Cours RSX103 : Réseaux Compléments et Applications

Chapitre II

Topologie des équipements d ’interconnexion

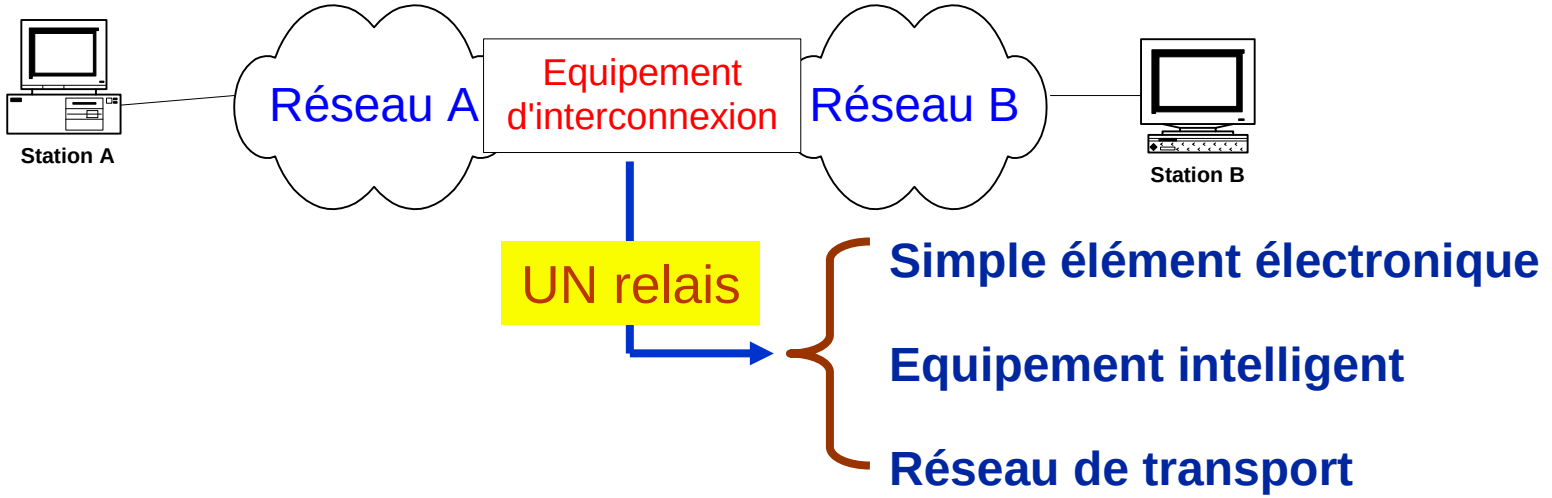
Algorithme STP

GENERALITES SUR L ’INTERCONNEXION

INTERCONNEXION ?

- Objectif : Nécessité de mettre en relation des machines :
  - Appartenant à des réseaux physiquement distincts,
  - Indépendamment de la distance qui les sépare
  - Et les protocoles qu'ils utilisent

- **Exemple :**

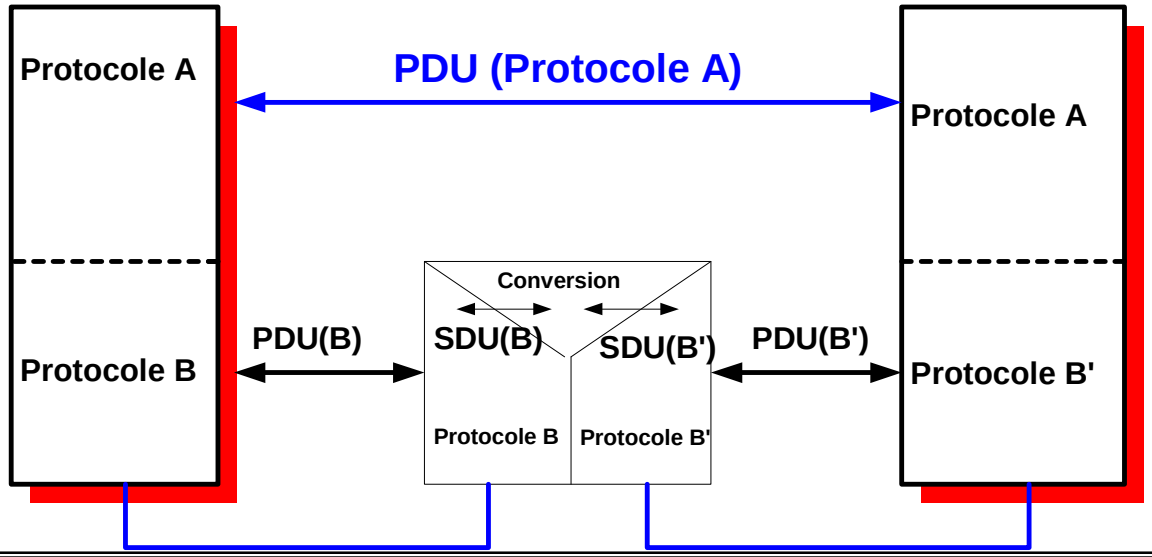


Rôle des relais (Unité d'Interfonctionnement : UIF) –1-

❑ UN relais (Unité d 'Interfonctionnement : UIF) doit assurer :

1. La conversion de service :

- **Le relais reçoit les messages selon le format A et les envoie en utilisant les services B.**
- **Cette technique est utilisable lorsque les protocoles à mettre en relation sont différents mais compatibles**
  - **Exemple : Passage d 'un réseau 802.3 -10 Base 5 à un réseau 802.3 -10 Base T**



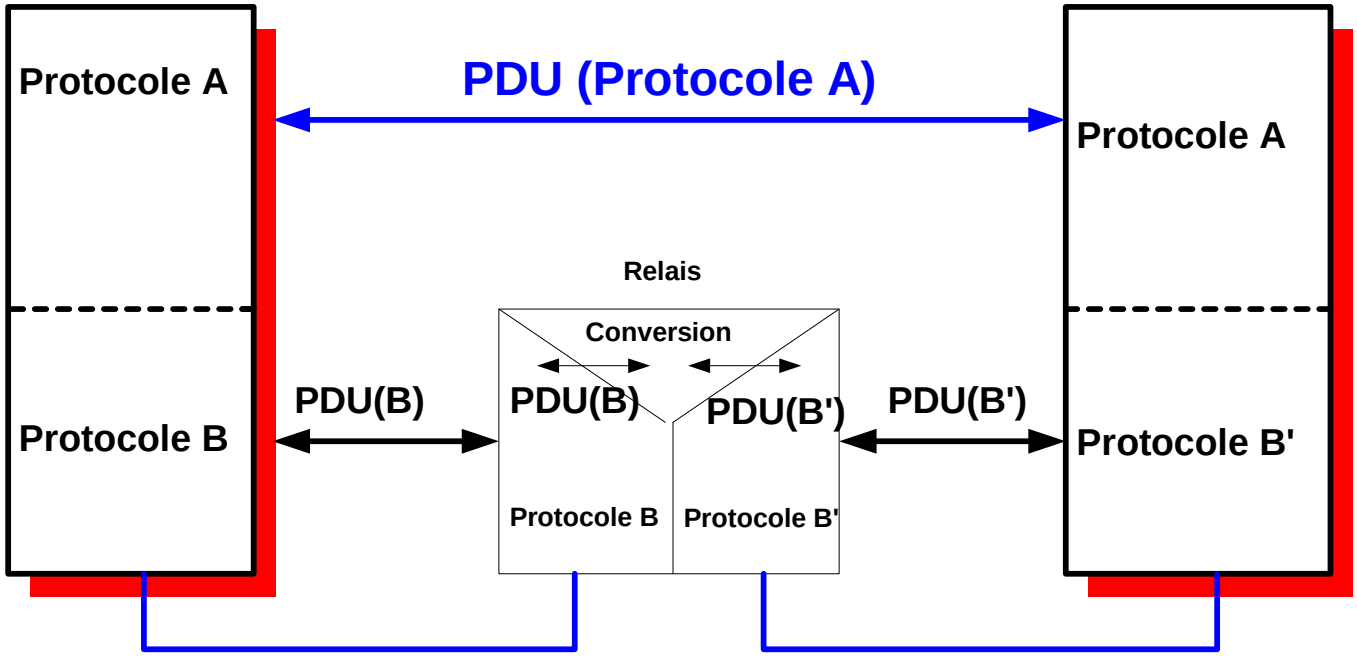
Rappel

version	vitesse	médium
10 Base 5	10 Mbps	50 ohm coaxial (épais)
10 Base T	10 Mbps	paire torsadée

- 10Base2 – Appelée Ethernet à câble fin, cette norme autorise des segments de réseau pouvant atteindre **185 mètres sur câble coaxial.**
- 10Base5 – Appelée ThickEthernet, cette norme autorise des segments de réseau pouvant atteindre **500 mètres sur câble coaxial.**
- 10BaseT – Permet de transporter des trames Ethernet sur des **câbles à paires torsadées bas de gamme.**

2.La conversion de protocole,

- **Lorsque les protocoles sont différents et incompatibles**
- **Exemple : Interconnexion Ethernet ⇔ Token Ring**

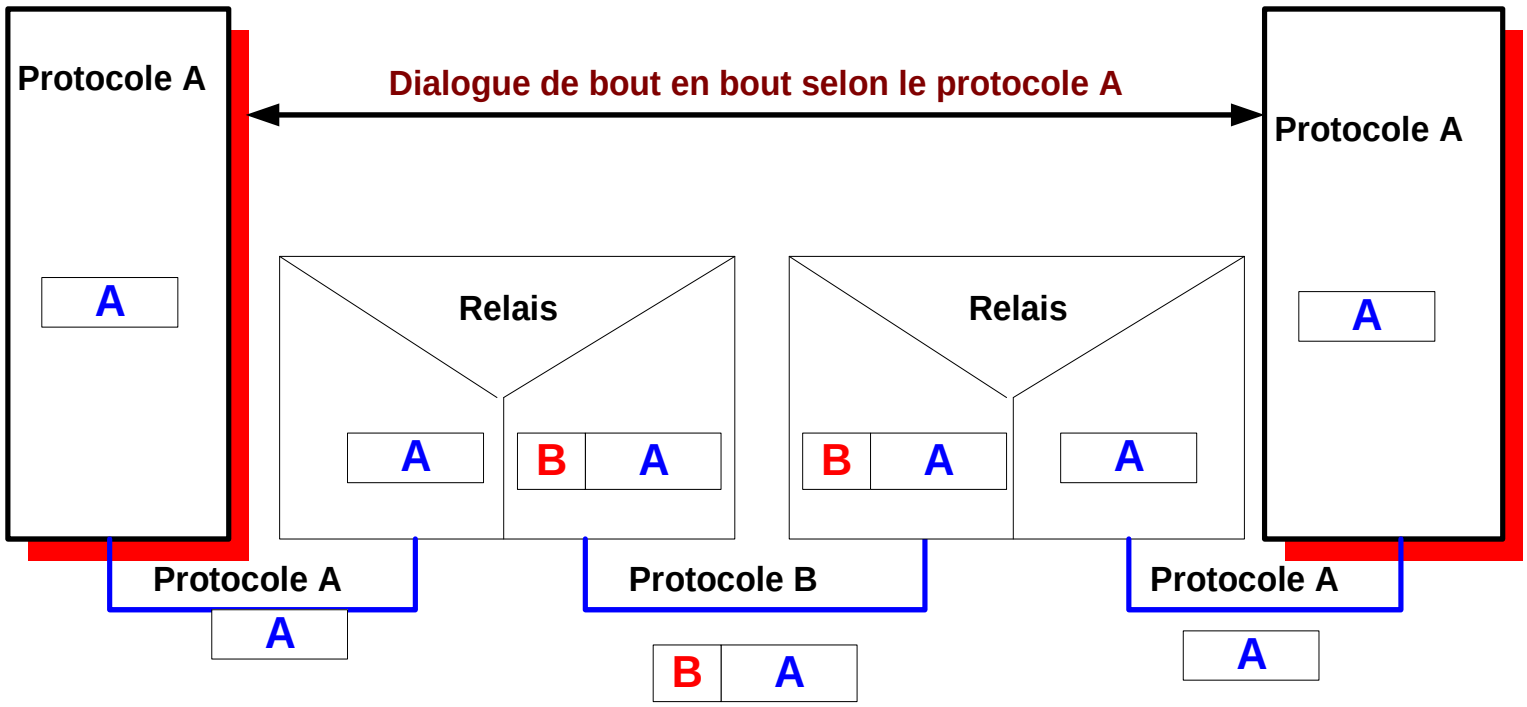


3. L'encapsulation :

⇨ **Lorsque le système d 'interconnexion utilise des systèmes de transport incompatibles avec les protocoles locaux à mettre en relation**

⇨ ***« Dans ce cas, les données du protocole A sont transportées dans les unités du protocole B »***

⇨ **Exemple : interconnexion du réseau TCP/IP via un réseau de transport X25.**



Interconnexion –1-

- **L ’interconnexion est une fonction pour réaliser l ’inter-fonctionnement des réseaux hétérogènes**

- **L ’hétérogénéité des réseaux peut être à plusieurs niveaux :**

  - **Matériel**
  - **Capacité des réseaux**
  - **Taille des paquets**
  - **Protocoles**
  - **Services**

- **Méthode :**

  Identifier le niveau d ’hétérogénéité afin de déterminer les fonctions requises pour établir l ’interconnexion (modèle ISO)

- **Selon le niveau d ’ hétérogénéité considéré :**
  - **Mise en œuvre d ’un dispositif d ’inter-fonctionnement**

Interconnexion –2-

- **Equipements employés :**
  - **Répéteurs / Concentrateur (Hub)**
  - **Ponts / Commutateurs (Switch)**
  - **Routeurs**
  - **Passerelles**
  - **Pare-Feux (…?)**
- **Techniques employées :**
  - **Amplification**
  - **Encapsulation**
  - **Fragmentation / Réassemblage**
  - **Conversion de protocoles / services**

ETHERNET / IEEE 802.3

« Rappels »

IEEE 802.3 - rappels

- **Réseau local le plus répandu**
  - **Initialement proposé par Digital, Intel et Xerox**
  - **puis standardisé par IEEE et ISO**

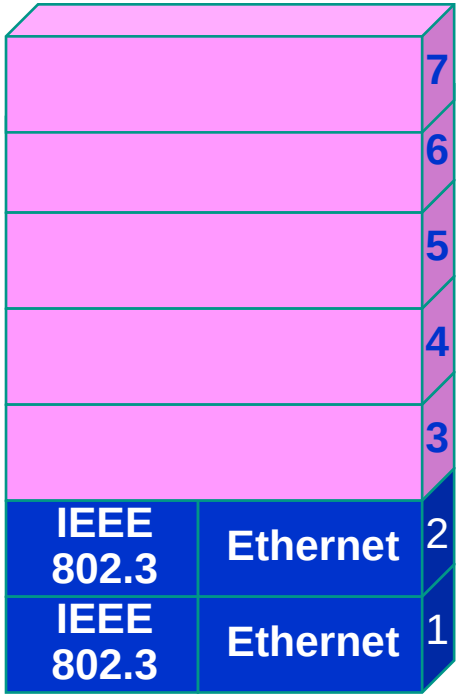
- **Utilise la méthode d'accès : CSMA / CD**

- **Caractéristiques techniques :**

1. **Topologie en** :
   - **bus sur coaxial : 10 base5 ou 10 base2**
   - **en étoile (hub) : sur paire téléphonique : 10 base T**

2. **Débit : 10 Mbps**
3. **Délai aller-retour maximum du bus**
   - **51. 2 µsec imposé dans la norme Ethernet/ 802. 3**
     - **taille minimale des trames : 512 bits**
       - **(pour un débit de 10 Mbps)**
4. **Câblage**

   - **Initialement coaxial, mais aujourd'hui paires torsadées**
   - **10Base5 : coaxial 500m maximum et 100 stations**
   - **10Base2 : coaxial 200m maximum et 30 stations**



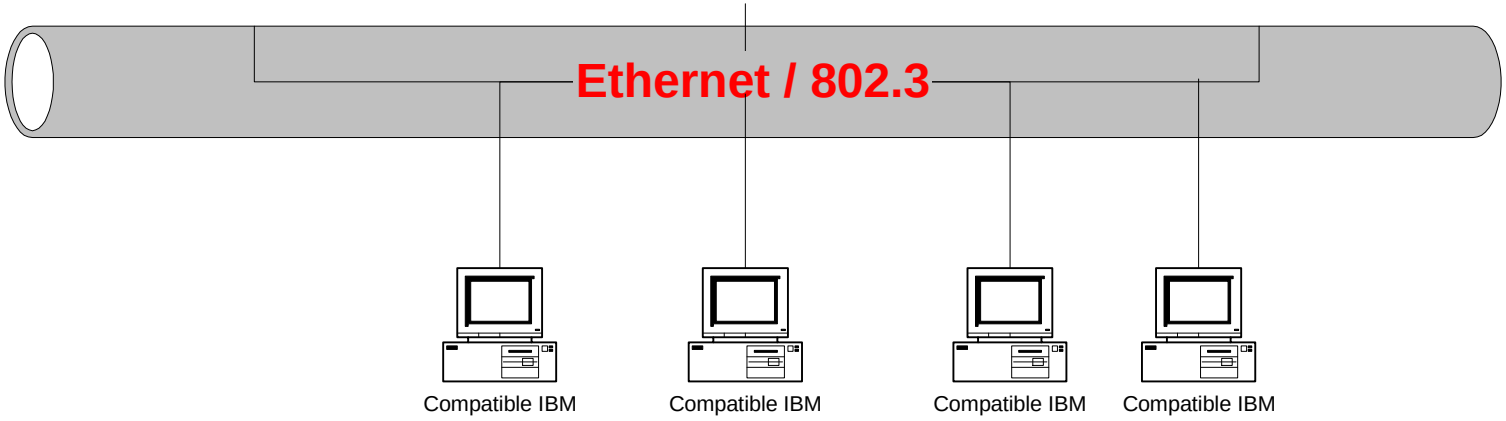
♦ Configuration originale

- **Réseau en bus sur câble coaxial**
- **Problème à résoudre:**

  ✓ **N ordinateurs cherchent à accéder à la ressource partagée (canal de transmission).**

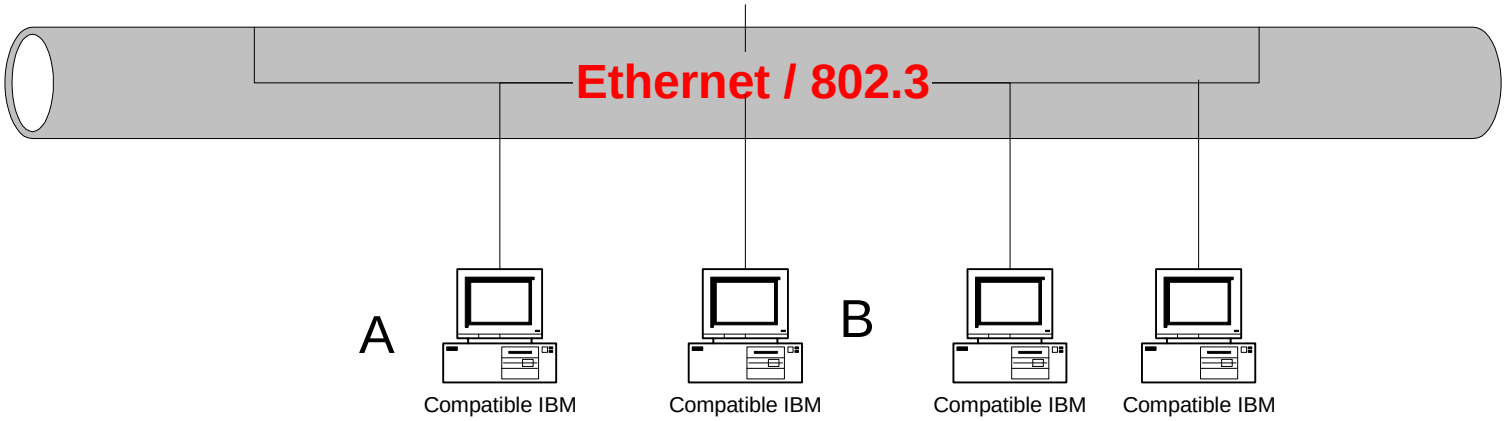
  ✓ **Un seul canal disponible pour la communication**

- **Solution : Algorithme CSMA/CD**



- **D ’autres Problèmes à résoudre ?**

1. **Problème N°. 1:**
   - **Comment transmettre une trame de A vers B ?**
   - **Comment B détecte qu’une trame lui est destinée ?**
2. **Problème N°. 2 :**
   - **Comment transmettre une trame à toutes les stations connectées au réseau ?**
3. **Problème N°. 3:**
   - **Comment transmettre une trame à un groupe de stations connectées au réseau**

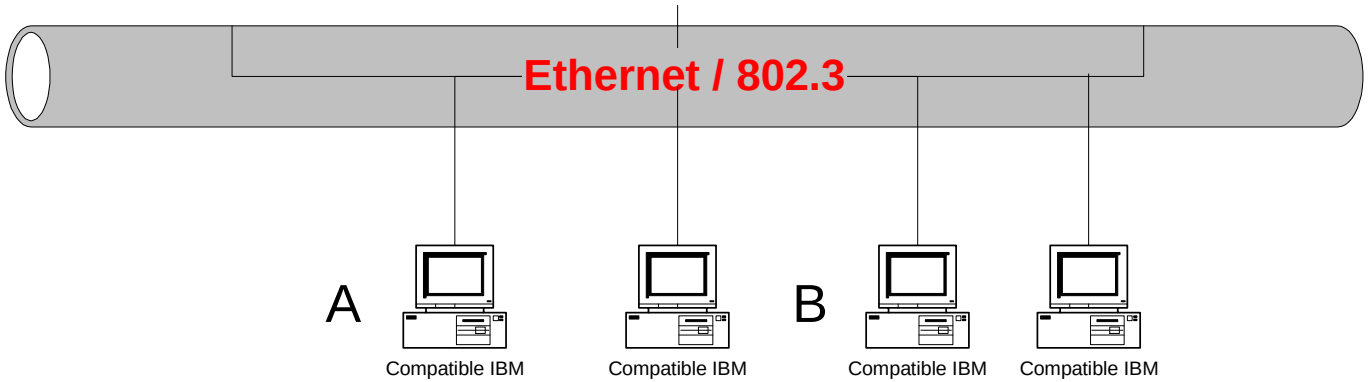


Adresses Ethernet

- **En effet, sur un bus, toutes les stations voient toutes les trames**

  ☞Alors, Comment une station peut-elle savoir qu'une trame qui passe lui est destinée ?

- **Solution : Chaque trame contient :**
  - l'adresse de la station source qui a envoyé la trame
  - l'adresse de la station destination qui doit recevoir la trame
- **Comment choisir l'adresse d'une station ?**
  1. Par configuration manuelle : adresses locales
  2. Par construction : adresse encodée en ROM par le fabriquant de la carte Ethernet

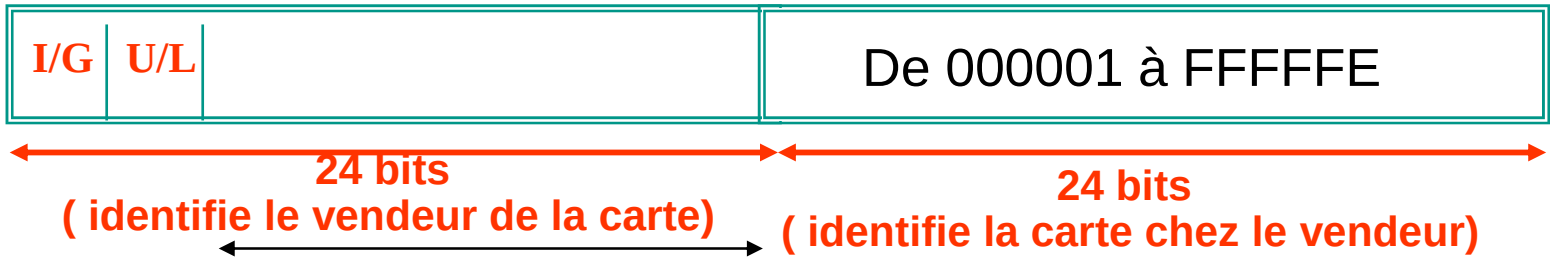


Adresses Ethernet

- **Chaque carte Ethernet contient une adresse Ethernet unique (adresse MAC)**

Permet de garantir que sur n'importe quel réseau, deux stations n'auront pas la même adresse

- **Format des adresses Ethernet (MAC)**
  - **Adresses encodées sur 48 bits**
    - **Adresse source**



    - **Adresse destination**
      - si bit de poids fort= 0, adresse d'une station
      - si bit de poids fort= 1, adresse multicast
    - **adresse destination = 111111.. 111: broadcast (FF:FF:FF:FF:FF:FF)**

Adresses Ethernet

La norme définit les adresses sur 6 octets (48 bits).

I/G U/L adresse de 46 bits (48 bits)

I/G	U/L	adresse de 46 bits

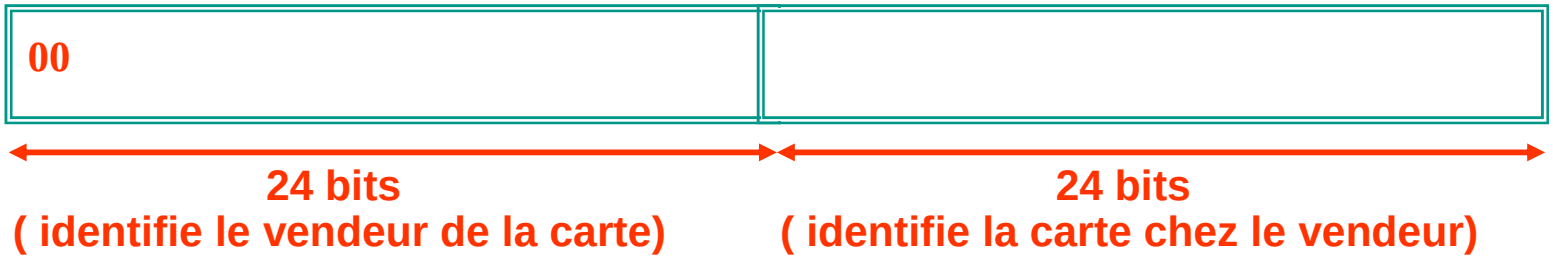
Adresse individuelle	I/G =0
Adresse de groupe (multicast)	I/G =1
Adresse globale attribuée par IEEE	U/L =0
Adresse locale	U/L =1

Si tous les 48 bits sont égal ` a 1, il s'agit d'une adresse à diffusion générale (broadcast).

Avec les 46 bits disponible, on peut former $2^{46} \sim 7 \times 10^{13}$ **adresses distinctes.**

Adresses Ethernet

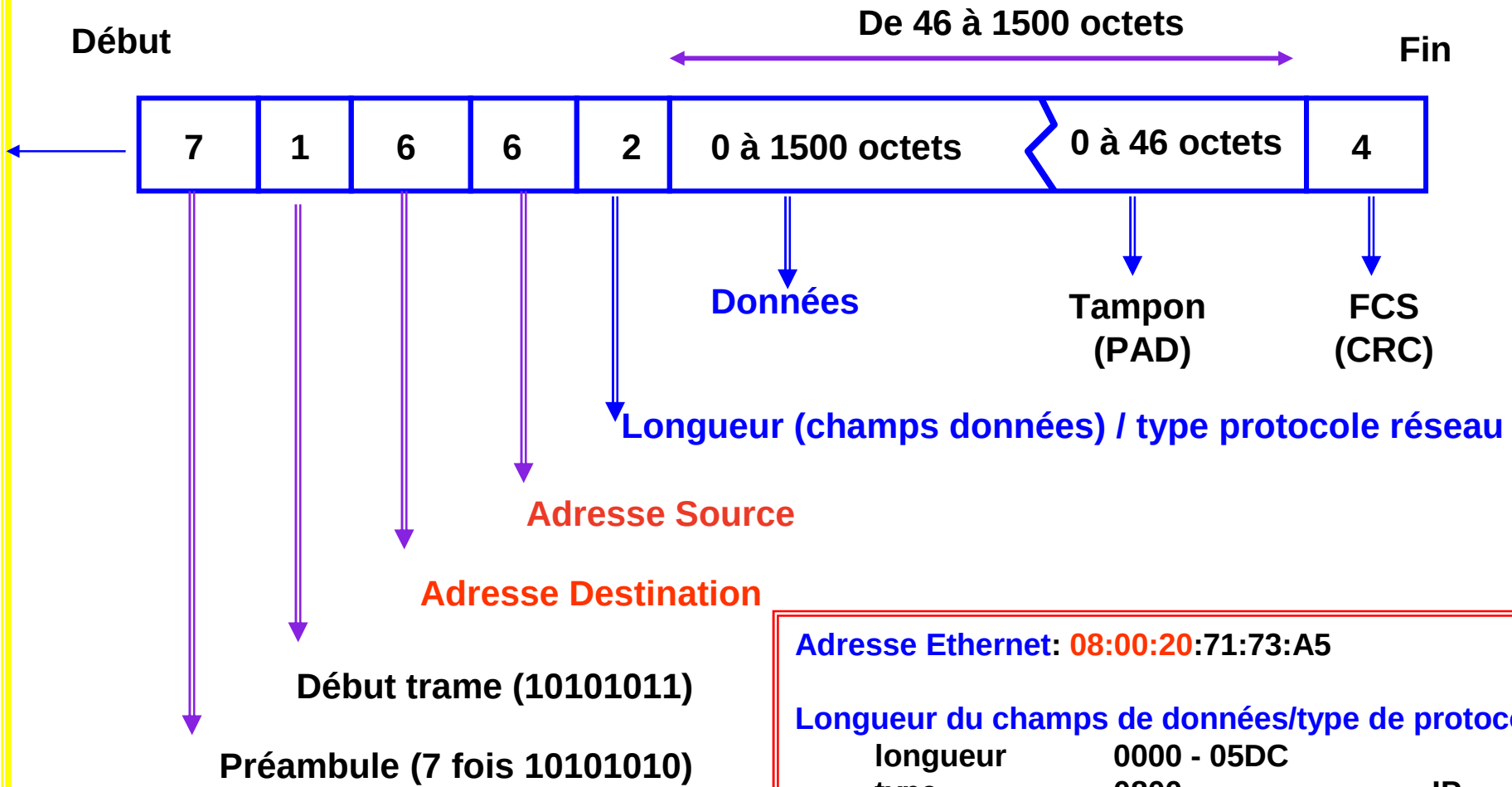
- L'adresse Ethernet est composée de deux identificateurs de 3 octets chacun:



- Exemples d'identificateurs de vendeurs:
  - 00-00-0C : Cisco
  - 08-00-20 : SUN
  - 08-00-5A : IBM

- Exemples de préfixes pour la communication de groupe:
  - 01-00-5E : Multicast Internet
  - 0D-1E-15 : Hewlett & Packard

Trame Ethernet – Format de base



Adresse Ethernet: **08:00:20**:71:73:A5

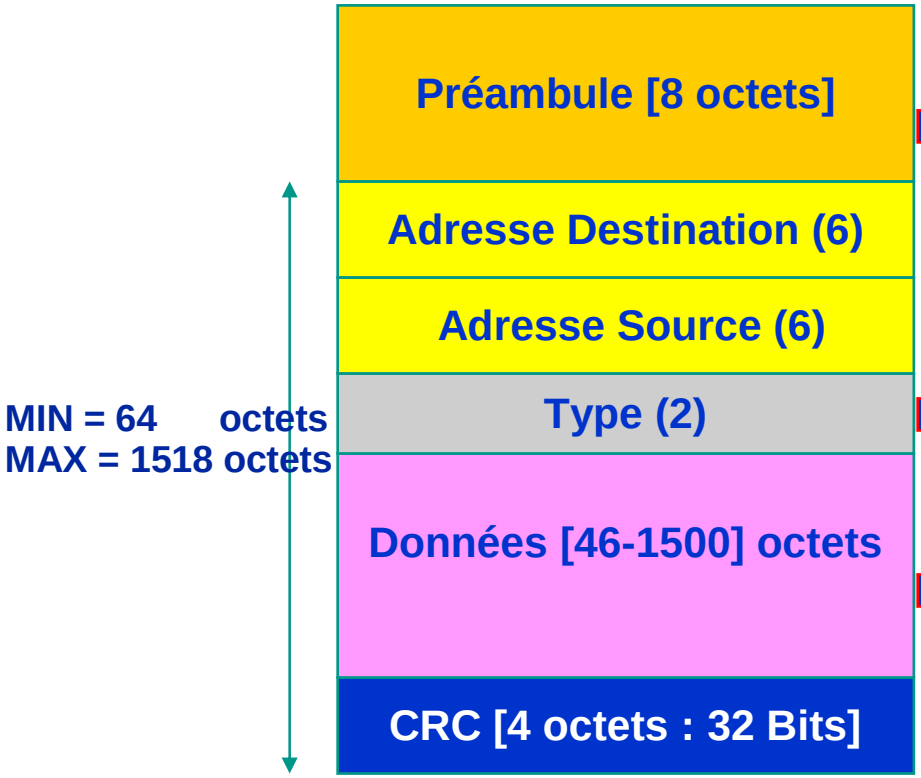
Longueur du champs de données/type de protocole:

longueur	0000 - 05DC	
type	0800	IP
	0805	X.25
	0806	Ethernet ARP
	809B	Appletalk
	814C	SNMP
	0600	XNS Internet

