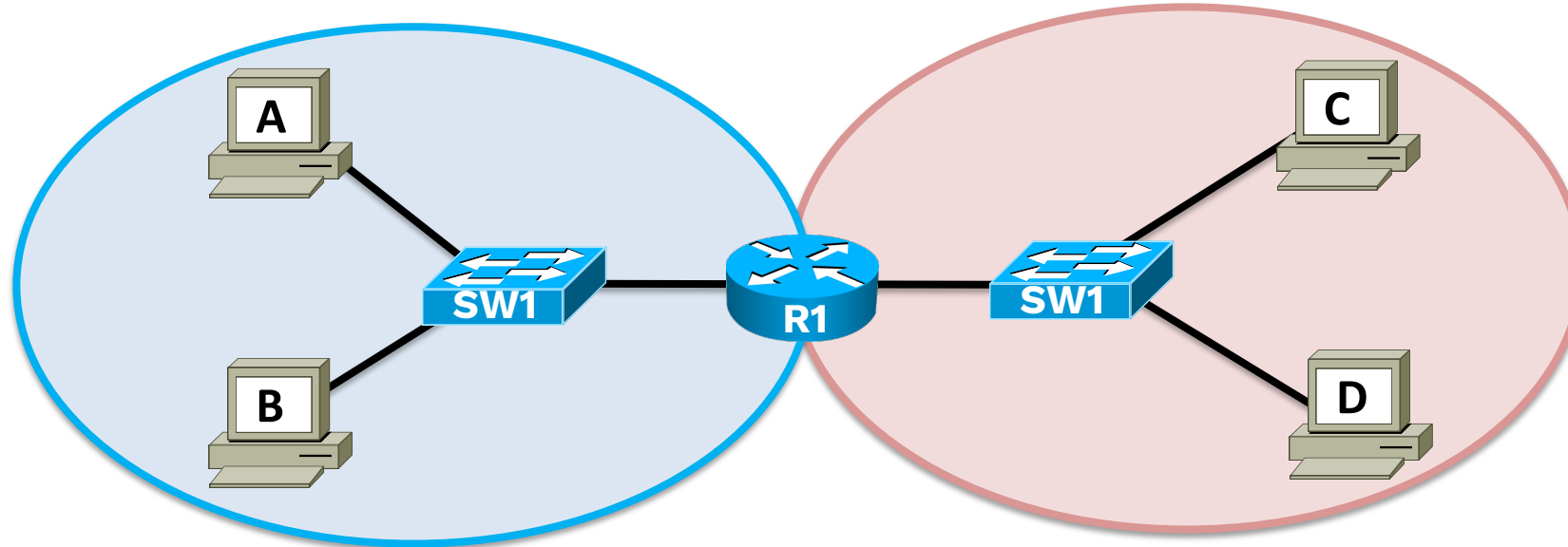
# Necesitatea rutării



- **A** vrea să comunice cu **E**; cum ar putea trimite un cadru către E în topologia de mai sus?
  - R: nu se poate, este necesar un **Ruter**

# Ruterul

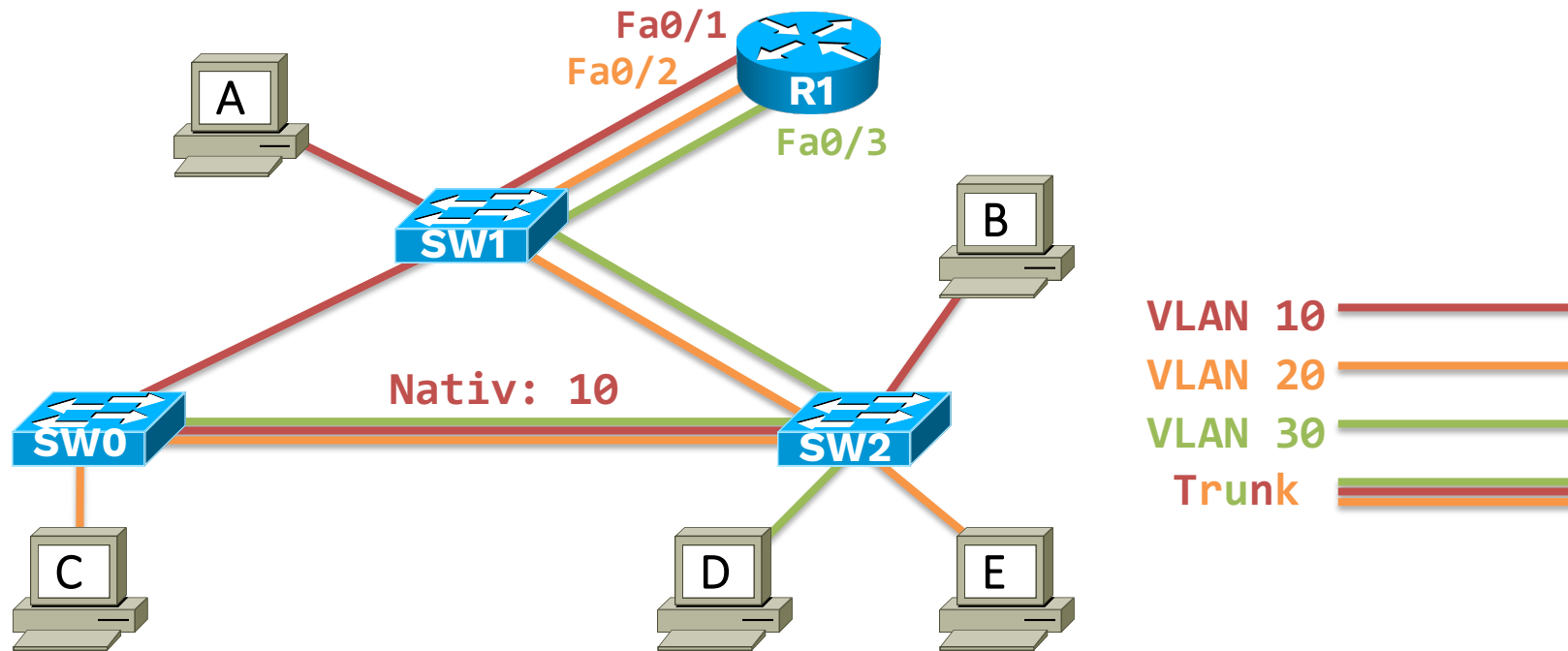
- Ruterul este un echipament ce funcționează la nivelul Rețea al stivei OSI
- Ruterul dirijează traficul între domenii de broadcast distincte



# Rutare Inter-VLAN

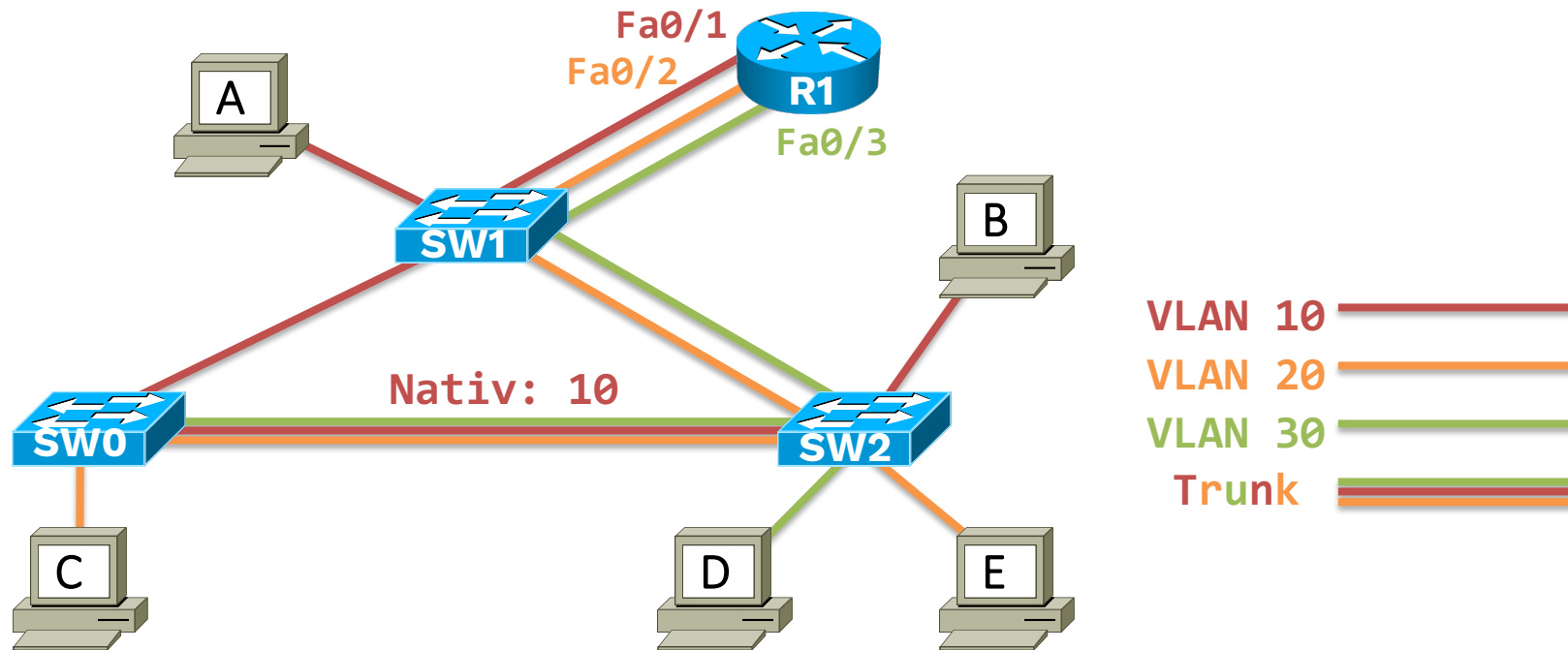
- Putem folosi un ruter pentru a asigura conectivitatea între VLAN-uri diferite
- Traficul va intra în ruter pe un VLAN și va ieși pe un altul
- Există două soluții:
  - Soluția “clasică”
  - Soluția “router-on-a-stick”

# Soluția clasică



- Folosește multiple interfețe pe ruter
  - fiecare interfață se va găsi într-un VLAN diferit

