Réseaux
Couche Liaison

E. Jeandel

Emmanuel.Jeandel at lif.univ-mrs.fr

Résumé des épisodes précédents

La couche physique nous permet une transmission brute de bits par un medium reliant une machine *A* à une machine *B*.

La couche liaison

7.6.2.1 *La couche liaison de données fournit les moyens fonctionnels et procéduraux nécessaires d'une part à la transmission en mode sans connexion entre entités de réseau et, d'autre part, en mode connexion, à l'établissement, au maintien et à la libération des connexions de liaison de données entre entités de réseau, ainsi qu'au transfert des unités de données de service de liaison de données. Une connexion de liaison de données est construite à partir d'une ou plusieurs connexions physiques.*

Plusieurs services

- Service sans connexion, sans acquittement
- Service sans connexion, avec acquittement
- Service avec connexion, avec acquittement, fiable

Plusieurs services

- Acquittement : L'emetteur sait que le message a été reçu.
- Connexion : Dialogue préalable pour se "connecter".
  - Téléphone vs Lettres vs Lettres+AR

Plusieurs services

- Envoi de fichier ?
- Multimédia ?
- VoIP ?

- Liaison cablée point-à-point ?
- Liaison sans fil ?
- Liaison modem ?

Que faut-il faire ?

- Être capable d'envoyer des messages sur le medium
- Détecter les erreurs
- Que faire s'il y a des erreurs ?

Contenu

Trames

Une donnée échangée au niveau liaison s'appelle une *trame*.

- Données + Commandes

Problème : comment délimiter les trames ?

Découpage de trames

Solutions

- Indiquer la taille de chaque trame
- Codage de début et de fin
- Reporter le problème à la couche physique

Codage de début et de fin

Mode octet

M	a	r	s	e	i	l	l	e

#	(header)	M	a	r	s	e	i	l	l	e	(footer)	#

On repère le début (resp. fin) de la trame par le caractère #.

Codage de début et de fin

Mode octet

R	o	o	m	:	#	1	3	4	9

#	(header)	R	o	o	m	:	#	1	3	4	9	(footer)	#

Codage de début et de fin

Mode octet

R	o	o	m	:	#	1	3	4	9

#	(header)	R	o	o	m	:	!	#	1	3	4	9	(footer)	#

Codage de début et de fin

Mode octet

Y	E	S	!		I		W	o	n

#	(header)	Y	E	S	!	!		I		W	o	n	(footer)	#

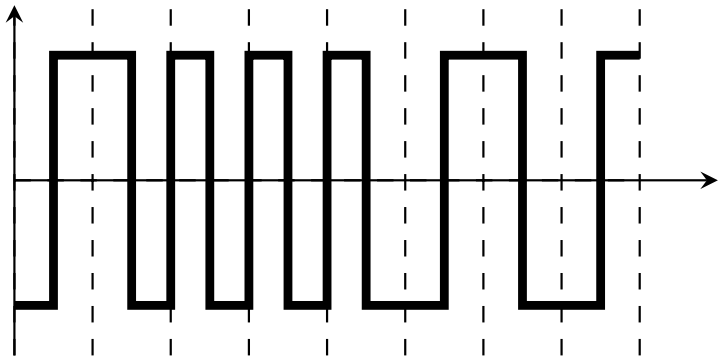
Codage de début et de fin

Mode bit

- On code le début et la fin par le fanion `01111110`
- Dans tout le texte, une suite de 5 caractères `1` consécutifs est remplacée par `111110`

Codage en couche physique

Example du Codage Manchester.



Contenu

Problématique

- La transmission dans le medium n'est pas parfaite.
- La couche physique n'offre aucune garantie.

Erreurs

- Un code de paramètre (n, m) est un procédé qui transforme n bits en $m > n$ bits. On note $f(x)$ la fonction.
- En général, les codes ajoutent aux n bits existants, $m - n$ bits de contrôle. Ce n'est pas toujours le cas.
- Un mot x est transmis avec k erreurs si le résultat de la transmission contient k bits différents de leur valeur initiale.
- Un code détecte k erreurs si, lorsqu'un mot $f(x)$ est transmis avec moins de k erreurs, on peut se rendre compte que le message est erroné.
- Un code corrige k erreurs si, lorsqu'un mot $f(x)$ est transmis avec moins de k erreurs, on est capable de retrouver $f(x)$.

