

4 — Interconnexion des réseaux Ethernet

Réseaux
IUT de Villetaneuse

Département Informatique, Formation Continue

Année 2012–2013

<http://www.lipn.univ-paris13.fr/~evangelista/cours/2012-2013/reseaux-fc>

Introduction

Les ponts Ethernet

Stratégies de commutation

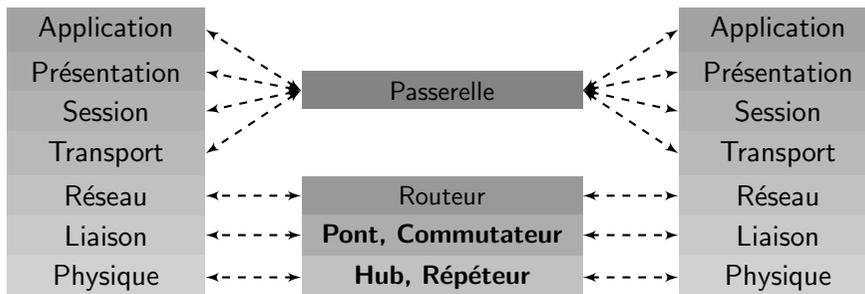
Construction d'un arbre couvrant

Introduction aux réseaux locaux virtuels

Equipements d'interconnexion

3/54

- ▶ Comment interconnecter des réseaux?
- ▶ Plusieurs possibilités selon le niveau auquel on se situe.

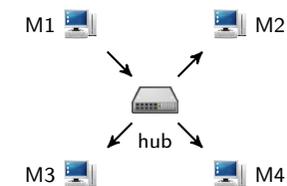


Dans ce cours : étude des ponts, commutateurs, hubs et répéteurs.

Au niveau physique

4/54

- ▶ Le répéteur
 - ▶ Il a deux ports en entrée.
 - ▶ Il régénère et amplifie les signaux reçus qui se dégradent sur des longues distances.
- ▶ Le hub
 - ▶ Il a plusieurs ports sur lesquels on peut brancher des cables Ethernet.
 - ▶ Un signal reçu sur un port est reproduit sur tous les autres ports.



- ▶ Ces équipements se situent au niveau physique car ils n'interprètent pas les bits reçus : il se contentent de les reproduire.

Des LANs de types différents

- ▶ Exemple : un réseau Ethernet ↔ un réseau Token Ring
- ▶ Plusieurs difficultés :
 - ▶ débits différents
 - ▶ Exemple : un LAN Token Ring à 16 Mbit/s ↔ LAN Ethernet à 1 Gbit/s
 - ⇒ le pont sera submergé de trames Ethernet
 - ▶ protocoles différents
 - ▶ Exemple : système de priorité dans Token Ring et pas dans Ethernet
 - ▶ tailles de trames maximales différentes
 - ▶ 1500 en Ethernet contre \approx 4000 octets en Token Ring
 - ⇒ nécessité de reformater les trames

⇒ peu de solutions satisfaisantes

Des LANs de même type

- ▶ Le même protocole est utilisé sur tous les réseaux connectés.
- ⇒ plus simple à mettre en œuvre
- ▶ Dans ce cours : interconnexion des réseaux Ethernet.

- ▶ Pont est un terme générique pour désigner un équipement qui relie
 - ▶ différents LANs
 - ▶ qui peuvent être de différents types (Ethernet, Token Ring).
- ▶ Un commutateur (ou switch) relie des LANs ou des équipements de même type.

Introduction

Les ponts Ethernet

Stratégies de commutation

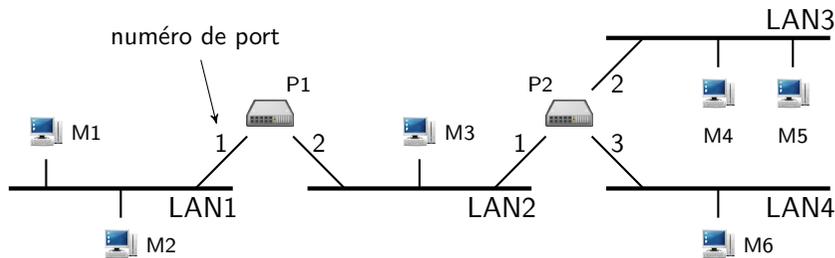
Construction d'un arbre couvrant

Introduction aux réseaux locaux virtuels



- ▶ marque : D-LINK
- ▶ 16 ports en façade pour brancher d'autres équipements (hub, switches, machines, ...).

- ▶ Un pont Ethernet connecte plusieurs LANs Ethernet.
- ▶ Exemple : 4 réseaux Ethernet connectés par 2 ponts P1 et P2.



- ▶ Son rôle est simple :
 - ▶ attendre une trame
 - ▶ déterminer le(s) LAN(s) sur lequel(s) la trame doit être redirigée en utilisant une **table de commutation**
 - ▶ retransmettre la trame sur ce(s) LAN(s)
- ▶ **commutation** = action de mettre en relation deux correspondants (l'émetteur et le récepteur de la trame)

Composants d'un pont

11/54

- ▶ **ports** — permet de relier le pont à plusieurs LANs Ethernet. 1 port / LAN
- ▶ **mémoire tampon** — y sont stockées les trames en attente de retransmission.
- ▶ **table de commutation** — utilisée pour déterminer le port vers lequel une trame reçue doit être redirigée.
- ▶ **unité de calcul** — permet d'exécuter le protocole CSMA/CD, calculer des CRCs pour détecter des erreurs sur les trames, ...

