

SEMAINE 2

Concepts et modèle de référence ISO/OSI

Pierre ROLIN

Maître de conférences à Télécom Bretagne

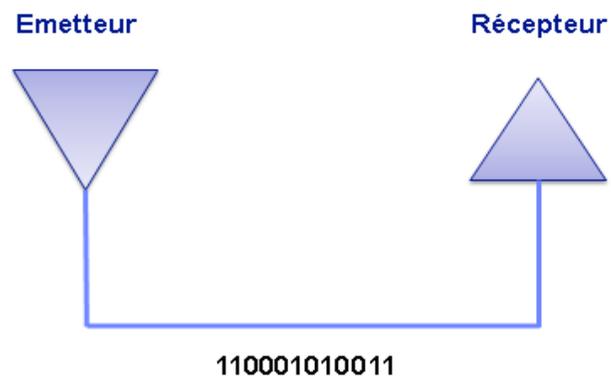
Bonjour, je m'appelle Pierre ROLIN. Je vais vous accompagner pendant cette deuxième semaine.

Semaine 2, Leçon 1 : Notions de voie de communication

Une voie de communication est constituée de 3 parties :

- un émetteur source des données,
- un canal de transmission et
- un récepteur.

Couple Emetteur – Récepteur



Le but de tout réseau de communication est de permettre de construire des voies de communication entre les utilisateurs.

Illustrons ce modèle sur une liaison physique composée d'une paire téléphonique, par exemple.

Le câble de cuivre téléphonique constitue le canal de communication physique. L'émetteur module ses données, par exemple sous forme de niveaux électriques. Le récepteur les démodule pour retrouver les informations émises.

L'information la plus élémentaire est le bit : binary digit, chiffres 0 ou 1. Les ordinateurs travaillent au niveau physique uniquement avec ces deux chiffres.

L'équipement d'émission doit donc transformer les octets qu'il lit dans la mémoire de l'ordinateur en une série de 1 et 0 contenus dans ces octets. On dit que les données sont sérialisées. A l'inverse, le récepteur doit ré-assembler les bits reçus sous forme d'octets qu'il peut ranger en mémoire.

On appelle Modem, pour Modulateur Démodulateur, l'équipement électronique qui réalise ces deux fonctions.

Le **délat de propagation** sur ce câble de cuivre est inversement proportionnel à la vitesse de propagation de l'électricité sur le cuivre et la longueur du câble L/V_p . Le délat de propagation est une composante du délat d'acheminement.

Exemple de voies physiques

Emetteur	Canal	Récepteur
Modulateur électrique	Câble téléphonique	Démodulateur électrique
Modulateur électronique -> optique	Fibre optique	Démodulateur optique -> électronique
Modulateur électronique -> antenne	Fréquence hertzienne	Modulateur antenne réception -> électronique

délat de propagation = Longueur du câble N_p

Mais d'autres supports physiques sont utilisés : la fibre optique, les fréquences hertziennes (Wifi, téléphonie mobile...)... Nous-mêmes, pour être entendus, nous modulons l'air.

Attention, un canal de communication est un concept qui ne se résume pas au canal physique.

Unités de mesure de débit

Unité	Notation	Grandeur
Bits / seconde	B/s	10^0
KiloBits / seconde	Kb/s	10^3
MegaBits / seconde	Mb/s	10^6
GigaBits / seconde	Gb/s	10^9
TeraBits / seconde	Tb/s	10^{12}
PetaBits / seconde	Pb/s	10^{15}

$$\text{Durée de transmission} = T_m / D$$

On appelle **débit**, le nombre de bits émis par seconde sur une voie de communication. Nous le noterons souvent D . Les débits sont notés par multiples de 10 puissance 3 : bit/s, $10^3 \Rightarrow$ Kb/s, $10^6 \Rightarrow$ Mb/s... Les débits ont explosé ces dernières années.

N.B. : En informatique, du fait de l'adressage des mémoires, les quantités de données (par exemple la taille des fichiers) sont des puissances de 2 : Kibi-octets 1024 noté Kio 2^{10} , Mébi-octets noté Mio 2^{20} , Gibi-octets noté Gio 2^{30} .

Durée d'émission



■ Soit

- **D** le débit de la voie de communication
- **T_m** la taille du message

■ Durée d'émission = **T_m/D**

- La Durée d'émission est une composante du délai d'acheminement

■ Exemple :

- Transmettre une image de 4 Mo sur une voie à 1 Mbs aura une durée d'émission de $8 \cdot 4 \cdot 10^6 / 10^6 = 32$ s

La taille d'un message, notée T_m , est limitée. Aussi, la durée d'émission est le rapport « taille du message » sur « débit » : T_m/D . La durée d'émission est une composante du délai d'acheminement.

Lorsque vous avez un document à transmettre, sa durée d'émission est proportionnelle à sa taille divisée par le débit.

Le délai d'acheminement sur un canal physique est la somme du délai de propagation + la durée d'émission.

Au même instant, du fait du délai de propagation, le modem émetteur et le modem récepteur ne travaillent pas sur la même donnée. Ces entités étant distantes, **il n'y a pas d'horloge commune** comme sur un ordinateur. Les deux processus émetteur et récepteur sont dits asynchrones. Les deux processus travaillent en parallèle.

On retrouve, à ce niveau, le parallélisme déjà évoqué mais, à la différence d'un ordinateur, sans possibilité de mémoire partagée ni d'horloge commune.

Voie bidirectionnelle



La voie que nous avons dessinée au début est unidirectionnelle. On dit aussi **simplex**. Par exemple : un canal radio ou un canal satellite.

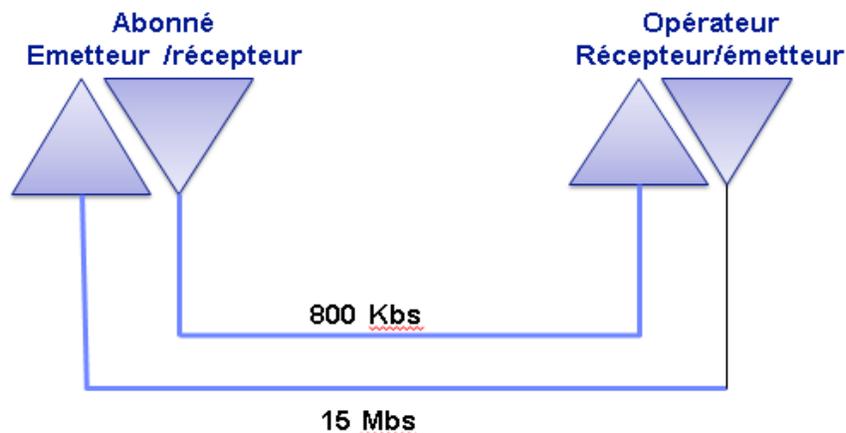
Un canal **half-duplex** est un canal simplex utilisé à l'alternat dans un sens puis dans l'autre.

La plupart des voies sont bidirectionnelles, ou **full-duplex**. Un modem permet l'émission et la réception afin de construire une voie bidirectionnelle. Les fonctions émission et réception travaillent en parallèle.

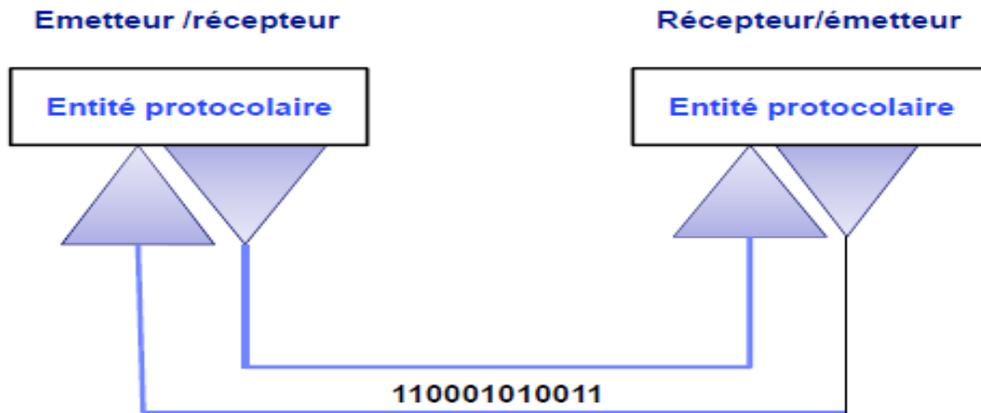
Par contre, chacun des sens de la voie peut avoir un débit propre. C'est le cas de aDSL ou le débit descendant de l'opérateur est plus élevé que celui remontant vers l'opérateur.