Premier exemple

Bit de parité

- On ajoute à un message de n bits un bit de parité
- Le bit vaut 0 s'il y a un nombre pair de 1 dans le message, et 1 sinon.
- Detecte une erreur.
- N'en corrige aucune.

Distance de Hamming

- La distance de Hamming entre deux mots x et y est le nombre de bits sur lesquels ils diffèrent.

$$\begin{aligned} d(00010, 01010) &= \\ d(00011010, 00001011) &= \end{aligned}$$

- On note $B(x, k)$ la boule de centre x et de rayon k : c'est l'ensemble des mots qui diffèrent de x par moins de k bits.

$$|B(x, k)| = C_m^0 + C_m^1 + \cdots + C_m^k$$

Distance de Hamming

- La distance de Hamming entre deux mots x et y est le nombre de bits sur lesquels ils diffèrent.

$$\begin{aligned} d(00010, 01010) &= 1 \\ d(000110101, 000010111) &= 2 \end{aligned}$$

- On note $B(x, k)$ la boule de centre x et de rayon k : c'est l'ensemble des mots qui diffèrent de x par moins de k bits.

$$|B(x, k)| = C_m^0 + C_m^1 + \cdots + C_m^k$$

Distance de Hamming

- La distance de Hamming entre deux mots x et y est le nombre de bits sur lesquels ils diffèrent.

$$\begin{aligned} d(00010, 01010) &= 1 \\ d(000110100, 000010111) &= 2 \end{aligned}$$

- On note $B(x, k)$ la boule de centre x et de rayon k : c'est l'ensemble des mots qui diffèrent de x par moins de k bits.

$$|B(x, k)| = C_m^0 + C_m^1 + \cdots + C_m^k$$

Distance de Hamming

Reformulations

- Un code détecte k erreurs si pour tout $x \neq y$, $B(x, k)$ ne contient pas y.
- Un code corrige k erreurs si pour tout $x \neq y$, $B(x, k)$ et $B(y, k)$ ne s'intersectent pas.
- Un code détecte k erreurs si la distance entre deux mots de code est supérieure à $k + 1$
- Un code corrige k erreurs si la distance entre deux mots de code est supérieure à $2k + 1$

Problématique

Compromis à trouver :

- Efficacité du code : rapport n/m
- Propriétés de correction du code

Limite théorique :

$$2^n(C_m^0 + C_m^1 + \cdots + C_m^k) \leq 2^m$$

Pour $k = 1$:

$$2^n(m+1) \leq 2^m$$

Codes de Hamming

- Code optimal corrigeant une seule erreur
- Les bits de contrôle sont situés aux puissances de 2.
- Le bit de contrôle en position 2^i est la parité de tous les bits ayant 2^i dans leur décomposition en binaire.

a	b	c	d

Codes de Hamming

- Code optimal corrigeant une seule erreur
- Les bits de contrôle sont situés aux puissances de 2.
- Le bit de contrôle en position 2^i est la parité de tous les bits ayant 2^i dans leur décomposition en binaire.

a	b	c	d

1	2	3	4	5	6	7
x	y	a	z	b	c	d

Codes de Hamming

- Code optimal corrigeant une seule erreur
- Les bits de contrôle sont situés aux puissances de 2.
- Le bit de contrôle en position 2^i est la parité de tous les bits ayant 2^i dans leur décomposition en binaire.

a	b	c	d

001	010	011	100	101	110	111
x	y	a	z	b	c	d

Codes de Hamming

- Code optimal corrigeant une seule erreur
- Les bits de contrôle sont situés aux puissances de 2.
- Le bit de contrôle en position 2^i est la parité de tous les bits ayant 2^i dans leur décomposition en binaire.

