

1 Étude physique d'une ligne de transmission

Les signaux électriques utilisés pour transmettre les données subissent des dégradations physiques "normales" telles que :

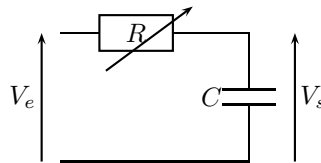
- pertes ohmiques ;
- déphasage ;
- "filtrage" importun parce que les lignes sont toujours assimilables à des circuits RC ;
- induction mutuelle entre conducteurs ;
- lignes non accordées (rebonds).

En outre, les signaux subissent différentes agressions ("parasites") dues au milieu :

- parasites industriels induits par les démarrages de moteurs, allumages d'éclairage fluorescent ;
- parasites véhiculés par les alimentations ;
- intermodulation (crosstalk) par les signaux de transmission de données voisins (ex. câble multiple, GSM).

Exercice 1.1 : Étude physique de la ligne

Question 1 : Le premier objectif est de mettre en évidence certains des facteurs perturbant la transmission sur une ligne. Pour cela, réalisez une maquette de ligne dégradée :



$R = 1M\Omega$ (potentiomètre)

$C = 100pF$

Question 2 : Mettez en entrée de ce circuit un générateur de signaux. Observez le signal en entrée et en sortie et essayez, pour plusieurs formes de signaux, de mettre en évidence les deux paramètres importants : l'atténuation et le déphasage.

Question 3 : Faites varier la résistance pour obtenir une bande passante permettant d'avoir un signal à peu près carré à $10kHz$. Mesurez la résistance pour en déduire la bande passante.

Exercice 1.2 : Étude de la réflexion d'un signal

Question 1 : Proposez un montage pour envoyer un signal carré à une certaine fréquence dans le câble de grande longueur et pour visualiser en même temps le signal à l'entrée et à la sortie du câble, sachant qu'en sortie vous devez pouvoir connecter une charge quelconque.

Question 2 : Relevez les signaux en entrée et en sortie lorsque vous connectez en sortie :

1. un circuit ouvert
2. un court-circuit
3. une charge de 50Ω que l'on appelle bouchon.

Question 3 : Que devrait-on observer en entrée du câble ? Avec quelle charge obtient-on le bon résultat ? Que se passe-t-il dans les 3 cas ?

Question 4 : Avec la charge de 50Ω , mesurez l'atténuation.

Question 5 : Le câble comporte une étiquette sur laquelle est inscrite sa longueur. Quelle est la vitesse de propagation ? Comparez vos résultats avec ceux d'un autre binôme ayant un câble de longueur différente.

Question 6 : Faites varier la fréquence pour avoir une idée de la bande passante du câble.

Exercice 1.3 : Émission d'un caractère à partir d'un PC

La liaison série asynchrone et le composant électronique UART qui lui est associée sont les descendants directs d'un système de transmission de données vieux de plus d'un siècle, le célèbre télégraphe système Baudot.

La trame Baudot a la structure temporelle très générale suivante :

START	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parité	STOP	STOP
-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	------	------

où les bits 5, 6, 7, parité et le premier bit stop sont optionnels.

Cette trame ne peut s'interpréter que sur une ligne de transmission à deux états :

- état de *repos* en l'absence de transmission (en anglais *mark*) ;
- état *actif* (en anglais *space*).

Habituellement, on choisit un état de *repos* tel que l'émetteur doive injecter de l'énergie dans le médium pour entretenir cet état ; de telle sorte qu'une absence prolongée de cet état peut s'interpréter à coup sûr :

- soit comme une panne d'émetteur ;
- soit comme une coupure physique du médium.

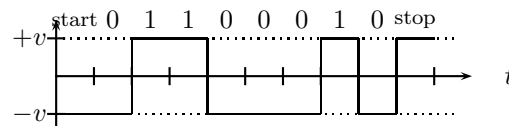
START est un état *space* durant un temps-bit, qui délimite le début de trame. A partir de la transmission *mark* \rightarrow *space* on cale une horloge d'échantillonnage au milieu des temps-bits 1. Il se traduit par une tension $-v$.

STOP est un état de *repos* imposé pendant (au moins) un temps-bit, avant toute nouvelle trame. Il correspond à l'envoi d'une tension $+v$.

