



Google

Recherche | Applications

Réseau



Laurent Noé –
Cristian Versari
(matériel pédagogique
Gilles Grimaud)

Objectifs du cours



```
suki4:~ laurent$ dig www.google.com
;; global options: +and
;; Got answer:
;; --HEADER-- opcodes: QUERY, status
;; flags: qr rd ra QUERY: 1, ANSWER:
;; QUESTION SECTION:
;www.google.com.
;; ANSWER SECTION:
www.google.com.      53      IN      A
www.google.com.      53      IN      AAAA
www.google.com.      53      IN      NS
www.google.com.      53      IN      MX
www.google.com.      54      IN      TXT
www.google.com.      51      IN      SRV
;; AUTHORITY SECTION:
google.com.          72293   IN      NS
google.com.          72293   IN      NS
google.com.          72293   IN      NS
google.com.          72293   IN      NS
;; ADDITIONAL SECTION:
ns1.google.com.      215688  IN      A
ns1.google.com.      215688  IN      AAAA
```

```
suki4:~ laurent$ traceroute google.com
traceroute to google.com (216.58.204.110)
 1  gw324-crl11f1.fr (134.206.24.1) 0.
 2  ciscoluniv-lille1.fr (193.49.225.3)
 3  193.51.250.1 (193.51.250.1) 0.870 m
```

```
suki4:~ laurent$ ping -c 3 www.google.com
PING www.google.com (64.233.184.184): 56
64 bytes from 64.233.184.184: icmp_seq=
64 bytes from 64.233.184.184: icmp_seq=
64 bytes from 64.233.184.184: icmp_seq=
```

```
File Edit View Go Capture Analyze Statistics Telephony I
Title:
No. Time Source Destination Protocol Length In
```

No.	Time	Source	Destination	Protocol	Length	In
7	0.04261000	192.168.1.1	192.168.1.10	ICMP	28	0.04261000
8	0.84251000	192.168.1.10	192.168.1.1	ICMP	28	0.84251000
9	0.851542000	192.168.1.10	192.168.1.1	ICMP	53	0.851542000
10	0.855634000	192.168.1.10	192.168.1.1	ICMP	135	0.855634000
11	0.853660000	192.168.1.10	192.168.1.1	ICMP	135	0.853660000
12	0.858350000	192.168.1.10	192.168.1.1	ICMP	112	0.858350000
13	0.858380000	192.168.1.10	192.168.1.1	ICMP	103	0.858380000
14	0.861681000	192.168.1.10	192.168.1.1	ICMP	342	0.861681000
15	0.862081000	192.168.1.10	192.168.1.1	ICMP	62	0.862081000
16	1.861150000	192.168.1.10	192.168.1.1	ICMP	116	1.861150000
17	1.771264000	192.168.1.10	192.168.1.1	ICMP	78	1.771264000
18	1.495610000	192.168.1.10	192.168.1.1	ICMP	342	1.495610000
19	2.221325000	192.168.1.10	192.168.1.1	ICMP	62	2.221325000
20	3.882433000	192.168.1.10	192.168.1.1	ICMP	116	3.882433000
21	3.695890000	192.168.1.10	192.168.1.1	ICMP	342	3.695890000
22	3.782486000	192.168.1.10	192.168.1.1	ICMP	362	3.782486000

```
run 1: 142 bytes in wire (2735 bits), 342 by
Ethernet II, Src: Apple ee:08:17 (28:1c:9d:0d:
Internet Protocol version 4, Src: 8.8.8.8, Dst:
User Datagram Protocol, Src Port: 68 (68), Dst
Bootstrap Protocol (Discover)
```

No.	Time	Source	Destination	Protocol	Length	In
6000	11.11.11.11.11.11.26.20	192.168.1.10	192.168.1.1	ICMP	37	11.11.11.11.11.11.26.20
6010	31.48.61.80.33.88.11.11	192.168.1.10	192.168.1.1	ICMP	29	31.48.61.80.33.88.11.11
6020	11.11.00.44.00.43.02.34	192.168.1.10	192.168.1.1	ICMP	01	11.11.00.44.00.43.02.34
6030	95.e2.00.00.00.00.00.00	192.168.1.10	192.168.1.1	ICMP	00	95.e2.00.00.00.00.00.00
6040	00.00.00.00.00.00.20.29	192.168.1.10	192.168.1.1	ICMP	17	00.00.00.00.00.00.20.29

```
... style="rgb(233, 234, 214)
-moz-background-clip: initial;
initial;" bgcolor="#e9ead6" val
</a><br>
</td>
<td bgcolor="#b0b190" width=
<td align="right" bgcolor="#d9d
<table border="0" cellpadding=
<tbody><tr>
<td bgcolor="#d9dac6" height=
<td rowspan="3" align="right" b
<target="_blank">
<td colspan="5" align="center"
<table width="100%">
<tbody><tr>
<td rowspan="2" class="h"
nowrap="nowrap" width="300"
<input name="exclude" value="index/man.shtml"
class="h"><u>?????</u></a>&nbsp;&nbsp;&nbsp;<input size="20" name="
width="1" type="text"></div>
```

10 Semaines :

Cours (1h30 × 10 séances)

Travaux Dirigés (1h30 × 10 séances)

Travaux Pratiques (2h × 10 séances)

dont 5 au pôle télécom (M5 salle B05)

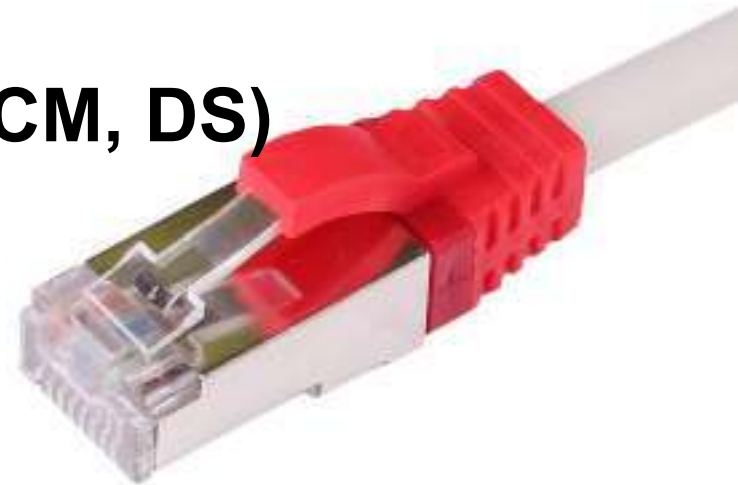


Evaluations :

