

le cnam

# Exercices dirigés

## UTC505/USRS4D -Introduction- E. Gressier-Soudan

2023-2024

*Ce polycopié a été élaboré par l'équipe enseignante "Réseaux et protocoles" à partir d'exercices rédigés par MM. Berthelin, Florin, Gressier-Soudan qu'ils en soient ici remerciés.*

# ED•Encapsulation et Les 7 couches de protocoles

L'exercice 0 correspond à des exercices d'exploration, il n'y a pas de correction donc il ne figure pas dans ce fichier. L'exercice 1 a pour but d'introduire les différentes couches protocolaires et leur rôle. Mais surtout faire découvrir le mécanisme d'encapsulation qui est fondamental et dont on se servira pendant toutes les séances ou presque. Les couches et l'encapsulation sont indissociables.

## Exercice 1 : Les couches Réseaux et Transport fiable TCP<sup>1</sup>, et les niveaux d'encapsulation depuis la trame (couche Liaison) ou le principe des poupées russes/cubes gigognes



©OmniSecu.com

Sources : <https://www.lecoledemesreves.com/les-poupees-russes/> (28/04/2020) ou  
<https://www.doudouplanet.com/Playskool/Cubes-gigognes.2910.html> (06/09/2021) ou  
<https://www.omniseку.com/tcpip/tcpip-encapsulation-decapsulation.php><sup>2</sup> (22/04/2021).

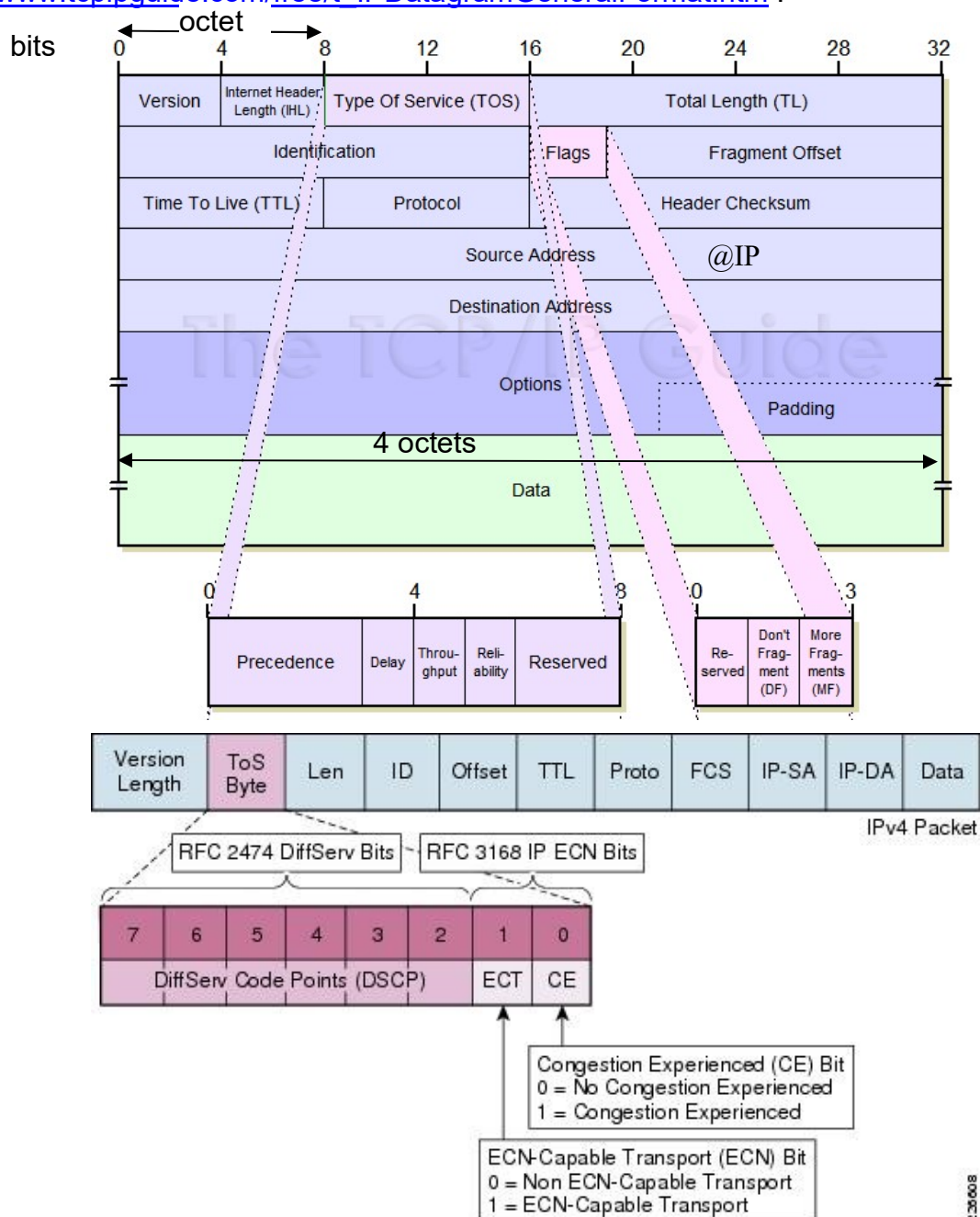
<sup>1</sup> Transmission Control Protocol