Les variantes de format sont les suivantes :

- données, de 5 bits (ex. télex) à 7 ou 8 bits ;
- sans parité, avec parité paire ou impaire ;
- durée du STOP égale 1 ou 2 temps bit.

Par conséquent, pour envoyer 0100 0110, les tensions sont :



Par définition, un composant ayant la fonction UART s'intercale entre :

- un bus interne d'ordinateur ;
- une jonction physique de raccordement à un médium de transmission en série.

Les signaux sur la jonction sont en général d'une nature différente des signaux logiques usuels dans l'ordinateur, d'où une couche de circuits *translateurs* (drivers et récepteurs).

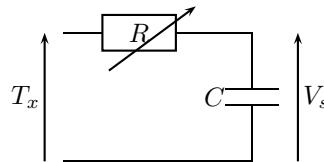
Nous vous fournissons un premier programme dont voici l'algorithme :

```
Algorithme Prog1_1
  variable e : caractère;

Début
  initialiser(COM_1); /* COM_1 = sortie série numéro 1 */
  saisir_au_clavier(e);
  tantque (aucune touche du clavier n'est actionnée) faire
    émettre_via(COM_1,e);
  fin tantque;
Fin
```

On peut, en jouant sur les paramètres passés à la fonction d'initialisation de l'uart, modifier la rapidité de modulation, le nombre de bits transmis, le nombre de bits stop, exercer ou non un contrôle de parité, et évidemment, choisir le caractère à émettre. Une fois le programme lancé, le même caractère est émis continuellement : on dispose de tout son temps pour étudier le signal transmis en utilisant un demi-câble NULL MODEM simplifié associé à un oscilloscope.

Branchez le câble sur votre maquette de la manière suivante :



Question 1 : Lancez plusieurs fois le programme en modifiant les paramètres. À chaque fois :

- Observez à l'oscilloscope et relevez le chronogramme de la trame émise en sortie de l'UART (donc à l'entrée de la ligne). Vérifiez la rapidité de modulation.
- Observez le signal transmis en sortie de ligne. Faites varier la valeur de la résistance. Déterminez à quel moment le signal n'est plus utilisable, et déduisez-en la bande passante.

Exercice 1.4 : Émission/réception d'un caractère à partir d'un PC

C'est le même PC qui émet et qui reçoit, en utilisant le second programme fourni, dont l'algorithme est :

```
Algorithme Prog1_2
  variables e,r : caractère;

Début
  initialiser(COM_1);
  saisir_au_clavier(e);
  tantque (aucune touche du clavier n'est actionnée) faire
    émettre_via(COM_1,e);
    r = lire_via (COM_1);
    afficherEcran(r) ;
  fin tantque;
Fin
```

Question 1 : Court-circuitez T_x sur R_x et faites fonctionner le programme.

Question 2 : Envoyez le signal d'émission T_x sur l'entrée de la ligne et branchez la sortie sur R_x .
Que constatez-vous ?

Question 3 : Sachant que la résistance d'entrée du circuit Rx est de quelques Ω , que faut il faire ?

2 Analyse de trames

Exercice 2.1 : Réalisation d'un câble 10baseT

Il s'agit de prendre un morceau de câble et de sertir une prise RJ45 à chaque bout. Pour cela :

1. Dénuder 3cm de gaine isolante.
2. Positionner les brins les uns à côté des autres en respectant l'ordre suivant : marron, blanc/marron, vert, blanc/vert, bleu, blanc/bleu, orange, blanc/orange.
3. Couper les brins à environ 12mm de la gaine. Ne pas les dénuder.
4. Insérer les brins dans le connecteur. Ils doivent s'enfoncer dans le bon ordre, et jusqu'au fond : on doit voir apparaître le contact de l'autre côté.
5. Faire vérifier votre connection à l'enseignant. Celui-ci vous montrera comment utiliser la pince à sertir.

Exercice 2.2 : Analyse de trames

L'exercice a un but double : d'une part analyser des trames de protocoles vus en cours, et d'autre part apprendre à manipuler un outil d'analyse de trames pouvant servir par la suite.

L'outil d'analyse de trames à utiliser est **ethereal**. Il comporte une interface graphique conviviale présentant les différents éléments d'une trame.