- Devoir surveillé (3h)**
- + Contrôle intermédiaire QCM (1h30)**
- + Evaluation des TPs (compte rendu + 2h)**

$\frac{1}{4}$ TP + $\frac{3}{4}$ MAX($\frac{2}{3}$ DS + $\frac{1}{3}$ QCM, DS)

$\frac{1}{4}$ TP + $\frac{3}{4}$ Rattrapage



Définir et classer les réseaux numériques

Les supports matériels de la communication

Les protocoles de l'Internet et les réseaux

Les flux de données applicatifs

Programmes d'échanges de données

Le modèle OSI



Vue d'ensemble

Introduction aux Réseaux

Un **réseau numérique** permet l'échange de données entre des machines distantes. Ces données sont, si nécessaire, **relayées** de liaison en liaison par les machines intermédiaires.

Échange de donnée en mode connecté



Échange de donnée en mode non-connecté



Modalité d'échange de donnée

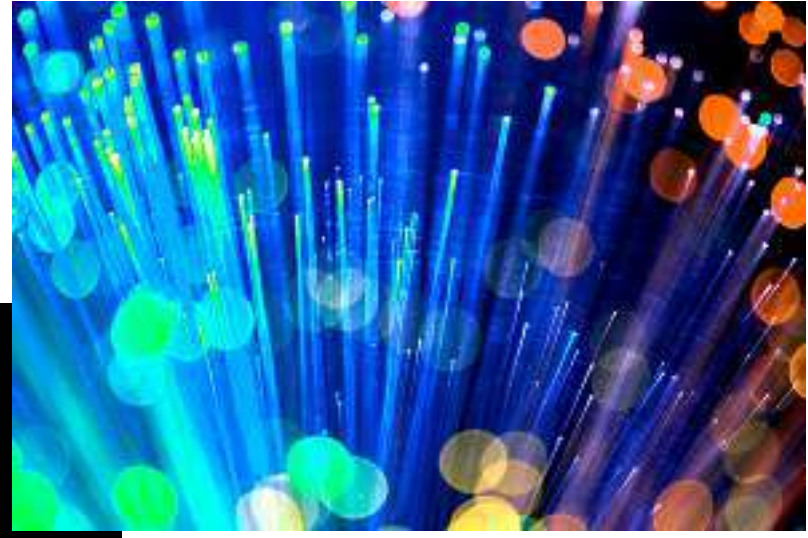
LAN, MAN et WAN



LAN, **MAN** et WAN



LAN, MAN et **WAN**



Topologie des liaisons

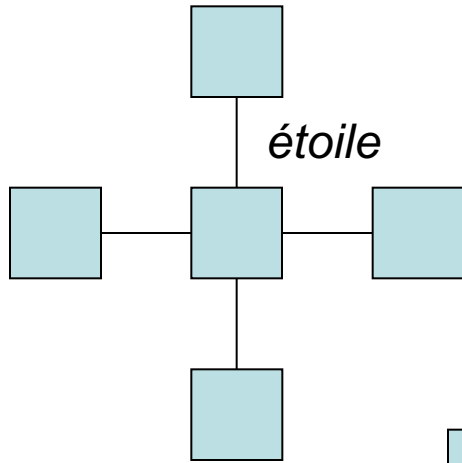


Liaisons « point à point »

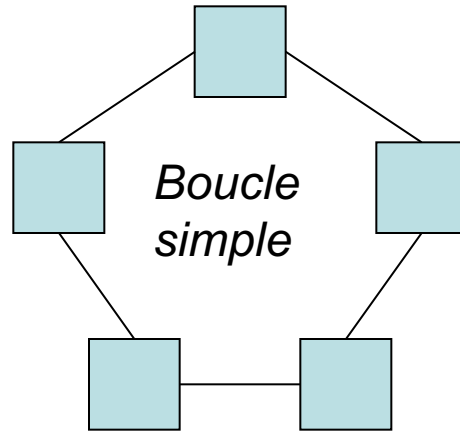
Liaisons « en diffusion »



