

Cours de Réseaux — Partie 2

IUT de Villetaneuse — R&T 1^{ère} année

Laure Petrucci

27 novembre 2011

Table des matières

Chapitre 1

Couche liaison de données

1.1 Introduction

La *couche liaison de données* s'assure que les données sont transférées d'un point à l'autre de la liaison. Elle fournit des services de transmission et de réception à la couche réseau. Elle reçoit de la couche réseau des PDU_3 (ou NPDU) appelées *paquets*. Elle ajoute ensuite des *informations de contrôle* (LPCI), formant ainsi le PDU_2 (ou LPDU, ou *trame*) qui sera envoyé à la couche physique. Le LPCI est utilisé pour assurer le transfert correct du NPDU, c'est-à-dire pour vérifier qu'il n'y a ni erreur, ni perte, ni duplication.

La couche liaison vérifie le LPCI transmis. Si celui-ci est correct, elle extrait le LSDU et l'envoie à la couche réseau. Sinon, la trame est rejetée et un dialogue s'établit pour retransmettre correctement l'information.

La taille des trames peut être fixe ou variable, selon le protocole mis en œuvre. Des *fanions* aux extrémités d'une trame (début et fin) permettent de la délimiter. Les données sont généralement encadrées par un *entête* et une *terminaison* formant le LPCI.

1.2 Protocole LAP-B

Le protocole *LAP-B* (Link Access Protocol – Balanced) est un sous-ensemble du protocole *HDLC* (High-Level Data Link Control), mis en œuvre dans le réseau *RNIS* (Réseau Numérique à Intégration de Services).

La connexion s'effectue en *mode connecté*. Il y a par conséquent trois phases : connexion, échange et déconnexion. Pour garantir une *réception* (délivrance à la couche 3) des informations transmises à la fois *correcte, sans duplication, et dans l'ordre*, le protocole LAP-B utilise plusieurs techniques :

- un *code polynômial*, de polynôme générateur $G(x) = x^{16} \oplus x^{15} \oplus x^2 \oplus 1$;
- une *numérotation des trames* ;
- des *acquitements groupés* : le récepteur peut acquitter avec une seule trame un ensemble de trames reçues. Ainsi, l'acquitterment d'une trame n permet d'acquitter en même temps toutes les trames précédentes (de la dernière acquittée à $n - 1$) ;
- le *piggypacking* : le récepteur peut acquitter une trame d'information par une autre trame d'information ;
- l'*anticipation* : l'émetteur peut envoyer w trames sans recevoir d'acquitterment. On dit alors qu'il a une *fenêtre d'émission* (window) de taille w .

Pour pouvoir utiliser ces techniques, les *trames d'information* sont *identifiées par un numéro*. Si n est le nombre de bits utilisés pour coder le numéro de la trame, la numérotation se fait modulo 2^n . Le protocole LAP-B utilise $n = 3$ bits, les trames sont donc numérotées de 0 à 7.

1.2.1 Structure des trames

Une trame LAP-B comporte plusieurs champs décrits dans la figure ??, avec leur taille (en octets).

Fanion	Adresse	Contrôle	Données	FCS	Fanion
1o	1o	1o		2o	1o

FIGURE 1.1 – Structure d’une trame LAP-B

Les *fanions* indiquent le début et la fin de la trame. Il sont formés des 8 bits 0111 1110.

Pour que l’on puisse reconnaître les fanions, ils ne doivent pas apparaître dans le reste du message. Pour que cela ne se produise pas, lors de l’émission de la trame, on insère, après chaque séquence de 5 bits à 1, un bit à 0, appelé *bit de transparence*. Ainsi, seuls les fanions comportent 6 bits à 1 consécutifs. À la réception de la trame, un bit 0 est supprimé après chaque séquence de 5 bits à 1 consécutifs.

L’*adresse* permet de connaître la machine ayant émis une commande ou une réponse. Par convention, l’émetteur d’une *commande* indique l’*adresse du destinataire*, alors que l’émetteur d’une *réponse* spécifie sa propre adresse. Par exemple, dans le réseau X25, qui utilise le protocole LAP-B, les ETTD ont pour adresse 03, et les ETCD 01.

La zone de données est vide lorsque des trames de service sont échangées, et contient le LSDU sinon. Par conséquent, une trame LAP-B a une *taille variable* d’au moins 6 octets.

Le *FCS* (Frame Check Sequence) est calculé par l’émetteur de la trame. C’est un CRC de polynôme générateur $G(x) = x^{16} \oplus x^{15} \oplus x^2 \oplus 1$. Il y a plus de 2 termes, donc les erreurs simples sont détectables. On peut écrire $G(x)$ sous la forme $G(x) = (x \oplus 1)(x^{15} \oplus x \oplus 1)$. On en déduit que les erreurs en nombre impair peuvent être détectées (multiple de $x \oplus 1$) ainsi que les erreurs doubles (facteur irréductible de 3 termes).

1.2.2 Types des trames

Les trames peuvent être de plusieurs *types*, comme indiqué dans le champ *contrôle* :

- les *trames d’information* (I) servent au *transfert des LSDU* ;
- les *trames de supervision* (S) sont utilisées pour réaliser les *commandes de supervision de la liaison*, telles que l’*accusé de réception*, la *demande de retransmission* et la *suspension temporaire* de transmission ;
- les *trames non numérotées* (U) permettent de réaliser les *fonctions de commande de la liaison* ainsi que le transfert d’information non numérotée.

1.2.3 Champ de contrôle

La structure du *champ de contrôle* dépend du type de trame :

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Trame I	N_r			P/F	N_s			0
Trame S	N_r			P/F	S_1	S_0	0	1
Trame U	U_4	U_3	U_2	P/F	U_1	U_0	1	1

FIGURE 1.2 – Champ de contrôle d’une trame LAP-B

Dans une trame d’information, les 3 bits N_s indiquent un numéro de trame modulo 8, et les 3 bits

N_r , un numéro d'accusé de réception acquittant une trame reçue antérieurement.

Dans une trame de supervision, les bits S_1 et S_0 peuvent prendre 3 valeurs :

- 00 : RR (Received and Ready). Une trame RR permet d'acquitter les messages déjà reçus et indiquer que l'on est prêt à recevoir une autre trame ;
- 01 : RNR (Received and Not Ready). Une trame RNR indique que le récepteur ne peut temporairement plus recevoir de trame d'information ;
- 11 : REJ (REJect) indique que la trame reçue n'a pas le numéro attendu. Elle indique aussi quel est le numéro attendu, accusant ainsi réception de toutes les trames précédentes.