<sup>2</sup> Sur <https://www.omniseку.com/tcpip/tcpip-encapsulation-decapsulation.php>, la métaphore des cartons est bien détaillée et intéressante. A regarder !!!!

On donne la structure d'une trame Ethernet :

Adresse Destination MAC	Adresse Source MAC	Type	Informations	FCS
6 octets	6 octets	2 octets	46 à 1500 octets	4 octets

On donne la structure de l'entête IP, consultés le 23 décembre 2013, source [http://www.tcpipguide.com/free/t\\_IPDatagramGeneralFormat.htm](http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm) :



le même entête avec une représentation en ligne, mais sans la taille des champs (source [http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/Medianet\\_Ref\\_Gd/chap4.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/Medianet_Ref_Gd/chap4.html))

et la structure de l'entête TCP, consultée le 23 décembre 2013 source <http://caleudum.wordpress.com/2011/05/08/tcp-header-format/> :

<div>← octet →</div>																																
TCP Header																																
Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Source port																Destination port															
32	Sequence number																															
64	Acknowledgment number (if ACK set)																															
96	Data offset	Reserved		C W R	E C E	U R G	A C K	P S H	R S T	S S Y N	F I N	Window Size																				
128	Checksum																Urgent pointer (if URG set)															
160	Options (if Data Offset > 5)																								padding							
...	...																															

IPv4 est défini dans la RFC (Request For Comment) 791 <http://www.ietf.org/rfc/rfc791.txt>, c'est la version de 1981, Jon Postel, certains champs ont été mis à jour depuis. Par exemple le champ DS vu ci-dessous se superpose au champ TOS <http://tools.ietf.org/html/rfc2474> (RFC2474).

TCP est défini dans la RFC 793 <http://tools.ietf.org/html/rfc793>, c'est la version de 1981, Jon Postel. Depuis, le protocole a été mis à jour. Par exemple le champ ECN, Explicit Congestion Notification <http://tools.ietf.org/html/rfc3168> (RFC3168) a été ajouté et ses circonstances d'utilisation.

Aujourd'hui, la QoS étant déployée dans de nombreux réseaux, le champ TOS peut être redéfini en DSCP Differentiated Service Code qu'on voit dans un autre cours.