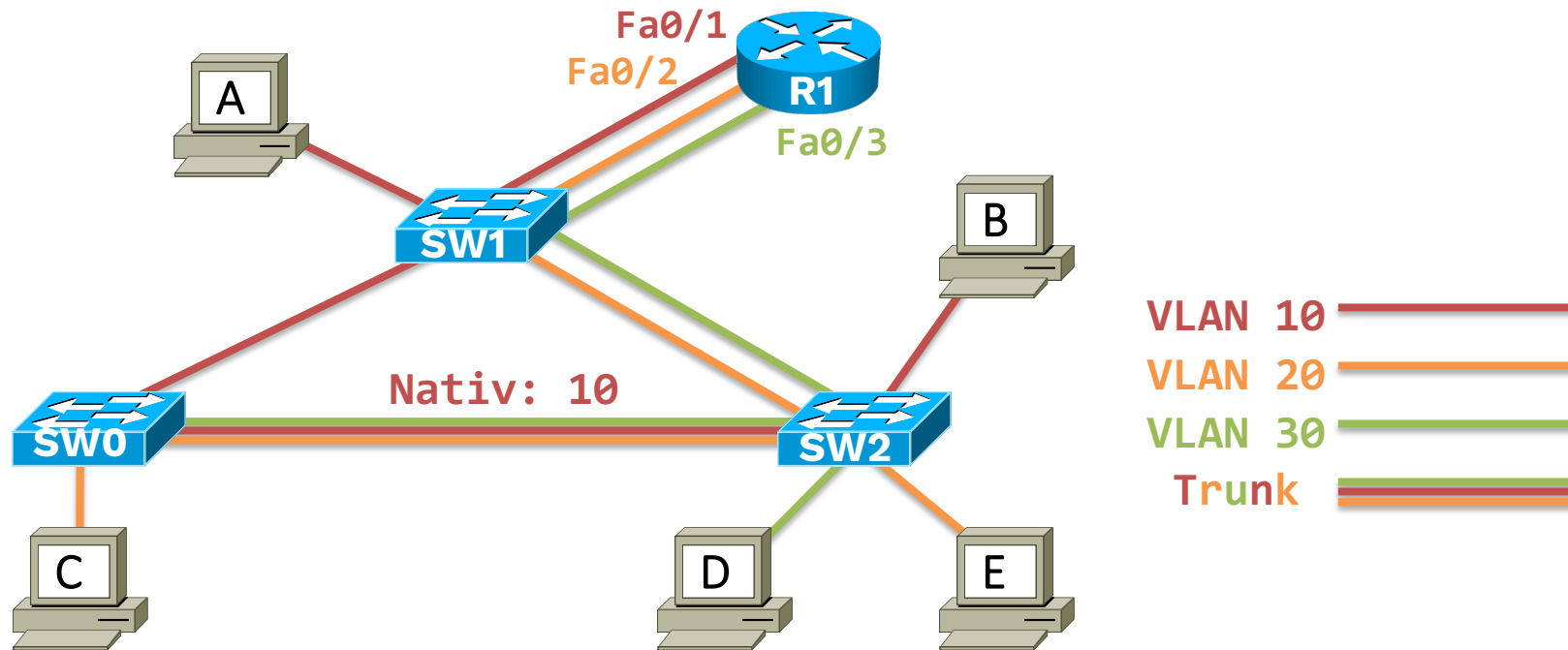
# Soluția clasică: Exemplu



- A îi trimite un cadru lui E; switch-urile au tabele CAM complete



# Soluția clasică: Exemplu

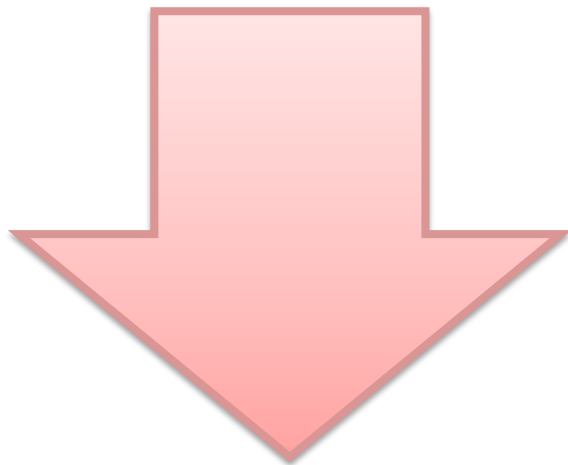
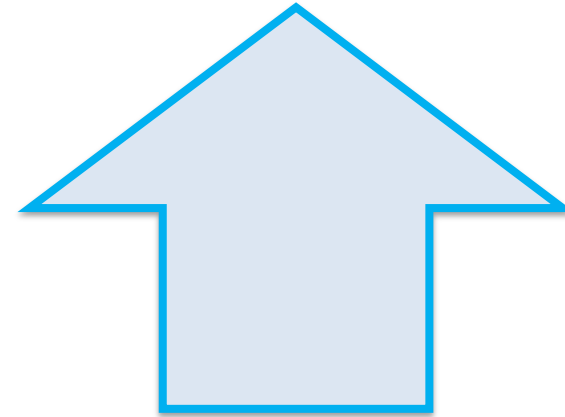


- A îi trimite un cadru lui E; switch-urile au tabele CAM complete
  - A → SW1 → Fa0/1 R1
  - Are loc procesul de rutare în R1: Fa0/1 R1 → Fa0/2 R1
  - Fa0/2 R1 → SW1 → SW2 → E

# Soluția clasică

## Avantaje:

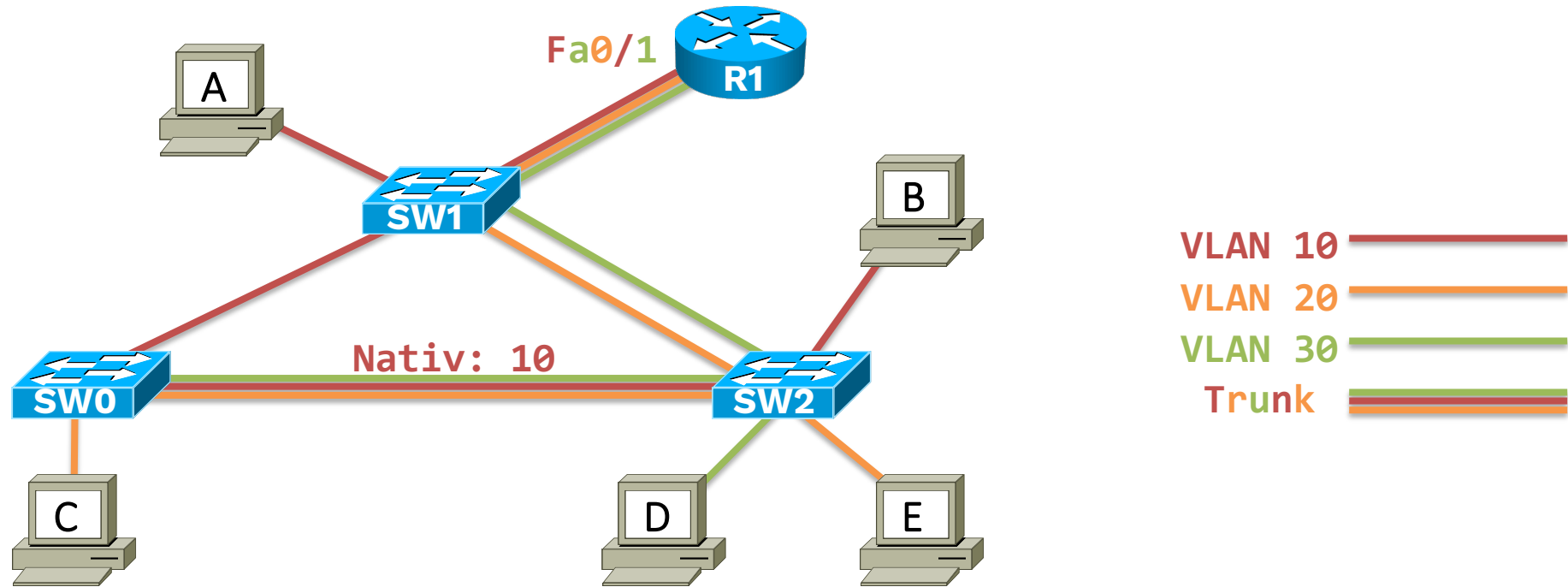
- Apartenența la VLAN-uri este transparentă ruterului
- Folosește eficient capacitatea de transfer a mediului



## Dezavantaje:

- Interfețele pe rutere sunt puține și abordarea consumă un număr mare de interfețe
- Este necesară o cantitate mare de cabluri pentru a realiza legăturile
- Nu scalează

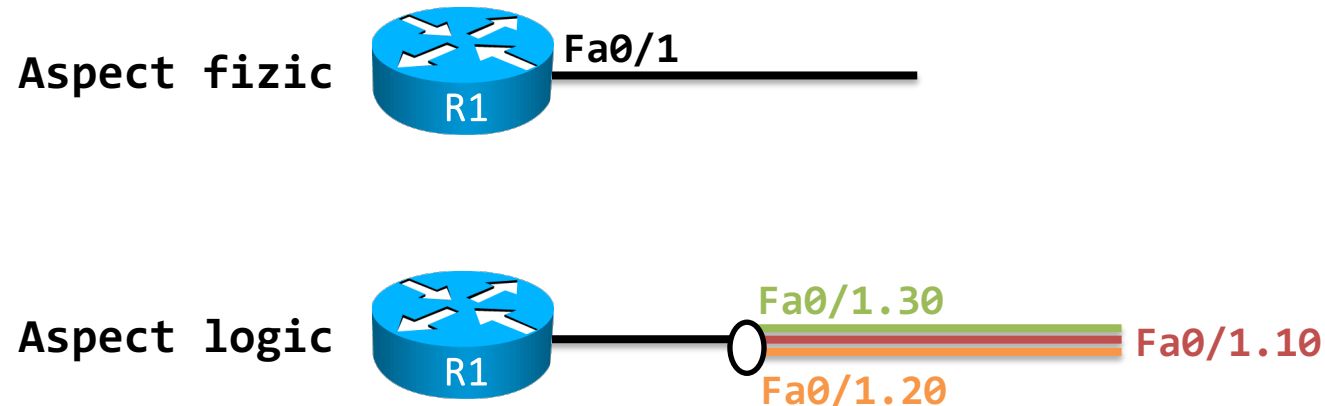
# Soluția Router-on-a-stick



- Folosește o singură interfață fizică
  - Interfața fizică este separată în mai multe interfețe logice numite **subinterfețe**

# Soluția Router-on-a-stick: Subinterfețe

- O interfață fizică poate fi împărțită în mai multe subinterfețe
- Abordarea router-on-a-stick presupune crearea unei subinterfețe pentru fiecare VLAN
- Fiecare subinterfață va avea adresa sa proprie de nivel 3
- Subinterfețele sunt identificate prin id-ul de subinterfață (de exemplu Fa0/1 poate avea subinterfața cu id-ul 42: Fa0/1.42)



# Soluția Router-on-a-stick: Subinterfețe

- Legătura dintre switch și ruter va fi configurată ca trunk
- Fiecare subinterfață trebuie informată că traficul va veni în format 802.1q și nu Ethernet
- Când se configurează încapsularea 802.1q se asociază și VLAN-ul corespunzător subinterfeței

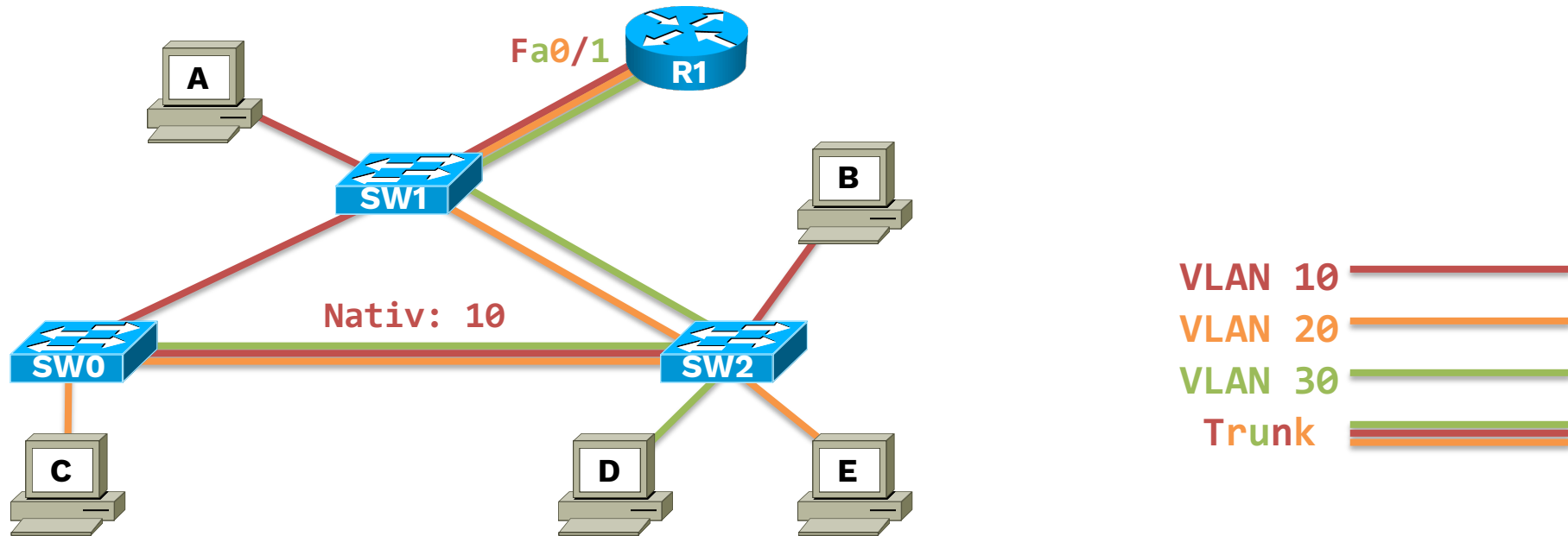


Fa0/1.30 - 802.1q; VLAN 30

Fa0/1.10 - 802.1q, VLAN 10

Fa0/1.20 - 802.1q; VLAN 20

# Soluția Router-on-a-stick: Exemplu

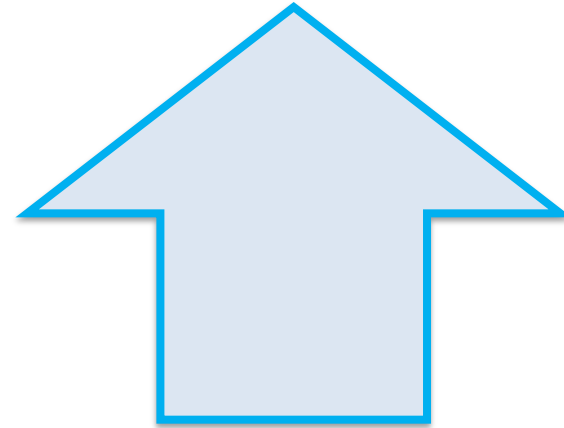


- A îi trimite un cadru lui E; switch-urile au tabele CAM complete
  - A → SW1 → Fa0/1 R1
  - R1 vede în tag-ul 802.1q că VLAN-ul e 10 și primește pe Fa0/1.10
  - Are loc procesul de rutare în R1: Fa0/1.10 → Fa0/1.20
  - R1 trimite pe Fa0/1.20 cadrul în format 802.1q cu VLAN-ul 20
  - Fa0/1 R1 → SW1 → SW2 → E

