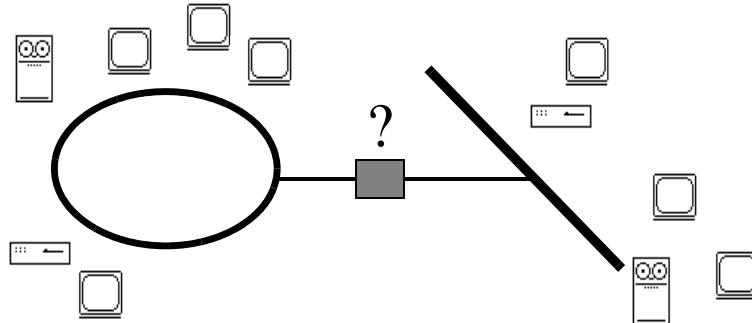


Interconnexion de réseaux locaux

(Z:\Polys\Reseaux-locaux\4.Pont.fm- 15 septembre 2008 09:36)



“Local Area Network - Media Access Control Bridge”, IEEE std 802.1D, 1993.

L.Toutain, “Resaux locaux et Internet”, Hermès, 1999.

W.Stallings, “LAN & MAN”, Prentice Hall, 1997.

PLAN

- . Présentation
- . Architecture générale
- . Répéteurs - concentrateurs
- . Ponts
- . Transparent bridging
- . Source routing
- . Conclusion

1. Présentation

Supposer l'existence d'un monde homogène est faux :

- . Réseaux grandes distances :
 - 20000 réseaux SNA (IBM) en 1990.
 - 25000 sous-réseaux IP (en sept. 94).
 - non OSI !
- . Réseaux locaux :
 - >1M.
 - plusieurs méthodes d'accès incompatibles.

L'hétérogénéité : temporaire ou permanente ?

- évolution de la localisation et du nombre de stations
- émergence de nouveaux standards
- technologies radicalement différentes (RxLx, satellite, RxGDist., sansfil, etc.)

Solution :




- un équipement d'interconnexion (répéteur, pont, routeur, passerelle).
- relié aux (2) réseaux à interconnecter.

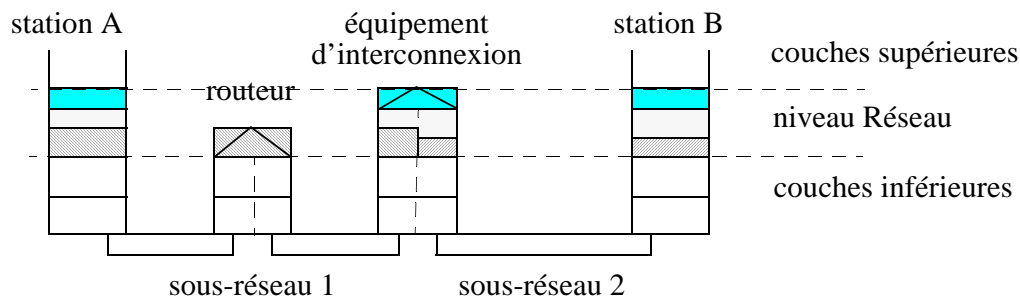
2. L'architecture générale

2.1. L'architecture d'interconnexion de l'OSI

De niveau Réseau, mais applicable à toutes les couches !

3 sous-couches :

- a) d'accès au sous-réseau 
- b) de mise à niveau 
- c) d'interconnexion 



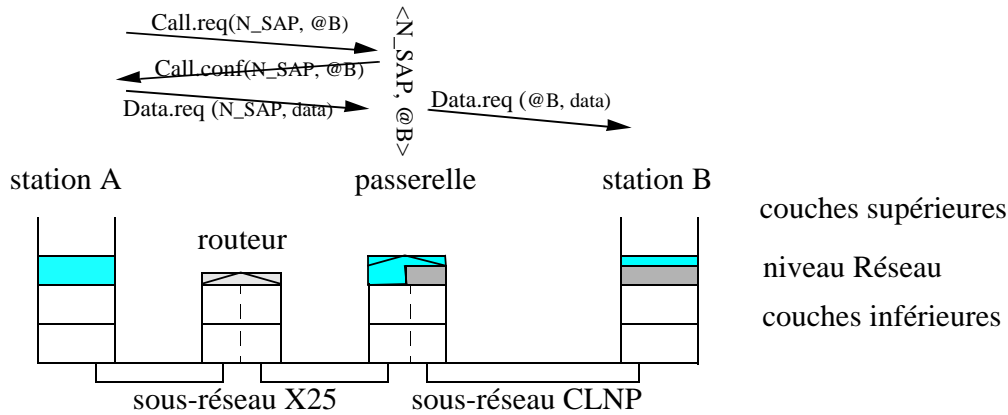
2.2. Exemple d'interconnexion

Interconnexion d'un sous-réseau en mode connecté à un sous-réseau en mode non-connecté, par exemple X25.3 et CLNP ("connectionless network protocol").

Définir la couche d'homogénéisation :

- soit mode connecté
- soit mode non connecté !
- soit autre !

