

# *MOOC Réseaux Locaux*

## *Le réseaux local Ethernet*

### **Formats de la trame**

#### Objectifs

Cette leçon a pour but de décrire avec quels formats des données sont encapsulées dans une trame Ethernet.

#### Prérequis

Connaissance globale d'Ethernet, notions de trame, d'adresse.

#### Connaissances

Structure de la trame Ethernet.

#### Compétences




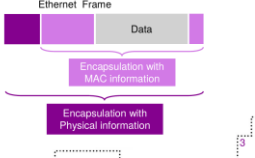

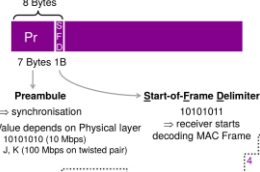
Analyser la structure d'une trame Ethernet.

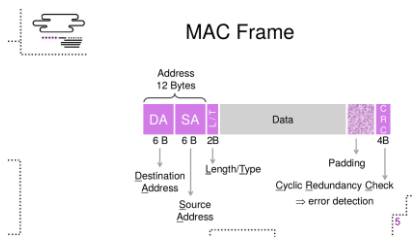
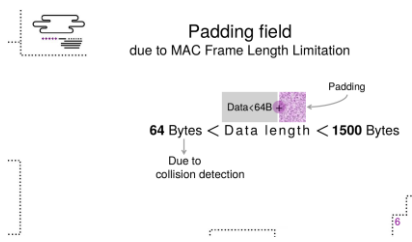
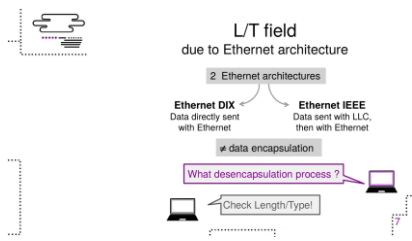
#### Évaluation des connaissances

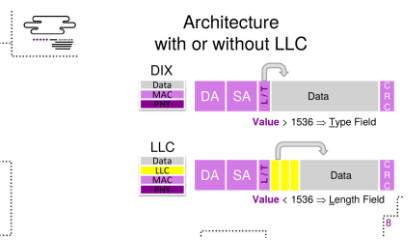
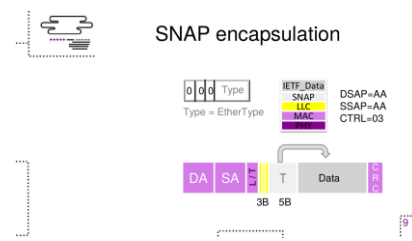
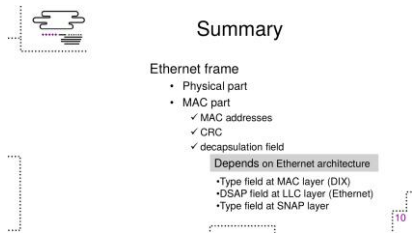
Identifier les composantes d'une trame Ethernet.

#### Évaluation des compétences

Analyser le contenu d'une trame Ethernet.

 <p><b>Ethernet</b> <i>Ethernet frame encapsulation</i></p> <p>Béatrice Paillassa</p>	<p>Nous allons examiner le mécanisme d'encapsulation des trames Ethernet.</p>
 <p><b>Objectives</b></p> <p>≠ Ethernet architectures ⇒ Compatibility ?</p>	<p>Nous allons voir qu'il y a différentes architectures. Nous allons comprendre comment ces architectures peuvent coexister.</p>
 <p><b>Ethernet Frame</b></p> 	<p>Voyons en premier lieu la trame Ethernet. Les données que l'on va vouloir transmettre vont être encapsulées par des informations utiles à la couche MAC. Nous aurons ensuite des informations pour la couche physique.</p>
 <p><b>Physical Frame</b></p> 	<p>Détaillons ces informations de la couche physique :</p> <p>Nous avons 7 octets de préambule qui vont être utilisés par le récepteur pour se synchroniser. La valeur de ce préambule est définie comme une alternance de codage d'uns et de codage zéro. Ceci pour l'implantation à 10 Mbps, mais nous pouvons avoir une définition différente ; par exemple, la version à 100 Mbps sera remplacée par des symboles J, K.</p> <p>Une fois le préambule transmis, l'émetteur va transmettre une séquence clé le <i>starting frame delimiter</i> ; lorsque le récepteur reçoit cette séquence, il est capable de la décoder, il va à ce moment être prêt à décoder les informations qui vont suivre de la trame MAC.</p>