# Soluția Router-on-a-stick

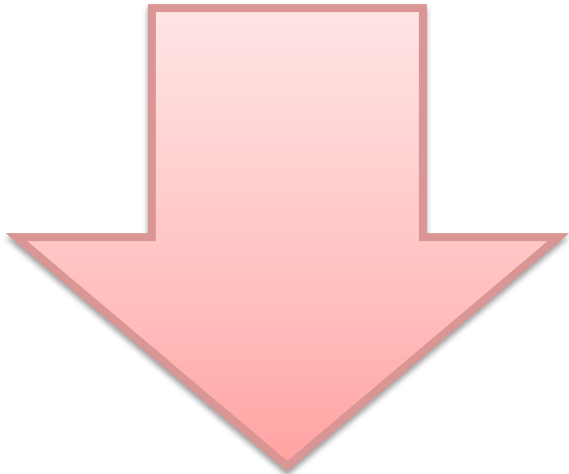
## Avantaje:

- Este utilizată o singură interfață a ruterului
- Este necesar un număr redus de legături
- Scalează bine



## Dezavantaje:

- Lățimea de bandă a interfeței fizice este împărțită între cele logice (poate apărea un bottleneck)
- Funcționalitatea nu este disponibilă pe toate ruterele
- VLAN-urile nu mai sunt transparente ruterului



# Spanning Tree Protocol

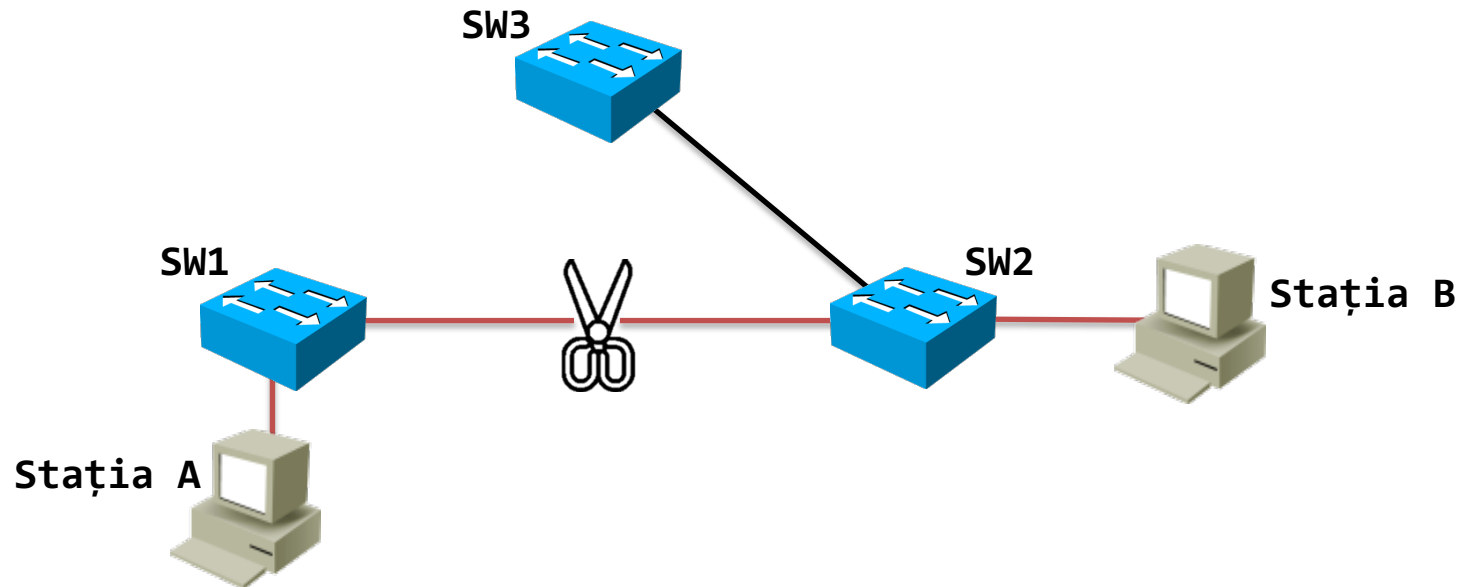




# ARP

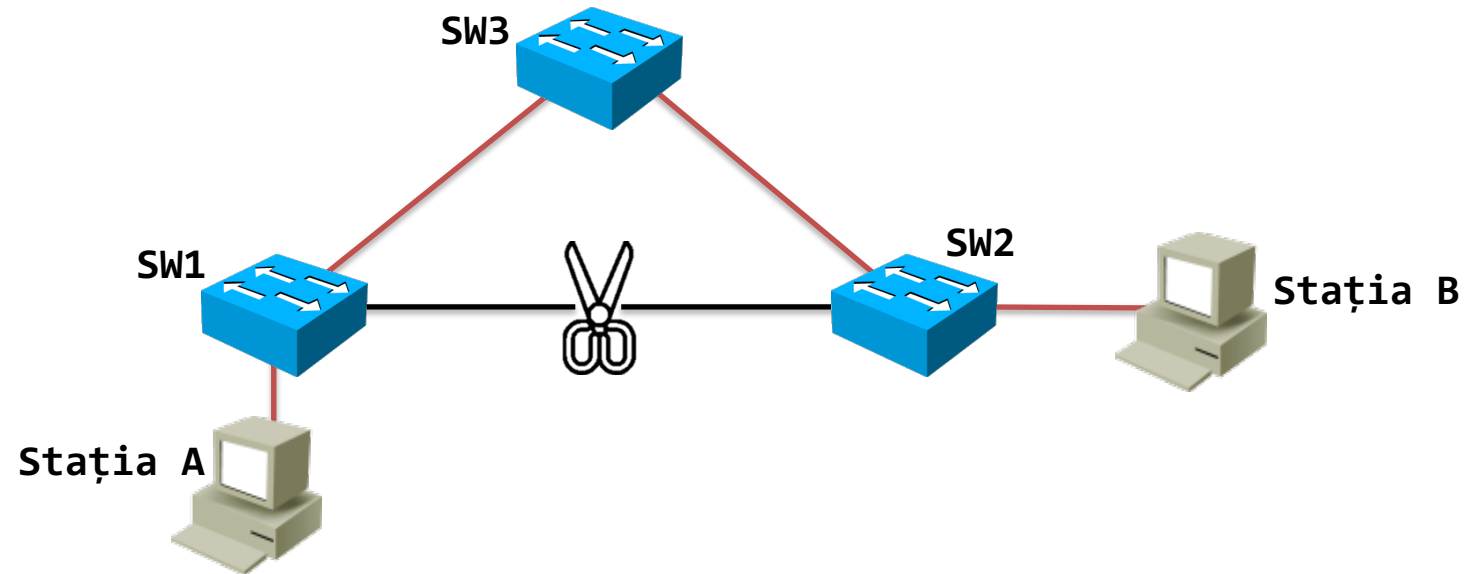
- Când o stație vrea să trimită un pachet într-o rețea Ethernet, aceasta dispune de adresa IP dar nu și de adresa MAC
- Pentru a putea transmite cadrul și a fi acceptat la destinație este necesară determinarea acestei adrese
- Protocolul care determină adresa MAC pornind de la adresa IP poartă numele de ARP (Address Resolution Protocol)

# Redundanța în rețele



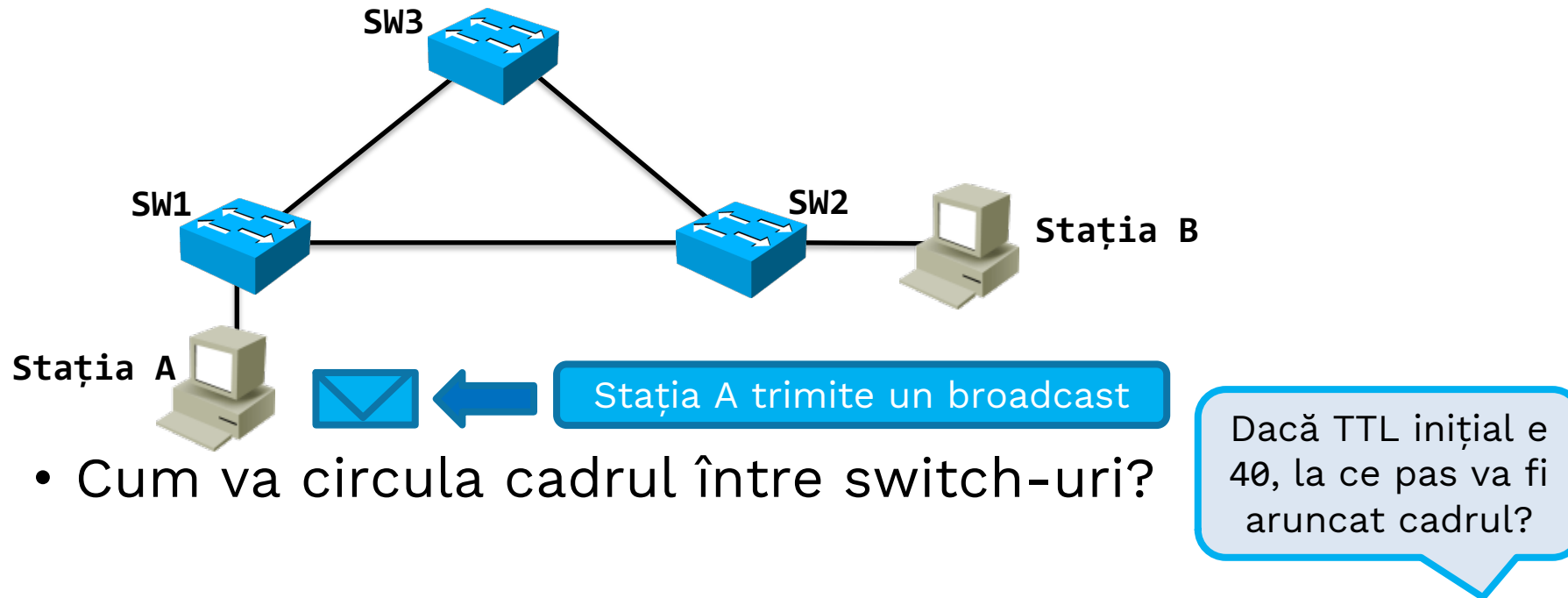
- Dacă legătura dintre SW1 și SW2 cade, stațiile nu mai pot comunica între ele
- Soluția este introducerea unei legături alternative ca backup în cazul căderii legăturii principale

# Redundanța în rețele



- Redundanța se poate implementa la niveluri diferite
  - La nivel de link (2 uplink-uri)
  - La nivel de dispozitiv de nivel 2 (multiple căi prin bucle fizice nivel 2)
  - La nivel de dispozitiv de nivel 3 (multiple gateway-uri – HSRP, VRRP)