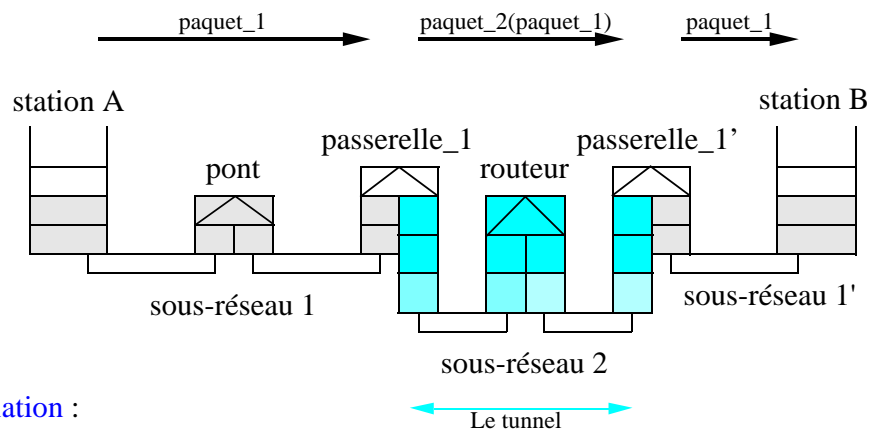
Quelques problèmes : l'adressage, l'absence de certains types de paquets, etc.



2.3. Le "Tunnelling"

Interconnexion de 2 (sous-)réseaux de type identique à travers un réseau d'un autre type :

- les messages des 2 sous-réseaux d'extrémité sont transportés tels quels par le réseau d'interconnexion.
- exemple : 2 réseaux Ethernet interconnectés par Internet, ou le mbone.



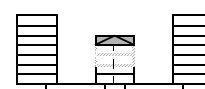
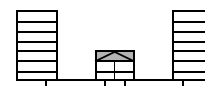
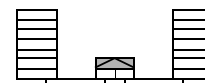
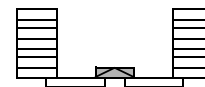
Encapsulation :

Chaque paquet_1 est placé en tant que données (c.-à-d. encapsulé) dans un paquet_2.

2.4. Les niveaux d'interconnexion

L'interconnexion peut être effectuée à tous les niveaux :

- couche 1 (Physique) : **modem, répéteur, concentrateur**
 - . techniques de modulation adaptées au support physique
 - . par ex.: interconnexion entre brins (segments) d'un seul Ethernet
- couche 2 (Liaison de données) : **pont**
 - . conversion entre différentes méthodes d'accès
 - . par ex.: interconnexion de réseaux locaux
- couche 3 (Réseau) : **routeur**
 - . prévue pour !
- couches supérieures : **passerelle**, (relai, convertisseur de protocoles)
 - . interopérabilité de niveau applicatif
 - . par ex.: messagerie SMTP<-> X400



3. Modems, répéteurs et concentrateurs

3.1. Introduction

Traitement au niveau du signal dû aux limitations physiques,
Pas de traitement sémantique (pas de notion de bit ou de trame)

Modem :

- rôle :
 - . adaptation entre deux supports physiques de communication différents

Répéteur :

- rôle :
 - . par ex. relie deux segments Ethernet (au sein d'un seul réseau Ethernet)
 - . extension de la couverture du réseau
 - . compatibilité entre deux segments utilisant des supports de type différents

Concentrateur :

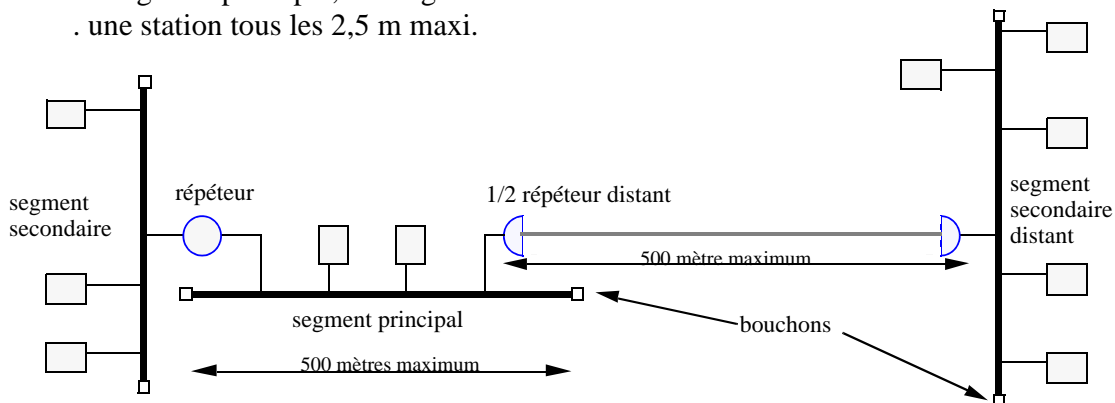
- rôle :
 - . permet le partage d'un seul point d'accès au réseau local entre plusieurs stations.
- dû aux limitations :
 - . distance minimale entre points d'accès
 - . difficulté de la pose des prises sur le câble coaxial

Hub : répéteur + concentrateur

3.2. Répéteurs Ethernet

Un réseau Ethernet utilise des répéteurs pour interconnecter ses différents segments (brins).

