Réseau II : la couche physique

Juliusz Chroboczek

24 septembre 2021

La couche physique est la plus basse de nos cinq couches (numérotées de 1 à 7). Elle utilise les fonctionnalités fournies par la nature (le courant électrique, la lumière, etc.) et fournit à la couche lien un service très primitif : elle permet de transmettre des *symboles*, dont la nature précise dépend de la nature du lien physique, localement au lien et sans aucune détection ou correction d'erreurs.

1 Quelques protocoles de couche physique

Comme la couche physique se trouve très bas dans la pile, les différents protocoles de couche physique ont très peu de choses en commun (voyez « structure en sablier » dans le cours précédent). Dans cette partie, je donne quelques exemples de couches physiques, dont plusieurs qui serviront dans le cours sur la couche lien.

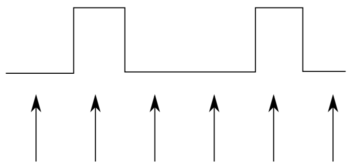
1.1 Ligne série

Une ligne série est un canal de communication qui peut être dans deux états, conventionnellement appelés *mark* et *space*. Les détais d'implémentation ne nous intéressent pas ; par exemple, un symbole peut être représenté par du courant et l'autre par l'absence du courant (c'est la « boucle de courant », qui est obsolète), ou alors les deux symboles peuvent être représentés par deux tensions distinctes, indépendantes de la charge (ce qui consomme moins qu'une boucle de courant et est la technique habituelle depuis la fin des années 1950).

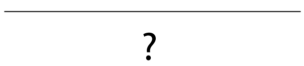
À tout moment, la ligne série est dans un des deux états *mark* ou *space* ; le rôle de la couche physique est de produire une suite de symboles *mark* et *space*, ce qui va demander d'échantillonner le signal à des moments bien définis : c'est le rôle du cadençage *(clocking)*. Par contre, la couche physique n'a pas pour rôle d'interpréter ces symboles comme des bits : c'est le rôle de la couche lien.

1.2 Cadençage

Pour pouvoir décoder la suite de symboles, il faut que le récepteur sache quand échantillonner le signal :

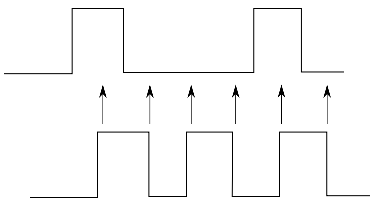


Déterminer le bon moment devient difficile si le signal a peu de transitions :



1.2.1 Horloge explicite

La solution la plus simple est d'ajouter une deuxième ligne, la ligne d'horloge, dont les transitions indiquent quand la ligne doit être échantillonnée :



Un avantage de l'horloge explicite est que l'émetteur peut suspendre la transmission à tout moment, ce qui permet de gérer le cas où l'émetteur est oisif (il n'a rien à émettre). Son désavantage, c'est qu'il est nécessaire d'ajouter une deuxième ligne, ce qui augmente les coûts.

La suite de symboles produite par la couche physique sera interprétée comme une suite de bits par la couche lien.

1.2.2 Communication synchrone

Pour éviter une horloge explicite et les coûts associés, une ligne série *synchrone* émet les symboles à intervalles réguliers. Le récepteur connaît le débit en symboles par secondes, et resynchronise son horloge avec celle de l'émetteur à chaque transition. Par conséquent :

- il faut s'assurer qu'il y a des transitions suffisamment souvent ;
- l'émetteur doit émettre quelque chose même lorsque la ligne est oisive.

Ces deux propriétés sont le problème de la couche lien, dont c'est la responsabilité de s'assurer qu'il y a un traffic constant qui cause suffisamment de transitions. La couche lien de l'émetteur devra insérer dans le traffic un motif qui indique que la ligne est oisive *(idle pattern)*, et elle devra insérer des bits supplémentaires *(bit stuffing)* pour garantir la présence de transmissions ; la couche lien de l'émetteur devra supprimer les motifs oisifs ainsi que les bits insérés.

1.2.3 Code manchester

Une autre approche consiste à utiliser un codage « *self-clocking* », à horloge implicite, qui contient suffisamment de transitions pour permettre au récepteur de se resynchroniser. Le plus simple est sans doute le *codage manchester*.

Le codage manchester utilise deux états pour chaque symbole : il représente un symbole par une suite *mark-space*, et l'autre par une suite *space-mark*. Comme avec une horloge implicite, le codage manchester permet de distinguer une suite de symboles de son absence : pour indiquer qu'il est oisif (qu'il n'a rien à émettre), l'émetteur cesse simplement d'émettre des symboles, ce que le récepteur est capable de détecter par l'absence de transitions.