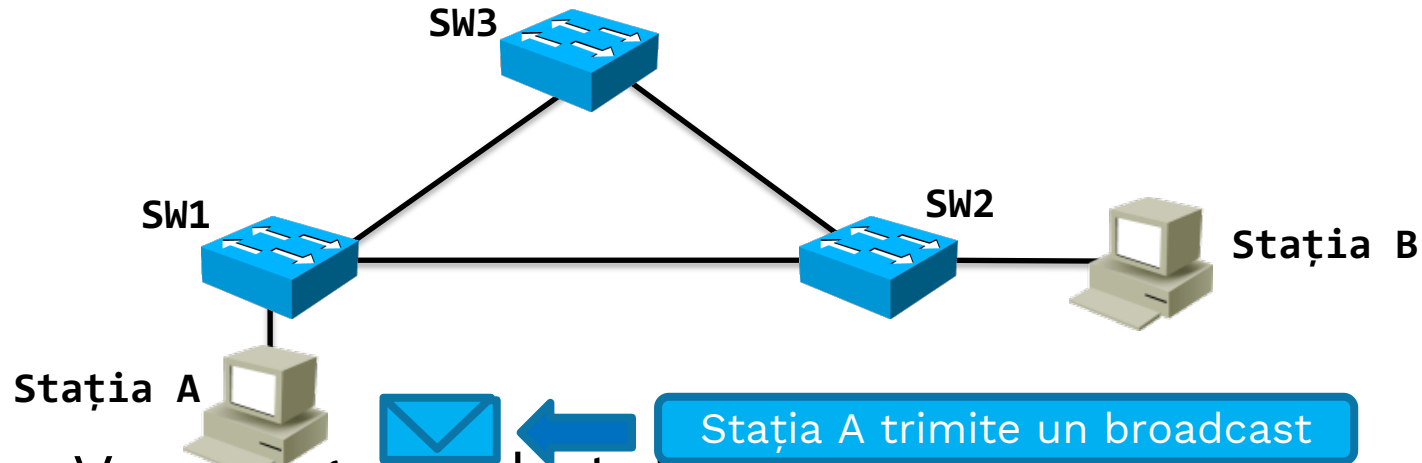
# Probleme introduse de redundanță - 1



- Cum va circula cadrul între switch-uri?

T	1	2	3	4	5	6	7
Cadre	A → SW1	SW1 → SW3 SW1 → SW2	SW3 → SW2 SW2 → SW3 SW2 → B	SW2 → SW1 SW2 → B SW3 → SW1	SW1 → SW3 SW1 → SW2 SW1 → A SW1 → A	SW3 → SW2 SW2 → SW3 SW2 → B	...

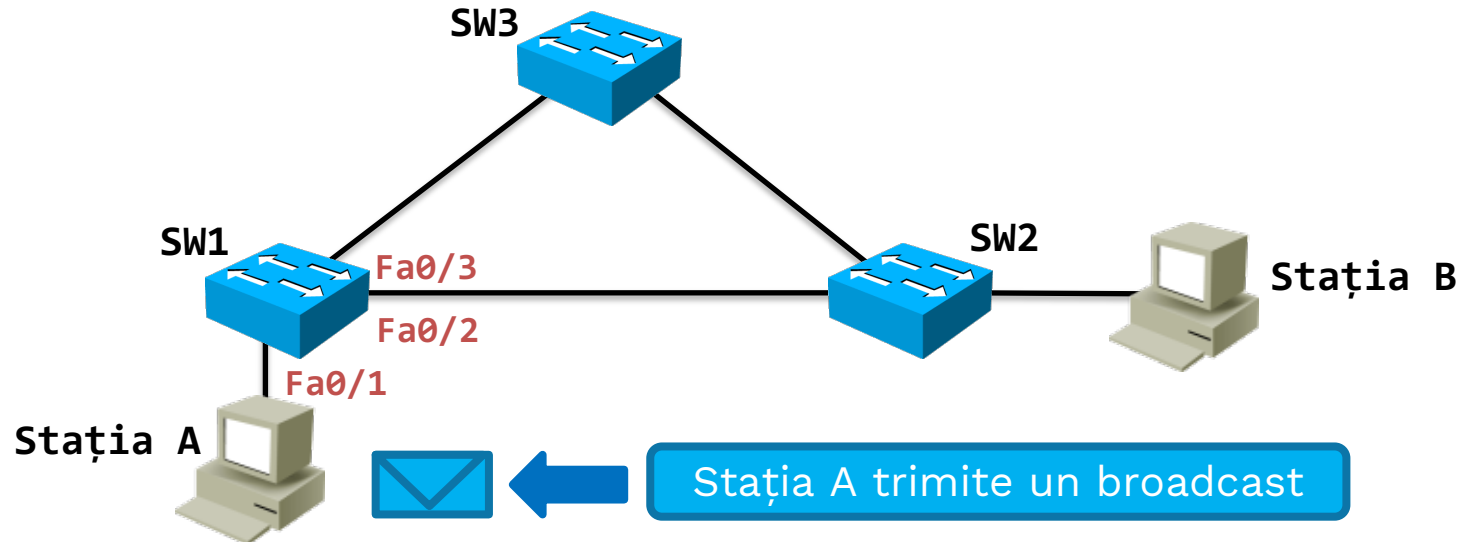
# Probleme introduse de redundanță - 2



- Va ajunge pachetul la destinație?
  - R: Da, de o infinitate de ori.

T	1	2	3	4	5	6	7
Cadre	A → SW1	SW1 → SW3 SW1 → SW2	SW3 → SW2 SW2 → SW3 SW2 → B	SW2 → SW1 SW2 → B SW3 → SW1	SW1 → SW3 SW1 → SW2 SW1 → A SW1 → A	SW3 → SW2 SW2 → SW3 SW2 → B	...

# Probleme introduse de redundanță - 3



- După câteva secunde, pe ce port crede SW1 că este stația A?

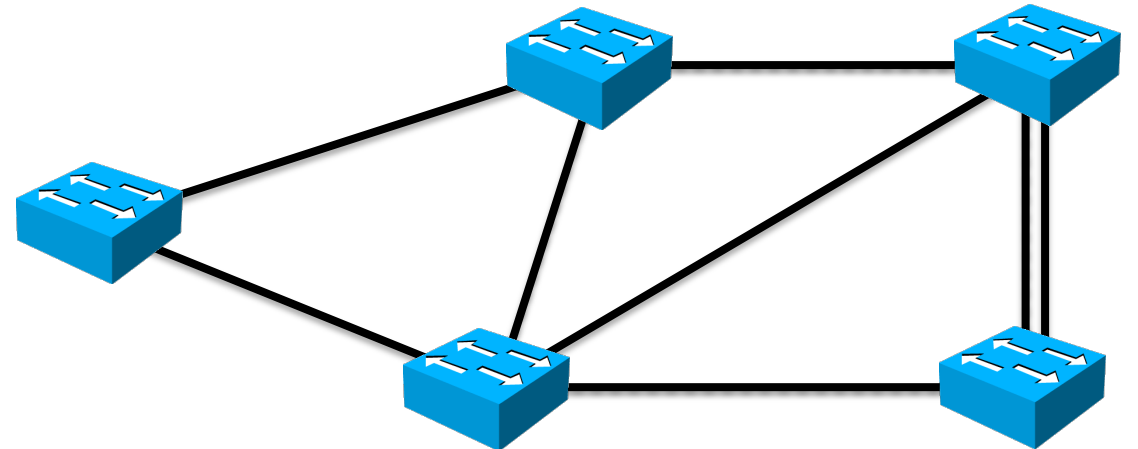
T	1	2	3	4	5	6	7
Cadre	A → SW1	SW1 → SW3 SW1 → SW2	SW3 → SW2 SW2 → SW3 SW2 → B	SW2 → SW1 SW2 → B SW3 → SW1	SW1 → SW3 SW1 → SW2 SW1 → A SW1 → A	SW3 → SW2 SW2 → SW3 SW2 → B	...

# Motivația pentru STP

- Avem nevoie de redundanță în rețea
  - ... dar creăm bucle (fizice și logice)
- Un broadcast storm este cauzat de buclele logice (din cauza modului în care funcționează switching-ul într-o buclă fizică)
  - trebuie deci eliminate buclele logice
- Ideea protocolului STP:
  - se acceptă existența unei bucle fizice (redundanță)
  - închiderea temporară a unei bucle logice prin închiderea la nivel logic a unui port din buclă
  - deschiderea portului blocat în cazul în care un uplink cedează

# STP

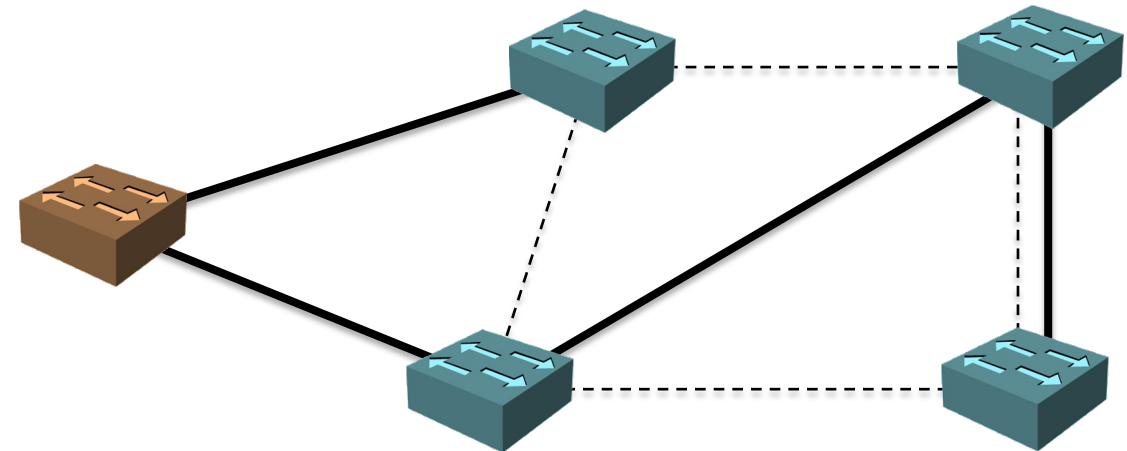
- Spanning Tree Protocol
- Specificat în standardul **802.1d**
- Operează pe o rețea de switch-uri
- Elimină buclele din rețea prin închiderea unor porturi
- Algoritmul STP poartă numele de **STA** (Spanning Tree Algorithm)
- Operație similară cu determinarea arborelui de acoperire pe un graf





# Rolurile switch-urilor

- În terminologia STP, switch-ul poartă numele de bridge
- Există două roluri pentru switch-uri:
  - **Root bridge** – rădăcina arborelui de switch-uri
  - **Non-root bridge** – toate celelalte switch-uri



# Rolurile porturilor

- Există trei roluri pentru porturi:
  - ▲ • **Designated port** – trimite și primește trafic de date
  - ▲ • **Root port** – trimite și primește trafic de date reprezintă calea cea mai eficientă spre root bridge
  - • **Blocked port** – nu trimite și nu primește trafic de date
- Pe o legătură, există următoarele două perechi de roluri:
  - **Designated – Root:**
    - Dacă legătura face parte din arborele de acoperire
  - **Designated – Blocked:**
    - Dacă legătura nu face parte din arborele de acoperire

# Costurile legăturilor

Costul unei muchii din graful STA este dependent de lățimea de bandă a legăturii respective.