Question 1 : Après avoir lancé **ethereal**, utilisez le menu **Capture->start** pour commencer à examiner les trames qui circulent. Vous capturerez des trames pendant une vingtaine de secondes. Les résultats sont affichés dans trois fenêtres. Les adresses source et destination se trouvant dans la première fenêtre sont des adresses IP (voir module 4 de réseau pour les détails sur IP). Expliquez quelles informations sont fournies dans chacune des fenêtres.

Question 2 : Quels sont, parmi ceux qui vous sont présentés, les types de trames vus en cours ?

Question 3 : Analysez ces trames et vérifiez que la structure correspond bien à celle donnée en cours. Vous devrez en particulier analyser la longueur des trames capturées, délimiter les différents champs. Donnez, dans votre compte-rendu, un exemple pour chaque type de trame.

Question 4 : La BPDU est une trame permettant aux ponts de se configurer. Quels sont les éléments constitutifs d'une BPDU ? Donnez un schéma présentant la structuration d'une BPDU.

Question 5 : Quelles sont les adresses MAC de la (carte réseau de la) machine que vous utilisez ? du pont le plus proche ?

Question 6 : Faites afficher les différents types de trames qui vous intéressent avec des couleurs différentes de manière à pouvoir les distinguer facilement.

Question 7 : Quelle est la répartition statistique des trames que vous avez capturées ?

Question 8 : Quelle est la distribution dans le temps des trames que vous avez capturées ? Celle-ci sera affichée sous forme graphique.

3 Commutation Ethernet

Les étudiants installés sur une même rangée travailleront en groupe. Chaque rangée dispose d'au moins 4 PCs, d'un *hub* et d'un *switch* CISCO 2950.

Exercice 3.1 : Configuration d'un *switch*

Question 1 : Installer l'image GTR numéro 4 sur 4 machines de la rangée.

Question 2 : Connecter un des PC au *switch* via un câble série. Attention, les machines disposent de 2 ports série, utiliser le B/2.

Question 3 : Se connecter sur la machine en tant que **etudiant**.

Question 4 : Ouvrir un terminal dans lequel vous allez travailler et utilisez la commande `su -` pour passer administrateur de la machine.

Question 5 : Lancer l'émulateur de terminal `minicom` avec l'option `-s`. Configurer le port série en 9.600 bauds (9600 8N1).

Question 6 : Éteindre le switch en le débranchant puis le rebrancher tout en appuyant sur le bouton *Mode*, qui doit être maintenu appuyé jusqu'à ce qu'un message s'affiche dans la console.

Question 7 : Initialiser le système du switch :

```
switch# flash_init
switch# load_helper
```

Supprimer l'ancienne configuration :

```
switch# del flash:config.text
```

Vous pouvez voir le contenu du système de fichier de la mémoire flash du switch en utilisant :

```
switch# dir flash:
```

Question 8 : Démarrer le système :

```
switch# boot
```

Le fichier de configuration précédent ayant été effacé, le système commence par proposer de lancer la dialogue de configuration initial. Répondre oui (Y).

Question 9 : Il faut ensuite répondre aux diverses questions concernant la configuration du switch. Une fois la configuration terminée, vous enlèverez la connection série, mais ne débranchez pas le switch.

Les caractéristiques importantes à indiquer lors de la configuration sont les suivantes :

- tous les mots de passe seront `switchR`, où R est le numéro de la rangée sur laquelle vous travaillez.
- l'adresse IP du switch sera `10.0.R.100`, où R est le numéro de la rangée.
- l'administration devra pouvoir se faire à partir du VLAN par défaut (`vlan1`).

Pour les autres questions, utilisez la valeur par défaut qui vous est proposée.

Exercice 3.2 : Configuration du réseau

Question 1 : Réaliser la configuration matérielle décrite dans la figure 1. Une machine est reliée à un équipement d'interconnexion par un *câble droit*, alors que deux équipements d'interconnexion sont reliés par un *câble croisé*. De plus, les PCs disposent de 2 cartes ethernet dont sortent des câbles de couleurs différentes : le *câble jaune* est connecté au réseau de l'IUT, il ne faut pas le déconnecter ; le *câble rouge* est connecté au réseau de la salle (armoire de répartition à côté du tableau), qui est prévu pour être complètement reconfiguré. Vous utiliserez donc la carte sur laquelle est branchée la câble rouge.

