

Le protocole 802.1d

Propriétés	Description
Intitulé long	Rôle et fonctionnement du protocole 802.1d
Date de publication	21-09-2004
Présentation	A partir d'une suite d'exercices s'appuyant sur différentes architectures composées de liaisons redondantes, on expose la nécessité d'un protocole prenant en charge cette redondance et son fonctionnement.
Formation concernée	BTS IG option ARLE
Public concerné	BTS IG option ARLE 2eme année
Matière	AMSI
Compétences	Assurer les fonctions de l'exploitation - Surveiller et optimiser le trafic sur un réseau
Savoirs	Techniques de routages et de commutation
Objectifs	Comprendre l'algorithme de « spanning tree »
Mots-clés	Ponts Commutateurs Spanning tree 802.1d
Auteur(es)	Roger Sanchez

Progression proposée

Cette suite d'exercice a pour objectif de présenter la nécessité et le fonctionnement du protocole 802.1d (spanning tree) mis en œuvre par les ponts et les commutateurs pour gérer les liaisons redondantes.

La nécessité du protocole sera exposée lors du premier exercice.

Le deuxième exercice présentera l'objectif d'un algorithme « spanning tree ».

Le troisième exercice montrera l'application d'un algorithme simplifié « spanning tree » sur un réseau comportant des liaisons redondantes gérées par des ponts puis l'application de l'algorithme utilisé par la norme 802.1d.

Le quatrième exercice propose une application de ce protocole sur un réseau comportant des liaisons redondantes gérées par des commutateurs.

Présentation du sujet

En tant que jeune assistant de l'administrateur du réseau un jeune titulaire du BTS vient d'être embauché dans une entreprise disposant d'un réseau Ethernet important (environ 600 postes) construit sur plusieurs années et juxtaposant différentes technologies.

L'administrateur réseau lui demande de se familiariser avec l'architecture du réseau.

En étudiant les différents schémas, il découvre des architectures particulières qui l'amènent à étudier les protocoles mis en œuvre par les matériels d'interconnexion.

Remarque : le nombre de PCs sur les schémas n'est pas significatif.

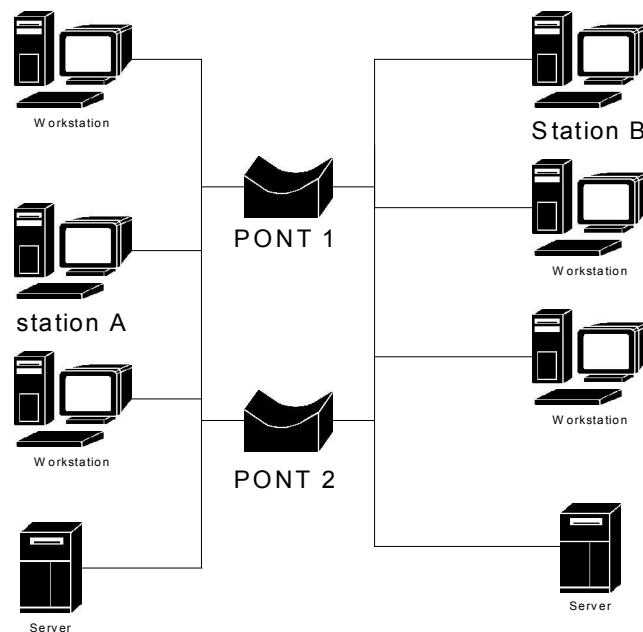
Première partie : les ponts redondants

Deux réseaux en 10 base2 sont reliés par deux ponts.

Un pont est un matériel d'interconnexion qui relie plusieurs segments Ethernet. Pour chacun des segments auquel il est connecté il gère une table des adresses MAC composant ce segment. Ces tables sont construites à l'issue d'un processus d'apprentissage. A chaque fois qu'une carte émet une trame, le pont stocke l'adresse MAC de l'émetteur de la trame dans la table associée au segment.

Ces tables vont ensuite lui permettre d'acheminer les trames. Lorsque le pont reçoit une trame, il relaye la trame vers l'autre segment dans trois cas :

- l'adresse du destinataire de la trame correspond à une adresse du segment
- il s'agit d'une adresse de diffusion (broadcast)
- l'adresse n'est pas connue par le pont



Travail à faire

Indépendamment de tout protocole particulier :