- . amplificateur du signal, détecteur de collision, générateur de brouillage
- . répéteur local ou distant (la liaison intermédiaire peut être en f.o.)
- . 1 segment principal, des segments secondaires
- . une station tous les 2,5 m maxi.



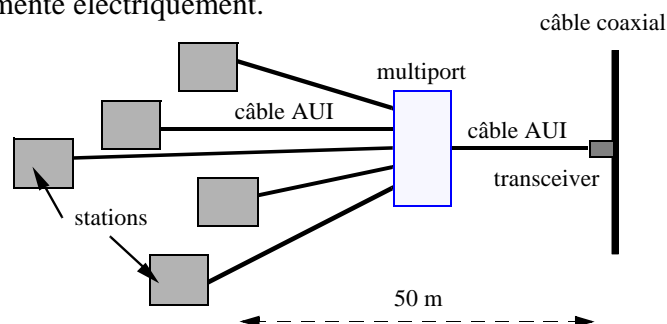
Ne pas confondre :

- . un segment d'un réseau local,
- . un réseau local (formé de plusieurs segments),
- . un ensemble de réseaux locaux interconnectés (par des ponts).

3.3. Concentrateurs

Fan-out (éventail), multiport :

- . distance minimale entre deux connecteurs,
- . topologie centrale (facilite la gestion),
- . équipement alimenté électriquement.



Utilisé historiquement en 10Base5.

3.4. Les "hubs" (moyeux)

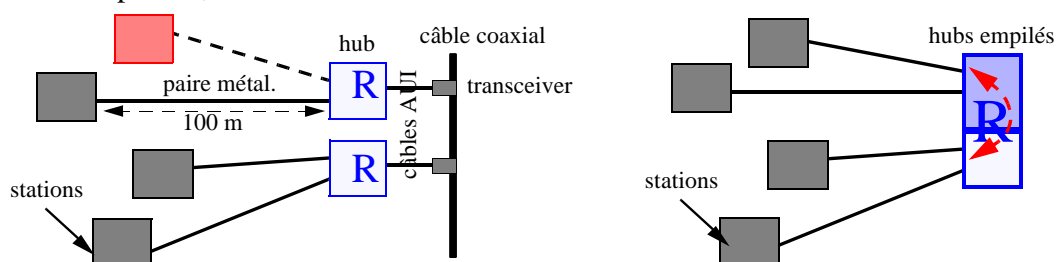
Indispensable en 10 base T

C'est un **répéteur** à multiple port, muni de trancivers intégrés

- placement en étoile, hiérarchique
- chaque station est considérée comme étant sur un segment particulier !

Équipement intelligent :

- détection et isolation des pannes,
- administrable à distance (pile SNMP)
- empilable, "mediumless": interconnexion entre bus internes



- Certains hubs (répéteurs) peuvent masquer les trames répétées sur certains ports afin d'éviter leur interception

4. Les Ponts

4.1. Introduction

Interconnexion au niveau de la couche **Liaison de données**.

- Le réseau (virtuel) étant formé de deux (ou +) réseaux locaux distincts
- interconnectés par un (ou +) équipement appelé pont ("bridge" ou "switch")

Augmentation de l'**étendue** couverte par l'ensemble des réseaux locaux interconnectés :

- potentiellement infinie.
- IRLE (Interconnexion de réseau local d'entreprise), "LAN_E"

Augmentation de la **bande passante** :

- le trafic local à un réseau local reste local
- les débits des réseaux locaux sont potentiellement additionnés

Augmentation de la **sécurité** :

- isolation : le trafic d'un réseau local ne pénètre pas dans l'autre réseau local
 - pont filtrant, "firewall"
- chiffrement de trames spécifiques circulant dans certaines parties du réseau

Contrôle d'accès et bufférisation

- . Les informations passant d'un RL à l'autre doivent obtenir le droit d'accès au médium du deuxième réseau.
- . Le pont fonctionne comme un récepteur sur le premier réseau et comme un émetteur sur le deuxième.
 - > gestion de l'espace de stockage

Adressage

- . @ IEEE 802 dite universelle : 6 octets
- . l'unicité de l'adresse attribuée à chaque station est assurée

Transparence

- . Grace à la structuration en couches, l'interconnexion à un certain niveau est insensible aux au type des protocoles supérieurs :
 - . ici, les ponts sont insensibles aux protocoles des couches 3 et plus

Service d'interconnexion minimum

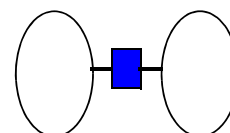
- . Le service assuré par l'interconnexion est généralement le service minimum commun :
 - . pas de gestion des priorités, pas de traitement des erreurs, pas de segmentation, etc.
 - . c'est-à-dire le service simple d'Ethernet

4.2. Les types de ponts

Un pont peut interconnecter deux RL identiques ou de types différents.

