

S22 - Architecture des réseaux

Commutation

Cédric Wemmert

IUT Robert Schuman – Département Informatique

wemmert@unistra.fr

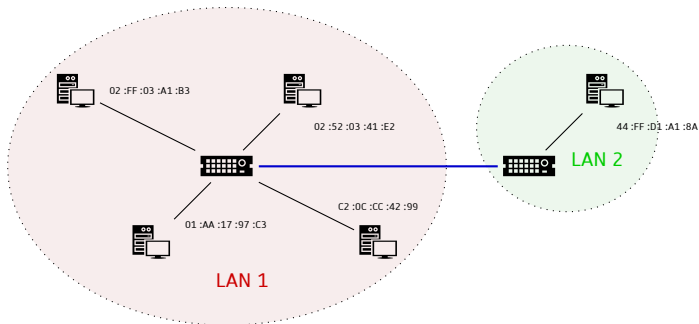
2020

Mécanismes de niveau liaison

- Local Area Network (LAN) et adressage Ethernet
- Accès au médium : CSMA/CD
- Ponts et commutation de trames
- Spanning Tree Protocol (STP)
- Virtual Local Area Network (VLAN)

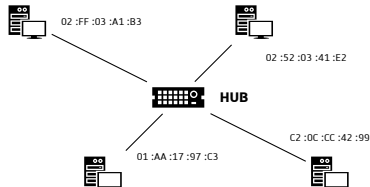
Ponts et LAN étendus

- Inter-connexions de plusieurs LAN au niveau 2 :
 - trames MAC commutées sur un ou + ports de sortie
 - commutation en fonction des infos de niveau MAC
- Formation d'un LAN étendu



Hub vs switch

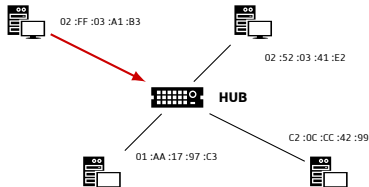
Hub (ou concentrateur)



- Un *hub* (ou concentrateur) fonctionne au niveau physique

Hub vs switch

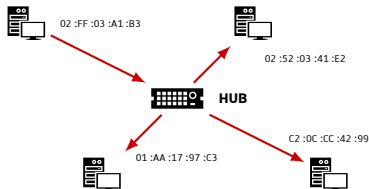
Hub (ou concentrateur)



- Un *hub* (ou concentrateur) fonctionne au niveau physique

Hub vs switch

Hub (ou concentrateur)



- Un *hub* (ou concentrateur) fonctionne au niveau physique
- La trame est diffusée sur tous les ports à toutes les stations connectées

Hub vs switch

Hub (ou concentrateur)

- Un hub ne choisit pas sur quel(s) port(s) envoyer chaque trame
- Les équipements qui y sont connectés partagent le même :
 - *domaine de diffusion* (broadcast domain)
 - *domaine de collision*

Domaine de diffusion

Une station peut communiquer avec n'importe quelle autre station du même domaine de diffusion

Domaine de collision

Les trames de deux stations au sein d'un même domaine de collision peuvent entrer en collision

Hub vs switch

Le switch (ou commutateur)

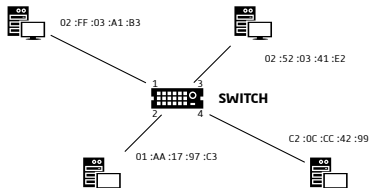


Table de commutation

Adresse MAC	Port
01:AA:17:97:C3	2
02:52:03:41:E2	3
...	

- Un *switch* (ou commutateur) fonctionne au niveau liaison
- Il possède une **table de commutation** contenant les liens entre port et adresse MAC

Hub vs switch

Le switch (ou commutateur)

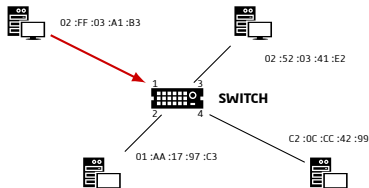


Table de commutation

Adresse MAC	Port
01:AA:17:97:C3	2
02:52:03:41:E2	3
...	

- Un *switch* (ou commutateur) fonctionne au niveau liaison
- Il possède une table de commutation contenant les liens entre port et adresse MAC
- **Les trames sont décodées** pour obtenir l'adresse MAC destination

Hub vs switch

Le switch (ou commutateur)

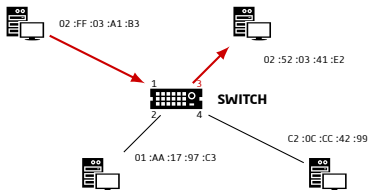


Table de commutation

Adresse MAC	Port
01:AA:17:97:C3	2
02:52:03:41:E2	3
...	