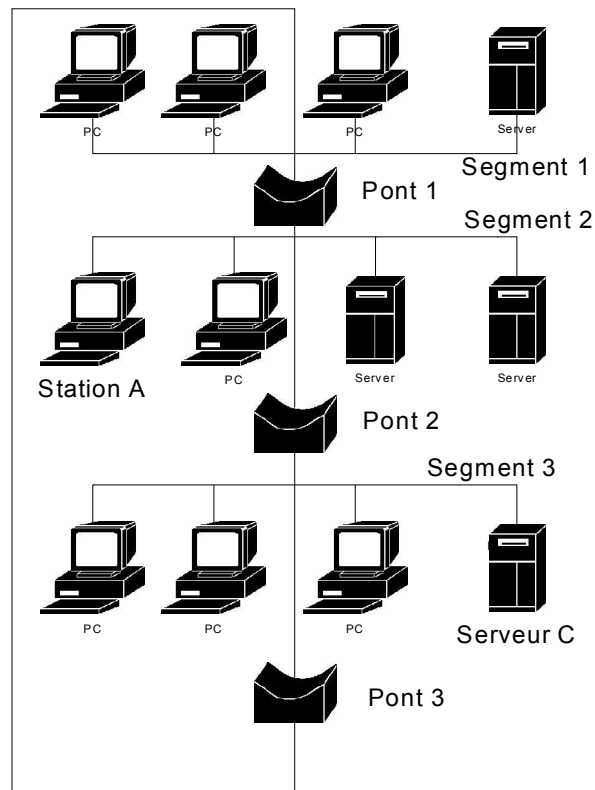
- Que se passe-t-il au niveau des trames échangées sur le réseau quand la station A émet une trame unicast vers la station B ?
- Que se passe-t-il quand une station émet une trame de broadcast ?

Le protocole 802.1d gérant la redondance a été mis en place

- Quel est le rôle de ce protocole ?

Deuxième partie : Les ponts cycliques

Trois autres services sont reliés entre eux par des ponts mais l'interconnexion des trois réseaux forme une boucle.



Travail à faire

Indépendamment de tout protocole particulier

A. Que se passe-t-il au niveau des trames échangées sur le réseau quand la station A émet une trame unicast vers le serveur C ?

A. Que se passe-t-il quand une station émet une trame de broadcast ?

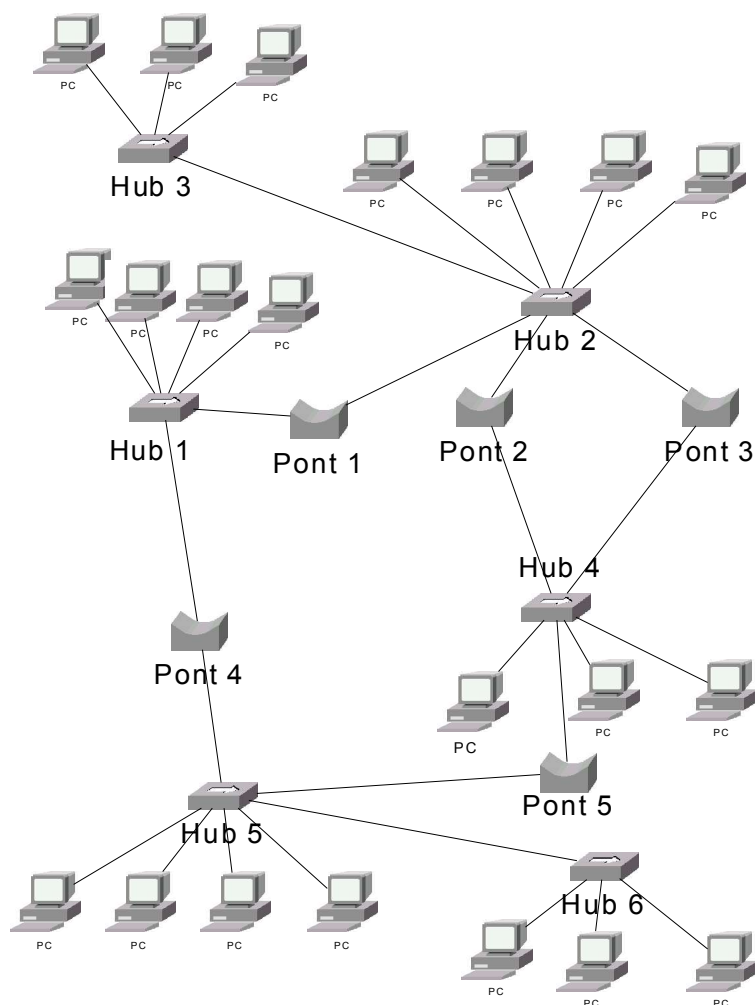
B. Dessiner le graphe correspondant à ce schéma. Chaque segment sera un nœud et chaque pont une branche.

Le protocole 802.1d gérant les graphes cycliques a été mis en place

C. Que doit faire ce protocole ?

Troisième partie : Construction d'un arbre à partir d'un graphe cyclique

Plusieurs services en 100baseT sont interconnectés soit par des hubs cascades (répéteur Classe 2) soit par des ponts. Les différentes liaisons forment un graphe cyclique.