Trace d'une communication point à point, prélevée par l'outil SNOOP :

```
ETHER: ----- Ether Header -----
ETHER: Packet 3 arrived at 11:42:27.64
ETHER: Packet size = 64 bytes
ETHER: Destination = 8:0:20:18:ba:40, Sun
ETHER: Source      = aa:0:4:0:1f:c8, DEC (DECNET)
ETHER: Ethertype = 0800 (IP)

IP: ----- IP Header -----
IP: Version = 4
IP: Header length = 20 bytes
IP: Type of service = 0x00
IP:   x xx. .... = 0 (precedence)
IP:   ...0 .... = normal delay
IP:   .... 0... = normal throughput
IP:   .... .0.. = normal reliability
IP: Total length = 40 bytes
IP: Identification = 41980
IP: Flags = 0x4
IP:   .1.. .... = do not fragment
IP:   ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 63 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = af63
IP: Source address = 163.173.32.65, papillon.cnam.fr
IP: Destination address = 163.173.128.212, jordan
IP: No options

TCP: ----- TCP Header -----
TCP: Source port = 1368
TCP: Destination port = 23 (TELNET)
```



```

TCP: Sequence number = 143515262
TCP: Acknowledgement number = 3128387273
TCP: Data offset = 20 bytes
TCP: Flags = 0x10
TCP:      ..0. .... = No urgent pointer
TCP:      ...1 .... = Acknowledgement
TCP:      .... 0... = No push
TCP:      .... .0.. = No reset
TCP:      .... ..0. = No Syn
TCP:      .... ...0 = No Fin
TCP: Window = 32120
TCP: Checksum = 0x3c30
TCP: Urgent pointer = 0
TCP: No options

```

```

TELNET: ----- TELNET: -----
TELNET: ""

```

A votre avis, à quoi correspondent les étiquettes TCP et TELNET (chercher sur le Web) ?

### Correction :

**TCP** : Transmission Control Protocol, indique les champs correspondants à ce protocole, qui est un protocole de couche transport de messages (4) d'application à application et de bout en bout (à travers tout l'Internet).

**TELNET** : c'est un protocole de terminal virtuel en mode caractère, c'est un protocole de couche application (7).

Trace hexadécimale d'une communication point à point :

```

3  0.00000 papillon.cnam.fr -> jordan      TELNET C port=1368

00: 0800 2018 ba40 aa00 0400 1fc8 0800 4500      .. ..@.....E.
16: 0028 a3fc 4000 3f06 af63 a3ad 2041 a3ad      .(.ü@.?..c.. A..
32: 80d4 0558 0017 088d de7e ba77 66c9 5010      ...X.....~.wf.P.
48: 7d78 3c30 0000 0000 0000 0000 0000 0000      }x<0.....

```

Déterminer le début du datagramme IPv4 dans cette trace.

### Correction :

On va d'abord trouver la fin de l'entête de la trame Ethernet, même approche que dans les exercices précédents :

```

00: 0800 2018 ba40 aa00 0400 1fc8 0800 4500      .. ..@.....E.
16: 0028 a3fc 4000 3f06 af63 a3ad 2041 a3ad      .(.ü@.?..c.. A..
32: 80d4 0558 0017 088d de7e ba77 66c9 5010      ...X.....~.wf.P.
48: 7d78 3c30 0000 0000 0000 0000 0000 0000      }x<0.....

```

Attention la première colonne numérote les lignes de la trace hexadécimale.

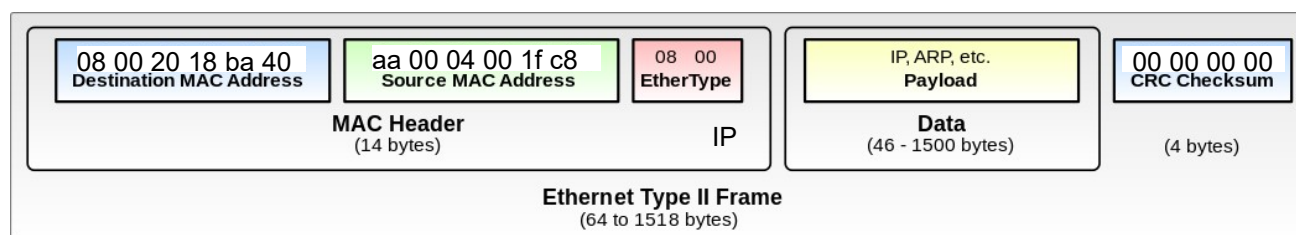
On a encadré aussi le CRC/FCS<sup>3</sup>. Attention, certains outils l'enlèvent complètement (Wireshark par exemple comme vu dans les exercices à base de trace Wireshark), d'autres le mettent tout à zéro. Ce qui reste correspond au datagramme IP. Le premier octet du datagramme est "45".