- Un *switch* (ou commutateur) fonctionne au niveau liaison
- Il possède une table de commutation contenant les liens entre port et adresse MAC
- Les trames sont décodées pour obtenir l'adresse MAC destination
- La trame n'est envoyée qu'à la station destination **si le port est connu** dans la table de commutation

Hub vs switch

Switch (ou commutateur)

- Un switch choisit sur quel(s) port(s) envoyer chaque trame
- Les équipements qui y sont connectés partagent le même *domaine de diffusion* (broadcast domain)
- Chaque port a son propre *domaine de collision*
- Un switch est « invisible » pour les stations
- Deux fonctions principales :
 - Construire la table de commutation (forwarding table) :
 - décide « où va quoi » : quel port pour quelles adresses ?
 - Gérer l'interconnexion des switches :
 - éviter les boucles mais autoriser la redondance : Spanning Tree Protocol

Hub vs switch

Switch (ou commutateur)

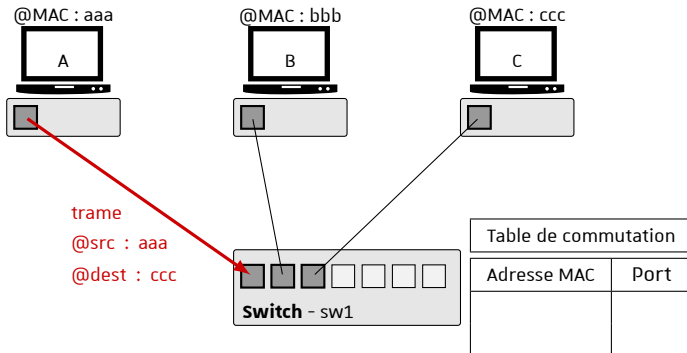
- Un switch choisit sur quel(s) port(s) envoyer chaque trame
- Les équipements qui y sont connectés partagent le même *domaine de diffusion* (broadcast domain)
- Chaque port a son propre *domaine de collision*
- Un switch est « invisible » pour les stations

Table de commutation

Une entrée par @MAC connue :

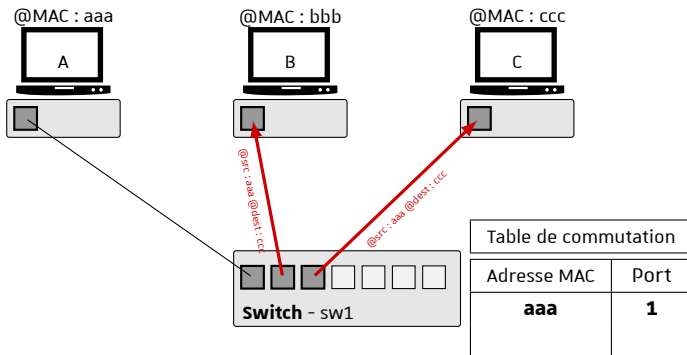
- @MAC : index de l'entrée
- interfaces / ports : sortie pour atteindre l'adresse index (plusieurs si multicast)
- âge : durée de validité de l'entrée (cache)
- divers flags : statique/dynamique, filtrage, ...