Caractéristiques des ponts

10/54

- ▶ Le pont se situe au niveau **2 - Liaison** du modèle OSI car :
 - ▶ Il a une adresse physique (MAC).
 - ▶ Il connaît les adresses MAC des machines qu'il relie.
 - ▶ Il exécute le protocole CSMA/CD.
 - ▶ Il comprend et interprète les trames qu'il reçoit.
- ▶ Le pont traite toutes les trames qu'il reçoit, quelle que soit l'adresse de destination.
- ▶ Un pont est un hub intelligent car :
 - ▶ il sait comment joindre la machine destinataire d'une trame
 - ▶ et il retransmet la trame uniquement sur le bon LAN
- ▶ Un pont est dit **transparent** :
 - ▶ Les machines voient tous les LANs connectés comme un unique LAN.
 - ▶ Les machines n'ont pas connaissance de l'existence des ponts.
- ▶ Un pont n'a pas de "carte" globale du réseau.
- ▶ L'acheminement d'une trame à son destinataire final se fait de **proche en proche**.
- ▶ La trame passe de pont en pont pour arriver à son destinataire final.

La table de commutation

12/54

- ▶ La table de commutation contient une liste d'enregistrements
(adresse, port)
- ▶ Un enregistrement (a,p) est interprété de la façon suivante :
Je sais que toute trame destinée à la machine dont l'adresse MAC est a doit être retransmise sur le port p .

4 réseaux Ethernet connectés avec 2 ponts

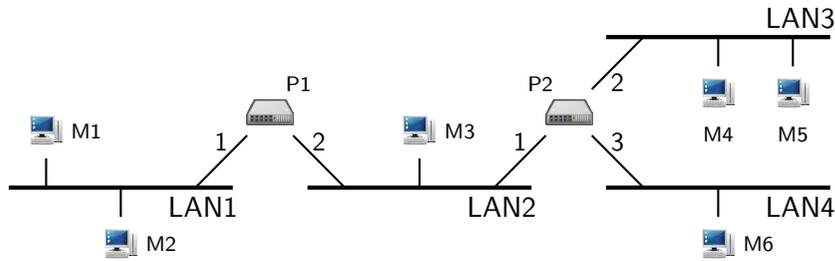


Table de P1	
adresse	port
M1	1
M2	1
M3	2
M4	2
M5	2
M6	2

Table de P2	
adresse	port
M1	1
M2	1
M3	1
M4	2
M5	2
M6	3

Remplissage des tables

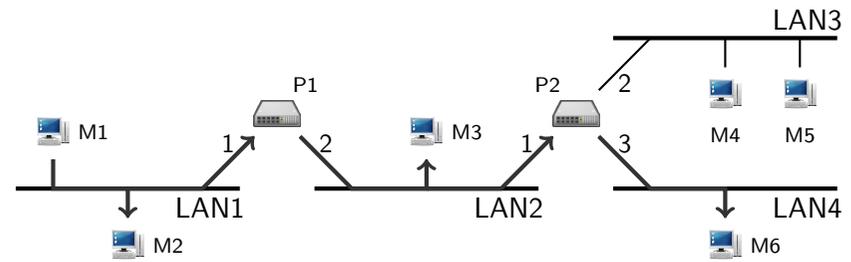
Dans l'exemple précédent, nous avons supposé que les tables de commutation étaient déjà remplies. Comment le sont-elles?

Manuellement ...

- ▶ L'administrateur du réseau crée un enregistrement par machine à l'installation du réseau.
- ▶ Plusieurs inconvénients :
 - ▶ fastidieux
 - ▶ erreur(s) possible(s) de l'administrateur
 - ▶ nécessité de modifier les tables quand on déplace une machine ou connecte une nouvelle machine

... ou automatiquement

- ▶ Les tables sont initialement vides.
- ▶ Les ponts remplissent leurs tables au fur et à mesure qu'ils découvrent la topologie du réseau.

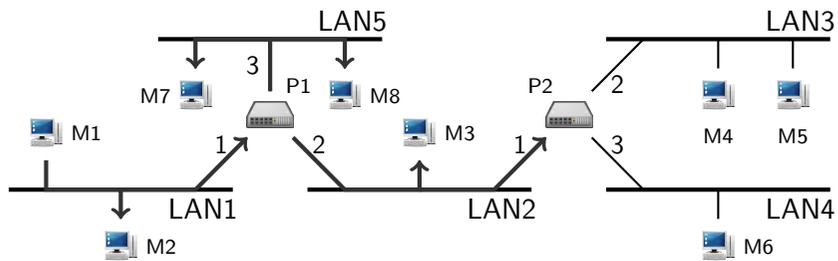


- ▶ $t = 0$ — M1 veut envoyer une trame à M6
⇒ il la transmet sur le LAN1.
- ▶ $t = 1$ — Le pont P1 reçoit la trame. Le destinataire est M6. Il consulte sa table et trouve l'enregistrement (M6, 2).
⇒ il transmet la trame sur le LAN2 en utilisant son port 2.
- ▶ $t = 2$ — Le pont P2 reçoit la trame. Le destinataire est M6. Il consulte sa table et trouve l'enregistrement (M6, 3).
⇒ il transmet la trame sur le LAN4 en utilisant son port 3.
- ▶ $t = 3$ — M6 reçoit la trame.