Travail à faire :

Utiliser l'annexe 1

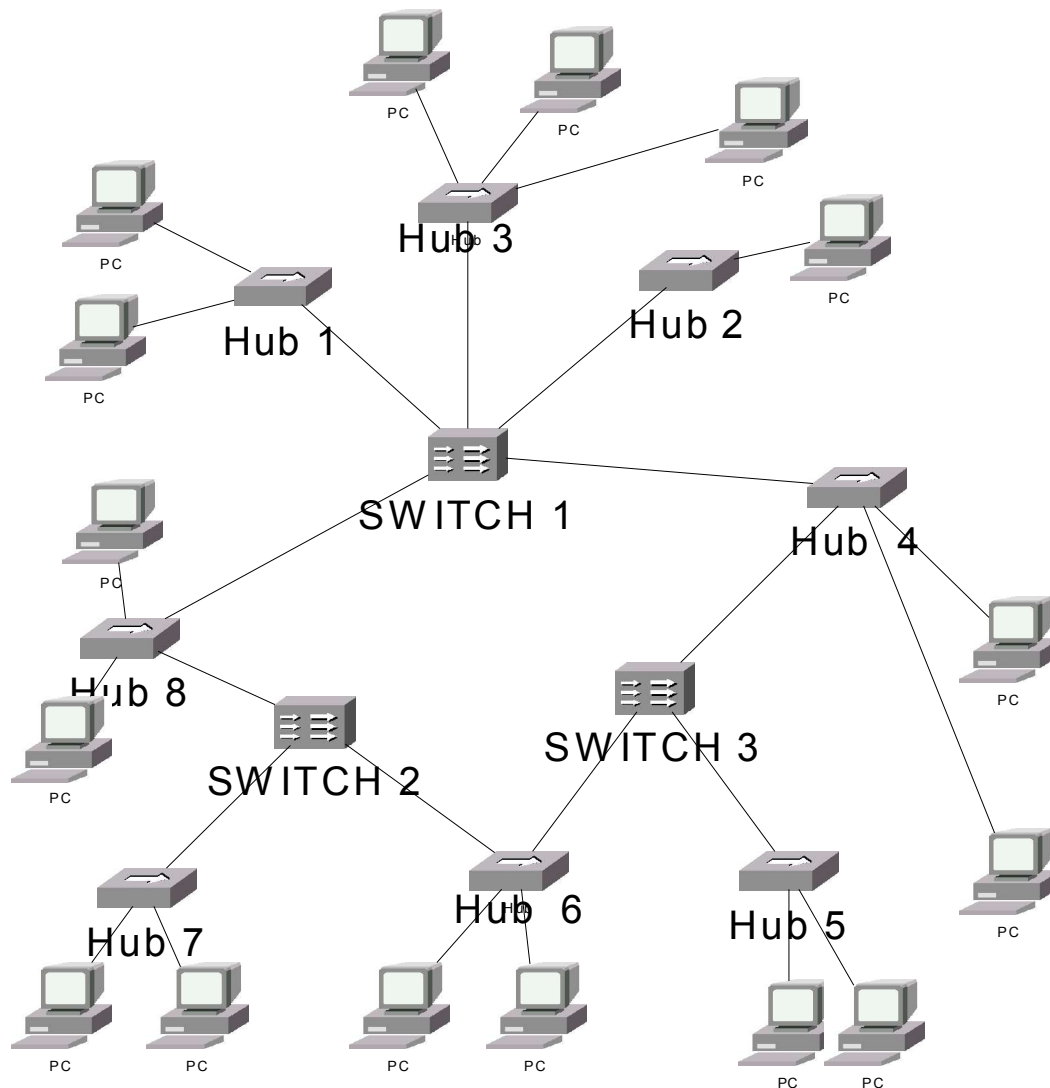
- Identifier les différents domaines de collisions et leur donner un nom (ex : segment 1 abréviation S1)
- Dessiner le graphe correspondant à ce schéma
- Quels ponts doit-on invalider pour transformer le graphe cyclique en arbre ? (plusieurs solutions possibles)
- Numéroter les ponts et les ports
- Déterminer le pont racine
- Affecter à chaque port un coût
- Déterminer **pour chaque pont** le port racine et le coût du chemin racine
- Déterminer pour chaque domaine de collision (segment) l'état des ports
- Noter sur le schéma les ports racines, les ports désignés, les ports bloqués.
- Dessiner le graphe correspondant au nouveau schéma

Utiliser l'annexe 2 (complément difficile)

- En utilisant l'annexe 2 donner la suite de trames échangées pour définir l'arbre (le coût de chaque port est fixé arbitrairement à 10)

Quatrième partie : Application à une architecture basée sur des commutateurs

Quelques services sont interconnectés par des commutateurs selon le schéma suivant.