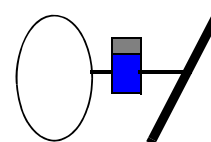
Homogène :

- . les deux réseaux étant de type identique :
 - le format des trames est compatible.
 - les adresses sont compatibles
- . simple : une seule entité de la sous-couche MAC est nécessaire dans le pont.



Hétérogène :

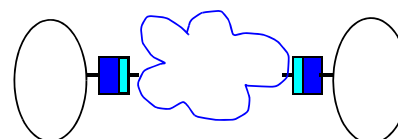
- . le pont fonctionne comme un "translater" (traducteur) :
 - de formats de trames (longueur max. des trames ?)
 - de services (niveau de priorité ?)



L'interconnexion peut être directe ou indirecte.

Indirecte :

- . le réseau intermédiaire fonctionne en "tunnelling".
- . les trames originelles sont encapsulées dans les trames du réseau intermédiaire.



4.3. Les ponts

Il existe deux principes de fonctionnement : les ponts simples et les ponts intelligents.

Pont simple :

- . Réémet toutes les trames reçues d'un réseau sur l'ensemble des autres réseaux.
- . Pas utilisé ... sauf par défaut... car inefficace.

Pont intelligent :

Agit alors comme un routeur

- . En fonction de l'adresse de destination (@ MAC, @ IEEE 802, @ Ethernet, ...).
- . Connaît le réseau local sur lequel est connectée chaque station.
- . Seules sont réémises les trames dont on sait qu'elles sont à destination d'une station connectée à un autre réseau local que le réseau local de la station émettrice.
- . Les transmissions locales restent locales : le trafic est optimisé.

Il existe deux grandes techniques de pontage :

- "Transparent bridging" ... + "Spanning tree"
- "Source routing"

5. Le “Transparent bridging”

5.1. Introduction

Utilisé généralement pour Ethernet

Principe :

- . auto-apprentissage
- . diffusion (par inondation) par défaut
- . optimisation par “caching”

Fonctionne en mode “promiscuous” :

- le pont traite toutes les trames circulant sur les réseaux locaux auxquels il est connecté.
- le pont doit être puissant.
- le pont est un point critique de sécurité.

Chaque pont possède une table (de pontage) :

- qui associe à chaque adresse le réseau local vers lequel le pont doit l'acheminer (cela peut ne pas être le réseau local sur lequel est connectée directement la station)
- la table est gérée comme une mémoire cache

Identification des ponts : [sur 8 octets]

- 2 octets : priorité (0= +prioritaire), gestion de la redondance par l'administrateur
- 6 octets : @IEEE 802

Identification des ports de chaque pont

- un numéro unique sur 2 octets

5.2. L'algorithme

Lorsque le pont reçoit une trame :

Si une entrée valide dans sa table de pontage correspond à l'[adresse de destination](#) de la trame
 alors /* la table de pontage contient l'adresse de destination */
 . si le réseau local associé à cette entrée est différent du réseau local dont est issue la trame,
 . alors [1] /* la trame doit être pontée */
 la trame (inchangée) est réémise vers ce nouveau réseau local.
 . sinon [2] /*rien*/
 sinon [3] /*la table de pontage ne contient pas l'adresse */
 . la trame est diffusée vers tous les réseaux locaux sauf le réseau local d'où elle provient.

Si aucune entrée valide dans sa table de pontage ne correspond à l'[adresse d'émission](#) de la trame
 alors [1'] /* la table de pontage ne contient pas l'adresse d'émission */
 . une entrée est créée dans la table de pontage associant l'adresse et le réseau local d'où est
 issue la trame.
 sinon /* la table de pontage contient déjà l'adresse */
 . si le réseau local dont est issue la trame et celui de l'entrée sont différents
 . alors [2']
 l'entrée est modifiée en conséquence
 . sinon [3'] /*rien*/

5.3. Conséquences

Le nombre d'adresses peut être grand :

- grand espace de stockage !
- le temps de recherche est important :
 ⇒ utilisation d'un procédé de hachage
- récupération des entrées les moins utilisées (LRU : Least recently used)

Utilisation d'un temporisateur :

- . détection des entrées inactives (environ 20 secondes)
- . suppression des entrées périmées (déplacement ou disparition des stations)

Par défaut l'algorithme fonctionne :

- . par inondation totale
- . robuste mais peu efficace

Plusieurs réseaux locaux peuvent être interconnectés par plusieurs ponts :

- . Assure une certaine redondance (volontaire ou par mégarde).
- . Le réseau peut présenter alors des cycles :
 ⇒ Une même trame peut être recopiée et reçue indéfiniment !

5.4. Le “Spanning tree”

Arbre de recouvrement :

- . construction d'un arbre recouvrant tous les ponts
 - . en désactivant certains ports de certains ponts, on élimine les cycles
 - . il existe plusieurs arbres recouvrants pour une même topologie !
- ==> “l'arbre des plus courts chemins”

Algorithme de construction d'un arbre de recouvrement total :

- . algorithme d'élection basé sur les **adresses + coût + n° port**.
 - . la racine de l'arbre sera le pont de + petite adresse
 - . les liaisons actives seront celles de + faible coût à partir de cette racine.
- . en cas d'égalité, on choisit
 - le chemin passant par le pont de plus petite adresse
 - le + petit n° de port (interface de communication).