Remplissage des tables — Méthode

Comment un pont remplit-il sa table de commutation?

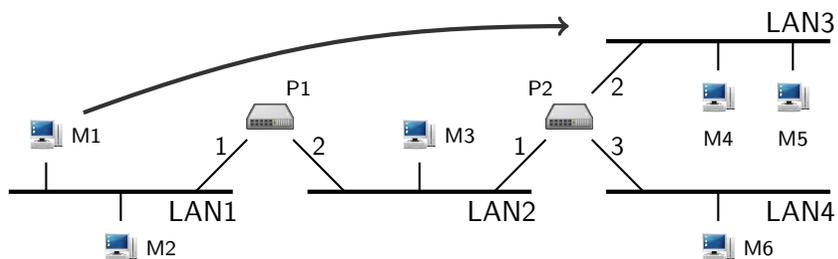
- ▶ À la réception d'une trame sur un port p, le pont
 - ▶ regarde l'adresse de l'émetteur de la trame (champ SA)
 - ⇒ il sait que la machine d'adresse SA peut être jointe en utilisant le port p
 - ▶ insère un enregistrement (SA, p) dans sa table de commutation
- ▶ Si il ne connaît pas le port à utiliser pour atteindre la machine destinataire (champ DA) il devra retransmettre la trame sur tous les ports possibles à l'exception du port p.



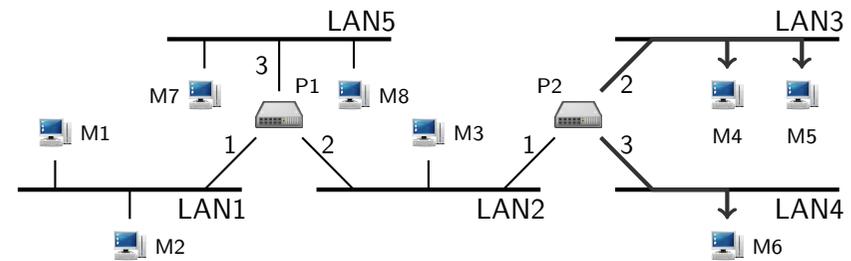
- ▶ $t = 0$ — Les tables de P1 et P2 sont vides.
M1 veut envoyer une trame à M6
- ⇒ Il la transmet sur le LAN1.
- ▶ $t = 1$ — Le pont P1 reçoit la trame sur son port 1.
- ⇒ **Il insère (M1, 1) dans sa table.**
Le destinataire est M6. Il consulte sa table et ne trouve pas M6.
- ⇒ Il transmet la trame sur le LAN2 et le LAN5 en utilisant ses ports 2 et 3.

Mobilité des machines

- ▶ Les machines peuvent être déplacées d'un LAN à un autre.
- ▶ Exemple : la machine M1 est déplacée du LAN1 au LAN3.



- ▶ Problème : pour les ponts P1 et P2, M1 est toujours sur le LAN1.
- ⇒ M1 recevra uniquement les trames de M4 et M5.
- ▶ Solution : les enregistrements de la table de commutation ont une durée de vie.
- ▶ Passé un certain délai ils sont automatiquement supprimés.
- ⇒ Les trames destinées à M1 seront, après la suppression, transmises par P1 et P2 sur tous leurs ports et reçues par M1.



- ▶ $t = 2$ — Le pont P2 reçoit la trame sur son port 1.
- ⇒ **Il insère (M1, 1) dans sa table.**
Le destinataire est M6. Il consulte sa table et ne trouve pas M6.
- ⇒ Il transmet la trame sur le LAN3 et le LAN4 en utilisant ses ports 2 et 3.
- ▶ $t = 3$ — M6 reçoit la trame.
- ▶ $t = 4$ — M6 veut répondre à M1. L'adresse de M1 est connue des deux ponts.
- ⇒ La trame passera par le LAN4 puis par le LAN2 et enfin par le LAN1 sans "polluer" le LAN3 et le LAN5.

Algorithme de commutation

Traitement par un pont d'une trame reçue sur un port p

extraire les champs DA et SA de la trame

on connaît maintenant le port pour joindre la machine d'adresse SA

⇒ *on met à jour de la table de commutation*

si il existe un enregistrement (adr, p') dans la table avec $adr = SA$

supprimer cet enregistrement

insérer (SA, p) dans la table

on retransmet la trame vers le bon port ou sur tous les ports si

on ne connaît pas le port pour joindre la machine d'adresse DA

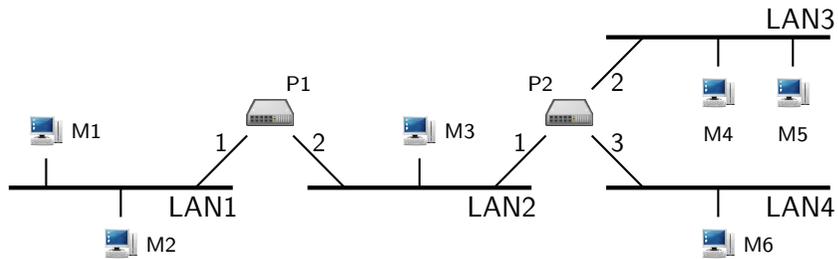
si il existe un enregistrement (adr, p') dans la table avec $adr = DA$

transmettre la trame sur le port p'

sinon

pour tout port $p' \neq p$

transmettre la trame sur le port p'