Trames Ethernet : Format DIX

Format original (DIX) :

proposé par Digital, Intel et Xerox



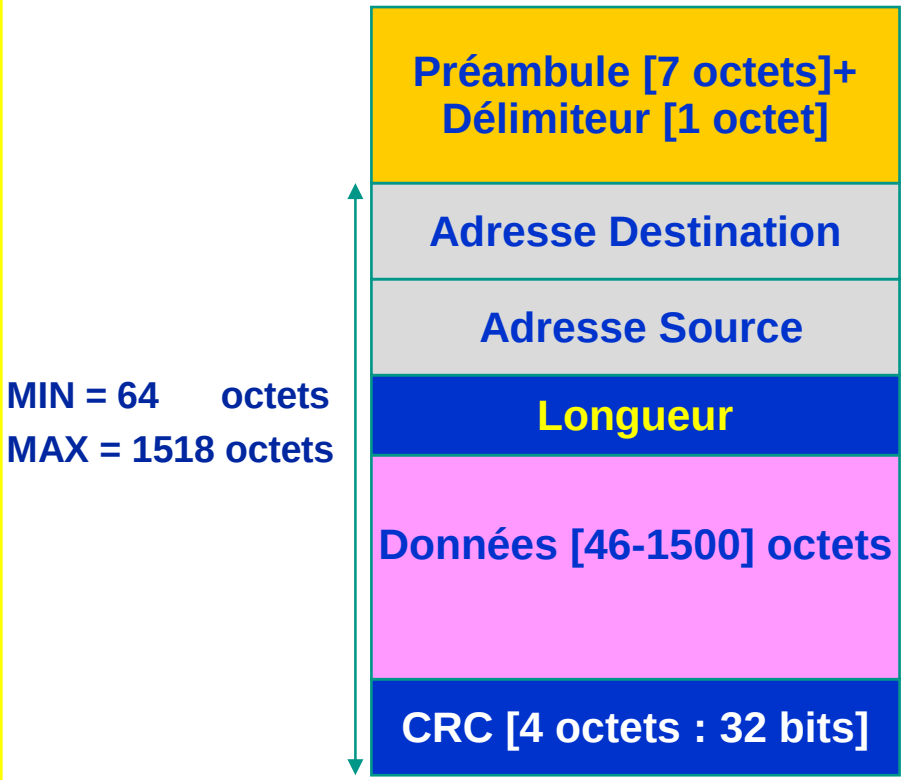
Utilisé pour marquer le début de la trame et permettre au receveur de synchroniser son horloge

Information sur le "type » d'utilisateur qui recevra le contenu de la trame à la destination « type protocole »

L'utilisateur source doit se débrouiller pour envoyer des paquets d'au moins 46 octets

Format 802. 3

standardisé par IEEE



Utilisé pour marquer le début de la trame et permettre au receveur de synchroniser son horloge

Indique la taille réelle du paquet qui a été transmis par la source
802.3 rajoute à la suite du paquet du padding (octets à la valeur 0) pour que la partie données de la trame contienne au moins 46 octets

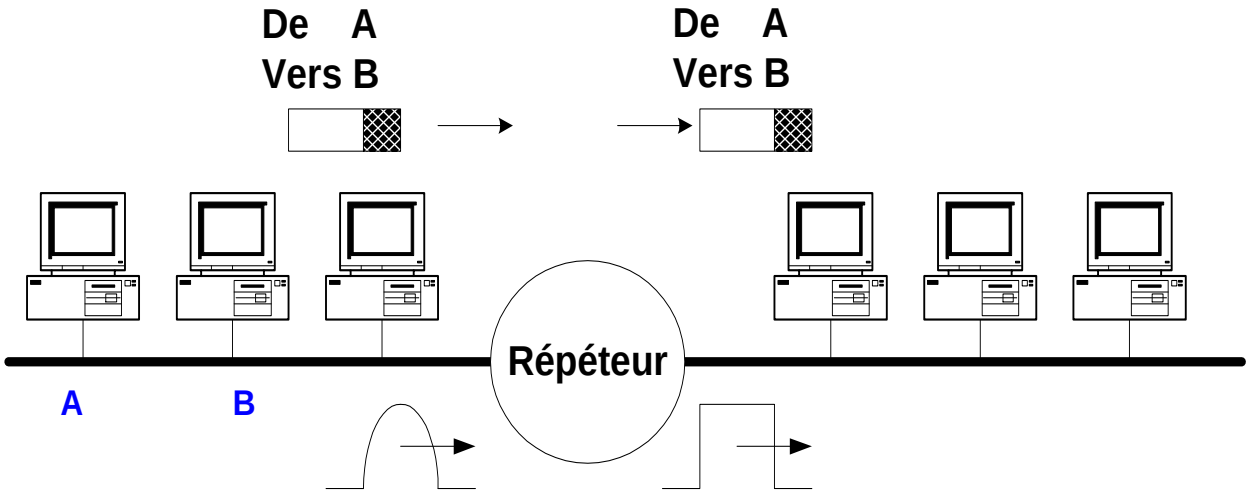
Services Ethernet

- **Le service fourni par un réseau local Ethernet est un service sans connexion et sans acquit (unacknowledged connectionless)**
- **Modes de transmission**
  - **unicast**
  - **multicast**
  - **broadcast**
- **Même si le service Ethernet est en théorie non-fiable, un bon Ethernet doit**
  - **délivrer les trames à la destination avec une très haute probabilité**
  - **ne pas modifier l'ordre des trames transmises**

DIFFERNTS TYPES DE RELAIS

Les Répéteurs

- **Réalisent une connexion physique entre deux segments d 'un même réseau logique**
- **Les réseaux interconnectés doivent être homogènes, car les répéteurs agissent au niveau physique (il ne fait que retransmettre d'un côté les bits reçus sur l 'autre)**
  - **« Il agit par diffusion »**
- **Fonction de régénération du signal**
  - **Réalise l'adaptation des supports**
  - **Pas d 'interconnexion fonctionnelle**

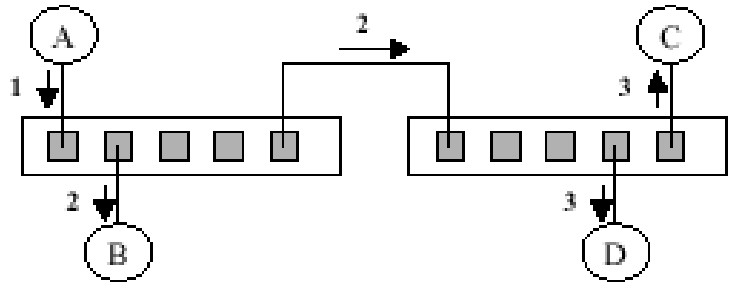


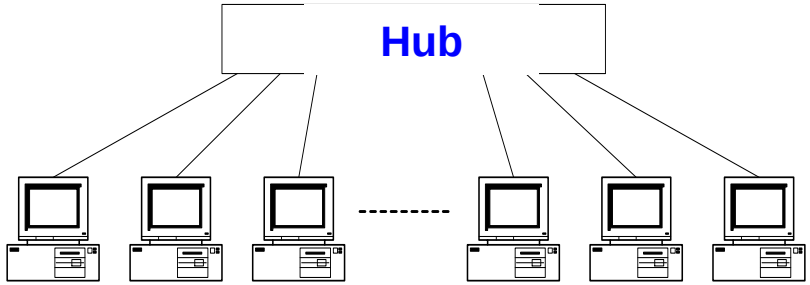
Les Répéteurs

- **Le REPETEUR est donc :**
  - **un dispositif actif non configurable**
  - **permet d 'augmenter la distance entre deux stations Ethernet**
  - **Reçoit, amplifie et retransmet les signaux (Travaille au niveau du bit)**
  - **Ne connaît pas la trame**
  - **Ne procède à aucun filtrage**
  - **Limité à 4 en série (Ethernet)**
- **Limitations :**
  - **Ne peuvent être utilisés que sur le même type de segments**
    - **(Ethernet ⇔Ethernet ou Token Ring ⇔ Token Ring)**
  - **Pas de conversion de signaux (Optique -> Electrique)**

Les HUB « Multi-répéteurs » / Répéteurs MultiPorts

- **Les Hubs sont employés dans les réseaux locaux ETHERNET**
  - **Ont une fonction répéteur**
  - **mais aussi ;**
    1. **Permettent de mixer différents médias :**
       (paires torsadées, AUI : Attachement Unit Interface/Transceiver, Thin Ethernet, Fibre Optique),
    2. **D'employer une topologie physique en étoile (Ethernet 10 Base T)**

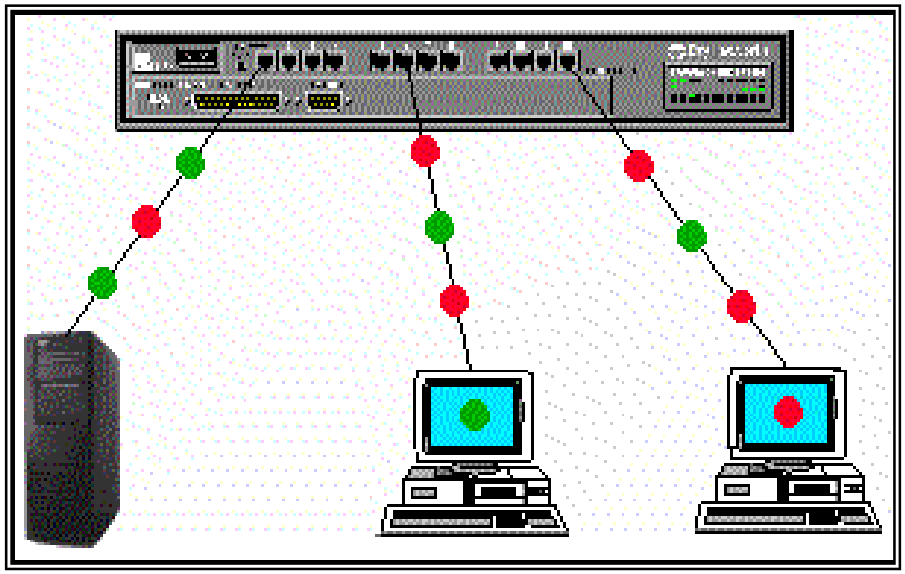




– Toute trame reçue sur un port est répétée sur tous les autres ports

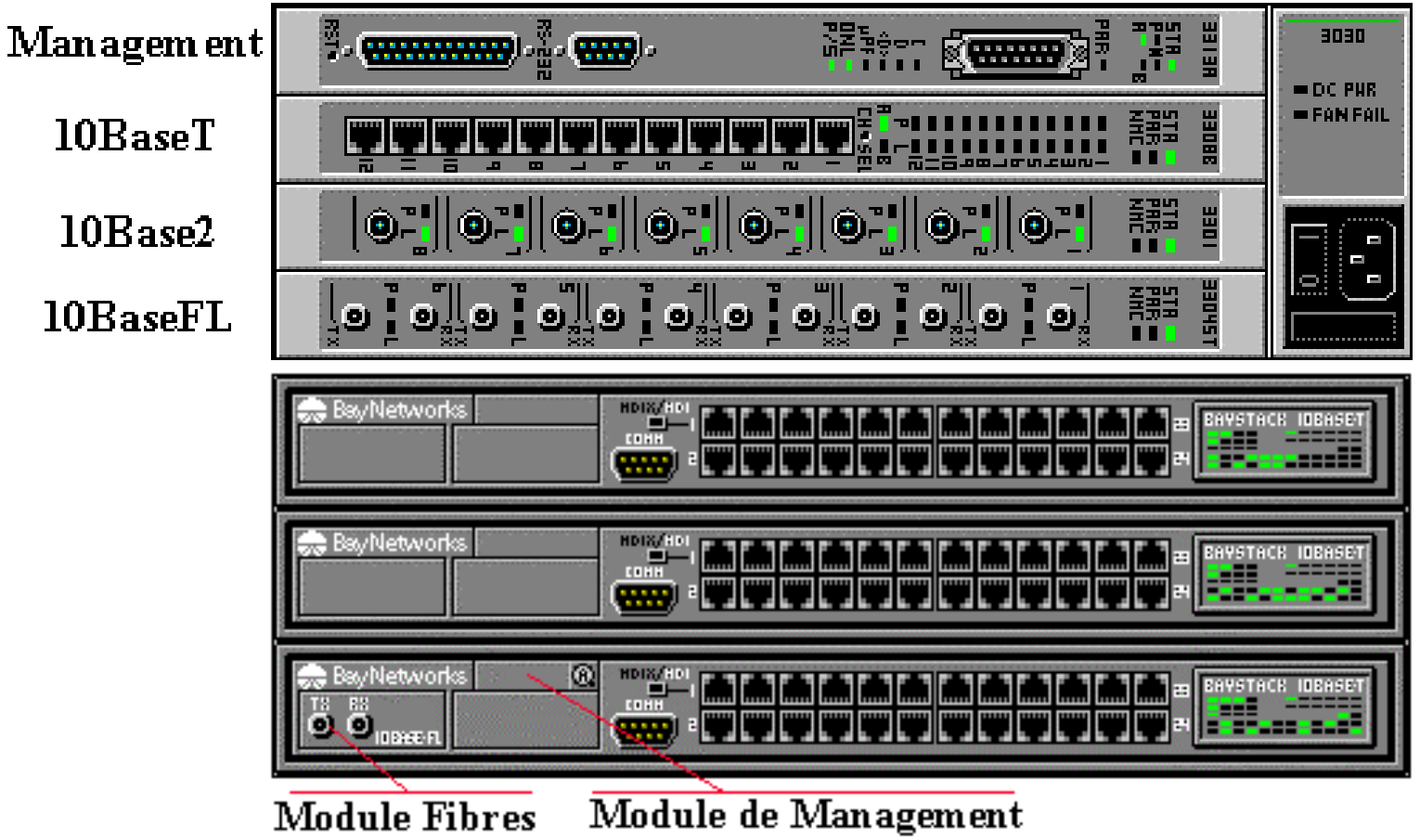
Les HUB « Multi-répéteurs » / Répéteurs MultiPorts

- **Un Hub est souvent composé d ’un châssis pouvant contenir N carte**
- **Peuvent comprendre aussi un agent SNMP**
- **Peuvent être empilables (un seul domaine de collision)**



- **Limitation principale**
  - Constituent des goulots d’étranglement
  - Tout le trafic transite par leurs bus de fond de panier
- **Exemple :**
  - *Soit un réseau 10baseT contenant 8 hubs de 12 ports reliant 80 stations*
  - *Si les stations émettent et reçoivent à 10 Mbps, elles disposent en fait d’une bande passante moyenne de 10 / 80 = 128 Kbps*
  - N.B : *En effet, chaque station est en concurrence avec toutes les autres*

Les HUBS : Exemple de châssis empilables



On peut ainsi empiler en les connectant au moyen d'un câble parallèle des modules de 12 ou 24 ports RJ45. (Hub stackable).

Les Ponts

- **Sont des éléments d'interconnexion de niveau 2.**
  - **Ils permettent d'interconnecter deux ou plusieurs réseaux (ponts multiports) dont les couches physiques sont dissemblables.**
  - **Interconnecter des segments LAN similaires (travail sur les trames au niveau liaison de données)**
  - **Sont transparents aux protocoles de niveau supérieur**
- **Intérêts :**
  1. **Etendre la portée des LANs au-delà des limites imposées par le répéteurs**
  2. **Isoler du trafic dans certains segments :**
     - **Segmenter le réseaux en sous réseaux indépendants**
     - **Créer des domaines de collision en réduisant le nombre de stations sur un même brin)**
     - **Capable de convertir des trames de formats différent (ex. Ethernet - Token Ring**
     - **Ils sont transparents aux protocoles de niveau supérieur**

symbole

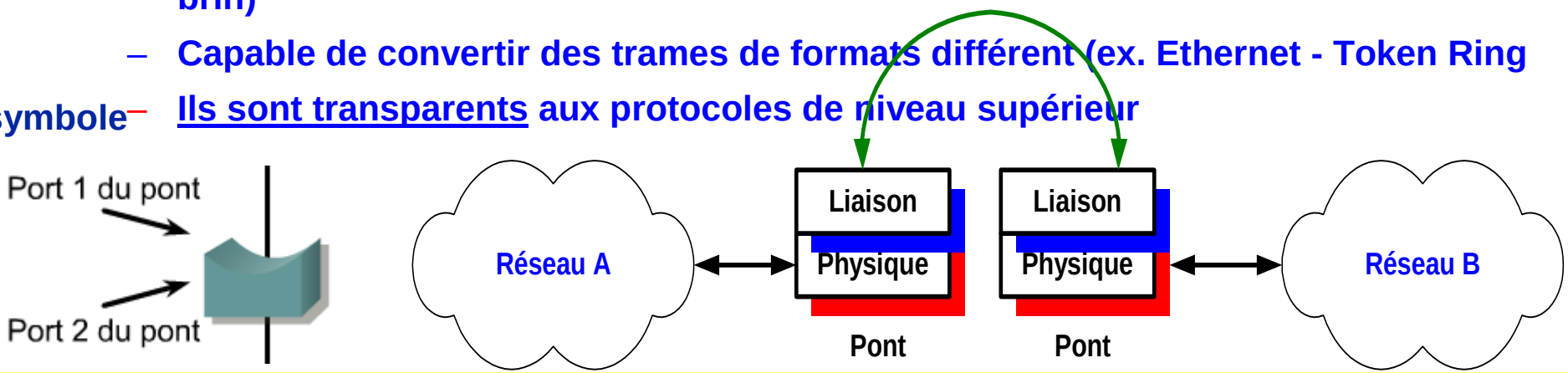
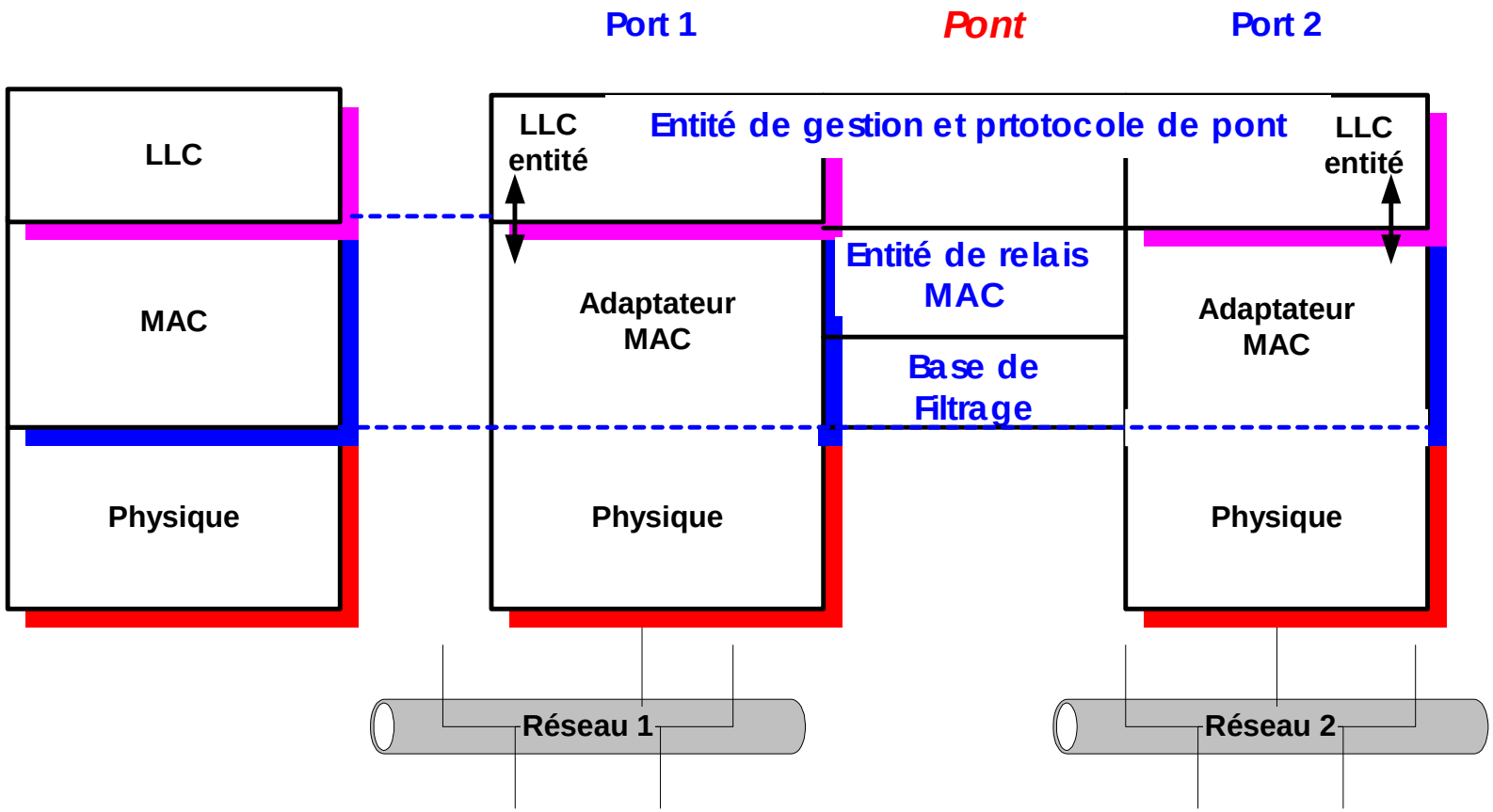
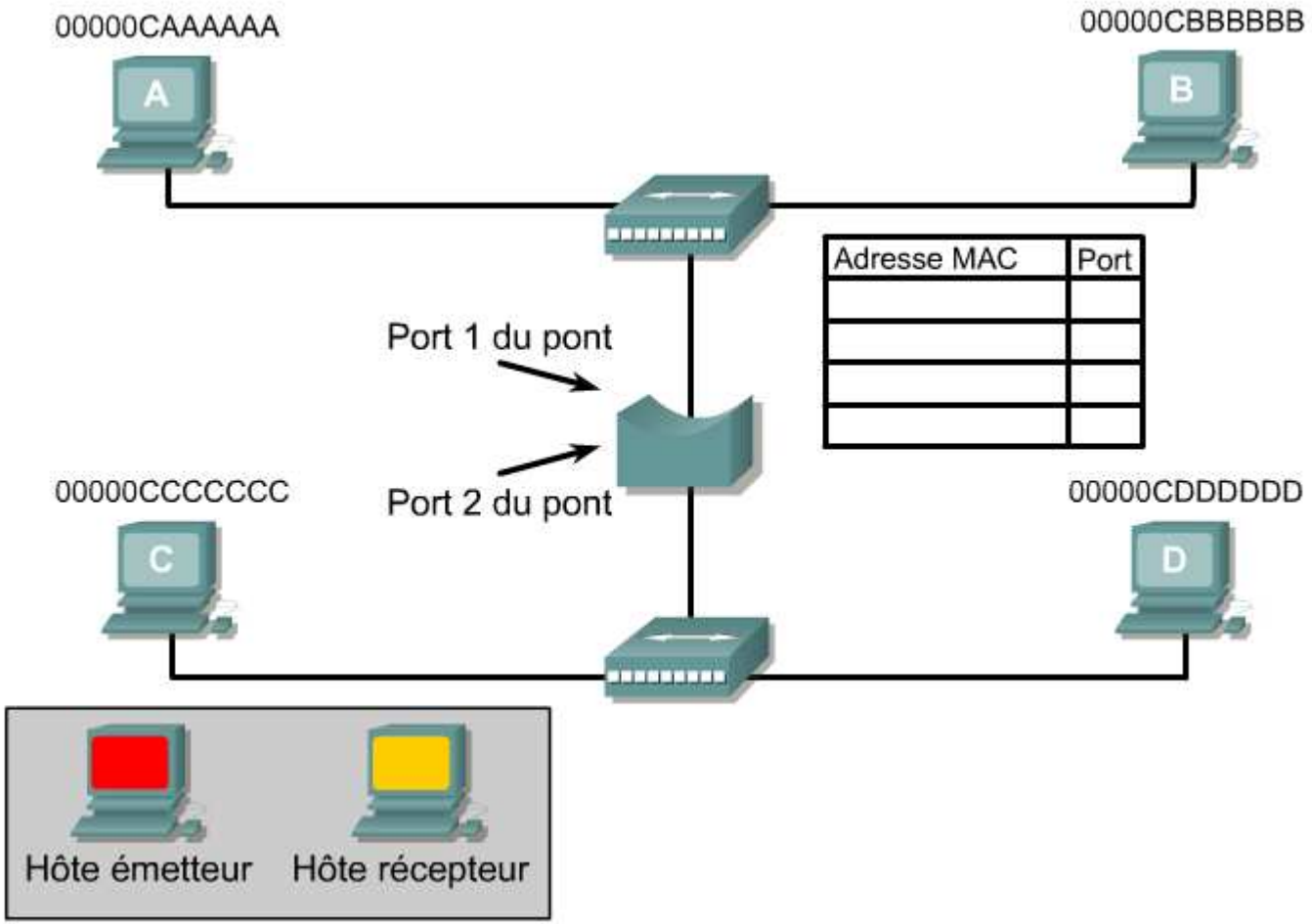


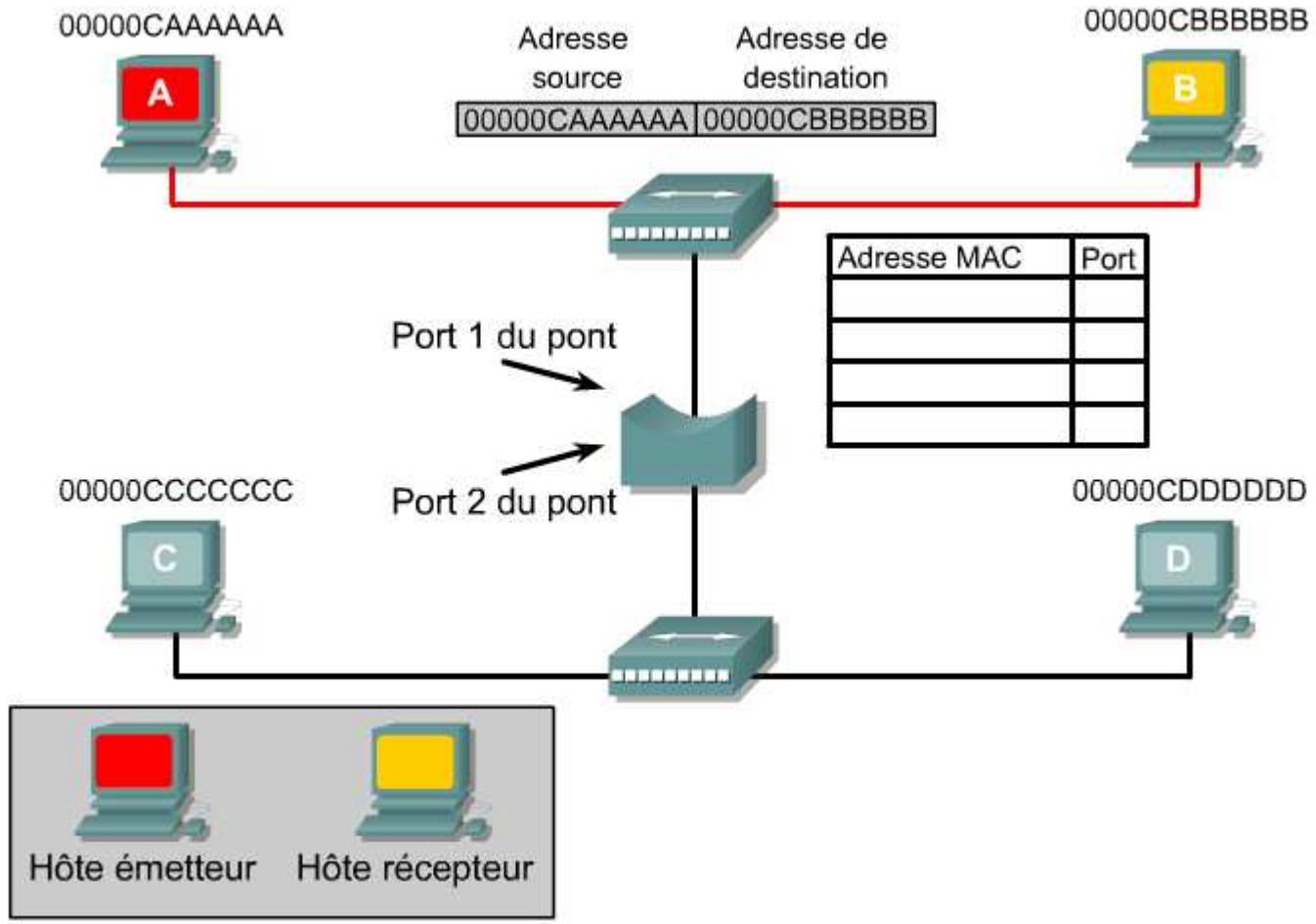
Schéma fonctionnel d'un pont



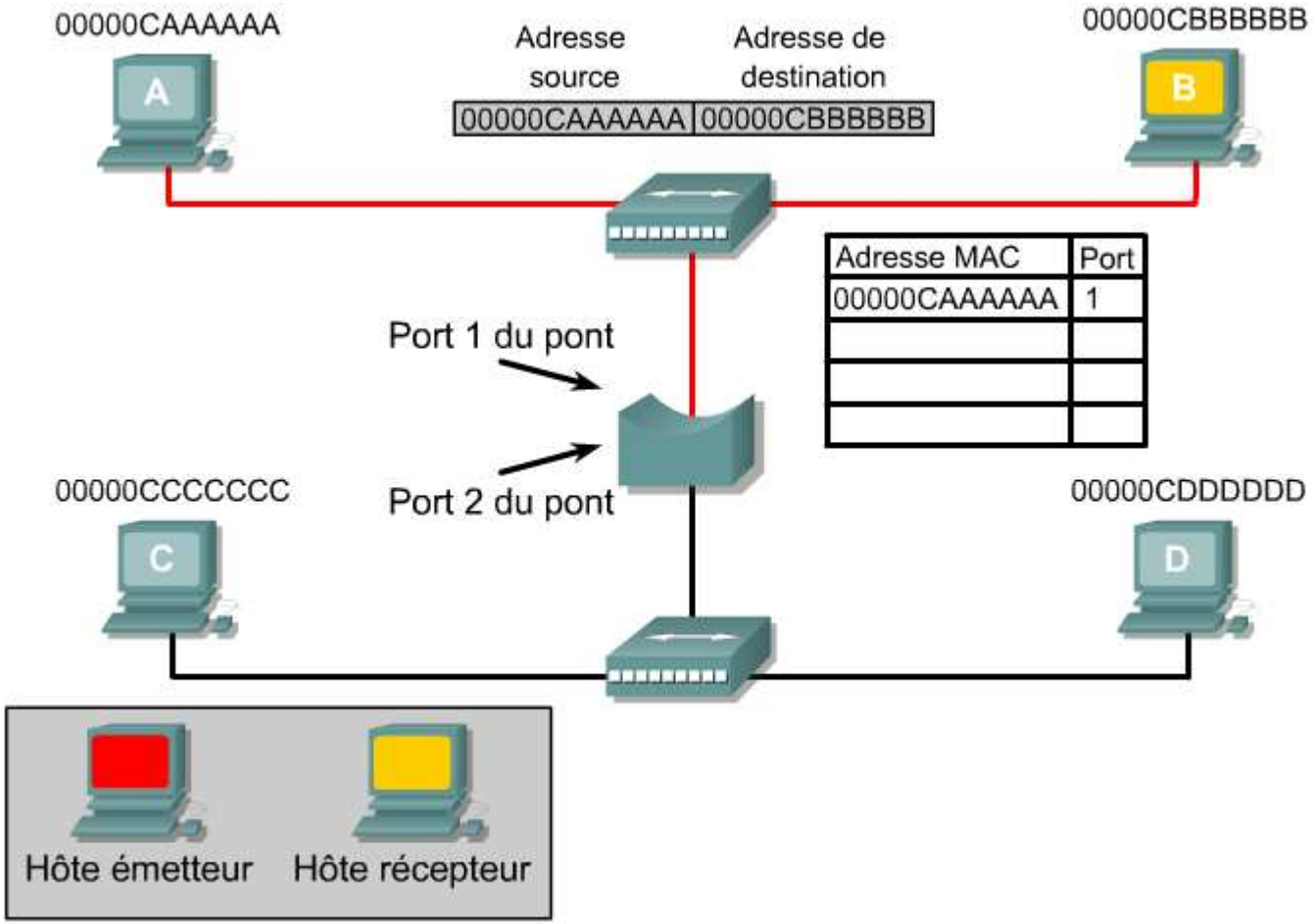
- **Utilisation d'une *table de correspondance* entre les adresses MAC et les ports associés.**
- **Le pont transmet ensuite les trames, ou les rejette, en fonction *des entrées de la table*.**
- **A la réception d'une trame, le pont examine l'adresse MAC destination**
  - **Il consulte la table d'acheminement (*on parle de pont sélectif*)**
    - La trame est dirigé vers **le port** sur lequel est raccordé **le segment de la station destinataire du message**
    - Si la trame appartient au segment raccordé au port d'arrivée, **elle est éliminée (action filtrage)**



