

Répétitions du Cours d'Introduction aux Réseaux Informatiques

Protocoles de ponts

François Cantin

Département Montefiore
Research Unit in Networking
Université de Liège

Année académique 2008–2009

Contact

- Bureau : Institut Montefiore (B28), I77a.
- Email : cantin@run.montefiore.ulg.ac.be.
- Téléphone : 04 366 26 94.

Table des matières

- 1 Protocoles de ponts
 - Principes de base
 - Fonctionnement des ponts transparents
 - Exercices
 - Synthèse sur l'état des ports

Table des matières

1 Protocoles de ponts

- Principes de base
- Fonctionnement des ponts transparents
- Exercices
- Synthèse sur l'état des ports

Emploi des ponts

Les **ponts (bridge)** sont utilisés pour interconnecter des LANs. Ils permettent :

- D'augmenter les performances du réseau en filtrant l'information.
- D'augmenter la sécurité du réseau en filtrant l'information.
- D'interconnecter des réseaux de types différents (par exemple, de l'ethernet avec du wi-fi).

Les **commutateurs (switch)** étudiés au cours théorique sont des ponts dont tous les ports utilisent le même format de trame (ils ne permettent pas d'interconnecter des réseaux de types différents).

Tout ce qui sera vu dans cette répétition au sujet des ponts s'applique également aux commutateurs.

Les ponts transparents

Les ponts sont dits **transparents**. Cela signifie que l'émetteur et le récepteur ne voient pas le pont et n'ont pas à s'en soucier : tout se passe comme si le destinataire était directement connecté au réseau local de l'émetteur.

Table des matières

- 1 Protocoles de ponts
 - Principes de base
 - **Fonctionnement des ponts transparents**
 - Exercices
 - Synthèse sur l'état des ports

La table d'acheminement

Définition

Le pont reçoit toutes les trames circulants sur les LANs auxquels il est connecté (mode *promiscuous*).

Quand une trame arrive sur un port d'entrée, le pont met à jour sa **table d'acheminement**. Cette table associe à chaque port un ensemble d'adresses MAC. Le filtrage des trames est basé sur cette table.

Forme des entrées de la table d'acheminement :

(Adresse MAC, Identificateur du port, Date de création)

Clé : Adresse MAC

La table d'acheminement

Mise à jour

Supposons qu'une trame d'adresse source *srcAdr* arrive sur port d'identificateur *portId* à l'instant *time*. A sa réception,

- S'il existe déjà une association entre le port d'arrivée et l'adresse source de la trame, la date de création de cette association est mise à jour.
- S'il existe une association entre cette adresse MAC et un autre port, elle est remplacée par (*srcAdr*, *portId*, *time*).
- S'il n'existe aucune association pour cette adresse MAC, l'entrée (*SrcAdr*, *portId*, *time*) est ajoutée.

La durée de vie des associations est limitée ce qui permet à l'algorithme de s'adapter à des changements de topologie.

Acheminement des trames

Le **filtrage** des trames s'opère de la façon suivante :

- Si le destinataire de la trame est associé au port d'entrée, le pont ignore la trame.
- Si le destinataire est associé à un port de sortie $\neq$ port d'entrée, le pont transmet la trame sur le port de sortie.
- Si le destinataire n'est pas connu, le pont transmet la trame sur tous ses ports sauf le port d'entrée.

Danger : boucles !

Solution : désactiver (bloquer) un ensemble de ports

Prévention des boucles

Pour éviter la création de boucles, on utilise l'algorithme de **Spanning Tree**, qui permet de voir la topologie du réseau comme un arbre (chemin unique entre deux noeuds). On réalise cela en désactivant un ensemble de ports dans chaque pont.

Tout ajout ou suppression d'un pont entraînera la construction d'un nouvel arbre.

Lorsque l'algorithme de Spanning Tree est stabilisé, un état est associé à chaque port. Les états possibles sont "Forwarding", "Root" et "Blocking". Un port dans l'état *Blocking* ne retransmet aucune trame de données. Les autres ports retransmettent les trames de données suivant les règles de filtrage.