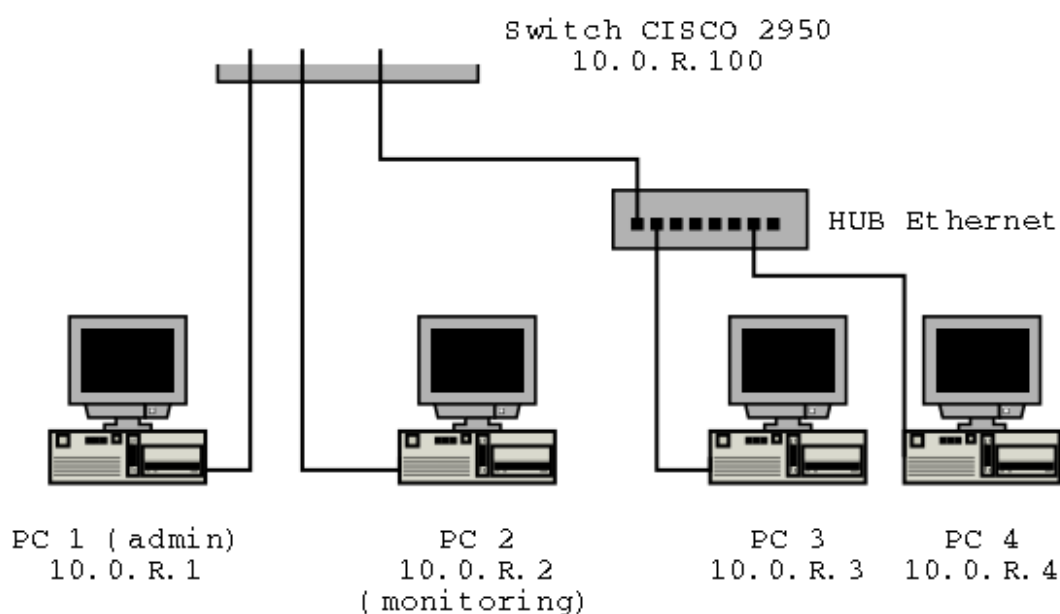


FIG. 1 – Câblage d'une rangée

Question 2 : Configurer les interfaces IP sur chaque PC :

```
ifconfig eth0 10.0.R.M
```

où R est le numéro de la rangée et M est le numéro de la machine (à partir de la fenêtre).

Question 3 : Activer l'interface IP sur chaque PC :

```
ifup eth0
```

Question 4 : Vérifier la configuration de chaque machine :

```
ifconfig
```

Question 5 : Utiliser la commande `ping` pour vérifier que les machines communiquent bien et que le switch répond.

Question 6 : Vérifier avec la commande `mii-tool` ou la commande `ethtool` l'état de la carte réseau utilisée (`eth0`).

Exercice 3.3 : Analyse du trafic

Le but de cet exercice est de mesurer les débits entre machines.

Question 1 : Lire le manuel en ligne de la commande `ttcp`.

Question 2 : Utiliser `ttcp` pour mesurer les débits TCP et UDP entre :

1. les deux machines connectées au switch ;
2. les deux machines connectées au hub ;
3. une machine connectée au switch et une machine connectée au hub.

Exercice 3.4 : Algorithme de l'arbre couvrant

Le but de cet exercice est de voir le fonctionnement de l'arbre couvrant.

Question 1 : Configurer 3 switches.

Question 2 : Connecter une machine sur chaque switch et configurer leur carte réseau.

Question 3 : Utiliser `etherreal` pour obtenir l'adresse MAC du switch connecté à chaque machine.

Question 4 : Lancer une capture de trames sur deux machines. Connecter (avec un câble croisé) les deux switches auxquels sont reliés ces machines. Comment les BPDUs sont-elles modifiées ? Pourquoi ?

Question 5 : Recommencer l'opération en connectant le troisième switch soit au premier, soit au second. Comment garantir que le pont racine se trouve à une des extrémités de la chaîne de switches ?

Question 6 : Relier les switches aux deux extrémités. Que remarque-t-on dans les BPDUs ? Pourquoi ?