Les trames U, de gestion, utilisent le bit P/F. Ce bit a deux significations différentes selon les cas. Lorsque ce bit vaut 1, on dit qu'il est positionné. Sa valeur est P si la trame est une trame de commande, F si c'est une trame de réponse. Les trames U ont des types différents selon la valeurs des bits U_4 à U_0 :

- 01111 : SABM (Set Asynchronous Balanced Mode) demande l'établissement d'une liaison en mode connecté ;
- 01100 : UA (Unnumbered Acknowledge) confirme une connexion ou une déconnexion ;
- 01011 : DISC (DISConnect) effectue une demande de déconnexion ;
- 11011 : FRMR (FRaMe Reject) est utilisée pour rejeter une trame en cas d'erreur.

1.2.4 Paramètres et variables d'état

L'ensemble des valeurs entrant en jeu dans les échanges du protocole LAP-B se divise en une *partie fixe* et une *partie variable*.

La partie fixe est identique aux deux extrémités de la liaison. Elle comporte, outre le *polynôme générateur*, un *délaï de garde* T_1 , un *délaï d'acquïttement* (T_2), un *délaï d'établissement de la liaison* (T_3), la *taille maximale* d'une trame (N_1), le *nombre maximum de réémissions* d'une trame (N_2) et la *taille w de la fenêtre d'anticipation*.

Le *délaï de garde* T_1 est armé à l'émission d'une trame I. Si l'acquïttement de la trame n'arrive pas dans le *délaï* T_1 , l'émetteur réémet automatiquement la trame.

À la réception d'une trame I, la station doit émettre un acquïttement dans un *délaï* égal au *délaï d'acquïttement* T_2 .

Une liaison de données LAP-B doit faire l'objet d'une demande d'établissement de connexion par une des deux entités qui communiquent, par l'envoi d'une trame U. Après plusieurs tentatives d'établissement de la connexion sans réception d'acquïttement durant un *délaï d'établissement de la connexion* T_3 , l'entité interrompt ses tentatives d'ouverture de connexion.

La partie variable comprend le *numéro de la prochaine trame d'information à émettre* ($V(s)$), le *numéro de la prochaine trame à recevoir* ($V(r)$) et le *numéro du dernier acquïttement reçu* ($DN(r)$). À l'établissement de la liaison, les valeurs de ces trois variables sont mises à zéro : $V(s) = V(r) = DN(r) = 0$.

Le champ N_s d'une trame d'information indique son numéro. Ce numéro étant codé sur 3 bits, on peut envoyer au maximum 8 trames avant de recevoir un accusé de réception. Si plus de trames étaient envoyées (avant d'être acquïtées), plusieurs pourraient avoir le même numéro. On ne saurait donc pas, lors de la réception d'un accusé ultérieur, laquelle des trames de numéro identique est concernée.

La variable d'état $V(s)$ indique le numéro de la prochaine trame d'information à envoyer. Cette variable est incrémentée lors de l'émission d'une trame d'information.

Le champ N_r présent dans les trames d'information et de supervision indique le numéro de la prochaine trame attendue. Par conséquent, l'émetteur de la trame contenant N_r accuse réception de toutes les trames de numéro inférieur, c'est-à-dire qu'il a reçu toutes les trames jusqu'à celle de numéro $N_r - 1$.

La variable d'état $V(r)$ indique le numéro de la prochaine trame d'information à recevoir. Cette variable est incrémentée à la réception de la trame I attendue si celle-ci ne comporte pas d'erreur.

1.2.5 Séquencement des trames

L'émission et la réception d'une trame peuvent comporter plusieurs actions. Détaillons les règles principales d'échange des trames :

Émission d'une trame d'information I :

- la trame $I_{V(s),V(r)}$ est envoyée ;
- cette trame est mémorisée, pour une éventuelle retransmission ultérieure ;
- le numéro de la prochaine trame à émettre est incrémenté. Il dépend de la taille w de la fenêtre d'émission : $V(s) = (V(s) + 1) \bmod w$;
- le temporisateur T_1 , correspondant au délai de garde, est armé.

Réception d'une trame d'information $I_{x,y}$:

- si une erreur est détectée, la trame reçue est ignorée ;
- si la trame reçue n'est pas valide, on la rejette en envoyant une trame U de gestion de type FRMR ;
- si $x \neq V(r)$, la trame reçue n'a pas le numéro attendu. Par conséquent, elle est rejetée en envoyant une trame S de supervision de type $REJ_{V(r)}$;
- la trame d'information reçue est traitée ;
- le numéro de la prochaine trame à recevoir est incrémenté. Il dépend de la taille w de la fenêtre d'émission : $V(r) = (V(r) + 1) \bmod w$;
- si $DN(r) \leq y < V(s)$ la trame d'information reçue acquitte toutes celles envoyées entre $DN(r)$ (la dernière déjà acquittée) et y . Par conséquent, tous les temporisateurs T_1 associés aux trames entre $DN(r)$ et y sont désarmés, et la valeur de $DN(r)$ est mise à jour : $DN(r) = y$.

Réception d'une trame de supervision S de type RR_y :

- si $DN(r) \leq y < V(s)$ la trame de supervision reçue acquitte toutes celles envoyées entre $DN(r)$ et y . Les temporisateurs T_1 sont désarmés, et $DN(r) = y$.

Réception d'une trame de supervision S de type REJ_y :

- si $DN(r) \leq y < V(s)$ la trame de supervision reçue acquitte toutes celles envoyées entre $DN(r)$ et y . Tous les temporisateurs T_1 associés sont désarmés, et $DN(r) = y$. De plus, la trame indique un rejet, suite à la réception d'une trame de numéro non attendu. On renvoie alors toutes les trames dont le numéro est compris entre y et $V(s)$. Pour cela, on change la valeur de $V(s)$: $V(s) = y$.

Scénario d'échange : On peut représenter les échanges de trames par un chronogramme sur lequel on note également les valeurs des variables $V(s)$, $V(r)$ et $DN(r)$ pour l'émetteur et le récepteur.

Soit l'échange représenté dans la figure ?? . Il comporte une connexion, l'envoi de trois trames d'information (avec une fenêtre en permettant plus), la perte de l'une d'elles, et enfin, lorsque toutes les trames sont arrivées, la déconnexion. Seules les variables modifiées dans ce scénario sont représentées.