a	b	c	d

001	010	011	100	101	110	111
x	y	a	z	b	c	d

$$\begin{aligned} x &= a+b+d \\ y &= a+c+d \\ z &= b+c+d \end{aligned}$$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- On représente le mot a_i par le polynôme $\sum a_i X^i$.
- On calcule le résultat modulo un polynôme P (et modulo 2).
- Exemple : Calcul du CRC de 110011110 pour le polynôme $x^5 + x^2 + 1$ (CRC-5-USB)

$$x^8 \quad +x^7 \quad +0 \quad +0 \quad +x^4 \quad +x^3 \quad +x^2 \quad +x \quad +0 \;\Big|\; x^5 + x^2 + 1$$

- Résultat : $x^2 + x + 1$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- On représente le mot a_i par le polynôme $\sum a_i X^i$.
- On calcule le résultat modulo un polynôme P (et modulo 2).
- Exemple : Calcul du CRC de 110011110 pour le polynôme $x^5 + x^2 + 1$ (CRC-5-USB)

$$x^8 \quad +x^7 \quad +0 \quad +0 \quad +x^4 \quad +x^3 \quad +x^2 \quad +x \quad +0 \;\Big|\; \underline{x^5 + x^2 + 1}$$

- Résultat : $x^2 + x + 1$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- On représente le mot a_i par le polynôme $\sum a_i X^i$.
- On calcule le résultat modulo un polynôme P (et modulo 2).
- Exemple : Calcul du CRC de 110011110 pour le polynôme $x^5 + x^2 + 1$ (CRC-5-USB)

$$
\begin{array}{cccccccccc|l}
x^8 & +x^7 & +0 & +0 & +x^4 & +x^3 & +x^2 & +x & +0 & x^5+x^2+1 \\
\cline{10-10}
x^8 & & & x^5 & & x^3 & & & & x^3 \\
\cline{1-9}
 & x^7 & 0 & x^5 & x^4 & 0 & x^2 & x & 0 &
\end{array}
$$

- Résultat : $x^2 + x + 1$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- On représente le mot a_i par le polynôme $\sum a_i X^i$.
- On calcule le résultat modulo un polynôme P (et modulo 2).
- Exemple : Calcul du CRC de 110011110 pour le polynôme $x^5 + x^2 + 1$ (CRC-5-USB)

$$
\begin{array}{cccccccccc|l}
x^8 & +x^7 & +0 & +0 & +x^4 & +x^3 & +x^2 & +x & +0 & x^5+x^2+1 \\
\cline{10-10}
x^8 & & & x^5 & & x^3 & & & & x^3+x^2 \\
\cline{1-9}
 & x^7 & 0 & x^5 & x^4 & 0 & x^2 & x & 0 & \\
 & x^7 & & & x^4 & & x^2 & & & \\
\cline{1-7}
 & & & x^5 & 0 & 0 & 0 & x & 0 &
\end{array}
$$

- Résultat : $x^2 + x + 1$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- On représente le mot a_i par le polynôme $\sum a_i X^i$.
- On calcule le résultat modulo un polynôme P (et modulo 2).
- Exemple : Calcul du CRC de 110011110 pour le polynôme $x^5 + x^2 + 1$ (CRC-5-USB)

$$
\begin{array}{llllllllll|l}
x^8 & +x^7 & +0 & +0 & +x^4 & +x^3 & +x^2 & +x & +0 & x^5+x^2+1 \\
\cline{10-10}
x^8 & & & x^5 & & x^3 & & & & x^3+x^2+1 \\
\cline{1-9}
 & x^7 & 0 & x^5 & x^4 & 0 & x^2 & x & 0 & \\
 & x^7 & & & x^4 & & x^2 & & & \\
\cline{1-8}
 & & & x^5 & 0 & 0 & 0 & x & 0 & \\
 & & & x^5 & & & x^2 & & 1 & \\
\cline{1-8}
 & & & & & & x^2 & x & 1 &
\end{array}
$$

- Résultat : $x^2 + x + 1$

Codes Polynomiaux

CRC (Cyclic Redundancy Check)

- Faciles à calculer par un circuit électronique
- Des petits polynomes peuvent détecter beaucoup d'erreurs.