Un commutateur peut être considéré comme un pont qui a plus de deux interfaces. On peut parler de micro segmentation ce qui consiste à avoir **un domaine de collision par port** si le commutateur est un commutateur *store and forward* (validation de trames : seules les trames valides sont réémises)

Le protocole "spanning tree" utilisé par les commutateurs est le même que pour les ponts.

Mais un commutateur a les caractéristiques suivantes :

- Un seul port racine par commutateur

- Zéro, un ou plusieurs ports désignés par commutateur.

- Zéro, un ou plusieurs ports bloqués par commutateur.

Travail à faire :

- déterminer les différents domaines de collision et les nommer
- numéroter les commutateurs (ex: SW1) et les ports des commutateurs (ex : SW1S1 et affecter un coût (10 par défaut)
- lister les trames échangées jusqu'à constitution de l'arbre de recouvrement (spanning tree)
- noter sur le schéma les ports racines, les ports désignés et les ports bloqués

Annexe 1 : Algorithme permettant de couper des circuits.

Pour éviter les chemins redondants et invalider des ponts on va transformer manuellement le graphe cyclique en arbre.

Pour cela on va utiliser l'algorithme suivant :

- ◆ Choisir un **pont racine**, puis numéroté les ponts et les ports des ponts

Pour faire un arbre il faut une racine. Pour choisir le pont racine, on donne à chaque pont un numéro unique. Le pont avec le numéro le plus petit sera le pont racine.

Dans le schéma précédent, si on considère que les numéros attribués aux ponts les identifient, le "**Pont 1**" sera le pont racine.

Pour numéroté les ports on fera ainsi, on construira le numéro à partir du numéro du pont et du numéro du segment relié au port. En appliquant cette règle, les numéros des ports du pont 1 seraient P1S1 et P1S2 (si on a nommé S1 et S2 les deux segments reliés au pont 1).

- ◆ Affecter un **coût** à chaque port (un coût est une valeur arbitraire mais qui sera défini par l'administrateur en fonction de la bande passante du réseau, du débit de la liaison, de sa qualité . etc.)

A chaque port on affecte un coût. Par exemple

P2S2 C = 10

P2S4 C = 10

P3S2 C = 10

P3S4 C = 10

- ◆ Déterminer le **port racine** de chaque pont et le **coût chemin racine** (CCR)

Chaque pont a deux ports, l'un des deux ports constitue un chemin plus court pour rejoindre le pont racine. Pour le déterminer, on additionne le coût de chaque port traversé pour rejoindre un des ports du pont racine à partir de chaque port. Le **port racine** est celui dont le total est le plus petit.

Exemple avec le pont 2, si on a choisit 10 comme coût à chaque port, le coût du chemin racine à partir de P2S2 est de 10 (on ne passe que par le port P2S2), et forcément supérieur à 10 à partir de P2S4 (qui doit traverser d'autres ponts).

Le pont racine est une exception, il a deux ports racines dont le coût est toujours égal à zéro.

Le **coût du chemin racine** pour un pont est le coût du chemin calculé avec son port racine (dans notre exemple pour le pont 2 le CCR est 10).

Le **CCR** pour le pont racine est toujours égal à zéro.

- ◆ Déterminer dans chaque segment l'**état** de chaque port des ponts

Un port est soit à l'état **actif** soit à l'état **bloqué**.

A l'état **actif** un port lit les trames sur le segment et transmet des trames sur ce segment.

A l'état **bloqué** un port lit les trames sur le segment mais ne transmet aucune trame.

Chaque pont a deux ports.

Le port racine est toujours actif car il est situé sur le segment permettant de rejoindre la racine.

Si le port n'est pas racine, il peut être bloqué. Pour qu'un port soit bloqué, il faut qu'il y ait sur le segment un port d'un autre pont dont le coût du chemin racine (CCR) est inférieur au sien ou en cas d'égalité dont l'identifiant du pont est inférieur.

Un port non racine et non bloqué est appelé **port désigné** (il a un rôle particulier dans le protocole réel).

Le calcul se fait ainsi :

On compare le CCR des deux ponts, le port ayant le CCR le plus petit devient le port désigné et les autres s'ils ne sont pas des ports racines sont bloqués. En cas d'égalité de CCR on compare l'identifiant des deux ponts, le port dont le pont a l'identifiant le plus petit devient le port désigné.

Le pont racine est une exception, ses deux ports sont toujours des **ports désignés**.