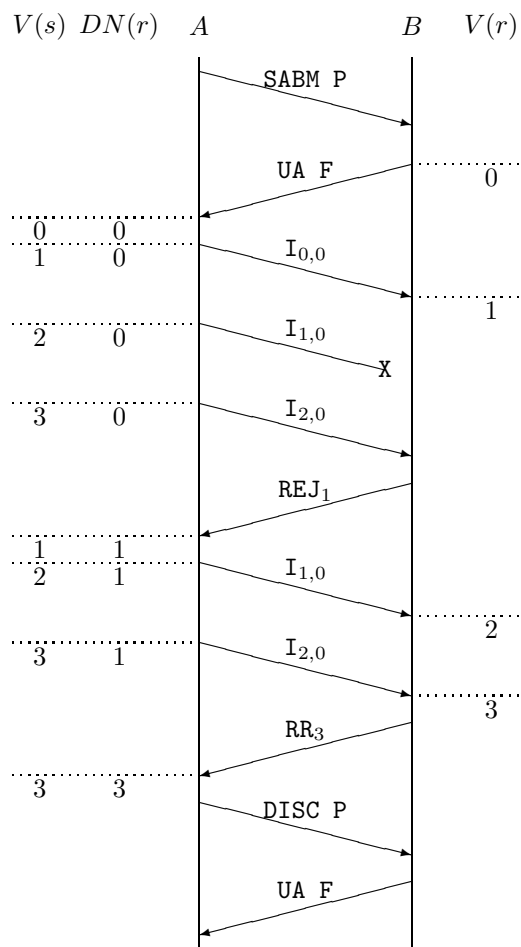


FIGURE 1.3 – Scénario d'échange LAP-B

Chapitre 2

Réseaux locaux

Un réseau local permet de relier un ensemble de machines réparties sur une zone géographique limitée.

2.1 Couches LLC et MAC

L'IEEE¹ a établi une norme pour les couches basses (1 et 2) des réseaux. Il a en particulier divisé la couche liaison de données en *deux sous-couches* :

- *LLC* : Logical Link Control
- *MAC* : Medium Access Control

Plusieurs protocoles existent pour la couche MAC, pour des méthodes d'accès et des supports physiques différents. La couche MAC offre des fonctions à la couche LLC, et fournit les méthodes d'accès au matériel. Les normes MAC les plus classiques sont :

802.3 Carrier Sense Multiple Access/Collision Detection method CSMA/CD (Ethernet)

802.4 Token-passing bus access method

802.5 Token Ring Access Method

802.6 DQDB Access Method

802.7 Broadband LAN

802.11 Wireless LAN

La couche LLC est décrite par la norme IEEE 802.2. Elle offre à la couche réseau un service de transfert de paquets tout en masquant l'hétérogénéité des méthodes d'accès.

2.2 Couche LLC (norme IEEE 802.2)

La norme LLC offre 3 types de services avec/sans connexion et avec/sans acquittement.

2.2.1 Les 3 types de services LLC

Le service *LLC1* fournit un mode de transfert *sans connexion* et *sans acquittement*. Chaque LPDU est transmise dans une *trame UI* (trame d'information non numérotée).

Le service *LLC2* fournit un mode de transfert *avec connexion*.

Le service *LLC3* effectue des transferts *sans connexion* mais *avec acquittement*. Chaque LPDU est acquittée et l'émetteur attend l'accusé de réception avant d'émettre la trame suivante. LLC3 ne peut être utilisé qu'avec une connexion point-à-point.

1. Institute of Electrical and Electronics Engineers

2.2.2 Structure des trames LLC

La couche LLC forme, à partir du NPDU transmis par la couche réseau, un LPDU. Ce LPDU est ensuite envoyé à la couche MAC qui lui ajoutera ses propres champs pour constituer la trame MAC.

La structure d'une trame LLC est décrite dans la figure ??.

DSAP 1o	SSAP 1o	Contrôle 1o ou 2o	LSDU
------------	------------	----------------------	------

FIGURE 2.1 – Structure d'une trame LLC

Les champs *DSAP* (Destination SAP) et *SSAP* (Source SAP) indiquent les adresses source et destination. Certains des bits qui les composent ont une fonction particulière. Le schéma des DSAP et SSAP se trouve dans la figure ??.

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
DSAP	SAP						I/G	L/U
SSAP	SAP						C/R	L/U

FIGURE 2.2 – Structure des adresses DSAP et SSAP

Le bit I/G du DSAP indique si l'adresse est individuelle ou groupée (LLC1 peut faire de la diffusion). Le bit C/R du SSAP vaut 0 lorsque l'on envoie une commande et 1 pour une réponse. Enfin, le bit L/U, présent à la fois dans le DSAP et le SSAP, indique si l'adresse est locale ou universelle.

Le *champ de contrôle* détermine le type de la trame. Le fonctionnement de LLC et la structure de son champ de contrôle sont similaires à ceux de LAP-B (voir figure ??). Les trames d'information (I) et de supervision (S) ont un champ de contrôle sur 2o alors que les trames de gestion U n'utilisent qu'1o.

	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8
Trame I	N_r							P/F
Trame S	N_r							P/F
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Trame I	N_s							0
Trame S	0	0	0	0	S_1	S_0	0	1
Trame U	U_4	U_3	U_2	P/F	U_1	U_0	1	1

FIGURE 2.3 – Champ de contrôle d'une trame LLC

Seul LLC2 utilise les trames d'information et de supervision. Ces dernières sont les mêmes que pour le protocole LAPB (voir section ??, page ??), c'est-à-dire RR, RNR et REJ.

Différentes trames U sont utilisées par les trois types de LLC. Elles sont indiquées dans le tableau ??.

Le fonctionnement de LLC2 est similaire à celui de LAP-B.

2.3 Sous-couche SNAP

Les SAP sont codés sur seulement 6 bits. La *sous-couche SNAP* (Sub-Network Access Protocol) permet de s'affranchir de cette limite en offrant une encapsulation supplémentaire entre la couche 3 et la couche LLC. L'entête de la couche SNAP est composé de 5 octets :

Type	Contrôle	Nom	Signification
LLC1	000 P 0011	UI	information non numérotée
	101 P/F 1111	XID	échange d'identification
	111 P/F 0011	TEST	test
LLC2	001 P 1111	SABME	demande de connexion
	011 F 0011	UA	acquittement non numéroté
	010 P 0011	DISC	déconnexion
	000 F 1111	DM	mode déconnecté
	100 F 0011	FRMR	rejet de trame
LLC3	011 P/F 0111	AC0	information non numérotée, séquence 0
	111 P/F 0111	AC1	information non numérotée, séquence 1

TABLE 2.1 – Trames U dans LLC

- 3 octets pour le champ OUI (*Organisational Unit Identifier*) qui spécifie un fabricant, une norme. . .
- 2 octets pour le champ *code* qui spécifie le SAP de niveau 3.

2.4 Accès au support

Il se peut, lorsque l'on transmet une trame sur le support physique, qu'une autre trame soit en train de se propager. D'où un risque de *collision* entre trames. Les méthodes d'accès au support visent à gérer le problème de la collision, selon deux approches principales :

- l'*approche optimiste* consiste à envoyer la trame, détecter si une collision a eu lieu, et si c'est le cas, appliquer une méthode de résolution de conflits. Cette méthode met en œuvre un accès au support par *compétition*. Il est impossible de prédire l'arrivée d'une trame : l'accès est dit *non déterministe* ou aléatoire ;
- l'*approche pessimiste* a pour principe de ne donner à chaque machine le droit exclusif d'émettre que pendant un temps limité. Une telle technique nécessite un mécanisme de négociation du droit d'émission. À un instant donné, seule une machine peut accéder au support, d'où un accès *déterministe*.