Structure des messages de configuration [valeur lors de l'initialisation]:

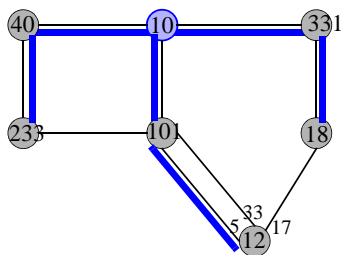
- . Identité de la racine (proposée) [l'émetteur].
- . Coût de la liaison entre l'émetteur et la racine [0].
- . Identité de l'émetteur.
- . Numéro du port de l'émetteur.

Coût des liens, en fonction inverse de leur débit :

- . $\text{Coût}(\text{lien}) = 10^9 / \text{débit}(\text{lien})$ (il existe une version adaptée aux débits supérieurs à 1 Gbit/s)

Exemple de construction d'un arbre couvrant :

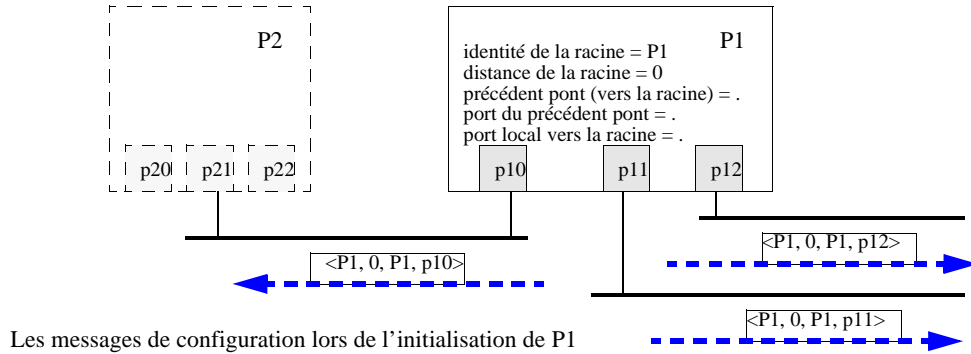
- hypothèse : tous les liens ont même coûts



5.5. L'algorithme du "Spanning tree"

5.5.1 Initialisation

A l'initialisation, chaque pont diffuse sur tous ses ports un message de configuration contenant la valeur initiale.

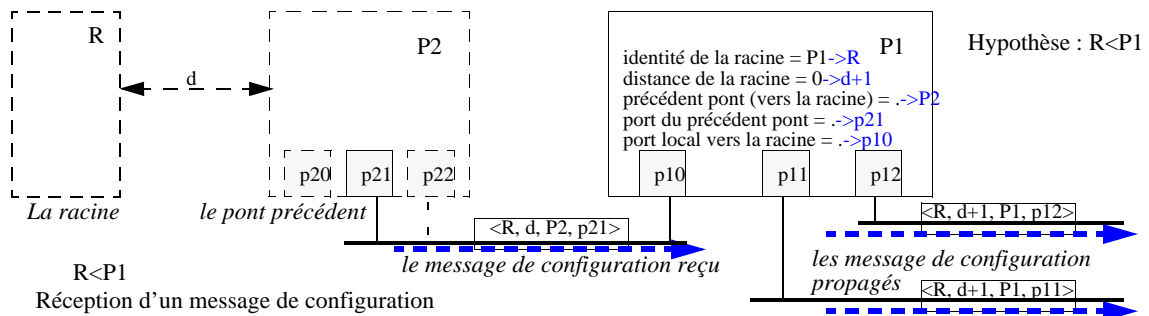


Chaque pont mémorise la plus petite valeur reçue sur chacun de ses ports.

5.5.2 Sélection

Lorsqu'un pont reçoit un message de configuration, il compare sa valeur courante locale avec la valeur reçue figurant dans le message (après avoir incrémenté son coût) :

- Si cette dernière est meilleure (+ petite)
 - les informations locales sont mises à jour à partir des données du message de configuration reçu et le port de réception
 - un nouveau message de configuration est propagé vers toutes les interfaces sauf vers le port de réception.
- sinon le message de configuration reçu est intercepté

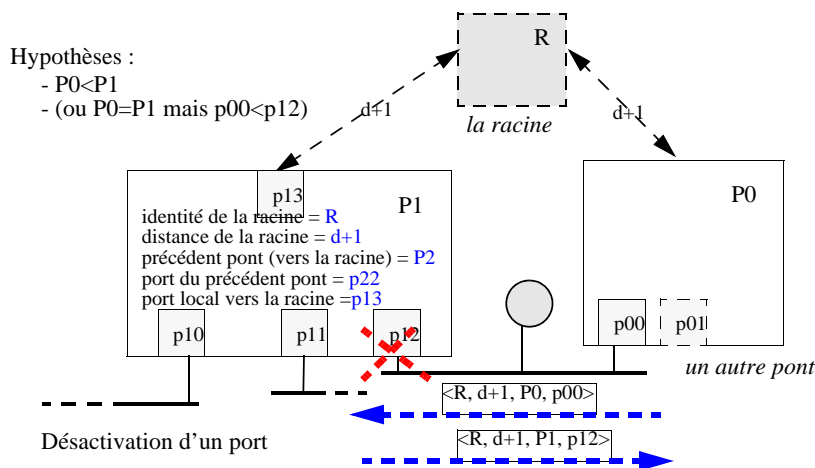


Cet algorithme sélectionne entre deux chemins de même coût celui qui emprunte le pont de plus petite adresse et celui utilisant le plus petit numéro de port.