Ci-après la structure globale de la trame, formalisée d'après la figure donnée dans

<sup>3</sup> CRC = Cyclic Redundancy Code, code détecteur d'erreur, ce principe est vu dans l'ED suivant. FCS = Frame Check Sequence, c'est une autre appellation pour la même chose. A considérer comme synonyme.



[https://en.wikipedia.org/wiki/Ethernet\\_frame](https://en.wikipedia.org/wiki/Ethernet_frame) (22/04/2021)



On peut proposer une métaphore en comparant la trame Ethernet à un wagon plat porte conteneur...



source : <https://www.wascosa.ch/fr/parc-de-wagons/wagons-intermodaux> (14/05/2021)

et l'extraction du datagramme dans la trame pourrait se comparer à l'extraction du conteneur hors du wagon plat :



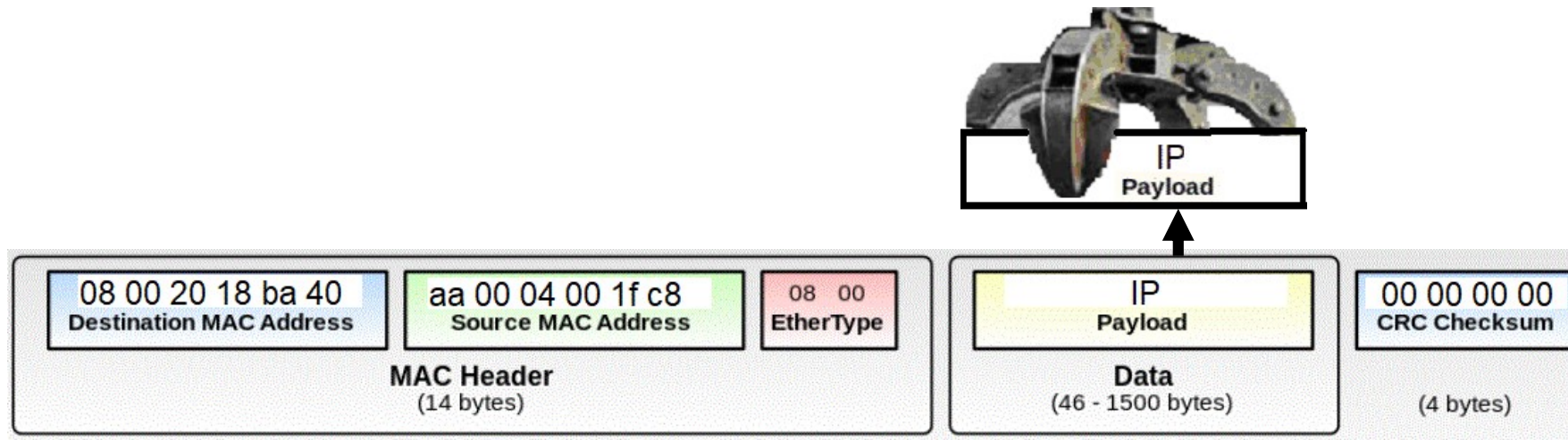
source : <https://www.republicain-lorrain.fr/edition-de-metz-ville/2017/03/14/photos-metz-le-port-se-developpe-par-les-conteneurs> photo 12 (14/05/2021)

enfin, le vidage du conteneur pourrait être comparé à l'extraction du segment TCP qui est dans le datagramme :



même source que ci-dessus, photo 10

Extraction du datagramme de la trame :



Déterminer la fin de l'entête du datagramme IPv4.