Annexe 2 : Le protocole 802.1d arbre de recouvrement (spanning tree) version simplifiée

Ce qu'on a fait manuellement doit être fait de façon dynamique par les ponts. Pour cela sur chaque pont est implanté le protocole spanning tree (SPT) normalisé par l'IEEE sous la norme 802.1d. Le protocole spanning tree doit supprimer les chemins redondants en construisant un arbre à partir d'un graphe cyclique. Ce protocole doit être implanté sur chaque pont. Pour le mettre en œuvre les ponts vont s'échanger des trames BPDU (Bridge Protocol Data Unit) de 34 octets. Ces trames sont adressées en multicast, seuls les ponts les liront.

- **Objectifs du protocole**

- Elire la racine de l'arbre
- calculer la distance à la racine
- déterminer les ports racines
- lorsque plusieurs ponts sont connectés au même segment, déterminer l'état de chaque port (actif ou bloqué) sur le segment
- déterminer le port désigné qui transmettra les messages 802.1d

- **Messages du protocole 802.1d**

- Le protocole 802.1d utilise des messages contenant l'information suivante
 - Root ID : identification actuelle de la racine
 - Cost : Coût du lien de plus faible coût entre le pont qui transmet le message et la racine
 - Transmitting ID : identification du pont qui transmet le message [identification unique sur 48 bits] L'identifiant est construit à partir de l'adresse MAC. C'est soit la meilleure des deux adresses MAC (la plus petite) soit l'adresse MAC fixé.
- Ces messages sont envoyés en multicast [adresse : all bridges]

- **Interprétation de deux messages 802.1d**

- $M1[R=R1, C=C1, T=T1]$ est meilleur que $M2[R=R2, C=C2, T=T2]$ si
 - $R1 < R2$
 - $R1=R2$ et $C1 < C2$
 - $R1=R2$ et $C1=C2$ et $T1 < T2$

- **Fonctionnement du protocole**

- Au démarrage, un pont se considère comme la racine et transmet un message 802.1d avec un coût de 0 sur tous ses ports
- Sur chaque port, le pont reçoit en permanence les messages 802.1d et sauvegarde pour chaque port le meilleur message
- Si un pont reçoit sur un port un meilleur message 802.1d que celui qu'il transmettrait, il arrête de transmettre son message 802.1
- Le protocole se stabilise lorsqu'un seul pont transmet des messages 802.1d sur chaque segment

- **Détermination de la racine**

- Dans un réseau la racine est le pont avec le plus petit identificateur
- Chaque pont peut déterminer quelle est la racine en analysant tous les messages 802.1d reçus

- **Port racine**

- le port racine d'un pont est le port qui se trouve topologiquement le plus proche de la racine. En fait le port qui a reçu le meilleur message 802.1d
- c'est via son port racine qu'un pont reçoit les messages 802.1d

- **Détermination de la distance à la racine (coût chemin racine CCR)**

- Si le pont est la racine, la distance est zéro par définition

- Sinon, la distance à la racine est le coût du port racine + le coût du meilleur message reçu sur le port "racine" du pont

Etats d'un port

- **Vis-à-vis des messages 802.1d**

- Port racine
 - port via lequel on reçoit les messages provenant de la racine par le plus court chemin
 - un port racine ne transmet pas de messages 802.1d mais les lit
 - un seul port racine par pont
- Port désigné
 - port via le(s)quel(s) le pont retransmet les messages 802.1d reçus de la racine
 - message 802.1d transmis par un pont
 - racine actuelle, identification du pont, coût jusque la racine
 - 0, ou un port désigné sur un pont
 - un port est désigné si le message qu'il transmet est meilleur que le meilleur message qu'il reçoit
- Port bloqué (reçoit uniquement les message. 802.1d)

- **Vis-à-vis des trames de données**

- port actif (port racine et port désigné)
 - le pont écoute les trames via ce port et les retransmet (sélectivement) vers les autres ports actifs du pont si nécessaire
- port inactif (port bloqué)
 - le pont n'écoute pas les trames de données via ce port et donc ne retransmet aucune trame de données reçue via ce port

- **Activité des ports**

- Initialement tous les ports sont inactifs
- les ports racines et désignés deviennent actifs lorsque la topologie du spanning tree (arbre de recouvrement) est stable

Proposition de corrigé

Première partie : Les ponts redondants

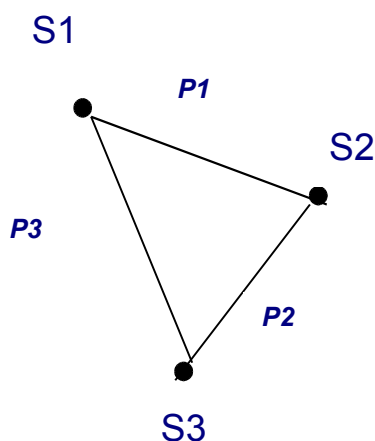
- A. La trame est retransmise par les deux ponts. La station B reçoit deux fois la même trame.
- B. La trame de broadcast est retransmise par les deux ponts à partir du premier segment. Deux trames de broadcast vont donc être émises sur le deuxième segment. Chaque pont va récupérer sur ce deuxième segment la trame de broadcast émise par l'autre pont et va la retransmettre de nouveau sur le premier segment. C'est une boucle sans fin.
- C. Le protocole doit invalider un des deux ponts.