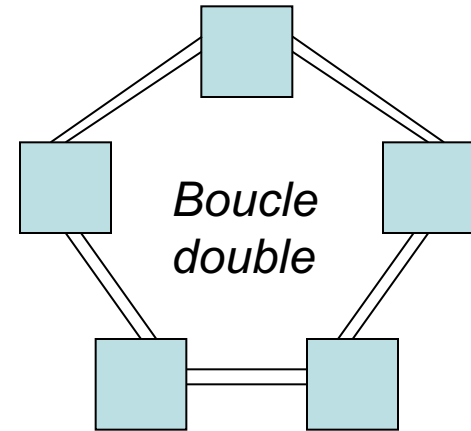
Topologie des réseaux



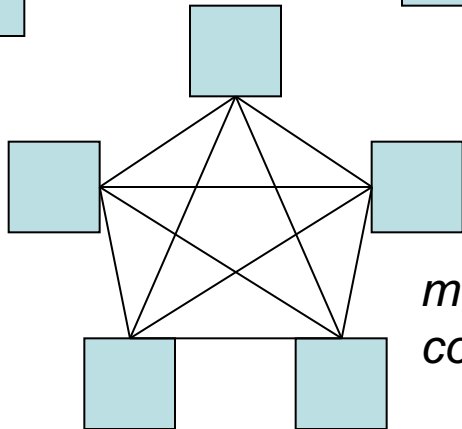
étoile



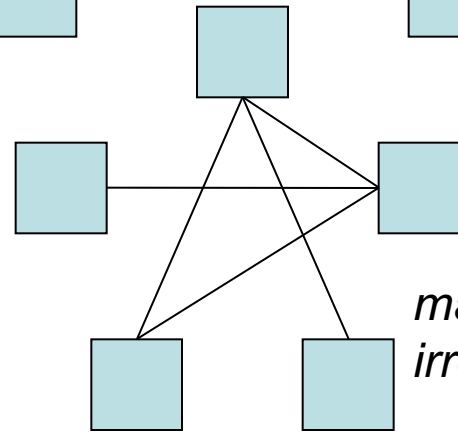
Boucle simple



Boucle double

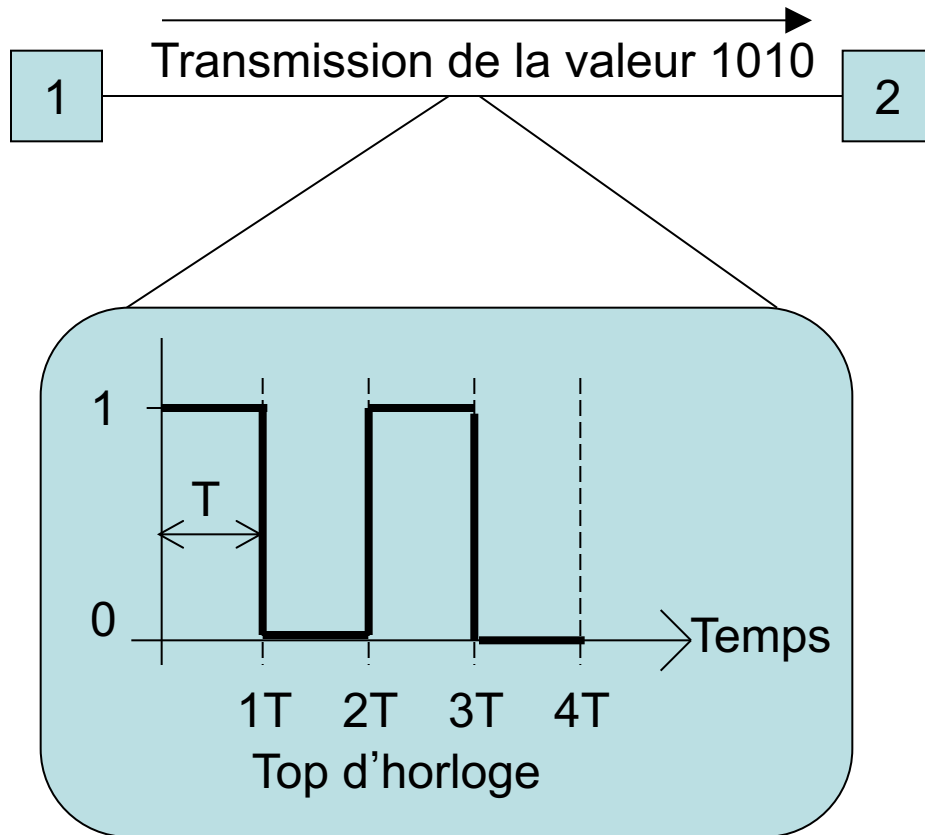


maillage complet



maillage irrégulier

Signaux et transmission de données



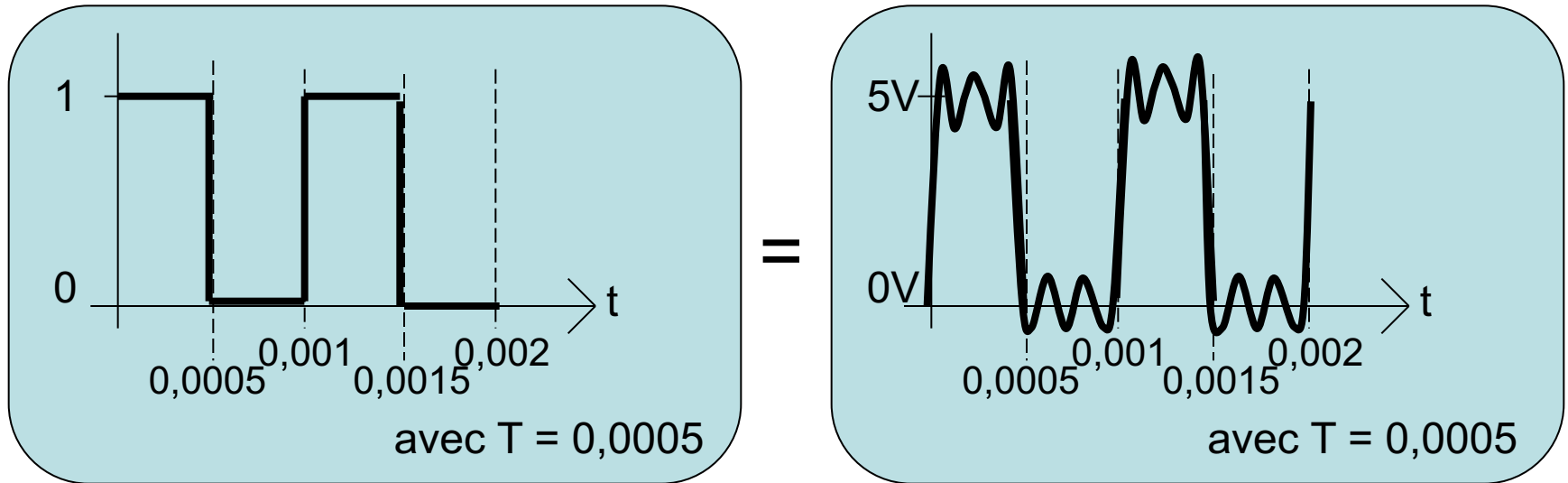
Baud = nombre de changements
par seconde
 $= 1 / T$
(T temps entre deux tops d'horloge)

Sur une liaison série 1 seul bits
transmit à chaque Top d'horloge.
1 Bauds \sim 1 bit/s.

Sur une liaison parallèle à n bits
1 Bauds \sim n bits/s.