Le codage manchester est facile à implémenter et évite d'utiliser une deuxième ligne. Cependant, il double la bande passante nécessaire pour un débit donné. Il est utilisé par l'Ethernet à 10 Mbit/s (10BASE5, 10BASE2, 10BASE-T). Les protocoles modernes utilisent des codages plus économes (8 symboles sont codés par 10 transitions dans le cas de 100BASE-TX, par exemple).

1.2.4 Communication asynchrone

Une approche différente consiste à grouper les symboles, par exemple par groupes de huit, et garantir qu'il y a une transition au début de chaque groupe. C'est le principe de la communication *asynchrone*.

L'émetteur procède comme suit :

- lorsque la ligne est libre, elle est dans l'état mark ;
- pour émettre un groupe, l'émetteur émet :
    - un symbole space (le « start bit ») ;
    - 8 symboles codant les données ;
    - au moins un symbole mark (le « stop bit »).

La transition entre le « stop bit » et le « start bit » garantit qu'il y a au moins une transition tous les 10 symboles. Lorsqu'il n'a rien à dire, l'émetteur maintient la ligne dans l'état *mark*, que le récepteur interprète comme un « stop bit » de très longue durée.

La ligne série asynchrone est facile à implémenter, et elle a un côut de 25% (deux symboles tous les huit symboles), ce qui est acceptable. De ce fait, c'était la connexion bon marché dominante jusqu'à l'arrivée de USB.

1.3 Plus de deux symboles

Dans les exemples précédents, le canal permettait de transmettre deux types de symboles (mark et space). Cependant, si le canal est propre, on peut envoyer trois niveaux ou même plus, codés par exemple comme plusieurs tensions différentes. Par exemple, l'Ethernet à 1 Gbit/s sur paire torsadée, 1000BASE-T, utilise cinq symboles distincts.

Comment interpréter ces symboles est le problème de la couche lien. Par exemple, si la couche physique est capable de transmettre trois symboles distincts, la couche lien pourra les interpréter par groupes de deux pour coder trois bits (car deux trits peuvent avoir 9 valeurs distinctes, ce qui permet de coder tois bits).

1.4 Détection et correction d'erreurs

Il n'est pas toujours possible de détecter les erreurs de transmission à la couche physique. Il faudra donc appliquer aux données reçues un algorithme de *détection d'erreurs* et jeter les données erronnées : c'est le rôle de la couche lien. Nous avons vu en TD quelques techniques de détection d'erreurs.

Si la ligne physique est très bruyante, il peut être avantageux d'appliquer des techniques de *correction d'erreurs* ; à la différence de la détection d'erreurs, qui est obligatoire, la correction d'erreurs est optionnelle aux couches basses, car la fiabilité est la responsabilité de la couche transport. Lorsqu'elle est implémentée, c'est généralement à la couche physique.

2 Conclusion

La couche physique permet de transmettre une suite de symboles. Cette suite de symboles n'est pas adaptée aux besoins des couches supérieures :

- c'est un flot de symboles individuels ou de petits groupes de symboles, la couche réseau recquiert des groupes plus grands ;
- il peut y avoir des contraintes arbitraires (symboles insérés, motif oisif) que la couche réseau n'a pas à gérer ;
- il peut y avoir plus de deux symboles, qu'il n'est pas évident d'interpréter comme des bits ;
- le flot peut contenir des erreurs.

Adapter ce flot de symboles aux besoin de la couche 3 est un des rôles de la couche lien.

3 Liens radio

Cette partie n'est pas au programme de ce cours : vous ne serez pas interrogés à ce sujet lors de l'examen.

Les liens radio ont plusieurs caractéristiques particulières qui les rendent difficiles à manipuler. Une de ces caractéristiques doit être gérée par la couche physique : un lien radio transmet la dérivée d'un signal électrique, et pas le signal lui-même, ce qui a deux conséquences :

- il est impossible de détecter le niveau d'un signal continu ;
- il est important d'éviter les variations importantes du signal, dont la dérivée aurait une valeur élevée, et dont la transmission demanderait une quantité d'énergie importante et causerait du bruit à travers tout le spectre.

Le codage d'un signal de façon à éviter le signal continu ainsi que les discontinuités s'appelle la *modulation*. Le décodage d'un signal modulé s'appelle la *démodulation*.

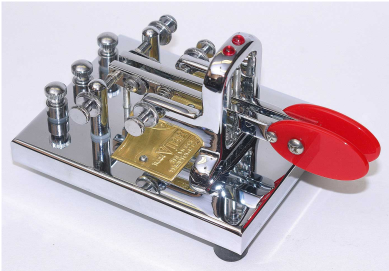
3.1 Modulation à bande étroite

La modulation à bande étroite permet de coder un signal par un signal oscillant autour d'une fréquence f choisie à priori.