Contenu

Problématique

Transmettre un message de A à B, en tenant compte

- Des erreurs
- Des pertes
- Du temps de traitement (couches supérieures...)

Protocole minimal

- *A* envoie toutes ses trames dès que possible
  `while(1) physical.put(network.nextframe());`
- *B* reçoit toutes les trames dès que possible
  `while(1) network.handleframe(physical.get());`

Problèmes

- *B* doit être capable de traiter la trame i avant que la trame $i+1$ ne soit arrivée.
- Ne marche pas si perte ou erreurs.

Protocole avec attente

- *A* attend que *B* ait traitée la trame avant d'envoyer la suivante

```
A: while (1)
 physical.put(network.nextframe());
 wait_acknowledgement();

B: while(1)
 network.handleframe(physical.get());
 acknowledge();
```

Pour prévenir *A*, le récepteur *B* peut utiliser le même medium :

```
acknowledge() { physical.put(frame_ack);}
```

ou un tout autre moyen/réseau/medium.

- Ne gère toujours pas les erreurs

Protocole avec erreurs

Essai 1

- *A* attend que *B* ait validé la trame avant d'envoyer la suivante

```
A: while (1)
   frame = network.frame[i];
   physical.put(frame);
   correct = physical.get();
   if (correct) i++;

B: while(1)
   frame = physical.get();
   if correct(frame)
    physical.put(1);
    network.handleframe(frame);
   else physical.put(0);
```

Protocole avec erreurs

Essai 2

On suppose qu'une trame non correcte (avec une CRC invalide) n'est vue ni de *A* ni de *B*

- Idem, mais *A* renvoie la trame si un certain délai est passé

```
A: while (1)
     frame = network.frame[i];
     physical.put(frame);
     received = physical.get() // timeout();
     if (received && !timeout) i++;

B: while(1)
     frame = physical.get();
     physical.put(1);
     network.handleframe(frame);
```

Protocole avec erreurs

Essai 3, correct

- Idem, mais *B* renvoie le numéro de la trame qu'il attendait.

```
A: while (1)
   frame = network.frame[i];
   physical.put(frame);
   correct = physical.get() // timeout();
   if ((correct == i) &&!timeout) i++;

B: while(1)
   frame = physical.get();
   if (frame.number == i)
    physical.put(i++);
    network.handleframe(frame);
```

Note : on peut se contenter d'envoyer $i \mod 2$ plutôt que i en général

Protocole avec fenêtre

- *A* envoie *N* frames avant d'attendre que *B* acquittex
- Voir démo

Deux protocoles

- Go-Back-N. Taille de fenêtre N, messages numérotés de 1 à $N + 1$.
- Possibilité d'envoyer un NAK "je n'ai pas reçu le message"
- Répétition sélective. Taille de fenêtre N_A, N_B, messages numérotés de 1 à $N_A + N_B$.

Bidirectionnel

- *A* envoie des données à *B* et *B* des données à *A*
- Le ACK est encapsulé dans les frames de données
- *piggybacking*

Contenu

HDLC

High-Level Data Link Control - ISO 13239

Flag	Adresse	Commande	Données	CRC	Flag
8	8	8/16	Variable	16	8

- Flag 01111110, un seul sépare deux trames.
- Trois types de trames : Information (i.e. Données) , Supervision (e.g. ACK), Autre

- Données : La commande contient le numéro de la trame envoyée, et celui de la trame attendue (ACK)
- Supervision : (ex.) Reject/Selective Reject : La commande contient un numéro de trame à réenvoyer

Utilisé par ex dans ISDN/RNIS (Numéris), ou X.25

PPP

Point to Point Protocol (RFC 1661)

- Encapsulée dans HDLC
- Le champ Adresse et le champ commande ne sont pas utilisés
- Le champ données contient le protocole, et les données associées

Utilisé partout

Contenu

La semaine prochaine

- On reste sur la couche liaison
- On s'intéresse aux réseaux locaux, et réseaux à diffusion