On ne peut pas faire de la répartition de charge entre les deux ponts car ils reçoivent systématiquement les trames. Si on voulait faire cela il faudrait un élément intermédiaire qui distribue les trames aux deux ponts.

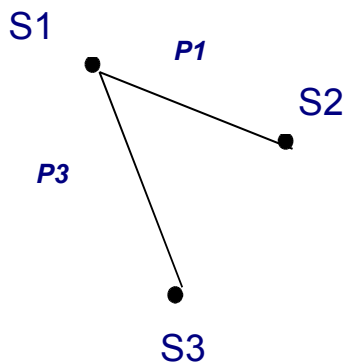
Le protocole doit permettre au pont invalidé de redevenir actif quand l'autre pont tombe en panne

Deuxième partie : Les ponts cycliques

- A. La trame est retransmise par le pont 1 puis par le pont 2 par un chemin, mais elle est aussi retransmise par le pont 3 sur un autre chemin. Le serveur reçoit deux fois la même trame.
- B. Comme avec les ponts redondants, la trame de broadcast est retransmise à l'infini par les différents ponts. En fait les ponts redondants sont un cas particulier de cycle. Dans les deux cas on a plusieurs chemins possibles qui forment forcément une boucle.
- C. Le graphe est cyclique.



- D. Le protocole doit invalider un des trois ponts, et donc transformer le graphe cyclique en arbre, en choisissant arbitrairement une racine. Dans le schéma suivant le pont 2 a été invalidé et le segment 1 est devenu racine d'un arbre à deux branches. :
En cas de défaillance d'un des deux autres ponts le protocole devra réactiver le pont invalidé.



Troisième partie : Construction d'un arbre à partir d'un graphe cyclique

Utilisation de l'annexe 1

A. Identifier les différents domaines de collision et les nommer

Chaque Hub non cascadié forme un domaine de collision.

Les hubs cascadiés forment un seul domaine de collision.

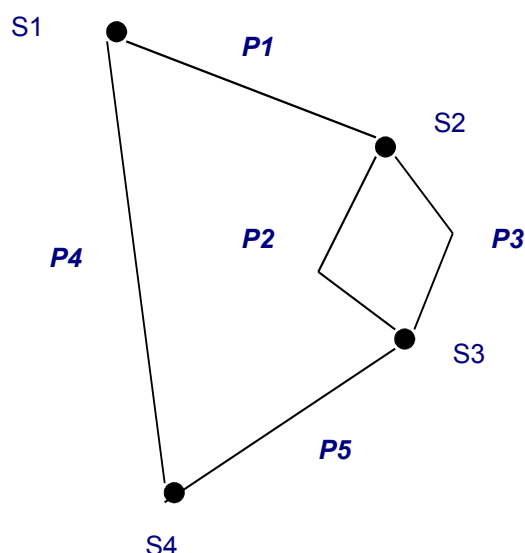
Il y a donc 4 domaines de collision.

Noms :

Hub1	segment 1	S1
Hub2 et Hub3	segment 2	S2
Hub4	segment 3	S3
Hub5 et Hub6	segment 4	S4

Remarque : en 100baseT on ne peut pas avoir plus de deux répéteurs de classe 2 entre deux stations. La distance entre la station et le répéteur ne doit pas excéder 100m et la distance entre les deux répéteurs ne doit pas excéder 5m.

B. Dessiner le graphe correspondant à ce schéma



C. Quels ponts doit-on invalider pour transformer le graphe cyclique en arbre ?

Il faut invalider deux ponts au total.

Sur le chemin de S2 à S3 il faut invalider soit P2 soit P3.

Sur le chemin de S1 à S4 il faut invalider soit P4 soit P5

D. Nommer les ponts et les ports

Ponts	Noms
Pont 1	P1
Pont 2	P2
Pont 3	P3
Pont 4	P4
Pont 5	P5

Ports	Noms	Noms
Ports de P1	P1S1	P1S2
Ports de P2	P2S2	P2S3
Ports de P3	P3S2	P3S3
Ports de P4	P4S1	P4S4
Ports de P5	P5S3	P5S4

E. Déterminer le pont racine

Le pont racine est P1 car il a l'identifiant le plus petit

F. Affecter à chaque port un coût

Ports	Noms	Coût	Noms	Coût
Ports de P1	P1S1	0	P1S2	0
Ports de P2	P2S2	10	P2S3	10
Ports de P3	P3S2	10	P3S3	10
Ports de P4	P4S1	10	P4S4	10
Ports de P5	P5S3	10	P5S4	10

En affectant des coûts différents on peut donner la priorité d'un chemin par rapport à un autre.