5.5.3 Désactivation

Il faut **désactiver** les ports qui créent des cycles. Ces ports alimenteraient de manière superflue certains réseaux, déjà alimentés par un autre (meilleur) pont.

Tout port, qui reçoit un message de configuration meilleur strictement (c'est-à-dire sans incrémentation du coût) que la configuration locale mais moins bon si l'on tient compte de l'incrémentation du coût, doit être désactivé.



Lorsque le même pont possède plusieurs ports sur le même réseau local, tous ses ports sont désactivés sauf un, grâce au dernier terme : le numéro de port.

5.5.4 Surveillance

La désactivation d'un port et le calcul d'une nouvelle reconfiguration sont retardés (forward delay : 15 s) :

- . afin que chaque pont du réseau ait le loisir d'arriver dans un état stable, le temps de faire parvenir les messages aux autres ponts

Les liaisons doivent être surveillées :

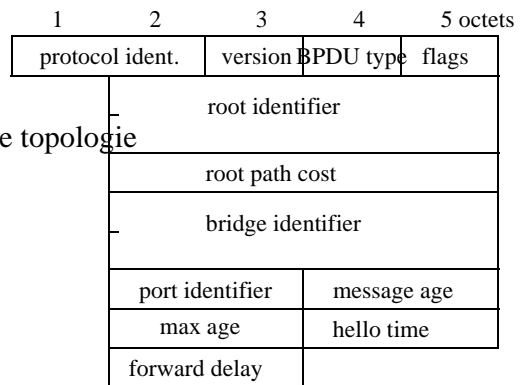
- . pour détecter les pannes des autres ponts.
- . un message de configuration est émis périodiquement (toutes les 2 s : hello time) par la racine et propagé par les ponts.
- . chaque message est estampillé (daté/numéroté) pour détecter les retards (message age).

Les informations reçues ont une durée limite de validité :

- . un message périmé est ignoré
- . une information périmée est invalidée
- . cette durée est conventionnellement de 20 s (max age).

5.6. Le format des messages : Bridge PDU

- . protocol identifieur : [0]
- . version : version du protocole[0]
- . message type :
 - 0 = message de configuration
 - 128 = message de notification de changement de topologie
- . flags :
 - Topology change bit (bit 8)
 - Topology change acknowledgment bit (bit 1)
- . root identifieur : le numéro de la racine
- . path cost : le coût du chemin entre l'émetteur et la racine
- . bridge identifieur : le numéro de l'émetteur du message
- . port identifieur : le numéro du port d'émission
- . message age : date d'émission du message (1/256s)
- . max age : date ultime de vie du message et de l'information qu'il contient (1/256s)
- . hello time : fréquence d'envoi des messages de configuration (1/256s)
- . forward delay : délai conditionnant le changement d'état du pont (1/256s)

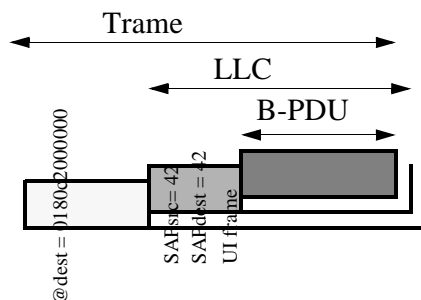


Bridge-PDU

5.7. Encapsulation

Les messages de configuration (B-PDU) utilisés par le "transparent bridging" sont encapsulés dans des trames LLC puis (par exemple) une trame Ethernet :

- adresse Mac de groupe correspondant aux ponts : 0x0180c2000000
- numéro de L-SAP correspondant au service d'administration des ponts : 0x42



- la structure des messages de configuration (un peu particulière) est prévue pour assurer l'alignement (en frontière de mots de 32 bits) de ses principaux champs, l'entête LLC faisant 3 octets (habituellement).

5.8. Conclusion

L'algorithme du "Transparent bridging" est un algorithme simple.

Dans un environnement d'interconnexion non-contrôlé, il nécessite la mise en oeuvre d'un algorithme complémentaire : le "Spanning tree".

Il est particulièrement utilisé pour réaliser l'interconnexion des réseaux locaux Ethernet.

Il fonctionne par défaut par inondation,

et utilise l'auto-apprentissage pour optimiser l'acheminement des trames.

Le "Spanning tree" supprime la redondance,

et ne permet donc pas d'utiliser tous les liens pour étaler la charge.