3.1.1 Onde continue

En onde continue *(continuous wave, CW)*, deux symboles sont modulés comme suit :

- space : pas de signal : 0 ;
- mark : $\cos(ft)$.



Il faut prendre soin de ne pas introduire de discontinuités aux frontières entre les symboles, en diminuant ou augmentant l'amplitude de façon progressive, ce qui explique le terme *onde continue*.
C'est la technique de modulation utilisée pour les transmissions en code morse.

3.1.2 Modulation d'amplitude

CW est un cas particulier de la *modulation d'amplitude*, qui utilise n symboles :

$$\begin{aligned} s_0 &= 0\cos(ft) \\ s_1 &= a_1\cos(ft) \\ s_2 &= a_2\cos(ft) \\ &\vdots \\ s_{n-1} &= a_{n-1}\cos(ft) \end{aligned}$$

Pour démoduler, on remarque que

$$\int a\cos(ft)\times\cos(ft)\,\mathrm{d}t \sim a$$
$$\int a\cos(ft)\times\cos(f't)\,\mathrm{d}t \approx 0$$

Il suffit donc de multiplier par un oscillateur réglé à la bonne fréquence (un circuit oscillant) puis d'intégrer (un condensateur).

3.1.3 Modulation de phase

En modulation de phase, c'est la phase qui varie :

$$\begin{aligned} s_0 &= a\cos(ft+0) \\ s_1 &= a\cos(ft+\phi_1) \\ s_2 &= a\cos(ft+\phi_2) \\ &\vdots \\ s_{n-1} &= a\cos(ft+\phi_{n-1}) \end{aligned}$$

3.1.4 Modulation de quadrature

La modulation de quadrature combine la modulation d'amplitude avec la modulation de phase :

$$s_{ij} = a_j \cos(ft + \phi_j).$$

Il y a donc un nombre important de symboles distincts (typiquement de l'ordre de la centaine) qui sont soigneusement choisis pour pouvoir être distingués (le nombre de déphasages possibles dépend de l'amplitude). L'ensemble des symboles s'appelle une *constellation*, est il généralement représenté par un ensemble de points du plan (de coordonnées polaires (a_j, ϕ_j)).

On peut montrer que le modulation de quadrature est équivalente à la somme de deux modulations d'amplitude décalées de $\pi/2$ (« en quadrature ») :

$$s_{ij} = b_j \cos(ft) + c_j \sin(ft).$$

Pour démoduler, il suffit de remarquer que :

$$\int (b\cos(ft) + c\sin(ft)) \times \cos(ft)\,\mathrm{d}t \sim b$$
$$\int (b\cos(ft) + c\sin(ft)) \times \sin(ft)\,\mathrm{d}t \sim c$$

Le démodulateur de quadrature consiste donc de deux démodulateurs d'amplitude, déphasés de $\pi/2$.

3.2 Passage aux nombres complexes

Tout devient plus facile si on considère notre onde comme la partie rélle d'une fonction complexe :

$$\begin{aligned} a\cos(ft + \phi) &= \mathrm{R}(ae^{ift+\phi}) \\ &= \mathrm{R}((ae^{i\phi})e^{ift}) \\ &= \mathrm{R}((b + ic)e^{ift}) \\ &= b\cos(ft) + c\cos(ft) \end{aligned}$$

Le nombre complexe $ae^{ift+\phi} = b + ic$ contient toute l'information de l'onde d'origine ; il suffit donc de faire les calculs sur ce nombre, ce qui simplifie drastiquement les notations.

3.3 Modulation à bande large

La modulation à bande large *(broadband)* consiste à superposer plusieurs ondes à base étroite, à des fréquences proches.

3.3.1 Multiplexage par division de fréquences

Un signal important dans la bande de fréquences utilisée *interfère* avec la modulation à bande étroite, même si le signal interférant est à bande très étroite :

$$\int (b\cos(ft) + c\sin(ft) + d\cos(f't)) \times \cos(ft)\mathrm{d}t \sim d$$
$$(d \ggg a, b, f' \approx f)$$

Or, il existe dans le monde moderne un nombre important de sources d'interférence à bande étroite, par exemple les fours à micro-ondes.

Le *mutliplexage par division de fréquences (FDM)* consiste à diviser la bande de fréquences disponible en deux sous-bandes, puis à envoyer simultanément un symbole dans chacune des sous-bandes. Le débit de chaque sous-bande est deux fois plus faible, le débit total reste donc inchangé. Cependant, une interférence à bande étroite n'interfère qu'avec une des deux sous-bandes, ce qui permet de préserver la moitié du signal (ce qui demande d'utiliser des techniques de correction d'erreurs).