Voie bidirectionnelle aDSL

Le débit dans chaque sens peut être différent



Entités homologues



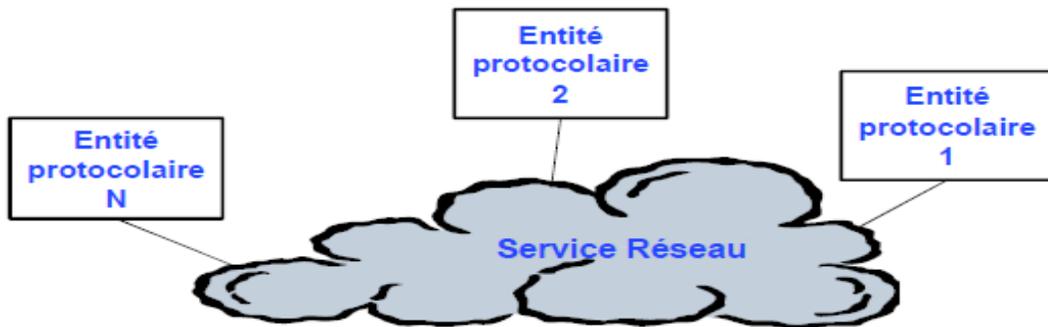
Une voie de communication n'est jamais parfaite. Lors de la description du service postal, nous avons vu que des erreurs peuvent se produire, que les données sont organisées en paquets, que le délai d'acheminement est variable, que plusieurs destinataires peuvent exister derrière une même boîte aux lettres...

Pour améliorer le service offert par un service de communication, donc des voies de communication qu'il fournit, un ensemble de règles sont définies et mises en œuvre. Ces règles sont appelées « protocole » de communication.

Nous appellerons « entité protocolaire » un composant logiciel ou matériel qui exécute ces règles. Une « entité protocolaire » met en œuvre un protocole particulier. Il existe dans les réseaux des centaines de protocoles différents.

Deux entités protocolaires sont dites homologues lorsqu'elles exécutent le même ensemble de règles aux deux extrémités d'une voie de communication.

Sur la figure, nous avons deux entités protocolaires homologues situées aux extrémités d'une même voie.



Un réseau a pour but de permettre de construire une voie de communication entre tout couple d'abonnés. Il est aussi possible de créer des voies multipoint ; donc entre des groupes d'abonnés. Mais ce ne sera pas traité dans ce cours.

Nous pouvons donc remplacer la voie de communication par un nuage qui symbolise le réseau.

Les entités homologues sont des abonnés de ce service réseau.

Les règles qui définissent un protocole sont définies avec la plus grande précision dans des normes et standards internationaux.

L'assemblage d'un protocole et d'un service réseau forme un nouveau service réseau.

Il est donc possible par emboîtement de services et protocoles de construire autant de services qu'on le souhaite.

Nouveau service S'

Emetteur /récepteur

Récepteur/émetteur

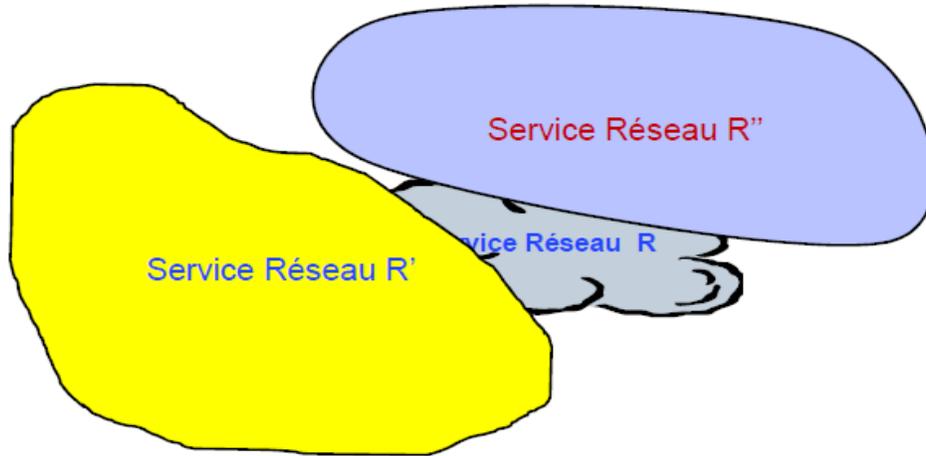


L'association d'un service de communication S et d'un protocole P construit un nouveau service de communication S' doté de propriétés différentes du service S.

On parle aussi de couche protocolaire. L'ensemble des entités qui mettent en œuvre un même protocole au dessus d'un même service réseau forment une couche protocolaire. Toutes ces entités protocolaires sont donc homologues. Dans chaque équipement, une instance de l'entité protocolaire de cette couche doit être implantée.

Vous avez sûrement déjà entendu l'expression « couche IP ». IP est le protocole qui fournit le « Service Internet ». L'entité IP doit être mise en œuvre dans chaque machine qui veut offrir à ses utilisateurs locaux le « service Internet ». La couche IP est composée de toutes ces entités. Nous verrons en semaine 4 ce qu'est le protocole IP.

Le service Réseau R associé au protocole P1 crée un nouveau service réseau R' et associé au protocole P2 un service Réseau R''

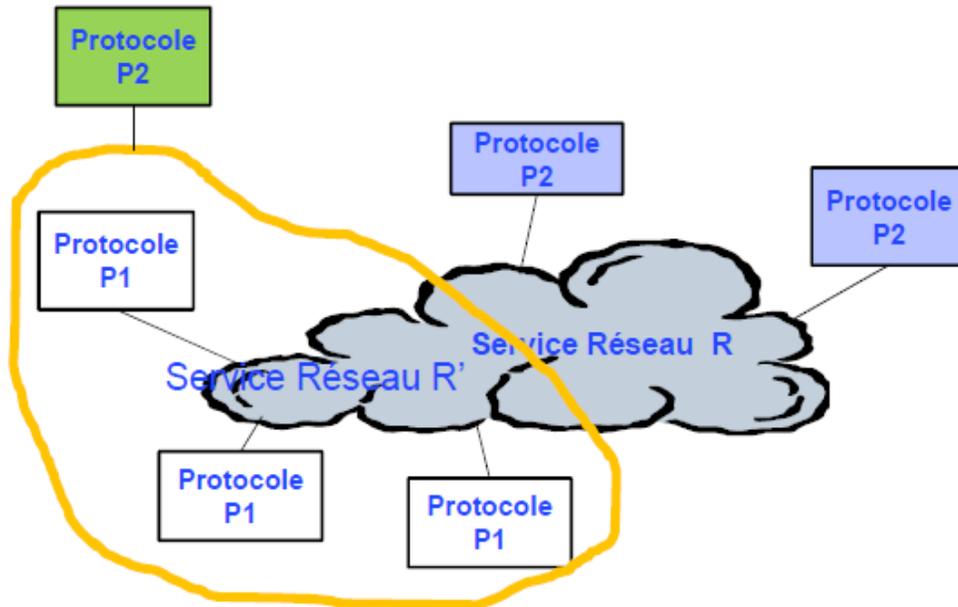


Un service réseau est en général partagé par plusieurs utilisateurs qui peuvent l'utiliser simultanément.

En particulier, plusieurs protocoles différents peuvent utiliser le même service réseau.

Ainsi, sur ce schéma, nous disposons de deux protocoles P1 et P2 qui savent utiliser le service réseau R. R associé au protocole P1 crée un nouveau service réseau R' et R associé au protocole P2 crée un autre service réseau R''.