1. connecter des machines
2. s'affranchir des contraintes de distance d'Ethernet et augmenter la couverture du réseau
3. meilleures performances grâce à la réduction du **domaine de collision**
 - ▶ Exemple : M4 peut envoyer une trame à M5 pendant que M1 envoie une trame à M2. Il ne peut pas y avoir de collision.
 - ▶ moins de collisions ⇒ meilleur débit utile
4. sécurité accrue : les machines ne voient plus circuler toutes les trames
 - ▶ Exemple : M4 et M5 peuvent s'échanger des données confidentielles. Aucune autre machine ne les recevra car elles seront filtrées par P2.

- ▶ À partir de quel instant un pont va-t-il relayer une trame reçue?
- ▶ plusieurs stratégies possibles
- ▶ Nous allons en voir 2 :
 - ▶ **Store and Forward** — réception complète, contrôle d'erreur puis retransmission
 - ▶ **Fast Forward** — retransmission au plus tôt/dès que possible

Introduction

Les ponts Ethernet

Stratégies de commutation

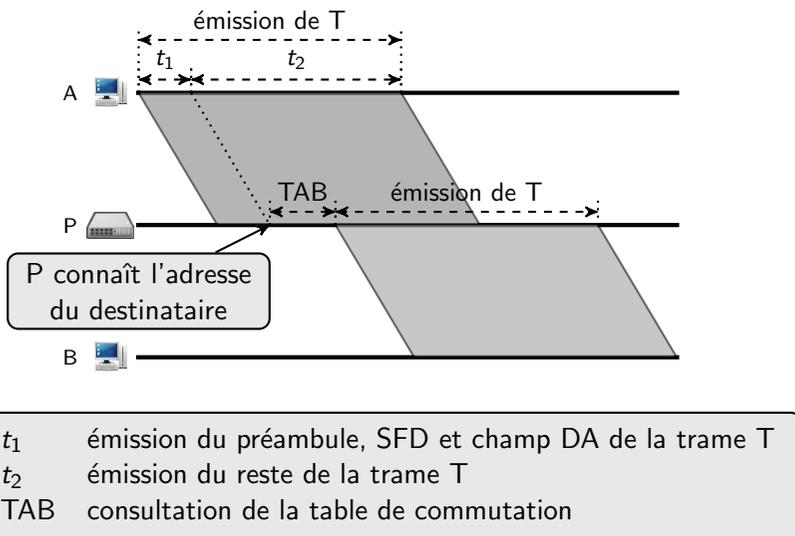
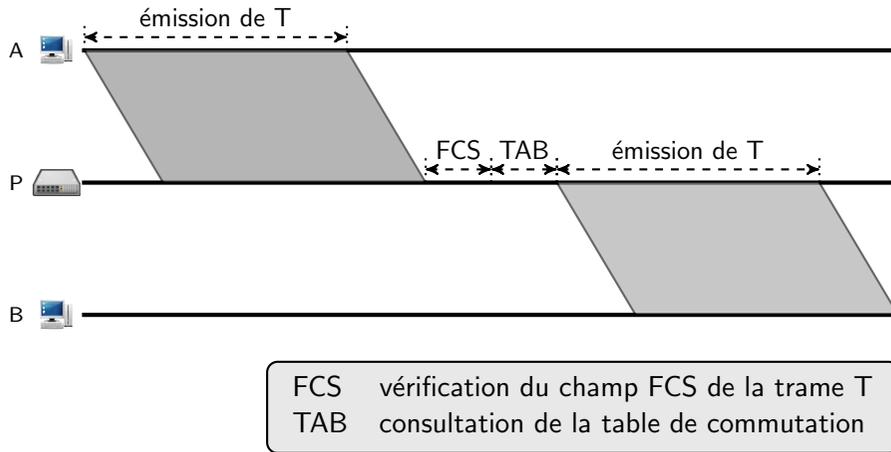
Construction d'un arbre couvrant

Introduction aux réseaux locaux virtuels

Algorithme de réception/retransmission d'une trame

1. Le pont attend l'arrivée de **tous les bits** de la trame.
2. Une fois que tous les bits de la trame sont arrivés, le pont vérifie qu'elle n'est pas erronée en calculant le FCS.
3. Si la trame est erronée elle est ignorée.
4. Sinon le pont extrait de la trame l'adresse du destinataire (champ DA).
5. Il consulte la table de commutation pour déterminer le(s) port(s) sur lequel(s) la trame doit être transmise.
6. Il retransmet la trame sur ces ports.

- ▶ La machine A émet une trame T à destination de la machine B.
- ▶ Un pont P relie les LANs sur lesquels se trouvent A et B.