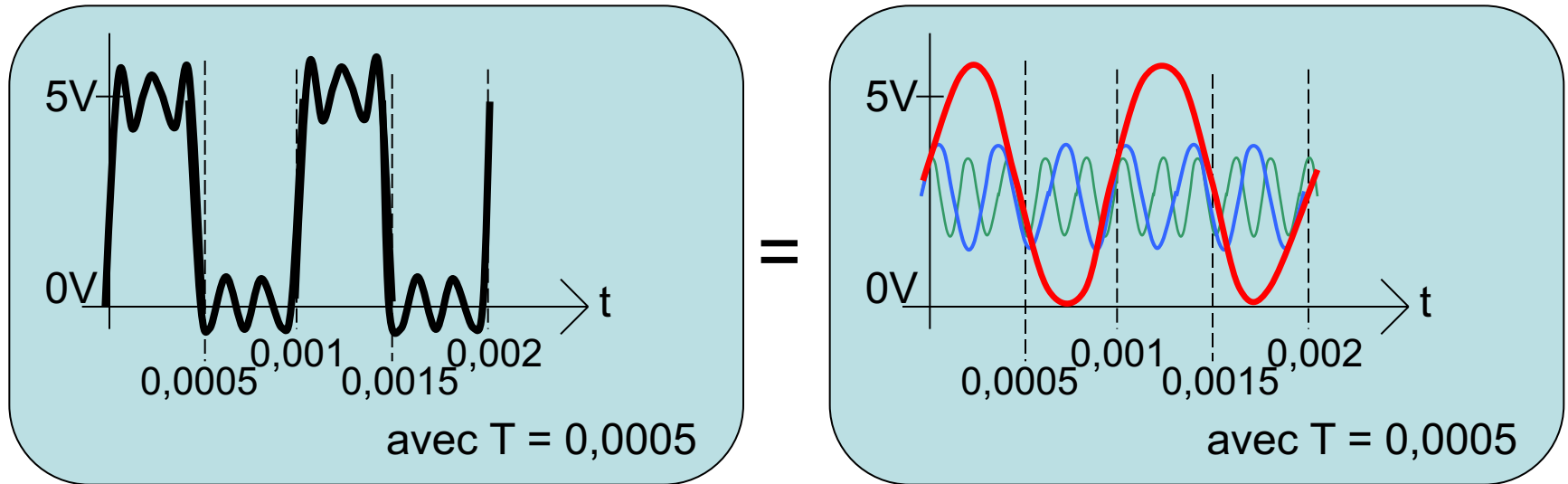
Transformer la valeur en un signal carré

Signaux et transmission de données



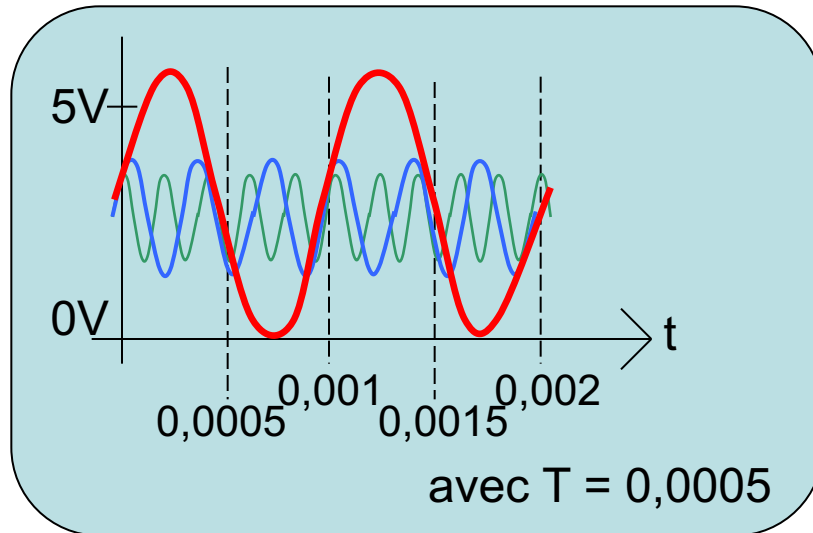
Transporter un signal carré sur un support analogique
(e.g. téléphone)

Signaux et transmission de données



un signal analogique est une somme (limitée) d'harmoniques.

Signaux et transmission de données



$$= g(t) = \frac{1}{2} + \frac{2}{\pi} \sin(2000 \pi t) + \frac{2}{3\pi} \sin(6000 \pi t) + \frac{2}{5\pi} \sin(10000 \pi t) + \dots$$

Chaque harmonique correspond à un signal sinusoïdal donné.

Signaux et transmission de données

$$\begin{aligned} &1/2 + \\ &2/\pi \sin(2000 \pi t) + \\ &2/3\pi \sin(6000 \pi t) + \\ &2/5\pi \sin(10000 \pi t) + \\ &\dots \end{aligned}$$

$$= g(t) = \frac{c}{2} + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t)$$

Avec $f = 1/T$ (fréquence fondamentale)