2.4.1 Token Ring (norme IEEE 802.5)

Le réseau *Token Ring* a été introduit par IBM en 1982, avec un débit de $4Mb/s$, puis a été normalisé par l'IEEE en 1985. La topologie du réseau est en anneau. Chaque station est reliée à la suivante par une liaison unidirectionnelle.

Le principe d'accès au support est régi par un système de possession de *jeton*. Le Token Ring utilise donc une approche pessimiste. Le jeton circule en permanence sur l'anneau. Lorsqu'une station souhaite émettre une trame, elle *capture le jeton* au passage, puis émet sa trame. Lorsque la trame revient à l'émetteur, soit le jeton, soit une autre trame est mis en circulation. Outre ce principe général de fonctionnement, il y a un temps limite de détention du jeton par une même machine ainsi qu'un mécanisme de priorités.

Le champ de contrôle de chaque trame a le format indiqué figure ??, où :

- P est le *niveau de priorité* ;
- T indique si la trame correspond à la circulation d'un *jeton* ($T = 0$) ou à des données ($T = 1$) ;
- M est un *bit de surveillance* ;
- R est un *niveau de réservation*. Il permet de gérer les priorités.

Chaque machine M_i gère 2 variables initialisées à -1 ("pas de valeur") :

JetonCréé mémorise la priorité du jeton créé par cette machine (s'il existe) ;

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
R			M	T	P		

FIGURE 2.4 – Champ de contrôle d’une trame Token Ring

JetonCapturé mémorise la valeur du jeton capturé par cette machine (s’il existe).

Chaque trame est représentée par son triplet (P, T, R) . Initialement, le jeton $(0, 0, 0)$ circule sur le réseau. L’algorithme ?? est utilisé lors de la réception d’une trame jeton, et l’algorithme ?? lors de la réception d’une trame de données.

Algorithme 2.1 : Réception d’un jeton $(P, 0, R)$ par M_i

```

début
   $P_i \leftarrow$  priorité maximale de trame à émettre par  $M_i$ 
  si  $P_i \geq P$  alors
    /* le jeton a une priorité inférieure, donc  $M_i$  peut émettre */
    envoyer  $(P, 1, 0)$ 
  sinon
    si  $P = \text{JetonCréé}$  alors
      /*  $M_i$  a créé le jeton et constate que ce n’est plus le niveau de
      priorité requis.  $M_i$  retire son jeton de la circulation */
      si  $R > \text{JetonCapturé}$  alors
        /*  $M_i$  crée un nouveau jeton de priorité  $R$  */
        envoyer  $(R, 0, 0)$ 
        JetonCréé  $\leftarrow R$ 
      sinon
        /*  $M_i$  remet en circulation le jeton capturé */
        envoyer  $(\text{JetonCapturé}, 0, R)$ 
        JetonCapturé  $\leftarrow -1$ 
        JetonCréé  $\leftarrow -1$ 
      finsi
    sinon
      /*  $M_i$  retransmet le jeton tout en réservant */
      envoyer  $(P, 0, \max(R, P_i))$ 
    finsi
  finsi
fin

```

Algorithme 2.2 : Réception d'une trame $(P, 1, R)$ par M_i

```
début
  si  $M_i$  est l'émetteur de la trame alors
    /* la trame est retirée de la circulation */
    si  $R > P$  alors
      /* un niveau de priorité plus grand a été demandé */
      si  $P = \text{JetonCréé}$  alors
        /*  $M_i$  crée un nouveau jeton */
        envoyer  $(\max(R, P_i), 0, 0)$ 
        JetonCréé  $\leftarrow \max(R, P_i)$ 
      sinon
        /*  $M_i$  capture le jeton de niveau  $P$  */
        JetonCapturé  $\leftarrow P$ 
        envoyer  $(\max(R, P_i), 0, 0)$ 
        JetonCréé  $\leftarrow \max(R, P_i)$ 
      finsi
    sinon
      /*  $M_i$  essaie d'émettre */
      si  $P_i \geq P$  alors
        /*  $M_i$  émet sa trame */
        envoyer  $(P, 1, 0)$ 
      sinon
        /*  $M_i$  transmet le jeton */
        envoyer  $(P, 0, \max(R, P_i))$ 
      finsi
    finsi
  sinon
    | envoyer  $(P, 1, \max(R, P_i))$ 
  finsi
fin
```

Exemple : Soient 3 machines M_1 , M_2 et M_3 souhaitant chacune émettre une trame, de priorités respectives 0, 4 et 6. Le déroulement des algorithmes est décrit dans le tableau ?? où dans chaque case figurent sur la première ligne le triplet (P, T, R) reçu et sur la seconde le couple $(\text{JetonCapturé}, \text{JetonCréé})$ après émission.

2.4.2 CSMA/CD (norme IEEE 802.3)

La méthode *CSMA/CD* (Carrier Sense Multiple Access/Collision Detection) suit une approche optimiste :

- on écoute la porteuse ;
- s'il n'y a pas de trafic sur le réseau, la trame est envoyée. Si elle entre en collision, la trame est renvoyée ultérieurement.

La technique doit garantir qu'une machine puisse d'une part détecter les collisions des trames qu'elle a envoyées, et d'autre part éviter qu'une collision se produise à nouveau lors d'une retransmission.

Soient les temps suivants, représentés sur la figure ?? :

- t_1 : début de l'émission d'une trame par la machine A ;
- $t_p = t_2 - t_1$: délai de propagation de la trame émise par A ;
- t_3 : fin de l'émission de la trame par A ;

M_1 $P_1 = 0$	M_2 $P_2 = 4$	M_3 $P_3 = 6$
(0, 0, 0) (-1, -1)	(0, 1, 0) (-1, -1)	(0, 1, 4) (-1, -1)
(0, 1, 6) (0, 6)	(6, 0, 0) (-1, -1)	(6, 0, 4) (-1, -1)
(6, 1, 0) (0, 6)	(6, 1, 0) (-1, -1)	(6, 1, 4) (-1, -1)
(6, 0, 4) (0, 4)	(4, 0, 0) (-1, -1)	(4, 1, 0) (-1, -1)
(4, 1, 0) (0, 4)	(4, 1, 0) (-1, -1)	(4, 0, 0) (-1, -1)
(4, 0, 0) (-1, -1)	(0, 0, 0) (-1, -1)	(0, 0, 0) (-1, -1)

TABLE 2.2 – Exemple de fonctionnement du Token Ring

- $t_t = t_3 - t_1$: temps de transmission de la trame envoyée par A ;
- t_4 : début de l'émission d'une trame par la machine B, plus son temps de propagation.