L'entité protoclaire verte n'est pas homologue des entités protoclairees bleues.
Ces 3 entités protoclairees mettent bien en œuvre le même protocole P2
Mais l'entité P2 verte est sur le service Réseau R' construit au dessus du service réseau R avec le protocole P1 alors que les entités P2 bleues sont directement sur le service R

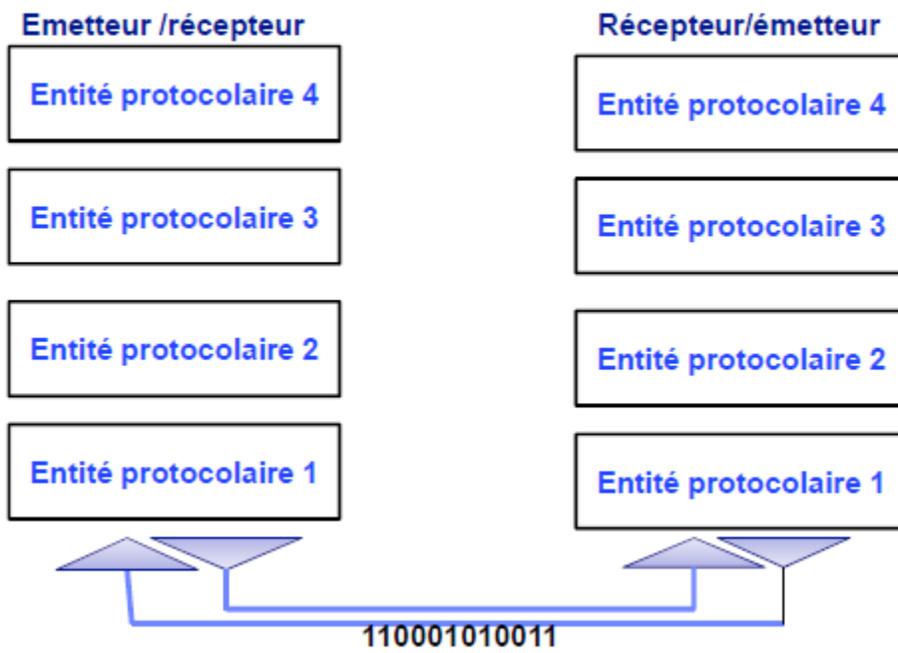


Des entités protoclairees identiques, c'est-à-dire mettant en œuvre le même protocole, ne sont homologues que si elles sont associées à un même service réseau.

Ainsi, sur cette figure, l'entité protoclaire verte n'est pas homologue des entités protoclairees bleues. Ces 3 entités protoclairees mettent bien en œuvre le même protocole P2.

Mais l'entité verte P2 est sur le service réseau R' construit au-dessus du service réseau R avec le protocole P1, alors que les entités bleues P2 sont directement sur le service R.

Pile protocolaire



Un utilisateur peut souhaiter un service qui garantit des propriétés dont il a besoin.

Par exemple, une banque pour ses distributeurs de billets, souhaite que le service de communication assure que les données arrivent dans l'ordre de soumission et sans erreurs, que celles-ci resteront confidentielles et que le délai d'acheminement ne fasse pas trop attendre le client.

Réaliser toutes ces propriétés de service dans une seule entité protocolaire n'est pas une bonne idée. Il est plus simple et efficace de concevoir un protocole pour chacun des besoins de service.

Une fois que l'on dispose d'un protocole pour chacun de ces services, il suffit, pour obtenir un service final qui rend toutes les fonctions souhaitées, **d'assembler les différents protocoles de manière hiérarchique**. Cet assemblage est dit en **couches**.



Modèle ISO/OSI

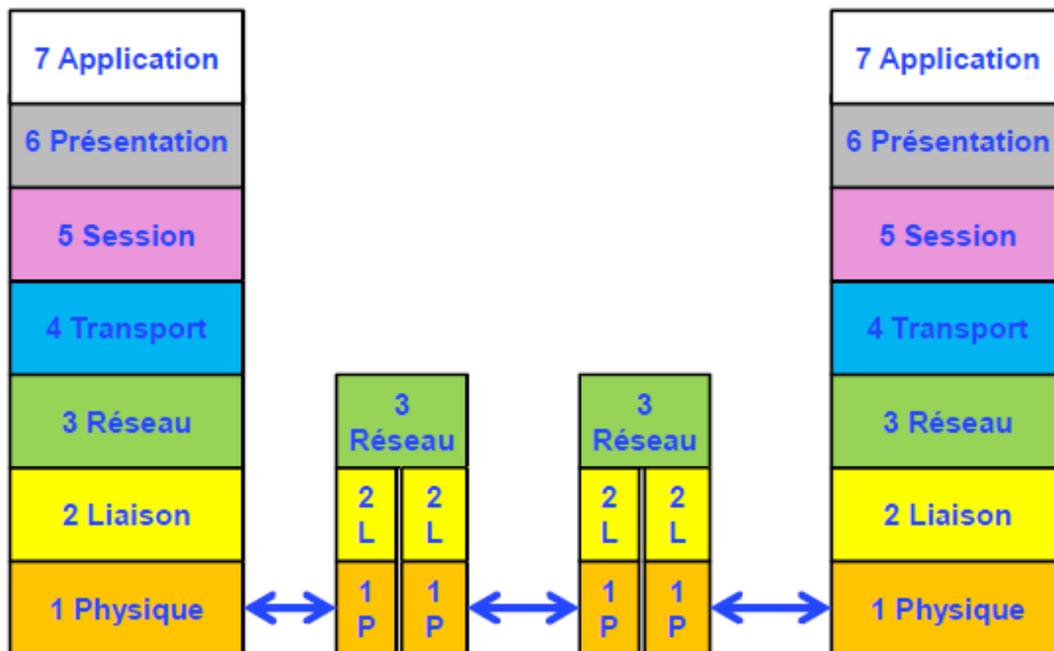
Défini dans les années 70

À la fin des années 70, l'International Standard Organisation a défini le modèle Open System Interconnexion, connu sous l'abréviation modèle ISO/OSI, qui sert aujourd'hui

encore de référence dans la conception des réseaux de données.

Ce modèle définit 7 couches. Au sein de chaque couche, différentes normes et standards définissent les services fournis.

Modèle ISO/OSI



- La couche **physique** définit les caractéristiques du support : métallique, optique, hertzien, micro-ondes, la puissance et portée du signal, les longueurs des câbles, la forme des prises, le codage utilisé pour transmettre des bits sur le câble.

- La couche **liaison** assure le transfert de l'information entre les machines sur la voie physique. Elle détecte en général les erreurs et peut éventuellement mettre en œuvre des mécanismes de correction.

- La couche **réseau** assure la fonction d'acheminement des messages à destination. Elle réalise le routage ou relayage ou commutation, des messages. Elle doit être capable de calculer et trouver les chemins. Pour cela, elle doit disposer de plusieurs liaisons vers des entités réseau voisines jusqu'à atteindre la destination désirée. Il existe de nombreux services réseau. Vous verrez en semaine 4 le protocole IP.

- La couche **transport** assure un transfert de données de manière transparente entre

utilisateurs en les déchargeant des détails d'exécution. Il existe de nombreux services de transport. Vous verrez en semaine 5 les services TCP et UDP.

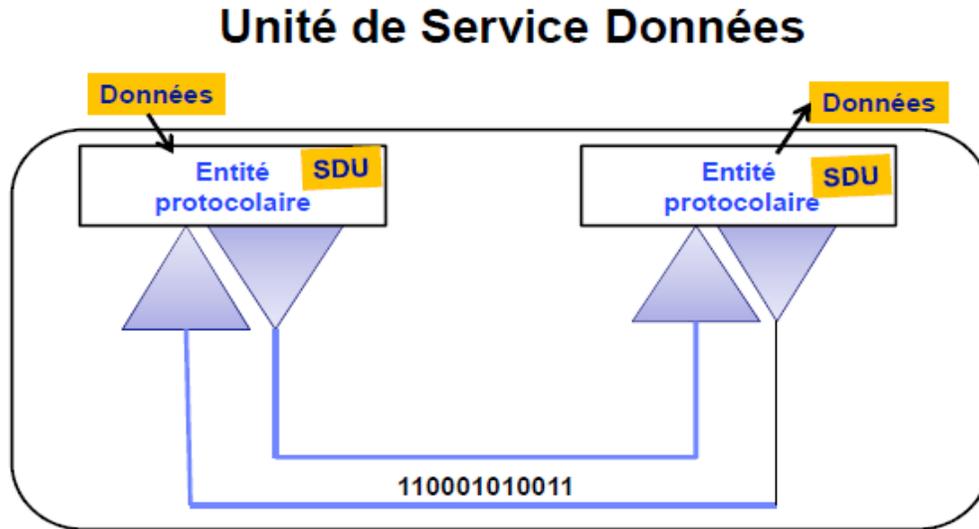
- La couche **session** a pour but de fournir des moyens de synchronisation. Elle est rarement implémentée au-dessus du transport.

- La couche **présentation** se charge de la représentation des informations que des entités d'application se communiquent. Elle est nécessaire du fait de la variété des représentations de données dans les différents systèmes.

La réalité des réseaux a beaucoup bousculé le placement des fonctions dans les couches. Lisez le document chapitre 3 du livre « les réseaux » qui est à votre disposition pour plus de détails.

Dans cette séance, nous avons vu les notions d'entités homologues et le modèle de référence ISO/OSI. La séance prochaine, nous verrons la notion de protocole.