3.3.2 OFDM

Si diviser la bande en deux sous-bandes améliore le comportement, il semblerait naturel d'itérer le processus et de diviser en plusieurs sous-bandes. Ce n'est pas possible dans le monde analogique : le nombre de modulateurs, de démodulateurs et de filtres nécessaires serait prohibitif (il faudrait une camionette pour transporter votre carte WiFi).

Ces limitations ne s'appliquent pas au monde numérique : la *transformée de fourier rapide (FFT)* est un algorithme qui permet d'analyser rapidement un nombre arbitraire de fréquences à condition qu'elles soient toutes séparées de la même fréquence (qu'elles soient *orthogonales*). Le *multiplexage par division de fréquences orthogonales (OFDM)* est la restriction de FDM au cas orthogonal, celui qui peut être implémenté de façon efficace en numérique.

Plus précisément, OFDM divise la bande de fréquences en n sous-bandes, avec n ridiculement grand (par exemple, $n = 56$ dans un canal de 20 MHz dans IEEE 802.11g ou n). L'émetteur transmet un vecteur de n symboles simultanément, en y appliquant un algorithme de correction d'erreurs (par exemple, dans IEEE 802.11n, la quantité de données dédiées à la correction d'erreurs varie dynamiquement entre 1/6 et 1/2). Le débit de chaque sous-bande est très lent (au moins n fois inférieur à celui de la bande d'origine), mais la division de fréquences permet de résister aux interférences et de mieux s'adapter aux conditions de propagation et à la législation en vigueur.

Depuis que l'on a appris à implémenter la transformée de fourier rapide en temps réel, OFDM domine les réseaux sans fil : il est utilisé par le WiFi depuis IEEE 802.11a (1999), et par la téléphonie cellulaire depuis LTE (la téléphonie dite « de quatrième génération », ou « 4G »).

3.4 Multiplexage spatial

À des niveaux de bruit réalistes, les débits obtenus par OFDM sont très proches de la limite théorique (la *limite de Shannon*) de capacité d'un canal d'un communication. Cependant, en utilisant plusieurs antennes, il est possible de violer les hypothèses du théorème de Shannon et d'augmenter encore davantage le débit.

SIMO La technique *Single Input-Multiple Output (SIMO)* vise à diminuer le niveau de bruit en recevant le même signal sur plusieurs antennes.

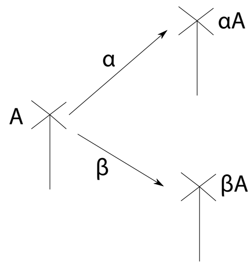
Considérons d'abord un canal OFDM de quadrature avec une antenne de réception (un système « SISO »). Dans chacune des sous-bandes, l'émetteur émet un symbole A qui peut être représenté par un nombre complexe (voir 3.2). Le récepter reçoit

$$\alpha A + n$$

où la partie rélle de α est l'attenuation du signal, la partie imaginaire est le déphasage, et n est le bruit *(noise)*.

Avec deux antennes, le récepteur reçoit deux signaux :

$$(\alpha A + n, \beta A + m)$$

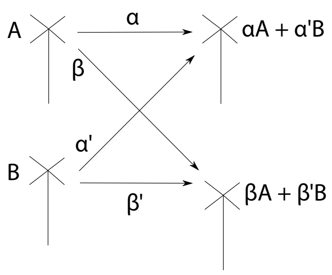


Or, α et β sont connus : il est facile au récepteur de les calculer si l'émetteur envoie un signal connu au début de chaque trame. Le récepteur calcule

$$\alpha^{-1}(\alpha A + n) + \beta^{-1}(\beta A + m) = 2A + (\alpha^{-1}n + \beta^{-1}m)$$

Les signaux s'ajoutent, mais pas les bruits, ce qui a pour effet d'augmenter le rapport signal-bruit, et donc de changer les paramètres du théorème de Shannon.

MIMO Avec deux antennes à l'émetteur, on peut émettre deux signaux indépendants : on parle alors de *Multiple Input-Multiple Output (MIMO)*. L'émetteur émet (A, B) ; le récepteur reçoit



$$\begin{aligned} U &= \alpha A + \alpha' B + n \approx \alpha A + \alpha' B \\ V &= \beta A + \beta' B + m \approx \beta A + \beta' B \end{aligned}$$

Pour récupérer le signal d'origine (A, B), il suffit d'inverser la matrice

$$\begin{vmatrix} \alpha & \alpha' \\ \beta & \beta' \end{vmatrix}$$

Comme les deux antennes ne sont pas au même endroit, les déphasages ne sont pas identiques (avec très forte probabilité), et cette matrice n'est donc pas singulière.