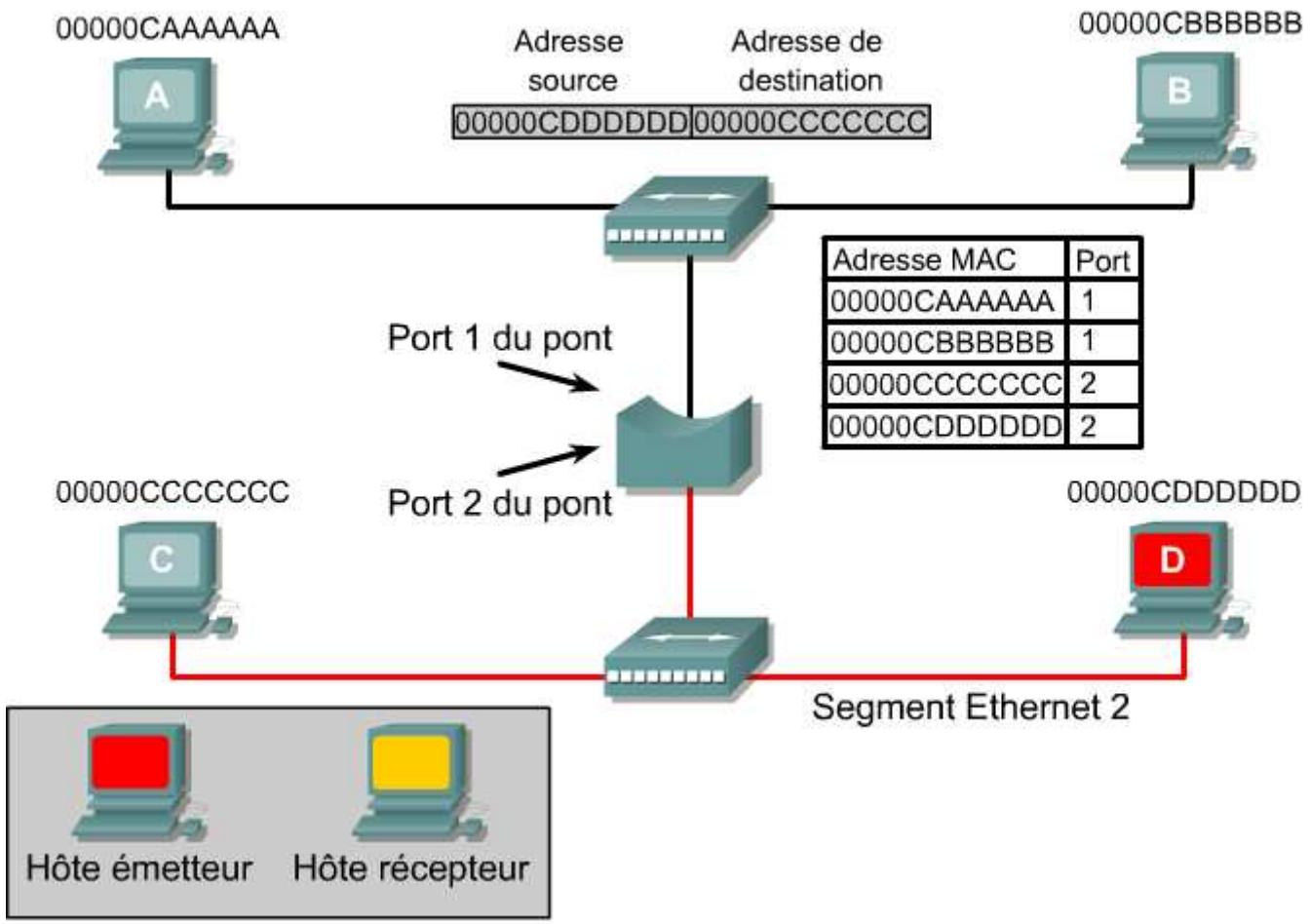
- **Lorsqu'un pont vient d'être installé, sa table est vide.**
- **Le pont attend le transfert de données sur le segment.**
  - **Dès qu'une transmission est détectée, le pont la traite.**



- **L'hôte A émet une requête vers l'hôte B.**
- **Les données étant transmises sur l'ensemble du segment de domaine de collision, le pont et l'hôte B traitent le paquet.**



- **Le pont ajoute l'adresse source de la trame à sa table de pontage.**
  - **Sachant que l'adresse se trouve dans le champ d'adresse source et que la trame est reçue sur le port 1, la trame doit être associée au port 1 dans la table.**


00000CAAAAAA
A
Adresse source
Adresse de destination
00000CDDDDDD
00000CCCCCCC
00000CBBBBBB
B
Adresse MAC
Port
00000CAAAAAA
1
00000CBBBBBB
1
00000CCCCCCC
2
00000CDDDDDD
2
Port 1 du pont
Port 2 du pont
00000CCCCCCC
C
00000CDDDDDD
D
Segment Ethernet 2
Hôte émetteur
Hôte récepteur

- **Etc.**

Types de Ponts

- **Les ponts simples (fonctionnement comme de HUBs)**
  - Pas de fonction d 'acheminement
  - Diffuse toutes les trames reçues sur tous les ports
  - Ils ne sont plus utilisés

- **Les ponts simples**
  - Avec fonction d 'acheminement
  - Dirige les trames reçues selon une table d 'acheminement introduite **à la configuration** (routage statique <=> Manuelle)

- **Les ponts transparents ou d 'apprentissage (learning bridge)**
  - Capables de construire dynamiquement la table de routage et de la maintenir à jour
  - (Protocole de Spanning Tree : STP/Spanning Tree Protocol: routage automatique)

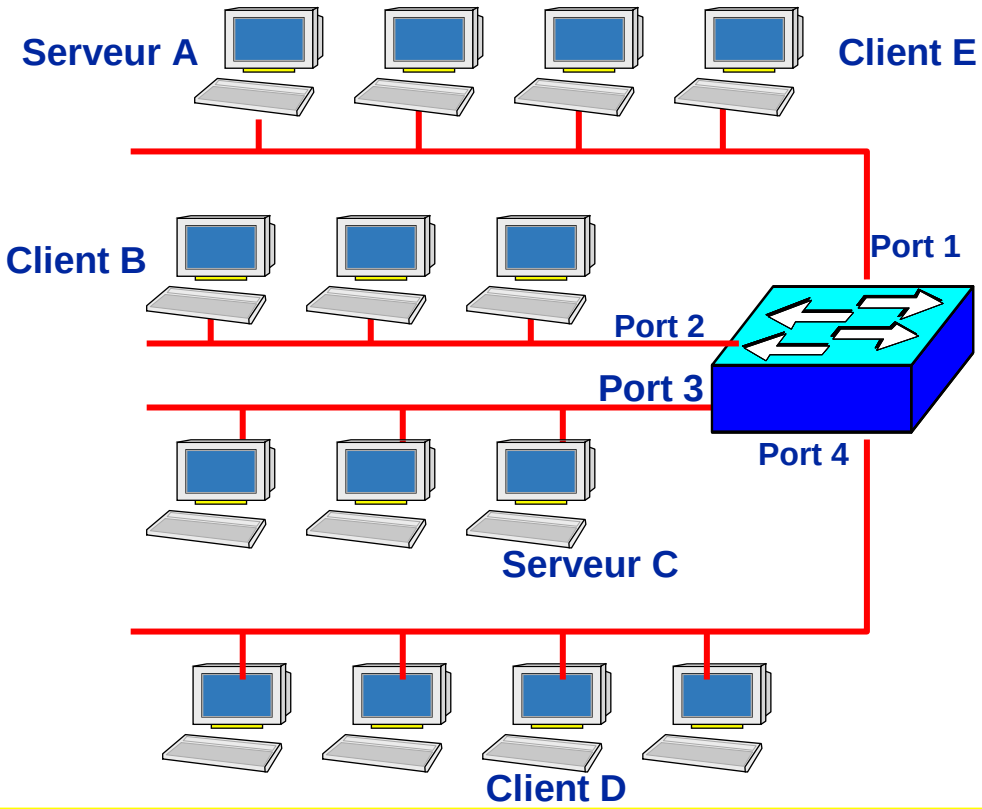
- **Les ponts à routage par la source ou contrôlés par l 'émetteur (Source Routing)**
  - Basé sur les adresses MAC. La route suivie par la trame est indiquée dans la trame elle-même (⇔ utilisation des ***trames de découvertes de routes*** préalables pour découvrir la route vers une destination – cette route est ensuite mémorisée dans la station source)

- **Les ponts distants (Remote Bridge) :**
  - interconnectent des réseaux locaux via une **liaison spécialisée** ou un réseau de transport

DES PONTS VERS LES SWITCHS / COMMUTATEURS

Les switchs (commutateurs)

- **Fonctionnent au niveau liaison**
- **Mêmes fonctions qu'un pont mais utilisent des ports dédiés et non partagés**
- **Commute les trames au niveau MAC**
- **Peut gérer simultanément plusieurs communications**



Echange simultanée :

A (port 1 <-> B (port 2)

C (port 3) <-> D (port 4)

Echange Non commuté :

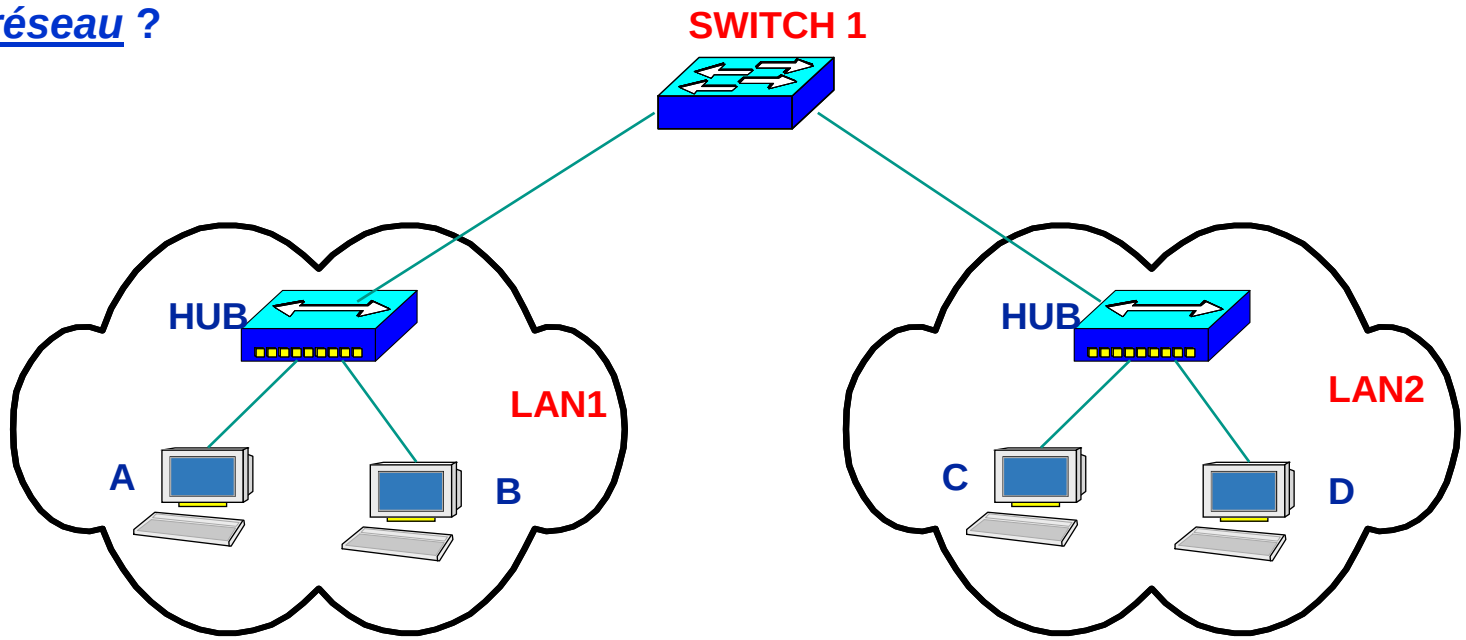
A (port 1) <-> E (port 1)

Débits sur les Switchs (commutateurs)

- **Débit maximal offert**
  - **Dans le meilleur des cas**
    - **Soit un commutateur Ethernet de N ports à Y Mbps**
    - ☛ ***N/2 connections (ou canaux virtuels) à Y Mbps peuvent être établies simultanément entre N/2 paires de ports***
  - **Exemple**
    - **Un commutateur Ethernet de 128 ports à 10 Mbps**
      - ✓ ***Peut offrir un débit maximal de 128/2*10 = 640 Mbps***
      - ✓ ***Chaque paire de ports (ou stations) connectées ayant droit à un canal de 10 Mbps***
    - **Un hub Ethernet de 128 ports à 10 Mbps**
      - ✓ ***Peut seulement offrir à chaque station un débit moyen de 10/128 = 80 Kbps***

Routage « commutation » dans les switch – protocole spanning Tree

- **Problème posé:**
  - Ethernet fonctionne bien à faible et moyenne charge
  - Et si on augmente le nbr de Hub ⇒ Domaine de collision ↗
  - Conséquence : Améliorer les performances ⇔ diminuer les *collisions* et la *charge d'un réseau* ?



- **Comment ? : Solution : Switch1 = Hub avec buffers pour réduire les collisions**
  - Reçoit la trame complète avant de la renvoyer (en respectant le CSMA/CD sur chaque port)
    - Exemple: A→C et D → B, A → B, broadcast (*pas de collisions possibles entre A et D*)

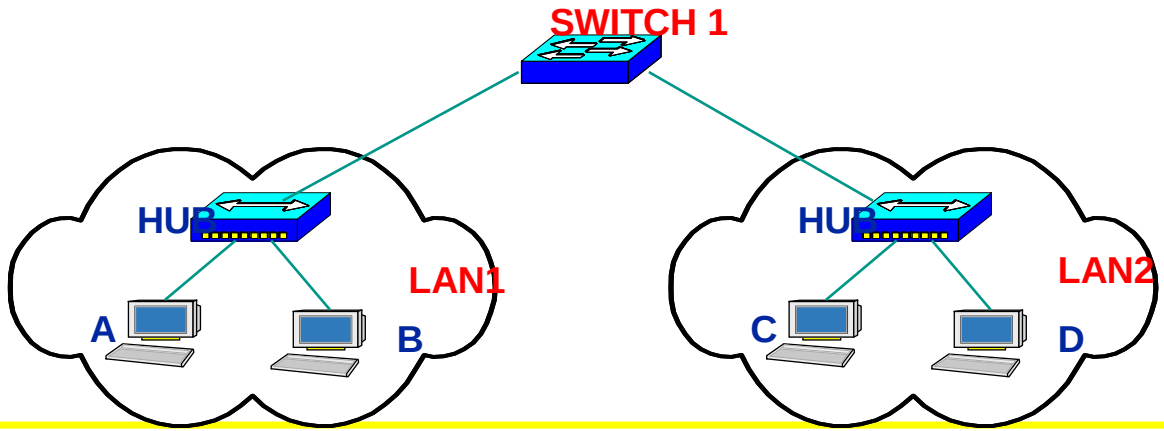
- **Donc, notre Switch1 avec des buffers :**

1. **Réduit les collisions en évitant grâce à des buffers que deux trames émises en même temps (ex: une sur LAN1 et une sur LAN2) n'entrent en collision**
2. **Par contre, il ne permet pas de réduire la charge. En effet, toutes les trames passent sur LAN1 et LAN2**

- **Question : Comment réduire la charge ?** ⟺ Le switch 1 : est un switch qui se contente de recevoir les trames sur ses ports d'entrée et les renvoyer sur ses ports de sortie en respectant la méthode **CSMA/CD**

  - **Idée de base de la solution : « modifier le fonctionnement du switch 1 en rajoutant une table qui *indique quel port joindre pour atteindre telle station»***

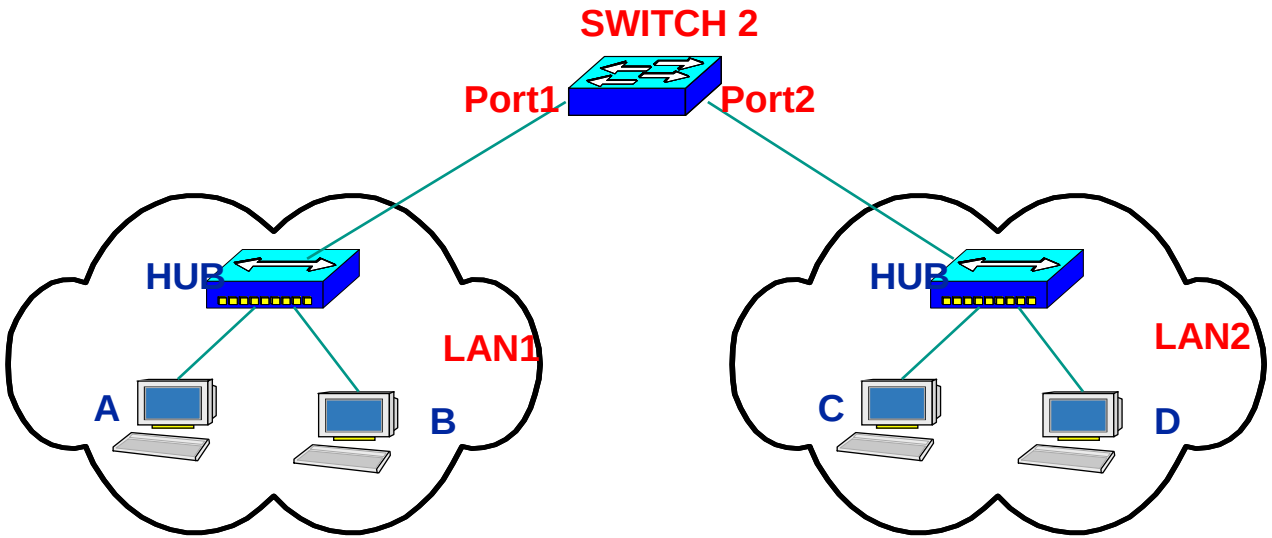
  ⇨ **construire switch2 qui ne transmet pas sur LAN2 les trames A → B et B → A**



⇨ **Switch Sélectif**

Comment construire un switch sélectif ?

Idée : inclure dans switch2 une table qui indique sur quel LAN (port) se trouve chaque adresse MAC (⇔ chaque station)



Port	Station
P1	A
P1	B
P2	C
P2	D

Construction de la table de routage

- **Fonctionnement du switch2**
  1) **Reçoit les trames sur port1 et port2**
  2) **A l'arrivée d'une trame, examen de l'adresse destination**
  3) **Consulter la table ports/ adresses pour voir si il faut retransmettre la trame sur un autre port**
     - **Si oui, transmettre la trame**
     - **Si non, la trame a déjà été émise sur le LAN où la destination se trouve et le switch peut la jeter**
- **Remarques**
  - **Un switch a, en pratique plus de 2 ports**
  - **Support du multicast et du broadcast**
    - **Donc, transmettre une trame sur tous les ports du switch sauf celui d'où la trame a été reçue**

Construction de la table de routage

- Switch sélectif : Comment construire la table port/ station ?

1. **Manuellement**
   - **Ce serait un cauchemar pour l'administrateur du réseau**
   - **Modifier la table chaque fois qu'une machine bouge**

2. **En choisissant les adresses MAC**
   - **Exemple :**
     - **adresses 1. A et 1. B dans LAN1, 2. C et 2. D dans LAN2**
   - **il faut configurer les switchs et les stations**

3. **Automatiquement (⇔ dynamiquement)**
   - **Solution idéale**
   - **Conforme à la philosophie Ethernet ⇔ on peut rajouter ou retirer une station du réseau sans devoir rien configurer sur les stations du réseau**

Construction de la table de routage

- Switch sélectif : Comment construire dynamiquement la table port/ station ?
  - Pour construire la table, il faut savoir à quel port du switch chaque station est attachée
- **Question :**
  - Le switch peut-il déterminer seul à quel port une station est attachée ?
- **Réponse :**
  - Oui en écoutant toutes les trames sur chaque port et en regardant l'adresse source de ces trames

Construction de la table de routage

- **Switch sélectif :**
  - Switch3 : switch sélectif avec ***construction automatique*** de la table port/ station
  - Comment faire lorsque l'on reçoit une trame dont la destination n'est pas dans la table ?

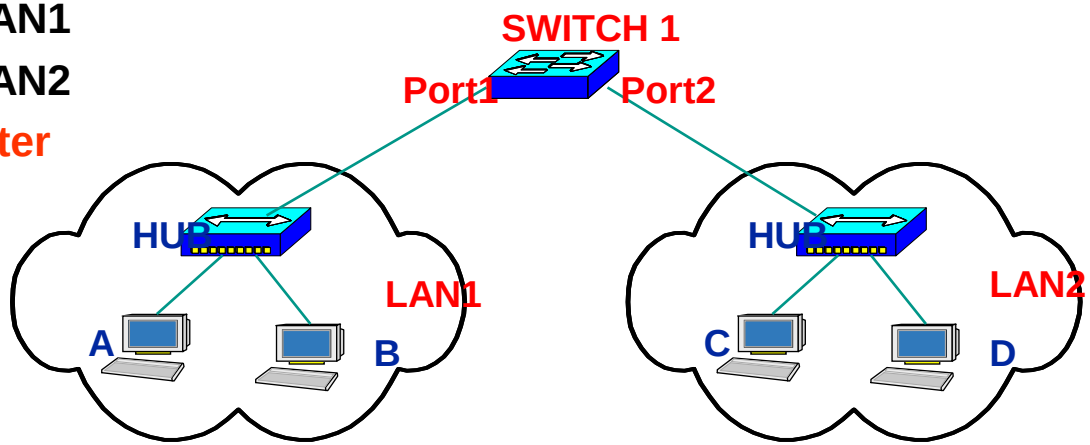
    → *L'envoyer sur tous les ports sauf celui d'où elle vient*
  - Fonctionnement de switch3 : arrivée d'une trame
    1. mise à jour de la table port/ station [addr.source]
    2. transmission éventuelle de la trame sur base de la table port/ station

Construction de la table de routage

- **Construction de la table port / Station**
- **Exemples :**
  - **trame A-> B [P1: A , P2: -] → LAN2**

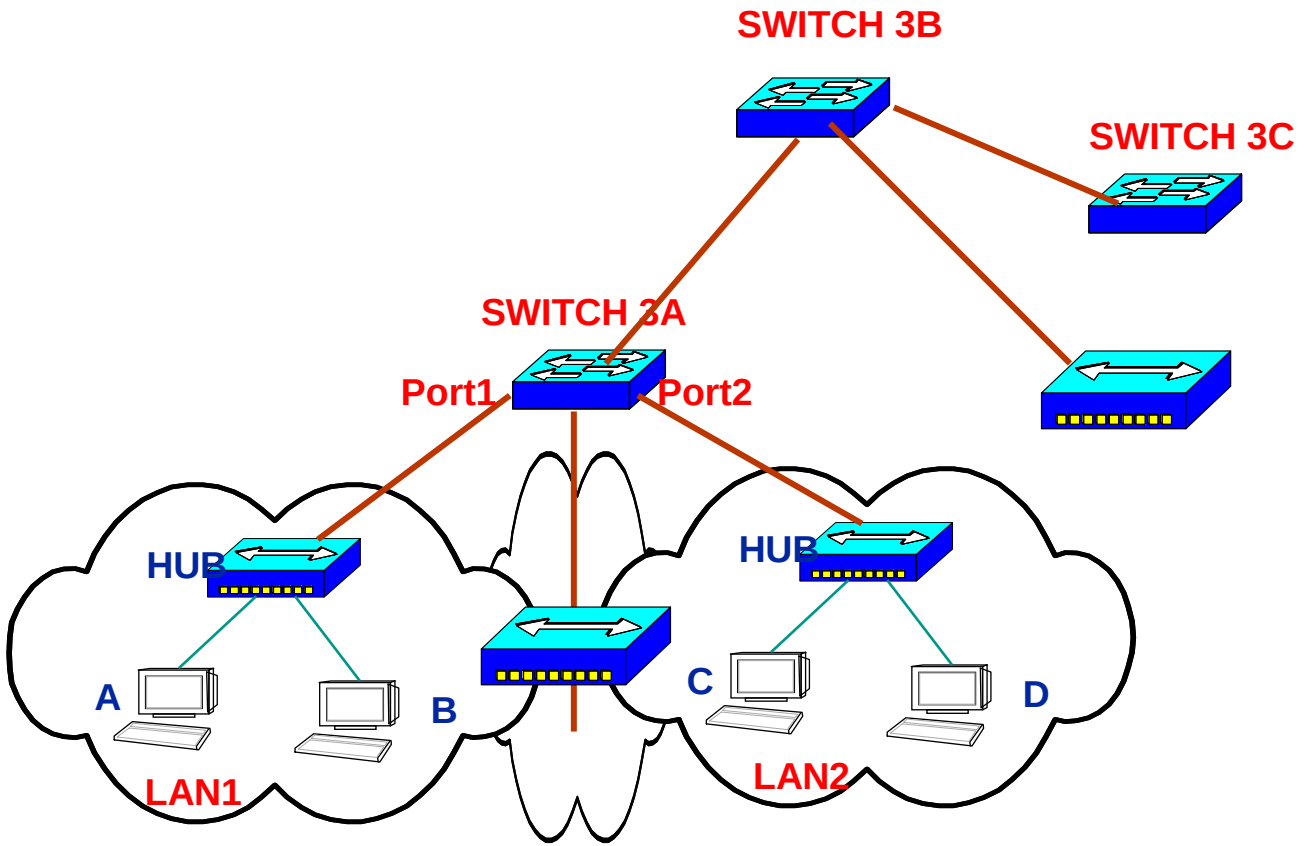
Trame envoyée de A -> B sauf que la destination n 'est pas encore dans la table
=> On diffuse la trame sur tous les autres ports
=> Sur LAN2

- **trame A-> C [P1: A , P2: -] → LAN2**
- **trame C-> D [P1: A , P2:C] → LAN1**
- **trame D-> C [P1: A , P2:C,D] → jeter**
- **trame A-> B [P1: A , P2: C,D] → LAN2**
- **trame C-> B [P1: A , P2:C,D] → LAN1**
- **trame B-> C [P1: A ,B, P2:C,D] → LAN2**
- **trame A-> B [P1: A,B , P2:C,D] → jeter**



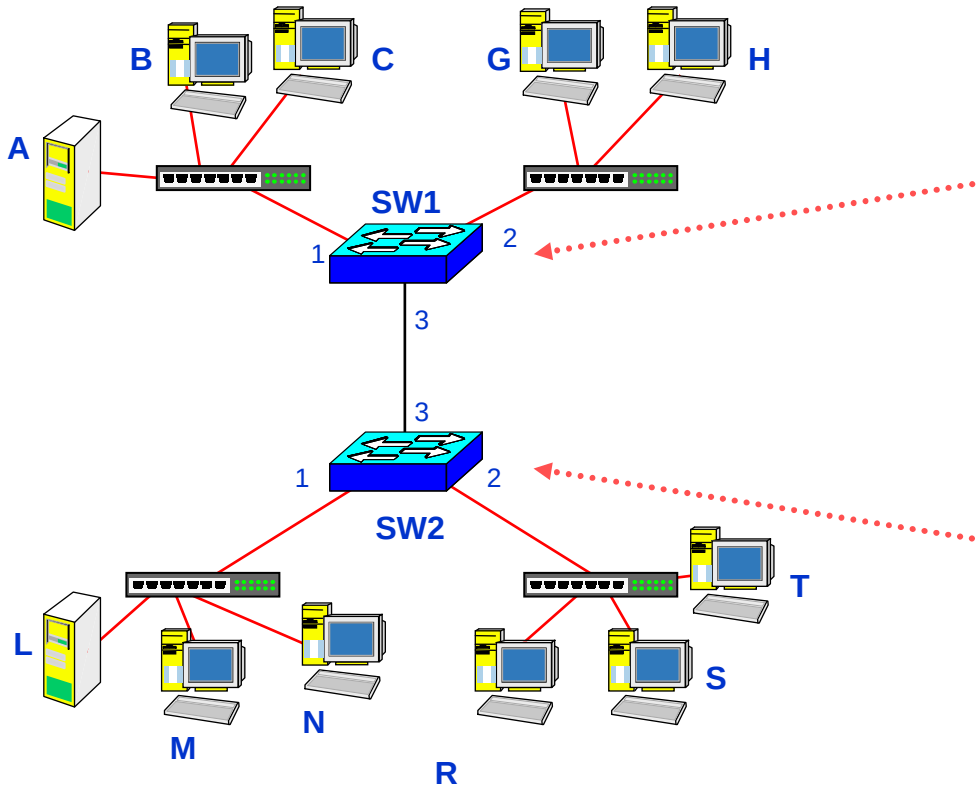
Construction de la table port/station

- **Fonctionne également pour des switchs avec plus de deux ports et dans des réseaux complexes (des switchs en cascade)**
- **N.B : un switch actuel peut stocker jusqu'au 8000 adresses MAC actifs (dans une période donnée)**



Exercice

- **Calculer les tables port station des deux switchs SW1 et SW2**

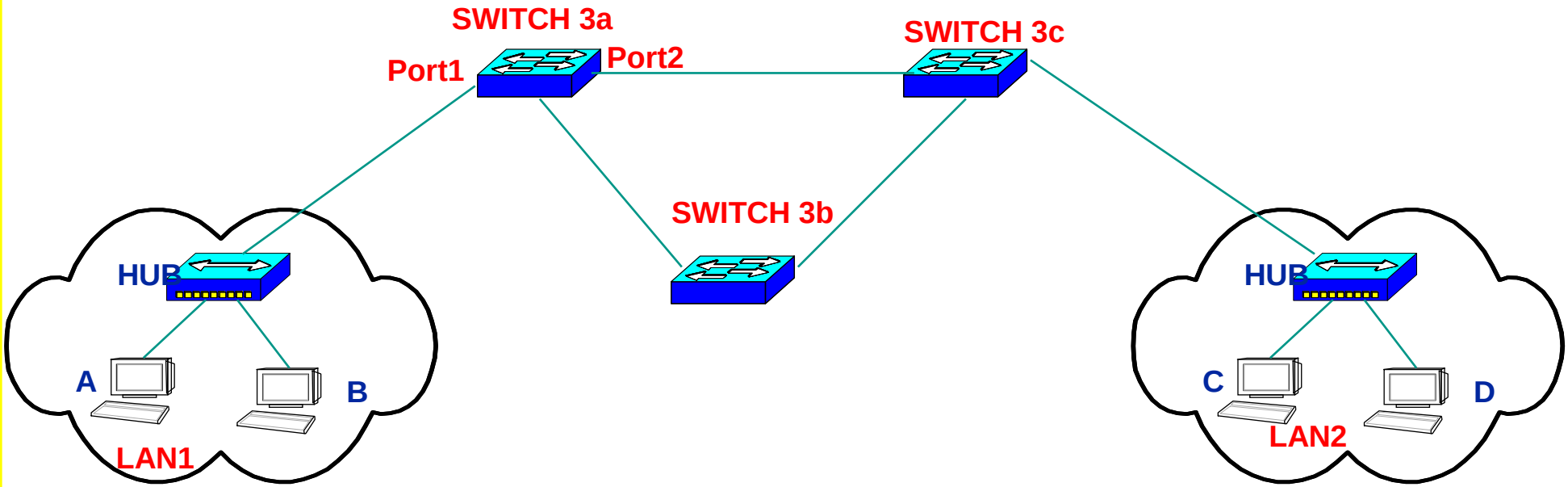


Port 1	Port 2	Port 3

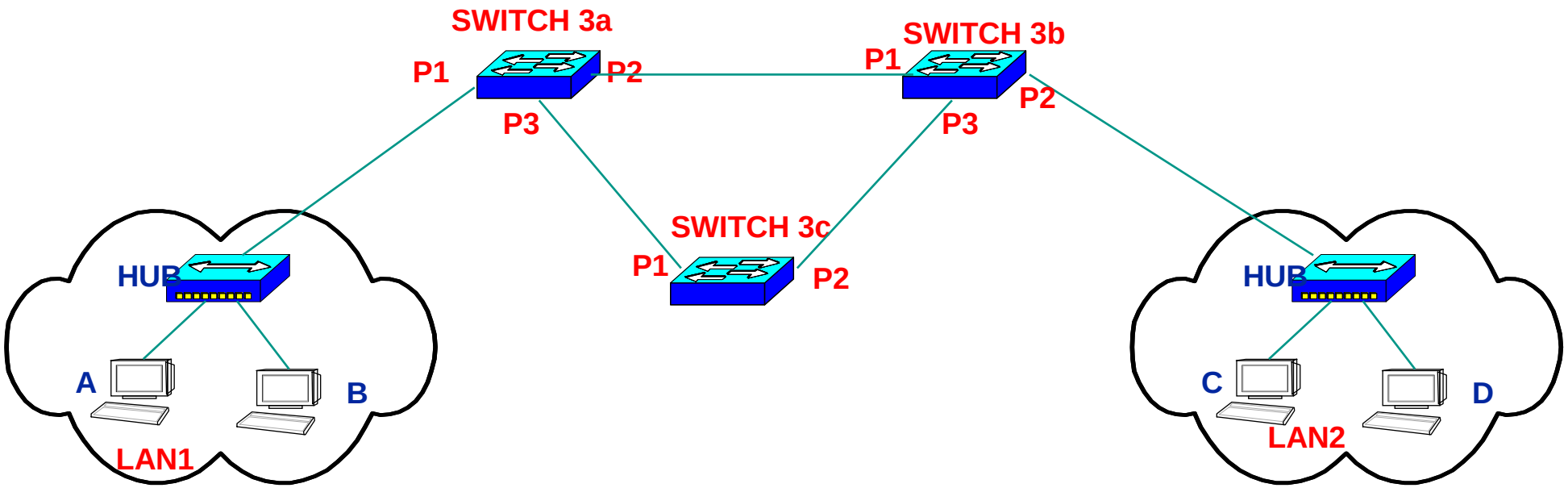
Port 1	Port 2	Port 3

solution

- **Construction de la table port/station : Fiabilité du réseau**
  - **Si un switch ou un hub tombe en panne, le réseau peut être coupé en deux**
  - **Comment améliorer la tolérance aux pannes ?**
  - **Idée**
    - **Introduire des liaisons redondantes**



TD N°2 : Effet de boucle ?