Semaine 2, Leçon 3



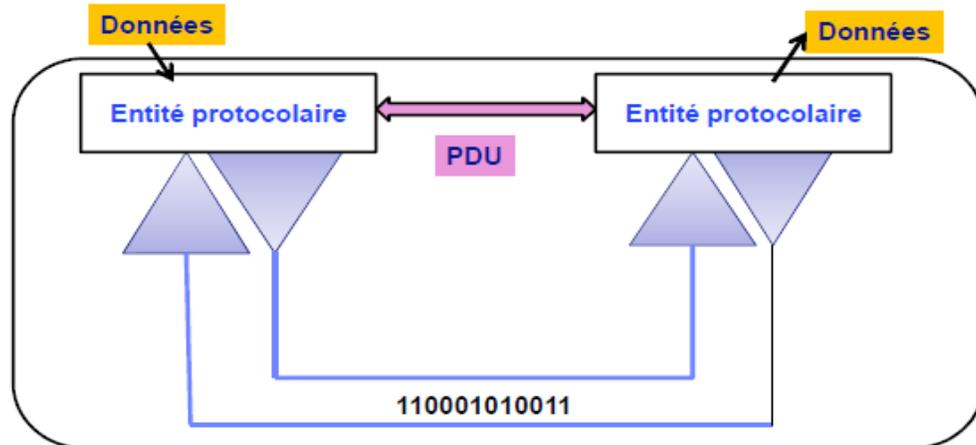
Dans la séance précédente, nous avons vu le modèle ISO/OSI et la notion d'entités homologues. Nous allons étudier dans cette séance la notion de protocole.

Pour réaliser les règles définies par un protocole, les entités homologues ont besoin d'échanger des informations.

Il y a au minimum les informations que l'utilisateur demande de transmettre que nous appellerons **SDU** (pour Service Data Unit) dans l'entité protocolaire.

L'objectif du service de communication est en général de délivrer la même donnée, le même SDU, à destination. Donc ce qui rentre et sort aux extrémités d'une voie de communication est identique.

Unité de Donnée Protocolaire

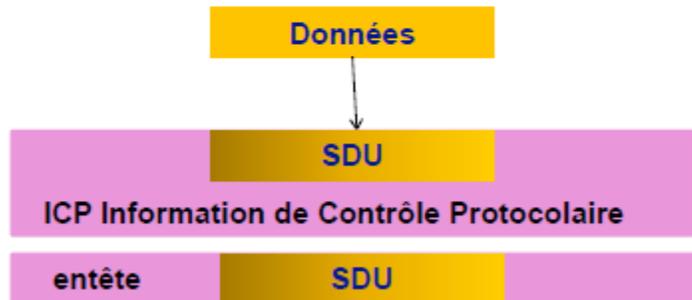


Pour réaliser le service de communication, les deux entités homologues vont s'échanger des messages que nous appellerons **PDU** pour Protocol Data Unit.

Ces PDU vont, en plus du contenu du SDU, contenir des informations dont le protocole a besoin pour construire les propriétés qu'il fournit. Ainsi, par exemple :

- un protocole qui assure que les SDU seront délivrés dans le bon ordre introduira un numéro de séquence.
- Un protocole qui assure que des SDU erronés ne sont pas délivrés introduira un code de détection d'erreur.
- Un protocole avec accusé de réception qui, comme le service du même nom à la poste, veut pouvoir informer l'émetteur que le SDU est bien délivré, introduira des PDU « accusés de réceptions ». Ces PDU circulent dans le sens inverse des données.

Unité de Service Données / Unités de Données Protocolaires



Deux représentations du PDU

On appellera ICP (Informations de Contrôle Protocolaire), toutes ces informations qui

sont ajoutées par les entités protocolaires homologues. On parle aussi d'entête, ou *header* en anglais, car ces informations sont souvent placées avant le SDU dans le PDU. Il en est aussi souvent ajouté à la fin aussi ; par exemple un code détecteur d'erreurs. On peut parler dans ce cas d'enqueue, ou *trailer*, en anglais.

Ces données ne sont utilisées que par ces entités protocolaires. Les utilisateurs du service n'en ont pas connaissance.

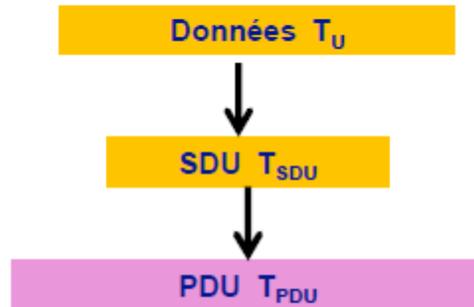
Il faut bien loger ces données supplémentaires. Elles vont donc s'ajouter au SDU. Si le PDU porte toutes les données du SDU, les données de protocole sont ajoutées. Si le PDU sert uniquement à la gestion du protocole, on parle de PDU de protocole.

Il est aussi possible que l'entité protocolaire fasse des opérations sur les données. Aussi, les données soumises peuvent être transformées pour former le SDU. C'est le cas par exemple pour un protocole de compression des données, d'un protocole de chiffrement ou de présentation des données. C'est évidemment le cas dans la couche physique où les données doivent être codées et modulées dans une grandeur physique. Vous verrez la semaine prochaine un exemple de modification des données dans le cas du protocole HDLC.

On appelle souvent « enveloppe » ces informations de contrôle protocolaires que nous avons notées ICP. On dit aussi que les données sont encapsulées dans le PDU.

La description du contenu de l'ICP est toujours décrite dans une norme et est souvent appelée **format de trame**.

Débit Utile / rendement d'un protocole



$$\text{Rendement : } R = T_u / T_{\text{PDU}}$$
$$\text{Débit utilisable : } D_u = R * D_{\text{voie}}$$

Voyons les conséquences de ce mécanisme d'encapsulation sur l'utilisation de la voie de communication.

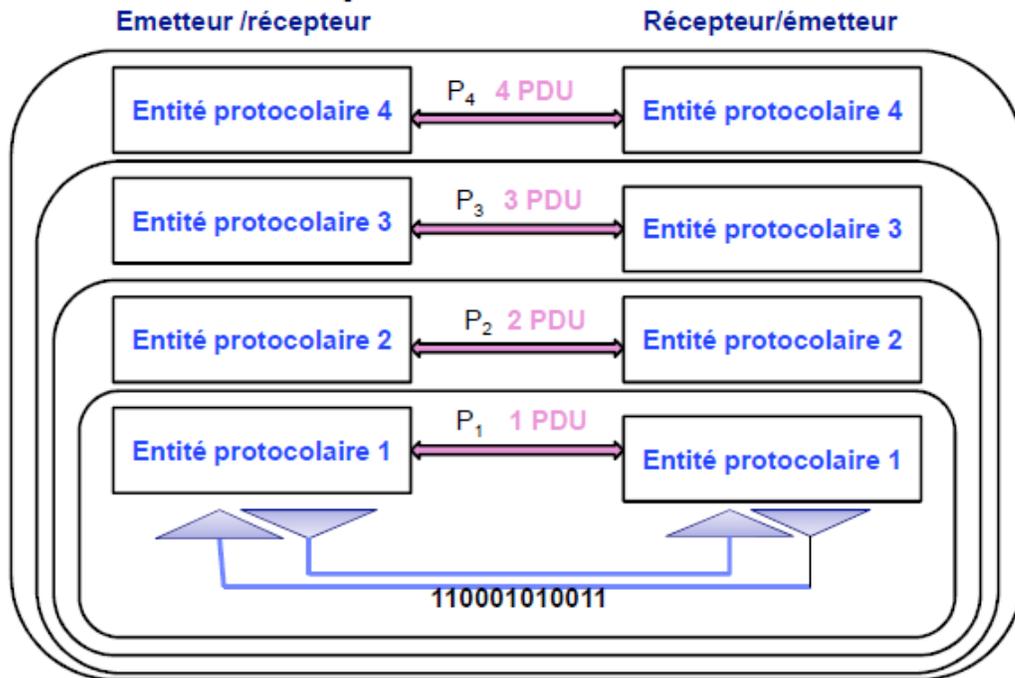
En général, la taille du PDU est supérieure à la taille des données. Une surcharge, overhead en anglais, est introduite. En conséquence, le débit utile pour l'utilisateur, D_u , est la quantité de données utiles, T_u , transmise par unité de temps. Si T_u est la taille des données soumises et T_{PDU} , la taille du PDU, une première définition simple du rendement R d'un protocole est le rapport T_u / T_{PDU} .