### Correction :

Le dernier octet du datagramme est juste avant le CRC ? Et s'il y avait du bourrage ? Comment le déterminer ?

Pour cela, une seule solution, examiner les informations fournies par l'entête du datagramme IPv4 fournie au début de l'exercice. On a mis en noir la partie correspondant à la trame et qui encapsule le datagramme IP. En langage familier, on pourrait prendre le synonyme "contient" pour encapsuler.

```

00: [redacted] 4500
16: 0028 a3fc 4000 3f06 af63 a3ad 2041 a3ad
32: 80d4 0558 0017 088d de7e ba77 66c9 5010
48: 7d78 3c30 0000 0000 0000 [redacted]
  
```

L'entête IP du datagramme est encadrée ci-dessus.

- En **jaune** le numéro de version du protocole, ici "4" correspond à IPv4. Il y a la v6 qui est en cours de déploiement mondial en ce moment.
- En **vert** la longueur de l'entête en mots de 32 bits soit 4 octets. L'entête fait donc 5\*4 octets, soit 20 octets, c'est ce qui est encadré. Il n'y a donc pas d'option dans ce datagramme sinon son entête serait plus longue. En fait, cette situation correspond à la majorité des situations. On verra plus tard dans le cours l'impact de la gestion des options sur la QoS.



- En **bleu turquoise** on a la longueur du datagramme exprimée en nombre d'octets. Le nombre est bien sur en hexadécimal. 28 se traduit en  $2 \times 16 + 8$  soit 40 octets. Le datagramme mesure 40 octets au total.
- Le **dernier octet** du datagramme est aussi en turquoise.

Comme le minimum de données dans une trame est 46 octets, il y a 6 octets de bourrage matérialisé en rouge ci-dessus.

Déterminer la fin de l'entête TCP.

### Correction :

```

00: 0800 2018 ba40 aa00 0400 1fc8 0800 4500
16: 0028 a3fc 4000 3f06 af63 a3ad 2041 a3ad
32: 80d4 0558 0017 088d de7e ba77 66c9 5010
48: 7d78 3c30 0000 0000 0000 0000 0000 0000

```

Là encore on va devoir se servir de la description de l'entête du segment TCP qui est donnée ci-dessus. L'entête IP a été grisée aussi pour résoudre cette question.

On sait que l'entête TCP fait 20 octets. Elle est encadrée sur le schéma ci-dessus. D'après le découpage, on s'aperçoit qu'il n'y a pas de données dans ce segment. C'est curieux... Pas tant que ça, certains segments ne contiennent que des informations protocolaires et aucune donnée applicative.

Comment trouve-t-on que ce segment est associé à TELNET ? C'est par le champ protocole transporté, soit par le **port source** (identifiant de l'émetteur) soit par le **port destination** (identifiant récepteur). Le port destination correspond à "0017" en hexadécimal, la valeur correspondante est 23 en décimal. C'est le port associé à un serveur telnet. C'est donc un segment TCP qui va du client telnet vers le serveur telnet.

La partie telnet est vide, mais cela correspond à une étape du protocole.