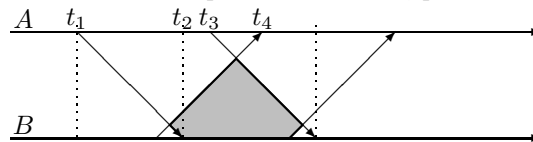


FIGURE 2.5 – Collision entre deux trames

Lorsque la machine B décide d'émettre, elle écoute la porteuse. Ne détectant aucune trame, elle envoie la sienne. Il y a ensuite collision entre les trames envoyées par les machines A et B (zone grisée dans la figure ??).

Propriété 1 Pour détecter les collisions, le temps de transmission d'une trame doit être supérieur au double du temps de propagation de la trame :

$$t_t > 2t_p$$

Preuve : Pour pouvoir détecter une collision, il suffit de garantir que $t_3 > t_4$. Pour que B puisse envoyer sa trame, il faut qu'il commence à l'émettre au plus tard en t_2 . Par conséquent, $t_4 \leq t_2 + t_p$, d'où $t_3 > t_2 + t_p$. Or $t_2 = t_1 + t_p$. Donc $t_3 > t_1 + 2t_p$, c'est-à-dire $t_3 - t_1 = t_t > 2t_p$. \diamond

La propriété ?? doit être satisfaite par les deux stations les plus éloignées sur le réseau.

Définition 1 (DAR) Le délai d'aller-retour (DAR) est le double du temps de propagation entre les deux stations les plus éloignées du réseau. C'est une caractéristique du réseau.

Pour que la propriété ?? de détection de collisions soit satisfaite, il faut imposer une *taille minimale* F_{min} aux trames à envoyer.

Définition 2 (bourrage) Si la taille d'une trame à envoyer est inférieure à la taille minimale F_{min} nécessaire pour assurer la détection de collision, la trame est complétée par des données dites de bourrage.

Il faut alors utiliser des mécanismes permettant de distinguer les données utiles des données de bourrage.

En cas de collision, la trame est retransmise ultérieurement. Le *délai de retransmission* ne doit pas toujours être le même. En effet, dans ce cas, les trames entrées en collision seraient retransmises avec un décalage identique à celui que l'on avait initialement et seraient de nouveau en collision. Pour éviter cela, un nombre *aléatoire* est utilisé dans le calcul du délai de retransmission t_r . L'*algorithme BEB* (Binary Exponential Backoff) est utilisé pour ce calcul (algorithme ??).

Algorithme 2.3 : Binary Exponential Backoff

```

début
   $k \leftarrow$  nombre de collisions successives de la trame
  si  $k > 16$  alors
    | réseau déclaré hors service
  sinon
    | si  $k \geq 10$  alors
    | |  $k \leftarrow 10$ 
    | finsi
  finsi
   $N \in [0, 2^k - 1]$  (nombre aléatoire)
   $t_r \leftarrow N \times DAR$ ;
fin

```

Lorsque l'on utilise l'algorithme BEB, le délai de retransmission t_r calculé dépend à la fois du nombre de collisions de la trame et du DAR.

2.5 Ethernet (couche MAC 802.3)

Ethernet a été introduit par Xerox en 1973. En 1975, Xerox, DEC et Intel entament une normalisation. Enfin, Ethernet devient en 1985 la *norme IEEE 802.3*. Ethernet utilise la méthode d'accès CSMA/CD. L'*Ethernet II* encapsule directement le NPDU (au lieu du LLC).

2.5.1 Structure des trames Ethernet

Une trame Ethernet 802.3 encapsule une trame LLC. La structure d'une trame Ethernet est décrite dans la figure ??.

Préambule	SFD	DA	SA	DL/EType	données	Bourrage	FCS
7o	1o	6o	6o	2o	LLC/NPDU	0-46o	4o

FIGURE 2.6 – Structure d'une trame Ethernet

Le *préambule* est composé de 7 octets valant chacun 1010 1010.

Le *SFD (Start Frame Delimiter)* est l'octet 1010 1011. Il ne diffère des octets du préambule que par son dernier bit. Le SFD annonce le début de la trame Ethernet.

L'*adresse de destination DA* est l'adresse MAC de l'entité destinataire. De même, l'*adresse source SA* est l'adresse MAC de l'entité émettrice. Les adresses sont codées sur 6 octets habituellement représentés par 6 nombres hexadécimaux sur 2 chiffres, séparés par des *:*. Par exemple 00:04:76:23:A1:D9 est une adresse MAC. Ces nombres sont attachés à la carte réseau de la machine : les trois premiers dépendent du constructeur de la carte et les trois derniers identifient la carte elle-même. Ces numéros sont *a priori* uniques. L'adresse FF:FF:FF:FF:FF:FF, composée uniquement de bits à 1 est utilisée pour envoyer une trame en *diffusion*, c'est-à-dire la transmettre à toutes les machines connectées (*broadcast*).

Le champ *DL (Data Length)* indique, dans Ethernet 802.3, la longueur, en octets, des données encapsulées, limitée à 1.500o. Les données étant une trame LLC, elles ont une longueur variable.

Le champ DL permet donc de savoir où les données se trouvent. Dans Ethernet II, ce champ se nomme *EType* et contient le code SAP du destinataire de la trame. Pour éviter la confusion entre les trames des deux protocoles Ethernet, le numéro de SAP est toujours supérieur à 1.500. Par exemple le code hexadécimal 0800 représente IPv4.

Si la trame Ethernet, hors préambule et SFD, a une taille inférieure à 64 octets, on *rajoute des octets* à 00 pour que la taille de la trame soit 64 octets. Ces octets sont dits *de bourrage*, car ils ne servent qu'à faire du remplissage.

Enfin, la trame se termine par un *FCS (Frame Control Sequence)* permettant de détecter les erreurs éventuelles.

2.5.2 Algorithme d'émission de trame Ethernet

Chaque nœud du réseau utilise l'algorithme ?? pour émettre une trame. Le *délati d'aller-retour (DAR)* est fixé à $51,2\mu s$.

Algorithme 2.4 : Émission d'une trame Ethernet

```
début
  Formation de la trame
  Écoute de la porteuse pendant  $9,6\mu s$ 
  répéter
    si le canal est libre alors
      /* on peut émettre */
      début de l'émission
      si un conflit est détecté pendant l'émission alors
        attendre que le support se libère en brouillant la ligne
        si le nombre maximum de retransmissions est atteint alors
          | échec de la transmission
        sinon
          | attendre un délai aléatoire
        finsi
      sinon
        | succès de la transmission
      finsi
    finsi
  jusqu'à fin de transmission
fin
```

Chapitre 3

Équipements d'interconnexion

3.1 Supports physiques

Ethernet est implanté en utilisant toute une panoplie de supports et d'architectures physiques. La *codification* des supports est sous la forme D-TT-S où :

- D donne le débit en Mb/s ;
- TT indique le type de transmission ;
- S renseigne sur l'architecture physique ou sur la longueur maximale d'un segment pour l'architecture bus.