Vous voyez ainsi que le débit qui est fourni par un protocole sera différent du débit de la voie utilisée par ce protocole.

$$D_{\text{utilisable}} = R * \text{Débit voie}$$

Vous verrez dans la suite du cours que cette définition basée uniquement sur la taille des messages est insuffisante.

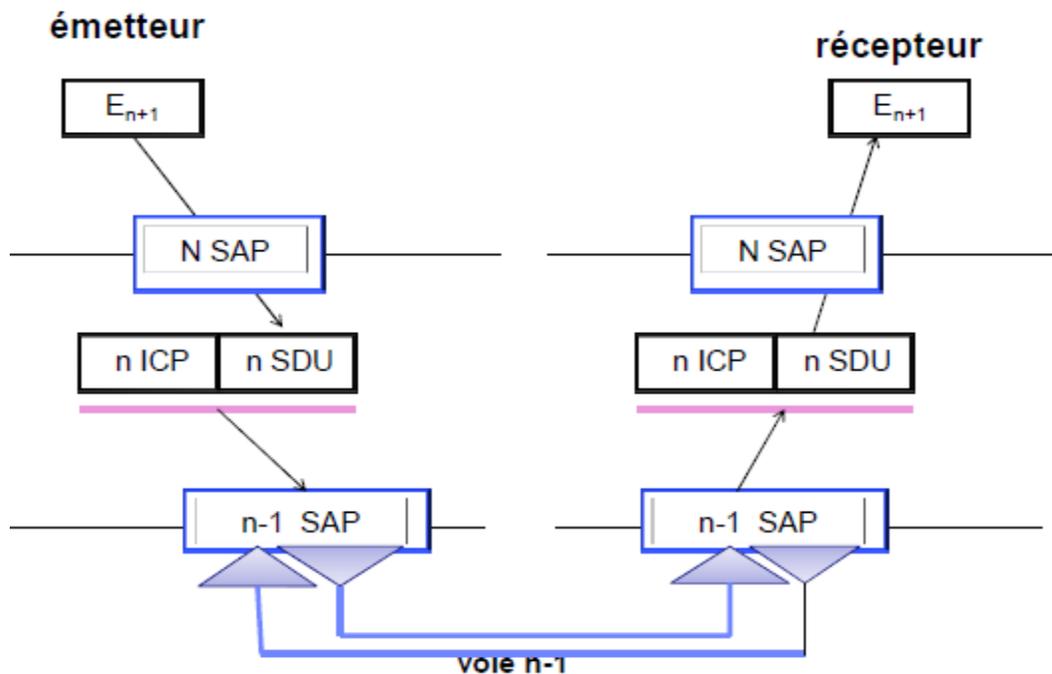
Pile protocolaire



Notre pile protocolaire peut maintenant être complétée par ces messages protocolaires.

Chaque couche utilise des PDU qui lui sont propres. Pour la couche « i » nous les noterons « i PDU ».

Ces i PDU circulent dans les deux sens entre les entités homologues de la couche i.



En conséquence, la taille des PDU évolue jusqu'à la couche physique ; seul niveau où l'information est effectivement transmise à l'équipement distant.

Les données E_{n+1} sont fournies par l'utilisateur qui est situé dans la couche immédiatement supérieure. Un PDU du protocole n est composé de l'information de contrôle protocolaire notée « n ICP », et du n SDU.

Entre la couche $n+1$ et la couche n , il existe une interface que nous appellerons SAP pour Service Access Point. Nous décrirons, dans la prochaine séquence, le rôle de ce Point d'Accès au Service.

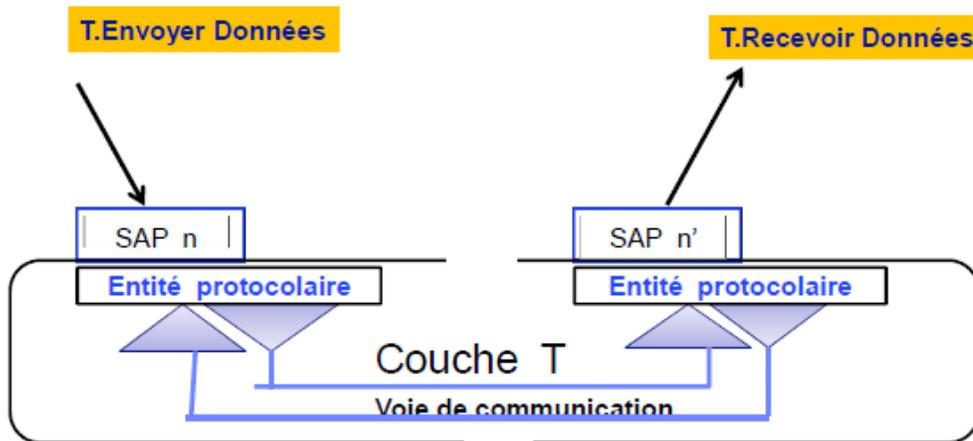
Du côté de l'émetteur, vous observez que la donnée initiale est complétée, on dit aussi encapsulée, dans chaque entité protocolaire traversée, par des informations de contrôle protocolaire. À l'inverse, du côté du récepteur, ces informations de contrôle protocolaire sont extraites et utilisées par le protocole. On dit que l'information utile est décapsulée pour être délivrée à la couche supérieure.

Ces couches et enveloppes sont clairement mises en évidence par l'analyseur de réseau Wireshark que vous installerez et utiliserez en TP en semaine 4.

Dans cette séance, nous avons vu la notion de protocole et ses conséquences. Dans la prochaine séance, nous verrons comment interagissent les couches.

Semaine 2, Leçon 4

Interactions utilisateur - service



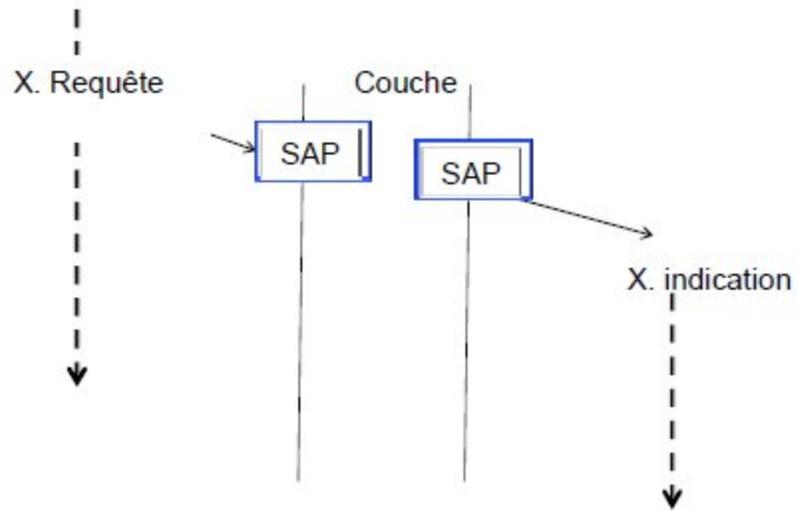
Dans la séance précédente, nous avons vu la notion de protocole et quelques unes de ses conséquences. Dans cette séance, nous allons nous intéresser aux interactions entre couches.

Pour interagir avec le service et réaliser ce qu'il souhaite faire sur le réseau, l'utilisateur d'un service a besoin de primitives et de points d'accès. Pour le service postal, il suffit de deux primitives « poster une lettre » et « retirer une lettre ».

Une couche fournit un ensemble de services au niveau supérieur qui sont invoquées par des primitives. On désigne usuellement ces primitives par un nom précédé de la première lettre du nom de la couche. Ainsi, T.CONNECT est une primitive de la couche transport, N.DATA est une primitive de la couche réseau (Network).

La mise en œuvre de ces primitives est réalisée par le langage de programmation : C, Python, C++, java... Nous décrirons ici seulement un modèle afin de ne pas être dépendant du langage de programmation.

Interaction asynchrone



Selon le sens et la nature de l'interaction, quatre types de primitives sont utiles :

Primitives de Requêtes, Indication, Réponse, Confirmation

Une requête permet à l'utilisateur de prendre l'initiative de transmettre des

informations.