Un exemple d'application de l'algorithme de propagation



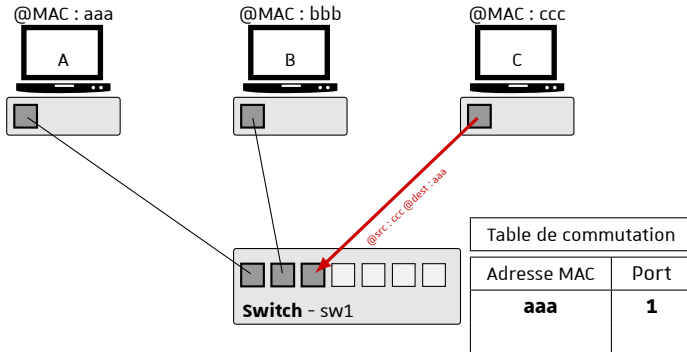
1. A veut parler à C

Un exemple d'application de l'algorithme de propagation



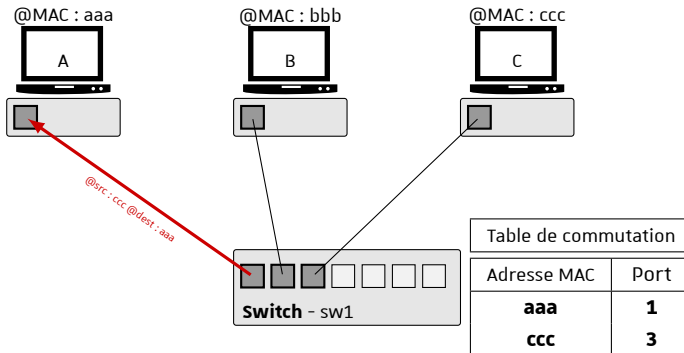
2. Le switch (commutateur) ne sait pas où est C. Il transmet la trame sur tous les ports. Au passage, il apprend l'adresse de A et l'associe au port sur lequel est arrivée la trame

Un exemple d'application de l'algorithme de propagation



3. C répond à A. B ignore la trame car elle n'est pas destinataire

Un exemple d'application de l'algorithme de propagation



4. Seule A reçoit la réponse de C. Le commutateur apprend où est C au passage.

Propagation et construction

Notes

- si @MAC destination présente partout :
 - la trame ne circule que là où il faut
 - du LAN émetteur, puis de switch en switch jusqu'au LAN récepteur
 - si source et destination dans le même LAN, pas de sortie du LAN
- si @MAC destination inconnue :
 - la trame circule dans tout le LAN étendu (comme avec un LAN simple)
 - LAN étendu = *domaine de diffusion*

Configuration manuelle/statique

- Peu pratique :
 - l'administrateur doit connaître les @MAC et leurs emplacements ☹️
 - nécessite une mise-à-jour à chaque modification du réseau

Propagation et construction

Configuration dynamique/automatique

- Apprentissage permanent
- À l'arrivée d'une trame avec @MAC source=S et via le port i :
 - ajouter/mettre à jour l'entrée d'index S avec out(S)=i
 - ré-initialiser le champ âge
- Si l'âge d'une ligne arrive à expiration :
 - suppression de l'entrée dans la table (p.ex. : 300s sans activité)
- À l'ajout d'une nouvelle station sur le réseau :
 - celle-ci émet une trame (broadcast ARP ou DHCP)
 - tous les switches apprennent sa localisation
 - l'info est rafraichie périodiquement (dynamique)
- Si une station est inactive durant un certain temps :
 - disparition de l'entrée concernée et gain de place

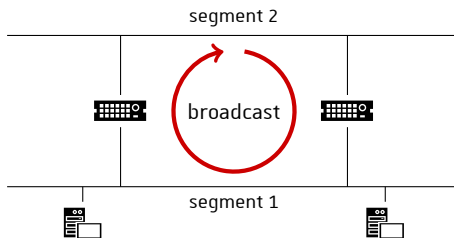
Fiabilité par redondance

- La panne d'un switch ou d'une liaison est critique car elle isole une partie du réseau
- Pour fiabiliser un réseau, on le rend **redondant** en plaçant des liaisons et/ou des switchs supplémentaires, utilisables en secours

Problème

Les boucles que cela crée empêchent le réseau de fonctionner correctement !

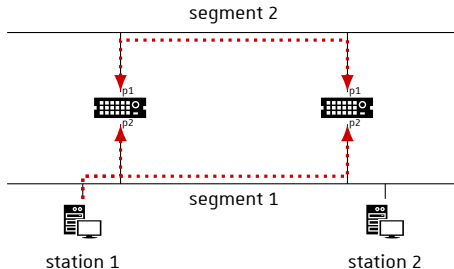
Problèmes dans les LAN étendus



Tempête de broadcast

- trame à destination de `ff:ff:ff:ff:ff:ff`
- risque de prolifération exponentielle : les trames se reproduisent à grande vitesse sous certaines conditions (cf. TD)
- réseau saturé = réseau bloqué

Problèmes dans les LAN étendus



Commutation instable

- p.ex. : trame de station 1 vers station 2
- boucle infinie de changement d'association (station 1, port)

Spanning Tree Protocol

Objectifs

- Autoriser la redondance physique...
 - boucles entre switches pour tolérance aux pannes des liens
 - boucles involontaires (erreur transitoires ou malveillance)
- ... en évitant les boucles!
- Supporter une topologie dynamique avec une configuration automatique