$$c = 2/T \int_0^T g(t) dt$$

$$a_n = 2/T \int_0^T g(t) \sin(2\pi n f t) dt$$

$$b_n = 2/T \int_0^T g(t) \cos(2\pi n f t) dt$$

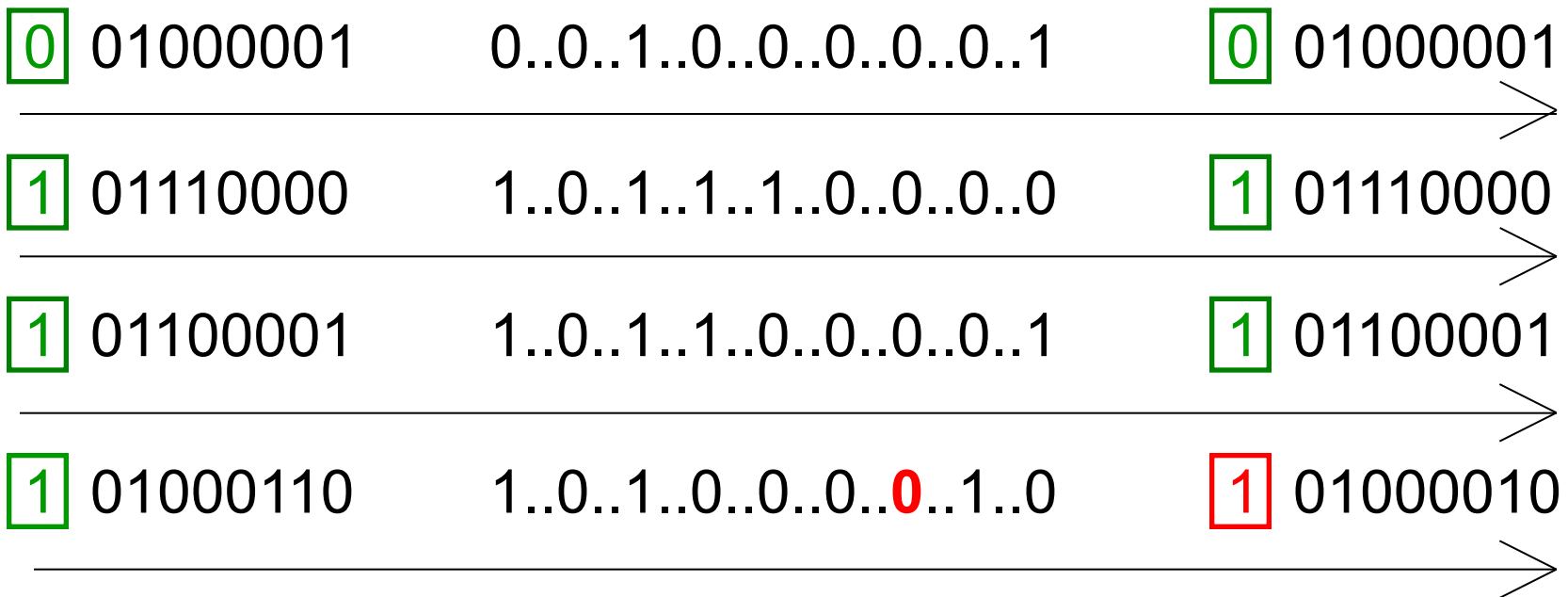
Dont la somme infinie constitue une **série de Fourier**.

Liaison des données

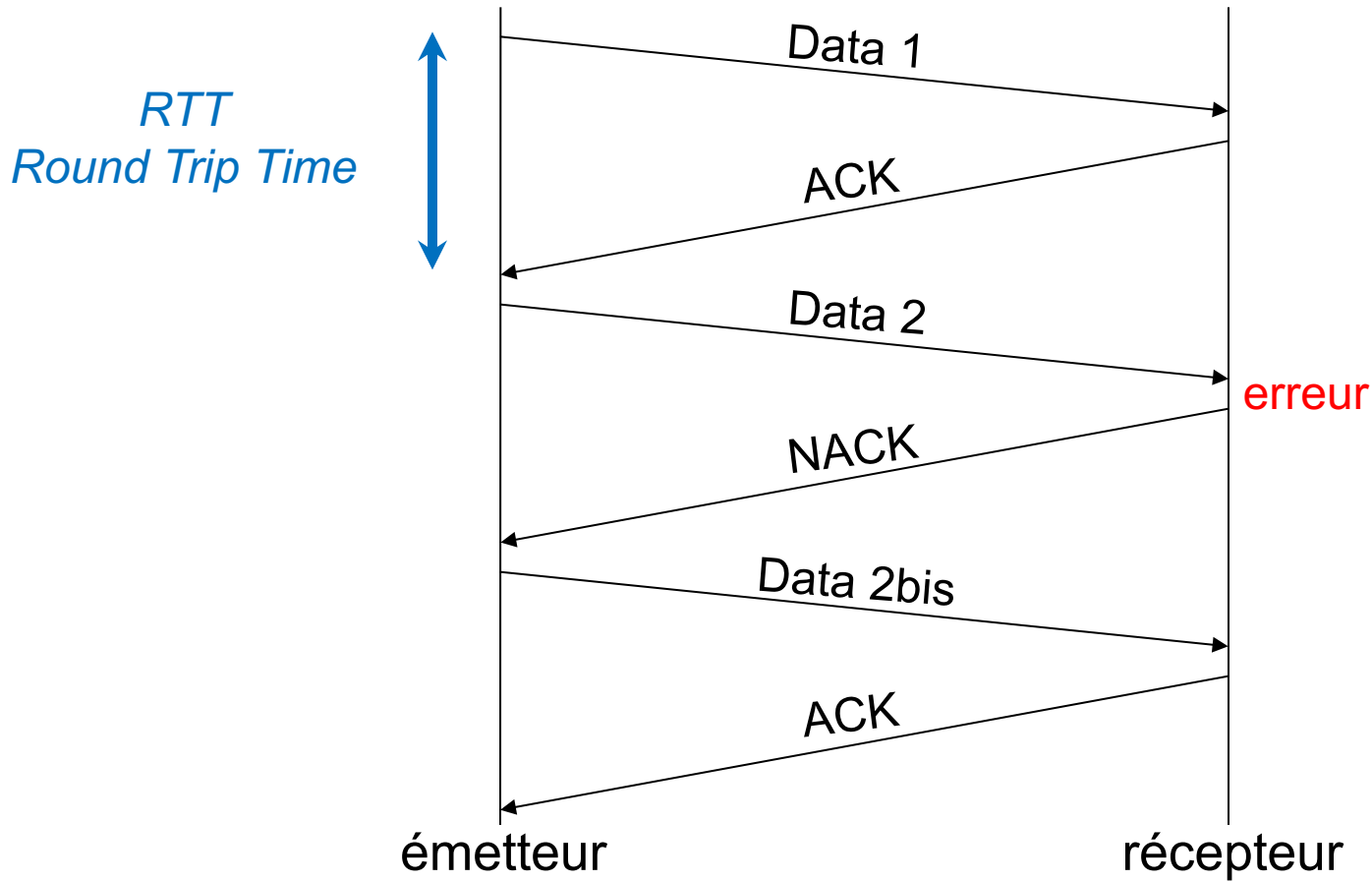
La transmission matérielle d'information étant sujette à des perturbations extérieures, elle induit de potentielles erreurs de transmissions. La liaison de données consiste en des procédures logicielles de détection et correction éventuelle de l'information reçue.