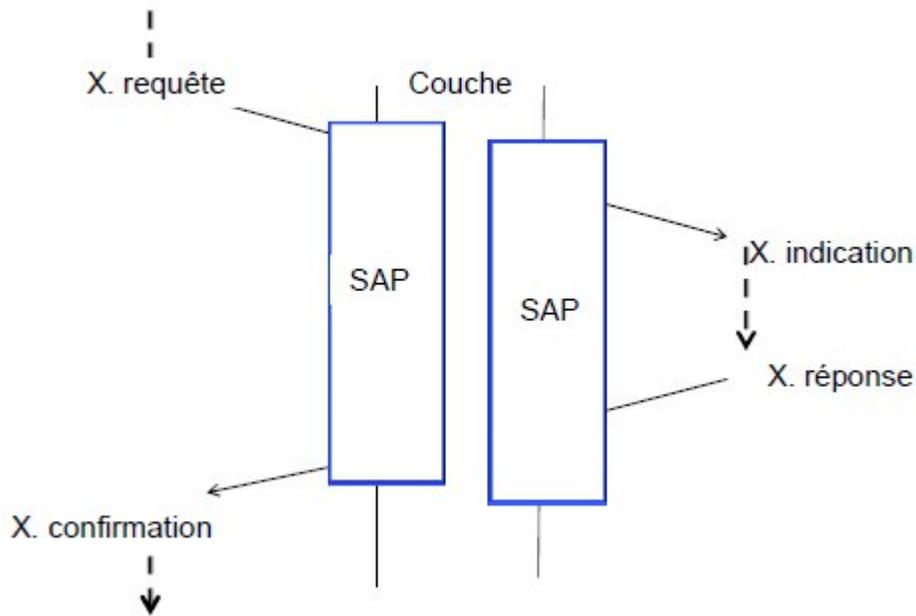
Une primitive de type requête permet à la couche $i+1$, utilisatrice des services de la couche i , de manifester une commande. La partie donnée, donc le message en paramètre de cette primitive, deviendra le i SDU. Le contenu est totalement transparent au prestataire de service, aussi appelé fournisseur de service.

Pour exécuter son protocole, le prestataire de service va échanger avec ses homologues des messages de protocoles, appelés PDU, nécessaires à l'exécution correcte de son service.

Le récepteur ne sait pas quand une donnée va arriver. Il doit donc être alerté par une primitive d'indication. L'esprit du modèle est qu'une indication est un « événement » qui réveille le récepteur. Par exemple, la sonnerie du téléphone alerte sur un appel entrant.

La mise en œuvre dans les langages est en général faite par une instruction qui met le programme en attente.

Interaction synchrone



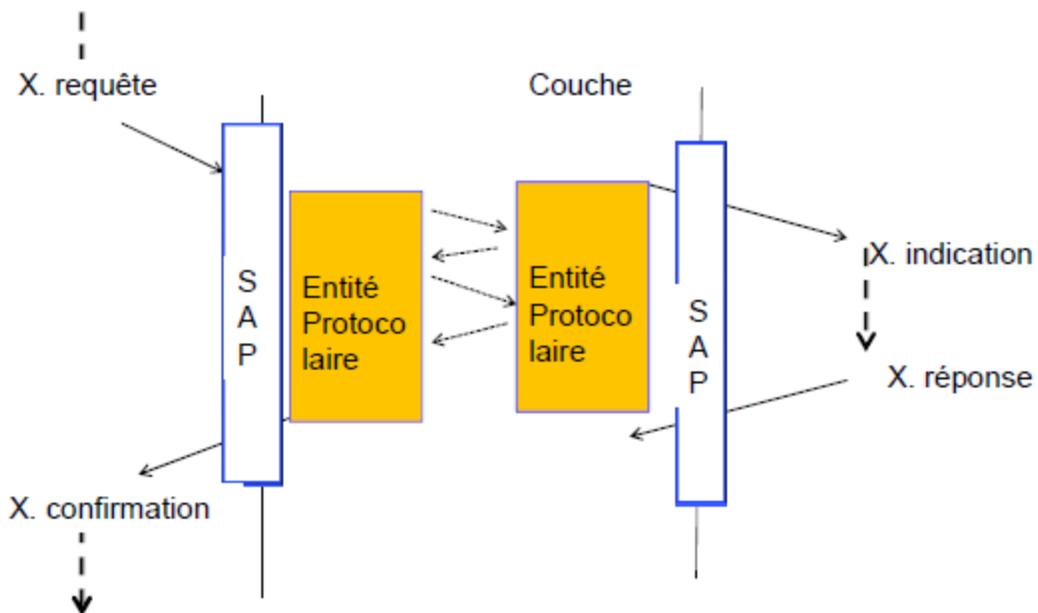
Deux autres primitives sont nécessaires :

- La primitive de réponse permet au récepteur de renvoyer des informations à l'émetteur. La primitive de confirmation permet à l'émetteur de savoir que sa requête s'est bien exécutée.

Dans une interaction asynchrone, la primitive de requête... n'est pas bloquante. C'est-à-dire qu'une fois exécutée, le programme qui l'invoque poursuit son activité. Il ne peut donc pas savoir le résultat de son opération. Dans le cas d'une indication, il s'agira d'une interruption.

Dans une interaction synchrone, la primitive, comme dans un appel de sous-programme, bloque le programme appelant. Une mise en œuvre d'une primitive d'indication synchrone sera réalisée par une demande de réception qui fera attendre le programme appelant jusqu'à ce qu'une donnée arrive.

La communication se fait donc entre deux SAP. Un SAP est donc adressable .

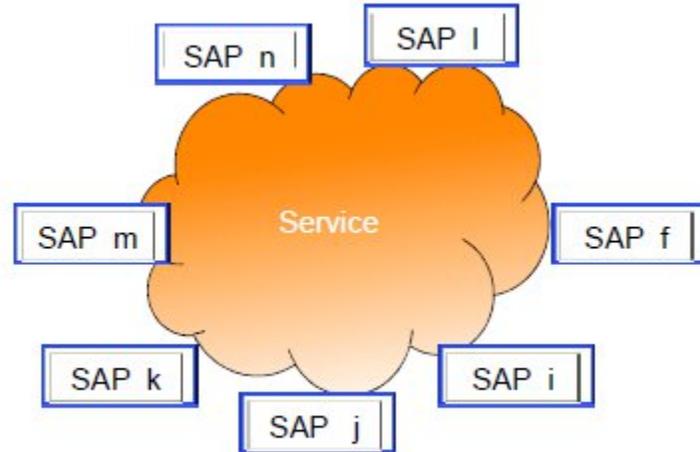


Pour réaliser ces opérations, les entités protocolaires vont échanger un certain nombre de PDU qui sont invisibles des utilisateurs.

Par exemple, les données peuvent être découpées en plus petits blocs, des accusés de réceptions peuvent être utilisés pour vérifier l'arrivée des données...

Nous avons vu en première semaine qu'il existe deux grandes familles de services : les services à datagramme et les services sur connexion.

Dans le cas d'un service sur connexion, il faudra des primitives spécifiques pour établir la connexion. Des données ne pourront être échangées que lorsque la connexion est établie. Les requêtes d'envoi de données auront en paramètre un identifiant de la connexion. Cet identifiant est similaire à la création d'un SAP de part et d'autre sur lequel seront appliqués les requêtes d'envoi de données et les indications d'arrivée de données.



Nous avons vu qu'un service est utilisable par un grand nombre d'utilisateurs et que l'interaction avec le service est réalisée au niveau du SAP. Un SAP possède donc un numéro, une adresse locale à l'entité protocolaire.

Vous pouvez donc voir le service rendu par une couche N comme un service « réseau » sur lequel sont accrochés des points d'accès.

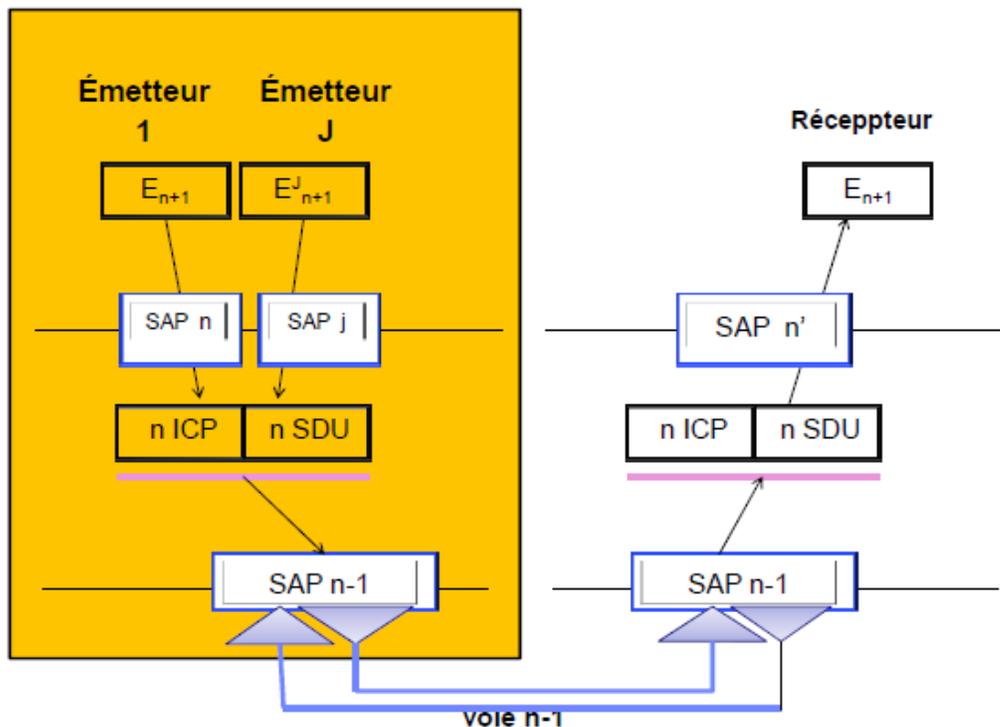
Chaque entité protocolaire peut desservir plusieurs applications au travers de SAP

locaux.