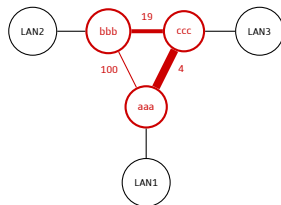
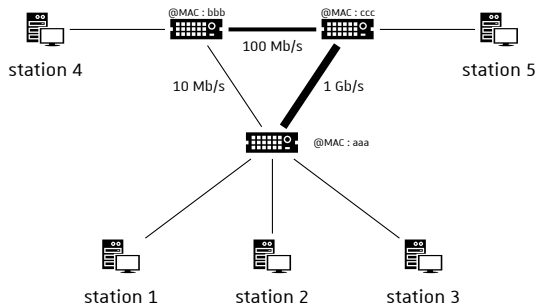
Moyens

- Définir automatiquement un arbre de recouvrement sans boucle
- Désactiver un sous ensemble d'arêtes

! STP est un protocole mis en œuvre par les switches en s'échangeant des messages spécifiques BPDU (*Bridge Protocol Data Unit*). Les hubs ne sont pas concernés et les boucles entre hubs restent interdites!

Principes du STP

- Le réseau est représenté par un graphe avec les switches et les LANs pour sommets
- Les switches sont étiquetés avec des identifiants uniques (priorité sur 16 bits + @MAC)
- Les arêtes sont étiquetées par un poids dépendant du débit : $10 \text{ Mb/s} = 100$, $100 \text{ Mb/s} = 19$, $1 \text{ Gb/s} = 4$

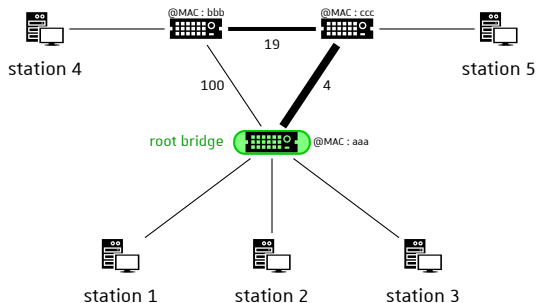


Algorithme STP

- Le switch qui possède la plus petite *Bridged Identity* (BID) est élu **root bridge**
 - Rappel : le BID est la concaténation d'une priorité entre 1 et 65536 (32768 par défaut) et l'adresse MAC du switch
- Toutes les 2s, chaque switch broadcast son BID ainsi que celle du *root bridge* donc il a connaissance
- Les ports des switchs sont mis dans une catégorie :
 - **root ports** s'ils sont connectés au root bridge
 - **designated ports** s'ils ne sont pas *root port* et s'ils autorisent le trafic à circuler
 - **non-designated ports** s'ils bloquent le trafic

Spanning Tree Protocol

Exemple

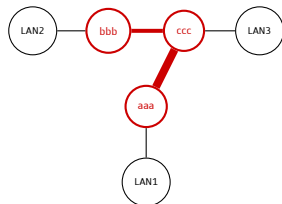
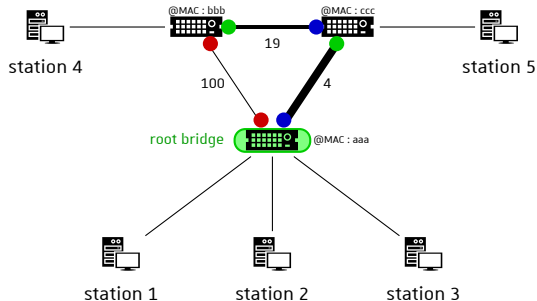


Sélection du switch racine

- Attribution à chaque switch d'un identifiant : priorité sur 16 bits + @MAC (priorité par défaut = 32 768)
- Switch racine \Rightarrow plus petit identifiant

Spanning Tree Protocol

Exemple



Détermine le chemin le moins coûteux vers le switch racine

- **root port** ⇒ utilisé pour le chemin le moins coûteux vers le switch racine (un seul par switch!)
- **designated ports** ⇒ port sur le chemin le moins coûteux entre un LAN et le switch racine
- **non-designated ports** ⇒ ni racine ni désigné

Spanning Tree Protocol

Construction d'un arbre recouvrant minimal

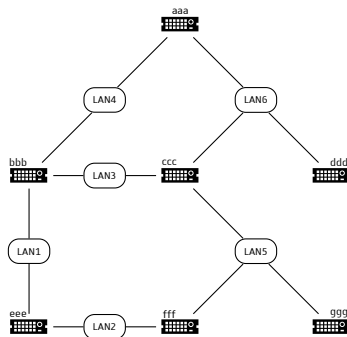
- Arbre : unicité de l'acheminement entre switches
- Recouvrant : tous les switches sont accessibles
- Minimal : entre deux segments, on sélectionne le plus performant (meilleur débit)