G. Déterminer pour chaque pont le port racine et le coût chemin racine

P1 est le pont racine, il a deux ports racines P1S1 et P1S2. Son CCR est égal à zéro.

Pour P2

Le CCR par P2S2 est de 10.

Le CCR par P2S3 est de 50 (coût cumulé de P2S3 P5S3 P5S4 P4S4 P4S1).

Le port racine est donc P2S2 et le CCR 10.

Pour P3 le port racine est P3S2 et le CCR 10

Pour P4 le port racine est P4S1 et le CCR 10

Pour P5 le calcul par P5S3 donne 30 (P5S3 + P2S3 + P2S2) et 30 aussi par P5S4 (P5S4 + P4S4 + P4S1). Les deux CCR sont équivalents; soit on modifie un coût soit le CCR sera le CCR du port avec l'identifiant le plus petit ici P5S3, dans ce cas le CCR est 30.

H. Déterminer pour chaque domaine de collision l'état des ports

Pour S1

On a P1S1 et P4S1

P1S1 est un port racine et désigné (pont racine). Il est actif.

P4S1 est un port racine. Il est actif.

Pour S2

On a P1S2 P2S2 et P3S2

P1S2 est un port racine et désigné (pont racine). Il est actif.

P2S2 est un port racine. Il est actif.

P2S3 est un port racine. Il est actif.

Pour S3

On a P2S3 P3S3 et P5S3

P5S3 est un port racine. Il est actif.

P2S3 et P3S3 ne sont pas des ports racines.

Les deux CCR sont égaux (encore une fois en jouant sur les coûts on aurait pu influencer le résultat)

L'identifiant P2S3 est inférieur à P3S3. P3S3 est bloqué, P2S3 est le port désigné actif.

Pour S4

On a P4S4 et P5S4, aucun des deux n'est un port racine.

Le CCR de P4 est 10.

Le CCR de P5 est 30.

P5S4 est bloqué. P4S4 est le port désigné actif.

Age Group	No opinion	Not a good idea	A good idea	A very good idea	Don't know
18-24	10%	10%	10%	10%	60%
25-34	10%	10%	10%	10%	60%
35-44	10%	10%	10%	10%	60%
45-54	10%	10%	10%	10%	60%
55-64	10%	10%	10%	10%	60%
65+	10%	10%	10%	10%	60%



<http://www.reseaucerta.org> © CERTA - Roger Sanchez - v1.0 septembre 2004 Page 1

A. Donner la suite de trames échangées

Pour déterminer le pont racine

Au démarrage (premier tour)

Sur S1

Message de P1S1 M_P1S1[R:=P1, C=0, T=P1]
Message de P4S1 M_P4S1[R:=P4, C=0, T=P4]

Le meilleur message est M_P1S1

Sur S2

Message de P1S2 M_P1S2[R:=P1, C=0, T=P1]
Message de P2S2 M_P2S2[R:=P2, C=0, T=P2]
Message de P3S2 M_P3S2[R:=P3, C=0, T=P3]

Le meilleur message est M_P1S2

Sur S3

Message de P2S3 M_P2S3[R:=P2, C=0, T=P2]
Message de P3S3 M_P3S3[R:=P3, C=0, T=P3]
Message de P5S3 M_P5S3[R:=P5, C=0, T=P5]

Le meilleur message est M_P2S3

Sur S4

Message de P4S4 M_P4S4[R:=P4, C=0, T=P4]
Message de P5S4 M_P5S4[R:=P4, C=0, T=P4]

Le meilleur message est M_P4S4

A l'issue de ce premier tour

P1 n'a pas reçu de meilleur message sur un de ses ports que celui qu'il émettrait. P1 continue à émettre des messages 802.1d.

P2 a reçu un meilleur message sur son port P2S2 il cesse d'émettre ses messages 802.1d

P3 a reçu un meilleur message sur son port P3S2 il cesse d'émettre ses messages 802.1d

P4 a reçu un meilleur message sur son port P4S1 il cesse d'émettre ses messages 802.1d

P5 a reçu un meilleur message sur son port P5S4 il cesse d'émettre ses messages 802.1d

P1 considère qu'il est le pont racine.

Pour déterminer le port racine et le port désigné provisoire

P2 P3 et P4 considère que P1 est le port racine (ils ont respectivement sauvegardé les messages M_P1S2 et M_P1S1) P2 détermine que son port racine est P2S2, pour P3 c'est P3S2 et pour P4 P4S1

P5 considère que le pont racine est P2 (il a sauvegardé le message M_P2S3). Pour lui son port racine est P5S3

Pour chaque pont l'autre port est un port désigné, sauf pour le pont racine ou on a deux ports racines qui sont aussi ports désignés.