L'algorithme de Spanning Tree

Terminologie (1)

- Pont racine (root bridge)** Le pont dont l'identificateur est le plus petit.
- Coût (path cost)** A chaque port d'un pont, on associe un coût de transmission d'une trame sur le LAN connecté à ce port. Il est alors possible d'associer un coût à chaque chemin reliant deux ponts.
- Port racine (root port)** Le port qui mène au root bridge par le chemin de coût le plus faible. Si deux ports ont le même coût (pour atteindre le root bridge), on choisit le port dont le numéro est le plus petit.
- Coût du chemin racine (root path cost)** Pour chaque pont, le coût du chemin au coût minimum qui relie le pont au root bridge (donc en passant par le root port du pont).

L'algorithme de Spanning Tree

Terminologie (2)

Pont désigné (designated bridge) Sur chaque LAN, un seul pont est le pont désigné. C'est le pont qui fournit le chemin de coût minimum vers la racine. C'est le seul pont qui peut transmettre des trames du LAN vers le pont racine. Le pont racine est le pont désigné de tous les LANs auxquels il est attaché.

Port désigné (designated port) Le port qui relie le pont désigné au LAN est le port désigné.

L'algorithme de Spanning Tree

Messages de contrôle

Les ponts s'échangent des messages de contrôle appelés BPDUs (*Bridge Protocol Data Units*). Un BPDU a la forme suivante :

$\langle \text{Root id.}, \text{Root path cost}, \text{Sender id.} \rangle$

où

- *Sender id.* est l'identificateur de l'émetteur du BPDU ($\neq$ de son adresse MAC).
- *Root id.* est l'identificateur du pont qu'il considère comme étant le pont racine.
- *Root path cost* est le coût de son chemin racine.

L'algorithme de Spanning Tree

Classification des messages de contrôle

On définit une relation d'ordre entre les BPDUs de la manière suivante. Soient deux BPDUs C1 et C2, de contenus respectifs :

$$\langle rootId_1, cost_1, senderId_1 \rangle \text{ et } \langle rootId_2, cost_2, senderId_2 \rangle$$

C1 est meilleur que C2

- 1 lorsque $rootId_1 < rootId_2$.
- 2 si les identificateurs de pont racine sont égaux, lorsque $cost_1 < cost_2$.
- 3 si les coûts aussi sont égaux, lorsque $senderId_1 < senderId_2$.

L'algorithme de Spanning Tree

Table

La table d'un pont contient une entrée par port. A tout moment, une entrée relative à un port indique l'état de ce port ainsi que le meilleur BPDU parmi :

- le BPDU qu'il peut envoyer sur ce port
- les BPDUs qu'il reçoit sur ce port

Chaque entrée donc a la forme suivante :

[Port id., Root id., Root path cost, Sender id., Port state]

où *Port state* $\in \{RP, FP, BP\}$

Attention : Il ne faut pas confondre la table d'acheminement et la table relative au Spanning Tree. La signification du terme "table" dépendra du contexte dans lequel il est employé.

L'algorithme de Spanning Tree

Initialisation

Initialement, chaque pont se considère comme étant le root bridge.

Tous les ports de tous les ponts sont dans l'état *Forwarding* (FP).

Chaque pont envoie un BPDU $\langle \textit{bridged}, 0, \textit{bridged} \rangle$ sur chacun de ses ports où *bridged* est son propre identificateur.

L'algorithme de Spanning Tree

Mise à jour de la table - Port de réception

Lorsqu'un pont reçoit un BPDU $\langle rootId, cost, senderId \rangle$ sur un port P , il compare le BPDU reçu avec l'entrée E_P de sa table relative à P .

- Si E_P est meilleure ou équivalente, il la conserve et le processus de mise à jour est terminé.
- Si E_P est moins bonne que le BPDU reçu, elle est mise à jour avec les informations contenues dans ce BPDU et elle est ensuite comparée aux autres entrées de la table.