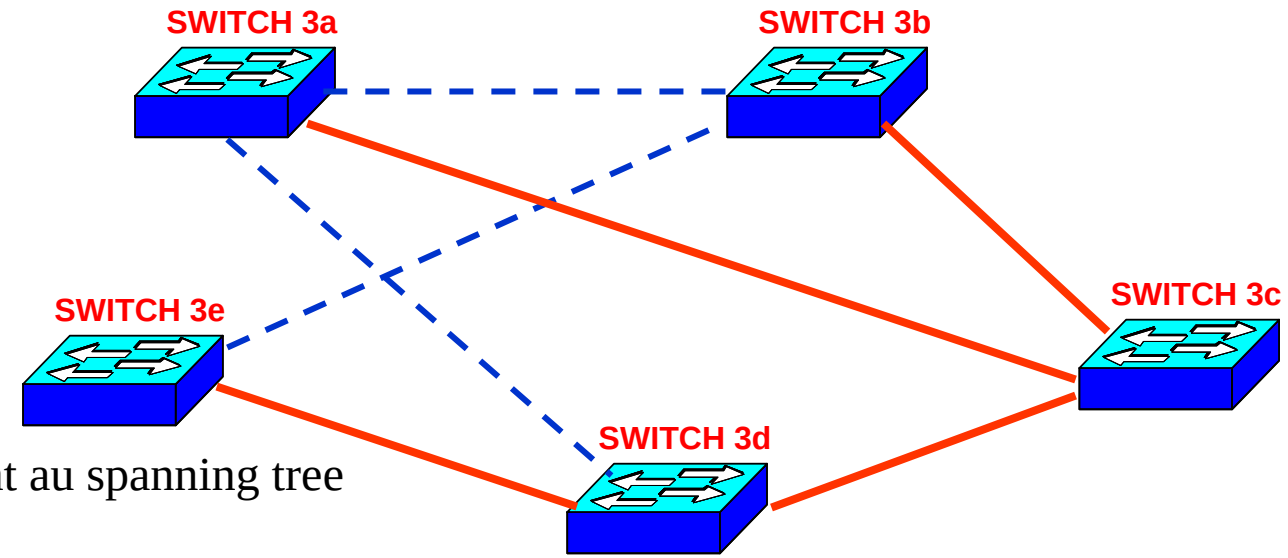
Solution

Construction de la table de routage

- Construction de la table de routage-Effets de la boucle :
- L'exemple précédent montre qu'une boucle peut avoir un effet désastreux dans un réseau Ethernet commuté
- Comment résoudre ce problème ?
- Solutions !

1. **Interdire les switchs et n'autoriser que des hubs (pratique mais solution pas intéressant de point de vue perf)**
2. **Interdire les boucles (enlever les liaisons de redondances ⇔ Pb de fiabilité)**
3. **Modifier les switchs pour qu'ils détectent les boucles (*avec des alarmes*) et s'arrêtent de fonctionner lorsqu'une boucle est découverte**
4. **Développer un algorithme qui *supprime automatiquement et dynamiquement* les boucles**

- **Spanning Tree (⇔ Arbre de recouvrement minimum)**
- **Dans un réseau / graphe, l'arbre de recouvrement minimum [Spanning tree] ne contient aucune boucle**
  - **Il est suffisant pour assurer que l'on peut atteindre toutes les stations du réseau (nœuds de l'arbres) à partir de la racine de l'arbre.**



Appartient au spanning tree

Hors du spanning tree

Au début de fonctionnement, on construit l'arbre de recouvrement minimum et on met inactifs tous les liens qui ne font pas partie de l'arbre

Spanning Tree

⊙ Construction du Spanning Tree ; caractéristiques du STP

☞ Algorithme distribué :

utilise un protocole (IEEE 802.1D) implémenté dans les switches [bridges]

☞ Objectifs du protocole : si on veut construire un arbre de recouvrement minimum, il faut construire un arbre. Chaque arbre a une racine, donc ;

1 - Elire la racine du Spanning tree

2 - Calculer la distance à la racine du Spanning Tree

3 - Lorsque plusieurs Switchs sont connectés au même LAN,

– déterminer celui qui doit transmettre les trames hors du LAN (et désactiver les autres

4 - Déterminer les ports/ liens à inclure dans le Spanning Tree

5 - Déterminer le port sur lequel les messages 802. 1D sont reçus

- Construction de Spanning Tree
- Protocole 802.1D:
  - Le protocole 802.1D utilise des messages avec les informations suivantes :
    1. **Root ID : identification actuelle de la racine**
       - **Chaque switch a un N° particulier et dans le message on transmet le N° de la racine actuelle.**
    2. **Cost : Coût du lien (chemin) de plus faible coût entre le switch qui transmet le message et la racine**
    3. **Transmitting ID : identification du switch qui transmet le message**
       - **identification unique sur 48 bits**
  - Ces messages sont envoyés en multicast [adresse: all switches] pour qu 'elles soient reçus par tous les switch du réseau
       - ***mais ne sont jamais retransmis par les switches »***

■Construction du Spanning Tree :

■**Définir une relation d ordre entre les messages 802.1 :**

permet de spécifier dans quel cas un message est meilleur qu 'un autre

Messages du protocole 802.1D

M1[R1, T1, C1] est meilleur que M2[R2, T2, C2]

Si :

- R1 < R2 (la racine la plus petit)
- Ou bien : R1 = R2 et C1 < C2 (le coût le plus faible)
- Ou bien : R1 = R2 et C1 = C2 et T1 < T2 (le transmitteur le plus petit)

M1			M2		
R1	T1	C1	R2	T2	C2
29	35	15	31	32	12
35	39	80	35	40	80
35	80	15	35	38	18

Remarque : On peut toujours avoir une relation d 'ordre entre les messages à partir du moment ou deux messages sont envoyés par deux switchs qui ont deux numéros différents

- *Construction du spanning tree / Fonctionnement du protocole :*

1. Au démarrage, un switch se considère comme la racine et transmet un message 802.1 avec un coût de **0** sur tous ses ports

2. Sur chaque port, le switch reçoit en permanence les messages ***802.1D*** et sauvegarde pour chaque port le meilleur message qui a été reçu sur ce port

3. Si un switch reçoit sur un port un meilleur message 802.1 que celui qu'il transmettrait, alors *il arrête de transmettre son message* ***802.1D*** *(sur ce port)*

4. Le protocole se stabilise lorsque un seul switch transmet ses messages 802.1D sur chaque LAN

Construction du Spanning Tree

Détermination de la racine

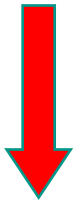
Principe :

- Dans un réseau la racine est le switch avec le plus petit identificateur
- Chaque switch peut déterminer quelle est la racine en analysant tous les messages 802.1 reçus sur ces différents ports

- **Exemple**
  - Messages reçus par le switch 18

	Root	Cost	Trasmitter
Port1	12	93	51
Port2	12	85	47
Port3	81	0	81
Port4	15	31	27

Meilleur Msg reçu de la racine ?

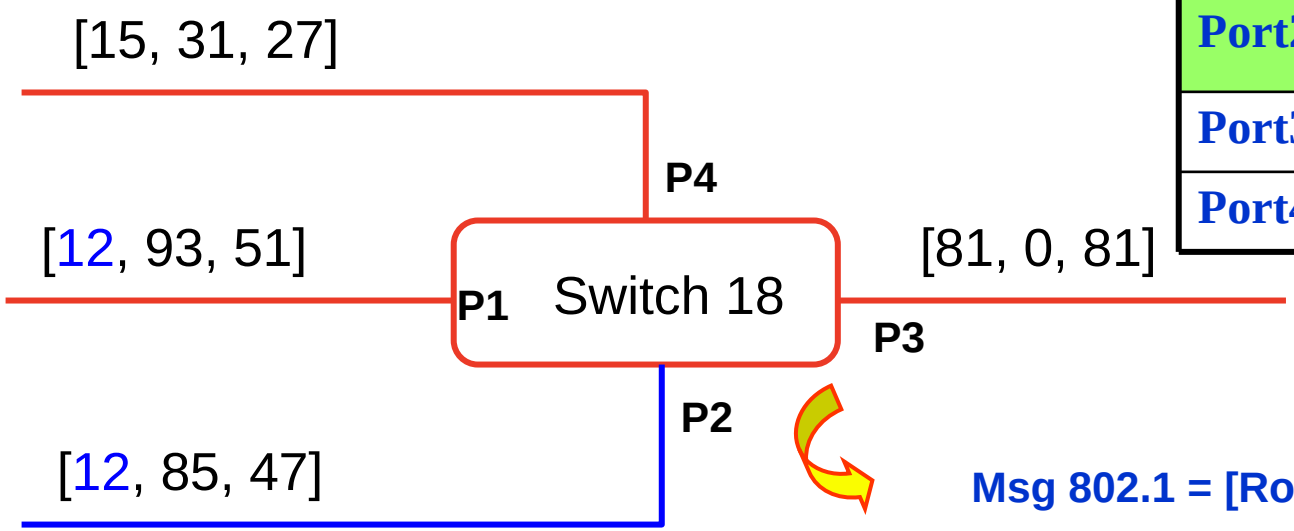


Meilleur racine qu 'il a entendu = 12

Construction du message 802.1 du switch

- Exemple : Messages reçus par le switch 18
  - Pour le switch 18, la racine est le switch 12 (meilleur racine qu'il a entendu ⇔ le **Root_Id** le plus petit)
  - Meilleur Msg reçu est celui qui est reçu sur port 2 (venant de la racine)
  - Et le plus court chemin vers la racine passe par le port2

	Root	Cost	Trasmeitter
Port1	12	93	51
Port2	12	85	47
Port3	81	0	81
Port4	15	31	27



Msg 802.1 = [Root, Cost, Transmitter]

Port racine ? / Détermination de la distance à la racine ?

♦ Port racine ?

- le port racine d'un switch est le port qui se trouve topologiquement le plus proche de la racine
  - ☞ *« port 2 dans l'exemple précédent »*
- c'est via son port racine qu'un switch reçoit les messages 802.1

♦ Détermination de la distance à la racine ?

- Si le switch est la racine, alors
  - ☞ la distance est zéro par définition
- Sinon,
  - ☞ la distance à la racine est 1+ la distance du message reçu sur le port "racine" du switch

Donc, les Msg 802.1D du switch 18 dans ce cas est : [Root, Cost, Transmitter]

M= [R=12, C =86, T=18]

Etats d ’un port Vis-à-vis des messages 802. 1

1.Port racine :

- Définition : port via lequel on reçoit les messages provenant de la racine par le plus court chemin
- un port racine ne retransmet pas de messages 802.1 reçus de la racine
- un seul port racine par switch

2.Port désigné

- Définition : un port est désigné si le message du switch est meilleur que le message reçu sur ce port
- port via le(s) quel(s) le switch retransmet les messages 802.1 reçus de la racine
- message 802. 1 transmis par un switch ?

  « racine actuelle, identification du switch et coût entre ce switch et la racine »
- zéro, un ou plusieurs ports désignés sur un switch

3.Port bloqué : reçoit uniquement les messages **802.1D**

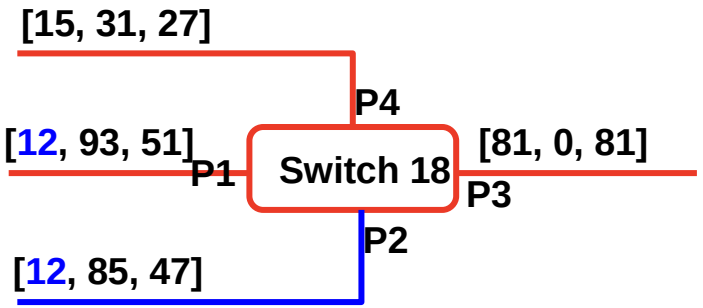
Détermination de l'état d'un port

- Exemple

	Root	Cost	Trasmitteur
Port1	12	93	51
Port2	12	85	47
Port3	81	0	81
Port4	15	31	27

R1 < R2
R1= R2 et C1 < C2
R1 = R2 et C1 = C2 et T1 < T2

M= [R=12, C =86, T=18]



port2 est le port racine

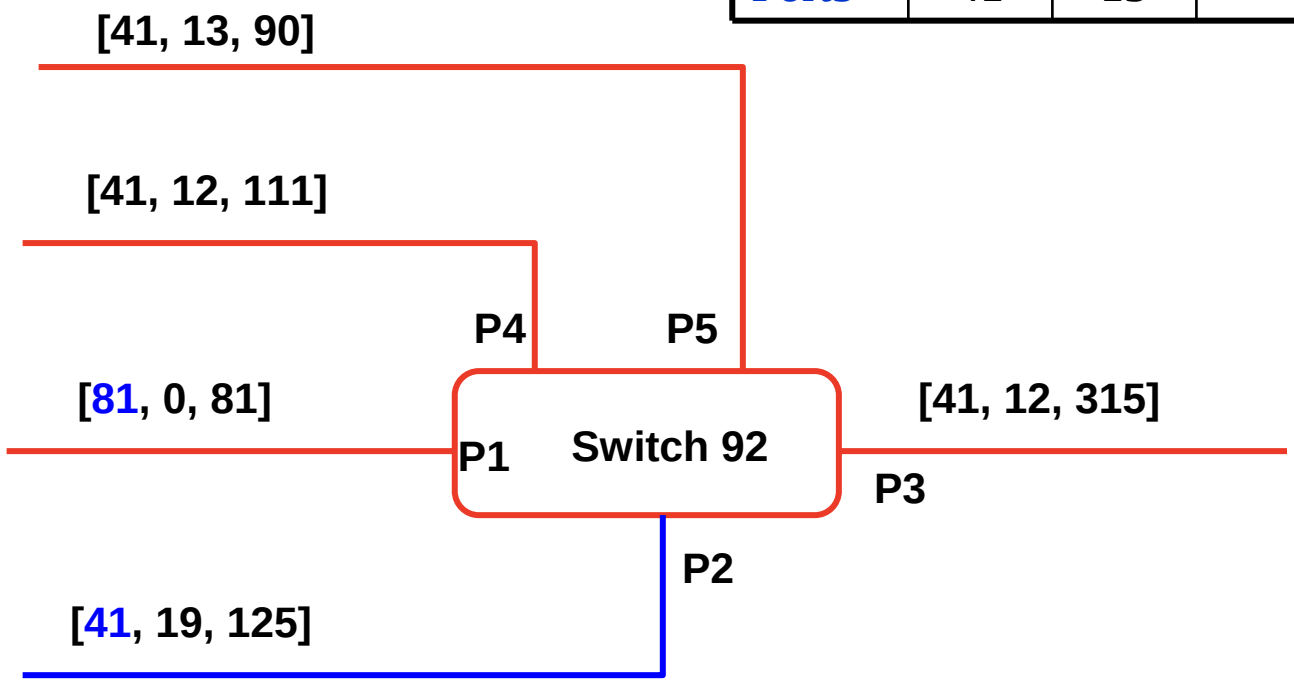
message du switch : R=12, T=18, C =86 (85+1)

Ce message est meilleur que celui reçu sur les ports : Port1, Port3 et Port4 et donc les trois ports sont en état désigné

Détermination de l'état d'un port : Un autre exemple

Message reçu par le switch 92

	Root	Cost	Trasmeitter
Port1	81	0	81
Port2	41	19	125
Port3	41	12	315
Port4	41	12	111
Port5	41	13	90



Exemple – suite

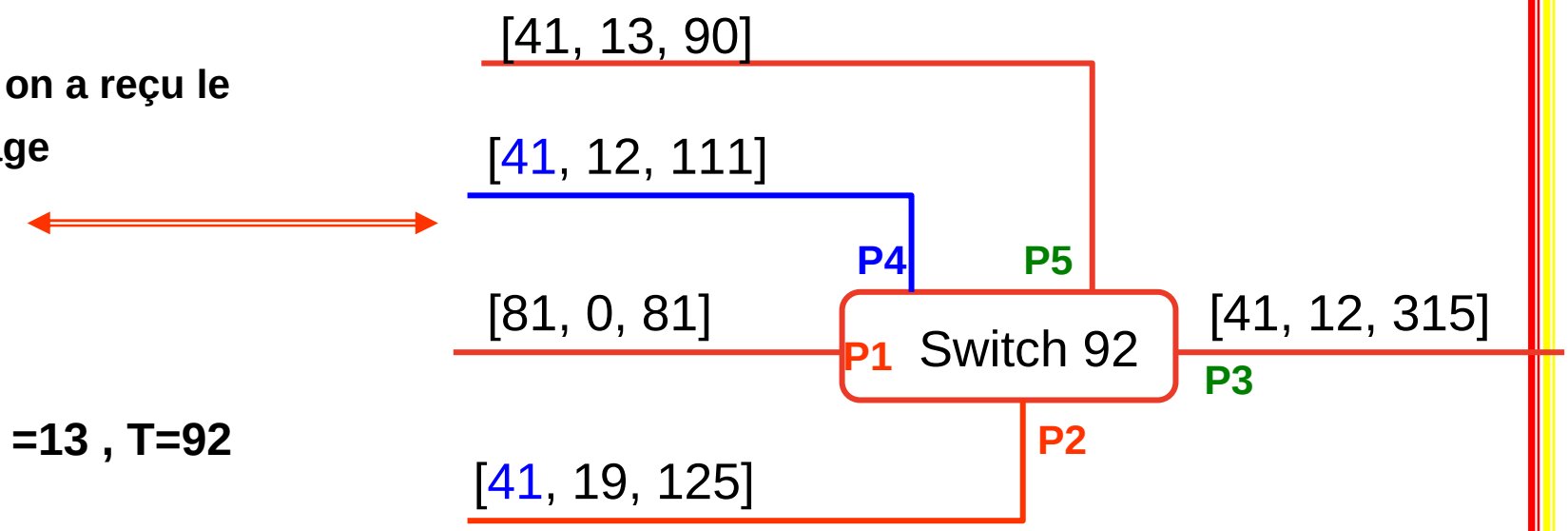
a. **Racine : switch 41**

b. **Port racine : port4 (⇔ le port sur lequel on a reçu le meilleur message)**

c. **construction du message du switch 92 : R= 41, T=92, C =13 (12+1)**

- Ce message est meilleur que celui reçu sur les ports 1, 2 donc les deux ports sont en état désigné
- Ce message n 'est pas meilleur que celui reçu sur les ports 3,5 donc les deux ports sont en état bloqué

Port sur lequel on a reçu le meilleur message



R= 41, C =13 , T=92

N.B : *Les états des ports (Racine, Désigné et Bloqué) sont définis par rapport aux messages* ***802.1D***

Etats d ’un port Vis-à-vis des trames de données

1. Port actif :
   - Le switch écoute les trames via ce port et les retransmet (sélectivement) vers les autres ports actifs du switch si nécessaire (selon la table de routage)
2. Port inactif
   - Le switch n’écoute pas les trames via ce port et ne retransmet aucune trame de données reçu via ce port.

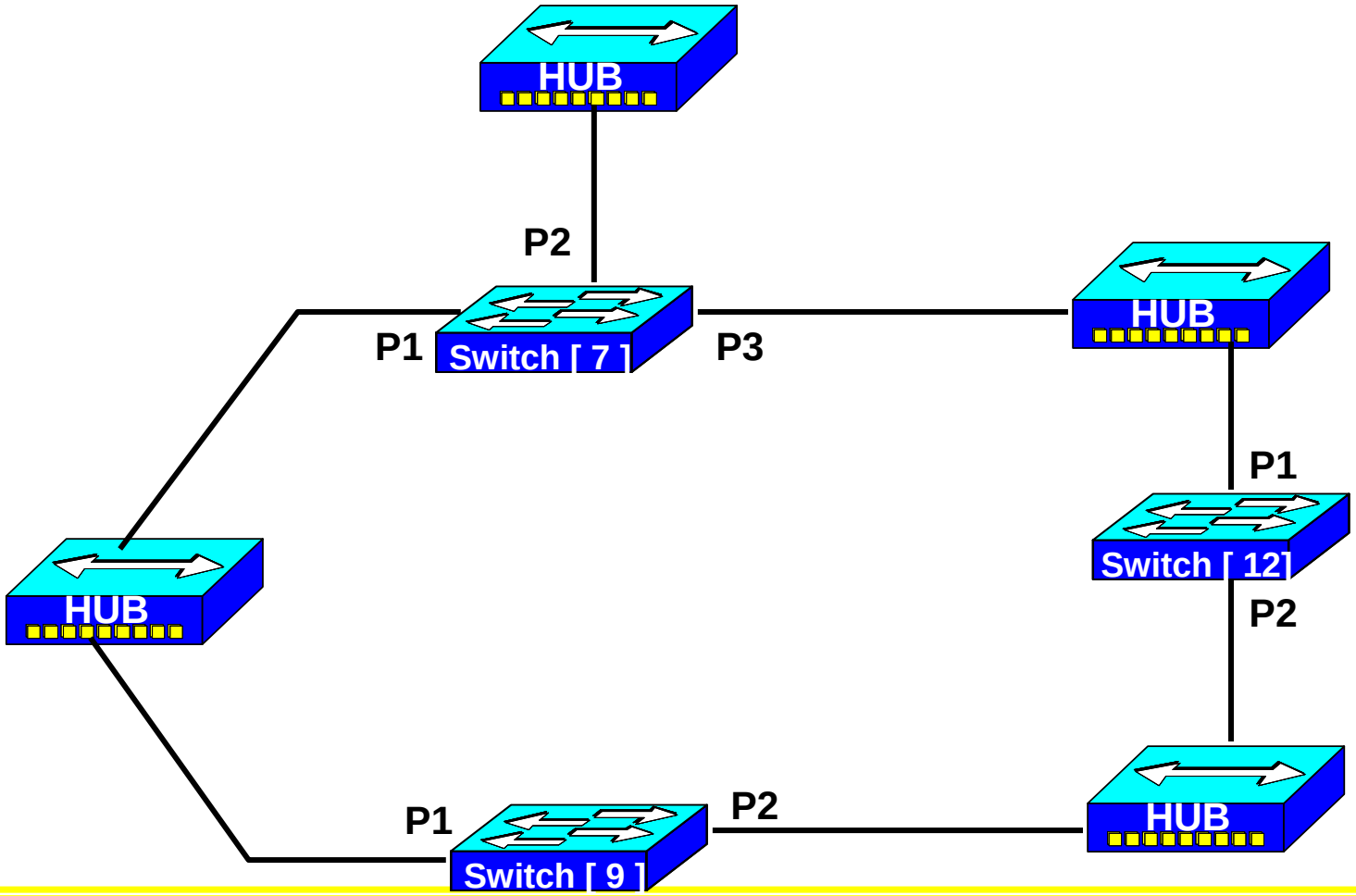
III. Activités des ports (détermination des activités des ports):

- Initialement, tous les ports sont inactifs
- Les ports racine et désignés deviennent actifs lorsque la topologie du STP est stable
- Les ports bloqués restent inactifs (un port bloqué ne fait que recevoir les msg 802.1D)

- Remarque :
  - Initialement, on met tous les ports à l'état inactif, car puisque on fait affaire à des réseaux avec boucles, ceci permet d 'éviter de transmettre des données au départ et attendre jusqu'à ce que l 'algorithme soit stabilisé.

TD N°3 – Algorithme de STP

- Au départ, on considère que les switchs ne sont pas allumés
  - On allume le switch 12
  - Ensuite, le switch 9
  - Et en fin, le switch 7
- On demande d'appliquer l'algorithme de STP pour construire une topologie stable et sans boucle



Solution

- **Comment réagir aux pannes ?**
  - **Supprimer une boucle dans un réseau existant c'est important pour éviter que les messages tournent indéfiniment dans le réseau.**
    - **Mais ce n'était pas le problème qu'on cherchait à résoudre.**
  - **Le problème à résoudre était d'avoir un réseaux de *switchs tolérant aux pannes* avec des liens multiples de telle façon qu'on puisse si une liaison ou un autre switch tombe en panne d'utiliser d'autres liens**
  - **Comment on va pouvoir faire ça au niveau du protocole 802.1D ?**
    - → **Ce qu'il faut faire, c'est que lorsqu'on a une panne sur un lien de forcer un re-calcul du spanning tree (⇔ reconstruction d'un arbre de recouvrement minimum).**
    - → **En pratique, on va ajouter une détection de pannes dans 802.1D**

Réaction Aux Pannes –2-

- Comment réagir aux pannes ?
  - **Ajout, dans le message 802.1D une information supplémentaire : age**
  - Détection des pannes
    1. *La racine transmet périodiquement un message 802.1 avec age = 0*
    2. *si un switch reçoit ce message (pour la première fois) via son port racine (soit directement soit indirectement), il le stocke et envoie son message 802.1*
    3. *Le switch associe Un timer au message reçu,*
       - *L'information age est incrémenté de 1 à chaque expiration du timer*
    4. *Si age dépasse une certaine limite (age_max ⇔ 20 secondes par défaut), le message est effacé et on recalcule le spanning tree*

- **Exemple**

	Root	Cost	Transmitter
Port1	81	0	81
Port2	41	19	125
Port3	41	12	315
Port4	41	12	111
Port5	41	13	90

- **Rappel : Etats des ports**
  - ports 1 et 2 sont désignés,
  - port 4 est racine
  - ports 3 et 5 sont bloqués

- **Exemple : expiration du message reçu sur port 4**
- **Périodiquement**, on va recevoir sur le port racine (port 4) les messages 802.1 qui proviennent par le chemin le plus court de la racine
- **Si ce message 802.1 expire** (⇔ on n'a pas reçu le rafraichissement du message 802.1 pendant un certain temps),
  - on va supprimer la ligne correspondante au **port 4** de la table
  - On force un recalcule de l'état du switch

    → en oubliant la ligne 4

	Root	Cost	Transmitter
Port1	81	0	81
Port2	41	19	125
Port3	41	12	315
Port5	41	13	90

- => Dans l'exemple, c'est le **port 3** qui va nous donner la **nouvelle racine**.
  - La **racine / port** du switch reste dans ce cas le **switch 41** et le **coût reste 13**, donc les message 802.1D du switch reste inchangé et la retransmission de ce message ne provoque aucun changement

- **Exemple port 3 (son temporisateur expire également) et port 4 sont en timeout**
  - **Messages reçus par le switch 92**

	Root	Cost	Transmitter
Port1	81	0	81
Port2	41	19	125
Port5	41	13	90

  - **Racine = 41**
  - **port racine = 5**
  - **message = [R=41, T=92 ,C=14], ce nouveau messages va provoquer un changement dans l'état des différents ports, les ports 1, 2, 3 et 4 vont passer dans l'état désigné**
  - **Et s'il y a un changement dans le switch 92, ça va provoquer un changement dans les autres Switchs par effet de cascade**

Implémentation du protocole de Spanning Tree

(IEEE 802.1D)

The Spanning Tree Protocol (IEEE 802.1D) dans la pratique

- STP est supporté par tous les switchs au niveau MAC
  - Est exécuté automatiquement (à moins qu'il soit désactivé)
- Les Switchs échangent des messages de topologie (BPDU's : Bridge Protocol Data Unit) pour construire l'arbre de STP
  - Topologie sans boucle
- Une fois STP est stable,
  - certains ports des bridges retransmet les trames "MAC" de données
  - Les autres ports sont bloqués
  - Ce qui permet de créer un seul chemin logique entre deux LANs
- Une fois, STP est configuré, le protocole s'exécute périodiquement (BPDU envoyés toutes les deux secondes – configurable-) pour prévenir des changements dynamiques de la topologie
  - Si une panne se produit dans la topologie, l'algorithme est déclenché pour recalculer une nouvelle topologie

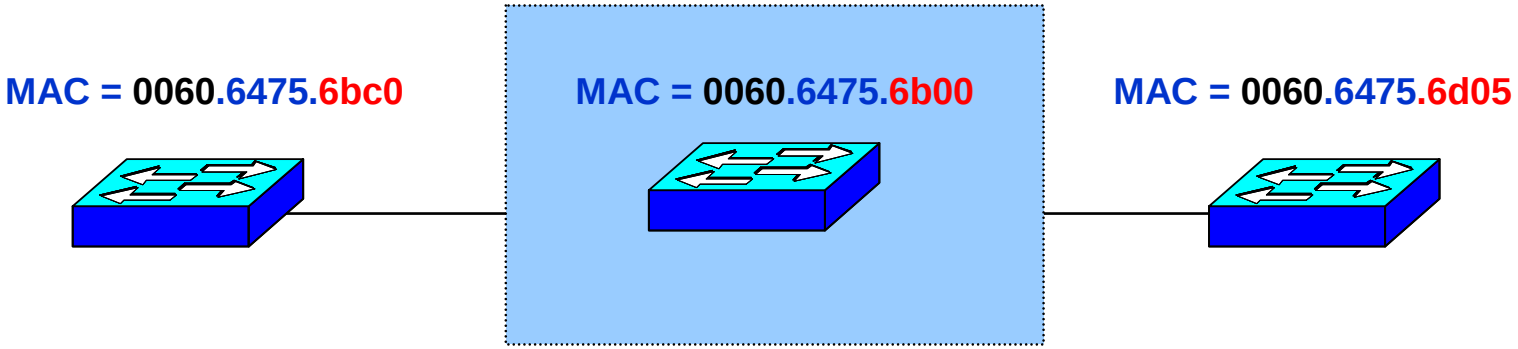
BPDU = bridge protocol data unit

- Les opérations de STP sont basées sur un certain nombre de paramètres :

1. Identificateur du Switch (Bridge identifier :BID)

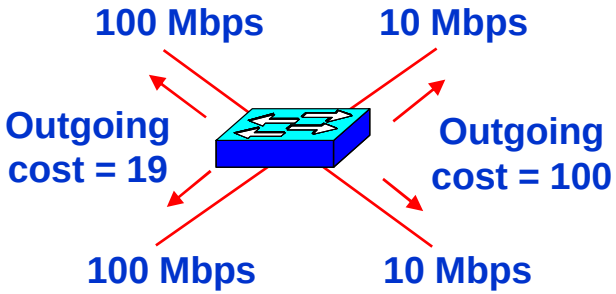
- **Composé de deux champs concaténés pour former un nombre de 64 bits**
  - ***Champs de Priorité sur 16 bits***
  - ***L'adresse MAC du switch (sur 48 bits attribué par le revendeur)***

  - **La Priorité est placée dans la partie (poids le plus fort)**
- **Valeur par défaut de la priorité est :**
  - ***Recommandation du standard : 32768 (0x8000)***
- **L'adresse MAC est utilisé pour départager les switchs en cas d'égalité des priorités**