Optimalité

⇒ Pas *globalement* optimal! Seulement optimal vis-à-vis de la racine

Spanning Tree Protocol

Exemple 2

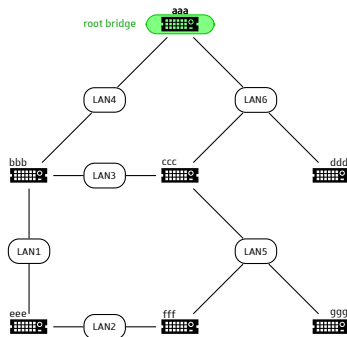


Hypothèses

- arêtes : poids identiques partout
- $aaa < bbb < \dots$

Spanning Tree Protocol

Exemple 2



Hypothèses

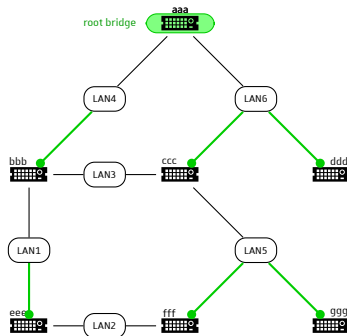
- arêtes : poids identiques partout
- $aaa < bbb < \dots$

Notes

- aaa est choisi comme *root bridge*

Spanning Tree Protocol

Exemple 2



Hypothèses

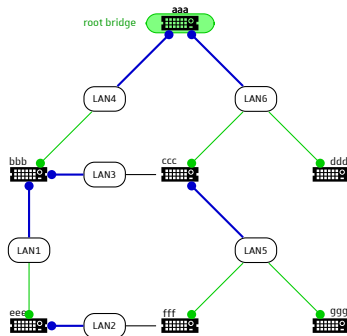
- Arêtes : poids identiques partout
- $aaa < bbb < \dots$

Notes

- aaa est choisi comme *root bridge*
- Les *root ports* sont choisis

Spanning Tree Protocol

Exemple 2



Hypothèses

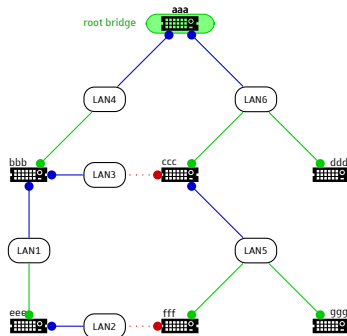
- Arêtes : poids identiques partout
- $aaa < bbb < \dots$

Notes

- aaa est choisi comme *root bridge*
- Les *root ports* sont choisis
- Les *designated ports* sont choisis

Spanning Tree Protocol

Exemple 2



Hypothèses

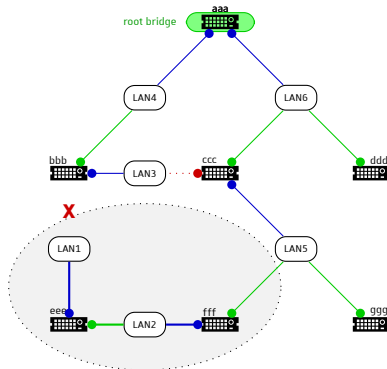
- Arêtes : poids identiques partout.
- $aaa < bbb < \dots$

Notes

- aaa est choisi comme *root bridge*
- Les *root ports* sont choisis
- Les *designated ports* sont choisis
- Les ports restants sont des *non-designated ports*, ils ne seront pas utilisés

Spanning Tree Protocol

Exemple 2



Hypothèses

- Arêtes : poids identiques partout
- $aaa < bbb < \dots$

Mise-à-jour

- En cas de perte d'un lien, l'arbre est recalculé

L'essentiel

- Hub et switch sont des équipements qui permettent de créer des LAN étendus
- Un switch choisit sur quel(s) port(s) envoyer chaque trame :
 - les équipements qui y sont connectés partagent le même domaine de diffusion
 - chaque port a son propre domaine de collision
- Problèmes dans LAN étendus en cas de cycle :
 - tempête de broadcast
 - commutation instable
- Solution : *Spanning Tree Protocol*
 - Transformation d'un graphe en arbre recouvrant minimal