Bit de parité : un exemple simple

Convenir d'un codage détecteur d'erreur pour les données transmises : **Le bit de Parité**



Protocole envoyer et attendre

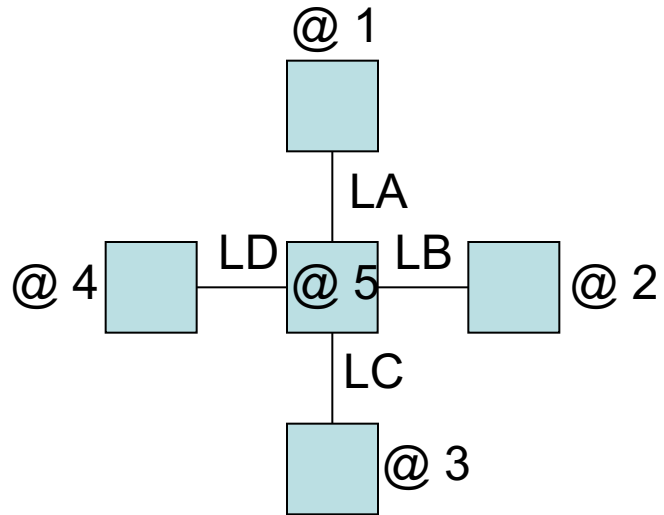


Notion de routage



Un réseau est constitué de différentes liaisons entre ordinateurs. La gestion d'un réseau nécessite l'existence de mécanismes d'**adressage** des différentes machines, de **routage** et de **contrôle de flux** des paquets de données transportés sur chaque liaison.

Adressage



Routage

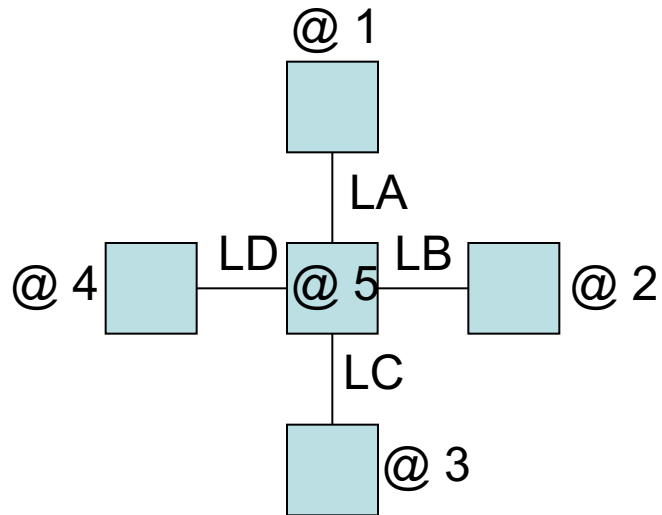


Table @3	
@ des	Liaison
@ 1-5	LC

Table @5	
@ des	Liaison
@ 1	LA
@ 2	LB
@ 3	LC
@ 4	LD

Routage

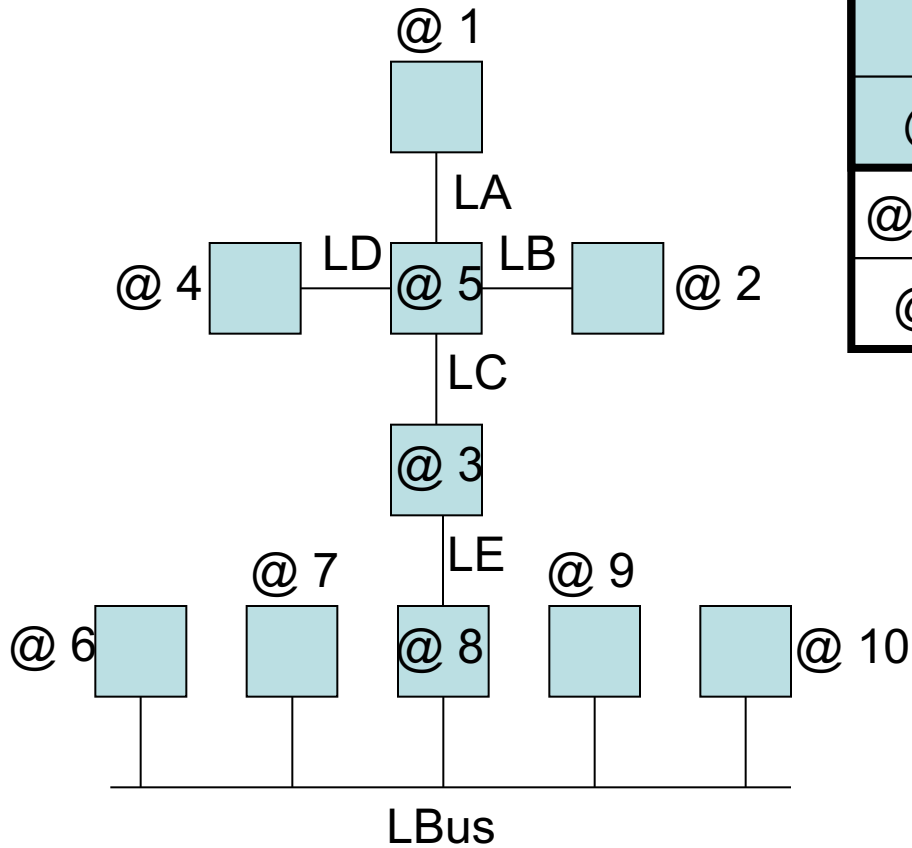


Table @3	
@ des	Liaison
@ 1-2;4-5	LC
@ 6-10	LE

Table @5	
@ des	Liaison
@ 1	LA
@ 2	LB
@ 3;6-10	LC
@ 4	LD

Table @8	
@ des	Liaison
@ 1-5	LE
@ 6-7;9-10	LBus

Routage

Gestion via un routage centralisé :

- **Fixe** : pas de mise à jour. Tables fixées une fois pour toute en *fact* de la topologie du réseau.
- **Synchrone** : Tables mises à jour au même moment par un centre de contrôle. (à partir d'informations reçues dynamiquement).
- **Asynchrone** : tables mises à jour indépendamment les unes des autres dans certaines parties du réseau (avec émission d'un compte-rendu de son état au centre de contrôle).