2. **Un identificateur de port par switch :Sur 16 bits**

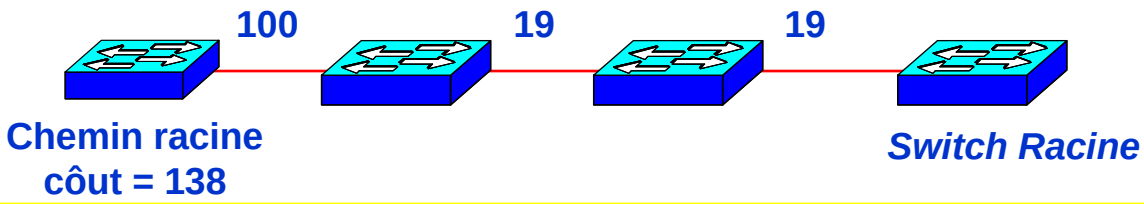
- Composé de deux parties :
  - Adresse (numéro de port sur 8 bits), fixé par le revendeur
  - Précédé par un champs de priorité sur 8 bits) (fixé par l'utilisateur)
    - Valeur par défaut : 0x80



⊙ **Coût des ports :**

3. **Chaque port d'un switch a un coût de sortie**

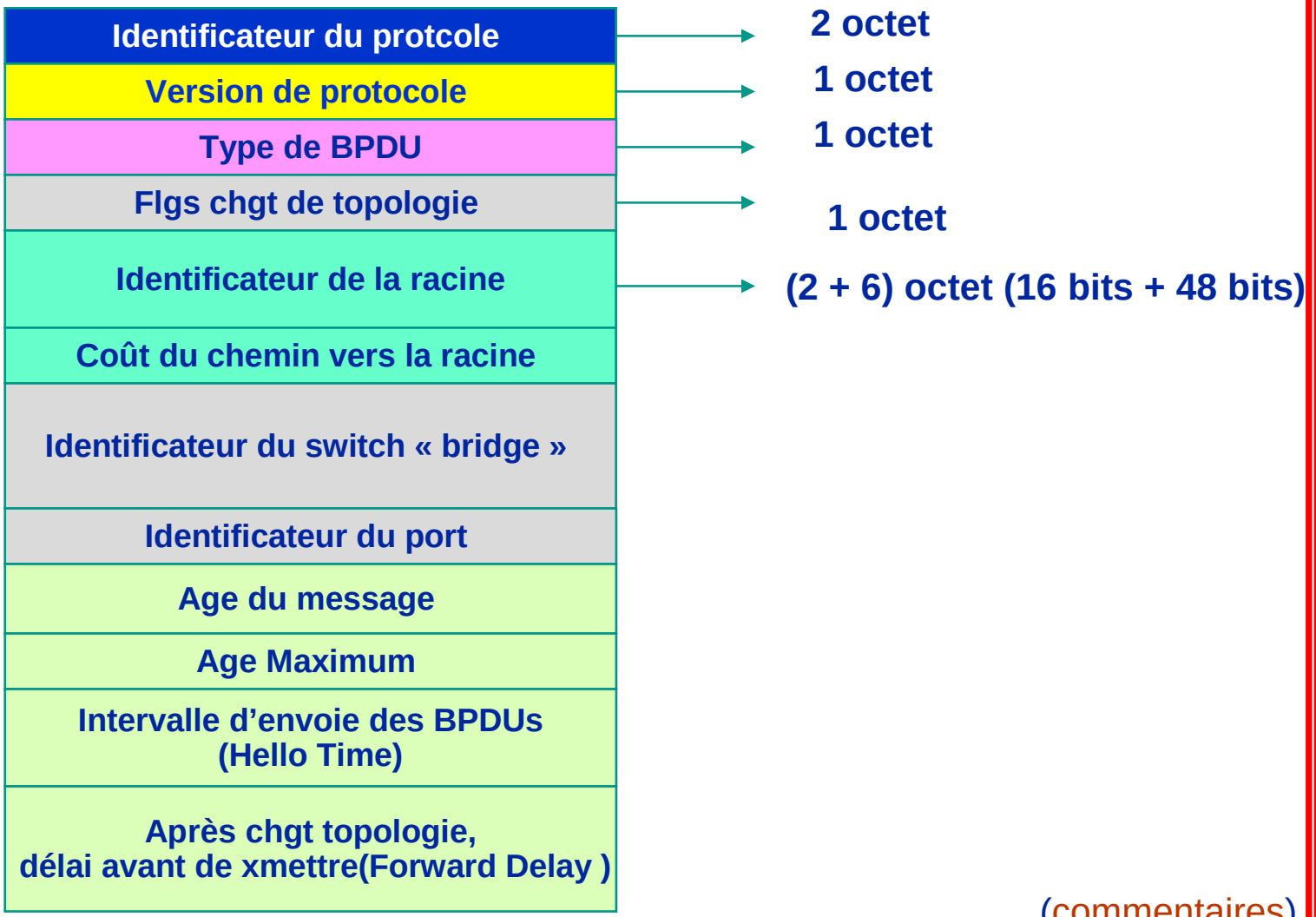
- Cas classique de calcul : 10^9/bande passante
- Les coûts des ports peuvent être fixés manuellement
- Le coût d'un chemin racine est la somme de tous les coûts des chemins vers la racine en partant de l'actuel switch
  - Le coût des ports racines est 0



Bandwidth	*Cost*
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

Normalisation IEEE

5. Format des messages BPDUs (Bridge Protocol Data Units (BPDUs)



Les BPDUs sont envoyés dans des trames Ethernets multicast ***: 01:80:C2:00:00:10***

(commentaires)

FIN DU CHAPITRE

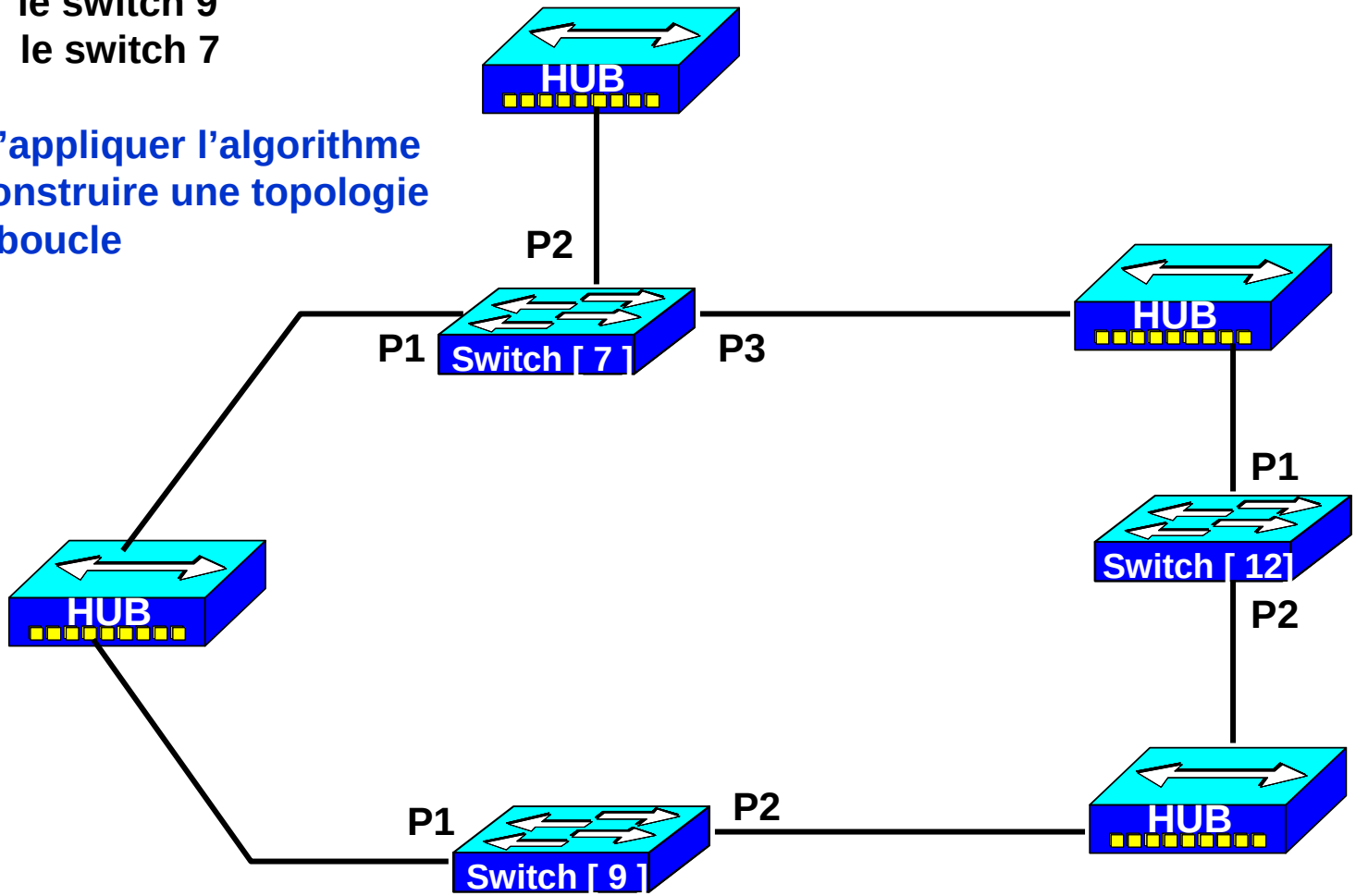
FIN DU CHAPITRE : Réveillez vous

Voir aussi :

http://www.cisco.com/warp/public/473/spanning_tree1.swf

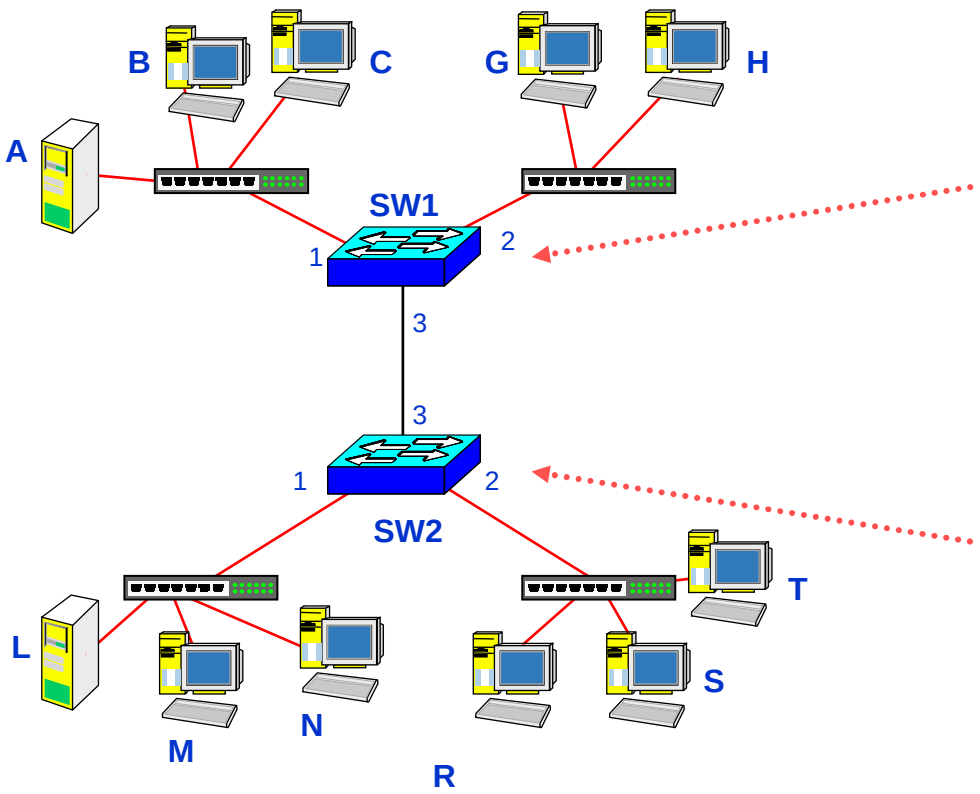
- **Au départ, on considère que les switchs ne sont pas allumés**
  - **On allume le switch 12**
  - **Ensuite, le switch 9**
  - **Et en fin, le switch 7**

- **On demande d'appliquer l'algorithme de STP pour construire une topologie stable et sans boucle**



Solution

SOLUTION DES TD

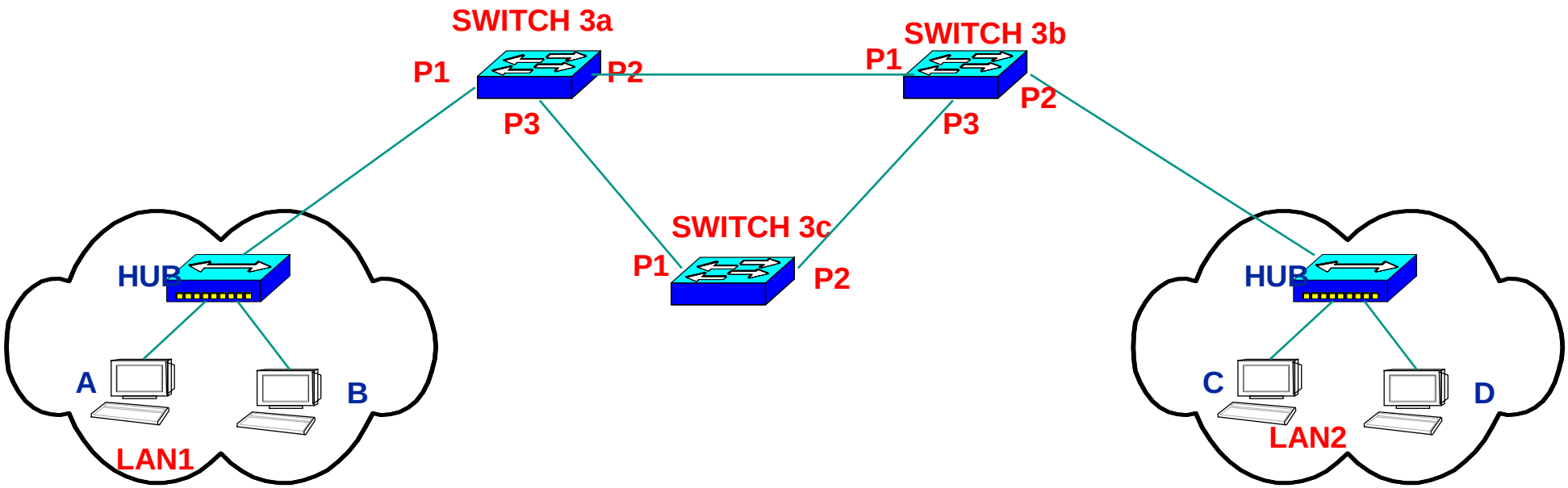


Port 1	Port 2	Port 3
A	G	L
B	H	M
C		N
		R
		S
		T

Port 1	Port 2	Port 3
L	R	A
M	S	B
N	T	C
		G
		H

back

- **Algorithme de Spanning Tree : Effet de boucle**
  - **A (sur LAN1) veut envoyer une trame à D (LAN2)**
- **Au départ les tables de routage sont vides dans les switchs**

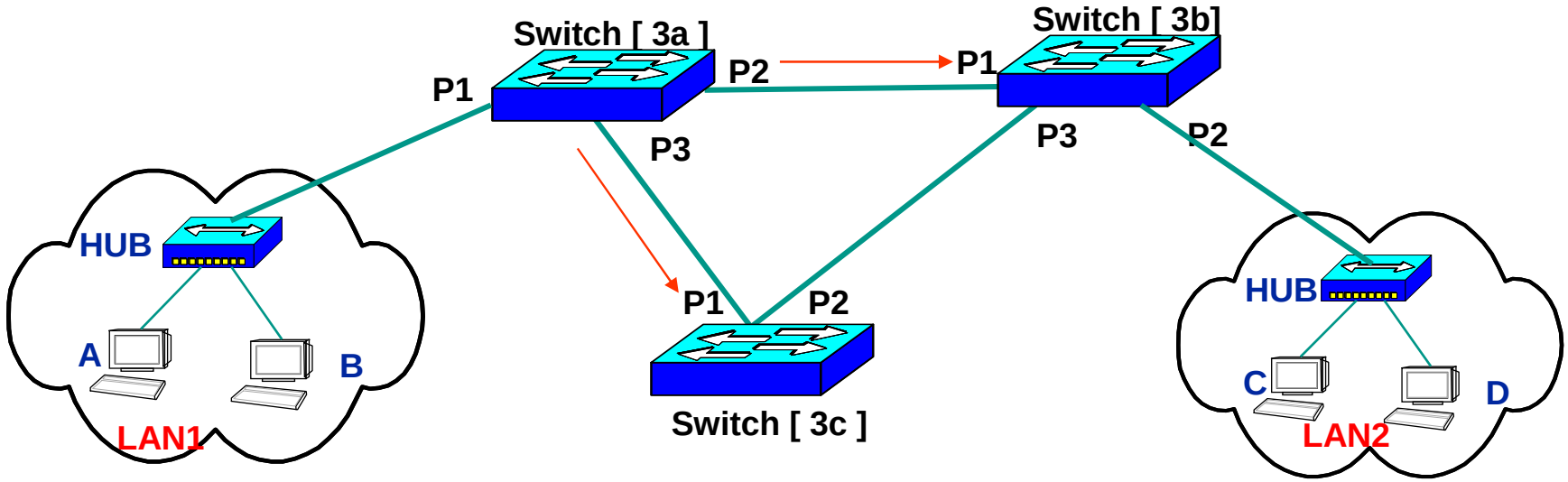


Algorithme de Spanning Tree : Effet de boucle

Le switch 3a reçoit la trame de A sur son port P1

=> le switch 3a identifie la station A mais pas la station D

Port	Station
P1	A

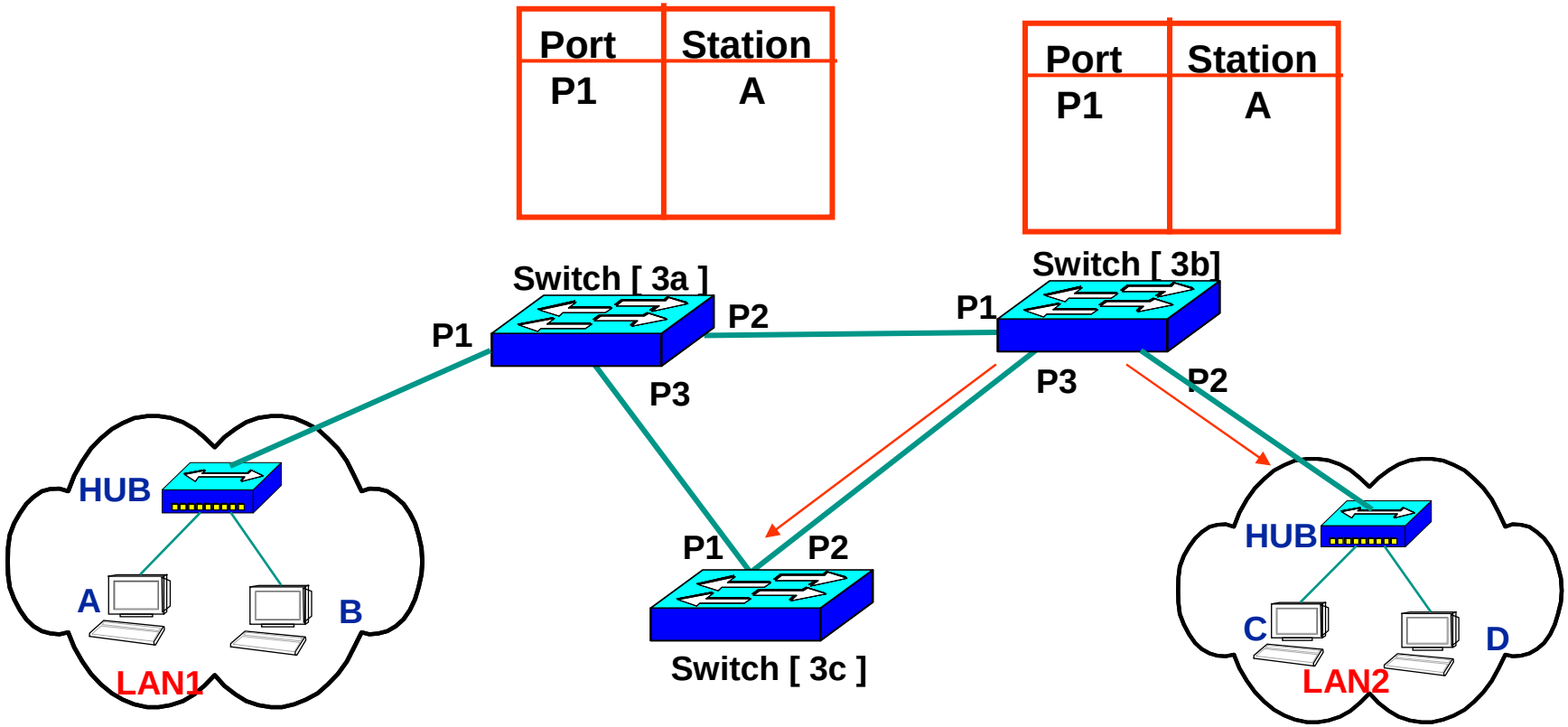


le switch 3a envoie la trame sur ses ports P2 et P3

Algorithme de Spanning Tree : Effet de boucle

Le switch 3b reçoit la trame de A sur son port P1

=> le switch 3b identifie la station A mais pas la station D

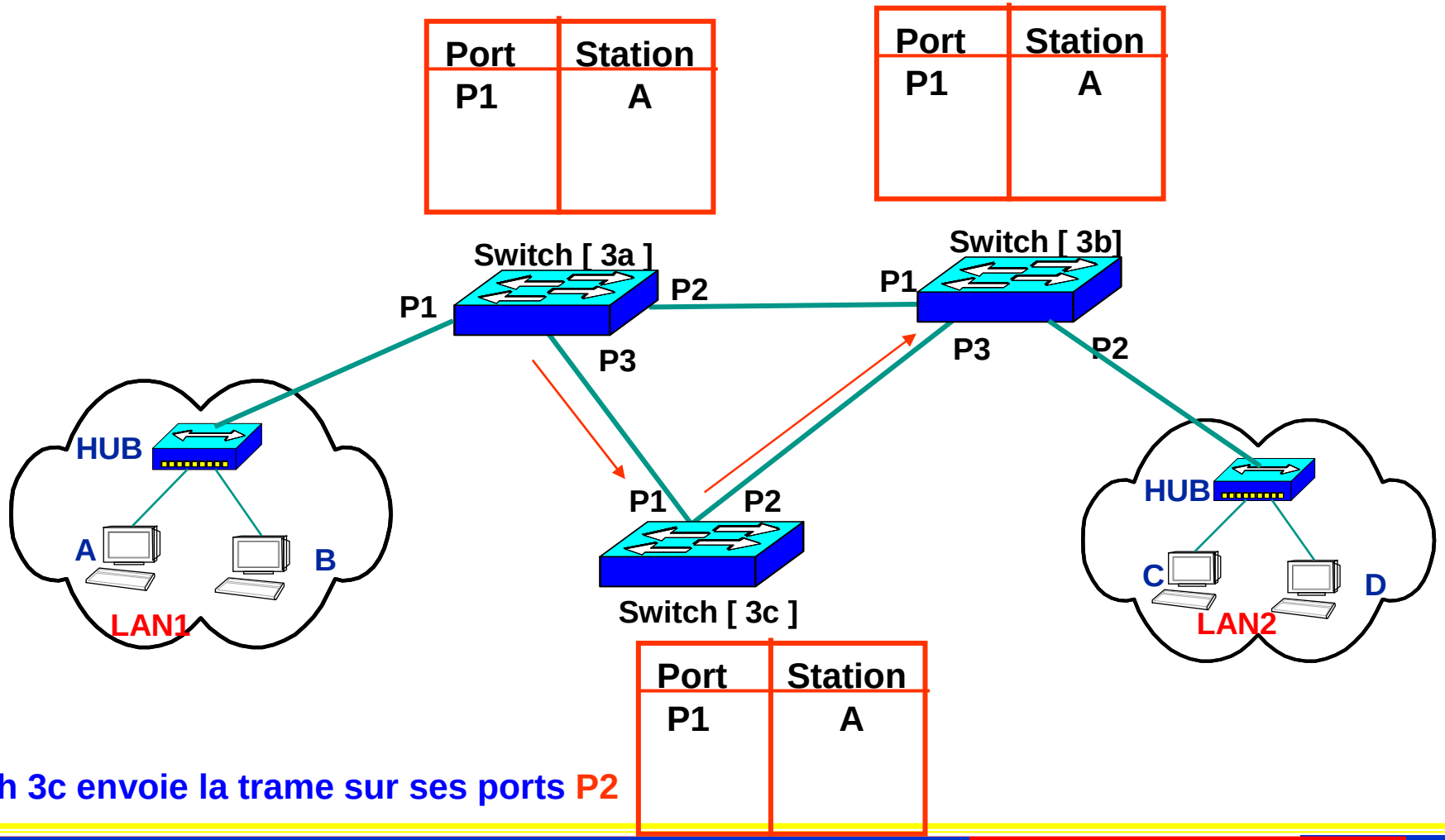


le switch 3b envoie la trame sur ses ports P2 et P3

Algorithme de Spanning Tree : Effet de boucle

Le switch 3c reçoit la trame de A sur sont port P1

=> le switch 3c identifie la station A mais pas la station D



le switch 3c envoie la trame sur ses ports P2

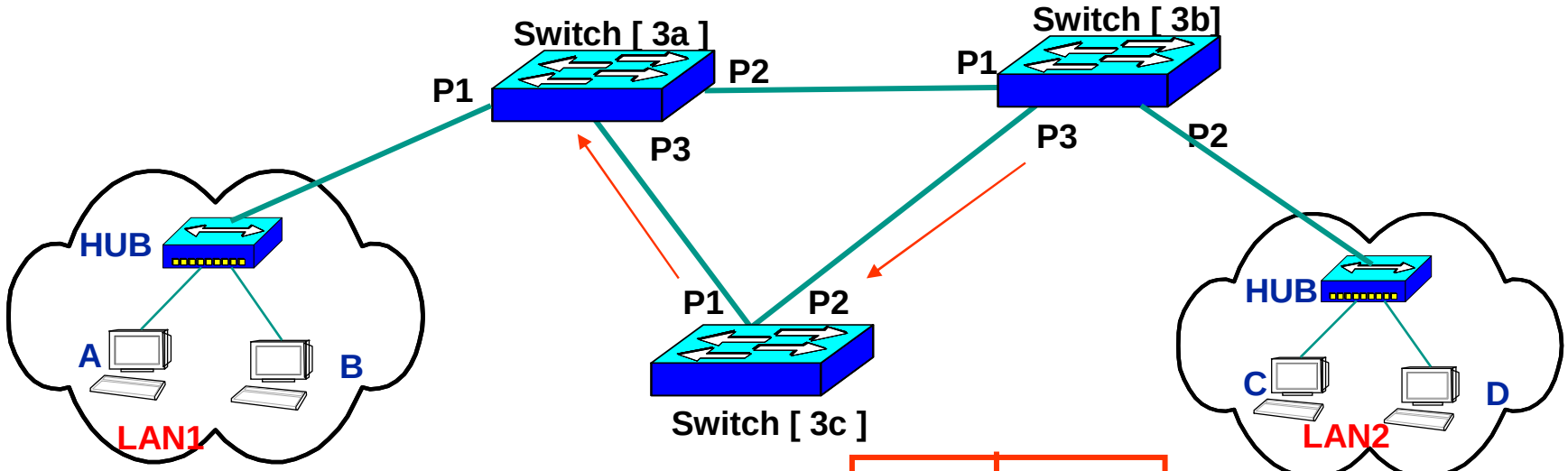
Algorithme de Spanning Tree : Effet de boucle

Le switch 3c reçoit la trame de A sur sont port P2 (via le switch 3b)