L'algorithme de Spanning Tree

Mise à jour de la table - Autres ports

Supposons que l'entrée $E_{P'}$ relative au port P' dans la table soit constituée des valeurs $rootld'$, $cost'$ et $senderld'$. Si

- $rootld < rootld'$,
- ou $rootld = rootld'$ et $cost + 1 < cost'$,
- ou $rootld = rootld'$ et $cost + 1 = cost'$ et $bridgeld < senderld'$,

alors $E_{P'}$ est mise à jour avec les informations $rootld$, $cost+1$ et **$bridgeld$** .

L'algorithme de Spanning Tree

Mise à jour de la table - Etat des ports

Lorsque les champs *Root id.*, *Root path cost* et *Sender id.* ont été mis à jour, l'état des ports est déterminé de la manière suivante :

- Le port pour lequel ces champs forment le meilleur BPDU est le port racine. Si plusieurs entrées sont équivalentes, c'est le port de plus petit identificateur qui est élu port racine. On lui attribue l'état *Root Port* (RP).
- Les ports dont le champ *Sender id.* contient la valeur *bridgeld* passent dans l'état *Forwarding* (FP). Ce sont des ports désignés. Ils forment un BPDU et l'envoient sur le LAN auquel ils sont attachés.
- Les autres ports passent dans l'état *Blocking* (BP).

L'algorithme de Spanning Tree

Détection des changements de topologie

Le root bridge envoie régulièrement des BPDUs sur ses ports pour signaler sa présence aux autres ponts.

Les ports de type *Forwarding* les retransmettent.

Un timer est associé à tout port de type *Root Port* ou *Blocking*. Le timer est relancé à la réception d'un BPDUs. Si le timer parvient à expiration, un changement de topologie est détecté. La table est recalculée en respect avec les règles décrites précédemment.

Table des matières

- 1 Protocoles de ponts
 - Principes de base
 - Fonctionnement des ponts transparents
 - **Exercices**
 - Synthèse sur l'état des ports

Exercice 1

Objectif : évolution de la table suite à la réception de divers BPDUs

Considérons un pont (host ID = 70) ayant 4 ports. Un port i donne accès au LAN i . Le pont vient de se mettre en service et n'a donc aucune connaissance du spanning tree. Considérons le scénario suivant :

- 1 Il reçoit le BPDU $\langle 1, 12, 3 \rangle$ sur le port 1.
- 2 Il reçoit le BPDU $\langle 1, 13, 56 \rangle$ sur le port 4.
- 3 Il reçoit le BPDU $\langle 1, 10, 5 \rangle$ sur le port 2.
- 4 Il reçoit le BPDU $\langle 1, 10, 45 \rangle$ sur le port 4.

Donner la table de spanning tree initiale et l'évolution de cette table à chaque étape du scénario. A la fin du scénario, donner la connaissance du spanning tree qu'a le pont 70.

Résolution exercice 1

Initialisation : Il construit sa table et transmet le BPDU $\langle 70, 0, 70 \rangle$ sur ses 4 ports.

table initiale

port	root ID	cost	sender ID	type
1	70	0	70	FP
2	70	0	70	FP
3	70	0	70	FP
4	70	0	70	FP

Résolution exercice 1 (2)

Il reçoit le BPDU $\langle 1, 12, 3 \rangle$ sur le port 1. Ce BPDU est meilleur que les BPDUs qu'il pourrait envoyer en se basant sur le contenu de sa table. Donc, il actualise sa table et transmet ensuite le BPDU incrémenté sur ses ports FP.

nouvelle table

port	root ID	cost	sender ID	type
1	1	12	3	RP
2	1	13	70	FP
3	1	13	70	FP
4	1	13	70	FP

Résolution exercice 1 (3)