Routage décentralisé

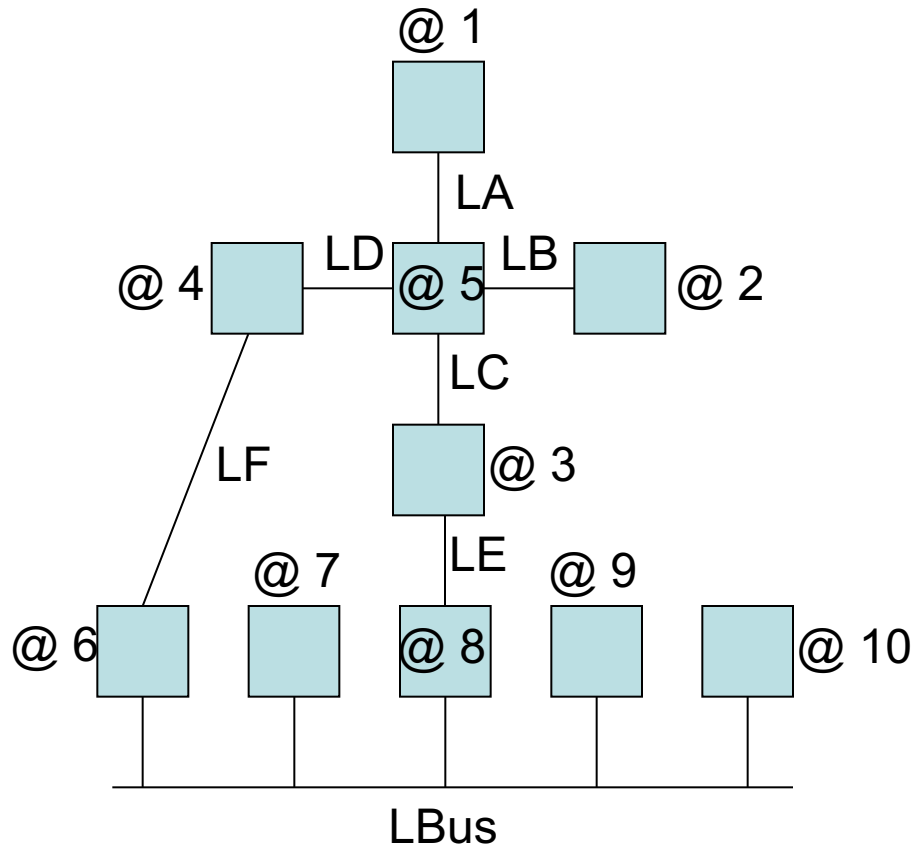
inondation, *hot potatoes*, routage adaptatif.

Table @3	
@ des	Liaison
@ 1-2;4-5	LC
@ 6-10	LE

Table @5	
@ des	Liaison
@ 1	LA
@ 2	LB
@ 3;6-10	LC
@ 4	LD

Table @8	
@ des	Liaison
@ 1-4	LE
@ 6-7;9-10	LBus

Contrôle de flux



Le contrôle de flux a pour **objectif** :

- minimiser le temps de transfert des paquets ;
- éviter la congestion du réseau ;

Techniques :

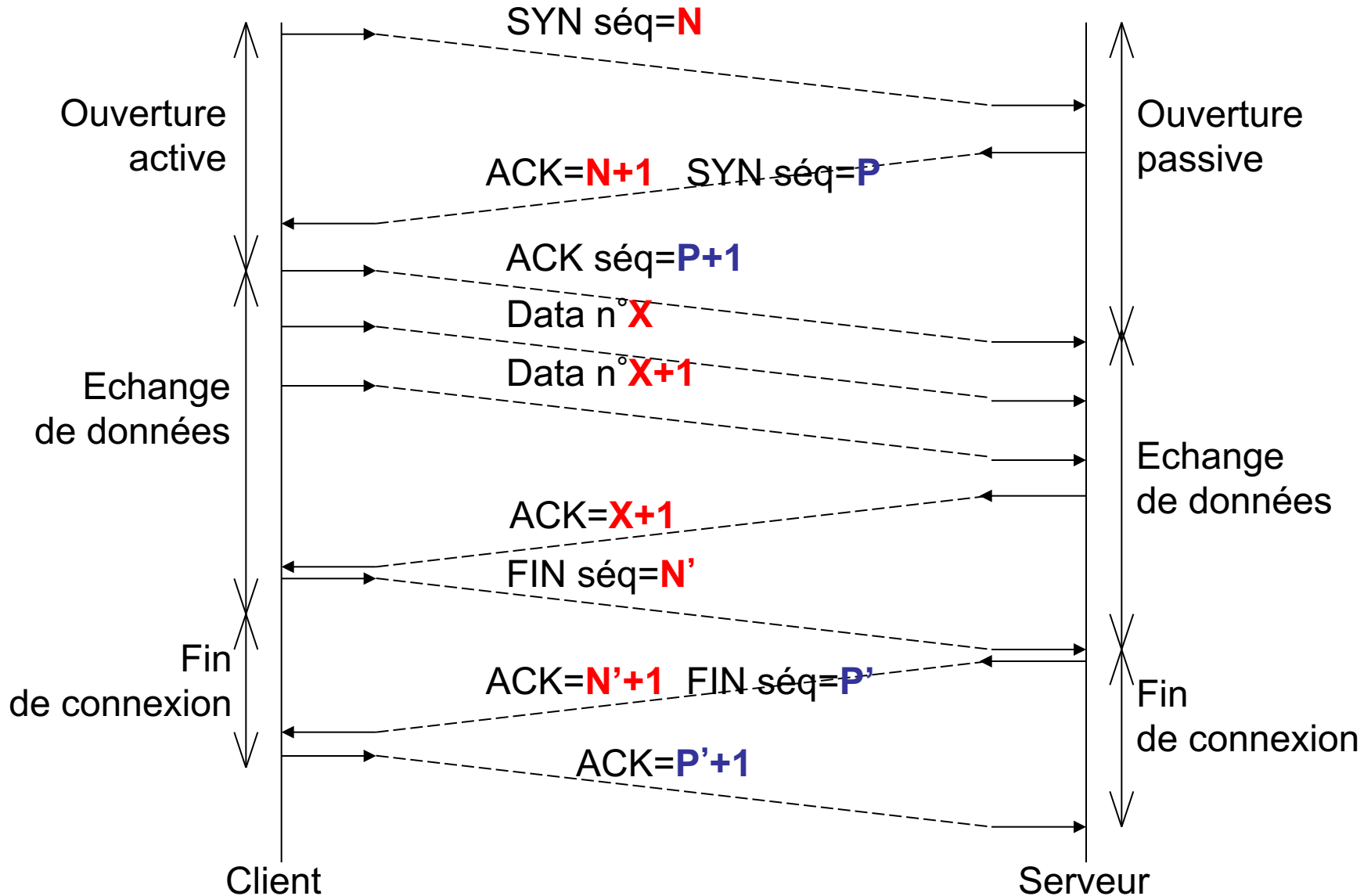
Contrôle par crédits & crédits dédiés. Contrôle par fenêtre.