N.B. : une machine physique, avec un seul CPU, ne peut exécuter qu'une seule tâche à la fois, dans le cas de notre pile de protocole . Ce sera le cas pris dans les exercices.

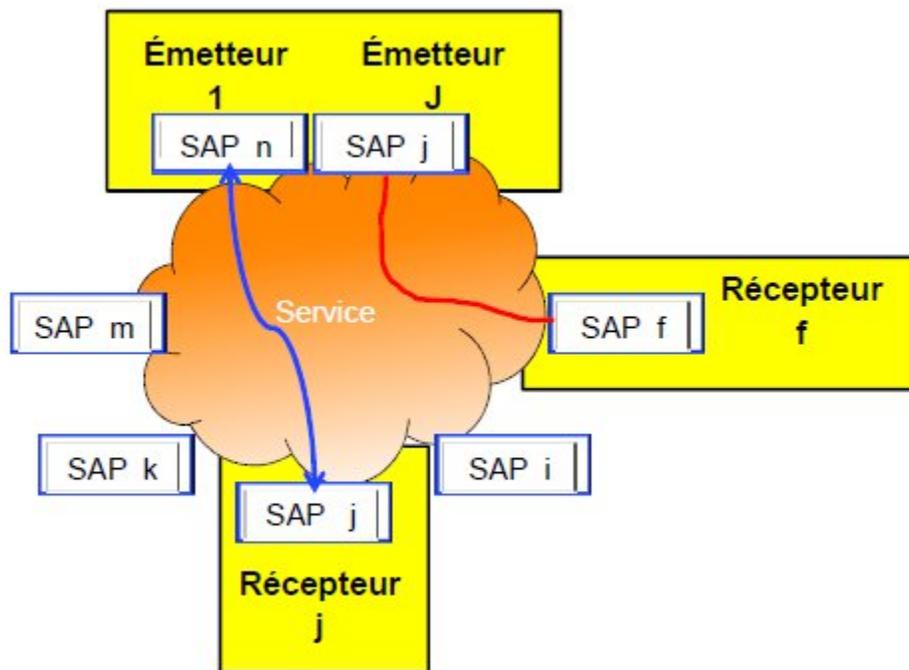
La couche physique, le MODEM, est en général réalisée par un coprocesseur dédié. Il peut donc travailler en parallèle du CPU.

Une machine physique dotée de plusieurs processeurs, multi-core, peut potentiellement exécuter le code d'une entité protocolaire par processeur.



Une même entité protocolaire est mise en œuvre dans une machine physique : le carré jaune sur la figure. Elle abrite J applications. Chacune de ces applications dispose d'au moins un SAP avec l'entité protocolaire n.

Ces applications peuvent être en relation avec des correspondants distincts ou le même, comme sur la figure.



On appelle « multiplexage », le fait de pouvoir faire partager l’usage d’une même voie par plusieurs couples « émetteur récepteur ».

Sur la figure, la voie de communication entre la machine jaune et la machine bleue est multiplexée, partagée entre trois couples d’utilisateurs émetteurs – récepteurs. Le multiplexage est très fréquent dans la couche physique afin d’optimiser l’usage de la capacité. Mais ce concept est général et s’applique à tous les niveaux.

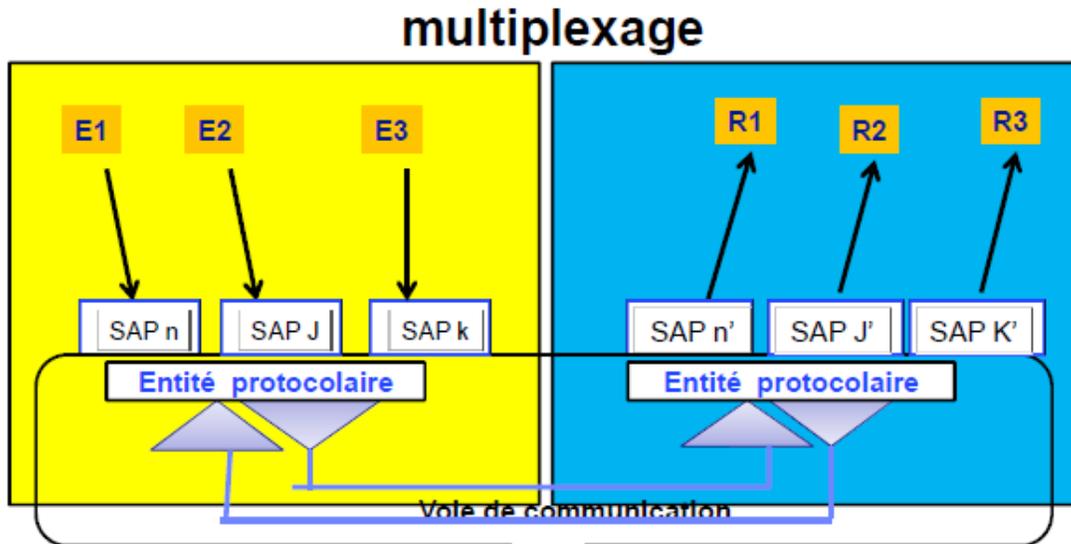
Vous verrez dans les réseaux plusieurs notions de multiplexage :

- Multiplexage temporel : chaque utilisateur dispose de la voie une fraction du temps. Le multiplexage temporel est très utilisé en téléphonie.
- Multiplexage spatial : chaque utilisateur dispose d’une fraction de la bande passante. En téléphonie mobile GSM, chaque utilisateur dispose d’une partie des fréquences.
- Multiplexage par codage : à chaque utilisateur est attribué un code pour la transmission de ses données.

Le découpage des données en paquets de taille limitée est une forme de multiplexage puisque chaque utilisateur a la possibilité de transmettre des PDU à tour de rôle.

Le partage d’une voie de communication suppose l’existence de mécanismes d’allocation

de la ressource partagée, du temps, des fréquences, de la capacité de transmission... Ce sujet n'est pas traité dans ce cours. Vous trouverez dans les chapitres 3 et 6 du livre « les réseaux » plusieurs illustrations de cette importante notion.



Dans cette séance, nous avons traité des primitives de service utilisables sur les Points d'Accès au Service SAP. Dans la prochaine séance, nous parlerons des instances de normalisation.

Semaine 2, Leçon 5

Dans la séance précédente, nous avons parlé des interactions entre couches. Dans cette séance, nous allons aborder les instances de normalisation.

J'ai cité, depuis le début de ce cours à plusieurs reprises, l'importance de la normalisation.

Il existe un très grand nombre d'acteurs : les constructeurs d'ordinateurs, de matériels télécom, de matériels informatique, les sociétés de logiciel, etc.

Pour que deux entités homologues puissent travailler correctement, il est indispensable qu'elles mettent en œuvre exactement le même protocole ; et ce, quelque soit le constructeur et la modalité de mise en œuvre (langages de programmation, matériel...).

Les systèmes en réseau se doivent d'être ouverts à tous types d'acteurs, d'où le nom de la norme OSI, Interconnexion des Systèmes Ouverts.

La loi de Metcalf, inventeur d'Ethernet et fondateur de la société 3Com, dit que la valeur d'un réseau est proportionnelle au carré du nombre d'utilisateurs. Une façon de dire que plus vous avez de correspondants possibles, plus le service a de valeur d'usage. Les systèmes propriétaires utilisant des protocoles non publics sont donc rares et réservés à quelques domaines très spécifiques.