Exemple : 10Base5 indique un débit de $10Mb/s$, en bande de base, sur une architecture bus dont les segments ne dépassent pas 500m.

Les *paires torsadées* sont utilisées pour 10BaseT, 100BaseT, 1000BaseT. Elles assurent une qualité similaire à celle de la téléphonie. Le codage physique utilisé est le codage Manchester avec des niveaux de $2,5V$ et $-2,5V$. Le câble est relié à la machine par des *connecteurs* de type *RJ45*, qui comportent 8 pins. Seules 2 des 4 paires torsadées du câble sont utilisées : l'une pour la transmission et l'autre pour la réception. Pour les réseaux 100BaseT et 1000BaseT, il faut modifier le délai d'aller-retour (DAR) pour que la taille minimale d'une trame soit toujours de 64 octets. 100BaseT utilise un codage 4b/5b, et 1000BaseT un codage 4D-PAM5. Les 4 paires torsadées sont utilisées par 1000BaseT.

La *fibres optiques* est utilisée pour 10BaseF, 100BaseFX et 1000BaseSX. Les deux derniers types de supports correspondent à une architecture en étoile.

3.2 Équipements d'interconnexion

Les équipements d'interconnexion permettent de constituer des *sous-réseaux locaux*, pour pouvoir s'affranchir des limitations de distance ou de nombre de stations connectées au réseau, répartir les stations sur des sous-réseaux différents pour diverses raisons telles que la topologie, les performances ou la sécurité, et relier des réseaux de types différents.

Les équipements d'interconnexion sont variés et se situent à différents niveaux du modèle en couches OSI, comme indiqué dans la figure ??.

L'équipement de base est le *répéteur* qui connecte deux segments de réseaux pour obtenir des distances plus importantes. Le répéteur amplifie le signal reçu, le reconstitue et le retransmet. Dans un réseau 10BaseT, les stations sont connectées à un *concentrateur (hub)* dont la fonction

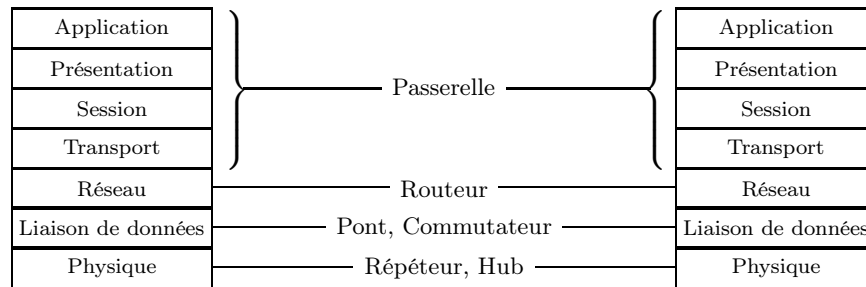


FIGURE 3.1 – Équipements d’interconnexion

est de répéter en diffusant. La distance maximale entre une station et le hub est de $100m$. On peut connecter jusqu’à 8 hubs en cascade.

3.3 Stratégies de commutation

On distingue deux types de commutateurs :

- les *commutateurs par port* : une seule station est connectée à chaque port ;
- les *commutateurs par segment* : un sous-réseau local est connecté à un port du commutateur.

Ces deux types de commutateurs permettent d’adresser respectivement une seule station ou un groupe de stations sur un port du commutateur.

De plus, les commutateurs disposent de différentes méthodes de commutation :

- la *commutation à la volée* : dès que l’adresse de destination (DA) a été analysée, la trame est envoyée à sa destination. La retransmission s’effectue pendant que l’on reçoit le reste de la trame. Le retard induit par le commutateur est alors assez faible ;
- la méthode *store and forward* : la trame est envoyée au destinataire une fois qu’elle a été complètement reçue et que le FCS a été vérifié ;
- une méthode dite *mixte* qui travaille tout d’abord à la volée, mais vérifie le FCS au passage. Si l’on détecte ainsi trop d’erreurs, la méthode à la volée est abandonnée au profit du store and forward.

3.4 Les ponts

Un *pont* permet de limiter le nombre de stations sur un segment du réseau et de dépasser les contraintes physiques telles que la longueur maximale en connectant plusieurs sous-réseaux. L’utilisation d’un pont augmente les performances locales de chaque sous-réseau en l’isolant du trafic circulant dans les sous-réseaux voisins. De plus, la sécurité est améliorée.

3.4.1 Fonctionnement d’un pont

Le pont écoute les trames circulant sur les sous-réseaux qu’il relie. Pour chaque trame il regarde l’adresse de destination. Le pont possède une *table de commutation* qui lui permet de savoir sur quel sous-réseau se trouve la station destinataire. S’il n’a pas encore l’information nécessaire, il diffuse la trame vers tous les sous-réseaux auxquels il est connecté, sauf celui dont provient la trame. Le pont dispose d’une mémoire (limitée) qui lui permet de stocker une trame jusqu’à l’avoir reçue dans son intégralité. Ensuite, il la retransmet sur un ou plusieurs sous-réseaux en fonction de l’adresse du destinataire. Cette méthode s’appelle *store and forward*.

L’utilisation d’un pont limite les risques de collisions. Les stations connectées au sous-réseau n’ont pas besoin de connaître les adresses des ports du pont. Le pont est donc qualifié de *transparent*.

3.4.2 Table de commutation

La *table de commutation* contient une ligne pour chaque station qu'elle connaît. Chaque ligne contient :

- l'adresse physique (adresse MAC) de la station ;
- le numéro du port du pont auquel est connecté le sous-réseau où la station se trouve ;
- la date de création de cette ligne d'informations.

Lorsque le pont écoute une trame, il crée une nouvelle ligne dans la table de commutation, contenant :

- l'adresse MAC source ;
- le numéro du port sur lequel l'écoute a lieu.

Les lignes de la table de commutation ont une durée de vie limitée, car l'information peut devenir obsolète, par exemple si l'on déplace des machines d'un sous-réseau à un autre. Lorsque l'on crée une ligne avec une adresse MAC déjà référencée dans la table de commutation, l'ancienne ligne est remplacée par la nouvelle.