Il reçoit le BPDU $\langle 1, 13, 56 \rangle$ sur le port 4. Ce BPDU est meilleur que le BPDU qu'il pourrait envoyer sur son port 4 mais pas meilleur que les BPDUs qu'il pourrait envoyer sur ses autres ports. En effet $\langle 1, 13 + 1, 70 \rangle$ est moins bon que $\langle 1, 13, 70 \rangle$, qui est lui-même moins bon que $\langle 1, 12, 3 \rangle$. Le port 4 devient bloquant car il existe un meilleur pont sur le LAN 4 mais pas meilleur que le root bridge actuel.

nouvelle table

port	root ID	cost	sender ID	type
1	1	12	3	RP
2	1	13	70	FP
3	1	13	70	FP
4	1	13	56	BP

Résolution exercice 1 (4)

Il reçoit le BPDU $\langle 1, 10, 5 \rangle$ sur le port 2. Comme ce BPDU est meilleur que $\langle 1, 12, 3 \rangle$, il modifie toutes ses entrées. Il transmet le BPDU $\langle 1, 11, 70 \rangle$ sur les ports 1, 3 et 4.

nouvelle table

port	root ID	cost	sender ID	type
1	1	11	70	FP
2	1	10	5	RP
3	1	11	70	FP
4	1	11	70	FP

Résolution exercice 1 (5)

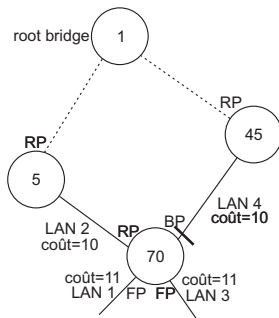
Il reçoit le BPDU $\langle 1, 10, 45 \rangle$ sur le port 4. Le BPDU est meilleur que le BPDU qu'il pourrait envoyer sur son port 4 mais pas meilleur que les BPDUs qu'il pourrait envoyer sur ses autres ports. En effet $\langle 1, 10 + 1, 70 \rangle$ est équivalent à $\langle 1, 11, 70 \rangle$ et est moins bon que $\langle 1, 10, 5 \rangle$. Le port 4 devient bloquant car il existe un meilleur pont sur le LAN 4 mais pas meilleur que le root bridge actuel.

nouvelle table

port	root ID	cost	sender ID	type
1	1	11	70	FP
2	1	10	5	RP
3	1	11	70	FP
4	1	10	45	BP

Résolution exercice 1 (6)

S'il ne reçoit plus de BPDU susceptibles de modifier sa table (algorithme stabilisé), il pourra transmettre des trames qui proviennent ou sont destinées aux ports 1, 2 et 3. Le pont 70 a la connaissance suivante du spanning tree :



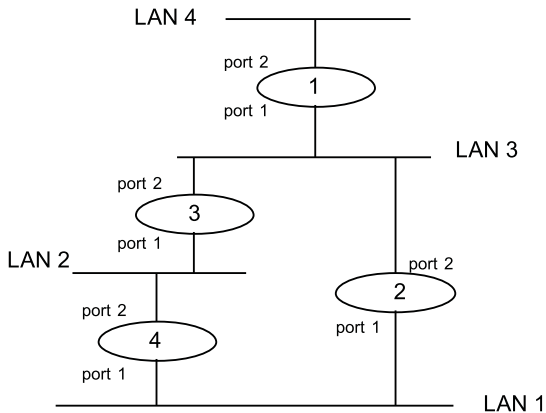
Exercice 2

Objectifs : état d'un ST stable, évolution des tables suite à un changement de topologie

Considérons le réseau de la figure sur le transparent suivant. On suppose au départ que le pont 3 n'est pas opérationnel. On demande de :

- ➊ Calculer la table de chaque pont si le spanning tree est stable.
- ➋ Donner le scénario possible d'échange de BPDU si l'on met en service le pont 3. Soyez précis en indiquant l'évolution de la table de chaque pont.
- ➌ Indiquer les BPDUs qui sont échangés quand tout est stabilisé.