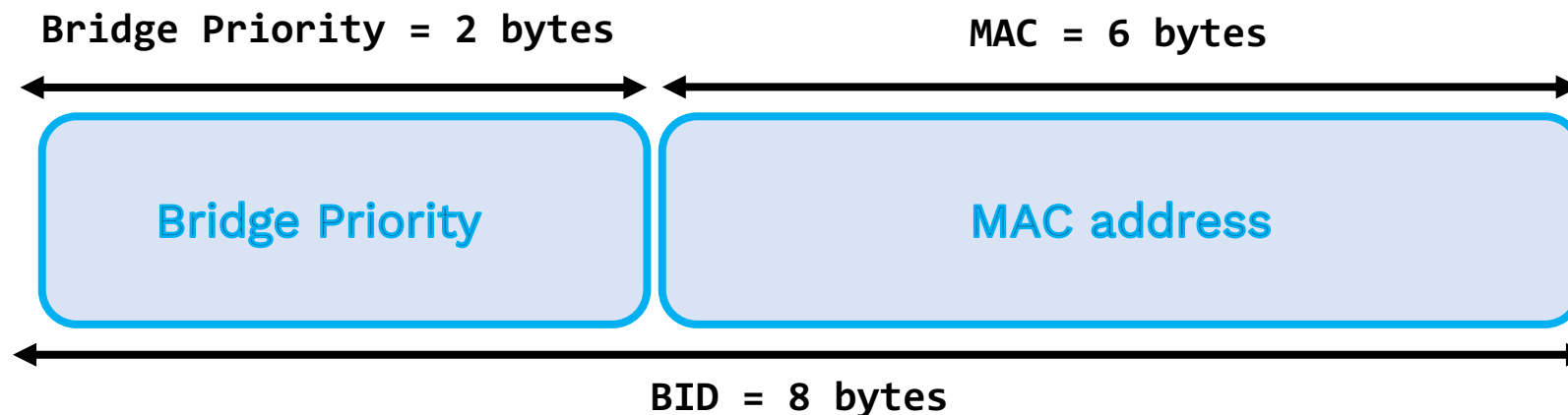
Lățime de bandă	Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Lățime de bandă	Cost
10 Mbps	2,000,000
100 Mbps	200,000
1 Gbps	20,000
10 Gbps	2,000

În cazul unor switch-uri cu legături mult mai rapide, se pot folosi alte sisteme de costuri.

# Bridge ID

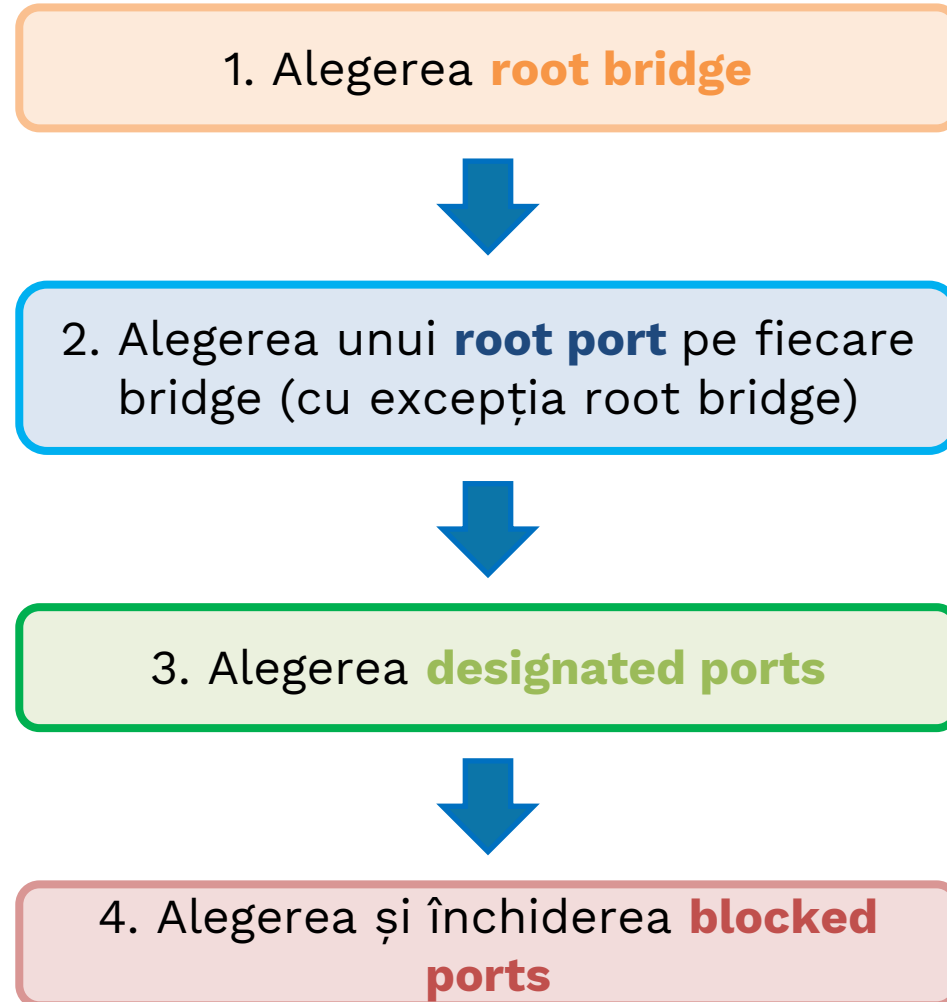
- Fiecare switch are un ID unic (BID)
- Valoare pe **64 biți**
  - 16 biți **prioritatea**
  - 48 biți **adresa MAC**
- Prioritatea este implicit 32768
- Switch-ul cu BID-ul cel mai mic va deveni **root bridge**



# BPDU

- Mesajele folosite de STP pentru a comunica informații între bridge-uri
- Transmise o dată la două secunde pe toate porturile
- Informații transmise:
  - root bridge ID
  - cost până la root bridge
  - bridge ID
  - port ID
- Observație: **blocked ports** încă primesc BPDU-uri

# Pașii STA



# Pasul 1 – Alegerea Root Bridge

- Bridge-urile trimit BPDU-uri până când toate cunosc cel mai mic BID din rețeaua de bridge-uri
- Bridge-ul cu ID-ul minim devine Root Bridge
- Cine ar deveni root bridge în fiecare din situațiile următoare?

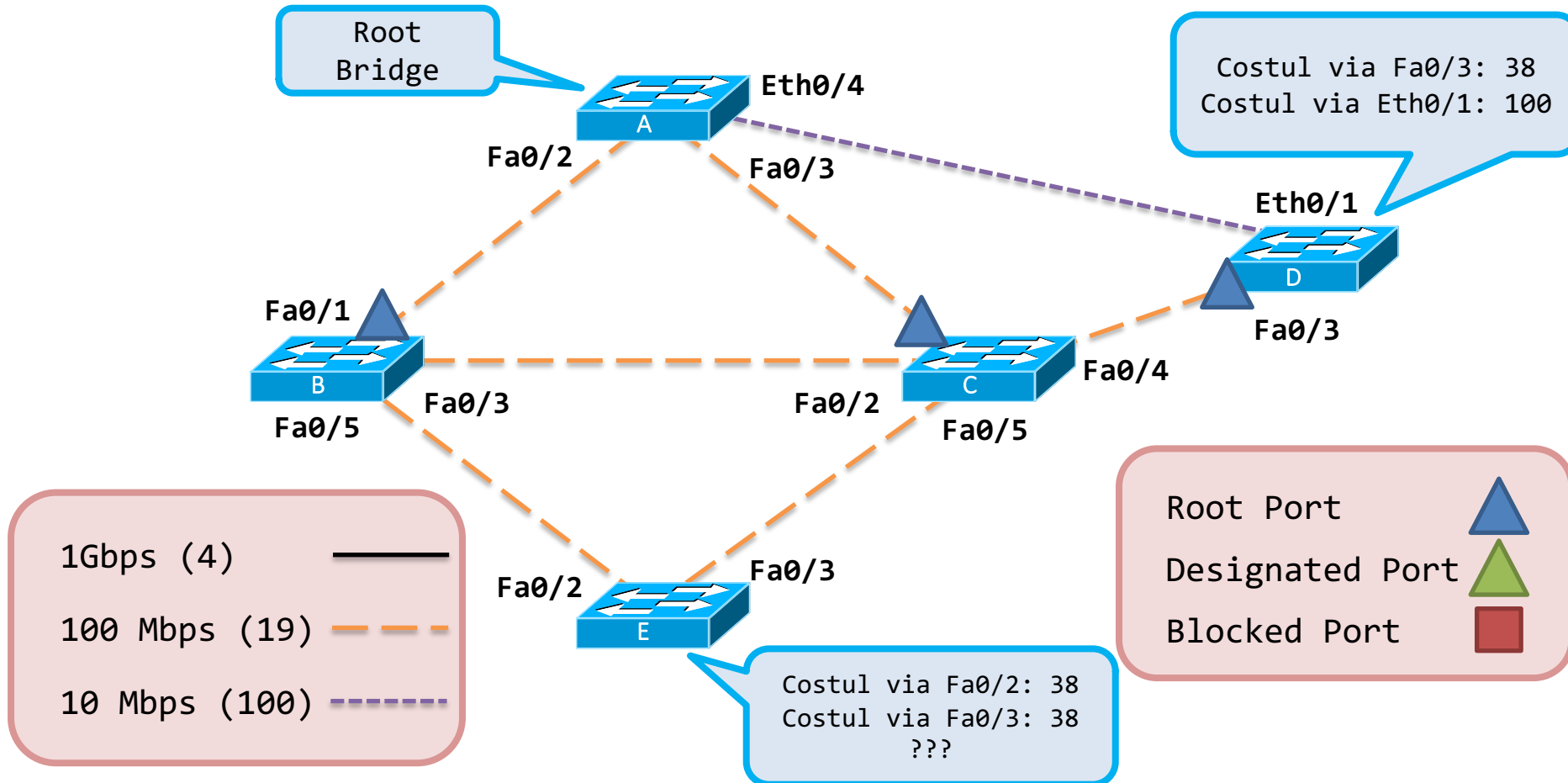
Nume	Prioritate	MAC
A	32768	00E0.A3C9.6AB8
B	32768	0001.97DA.86E8
C	32768	00D0.BC0C.844D
D	32768	0003.E496.C80E

Nume	Prioritate	MAC
A	16384	00E0.A3C9.6AB8
B	32768	0001.97DA.86E8
C	8192	00D0.BC0C.844D
D	16384	0003.E496.C80E
E	8192	0060.2F07.EB2B
F	8192	0060.7058.D0A5

- **R: B** în prima situație. **E** în a doua situație.

# Pasul 2: Root ports

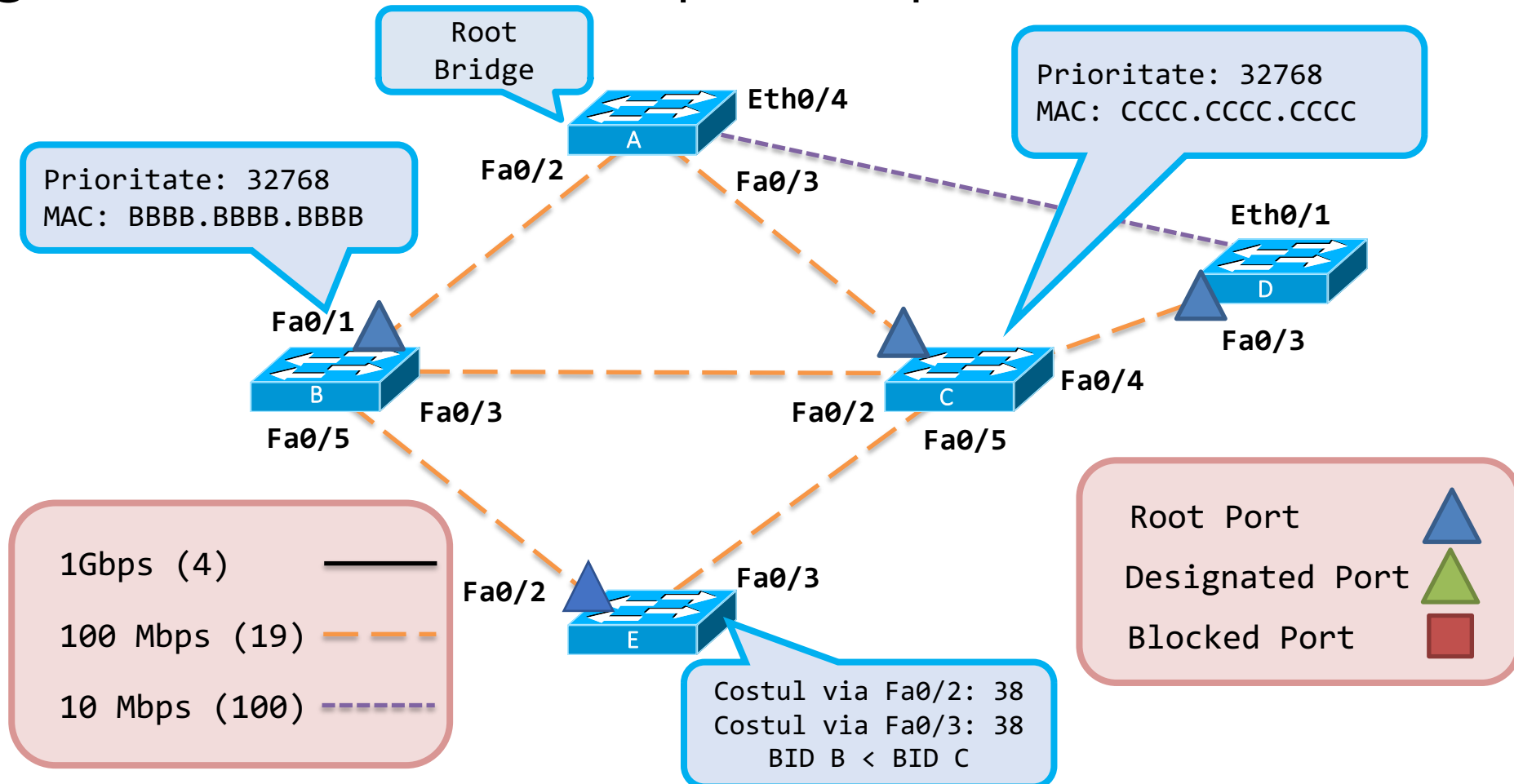
- Fiecare switch non-root trebuie să aibă un **root port**