Lorsque la topologie du réseau comporte des boucles, comme dans la figure ??, il faut assurer le bon fonctionnement des ponts. En effet, supposons qu'une trame soit émise de *A* vers *B* sur le sous-réseau local *LAN2*. Le pont 1 comprend que la machine *A* se trouve sur *LAN2*. Ne sachant pas où est *B*, il transmet la trame sur *LAN1*. Le pont 2 croit alors que *A* est sur *LAN1*. La trame va être transmise de nouveau sur *LAN2*, et ainsi de suite. Pour éviter ce problème, on choisit les chemins à emprunter de manière unique, en utilisant l'*algorithme de l'arbre couvrant*.

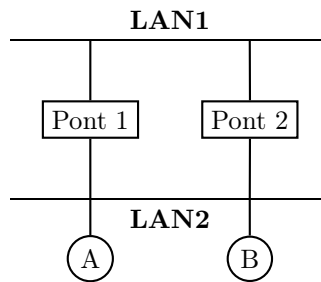


FIGURE 3.2 – Réseau comportant une boucle

3.4.3 Algorithme de l'arbre couvrant

L'*algorithme de l'arbre couvrant* (*spanning tree*) vise à choisir de manière unique un des plus courts chemins pour aller d'un sous-réseau local à un autre. Il transforme un graphe pouvant comporter des boucles en un arbre (donc sans boucle).

L'algorithme procède à l'élection d'un pont particulier, appelé *pont racine*. Pour cela, les ponts se transmettent des trames MAC spéciales, appelées *BPDU* (*Bridge Protocol Data Unit*), permettant de configurer les ponts. Une BPDU est émise par un pont sur un de ses ports. Elle est reçue et traitée par tous les autres ponts reliés au même sous-réseau local.

Définition 3 Les termes suivants sont utilisés dans l'algorithme :

- Le pont racine est le pont ayant l'identité la plus petite.
- Le port racine est le port par lequel il faut passer pour atteindre le pont racine, éventuellement en traversant plusieurs autres ponts.
- Les ports désignés sont les autres ports utilisés dans l'arbre couvrant, permettant d'atteindre des fils. Les autres ports, n'étant donc pas utilisés, sont appelés ports bloqués.

Propriété 2 Un graphe connexe admet un arbre couvrant pas nécessairement unique. L'arbre couvrant contient tous les nœuds du graphe. Le chemin entre deux nœuds dans l'arbre ouvrant est unique.

Trames BPDU

Définition 4 (BPDU) Une BPDU comporte les informations suivantes :

- adresse MAC du pont racine (sur 6 octets) ;
- distance entre l'émetteur de la BPDU et le pont racine. Cette distance est exprimée en nombre de sauts ;
- adresse MAC du pont émetteur de la BPDU ;
- numéro du port sur lequel la BPDU a été émise.

Une trame BPDU sera notée comme un quadruplet (R, D, E, P) .

Les ponts échangent des BPDUs. Ils doivent donc comparer la BPDU qu'ils ont envoyée avec celles qu'ils reçoivent pour garder la meilleure configuration. Pour cela, une comparaison des BPDUs est nécessaire.

Définition 5 (comparaison de BPDUs) Soient deux BPDUs, $BPDU_1 = (R_1, D_1, E_1, P_1)$ et $BPDU_2 = (R_2, D_2, E_2, P_2)$.

$$\begin{aligned} BPDU_1 < BPDU_2 \Leftrightarrow & (R_1 < R_2) \vee \\ & (R_1 = R_2 \wedge D_1 < D_2) \vee \\ & (R_1 = R_2 \wedge D_1 = D_2 \wedge E_1 < E_2) \vee \\ & (R_1 = R_2 \wedge D_1 = D_2 \wedge E_1 = E_2 \wedge P_1 < P_2) \end{aligned}$$

L'ordre ainsi défini sur les BPDUs n'est autre que l'ordre classique sur les quadruplets. La meilleure configuration est celle qui correspond à la BPDU la plus faible.

Fonctionnement de l'algorithme

Au départ, chaque pont ne connaît pas les autres et suppose donc qu'il est le pont racine. Il construit donc une configuration locale et la diffuse sur chacun de ses ports. Lorsqu'un pont reçoit une BPDU, il la compare à la configuration locale courante. Il en déduit la nouvelle configuration ainsi que des informations sur le pont racine et sur ses ports.

Algorithme 3.1 : arbre couvrant pour le pont i

```
début
   $config_i \leftarrow (i, 0, i)$  /* Initialisation de la configuration locale */
  répéter
    pour chaque port désigné  $p_j$  faire
      | envoyer ( $config_i, p_j$ ) sur le port  $p_j$ 
    finprch
    attendre la réception d'une BPDU
    /* Choix de la meilleure BPDU */
    soit  $(r, d, e, p)$  la meilleure BPDU reçue
    soit  $p_r$  le port sur lequel  $(r, d, e, p)$  a été reçue
    si  $(r, d + 1, e) < config_i$  alors
      |  $config_i \leftarrow (r, d + 1, i)$  /* Nouvelle configuration locale */
      |  $p_r$  devient le port racine
    finsi
    pour chaque port  $p_j \neq p_r$  sur lequel on a reçu une BPDU  $(r_j, d_j, e_j, p'_j)$  faire
      | si  $(r_j, d_j, e_j) < config_i$  alors
      | |  $p_j$  est bloqué
      | finsi
    finprch
  jusqu'à fin /* Les configurations locales ne changent plus */
fin
```

Exemple : Soit la topologie de réseaux décrite dans la figure ?? . Les numéros des ports sont indiqués à côté des sorties des ponts. Appliquons l'algorithme de l'arbre couvrant.

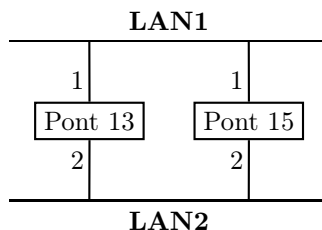


FIGURE 3.3 – Réseau comportant une boucle

Le pont 13 envoie $(13, 0, 13, 1)$ sur son port 1 et $(13, 0, 13, 2)$ sur le 2. De même, le pont 15 envoie $(15, 0, 15, 1)$ sur son port 1 et $(15, 0, 15, 2)$ sur le 2.

Pour le pont 13, la meilleure BPDU est reçue sur le port 1 : $(15, 0, 15, 1)$. La configuration locale est donc toujours $(13, 0, 13)$. Sur le port 1, la BPDU reçue est $(15, 0, 15, 1)$. Comme $(15, 0, 15) > (13, 0, 13)$, le port 1 est désigné. Sur le port 2, la BPDU reçue est $(15, 0, 15, 2)$. Comme $(15, 0, 15) > (13, 0, 13)$, le port 2 est désigné.

Pour le pont 15, la meilleure BPDU est reçue sur le port 1 : $(13, 0, 13, 1)$. La nouvelle configuration locale est donc $(13, 1, 15)$. Le port racine est le port 1. Sur le port 2, la BPDU reçue est $(13, 0, 13, 2)$. Comme $(13, 0, 13) < (13, 1, 15)$, le port 2 est bloqué.