Exercice 2 (2)



Résolution exercice 2

1 Le root bridge est le pont 1 (il a le plus petit identificateur).
Chaque pont est utile au spanning tree, ils sont donc tous actifs.
Les tables suivantes peuvent être facilement obtenues à partir de la situation initiale :

pont 1

port	root ID	cost	sender ID	type
1	1	0	1	FP
2	1	0	1	FP

pont 2

port	root ID	cost	sender ID	type
1	1	1	2	FP
2	1	0	1	RP

pont 4

port	root ID	cost	sender ID	type
1	1	1	2	RP
2	1	2	4	FP

Résolution exercice 2 (2)

2 Le pont 3 est mis en service. Il suppose qu'il est le root bridge et émet des BPDUs $\langle 3, 0, 3 \rangle$ sur tous ses ports.

pont 3

port	root ID	cost	sender ID	type
1	3	0	3	FP
2	3	0	3	FP

Les ponts 1, 2 et 4 vont recevoir le BPDU mais ils ne vont pas le retransmettre car le root ID de ce BPDU est plus grand que le root ID (=1) de ces ponts. Le pont 3 va finir par recevoir un BPDU $\langle 1, 0, 1 \rangle$ sur le port 2. Il va donc modifier sa table.

pont 3

port	root ID	cost	sender ID	type
1	1	1	3	FP
2	1	0	1	RP

Résolution exercice 2 (3)

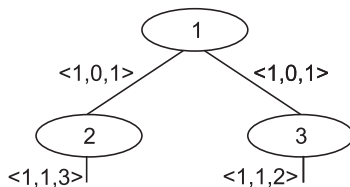
Après avoir modifié sa table, le pont 3 retransmet le BPDU incrémenté sur le port 1 et le pont 4 reçoit $\langle 1, 1, 3 \rangle$ sur son port 2. Il va modifier sa table car $\langle 1, 1, 3 \rangle$ est meilleur que $\langle 1, 2, 4 \rangle$. Il en déduit qu'il n'est plus désigné sur le LAN 2 et bloque le transfert de données sur ce port.

pont 4	port	root ID	cost	sender ID	type
	1	1	1	2	RP
	2	1	1	3	BP

Le pont 4 va recevoir périodiquement le BPDU $\langle 1, 1, 2 \rangle$ sur le port 1 mais celui-ci ne modifiera pas sa table. L'algorithme est maintenant stabilisé. Le pont 4 ne fait plus partie du spanning tree.

Résolution exercice 2 (4)

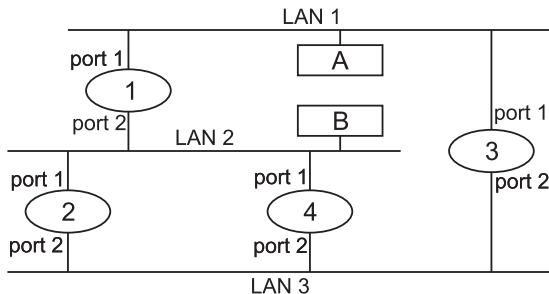
3 Les BPDUs $\langle 1, 1, 3 \rangle$ et $\langle 1, 1, 2 \rangle$ ne seront pas retransmis par le pont 4.



Exercice 3 (examen août 1998)

Objectifs : filtrage des trames, mise à jour des tables d'acheminement, emploi du protocole ARP, ponts désignés

Considérons le réseau suivant :



Exercice 3 (2)

- 1 Quels sont les BPDUs échangés par les ponts si le réseau est stabilisé ? Donnez la table de chaque pont.
- 2 On suppose que *B* n'a plus transmis de trame sur son réseau local depuis plus d'une heure. Que se passe-t-il dans le réseau et au niveau des ponts si la station *A* émet une trame à destination de *B* pour la première fois (*A* ne connaît pas l'adresse MAC de *B*) ? Donnez les adresses source et destination de chaque trame envoyée et détaillez l'évolution des tables d'acheminement des ponts.
- 3 Donnez un scénario possible d'échanges de BPDUs si le pont 2 sort subitement du réseau et si le pont 4 est le premier pont à remarquer l'absence du pont 2. Indiquez aussi le pont désigné pour chaque segment.