=> le switch 3c identifie la station A mais pas la station D

Port	Station
P1	A

Port	Station
P1	A



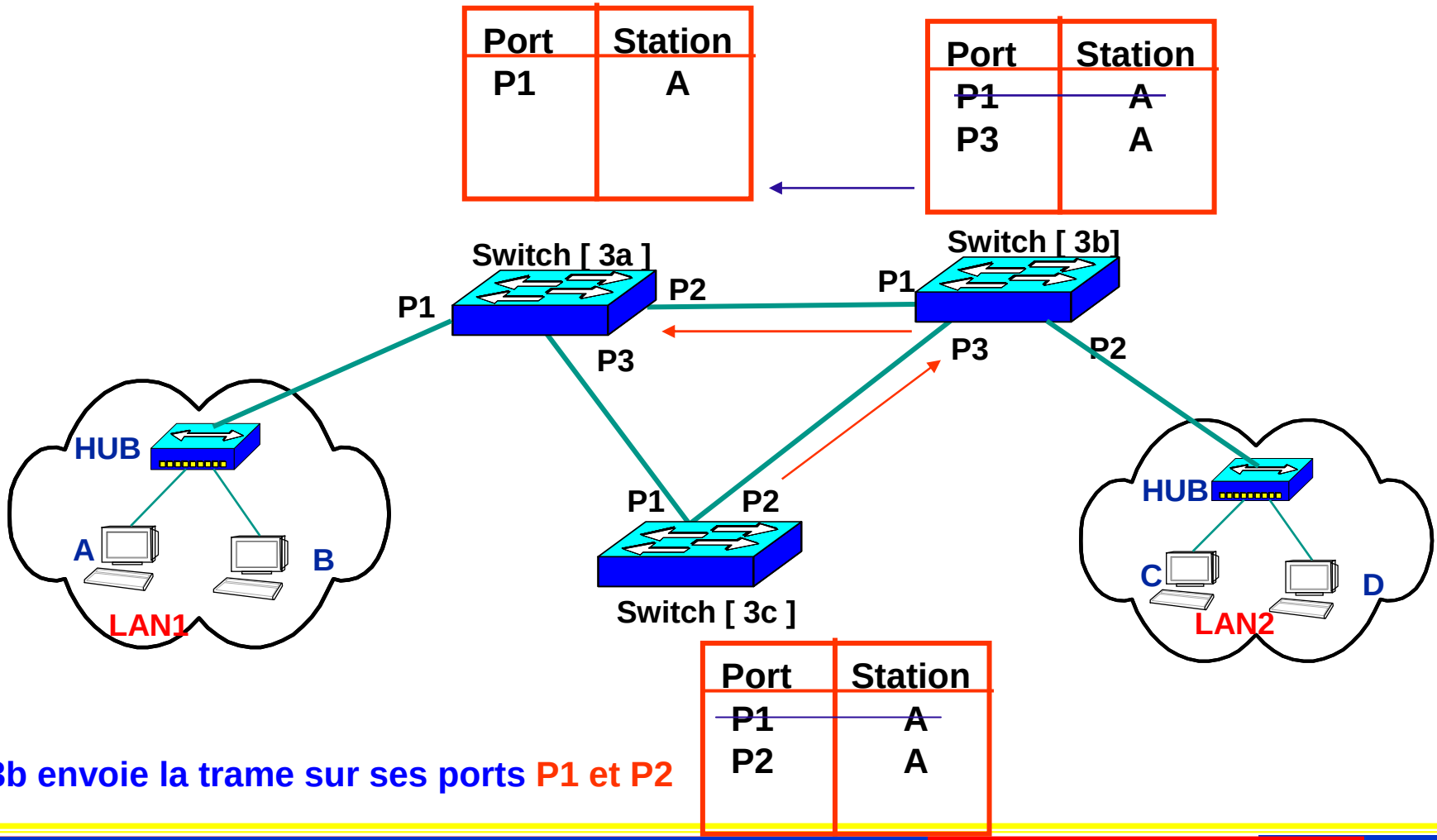
Port	Station
~~P1~~	~~A~~
P2	A

le switch 3c envoie la trame sur son port P1

Algorithme de Spanning Tree : Effet de boucle

Le switch 3b reçoit la trame du switch 3c sur sont port P3

=> le switch 3b identifie la station A mais pas la station D

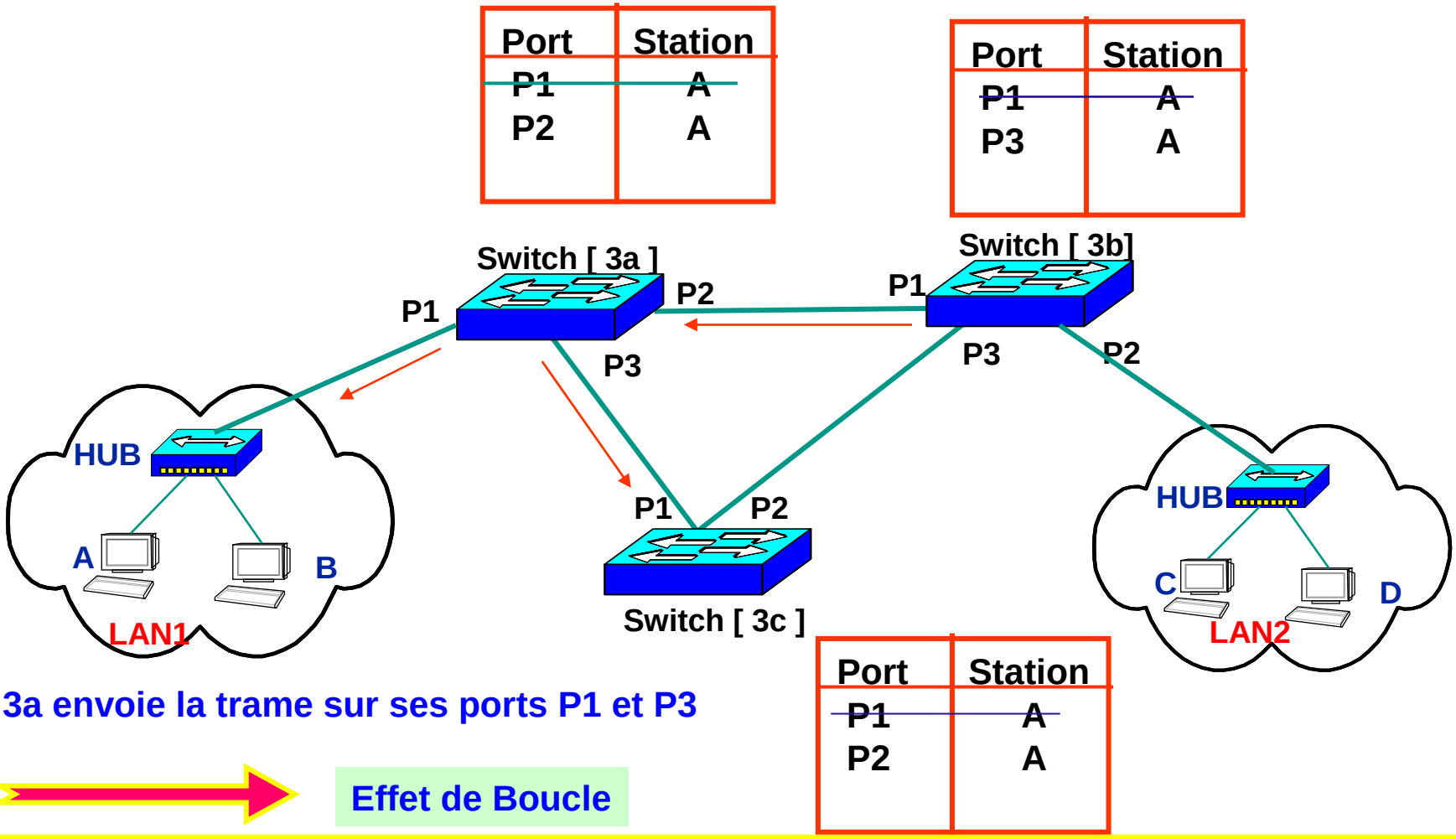


le switch 3b envoie la trame sur ses ports P1 et P2

Algorithme de Spanning Tree : Effet de boucle

Le switch 3a reçoit la trame du switch 3b sur sont port P2

=> le switch 3a identifie la station A mais pas la station D



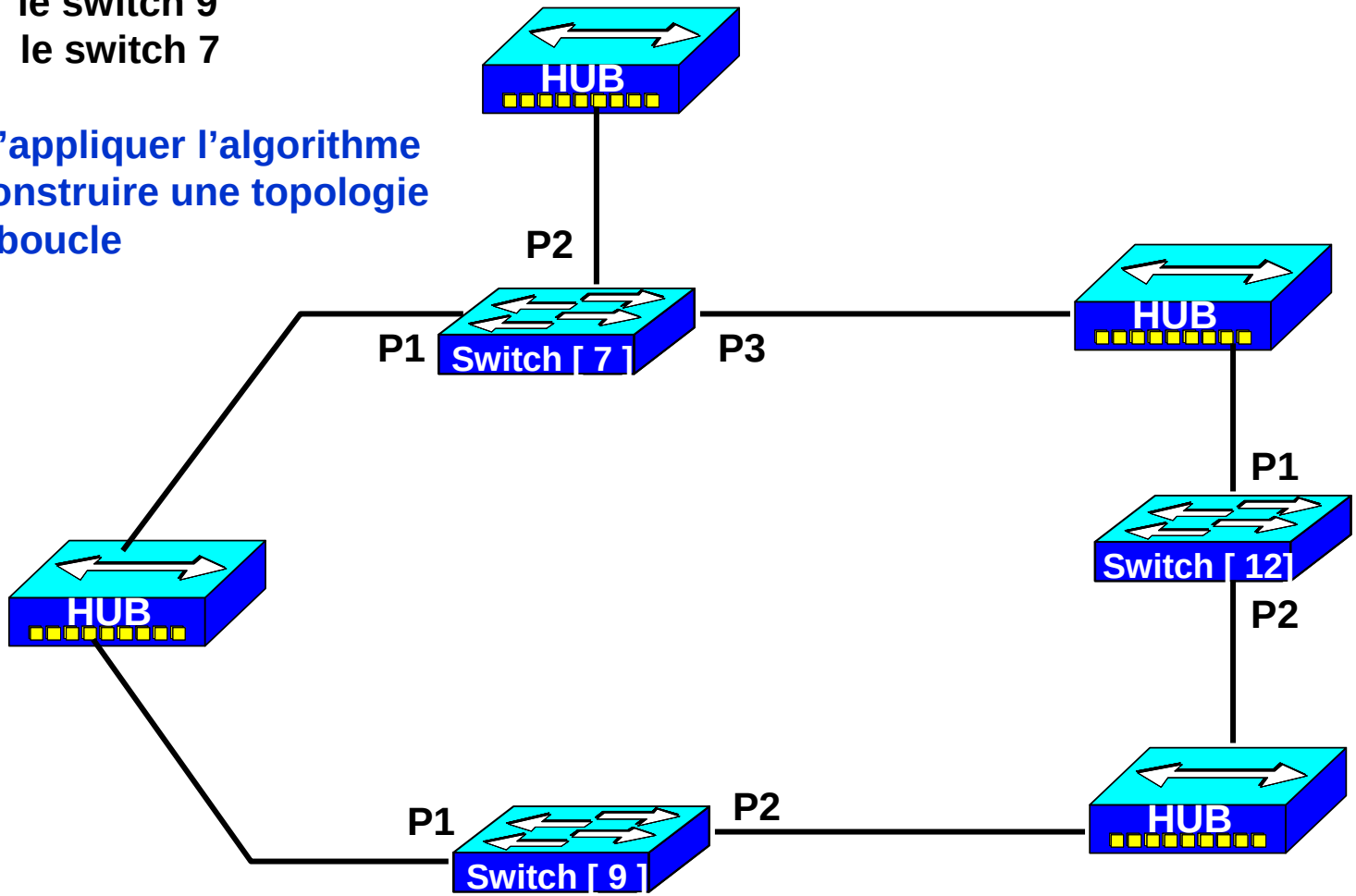
le switch 3a envoie la trame sur ses ports P1 et P3

Back

Effet de Boucle

- **Au départ, on considère que les switchs ne sont pas allumés**
  - **On allume le switch 12**
  - **Ensuite, le switch 9**
  - **Et en fin, le switch 7**

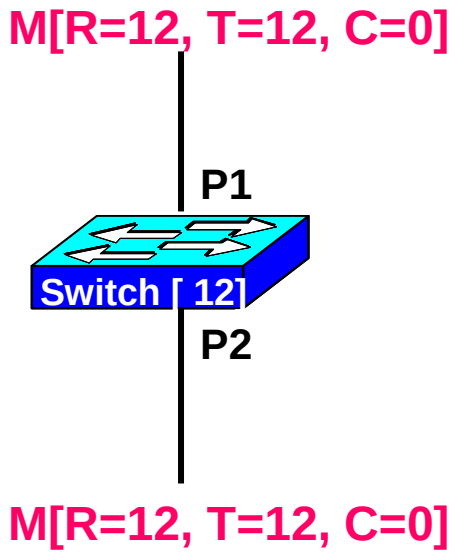
- **On demande d'appliquer l'algorithme de STP pour construire une topologie stable et sans boucle**



1. Supposons que le switch 12 est le premier à être allumé

- **Le switch [12] se considère comme racine de l'arbre (car il n'a reçu aucun message 802.1). Il va émettre un message 802.1 qui indique**
  - **que la racine est le switch 12,**
  - **le coût / à la racine est de 0,**
  - **l'identificateur de l'émetteur = 12**

⇒ **Ce message est émis sur ses ports P1 et P2**



Comme il n'y a aucun switch qui écoute à cet instant aux ports 1 et 2, les messages sont perdus

- **Etat du switch [12] :**

  - P1 : Best : - Port désigné

  - P2 : Best : - Port désigné

  - Message du switch est M[R = 12, T= 12, C= 0]

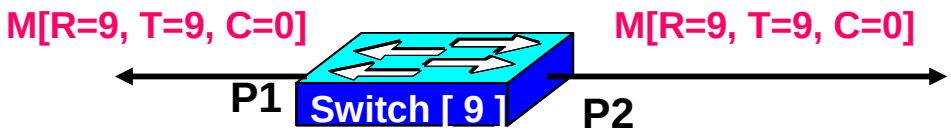
2. Supposons que maintenant, le switch 9 devient actif

- **Le switch [9] se considère comme racine de l'arbre (car il n'a reçu aucun message 802.1). Il va émettre un message 802.1 qui indique**
  - **que la racine est le switch 9,**
  - **le coût / à la racine est de 0,**
  - **l'identificateur de l'émetteur = 9**

M[R = 9, T = 9, C = 0]

⇒ Ce message est émis sur ses ports P1 et P2

⇔ le message M[R = 9, T = 9, C = 0] est envoyé sur les ports P1 et P2

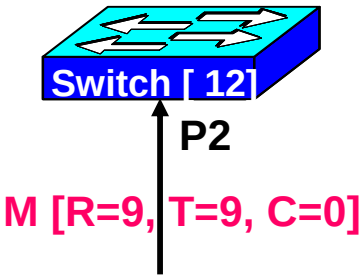


Etat du switch [9]

- **Se considère comme étant la racine**
- **P1 : Best = - Port désigné**
- **P2 : Best = - Port désigné**
- **Message du switch est M[R=9, T=9, C=0]**

Le message envoyé sur le port P2 va être reçu par le switch 12

- **Réception du message du switch 9 par le switch 12 :**



A. Etat du switch [12] avant réception du Message

Switch 12 : se considérait comme étant la racine

P1 : Best = - Port Etat désigné

P2 : Best = - Port Eta désigné

Le Message 802.1D du switch était : **M[R = 12, T = 12, C = 0]**

B. Etat du switch [12] après réception M [R=9, T=9, C=0] sur P2

- Sur P2, le meilleur message reçu va être maintenant M [R=9, T=9, C=0]
- Le switch 12 va analyser le meilleur message reçu sur les différents ports
- conclusion : nouvelle racine de Spanning Tree est : switch 9
- Le port 2 du switch va passer dans *l'état Racine*.
- Le port 1 reste dans *l'état désigné* (on a reçu aucun message 802.1 sur ce port)
- Puisque P2 devient racine, le message 802.1 du switch 12 va changer

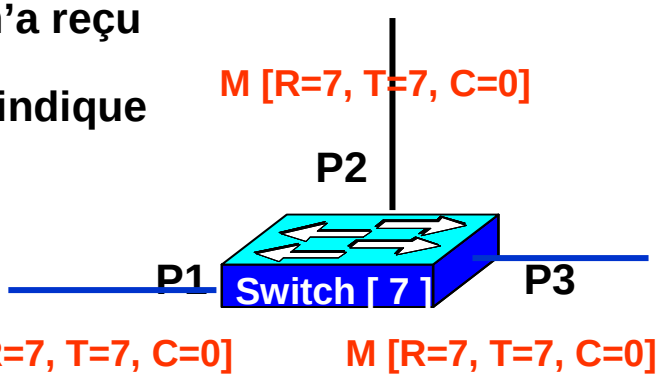
Switch 12 :

- **P1 : Port désigné /on a reçu aucun msg/**
- **P2 : [9, 9, 0] => devient port Racine**
- **Message du switch [12] devient :**
  - **M[R=9, T=12, C=1] à envoyer sur ses ports désignés (donc sur P1)**

3. Supposons que maintenant, le switch 7 devient actif

- **Le switch [7] se considère comme racine de l'arbre (car il n'a reçu aucun message 802.1). Il va émettre un message 802.1 qui indique**
  - **Que la racine est le switch 7,**
  - **Le coût / à la racine est de 0,**
  - **L'identificateur de l'émetteur =7**

⇒ **Ce message est émis sur ses ports P1, P2 et P3**



Etat du switch [7] :

- **Se considère comme étant la racine**
  - **P1 : Best = - Port désigné**
  - **P2 : Best = - Port désigné**
  - **P3 : Best = - port désigné**

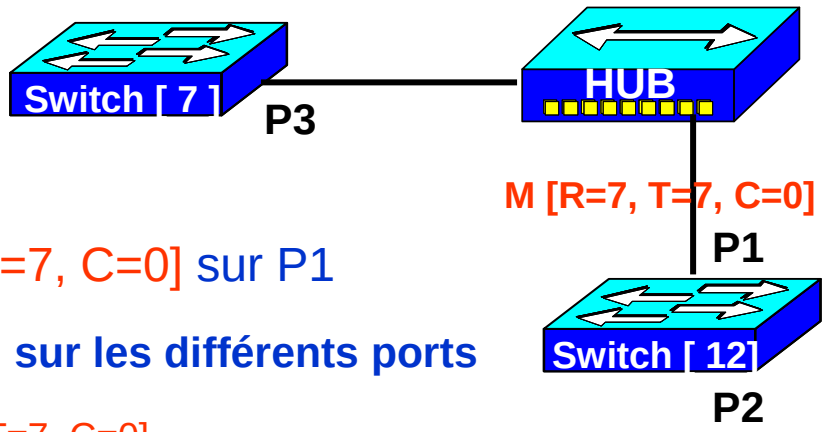
- Message du switch[7]est : M [R=7, T=7, C=0] sur P1, P2 et P3

Regardons ce qui se passe lorsque ce message est reçu par le switch 12

4 - Réception du Message du switch [7] par le switch [12]

A. Avant réception du message 802.1 du switch 7, le switch 12 était dans la situation

Switch 12 :
- **P1 : Port désigné**
- **P2 : [9, 9, 0] : port Racine**
- **Message du switch [12] M[R=9, T=12, C=1]**



B. Etat du switch [12] **après réception** M [R=7, T=7, C=0] sur P1

- **Le switch 12 va analyser le meilleur message reçu sur les différents ports**

  ☞ Le meilleur message reçu sur le port 1 : M [R=7, T=7, C=0]

- **Sur la base de ce message et son identificateur à lui, il va déterminer la racine (conclusion : nouvelle racine de spanning Tree est : switch 7)**
- **Le port 1 du switch va passer dans l'état Racine.**
- **=> le switch va devoir modifier Son message 802.1**

  ↳ Le message 802.1D du switch 12 va être : **M[R = 7, T = 12, C = 1]**

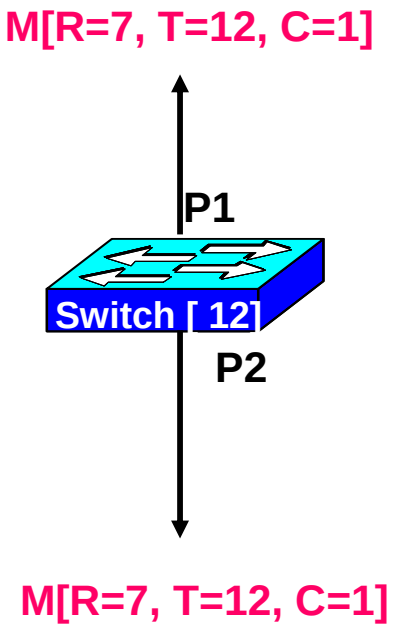
- **Connaissant le message du switch 12, on peut déterminer l'état du port 2 (par comparaison)**
- **Le port P2 devient désigné** ⟷ **M[R=7, T=12, C=1] est meilleur que [9, 9, 0]**

♦ **Nouveau message du switch 12 : M[R=7, T=12, C=1] envoyé sur ses ports désignés.**

→ **Donc sur le port P2**

Nouvel état du Switch 12 :

- **Messages reçus :**
  - **Sur P1 : M [R=7, T=7, C=0] : port racine**
  - **Sur P2 : [9, 9, 0] : port désigné**
- **Message du switch [12] est : M[R=7, T=12, C=1]**

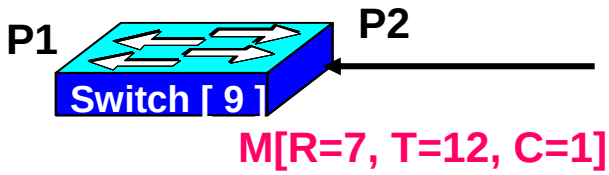


5 - Réception du Message du switch [12] par le switch [9]

A. Avant réception du message, le switch 9 était dans l'état suivant

Etat du switch [9]

- **Se considère comme étant la racine**
- **P1 : Best = - Port désigné**
- **P2 : Best = - Port désigné**
- **Message 802.1du switch était M[R=9, T=9, C=0]**



B. Après réception du message M[R=7, T=12, C=1] du switch 12

- **Le switch 9 va analyser le meilleur message reçu sur les différents ports**
  - **Le meilleur message reçu sur le port 2 : M[R=7, T=12, C=1]**
- **Sur la base de ce message et son identificateur à lui, il va déterminer la racine (conclusion : nouvelle racine de spanning Tree est : switch 7)**
  - ☞ Le port 2 du switch va passer dans l'état Racine.
- **=> le switch va devoir modifier Son message 802.1**
  - ↳ Va être : M[R = 7, T = 9, C = 2]
  - ↳ Le port P1 reste dans *l'état désigné*

C.Le switch 9 va émettre son nouveau message 802.1 sur ses ports désignés, donc sur P1

M[R = 7, T = 9, C = 2] **M[R = 7, T = 9, C = 2]**

Nouvel Etat du switch [9]

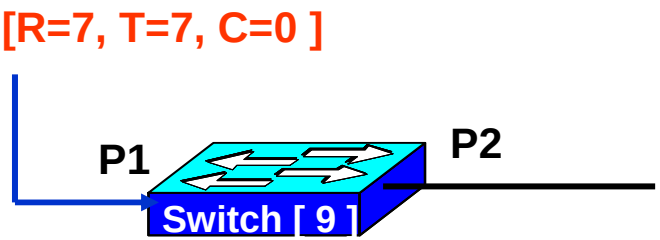
- **P1 : Best = - Etat désigné**
- **P2 : Best = - Etat racine / M[R=7, T=12, C=1]**
- **Message 802.1du switch 9 : M[R = 7, T = 9, C = 2]**

6. Réception du Message du switch [7] par le switch [9]

A. Avant réception du message venant du switch 7, le switch 9 était dans l'état suivant

Etat du switch [9]

-
- P1 : Best = - Etat désigné
- P2 : Best = - Etat racine / M[R=7, T=12, C=1]
- Message 802.1du switch 9 : M[R = 7, T = 9, C = 2]



B. Après réception du message M [R=7, T=7, C=0] sur son port 1

- Le switch 9 va analyser le meilleur message reçu sur les différents ports
  - Le meilleur message reçu sur le port 1 est : M [R=7, T=7, C=0]
  - Le meilleur message reçu sur le port 2 est : M[R=7, T=12, C=1]
- Sur la base de ce message et son identificateur à lui, il va déterminer la racine (conclusion : nouvelle racine de spanning Tree est : switch 7)
- Port racine et ports désignés ?

☞ Le message reçu sur le port 1 est meilleur que celui reçu sur le port 2 . Donc, le port 1 devient le port racine

- **=> on peut déterminer le message 802.1 du switch 9**

  Son message 802.1 va être maintenant : M[R = 7, T = 9, C = 1]

- **Connaissant le message du switch 9, on peut déterminer l'état du port 2:**

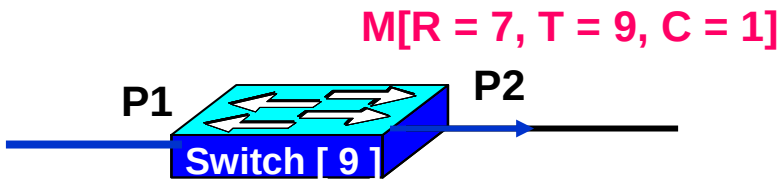
  → **Comparer**

  le Message du switch 9 **M[R = 7, T = 9, C = 1]**

  avec celui reçu sur le port 2 **M[R = 7, T=12, C = 1]**

  Le message du switch 9 est meilleur que le meilleur message reçu sur le pot 2

  => P2 passe dans l'état désigné



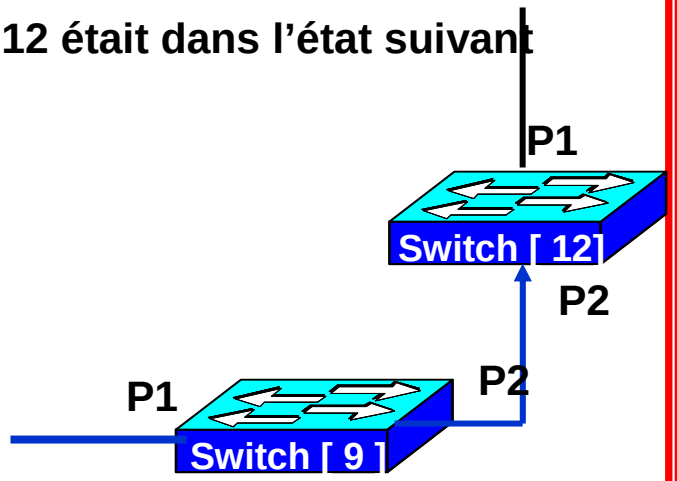
Le message du switch 9 **M[R = 7, T = 9, C = 1]** est envoyé sur les ports désignés (donc sur Port2)

7 - Réception du Message du switch [9] par le switch [12]

A. **Avant réception du message venant du switch 9, le switch 12 était dans l'état suivant**

Switch 12 :

- **P1 : M [R=7, T=7, C=0] : port racine**
- **P2 : [9, 9, 0] : port désigné**
- **Message du switch [12] est : M[R=7, T=12, C=1]**



B. **Après réception du message M[R=7, T=9, C=1] sur son port 2**

M[R = 7, T = 9, C = 1]

1. **Le switch 12 va analyser le meilleur message reçu sur les différents ports**
2. **Racine ? Identification du switch le plus petit est 7 (switch 7)**
3. **Meilleur message reçu ?**

   → **Est celui qui est arrivé sur le port 1**

   - **Le port 1 va se trouver dans l'état racine (Port1 : port racine)**

4. **Détermination du message 802.1D du switch 12 :**

   M[R=7, T=12, C=1]

C. Etat du port 2 :

- **Connaissant le message du switch12, on peut déterminer l'état du port 2:**
- **Comparer**

le nouveau Message du switch 12 [R = 7, T = 12, C = 1]

avec celui reçu sur le port 2 [R = 7, T = 9, C = 1]

Le message du port 2 est meilleur qui celui du switch 12

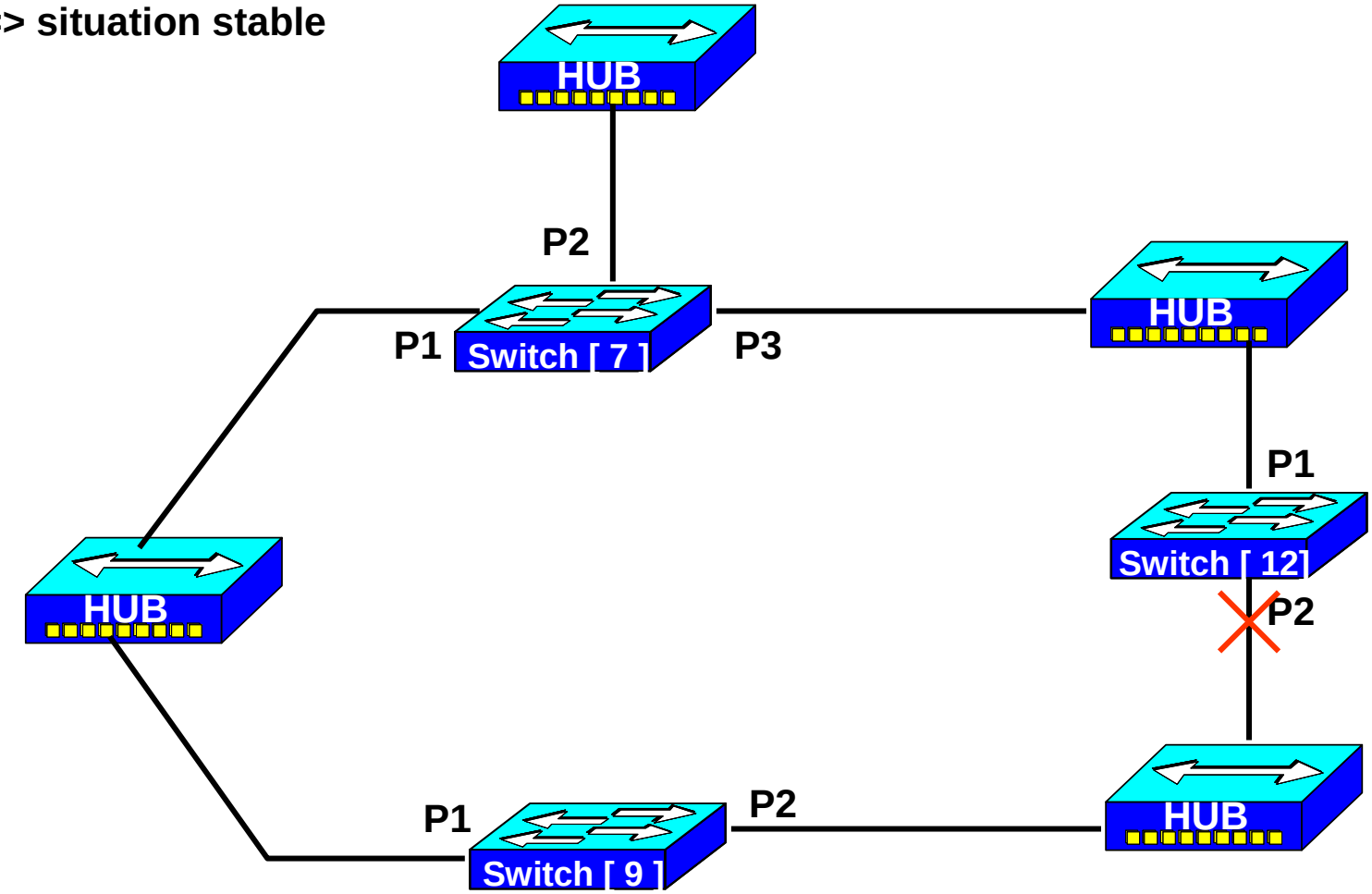
=> le port 2 passe dans l'état bloqué

8. à ce stade le switch 7 va envoyer son message 802.1D

il provoque aucun changement sur la situation de switch 9

Le message 802.1 du swicth 9 ne provoque aucun changement sur switch 12

=> situation stable



Back

Annexe
STP sur les switchs Cisco

Les états ont été définis au départ, puis ils ont été modifiés par le protocole STP.

- **Blocage**
- **Écoute**
- **Apprentissage**
- **Transmission**
- **Désactivé**

Il est possible de configurer les ports du serveur de sorte qu'ils passent automatiquement en mode de transmission STP

Timer	Primary Purpose	Default
Hello Time	The amount of time between the sending of Configuration BPDUs by the Root Bridge	2 seconds
Forward Delay	The duration of the Listening and Learning states	15 seconds
Max Age	How long a BPDU is stored	20 seconds

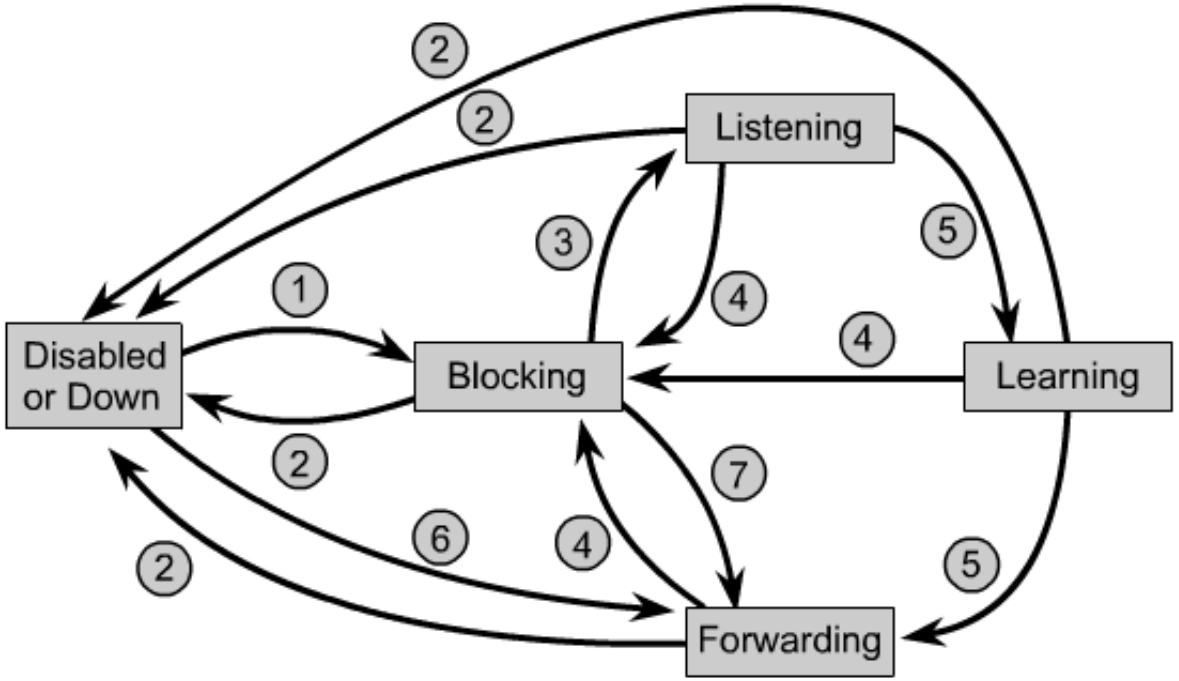
- Blocage - Aucune trame transmise, unités BPDU entendues.
- Écoute - Aucune trame transmise, écoute de trames.
- Apprentissage (learning) - Aucune trame transmise, acquisition des informations sur les adresses.
- Transmission (forwarding)- Trames transmises, acquisition des informations sur les adresses.
- Désactivé - Aucune trame transmise, aucune unité BPDU entendue.

Transmission
Apprentissage
Écoute
Blocage
Désactivé

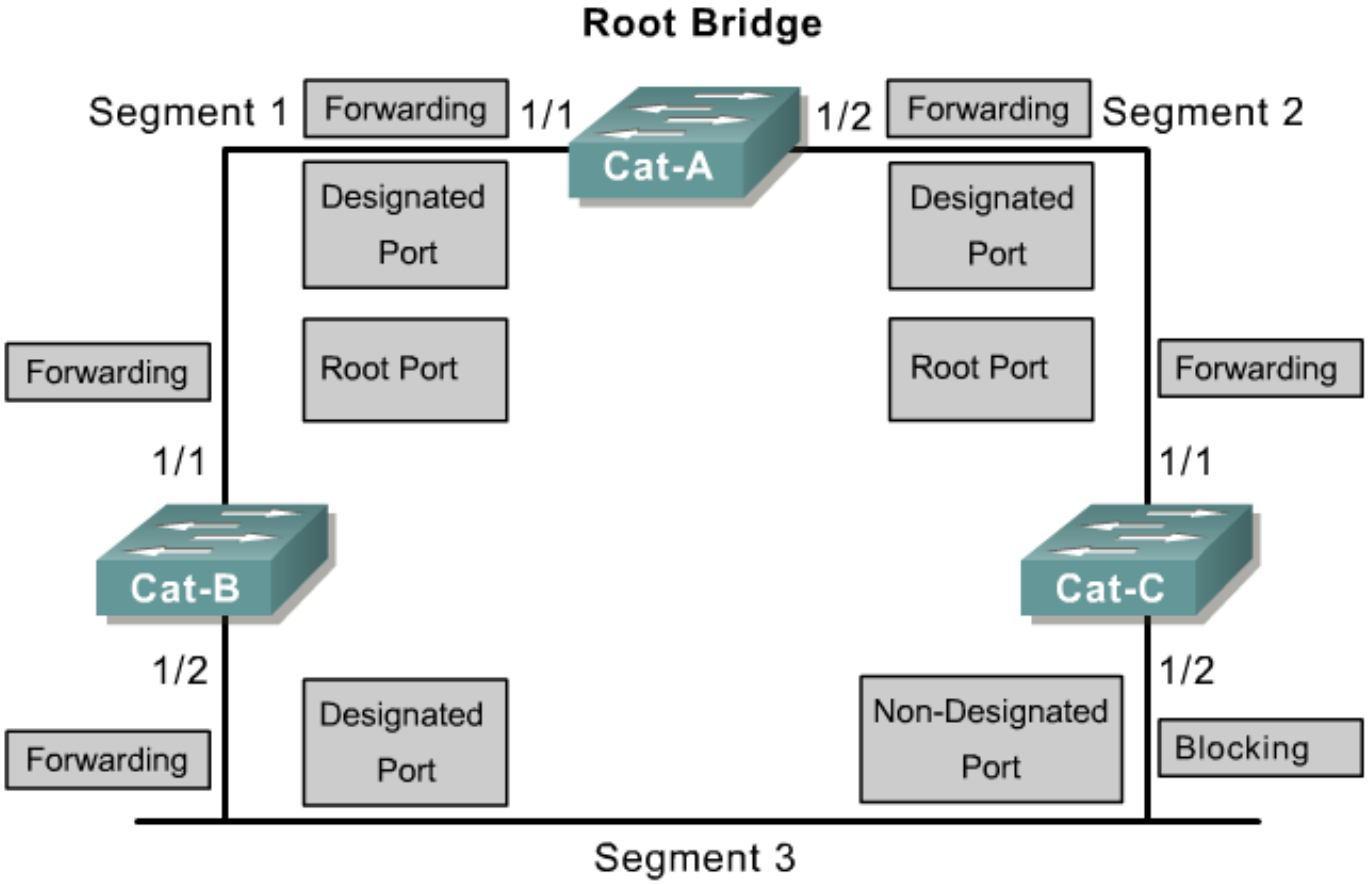
State	Purpose
Forwarding	Sending / receiving user data
Learning	Building bridging table
Listening	Building "active" topology
Blocking	Receives BPDUs only
Disabled	Administratively down

- Il fait partie de la norme **802.1d**.
- Le principe est simple : construire une arborescence sans boucle à partir d'un point identifié, connu sous le nom de racine.
- Les chemins redondants sont autorisés, mais un seul peut être le chemin actif.
- Cet algorithme a été développé par Radia Perlman.