Algorithme de réception/retransmission d'une trame

1. Le pont attend l'arrivée de **l'adresse de destination** (champ DA) de la trame.

Rappel sur la structure des trames Ethernet :

Préambule	SFD	DA	SA	DL/EType	Données	Bourrage	FCS
7 octets	1 o.	6 o.	6 o.	2 o.	0–N o.	0-46 o.	4 o.

↑

- 2 Dès que l'adresse du destinataire est connu, le pont consulte sa table de commutation pour déterminer le(s) port(s) sur lequel(s) la trame doit être transmise.
- 3 Il commence aussitôt la retransmission de la trame sur ces ports.

- ▶ Fast Forward est (potentiellement) plus rapide :
 - ▶ Dès que l'adresse du destinataire est connue, le pont relaie la trame. ⇒ pas besoin d'avoir reçu toute la trame pour commencer à retransmettre
 - ▶ Mais en cas de problème sur la trame à relayer le pont utilisant la méthode fast forward retransmettra une trame erronée :
 - ▶ si au moins un bit de la trame relayée est incorrect (pas de vérification du champ FCS en Fast Forward)
 - ▶ ou si une collision survient et est détectée par le pont après la réception de l'adresse du destinataire
- ⇒ La méthode Fast Forward peut augmenter le nombre de collisions.
 ⇒ pas de meilleure méthode

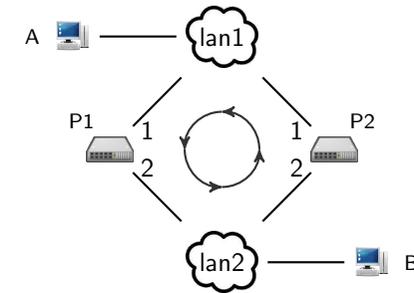
Introduction

Les ponts Ethernet

Stratégies de commutation

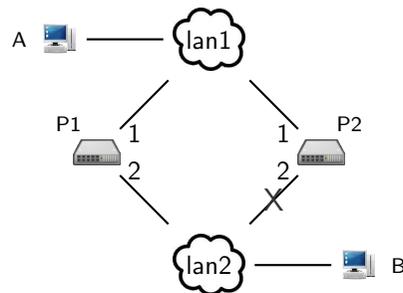
Construction d'un arbre couvrant

Introduction aux réseaux locaux virtuels



- ▶ Hypothèses
 - ▶ Les tables de commutation de P1 et P2 sont vides.
 - ▶ La machine A envoie une trame à la machine B.
- ▶ La trame est reçue par P1 et P2 sur le LAN1. Ils vont ensuite se l'échanger sur le LAN2, puis sur le LAN1, puis sur le LAN2 ...
- ⇒ La trame va circuler indéfiniment sur le réseau.
- ▶ Ce problème se pose lorsqu'il existe plusieurs chemins pour arriver d'un pont à un autre.

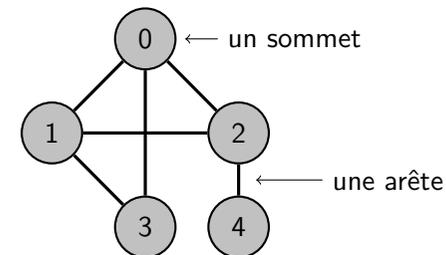
Solution au problème de cycle



- ▶ éliminer les cycles de manière **logique**
- ▶ en bloquant certains ports
- ▶ à l'aide du protocole de l'**arbre couvrant**
 - ▶ protocole **STP** (Spanning Tree Protocol), norme IEEE 802.1D
 - ▶ STP est un protocole utilisé par les ponts uniquement.
- ▶ L'émission et la réception sur un port bloqué sont interdites.
- ⇒ 2 machines ne pourront pas être reliées par 2 chemins différents
- ▶ En bloquant le port 2 du pont P2 il n'y a plus de cycle possible.

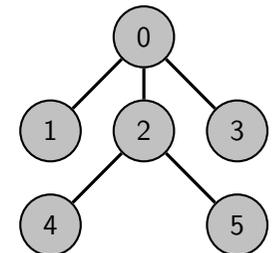
Notions de graphe, d'arbre et d'arbre couvrant (1/2)

Un graphe



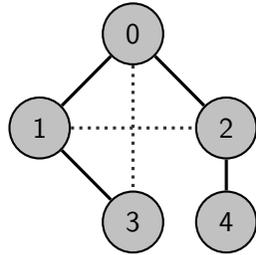
- ▶ 1 cycle = 1 chemin qui part d'un sommet et revient à ce sommet sans emprunter deux fois la même arête
- ▶ Exemples de cycles :
 - ▶ 0, 1, 3, 0
 - ▶ 0, 3, 1, 2, 0