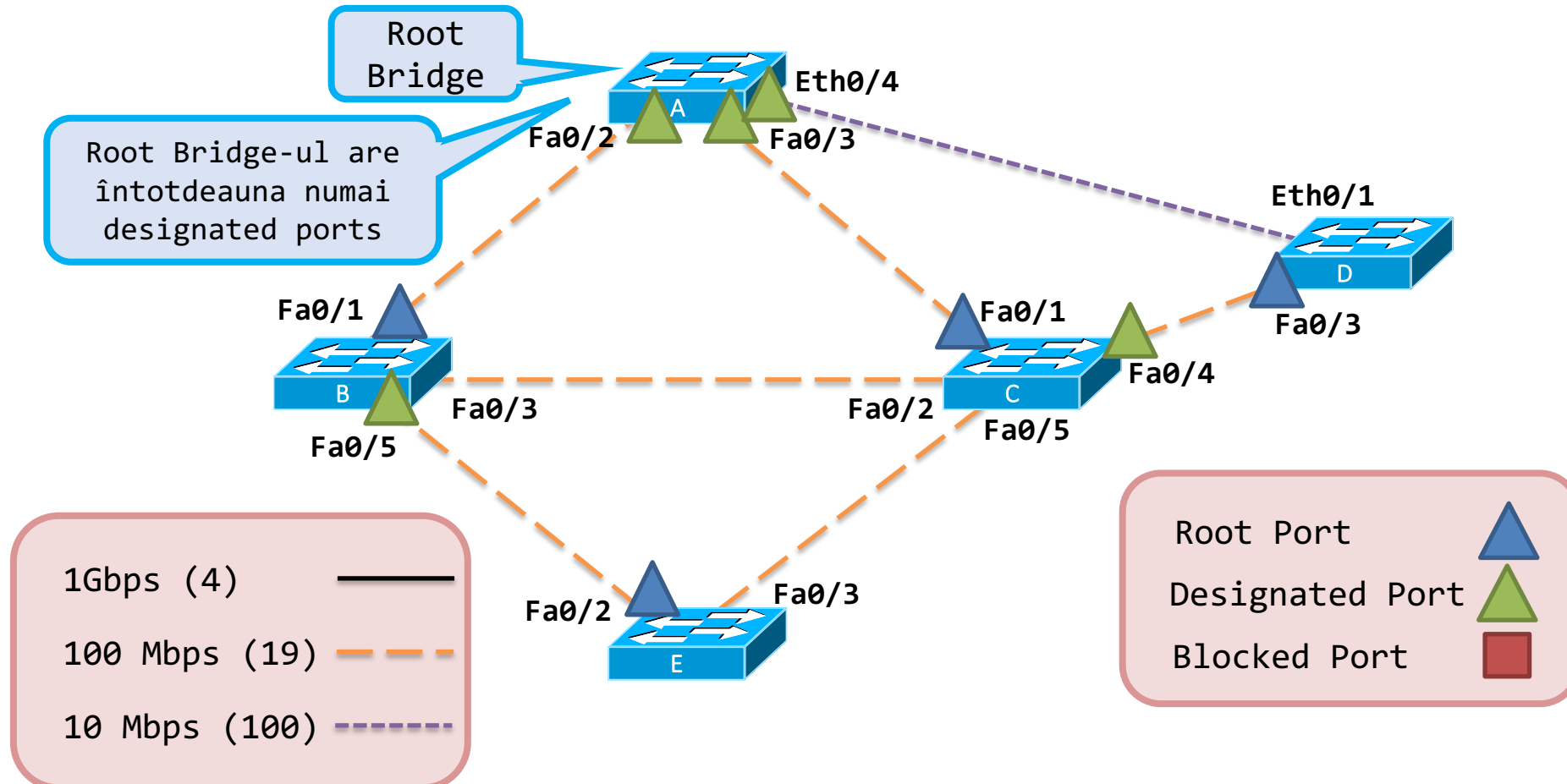
# Pasul 2: tiebreaker

- Bridge-ul E va decide root port-ul pe baza BID-ului vecinului



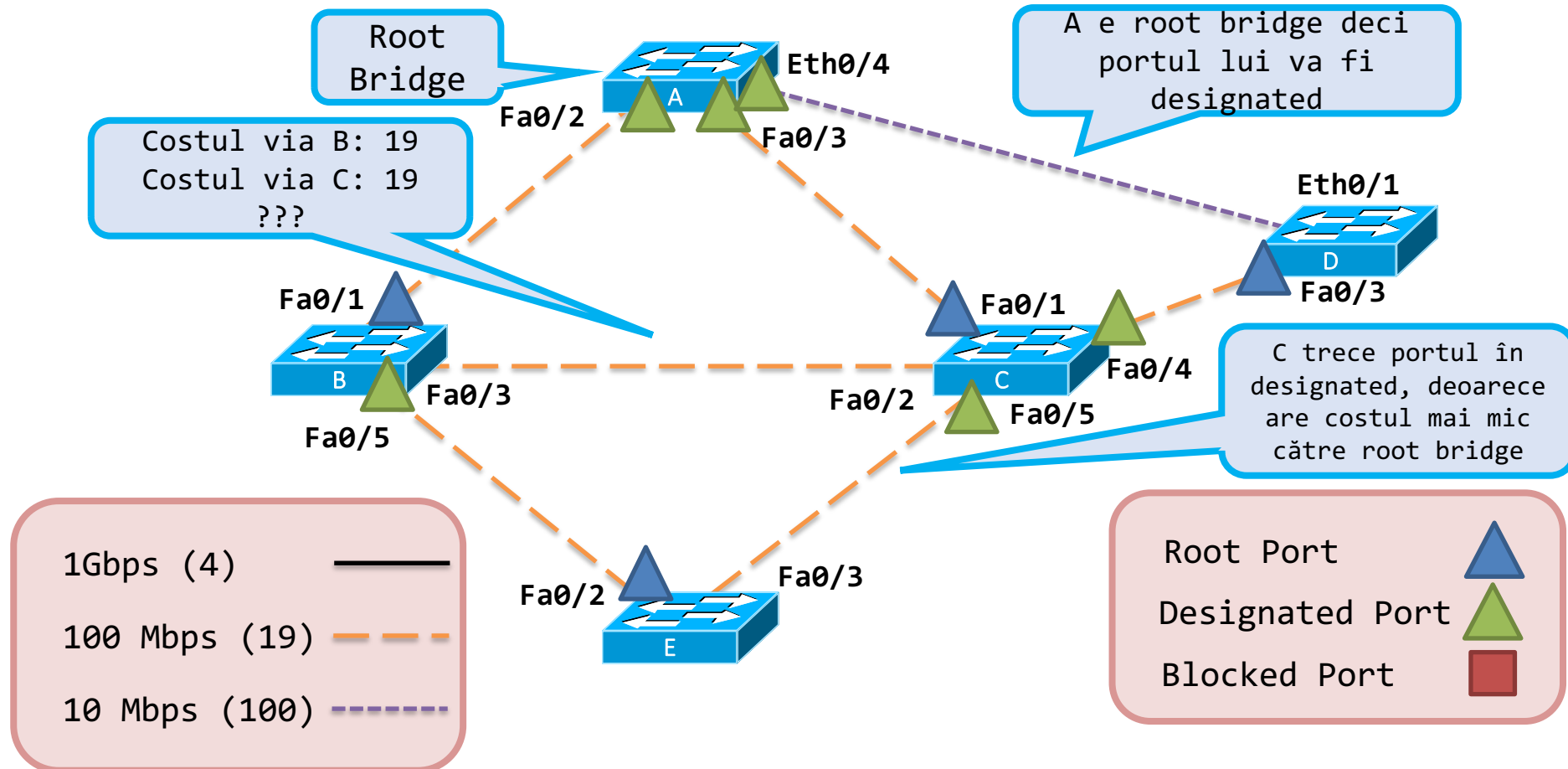
# Pasul 3: Designated ports

- Un **root port** este cuplat pe link cu un **designated port**



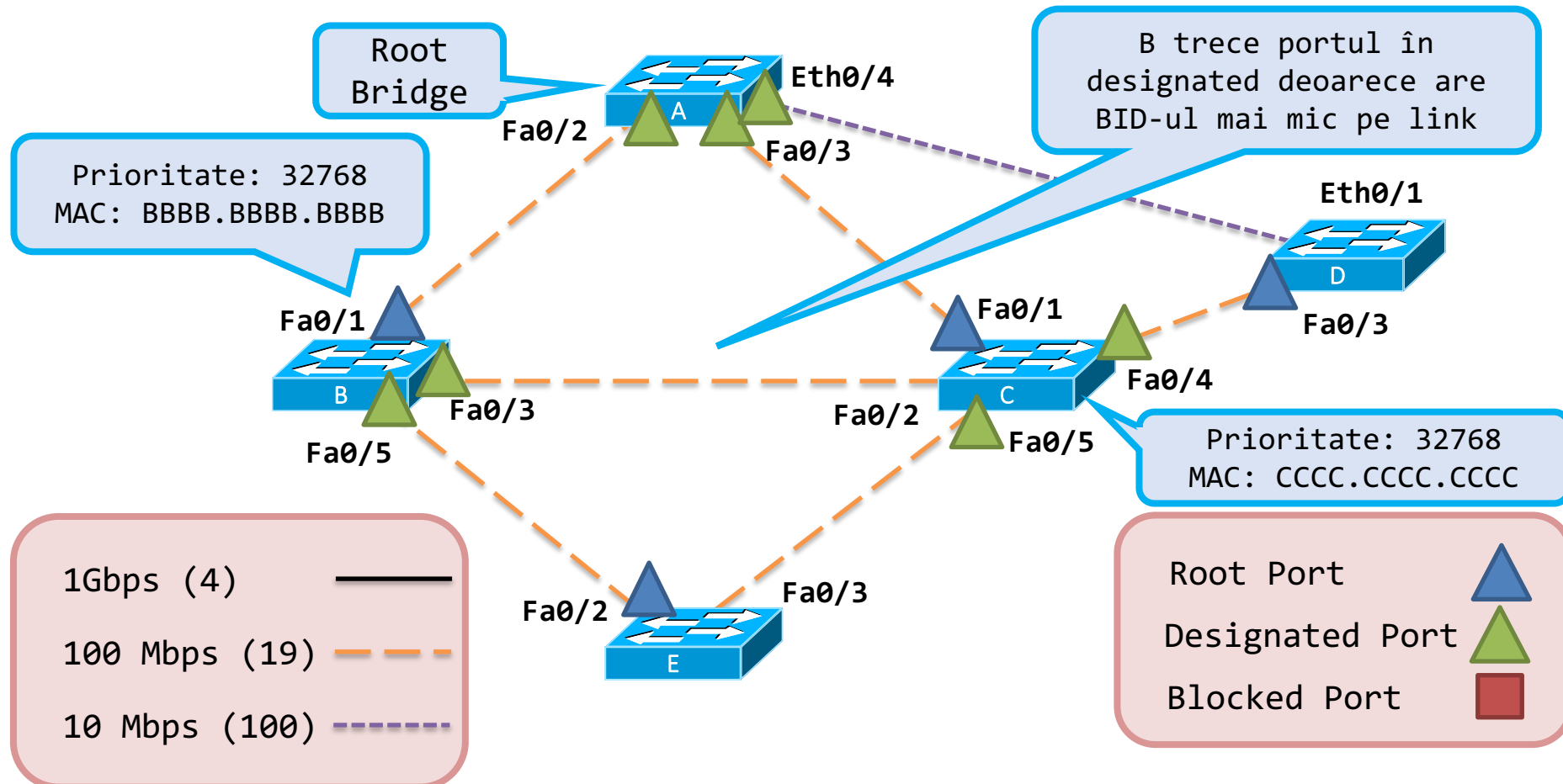
# Pasul 3: Designated ports - tiebreaker