Quelques développements regroupés dans la norme 802.1q :

- VLAN
- "Rapid Spanning Tree Protocol" :
 - . convergence plus rapide en cas de panne
- "Multiple Spanning Tree Protocol" : un arbre couvrant par VLAN

6. Le "Source Routing"

6.1. Introduction

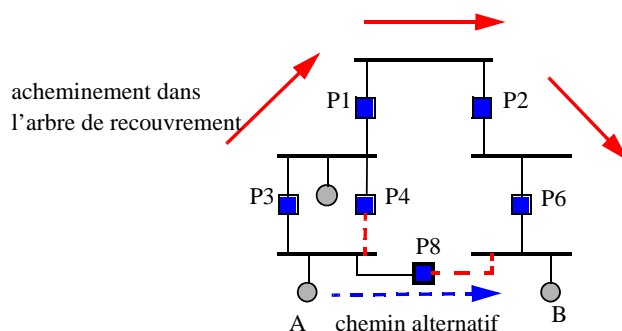
Proposé par IBM pour le Token Ring.

Problèmes du "Transparent bridging" :

- . les ponts redondants ne sont pas utilisés
- . un seul pont (un seul chemin) entre 2 points du réseau
- . les chemins optimaux ne sont pas suivis

Les ponts en mode "promiscuous" travaillent beaucoup :

=> les ponts "Source routing" n'explorent que les trames marquées



6.2. Constitution des trames

Principe :

Chaque station connaît le meilleur chemin à suivre pour parvenir au destinataire

. pour les trames destinées à une station située sur le même réseau

local : émission normale (directe)

. pour les trames destinées à des stations distantes :

- elles sont marquées (bit de poids fort de l'adresse d'émission à 1)

- en tête du champ de données : le chemin à suivre, c'est-à-dire :

⇒ Liste de couples <n° RL, n° pont> :

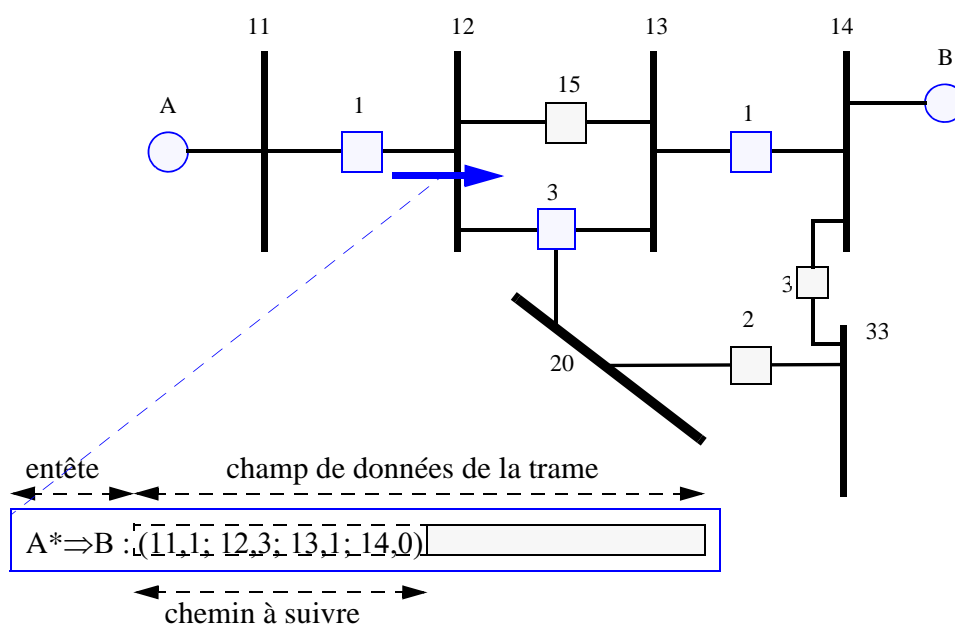
. n° RL identifie de manière unique le réseau local (12 bits)

. n° pont identifie le pont dans son réseau local (4 bits)

note : deux ponts situés sur des réseaux locaux différents peuvent avoir même numéro.

note : le n° de pont 0 est réservé pour identifier la fin de la liste.

6.3. Exemple de "Source routing"



6.4. L'acheminement par le "Source routing"

Les ponts ne s'intéressent qu'aux trames marquées.

Ils analysent le chemin présent dans la trame,

s'il contient le n° de RL d'où provient la trame

- . si ce n° est suivi du n° du pont
- . alors il retransmet la trame vers le RL dont le n° suit
 - . s'il existe
 - . sinon le chemin est incorrect.

S'il n'existe pas de composant spécialisé, la scrutation de la marque se fait par logiciel en mode "promiscuous" !