Un arbre



- ▶ 1 arbre = 1 graphe sans cycle

Un arbre couvrant

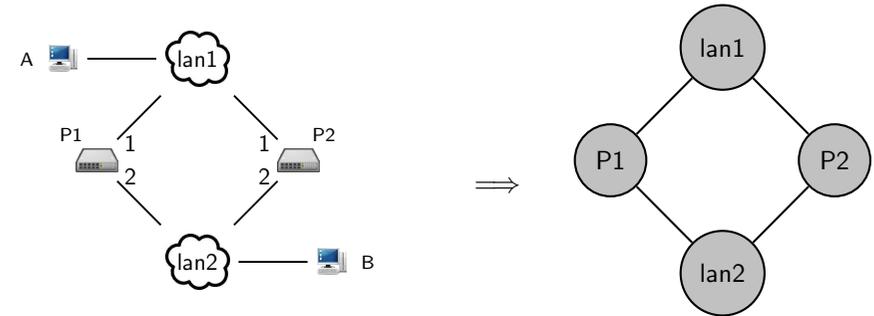


- ▶ On supprime des arêtes du graphe pour obtenir un arbre.
- ▶ Tous les sommets doivent rester connectés.
- ▶ Pour un graphe, on peut avoir plusieurs arbres couvrants.

Principe du protocole de l'arbre couvrant

- ▶ Un **pont racine** va être désigné.
- ▶ C'est le pont ayant la plus petite adresse MAC (parmi tous les ponts).
- ▶ Au début, le pont racine n'est pas connu et chaque pont se considère comme le pont racine.
- ▶ Les ponts vont progressivement découvrir l'identité du pont racine.
- ▶ Chaque pont va découvrir un **chemin de longueur minimale** qui le mène au pont racine.
- ▶ Tous les ports qui n'apparaissent pas sur un des chemins calculés sont bloqués : ils ne font pas partie de l'arbre couvrant.
- ▶ Mais les ports bloqués sont toujours utilisés par le protocole STP.
 - ▶ Les trames du protocole reçues sur un port bloqué sont traitées.

- ▶ Les LANs et ponts correspondent aux sommets.
- ▶ Les connexions pont↔LAN correspondent aux arêtes.



Classification des ports

A l'issue du protocole un pont associera un des trois états suivants à chacun de ses ports :

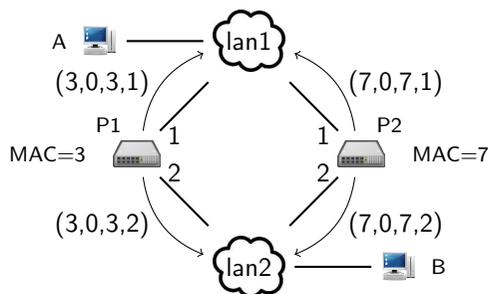
- ▶ **bloqué** — Le pont ignorera les trames qu'il recevra sur ce port (sauf les BPDUs).
- ▶ **racine** — C'est le port qu'utilise le pont pour joindre le pont racine par le chemin de longueur minimale qu'il a calculé.
- ▶ **désigné** — Ce sont les ports qu'utilisent un pont pour joindre les autres ponts qui sont plus éloignés que lui du pont racine.

- ▶ Chaque pont a trois variables qui forment sa **configuration** :
 - ▶ **racine** — l'adresse MAC du pont qu'il considère (pour l'instant) comme le pont racine
 - ▶ **distance** — sa distance par rapport à racine
 - ▶ **portRacine** — le port utilisé pour atteindre racine
- ▶ Au début, tout pont se considère comme le pont racine et initialise
 - ▶ racine à son adresse MAC
 - ▶ distance à 0
 - ▶ et portRacine à -1 (un port inexistant)
- ▶ Les ponts s'échangent ensuite des BPDUs.
- ▶ La réception d'une BPDu peut éventuellement modifier la configuration d'une machine.
- ▶ Le protocole termine quand les configurations sont stables : l'envoi d'une BPDu ne peut plus changer la configuration d'une machine.
- ▶ A la terminaison du protocole, tous les ponts connaissent le pont racine et ont bloqué des ports qui créent des cycles.

```

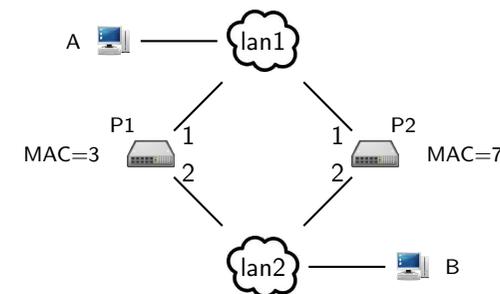
racine = moi ; distance = 0
portRacine = -1
} initialisation de ma configuration
tant que les configurations changent
    pour chaque port p non bloqué
        et ≠ de portRacine
        envoyer (racine, distance, moi, p)
            sur le port p
        } j'envoie des BPDUs sur tous les ports non bloqués et ≠ du port racine
    attendre une (ou des) BPDUs
    soit (r,d,e,p) la meilleure BPDu reçue
    soit pr le port de réception de (r,d,e,p)
    } je réceptonne des BPDUs et je sélectionne la meilleure
    si (r, d + 1) < (racine, distance)
        racine = r
        distance = d + 1
        portRacine = pr
        } j'ai trouvé un nouveau pont racine ou un chemin plus court qui y mène ⇒ je mets à jour ma configuration
    pour chaque port pr ≠ portRacine
        sur lequel j'ai reçu (r,d,e,p)
        } je bloque les ports qui mènent au pont racine et qui créent un cycle
        bloquer le port pr
    
```

Exemple (1/4)



- ▶ Initialisation des configurations :
 - ▶ pour P1 : *racine* = 3, *distance* = 0 et *portRacine* = -1
 - ▶ pour P2 : *racine* = 7, *distance* = 0 et *portRacine* = -1
- ▶ P1 envoie (3,0,3,1) sur le LAN1 et (3,0,3,2) sur le LAN2.
- ▶ P2 envoie (7,0,7,1) sur le LAN1 et (7,0,7,2) sur le LAN2.