- Pe fiecare legătură trebuie să existe un designated port



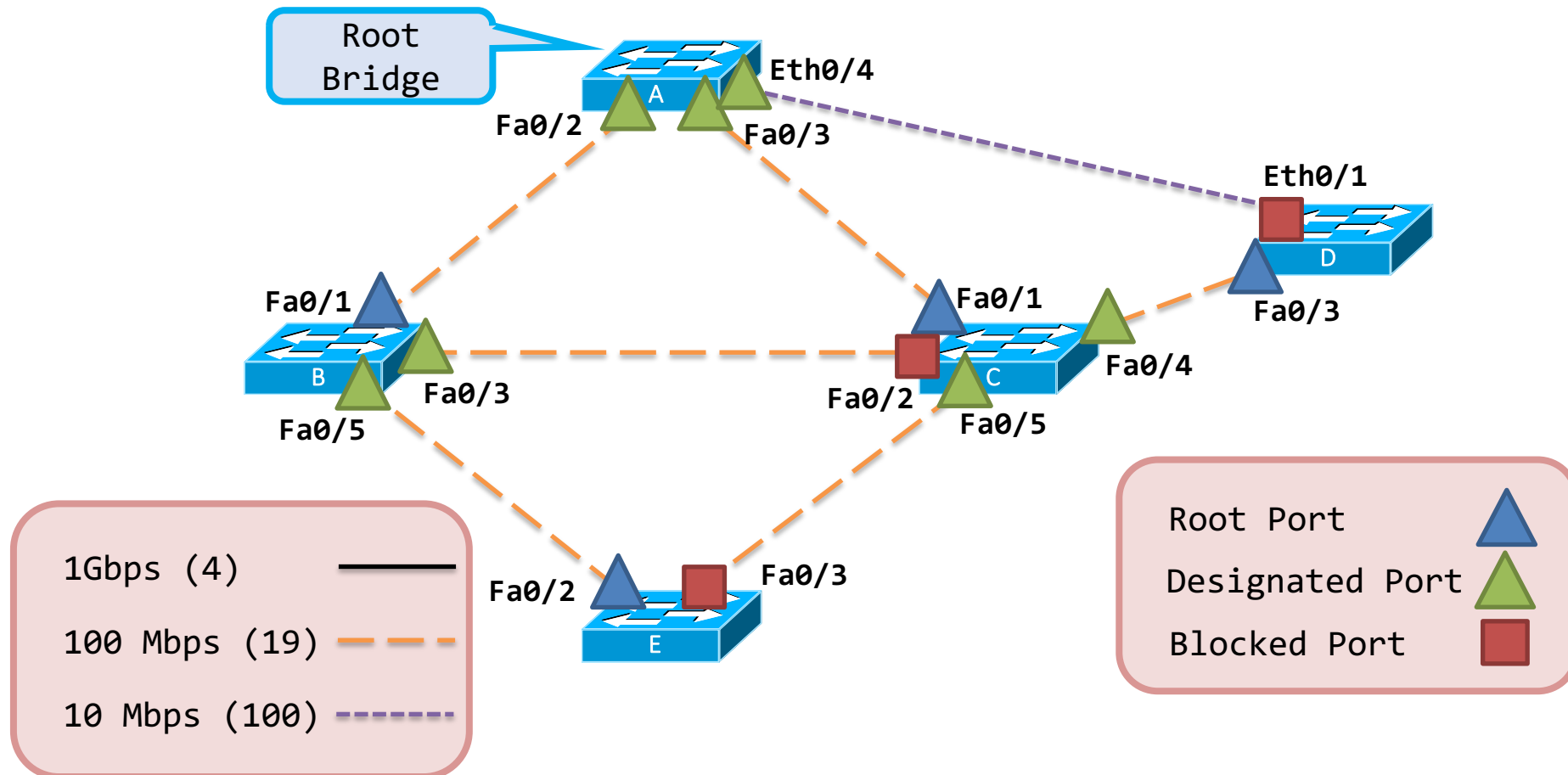
# Pasul 3: Designated ports - tiebreaker

- Pe fiecare legătură trebuie să existe un designated port

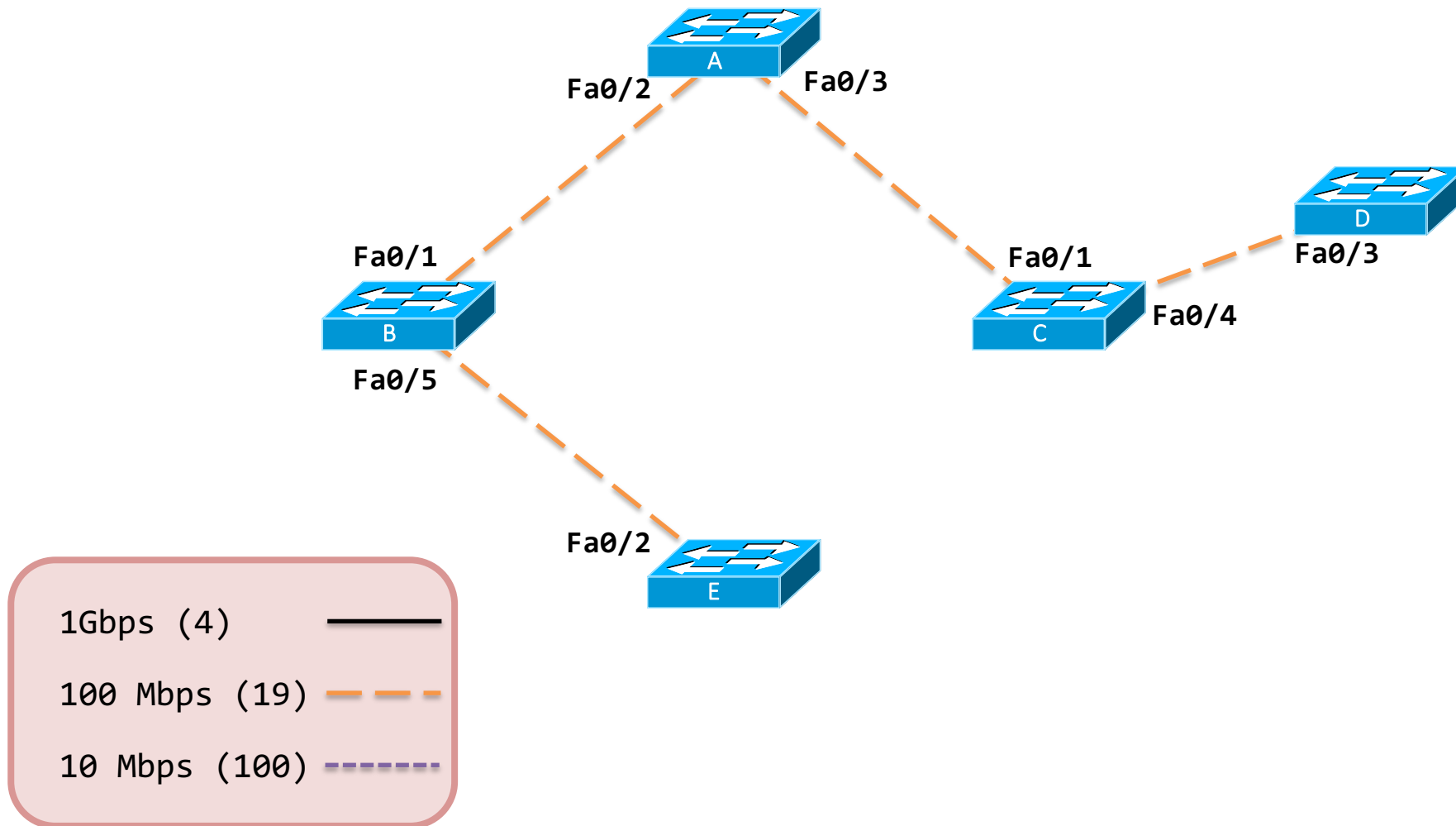


# Pasul 4: Blocked ports

- Toate porturile rămase sunt **blocked ports**

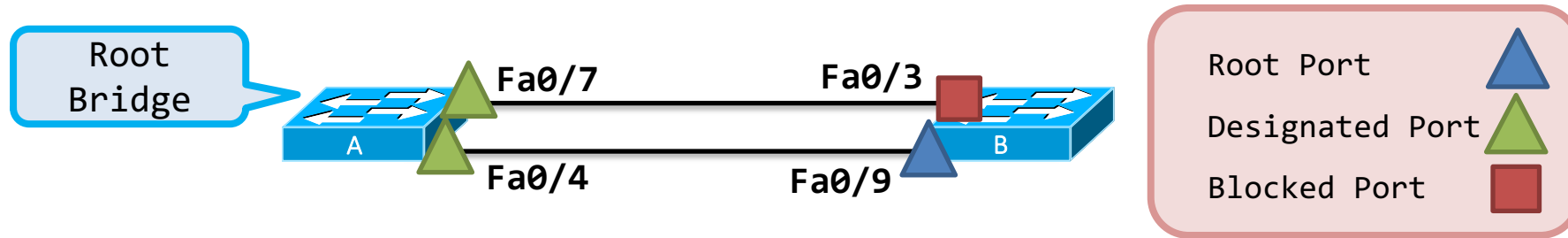


# Topologie logică finală



# Un ultim tiebreaker

- Poate apărea situația în care costurile și BID-urile sunt egale:



- Pentru această situație se definește conceptul de PID (Port ID), care este un număr format din:
  - prioritatea portului (configurată static de administrator)
  - indexul portului (de exemplu 7 pentru Fa0/7)
- Va fi folosită legătura care are PID-ul mai mic pe bridge-ul mai prioritar (root bridge, cost minim către root, BID mai mic)
- În cazul acesta, Fa0/9 devine root port deoarece Fa0/4 are un port id mai mic decât Fa0/7

# Stări Porturi în STP

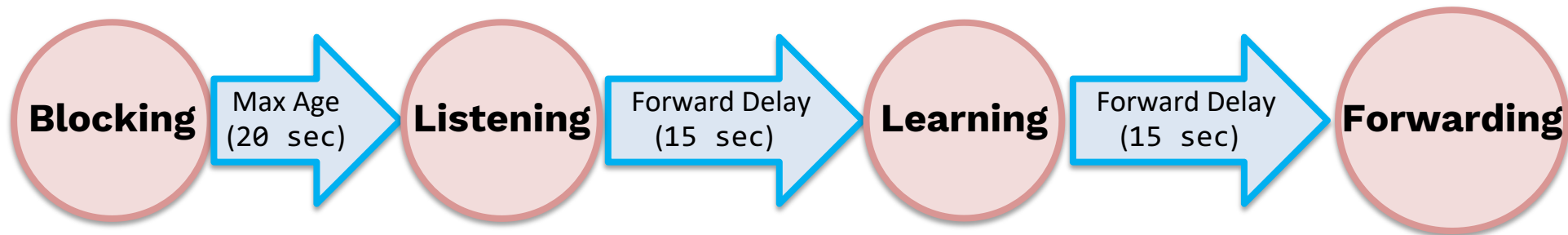
- În decursul STA, un port face tranziția între mai multe stări:

Stare port	Acțiune la nivel de Switch	Acțiune la nivel de Port
Disabled	Nu se acceptă nici un fel de trafic	Nu se transmit cadre Nu se transmit BPDU-uri
Blocking	Se primesc doar BPDU-uri	Nu se transmit cadre Se primesc BPDU-uri
Listening	Se construiește topologia STP	Nu se transmit cadre Se transmit BPDU-uri
Learning	Se construiește tabela de adrese MAC	Nu se transmit cadre Se învață adrese MAC Se transmit BPDU-uri
Forwarding	Se transmite traficul normal	Se transmit cadre Se învață adrese MAC Se transmit BPDU-uri



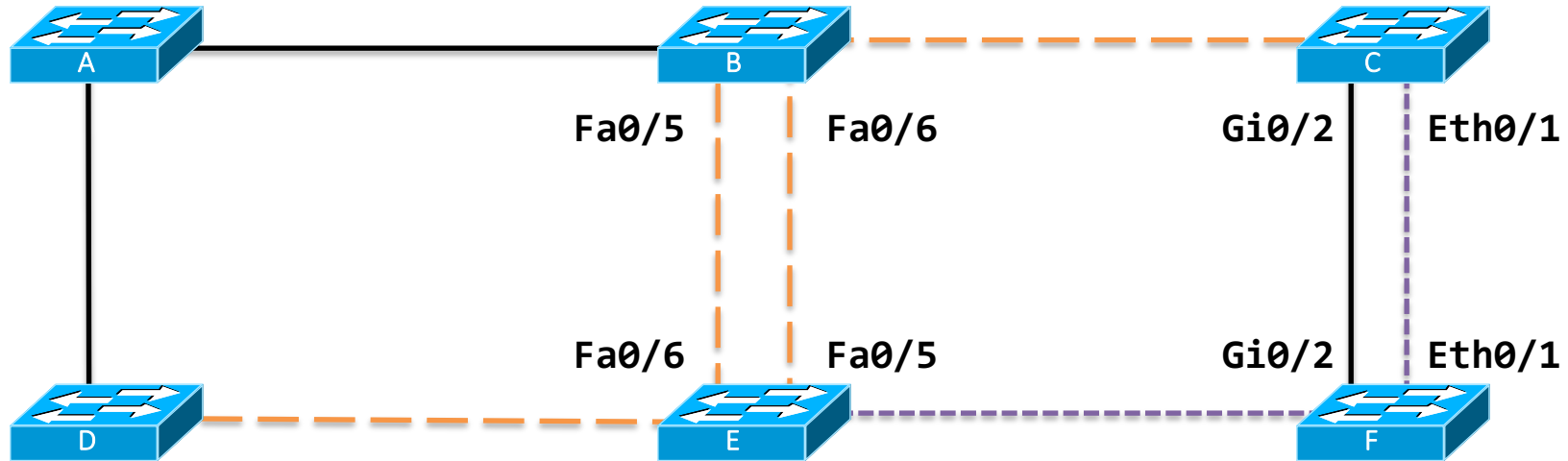
# Timpi de tranziție

- Timere de tranziție
  - stabilite de root bridge
  - **Hello time:** 2 sec
  - **Forwarding delay:** 15 sec
  - **Max Age:** 20 sec



- timp total de convergență: 50 sec

# Exemplu



Name	Prioritate	MAC
A	16384	00E0.A3C9.6AB8
B	32768	0001.97DA.86E8
C	8192	00D0.BC0C.844D
D	16384	0003.E496.C80E
E	8192	0060.7058.EB2B
F	8192	0060.702E.D0A5

1Gbps (4) ———

100 Mbps (19) - - - -

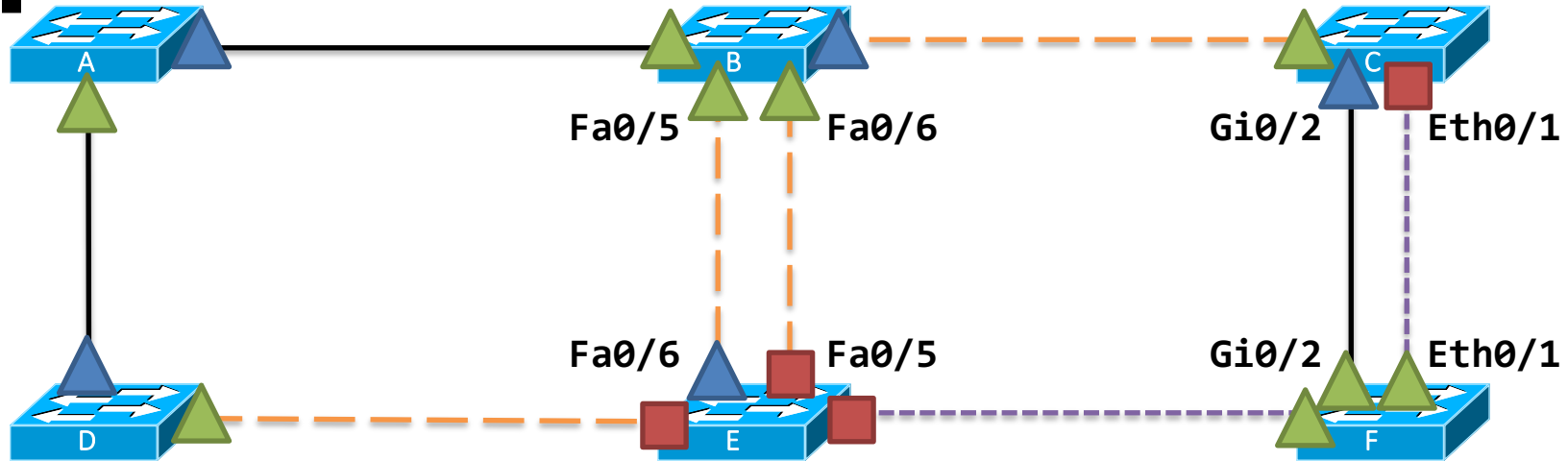
10 Mbps (100) - - - - -

Root Port ▲

Designated Port ▲

Blocked Port ■

# Exemplu



Nume	Prioritate	MAC
A	16384	00E0.A3C9.6AB8
B	32768	0001.97DA.86E8
C	8192	00D0.BC0C.844D
D	16384	0003.E496.C80E
E	8192	0060.7058.EB2B
F	8192	0060.702E.D0A5

1Gbps (4) ———

100 Mbps (19) - - - -

10 Mbps (100) - - - - -

Root Port ▲

Designated Port ▲

Blocked Port ■

# Variante STP

- Deoarece calculele STP durează foarte mult, s-a introdus RSTP care are o viteză de calcul a arborelui mult mai bună
- Deoarece VLAN-urile separă domeniile de broadcast, deși există bucle fizice pot să nu fie bucle logice
- Pentru a funcționa în rețele cu VLAN-uri, au fost introduse variante noi de STP:
  - PVST, RPVST (Cisco)
  - MSTP (IEEE)

# Etherchannel

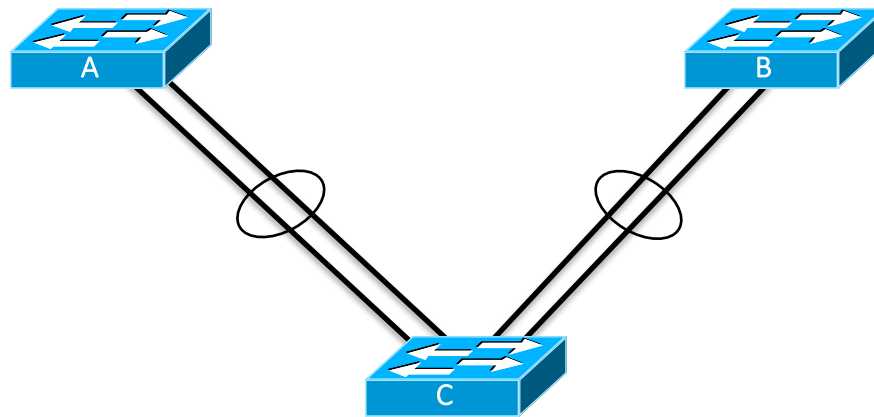


# Agregarea legăturilor

- Folosită la nivelul 2 – legătură de date
- Mai multe legături fizice devin o singură legătură la nivel logic
- Evită problemele de convergență din STP și congestiile de trafic
- Procesul se numește EtherChannel

# Etherchannel

- Două sau mai multe interfețe fizice agregate
- La nivel logic văzute ca o singură interfață
- Configurat prin diferite protocoale



# Terminologie

- **Port-Channel/Channel Group** – interfețele logice EtherChannel create prin agregarea legăturilor fizice
- **Member interface** – interfață fizică aparținând unui channel group
- **Protocol de negociere** – protocol folosit pentru stabilirea unei legături agregate



# Avantaje

- Configurare per interfață logică
- Load balancing
- Channel Group-ul este văzut ca un singur link de către STP
- Oferă redundanță
- Lățime de bandă, costuri mici (nu sunt necesare porturi suplimentare)

# Restricții de implementare

- Maximum 8 interfețe fizice într-un Channel Group
- Interfețele fizice din același Channel Group trebuie să fie identice/aceleași caracteristici
  - Viteză
  - Duplex
  - Mod (trunk/access)
  - Allowed VLAN/Access VLAN
  - Native VLAN
  - Trunking protocol
- Numărul de interfețe logice EtherChannel depind de platformă

# PAgP

- Port Aggregation Protocol – proprietar Ciscp
- Creează Channel Group-uri pe baza negocierilor active
- Pachetele PAgP sunt trimise o dată la 30 de secunde
- Modurile setate pe interfețe trebuie să fie compatibile în ambele părți ale legăturii fizice

# PAgP – moduri de negociere

- **On**

- Interfața formează un channel fără PAgP
- Nu primește/trimite pachete PAgP
- Recomandat pentru switch-uri cu vendori diferiți

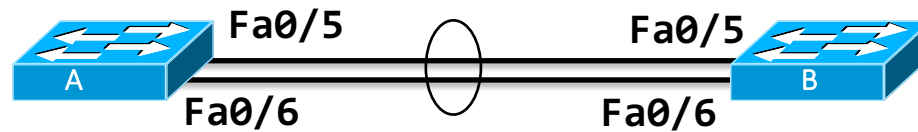
- **Desirable**

- Interfața este într-o stare de negociere activă
- Inițiază negocieri cu alte interfețe trimițând pachete PAgP

- **Auto**

- Interfața este într-o stare de negociere pasivă
- Răspunde la pachete PAgP primite, dar nu inițiază negocieri

# PAgP - negocieri



SwA	SwB	Channel Establishment
On	On	Yes
On	Desirable/Auto	No
Desirable	Desirable	Yes
Desirable	Auto	Yes
Auto	Desirable	Yes
Auto	Auto	No

# LACP

- Link Aggregation Control Protocol
- IEEE 802.3ad (mai nou IEEE 802.1AX)
- Funcții similare cu PAgP, doar modurile diferă

# LACP – moduri de negociere

- **On**

- Interfața formează un channel fără LACP
- Nu primește/trimite pachete LACP
- Recomandat pentru switch-uri cu vendori diferiți

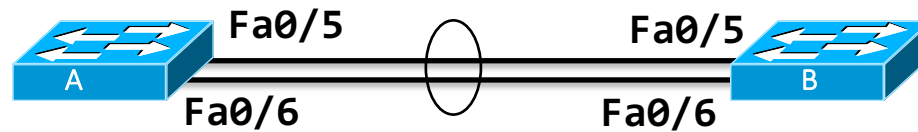
- **Active**

- Interfața este într-o stare de negociere activă
- Inițiază negocieri cu alte interfețe trimițând pachete LACP

- **Passive**

- Interfața este într-o stare de negociere pasivă
- Răspunde la pachete LACP primite, dar nu inițiază negocieri

# LACP - negocieri



SwA	SwB	Channel Establishment
On	On	Yes
On	Active/Passive	No
Active	Active	Yes
Active	Passive	Yes
Passive	Active	Yes
Passive	Passive	No



# Sumar