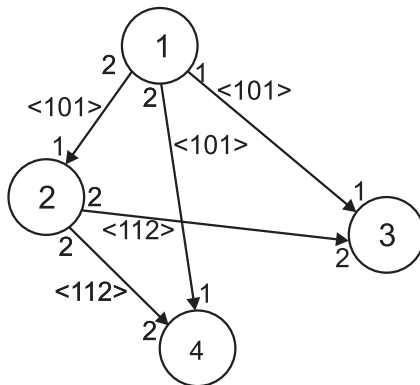
Résolution exercice 3

1 La table de spanning tree stabilisée peut être rapidement obtenue à partir de la situation initiale :

pont	port	root ID	cost	sender ID	type
pont 1	1	1	0	1	FP
	2	1	0	1	FP
pont 2	1	1	0	1	RP
	2	1	1	2	FP
pont 3	1	1	0	1	RP
	2	1	1	2	BP
pont 4	1	1	0	1	RP
	2	1	1	2	BP

Résolution exercice 3 (2)

Les BPDUs de contrôle échangés sont :



Résolution exercice 3 (3)

2 A va se servir de ARP pour découvrir l'adresse MAC de B à partir de son adresse IP.

Une **requête ARP** est de type "Quelle est l'adresse MAC qui correspond à l'adresse *adresseIP* ? Répondez à *adresseMAC*". Cette requête est envoyé en broadcast sur le réseau local. Tous les ponts du réseau local vont propager cette requête sur tous leurs ports FP et RP (sauf celui sur lequel la requête est arrivée) et mettre à jour leur table d'acheminement.

Quand la station qui a l'adresse *adresseIP* reçoit la requête, elle y répond avec une trame directement adressée à l'émetteur de la requête. La réponse va suivre le chemin prévu par le spanning tree pour rejoindre l'émetteur et tous les ponts situés sur ce chemin mettent à jour leur table d'acheminement.

Résolution exercice 3 (4)

A émet donc une trame contenant une requête ARP portant sur l'adresse IP_B .

Adresse source : MAC_A

Adresse destination : *Broadcast* (FF-FF-FF-FF-FF-FF)

Le pont 3 reçoit cette trame sur son port 1. Il met à jour sa table d'acheminement en associant MAC_A à son port 1. Il ne propage pas la requête sur son port 2 car il est BP.

Le pont 1 reçoit également cette trame sur son port 1. Il associe MAC_A à son port 1. Il propage la trame sur son port 2.

Le pont 2 reçoit la trame propagée sur son port 1. Il associe MAC_A à son port 1 et il propage la trame sur son port 2.

Le pont 4 reçoit aussi la trame sur son port 1. Il associe MAC_A à son port 1. Il ne propage pas la requête sur son port 2 car il est BP.

Résolution exercice 3 (5)

B reçoit la trame propagée par le pont 1. Il constate que la requête porte sur son adresse IP et il y répond en émettant une trame à destination de A .

Adresse source : MAC_B
Adresse destination : MAC_A

Le pont 4 reçoit cette trame sur son port 1. Il associe MAC_B à son port 1 et ne la propage pas.

Le pont 2 reçoit cette trame sur son port 1. Il associe MAC_B à son port 1 et ne la propage pas car MAC_A est associée à son port 1.

Le pont 1 reçoit la trame sur son port 2. Il associe MAC_B à son port 2 et propage la trame sur son port 1.

Le pont 3 reçoit la trame propagée sur son port 1. Il associe MAC_B à son port 1 et ne propage pas la trame.

Résolution exercice 3 (6)

A reçoit la réponse à la requête et il peut envoyer sa trame à B.

Adresse source : MAC_A

Adresse destination : MAC_B

Le pont 1 et le pont 3 reçoivent cette trame sur le port 1. Ils rafraîchissent l'entrée pour MAC_A dans leur table d'acheminement et le pont 1 propage la trame sur son port 2.