L'arbre couvrant est présenté dans la figure ??.

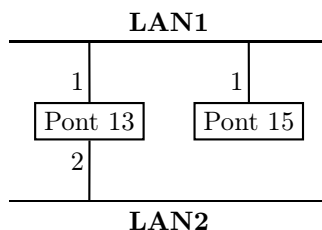


FIGURE 3.4 – Arbre couvrant du réseau de la figure ??

Chapitre 4

Réseaux Locaux Virtuels (VLAN)

4.1 Concepts

Les réseaux locaux imposent une structure statique. Toutefois, il est parfois préférable de regrouper des machines ayant des caractéristiques communes, même si celles-ci sont physiquement connectées à des réseaux locaux différents. Pour ce faire, on utilise le concept de *réseau local virtuel* (VLAN, Virtual Local Area Network).

Définition 6 (VLAN) *Un Réseau Local Virtuel (VLAN) est un regroupement logique de stations appartenant à des réseaux locaux existants.*

La figure ?? illustre un réseau comportant deux sous-réseaux locaux et trois réseaux locaux virtuels. Le réseau local LAN1 comprend les stations A, B et C, et le réseau local LAN2 les stations D, E et F. Le réseau local virtuel VLAN1 contient les stations A et B du LAN1, VLAN3 la station F du LAN2. Le réseau local virtuel VLAN2 regroupe des stations des deux réseaux locaux LAN1 et LAN2 : C, D et E.

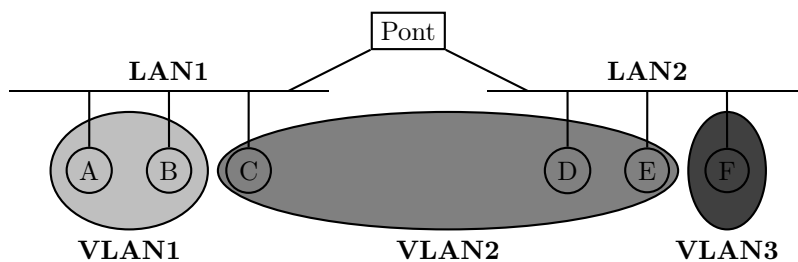


FIGURE 4.1 – LANs et VLANs

Propriété 3 *Les stations d'un VLAN communiquent comme si elles appartenait au même segment.*

Propriété 4 *Un VLAN définit un domaine de diffusion.*

Par conséquent, les messages émis en diffusion par une station d'un VLAN ne sont reçus que par les stations de ce même VLAN.

Les VLANs offrent des avantages similaires à ceux fournis par les équipements d'interconnexion, à savoir :

- l'amélioration de la bande passante ;
- une administration du réseau plus facile ;

– une plus grande sécurité.

De plus, les VLANs permettent dans une certaine mesure de gérer aisément la mobilité. Soit le réseau configuré comme dans la figure ?? . La station *S* peut également être connectée comme indiqué dans la figure ?? sans changement majeur.

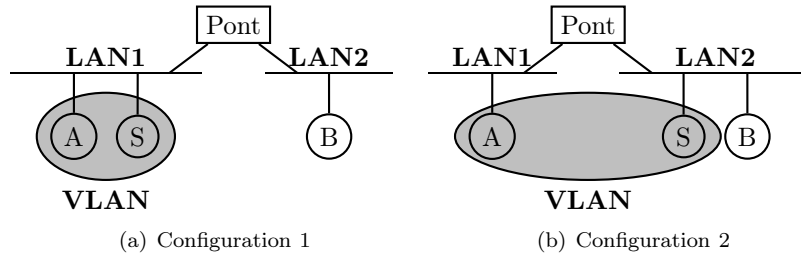


FIGURE 4.2 – VLANs et mobilité

4.2 Mise en œuvre de VLANs

4.2.1 Équipements VLAN

Plusieurs équipements d’interconnexion permettent de créer et gérer des VLANs :

- les *HUBs intelligents* sont programmables. Ils permettent d’associer un VLAN à un sous-ensemble des ports du HUB ;
- les *commutateurs LAN* ;
- les *routeurs*.

La définition des VLANs se fait par la *programmation* des équipements d’interconnexion.

4.2.2 Construction de VLANs

Pour construire un VLAN, il faut commencer par définir un critère de regroupement des stations. Cela permet de savoir quelles stations doivent appartenir à un même VLAN.

Ensuite, la mise en œuvre de ces regroupements peut s’effectuer à différents niveaux, selon l’équipement utilisé :

- niveau 1 : on utilise le numéro de port du HUB auquel la machine est connectée ;
- niveau 2 : l’adresse MAC de la carte réseau de la station est utilisée ;
- niveau 3 : l’adresse réseau (IP) de la station permet de spécifier à quel VLAN elle appartient.

Des mécanismes plus sophistiqués tels que des *règles de filtrage* permettent une définition plus souple des VLANs.

Exemple : Soit un commutateur de niveau 1 possédant 6 ports. Une assignation des ports à 3 VLANs différents peut se programmer de la manière suivante :

```
assign port 1,3 to VLAN1
assign port 4..6 to VLAN2
assign port 2 to VLAN3
```

Le VLAN1 est alors composé des ports 1 et 3, le VLAN2 des ports 4 à 6 et le VLAN3 ne contient que le port 2.

La programmation aux niveaux 2 et 3 s’effectue de manière similaire.

De nombreux équipements disposent d’applications embarquées permettant de configurer les VLANs à partir d’une interface web.

4.3 Protocole ISL

Le *protocole ISL* (Inter Switch Link) de Cisco (constructeur d'équipements d'interconnexion), permet d'étendre la définition d'un VLAN à plusieurs commutateurs.

Les commutateurs échangent alors des trames MAC encapsulant des trames ISL qui elles-même encapsulent les trames MAC reçues. L'entête de la trame ISL fournit des informations quant à l'appartenance des éléments aux réseaux locaux virtuels. La trame ISL est acheminée vers le commutateur destination. Celui-ci supprime l'entête et transmet la trame MAC ainsi obtenue (celle encapsulée par ISL) uniquement aux stations concernées.

Le format de la trame ISL est décrit dans la figure ??, sur laquelle chaque ligne a une taille de 32 bits.

DA (40 bits)			
DA (suite)	Type (4)	User (4)	SA (48 bits)
SA (suite)			
Longueur (16 bits)		Constante (24 bits)	
Constante (suite)	Constructeur (24 bits)		
Id VLAN (15 bits)	BPDU (1)	Index (16 bits)	
Réservé (16 bits)			
Trame encapsulée (8 à 196.600 bits)			
FCS (32 bits)			

FIGURE 4.3 – Format d'une trame ISL