Pour déterminer l'état de chaque port

Deuxième tour

Seul P1 transmet un nouveau message 802.1d, c'est le seul pont à se considérer comme pont Racine.

Les autres ponts lisent le message 802.1D sur le port racine et le retransmettent sur l'autre port en modifiant le coût et le transmetteur.

P1 transmet un nouveau message 802.1d sur S1 et S2

M_P1S1(R=P1,C=0,T=P1)

M_P1S2(R=P1,C=0,T=P1)

Les autres ponts sauvegardent le message en modifiant le coût et le transmetteur et le retransmettent sur le port désigné.

Le pont 2 retransmet le message

M_P2S3(R=P1,C=10,T=P2)

Le pont 3 retransmet le message

M_P3S3(R=P1,C=10,T=P3)

Le pont 4 retransmet le message

M_P4S4(R=P1,C=10,T=P4)

Le pont 5 retransmet le message

M_P5S4(R=P1,C=20,T=P5)

A l'issue du deuxième tour tous les ponts considèrent P1 comme le pont racine.

Le CCR pour P1 est zéro.

Le CCR pour P2 est 10.

Le CCR pour P3 est 10.

Le CCR pour P4 est 10.

Le CCR pour P5 est 20.

Lorsqu'un pont reçoit un meilleur message sur un port non racine que celui qu'il transmet, il bloque ce port.

P3 bloque son port P3S3.

P5 bloque son port P5S4.

P2S3 est le port désigné de P2.

P4S4 est le port désigné de P4.

Chaque pont a un port racine, et un port désigné ou bloqué l'arbre est stable. Les ponts commencent à construire les tables d'adresses MAC et à jouer leur rôle.

Un "age" fixé à zéro est associé à chaque message sauvegardé par les ponts. Un timer est déclenché qui incrémente de un cet "age". Au bout d'une certaine limite si le pont n'a pas reçu de message sur son port racine, l'arbre de recouvrement doit être recalculé.

L'introduction d'un nouveau pont crée une période d'instabilité. En réalité dans ce que l'on vient de décrire, on a considéré que tous les ponts étaient initialisés en même temps, ce qui ne correspond pas à la réalité. Mais on peut appliquer l'algorithme en décalant l'initialisation des ponts et constater que celui-ci provoquera un résultat stable.

Quatrième partie : Application à une architecture basée sur des commutateurs

A. Nommer les différents domaines de collision

Dans le schéma chaque Hub correspond à un domaine de collision :

Hub1	segment 1	S1
Hub2	segment 2	S2
Hub3	segment 3	S3
Hub4	segment 4	S4
Hub5	segment 5	S5
Hub6	segment 6	S6
Hub7	segment 7	S7
Hub8	segment 8	S8

B. Nommer les commutateurs et les ports. Affecter un coût

Commutateurs	Noms
Switch 1	SW1
Switch 2	SW2
Switch 3	SW3

Ports	Noms	Noms	Noms	Noms	Noms
-------	------	------	------	------	------

Ports de SW1	SW1S1	SW1S2	SW1S3	SW1S4	SW1S8
Ports de SW2	SW2S6	SW2S7	SW2S8		
Ports de SW3	SW2S4	SW2S5	SW2S6		

Tous les ports ont un coût de 10

C. Lister les trames

A l'initialisation chaque switch envoie le message suivant sur chacun de ses ports

M_SW1 [R=SW1,C=0,T=SW1]

M_SW2 [R=SW2,C=0,T=SW2]

M_SW3 [R=SW3,C=0,T=SW3]

SW1 reçoit

Sur le port SW1S4 le message M_SW3

Sur le port SW1S8 le message M_SW2

SW2 reçoit

Sur le port SW2S8 le message M_SW1

Sur le port SW2S6 le message M_SW3

SW3 reçoit

Sur le port SW3S4 le message M_SW1

Sur le port SW3S6 le message M_SW2

A la réception de ces messages, tous les commutateurs considèrent que SW1 est le commutateur racine.

Pour SW1

Tous les ports deviennent des ports désignés et sont actifs

Pour SW2

Le port SW2S8 devient le port racine

Le port SW2S6 qui a émis le message M_SW2 meilleur que le message M_SW3 devient un port désigné

Les autres ports deviennent des ports désignés

Pour SW3

Le port SW3S4 devient le port racine

Le port SW3S6 qui a reçu le message M_SW2 meilleur que le message M_SW3 qu'il a émis devient bloqué

Les autres ports deviennent des ports désignés

A l'issue du premier tour l'arbre est constitué.

D. Noter sur le schéma les ports racines, les ports désignés et les ports bloqués.