 <p>MAC Frame</p> <p>Address 12 Bytes</p> <p>DA 6 B SA 6 B</p> <p>Length/Type 2 B</p> <p>Data 28 B</p> <p>Padding 4 B</p> <p>Cyclic Redundancy Check → error detection</p> <p>50 B</p>	<p>Les informations qui vont être décodées sont d'abord les 12 octets d'adresse : 6 octets d'adresse destination suivis de 6 octets d'adresse source.</p> <p>Nous avons également 4 octets rajoutés à la fin de la trame : le CRC ; ce sont des informations de redondance pour vérifier que la trame qui est reçue est une trame correcte, qu'il n'y a pas eu d'erreurs de transmission.</p> <p>Cette trame MAC contient également deux champs :</p> <ul style="list-style-type: none"> <li>• Un champ dit longueur/type</li> <li>• Et un champ de <i>padding</i>.</li> </ul> <p>Le champ <i>padding</i>, bourrage en français, est utilisé pour pouvoir transmettre des informations de petite taille.</p>
 <p>Padding field due to MAC Frame Length Limitation</p> <p>Data: 64 B</p> <p>Padding 4 B</p> <p>64 Bytes &lt; Data length &lt; 1500 Bytes</p> <p>Due to collision detection</p> <p>68 B</p>	<p>En effet, dans Ethernet, il y a des limitations sur la taille de données qui peuvent être transmises. Nous allons avoir une limite maximale, 1500 octets, pour les trames de base. Et nous avons, surtout, une limitation sur la taille minimale des trames en raison de l'algorithme utilisé pour détecter les collisions, cette limite est de 64 octets.</p> <p>Lorsque l'émetteur veut transmettre une information de petite taille, inférieure à ces 64 octets, il va falloir rajouter dans la trame du bourrage ; ce sont des informations inutiles qui seront ensuite enlevées par le récepteur.</p>
 <p>L/T field due to Ethernet architecture</p> <p>2 Ethernet architectures</p> <p>Ethernet DIX Data directly sent with Ethernet</p> <p>Ethernet IEEE Data sent with LLC, then with Ethernet</p> <p>data encapsulation</p> <p>What des encapsulation process ?</p> <p>Check Length/Type</p>	<p>Le dernier champ de la trame MAC est le champ longueur/type qui, comme son nom l'indique a un double rôle en fonction de l'architecture Ethernet qui va être utilisée.</p> <p>Précisons, il y a 2 architectures Ethernet :</p> <ul style="list-style-type: none"> <li>• une première architecture qui a été faite par les constructeurs d'Ethernet, va transmettre les données sans utiliser de couches LLC</li> <li>• une deuxième architecture élaborée par IEEE, envoie les informations dans une sous couche LLC.</li> </ul> <p>Le récepteur va avoir un problème pour savoir comment il doit comprendre les données, comment il doit les « dés-encapsuler ».</p> <p>La solution qui a été retenue est d'utiliser un champ longueur/type qui va se situer juste derrière le champ adresse/source de la trame MAC.</p>

 <p>Architecture with or without LLC</p> <p>DIX</p> <p>LLC</p>	<p>En haut nous avons une architecture DIX où il n'y a pas de couche LLC. Le champ qui suit l'adresse/source va être un champ que l'on appelle <i>type</i>.</p> <p>On sait que c'est un champ type parce que sa valeur est supérieure à 1536. Comme les longueurs de trame Ethernet sont inférieures à 1518 (1500 plus 18 octets d'en-tête), nous sommes sûres que ce qui suit n'est pas une encapsulation de type LLC. La valeur de ce champ va indiquer comment décoder les données qui suivent.</p> <p>Dans la trame que se situe dessous, nous avons une encapsulation LLC. Nous allons retrouver le champ L/T et derrière nous aurons donc les 3 octets de l'encapsulation LLC. Nous savons que ces 3 octets sont dans la trame parce que la valeur de champ L/T est inférieure à 1500. Le récepteur sait qu'il doit alors décoder l'encapsulation LLC et que, selon la valeur qu'il trouvera dans les champs SAP (DSAP et SSAP), il comprendra les données qui sont transportées.</p>
 <p>SNAP encapsulation</p>	<p>Dans le cas d'une architecture LLC, nous avons également l'encapsulation SNAP définie par l'IETF pour pouvoir transférer des protocoles IP sur différents réseaux et, en particulier, les réseaux locaux.</p> <p>Cette encapsulation SNAP va rajouter 5 octets. Ces 5 octets se situent après l'en-tête LLC, en jaune sur le schéma, et l'on saura que ces 5 octets sont présents grâce aux valeurs de DSAP et de SSAP, ainsi que la valeur de champ contrôle du LLC. Vous avez les valeurs AA pour le DSAP, AA pour le SSAP, et 03 pour CTRL.</p> <p>Ce champ type va être constitué de 2 octets significatifs qui vont reprendre les valeurs des champs types Ethernet, de l'encapsulation DIX qui avait été définie par les constructeurs. Grâce à eux, on va pouvoir décoder le champ de données qui suit.</p>
 <p>Summary</p> <p>Ethernet frame</p> <ul style="list-style-type: none"> <li>Physical part</li> <li>MAC part <ul style="list-style-type: none"> <li>MAC addresses</li> <li>CRC</li> <li>decapsulation field</li> </ul> </li> <li>Depends on Ethernet architecture <ul style="list-style-type: none"> <li>Type field at MAC layer (DIX)</li> <li>DSAP field at LLC layer (Ethernet)</li> <li>Type field at SNAP layer</li> </ul> </li> </ul>	<p>En résumé, nous avons détaillée la trame Ethernet avec la partie des adresses, du CRC, et surtout le champ d'encapsulation qui permet de comprendre quelles sont les données envoyées dans la trame Ethernet.</p> <p>De ce champ d'encapsulation nous avons vu qu'il n'était pas le même selon l'architecture utilisée. Nous avons vu 2 architectures : l'architecture DIX et l'architecture Ethernet</p>

	<p>L'architecture DIX va utiliser le champ que l'on appelle type, qui se situe au niveau du MAC, la deuxième va utiliser le champ LLC qui grâce aux valeurs de DSAP indiquera les données et nous avons également vu, une troisième façon de dés-encapsuler par l'utilisation du champ type du protocole SNAP.</p>
--	--