Exemple (2/4)



- ▶ meilleure BPDu reçue par P1 : (r, d, e, p) = (7, 0, 7, 1)
- ▶ P1 vérifie si sa configuration doit changer :

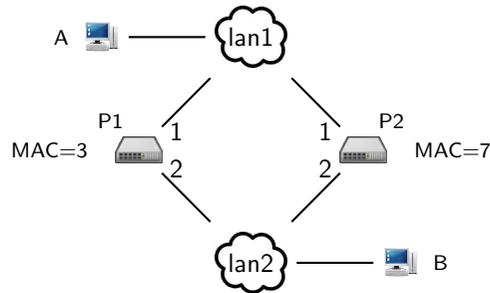
$$\text{est-ce que } (r, d + 1) < (\text{racine}, \text{distance}) ?$$

$$(7, 1) < (3, 0) \quad ?$$

non ⇒ la configuration de P1 ne change pas

Exemple (3/4)

45/54



- ▶ meilleure BPDU reçue par P2 : $(r, d, e, p) = (3, 0, 3, 1)$
- ▶ P2 vérifie si sa configuration doit changer :

$$\begin{array}{l} \text{est-ce que } (r, d + 1) < (\text{racine}, \text{distance}) ? \\ (3, 1) < (7, 0) \end{array} ?$$

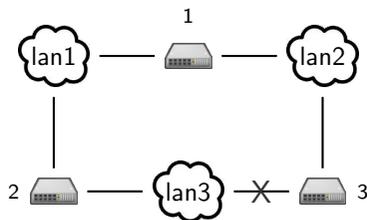
oui \Rightarrow la configuration de P2 change :

- ▶ $\text{racine} = r = 3$ et $\text{distance} = d + 1 = 1$
- ▶ $\text{portRacine} = \text{port de réception de la BPDU } (3, 0, 3, 1) = 1$

Situations de blocage (1/2)

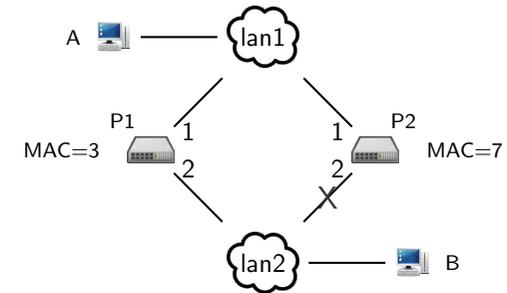
47/54

- ▶ 1^{ère} situation : 2 ponts sont à égale distance du pont racine et reliés à un même LAN.
- \Rightarrow C'est le pont ayant la plus grande adresse MAC qui bloque son port le menant à ce LAN.
- ▶ Exemple :



Exemple (4/4)

46/54



- ▶ P2 teste si il doit bloquer des ports.
La BPDU reçue sur son port 2 est $(r, d, e, p) = (3, 0, 3, 2)$.

$$\begin{array}{l} \text{est-ce que } r = \text{racine et } (d, e) < (\text{distance}, \text{moi}) ? \\ 3 = 3 \quad \text{et } (0, 3) < (1, 7) \end{array} ?$$

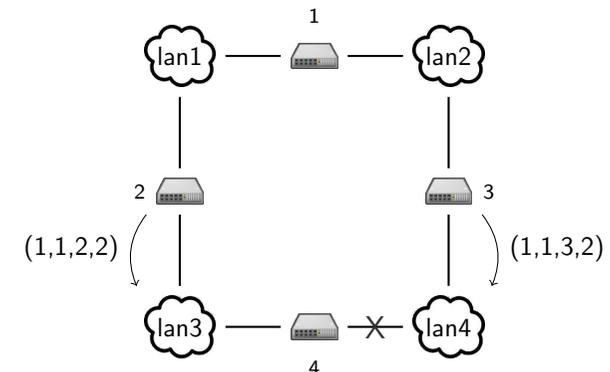
oui \Rightarrow P2 bloque le port de réception de la BPDU $(3, 0, 3, 2)$: le port 2

- ▶ À l'étape suivante, les ponts s'échangent des BPDUs mais leurs configurations ne changent pas.
- \Rightarrow Le protocole termine.