Etats du STP



Standard States	Events
1. Port enabled or initalized.	6. PortFast
2. Port disabled or fails.	7. UplinkFast
3. Port selected as a Root or Designated Port.	
4. Port ceases to be a Root or Designated Port.	
5. Forwarding timer expires.	


Root Bridge
Segment 1
Forwarding
1/1
Cat-A
1/2
Forwarding
Segment 2
Designated Port
Designated Port
Root Port
Root Port
Forwarding
Forwarding
1/1
1/1
Cat-B
Cat-C
1/2
1/2
Forwarding
Designated Port
Non-Designated Port
Blocking
Segment 3

Étape 1: Désignation d'un pont racine

Étape 2 : Désignation des ports racine

Étape 3 : Choix des ports désignés

- Tous les commutateurs envoient des unités BPDU (Bridge Protocol Data Units) de configuration.
- Les unités BPDU sont envoyées vers toutes les interfaces toutes les deux secondes (valeur par défaut - paramétrable) (*intervalle Hello*).
- Tous les ports se trouvent en mode Blocage durant le processus " Spanning Tree " initial.

Identificateur de protocole (2 octets)
Version (1 octet)
Type de message (1 octet)
Indicateurs (1 octet)
ID de la racine (8 octets)
Coût jusqu'à la racine (4 octets)
ID de pont (8 octets)
ID du port (2 octets)
Âge du message (2 octets)
Âge maximum (2 octets)
Heure HELLO (2 octets)
Délai de transmission (2 octets)

Algorithme " spanning tree " (STA) : Champs BPDU (à titre d'information)

- Les champs utilisés dans l'unité BPDU STA sont donnés à titre purement informatif.
- Lorsqu'il sera question de l'algorithme STA, vous voudrez peut-être vous reporter à ce protocole pour savoir comment se déroulent l'envoi et la réception des informations.

Champs BPDU (à titre d'information)

- Identificateur de protocole (2 octets),
- Version (1 octet),
- Type de message (1 octet) : Pas vraiment utilisés
- Indicateurs (1 octet) : Ce champ est utilisé avec les modifications de topologie
  ID de la racine (8 octets) : Ce champ indique le pont racine actuel sur le réseau. Il comprend :

  → *La priorité du pont* (2 octets) [1 - 65 526]

  → *L'adresse MAC du pont* (6 octets)

  - Il est connu sous le nom d'identificateur du pont racine.

Champs BPDU (à titre d'information)

- Coût jusqu'à la racine (4 octets) : C'est le coût de la route depuis le pont qui émet l'unité BPDU jusqu'au pont racine indiqué dans le champ ID de la racine.
  Le coût est basé sur la bande passante.
- ID de pont (8 octets) : C'est le pont qui émet l'unité BDPU.
  - 2 octets : Priorité du pont
  - 6 octets : Adresse MAC
- ID du port (2 octets) : Il s'agit du port situé sur le pont qui émet l'unité BDPU, y compris la valeur Priorité du port.

Champs BPDU (à titre d'information)

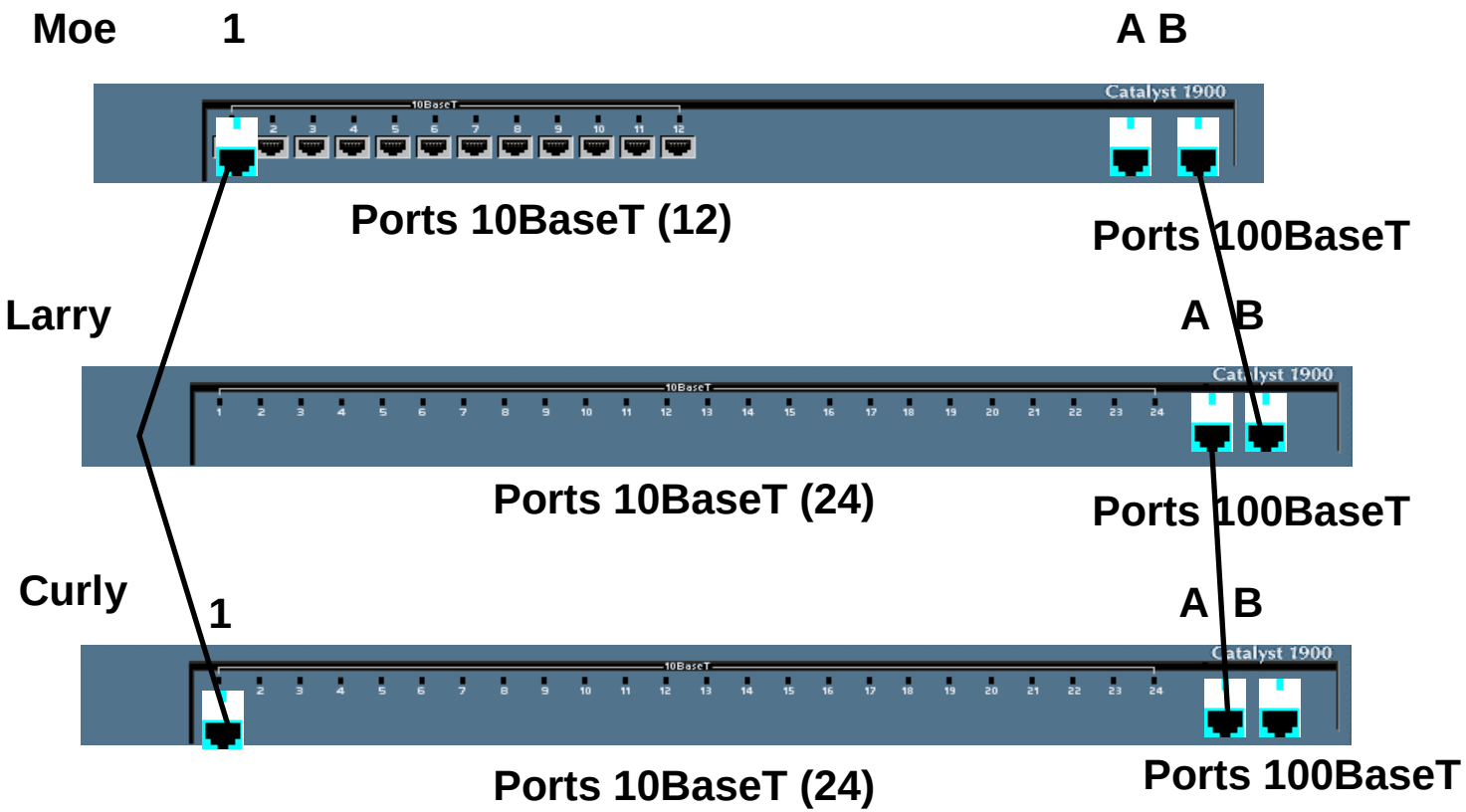
- Âge du message (2 octets) : Il s'agit de l'âge de l'unité BPDU
- Âge maximum (2 octets) : Ce champ indique quand l'unité BPDU doit être éliminée
- Heure HELLO (2 octets) : Ce champ indique la fréquence d'envoi des unités BPDU
- Délai de transmission (2 octets) : Ce champ indique combien de temps le pont doit rester en état « Écoute et Apprentissage »

Timer	Primary Purpose	Default
Hello Time	The amount of time between the sending of Configuration BPDUs by the Root Bridge	2 seconds
Forward Delay	The duration of the Listening and Learning states	15 seconds
Max Age	How long a BPDU is stored	20 seconds

(retour)

Il y a 3 commutateurs avec des chemins redondants.

Êtes-vous capable de les identifier ?



Étape 1: Désignation d'un pont racine

- Priorité du pont
- ID de pont
- Pont racine

Étape 2 : Désignation des ports racine

- Coût de la route ou Coût du port
- Coût du chemin racine
- Port racine

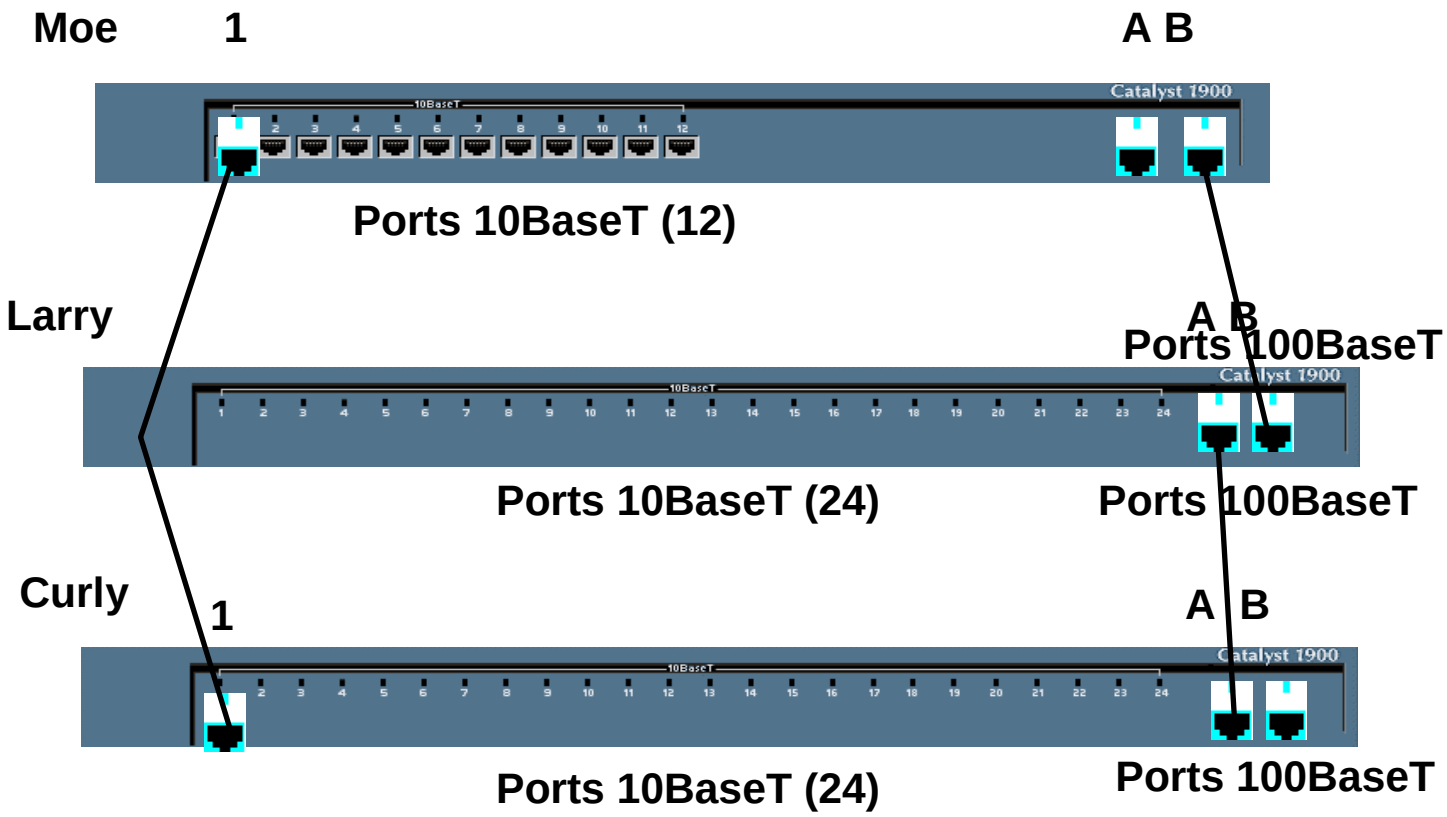
Étape 3 : Choix des ports désignés

- Coût de la route ou Coût du port
- Coût du chemin racine

- La première étape pour les commutateurs consiste à sélectionner un pont racine.
  - Le **pont racine** est le pont à partir duquel tous les *autres chemins sont décidés.*
  - Un seul commutateur peut prétendre au titre de pont racine.
  - Le choix d'un pont est déterminé par :
    1. La priorité de pont la *plus faible*
    2. L'ID de pont *le plus petit* (tiebreaker ou pont de départage)

Priorité du pont

- Il s'agit d'une valeur numérique.
- Le commutateur qui présente la priorité de pont la plus faible est le pont racine..
- Pour ce faire, les commutateurs utilisent des unités BPDU.
- Chaque commutateur se considère comme le pont racine jusqu'à ce qu'il découvre qu'il en va autrement.
- Sur tous les commutateurs Cisco Catalyst, la priorité de pont par défaut est 32768.
- *On se trouve donc dans une situation d'égalité ! Et après ?*


Moe
1
A B
Catalyst 1900
Ports 10BaseT (12)
Larry
A B
Ports 100BaseT
Catalyst 1900
Ports 10BaseT (24)
Ports 100BaseT
Curly
1
A B
Catalyst 1900
Ports 10BaseT (24)
Ports 100BaseT

Catalyst 1900 - Configuration « spanning tree » - Option 1

---------- Informations ----------

[V] VLAN affecté à l'option 1 -1005

---------- Paramètres ---------- ----------

[B] Priorité du pont **32768 (8000 hex)**

[M] Âge maximum en tant que racine 20 secondes

[H] Heure HELLO en tant que racine 2 secondes

[F] Délai de transmission en tant que racine 15 secondes

- *En cas de partage, l'ID de pont est utilisé...*
- ID de pont :

  → L'ID de pont est l'adresse MAC affectée à chaque commutateur.

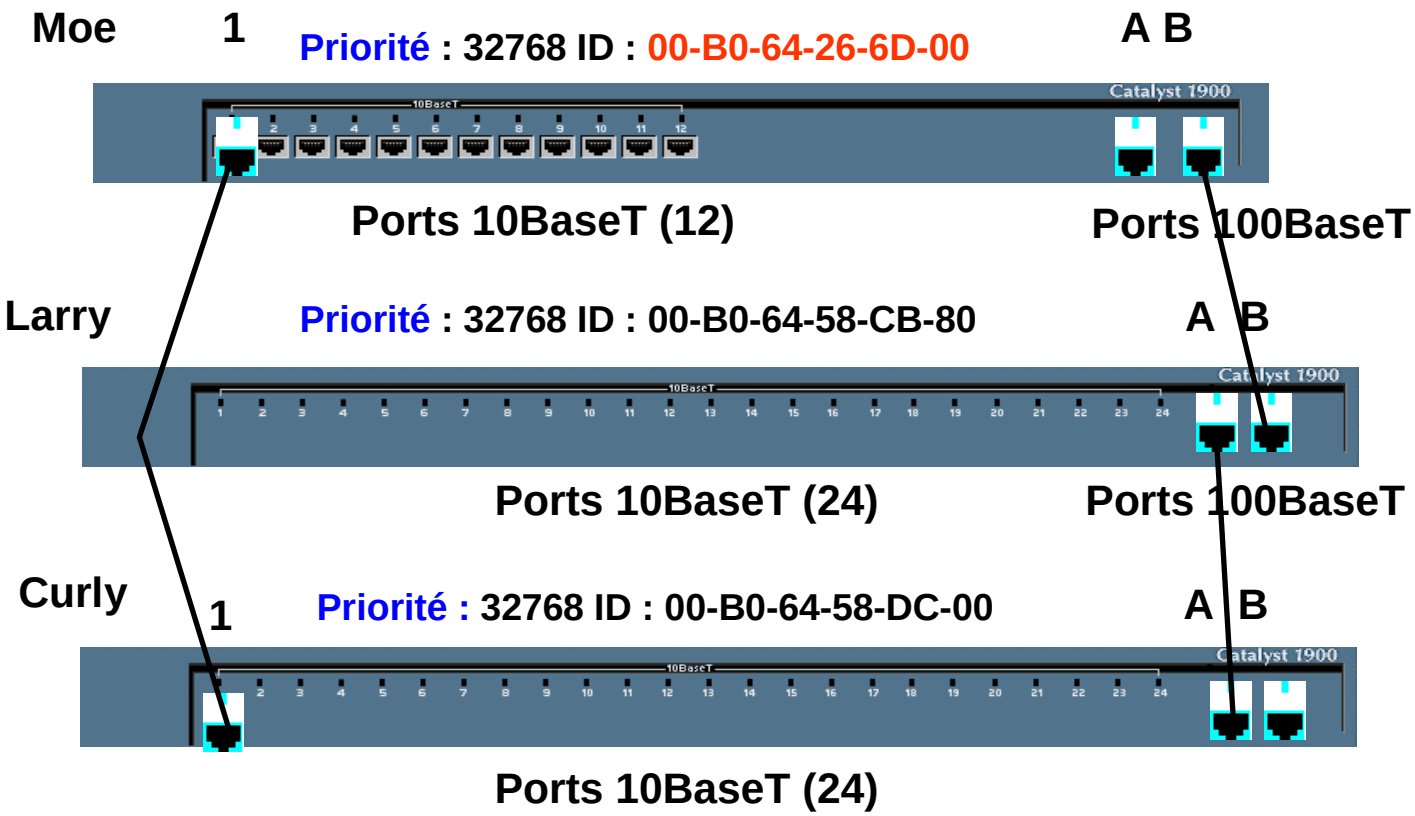
  → L'ID de pont inférieur (adresse MAC) permet de départager les ponts.

  → Chaque adresse MAC étant unique, vous avez ainsi la garantie qu'un seul pont aura la valeur la plus faible.

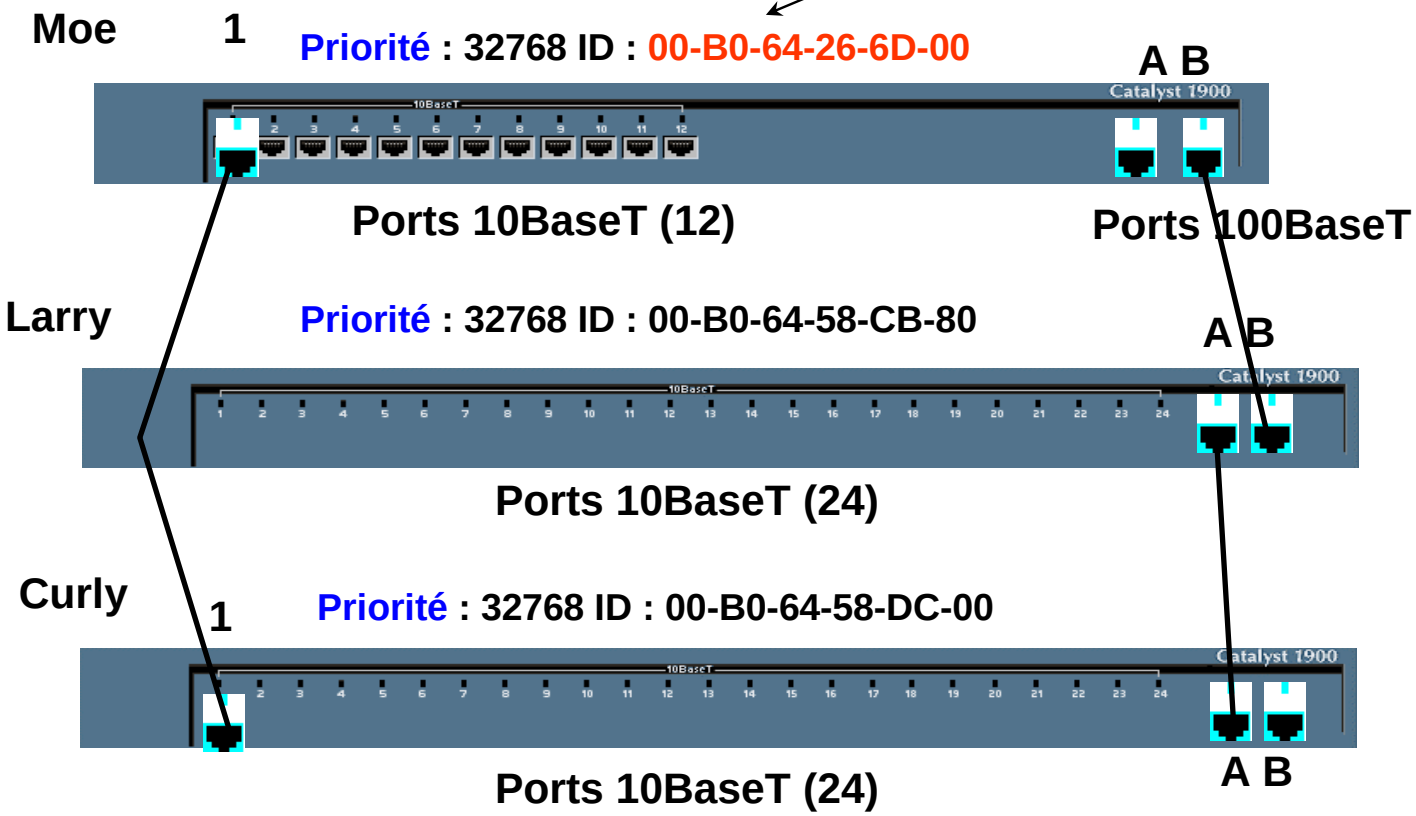
Catalyst 1900 Management Console
Copyright (c) Cisco Systems, Inc. 1993 -1998
Tous droits réservés.
Enterprise Edition Software
Addresse Ethernet : 00 -B0-64-26-6D-00

Numéro PCA : 73 -3122-04
Numéro de série PCA : FAB03503222
Numéro de modèle : WS -C1912 -EN
Numéro de série système : FAB0351U08M
N° de série alimentation : PHI033301VQ
Numéro de série PCB : FAB03503222,73 -3122-04

Qui a la valeur la plus faible ?

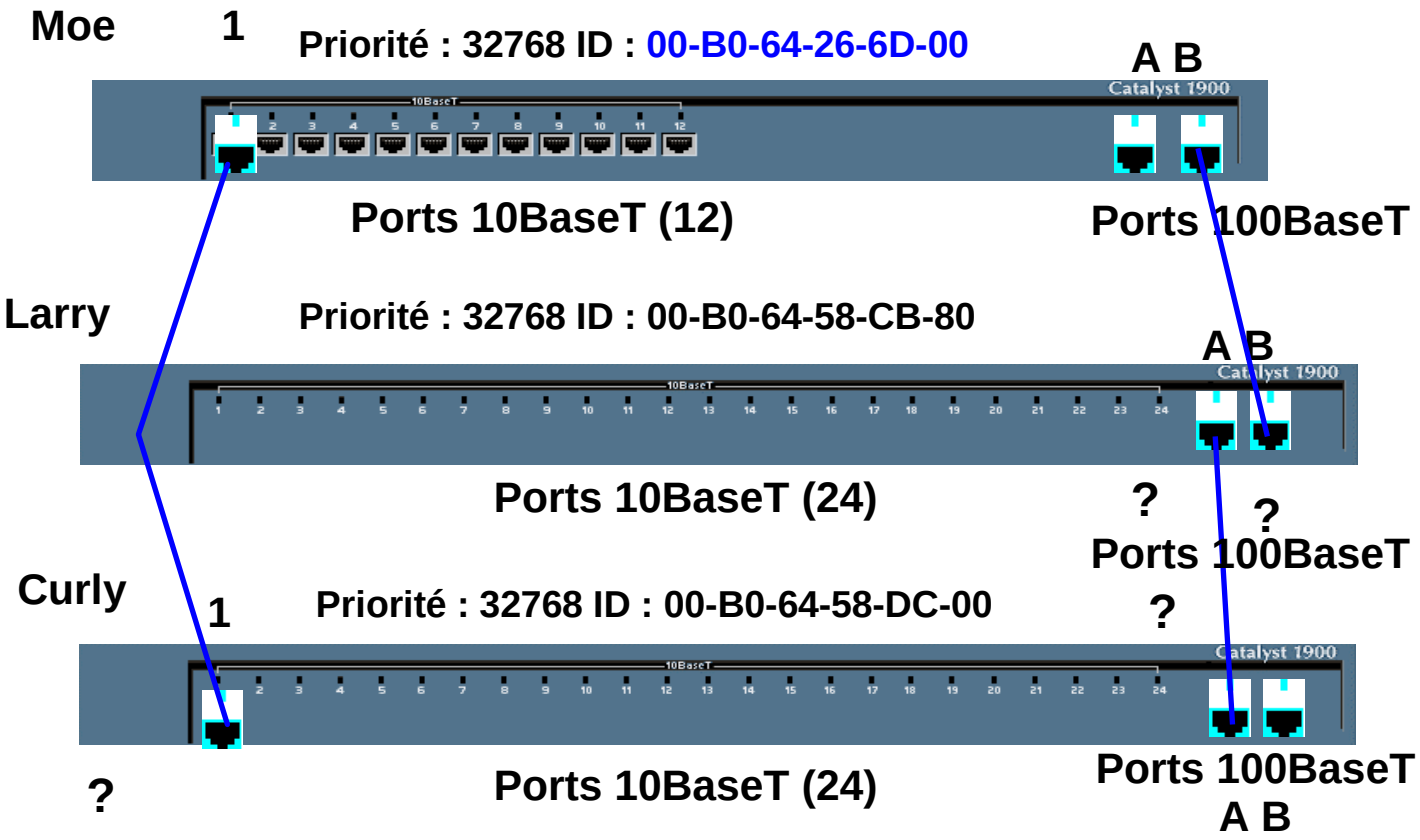


Valeur la plus basse : Moe devient le pont racine.



Étape 2 : Désignation des ports racine

- Objectif :
  - Une fois le pont racine sélectionné, les commutateurs (ponts) doivent localiser les *chemins redondants* vers le pont racine et ne laisser accessible qu'un seul de ces chemins (ce qui implique de *bloquer tous les autres*).
- Pour ce faire, les commutateurs utilisent des unités BPDU.
  - Comment le commutateur détermine-t-il le port à utiliser, connu sous le nom de port racine, et celui qui doit être bloqué ?


Moe
1
Priorité : 32768 ID : 00-B0-64-26-6D-00
A B
Catalyst 1900
Ports 10BaseT (12)
Ports 100BaseT
Larry
Priorité : 32768 ID : 00-B0-64-58-CB-80
A B
Catalyst 1900
Ports 10BaseT (24)
?
?
Ports 100BaseT
Curly
1
Priorité : 32768 ID : 00-B0-64-58-DC-00
?
Catalyst 1900
?
Ports 10BaseT (24)
Ports 100BaseT
A B

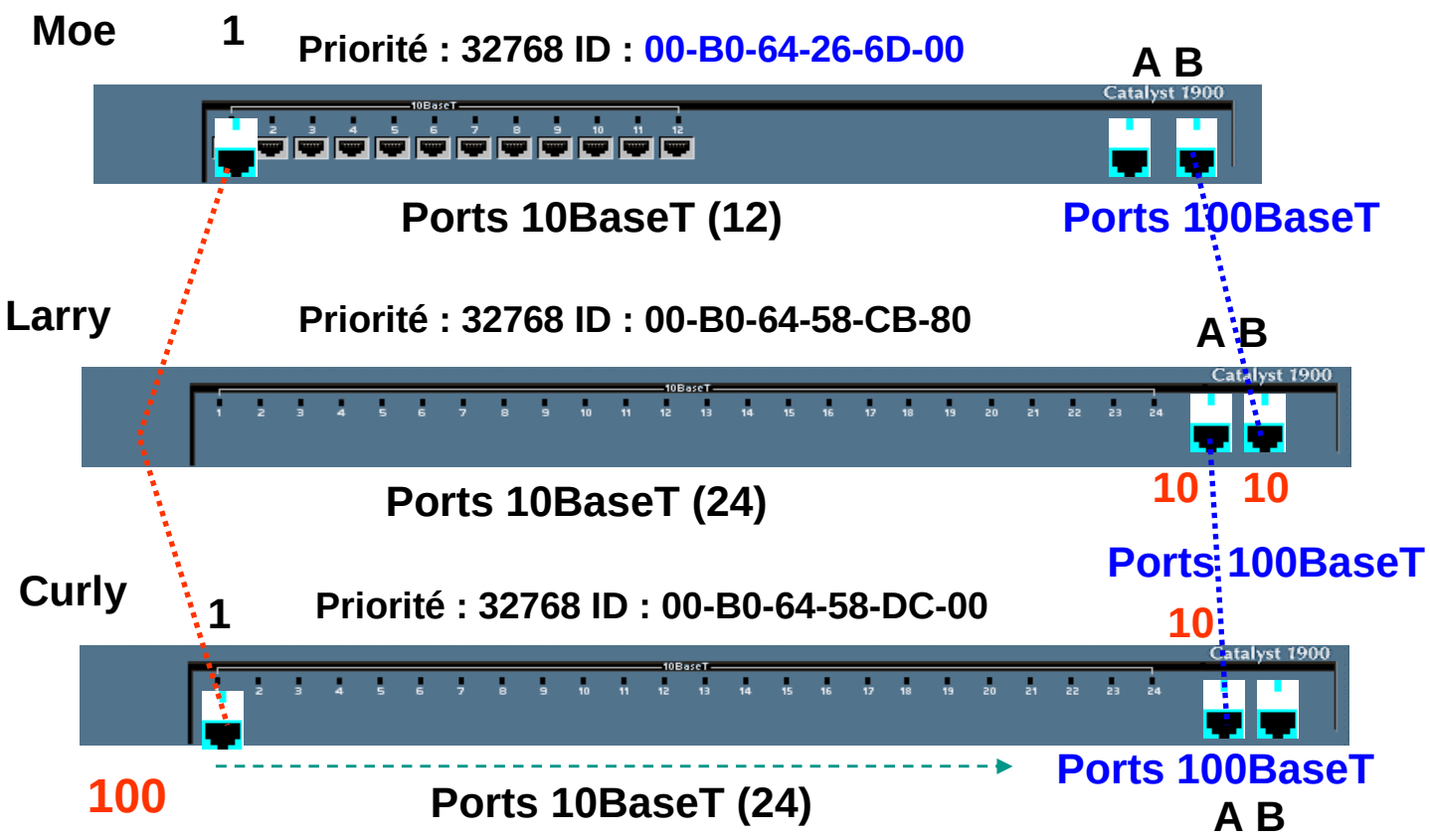
Coût de la route (ou Coût du port)

- Le coût du port est utilisé pour identifier le chemin " le moins cher " ou " le plus rapide " vers le pont racine.
- Par défaut, le coût du port est basé sur le média ou la bande passante du port.
- Sur les commutateurs Cisco Catalyst, on obtient cette valeur en divisant 1 000 par la vitesse du média, exprimée en mégabits par secondes (Mbit/s).
- Exemples :
  - Ethernet Standard : 1 000/10 = **100**
  - Fast Ethernet : 1 000/100 = **10**