Les ponts 2 et 4 reçoivent la trame propagée sur leur port 1. Ils rafraîchissent l'entrée pour MAC_A dans leur table d'acheminement et ils ne propagent pas la trame.

La station B reçoit la trame.

Attention : quand un pont ne faisant pas partie du spanning tree (ponts 3 et 4) reçoit une trame, il met quand même à jour sa table d'adresses.

Résolution exercice 3 (5)

3 Un échange possible de BPDUs est le suivant :

- Au niveau du pont 4, l'entrée de la table concernant le port 2 expire car le pont 2 ne transmet plus son BPDU $\langle 1, 1, 2 \rangle$.
Quand le pont 4 reçoit $\langle 1, 0, 1 \rangle$, il transmet $\langle 1, 1, 4 \rangle$ sur le port 2. Si rien ne bouge, son port 2 va passer d'un état bloquant à un état actif (FP) : avant de faire repasser le port 2 à l'état actif, on attend un peu pour voir si aucun pont ne propose un meilleur BPDU.
- Le pont 3 reçoit $\langle 1, 1, 4 \rangle$ et ne le retransmet pas sur le port 1 car il est moins bon que $\langle 1, 0, 1 \rangle$. Le pont 3 réagit comme le pont 4 et il transmet $\langle 1, 1, 3 \rangle$ sur le port 2 quand il reçoit $\langle 1, 0, 1 \rangle$ sur son port 1.

Résolution exercice 3 (6)

- Le pont 4 va modifier sa table car $\langle 1, 1, 3 \rangle$ est meilleur que $\langle 1, 1, 4 \rangle$. Il ne retransmet pas le BPDU sur le port 1 car il est moins bon que $\langle 1, 0, 1 \rangle$.
- Après un certain temps, le port 2 du pont 3 passe d'un état bloquant à un état actif. Le port 2 du pont 4 reste dans l'état bloquant.

Le pont 1 est le pont désigné pour le LAN 1 et le LAN 2. Le pont 3 est le pont désigné pour le LAN 3.

Table des matières

- 1 Protocoles de ponts
 - Principes de base
 - Fonctionnement des ponts transparents
 - Exercices
 - Synthèse sur l'état des ports

Synthèse sur l'état des ports (1)

Forwarding Port

- ❶ Le pont envoie régulièrement un BPDU sur ce port.
- ❷ Le pont est désigné sur le LAN auquel ce port donne accès.
- ❸ Le pont enregistre les adresses source des trames qui arrivent sur ce port.
- ❹ Le pont retransmet sur ce port les trames envoyées en broadcast.
- ❺ Le pont retransmet sur ce port les trames dont l'adresse de destination lui est inconnue.

Synthèse sur l'état des ports (2)

Root Port

- ❶ Ce port donne accès au chemin le plus court vers le pont racine.
- ❷ Il y a en a un et un seul par pont.¹
- ❸ Le pont écoute les BPDUs envoyés sur le LAN correspondant au port.
- ❹ Le pont n'envoie pas de BPDU via ce port.
- ❺ Le pont enregistre les adresses source des trames qui arrivent sur ce port.
- ❻ Le pont retransmet sur ce port les trames envoyées en broadcast.
- ❼ Le pont retransmet sur ce port les trames dont l'adresse de destination est associée avec ce port dans la table d'acheminement.
- ❽ Le pont retransmet sur ce port les trames dont l'adresse de destination lui est inconnue.

¹Sauf pour le root bridge (tous ses ports sont FP)

Synthèse sur l'état des ports (3)

Blocking Port

- 1 le pont écoute les BPDUs envoyés sur le LAN correspondant au port.
- 2 le pont n'envoie pas de BPDU via ce port.
- 3 le pont n'est pas désigné pour ce LAN.
- 4 le pont ignore les trames autres que les BPDUs en transit sur le LAN correspondant.
- 5 le pont ne retransmet aucune trame via ce port.