Situations de blocage (2/2)

48/54

- ▶ 2^{ème} situation : un pont a 2 chemins de même longueur le menant au pont racine
- \Rightarrow Il bloque le port sur lequel il reçoit des BPDUs émises par le pont avec la plus grande adresse MAC.
- ▶ Exemple :



Introduction

Les ponts Ethernet

Stratégies de commutation

Construction d'un arbre couvrant

Introduction aux réseaux locaux virtuels

Principe des réseaux locaux virtuels

51/54

regrouper les machines sur une base logique plutôt que physique

- ▶ On a un ensemble de réseaux locaux virtuels (ou VLANs, Virtual LANs).
 - ▶ Chaque VLAN a un **identifiant**.
 - ▶ Une machine est rattachée à un VLAN.
 - ▶ Certaines machines peuvent ne pas avoir été rattachées à un VLAN.
 - ⇒ Elles sont rattachées au VLAN par défaut (d'identifiant 1, généralement).
 - ▶ La décomposition en VLANs implique une **partition** du réseau :
 - ▶ Une machine ne voit plus que les machines du même VLAN.
 - ▶ Le découpage en VLANs est à la charge de l'administrateur du réseau.
 - ▶ Ce découpage logique se superpose à l'organisation physique.
 - ▶ Exemple de découpage : attribuer le VLAN1 au département Réseaux, le VLAN2 au département Informatique, ...
- ⇒ moyen économique d'avoir plusieurs LANs sur une même architecture physique

Limite des ponts et commutateurs

50/54

- ▶ diffusion/broadcast coûteux (trames envoyées vers FF:FF:FF:FF:FF:FF)
 - ▶ Toutes les machines et ponts/commutateurs de tous les LANs recevront la trame.
 - ⇒ inadapté pour les protocoles de plus haut niveau qui font beaucoup de diffusion (ex : ARP)
- ▶ découpage physique plutôt que logique
 - ▶ Les machines qui communiquent fréquemment doivent être placées sur le même LAN pour éviter d'encombrer les autres liaisons.
 - ▶ La structuration physique doit refléter la structure organisationnelle.
 - ⇒ manque de souplesse
 - ⇒ limite les déplacements de machines (⇒ inadapté pour les portables)
- ▶ sécurité limitée
 - ▶ Les machines peuvent voir les échanges de données privées circulant entre 2 machines situés sur des LANs différents.

Principe des réseaux locaux virtuels

51/54

Mise en œuvre des réseaux locaux virtuels

52/54

- ▶ Sur un réseau avec VLANs, les trames échangées ont une **étiquette**.
- ▶ Cette étiquette détermine le VLAN sur lequel circule la trame.
- ▶ L'étiquette est définie dans la norme IEEE 802.1Q.

Structure de la trame Ethernet IEEE 802.3 (hors préambule et SFD)

DA	SA	DL/EType	Données	Bourrage	FCS
6 o.	6 o.	2 o.	0–N o.	0–46 o.	4 o.

Structure de la trame Ethernet étiquetée IEEE 802.1Q

DA	SA	Id de prot.	Pri.	CFI	VLAN-id	DL/EType	Données	Bourrage	FCS
6 o.	6 o.	2 o.	3 bits	1 bit	12 bits	2 o.	0–N o.	0–46 o.	4 o.

← étiquette de 4 octets →

étiquette = 4 champs insérés entre les champs SA et DL/EType :

- ▶ Id de prot. = 0x8000
 - ▶ mêmes taille et position que le champ DL/Etype dans une trame standard
- ▶ Pri. = niveau de priorité de la trame
- ▶ CFI = 0 (utilisé pour la compatibilité Ethernet/Token Ring)
- ▶ VLAN-id = id. du VLAN sur lequel circule la trame

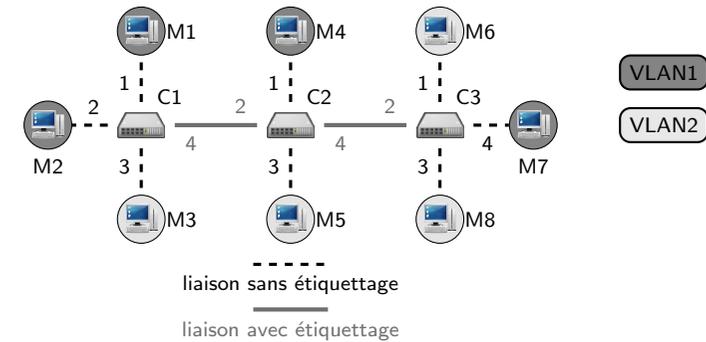
- ▶ On distingue deux types de liaison :
 - ▶ les liaisons sur lesquelles circulent des trames étiquetées ;
 - ▶ liaison commutateur ↔ commutateur
 - ▶ des liaisons sur lesquelles circulent des trames Ethernet standard.
 - ▶ liaison commutateur ↔ machine
- ▶ Un port d'un commutateur auquel est branché une machine est associé à un identifiant de VLAN.

Insertion de l'étiquette

- ▶ par un commutateur qui reçoit une trame d'une machine
- ▶ Le commutateur détermine le VLAN associé à ce port (ou le VLAN par défaut si aucun VLAN ne lui est associé).
- ▶ Il insère dans la trame l'étiquette contenant l'identifiant de ce VLAN puis retransmet sur le(s) port(s) adéquat(s).

Retrait de l'étiquette

- ▶ par un commutateur avant livraison au destinataire final



- ▶ M3 envoie une trame à M6.
 - ▶ C1 reçoit la trame. Il sait que son port 3 est associé au VLAN2.
- ⇒ Il rajoute à la trame une étiquette indiquant que la trame circule sur le VLAN2.
- ▶ La trame étiquetée va de C1 à C3.
 - ▶ Avant de la délivrer à M6, C3 retire l'étiquette.