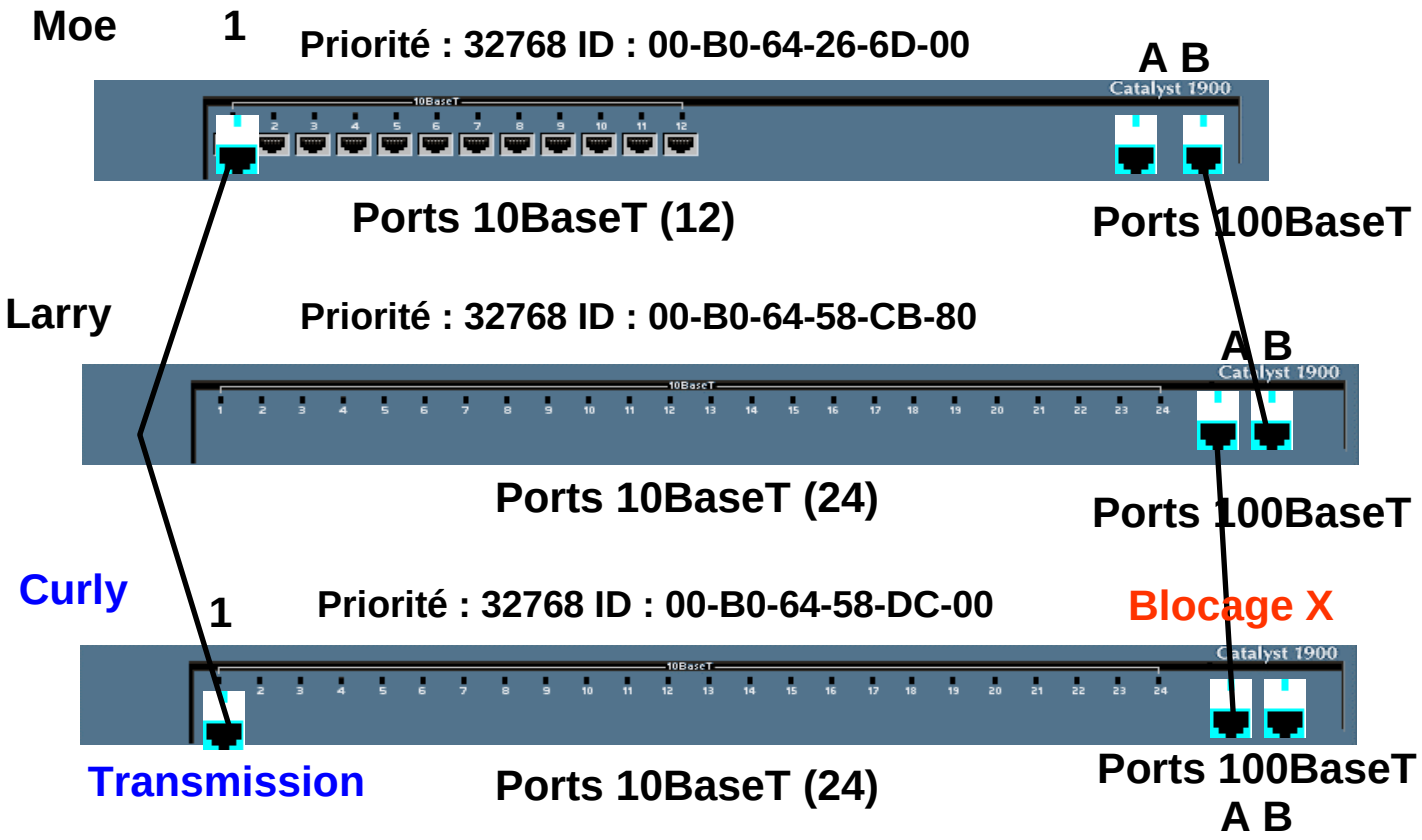
Coût du chemin racine

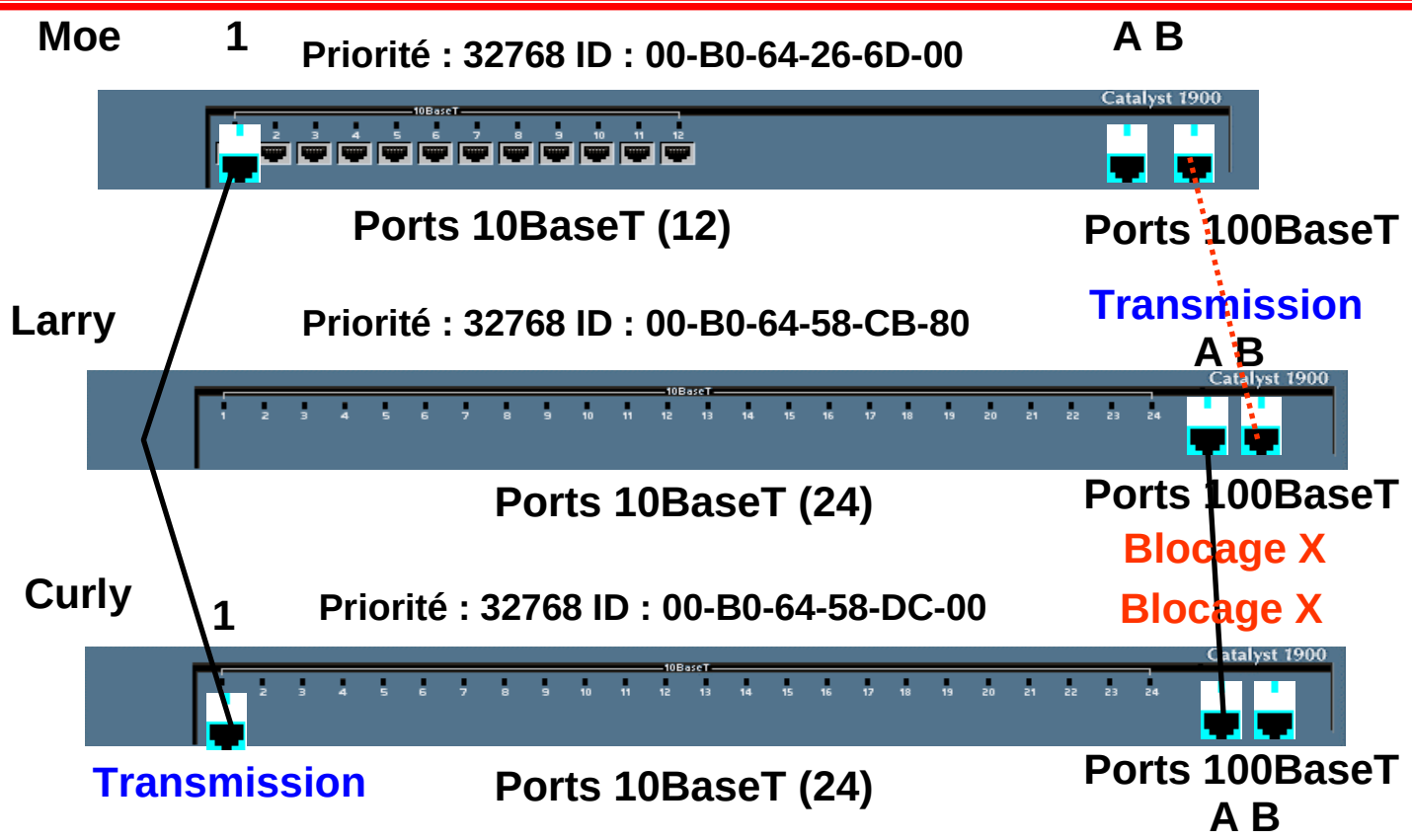
- Le coût du chemin racine correspond au *total des coûts de port* (coûts du chemin) *vers le pont racine*.
- Cette valeur est indiquée dans le champ de coût de l'unité BPDU.

- Cependant, tout est considéré par rapport au pont racine.
- Ports racine :
  - → Les ports *directement connectés* au pont racine seront *les ports racine*.
  - → Sinon, le port présentant le coût du chemin racine le plus bas sera le port racine.

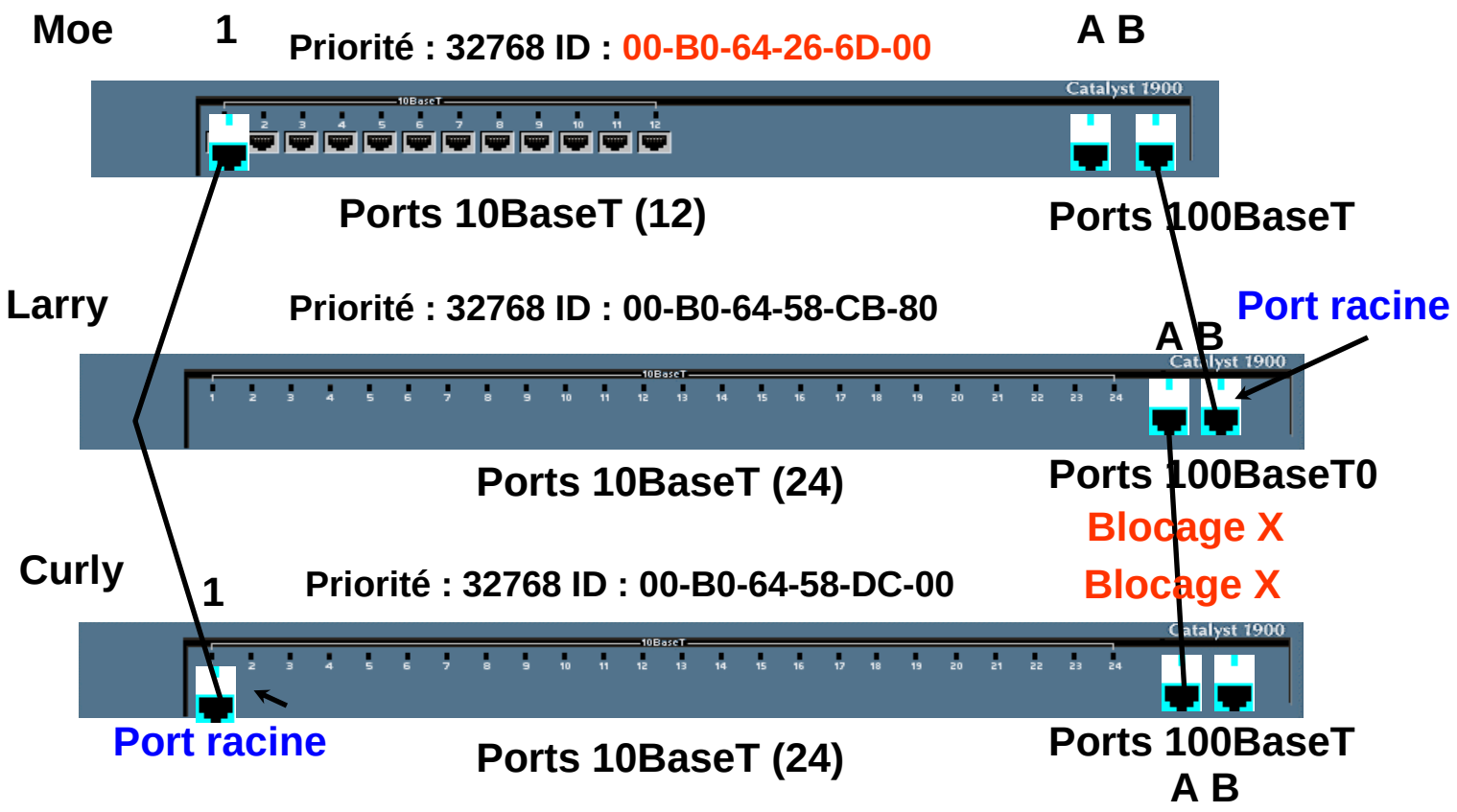


- **Bien que, pour Curly, le coût de la route vers le pont racine soit plus élevé en utilisant le port 1, le port 1 dispose d'une *connexion directe au pont racine*. Il devient donc le port racine.**
- **Le port 1 passe ensuite en mode Transmission, tandis que le chemin redondant du port A est placé en mode Blocage.**


Moe
1
Priorité : 32768 ID : 00-B0-64-26-6D-00
A B
Catalyst 1900
Ports 10BaseT (12)
Ports 100BaseT
Larry
Priorité : 32768 ID : 00-B0-64-58-CB-80
A B
Ports 10BaseT (24)
Ports 100BaseT
Curly
1
Priorité : 32768 ID : 00-B0-64-58-DC-00
Blocage X
Transmission
Ports 10BaseT (24)
Ports 100BaseT
A B

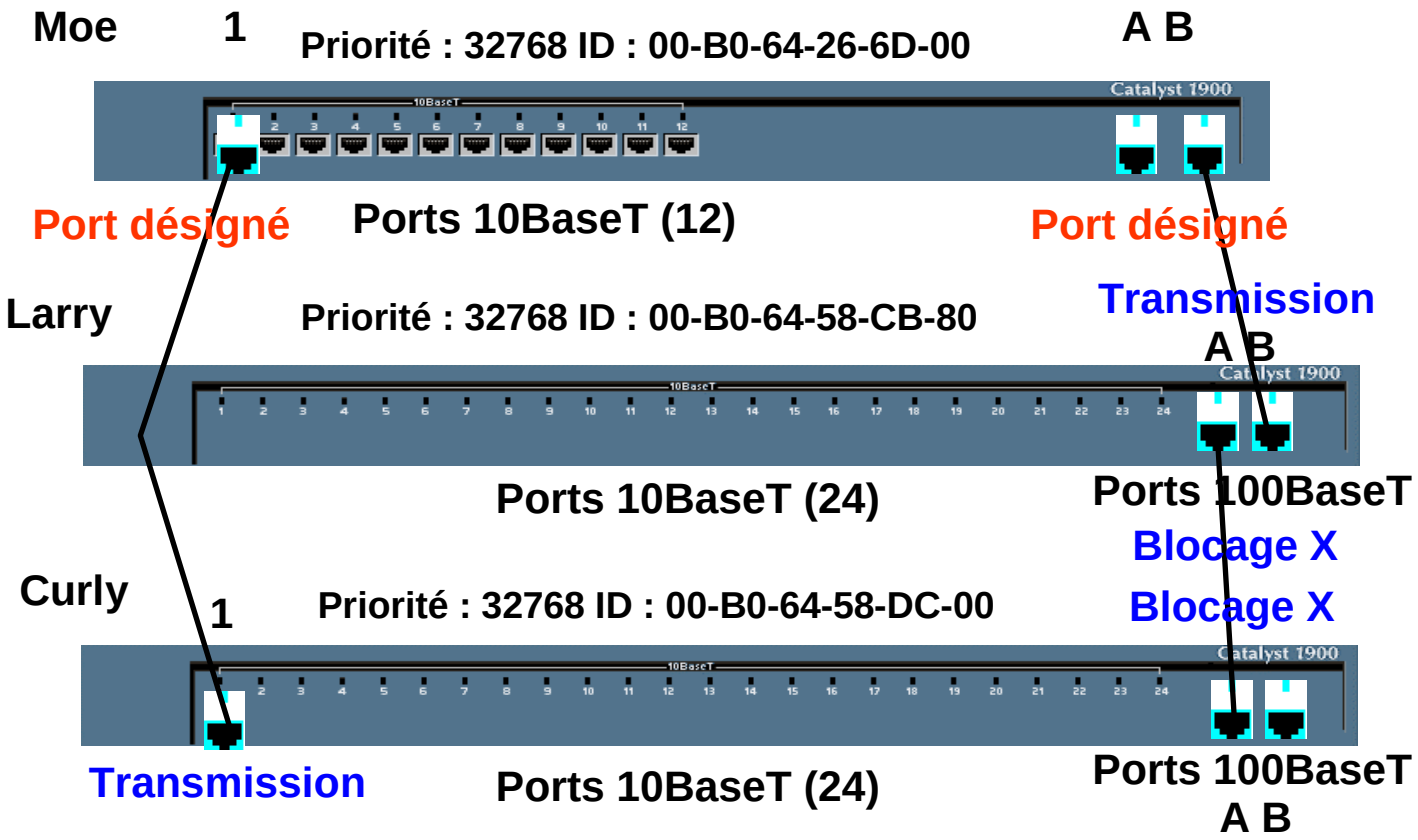


- Larry possède également un port racine, une connexion directe avec le pont racine, par le biais du port B.
  - Le port B passe ensuite en mode Transmission, tandis que le chemin redondant du port A est placé en mode Blocage.


Moe
1
Priorité : 32768 ID : 00-B0-64-26-6D-00
A B
Catalyst 1900
Ports 10BaseT (12)
Ports 100BaseT
Larry
Priorité : 32768 ID : 00-B0-64-58-CB-80
Port racine
A B
Ports 10BaseT (24)
Ports 100BaseT0
Blocage X
Curly
1
Priorité : 32768 ID : 00-B0-64-58-DC-00
Blocage X
Port racine
Ports 10BaseT (24)
Ports 100BaseT
A B

Étape 3 : Choix des ports désignés

- C'est le port unique d'un commutateur qui échange des données (trafic entrant et sortant) avec le pont racine.
- Il peut également être considéré comme le port annonçant le coût le plus bas vers le pont racine.
- Dans cet exemple, nous avons uniquement le choix entre les deux solutions évidentes, toutes deux au niveau du commutateur Moe.
- S'il y avait eu d'autres segments LAN, nous aurions pu donner une explication plus détaillée des ports désignés. Cependant, cela suffit pour l'instant.


Moe
1
Priorité : 32768 ID : 00-B0-64-26-6D-00
A B
Catalyst 1900
Port désigné
Ports 10BaseT (12)
Port désigné
Larry
Priorité : 32768 ID : 00-B0-64-58-CB-80
Transmission
A B
Catalyst 1900
Ports 10BaseT (24)
Ports 100BaseT
Blocage X
Curly
1
Priorité : 32768 ID : 00-B0-64-58-DC-00
Blocage X
Catalyst 1900
Transmission
Ports 10BaseT (24)
Ports 100BaseT
A B

La Spanning Tree est maintenant terminé.

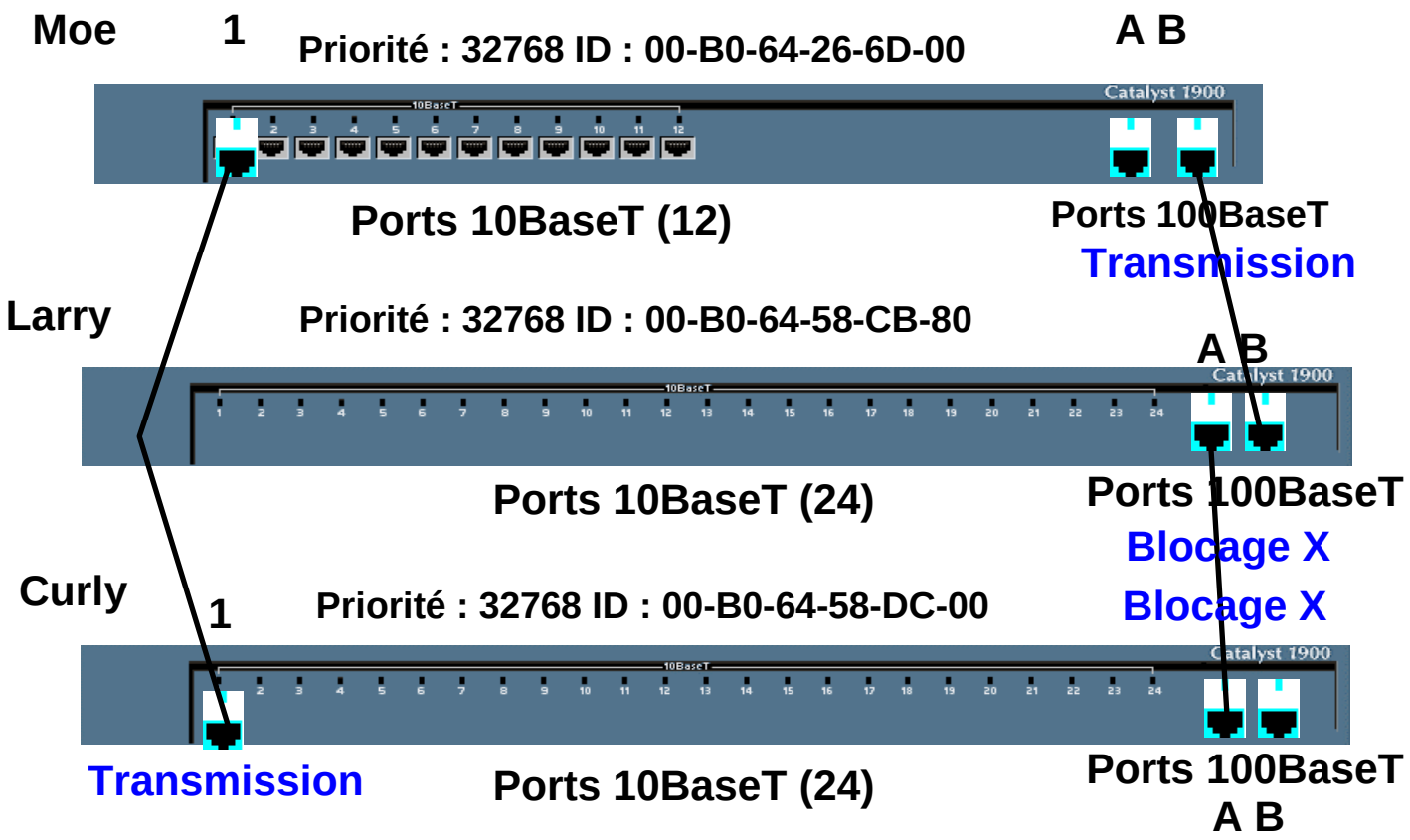
- Les commutateurs peuvent désormais commuter correctement les trames vers les ports appropriés avec les tables de commutation adéquates et sans créer de trames en double.

- La plus part des ouvrages traitant des réseaux locaux et des inter-réseaux commutés donnent des informations sur le protocole " spanning tree ". Pour obtenir des exemples plus complexes, nous vous invitons à consulter les ouvrages suivants :
  - Cisco Catalyst LAN Switching, par Rossi et Rossi, McGraw Hill (très accessible)
  - CCIE Professional Development: Cisco LAN Switching, par Clark et Hamilton, Cisco Press (niveau plus avancé)
  - Interconnections, par Radia Perlman, Addison Wesley (excellent mais très académique)

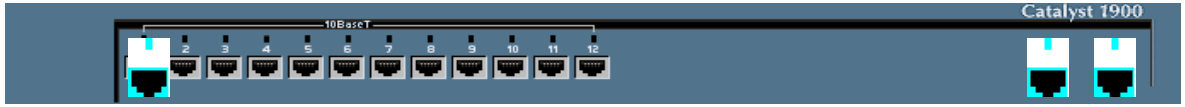
Mode rapide ou " Port Fast Mode " (extrait de la documentation Cisco)

- Ce mode fait passer immédiatement un port de l'état Blocage à l'état Transmission en éliminant le délai de transmission
  - c'est-à-dire la période d'attente observée par un port avant de quitter les états Apprentissage et Écoute STP et de passer à l'état Transmission

- Lors de la mise sous tension du commutateur, l'état Transmission est retardé (même si le mode rapide est activé) afin de permettre au protocole " spanning tree " de découvrir la topologie du réseau et de s'assurer qu'aucune boucle temporaire ne s'est formée.
  - Cette phase de détection dure environ 30 secondes, période pendant laquelle aucune transmission de paquets n'est effectuée.
  - À l'issue de la détection initiale, les ports en mode rapide passent directement de l'état Blocage à l'état Transmission.


Moe
1
Priorité : 32768 ID : 00-B0-64-26-6D-00
A B
Catalyst 1900
Ports 10BaseT (12)
Ports 100BaseT
Transmission
Larry
Priorité : 32768 ID : 00-B0-64-58-CB-80
A B
Catalyst 1900
Ports 10BaseT (24)
Ports 100BaseT
Blocage X
Curly
1
Priorité : 32768 ID : 00-B0-64-58-DC-00
Blocage X
Catalyst 1900
Transmission
Ports 10BaseT (24)
Ports 100BaseT
A B

Moe **1** **Priorité : 32768 ID : 00-B0-64-26-6D-00** **A B**



Port désigné **Ports 10BaseT (12)** **Ports 100BaseT**
Port désigné

Catalyst 1900 - Configuration du port 1

10Base-T intégré
802.1d - **État STP : Transmission** Transitions de transmission : **1 Activé**

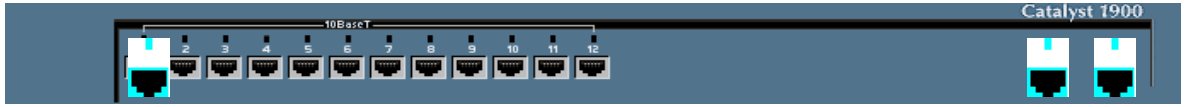
---------- Paramètres ----------

[D] Description/nom du port	
[S] Statut du port	Activé
[F] Full duplex	**Désactivé**
[I] Priorité du port (spanning tree)	**128 (80 hex)**
[C] Coût de la route (spanning tree)	**100**
[H] Mode rapide « Port fast » (spanning tree)	**Activé**

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal

Moe **1** **Priorité : 32768 ID : 00-B0-64-26-6D-00** **A B**



Ports 10BaseT (12) **Ports 100BaseT**

Catalyst 1900 - Configuration du port B

100Base-TX intégré
802.1d - **État STP : Transmission** Transitions de transmission : **1**
État de l'autonégociation : Full duplex

---------- Paramètres ----------

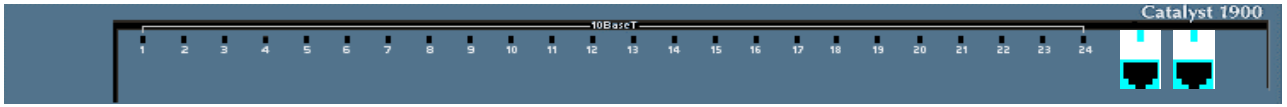
[D] Description/nom du port	
[S] Statut du port	**Activé**
[I] Priorité du port (spanning tree)	**128 (80 hex)**
[C] Coût de la route (spanning tree)	**10**
[H] Mode rapide « Port fast » (spanning tree)	**Désactivé**
[E] Contrôle de congestion étendu	**Désactivé**
[F] Full duplex / Contrôle de flux	**Autonégociation**

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal

Larry **Priorité : 32768 ID : 00-B0-64-58-CB-80**

A B



Ports 10BaseT (24) **Ports 100BaseT**

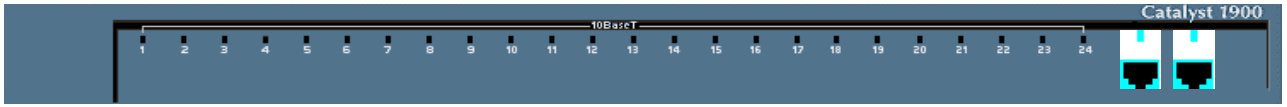
Catalyst 1900 - Configuration « spanning tree » des ponts de groupe 1
ID du pont : 8000 00 -B0 -64 -58 -CB -80

---------- Informations ----------
Racine désignée 8 000 00 -B0 -64 -26 -6D -00
Nombre de ports membres 27 Port racine **B**
Âge maximum (secondes) 20 Coût du chemin racine **10**
Délai de transmission (secondes) 15 Heure HELLO (secondes) **2**
Modifications topologiques 2
Dernière modification principale 0d00h48m58s [1]
---------- Paramètres ----------
[S] Algorithme et protocole « spanning tree » **Activé**
[B] Priorité du pont **32768 (8000 hex)**
[M] Âge maximum en tant que racine **20 secondes**
[H] Heure HELLO en tant que racine **2 secondes**
[F] Délai de transmission en tant que racine **15 secondes**

---------- Actions ----------
[N] Groupe de ponts suivant [G] Accès au groupe de ponts
[P] Groupe de p onts précédent [X] Retour au menu précédent

Larry **Priorité : 32768 ID : 00-B0-64-58-CB-80**

A B



Ports 10BaseT (24) **Ports 100BaseT**

Catalyst 1900 - Configuration du port A

100Base-TX intégré
802.1d - État STP : Blocage Transitions de transmission : 0
État de l'autonégociation : Autonégociation

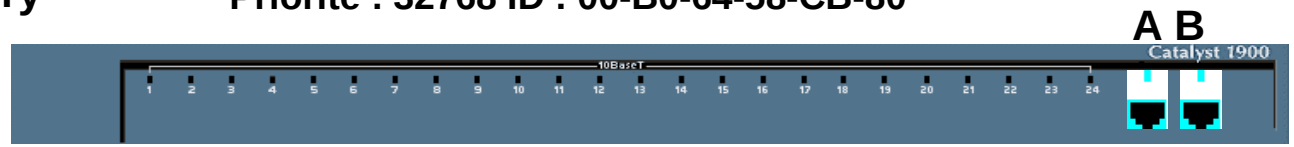
---------- Paramètres ----------

D] Description/nom du port	
[S] Statut du port	Interrompu - pas de signal de liaison
[I] Priorité du port (spanning tree)	128 (80 hex)
[C] Coût de la route (spanning tree)	10
[H] Mode rapide « Port fast » (spanning tree)	Désactivé
[E] Contrôle de congestion étendu	Désactivé
[F] Full duplex / Contrôle de flux	Autonégociation

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal

Larry **Priorité : 32768 ID : 00-B0-64-58-CB-80**



Ports 10BaseT (24) **Ports 100BaseT**

Catalyst 1900 - Configuration du port B

100Base-TX intégré
802.1d - État STP : Transmission Transitions de transmission : 1
État de l'autonégociation : Full duplex

---------- Paramètres ----------

[D] Description/nom du port	
[S] Statut du port	Activé
[I] Priorité du port (spanning tree)	128 (80 hex)
[C] Coût de la route (spanning tree)	10
[H] Mode rapide « Port fast » (spanning tree)	Désactivé
[E] Contrôle de congestion étendu	Désactivé
[F] Full duplex / Contrôle de flux	Autonégociation

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal

Catalyst 1900 - Configuration « spanning tree » des ponts de groupe 1
ID du pont : 8000 00-B0-64-58-DC-00

---------- Informations ----------

Racine désignée 8000 00-B0-64-26-6D-00

Nombre de ports membres	27	Port racine	1
Âge maximum (secondes)	20	Coût du chemin racine	100
Délai de transmission (secondes)	15	Heure HELLO (secondes)	2
Modifications topologiques	0		
Dernière modification principale	0d00h00m00s		

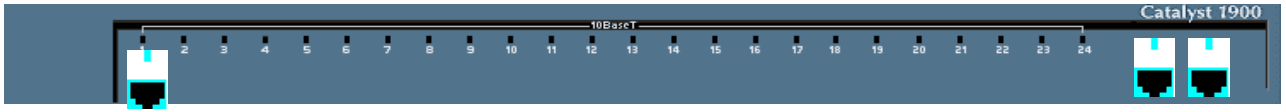
---------- Paramètres ----------

[S] Algorithme et protocole « spanning tree »	Activé
[B] Priorité du pont	32768 (8000 hex)
[M] Âge maximum en tant que racine	20 secondes
[H] Heure HELLO en tant que racine	2 secondes
[F] Délai de transmission en tant que racine	15 secondes

---------- Actions ----------

[N] Groupe de ponts suivant	[G] Accès au groupe de ponts
[P] Groupe de ponts précédent	[X] Retour au menu précédent

Curly **1** **Priorité : 32768 ID : 00-B0-64-58-DC-00** **Blocage X**



Transmission **Ports 10BaseT (24)** **Ports 100BaseT**
A B

Catalyst 1900 - Configuration du port 1

10Base-T intégré
802.1d - État STP : Transmission Transitions de transmission : 1

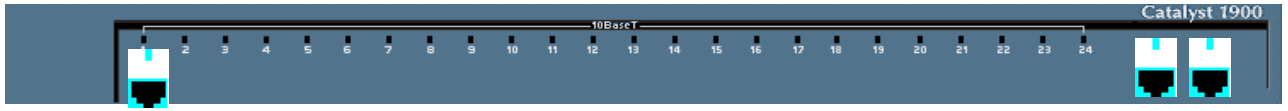
---------- Paramètres ----------

[D] Description/nom du port	
[S] Statut du port	Activé
[F] Full duplex	Désactivé
[I] Priorité du port (spanning tree)	128 (80 hex)
[C] Coût de la route (spanning tree)	100
[H] Mode rapide « Port fast » (spanning tree)	Activé

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal

Curly **1** **Priorité : 32768 ID : 00-B0-64-58-DC-00** **Blocage X**



Transmission **Ports 10BaseT (24)** **Ports 100BaseT**
A B

Catalyst 1900 - Configuration du port A

100Base-TX intégré
802.1d - État STP : Blocage Transitions de transmission : 0
État de l'autonégociation : Autonégociation

---------- Paramètres ----------

[D] Description/nom du port	
[S] Statut du port	Interrompu - pas de signal de liaison
[I] Priorité du port (spanning tree)	128 (80 hex)
[C] Coût de la route (spanning tree)	10
[H] Mode rapide « Port fast » (spanning tree)	Désactivé
[E] Contrôle de congestion étendu	Désactivé
[F] Full duplex / Contrôle de flux	Autonégociation

---------- Menus connexes ----------

[A] Adressage du port	[V] Affichage des statistiques sur le port
[N] Port suivant	[G] Accès au port
[P] Port précédent	[X] Retour au menu principal