6.5. Algorithme de recherche d'un chemin

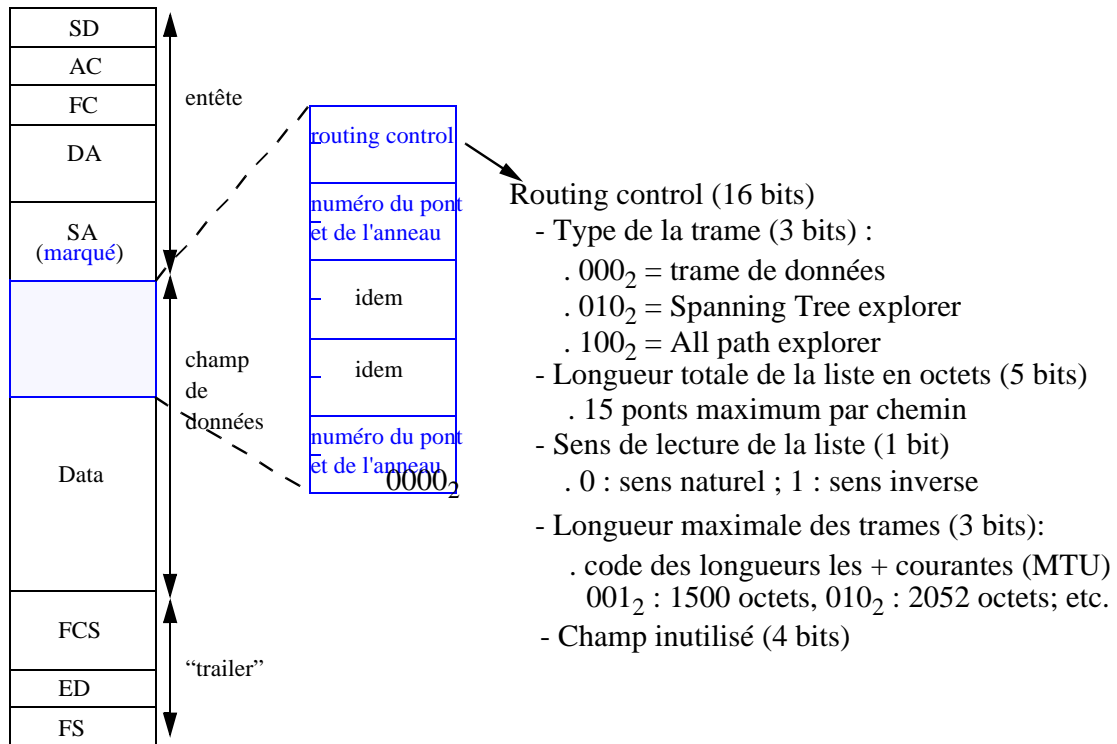
Principe :

- . Lorsque que le chemin vers une destination est inconnue,
- . L'émetteur diffuse une trame de recherche avec comme paramètre l'adresse de la station dont le chemin est recherché !
 - cette trame est munie initialement d'une liste vide.
- . Chaque pont rencontré recherche son numéro dans la trame :
 - s'il n'y figure pas, il place son identificateur et celui du réseau local d'où elle provient dans la trame,
 - sinon il détruit la trame (qui boucle).
- . Le destinataire retourne la trame de recherche à l'émetteur (**inversion du chemin**)
- . L'émetteur choisit parmi toutes les trames retournées le meilleur chemin :
 - généralement celui de la première retournée

Variantes :

- diffusion globale (par inondation)
- diffusion sur un arbre recouvrant

6.6. Le format des trames



6.7. Conclusion

Le “source routing” est adapté au Token Ring.

La phase d'apprentissage des chemins est coûteuse :

- . par défaut nécessité une diffusion totale
- . par inondation ou après construction d'un arbre

Robuste et très systématique (par inondation) :

- . trouve toujours le meilleur chemin

Rapide mais parfois inefficace (par arbre recouvrant) :

- . risque de congestion

Optimisation :

- . chaque station mémorise les chemins trouvés
- . cette mémoire “cache” est consultée lors de chaque émission
- . ces chemins ont une durée de vie limitée

7. Conclusion

Equipements d'interconnexion : répéteurs, concentrateurs et ponts.
 Quelques techniques d'interconnexion : le "tunnelling", la traduction.

Source routing versus Transparent bridging

. intelligence dans les stations ou dans les ponts

Approches complémentaires :

Si le bit de l'adresse source est positionné \Rightarrow Source routing

sinon \Rightarrow Transparent bridging

Les deux techniques utilisent la diffusion :

. adaptées aux réseaux locaux (où la diffusion est naturelle)

Les deux techniques :

- . + multi-protocoles
- . - ne permettent pas d'adapter la taille des trames (segmentation)
- . - sont peu efficaces s'il y a un trop de stations,
- . - ne gèrent pas les erreurs.

\Rightarrow Le routage et les routeurs (IP)

Extension : Les "Virtual LANs" (IEEE 802.1Q ou ISL "Inter Switch Link" of Cisco)

The Spanning Tree Algorithm: "Algorhythm"

(by Radia Perlman)

I think that I shall never see
 A graph more lovely than a tree.
 A tree whose crucial property
 Is loop free connectivity.
 A tree that must be sure to span.
 So packets can reach every LAN.
 First, the root must be selected.
 By ID, it is elected.
 Least cost paths from root are traced.
 In the tree, these paths are placed.
 A mesh is made by folks like me,
 Then bridges find a spanning tree.

(adapted from "Trees" by Joyce Kilmer)