- **ISO/OSI** est l'auteur du modèle de référence norme ISO 7498 . Elle publie de nombreuses normes
- **Internet Engineering Task Force (IETF)** est le principal acteurs pour les protocoles de l'Internet
- **IEEE** dans ses comités 802.x a normalisé l'essentiel des réseaux locaux **Ethernet, Wifi..**
- **UIT Union Internationale des Télécommunication** très actives sur les réseaux de téléphonie mobile

Hélas pour vous, il existe un grand nombre d'instances qui jouent un rôle majeur dans la normalisation ou standardisation. La normalisation a une valeur de droit alors que la standardisation est plus technique.

Nous n'allons pas citer ici toutes les instances, mais seulement celles auxquelles nous ferons référence dans ce cours :

- ISO/OSI est l'auteur du modèle de référence et publie de nombreuses normes.
- IETF, Internet Engineering Task Force, est le principal acteur pour les protocoles de l'Internet. Il publie des RFC, Request for comment.

- IEEE a normalisé dans ses comités 802.x l'essentiel des normes des réseaux locaux Ethernet, Wifi...
- ITU International Télécommunication Union, basée à Genève, est une instance de l'ONU très active dans les réseaux de téléphonie, en particulier mobiles.

ETSI, European Telecommunication Standard Institute, basé à Sophia Antipolis, est officiellement responsable de la normalisation des Technologies de l'Information et de la Communication (ICT) pour l'Europe en coopération avec les CEN et le CENELEC.

- l'ANSI, American National Standard Institute, joue aussi un rôle important.
- Des forums regroupant différents acteurs industriels, constructeurs, opérateurs, sont actifs et régulièrement créés pour faire progresser une technologie et la normaliser. C'est le cas du 3GPP pour la téléphonie mobile 4G.
- Les constructeurs de câbles, de connectiques... sont aussi regroupés en associations.

■ **ETSI European Telecommunications Standards Institute**

[http://fr.wikipedia.org/wiki/European Telecommunications Standards Institute](http://fr.wikipedia.org/wiki/European_Telecommunications_Standards_Institute)

Joue un rôle pivot en Europe puisque cet institut a pour mission la normalisation des Technologies de l'information et de la communication (ICT) pour l'Europe

Nous vous recommanderons, dans ce cours, de consulter divers documents de normalisation ou de standards. Les articles de Wikipédia sur ces sujets sont, la plupart du temps, une entrée suffisante pour les besoins de ce cours.

Dans cette séance, nous avons abordé les acteurs de la normalisation. Dans la dernière séance, nous allons étudier le protocole Ethernet et son PDU, la trame Ethernet.

Semaine 2, Leçon 6

Lors de la séance précédente, nous avons abordé la normalisation. Pour conclure cette semaine, je vous propose de regarder Ethernet.

Ethernet est normalisé dans le standard [IEEE 802.3](#) : Couche média CSMA/CD Ethernet.

Ethernet est un protocole de niveau Liaison, appelée Logical Link Control, car il permet le partage du support de communication.

Il utilise une architecture physique de type bus, c'est-à-dire que tous les équipements raccordés au bus reçoivent les messages. De même, tous les équipements peuvent émettre simultanément.

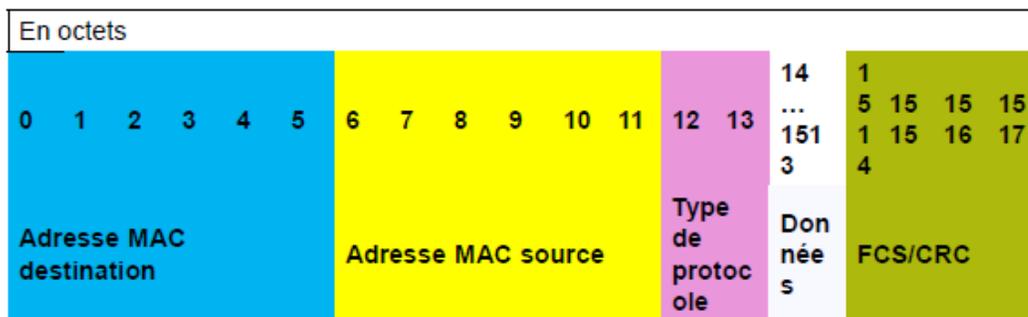
Le protocole Ethernet est destiné à faire en sorte qu'un seul équipement émette à un moment donné. Le mécanisme mis en œuvre par ce protocole est basé sur les règles de politesse. Tous les équipements écoutent le câble et aucun n'émet lorsqu'une émission est en cours. Si personne ne parle, n'émet, alors l'équipement peut émettre. S'il est seul tout va bien. Si 2 équipements ou plus parlent en même temps, tous se taisent et chacun tirera au hasard un délai d'attente avant de reparler.

Je veux surtout ici illustrer les mécanismes de SAP. Sur Ethernet, un SAP s'appelle une

Adresse MAC (Medium Access Control). Elle tient sur 6 octets. Chaque machine physique possède une adresse MAC unique et universelle.

Pour savoir si une trame leur est destinée, chaque machine doit regarder si l'adresse destinataire dans la trame est la sienne.

Format du PDU / trame Ethernet

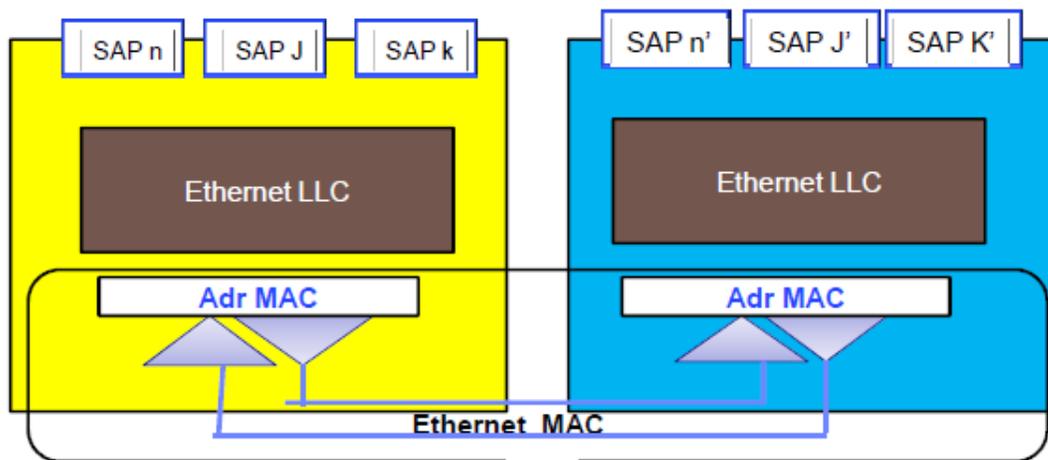


La trame Ethernet a une longueur maximale de 1518 octets et minimale de 64 octets.

Le format de la trame Ethernet est composé de 6 octets pour l'adresse destinataire, 6 octets pour l'adresse source, c'est-à-dire l'adresse de la machine qui a émis cette trame, ce PDU. Elle est utile au récepteur pour savoir à quelle machine répondre le cas échéant.

Puis, vous trouvez 2 octets appelés « type de protocole ». Ce champ est tout simplement le numéro du SAP sur lequel Ethernet doit délivrer le SDU qu'il porte.

La partie SDU est de longueur variable. Elle peut aller de 46 octets à 1500 octets. Il n'y a aucune raison rationnelle à ce choix de longueur maximale de 1500. Par contre, la taille minimale a été imposée par les propriétés physiques du premier réseau Ethernet pour détecter la présence de plusieurs émetteurs simultanés. Si le SDU ne contient pas 46 octets, la couche LLC introduira des octets inutiles dits de bourrage. Enfin, les 4 derniers octets sont un CRC, Code à Redondance Cyclique, destiné à détecter une erreur de transmission. Si une erreur est détectée en réception, la trame est jetée.



En résumé, l'adresse destination est le SAP de niveau inférieur. Le champ « type de protocole » désigne le SAP supérieur.

En travaux pratiques, vous aurez à regarder des trames qui circulent sur un réseau Ethernet et à identifier les protocoles destinataires de ces SDU. Pour cela, il vous faudra au préalable consulter plusieurs standard IEEE et RFC.

La prochaine semaine sera animée par Olivier PAUL qui décrira plusieurs protocoles.

Pour en savoir plus

<http://fr.wikipedia.org/wiki/Ethernet>

et

http://fr.wikipedia.org/wiki/IEEE_802.3#IEEE_802.3_et_standards

[RFC 1166](#) Internet Numbers