Transport



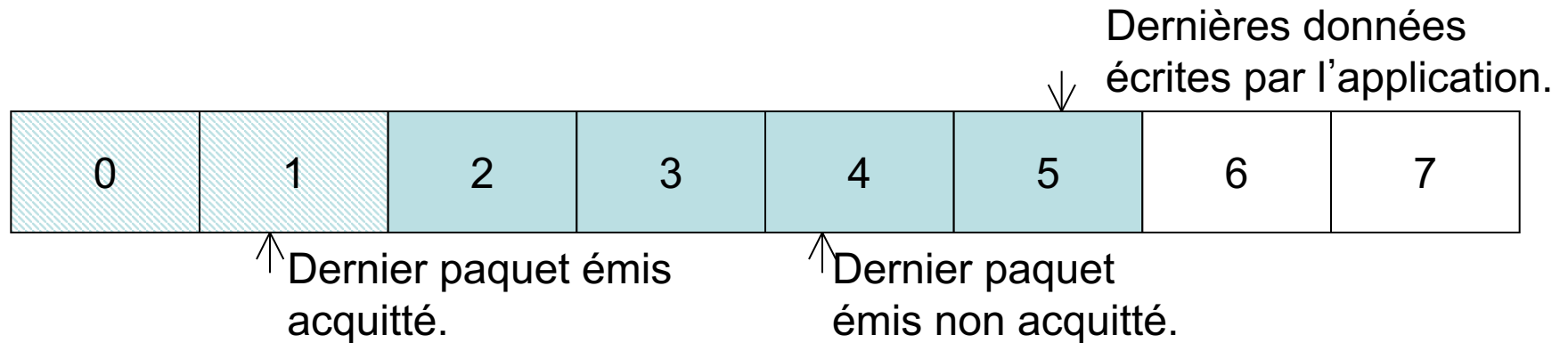
Les services de transport assurent aux applications la capacité de transférer des séquences de données de tailles indéterminées. A cette fin elle **segmente et désegmente** (fragmente et défragmente) ces données en paquets autonomes qui sont émis sur (reçus depuis) le réseau. Elle assure **la fiabilité des données** en gérant la perte de paquets ou leur corruption.

Mode connecté (e.g. TCP)

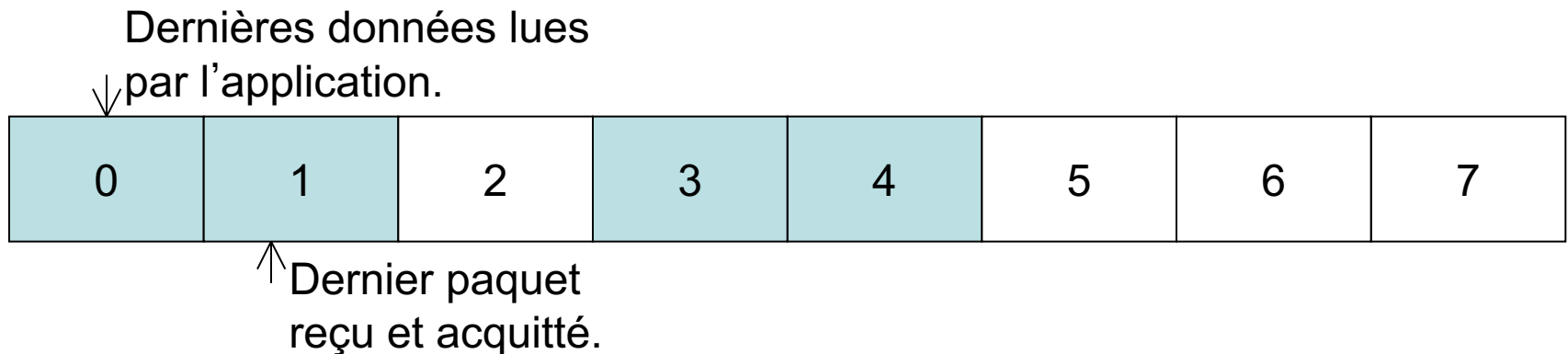


Connexion TCP

Gestion d'une fenêtre d'émission :



Gestion d'une fenêtre de réception :



Programmes d'échanges de données

Pour une **application**, le réseau apparaît comme un service logiciel (un ensemble de fonctions) qui permet **d'initier l'émission** à destination, ou **d'attendre la réception** de données en provenance de n'importe quelle autre application.

émission d'un paquet UDP

– avec Java –

```
DatagramPacket p;
```

```
DatagramSocket s;
```

```
InetAddress dst = InetAddress.getByName("brigant");  
int port = 1024 ;
```

```
byte array[] = new byte[2];  
array[0] = 1;  
array[1] = 2;
```

Contenu

Destinataire : @IP + Port

```
p = new DatagramPacket p(array, 2, dst, port);
```

```
s = new DatagramSocket();
```

Emetteur : @IP + Port

```
s.send(p);
```

```
s.close();
```

Réception d'un paquet UDP

– avec Java –

```
import java.net.*;
import java.io.*;
DatagramSocket s;
DatagramPacket p;
s = new DatagramSocket(1024);
p = new DatagramPacket(new byte[512], 512);
s.receive(p);
system.out.println("paquet reçu de :"+ p.getAddress()+
                    "port "+ p.getPort()+
                    "taille" + p.getLength());
byte array[] = p.getData();
For(i=0;i<p.getLength();i++)
    System.out.println("array["+i+"] = "+array[i]);
s.close();
```

Recepteur : ((@IP)) + Port

Emetteur : @IP + Port

**Affichage
Contenu**

Le modèle OSI