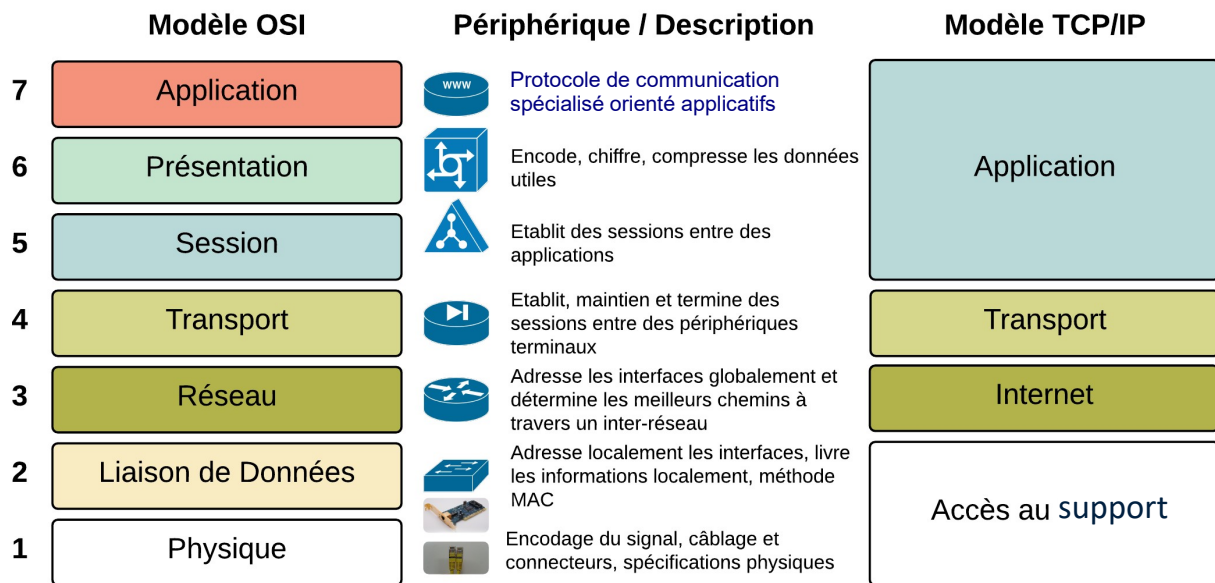
Combien y a-t-il d'encapsulations successives ?

### Correction :

- On n'encapsule pas de données puisque l'entête TCP occupe tout le segment. Ça fait donc 0 pour l'instant.
- On encapsule le segment TCP dans un datagramme IP, ça fait un.
- On encapsule le datagramme IP dans une trame, ça fait deux.



## Exercice 2 : Le modèle OSI et le modèle Internet



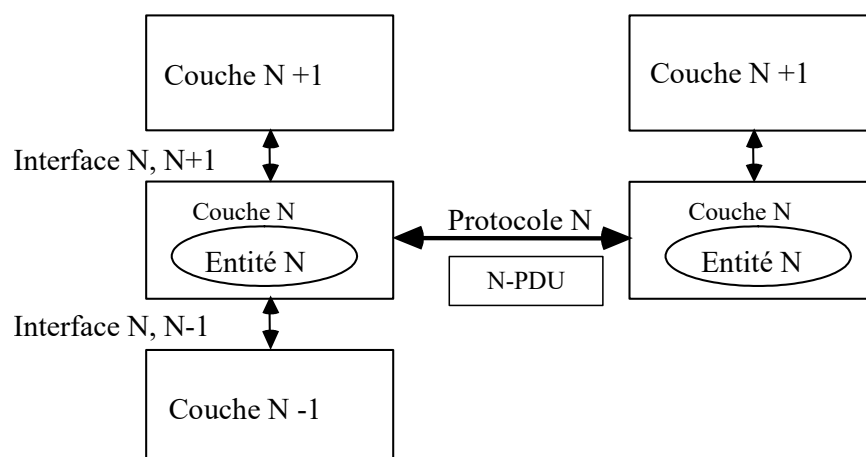
source : <https://cisco.goffinet.org/ccna/fondamentaux/modeles-tcp-ip-osi/>, consulté le 24/04/2019 17h45

Dans le contexte du modèle OSI, qu'est qu'une PDU ? Comment cela se décline pour l'Internet ?

### Correction :

Une PDU, Protocol Data Unit, Unité de données de protocole en français correspond à une unité de message (entête, données voire CRC) échangée entre entités protocolaires de même niveau. On parle plutôt de N-PDU pour PDU de couche N (ne pas confondre avec N pour Network, qui existe aussi).

On rappelle ci-dessous le dessin donné en cours :



Les PDU de l'Internet sont : Bit (Couche 1), Trame (2), Datagramme (3, IP), Segment (4, TCP), Datagramme (4, UDP), Message pour 5 à 7.

Le nombre de couches est différent entre le modèle ISO et le modèle des couches associé à l'architecture Internet.

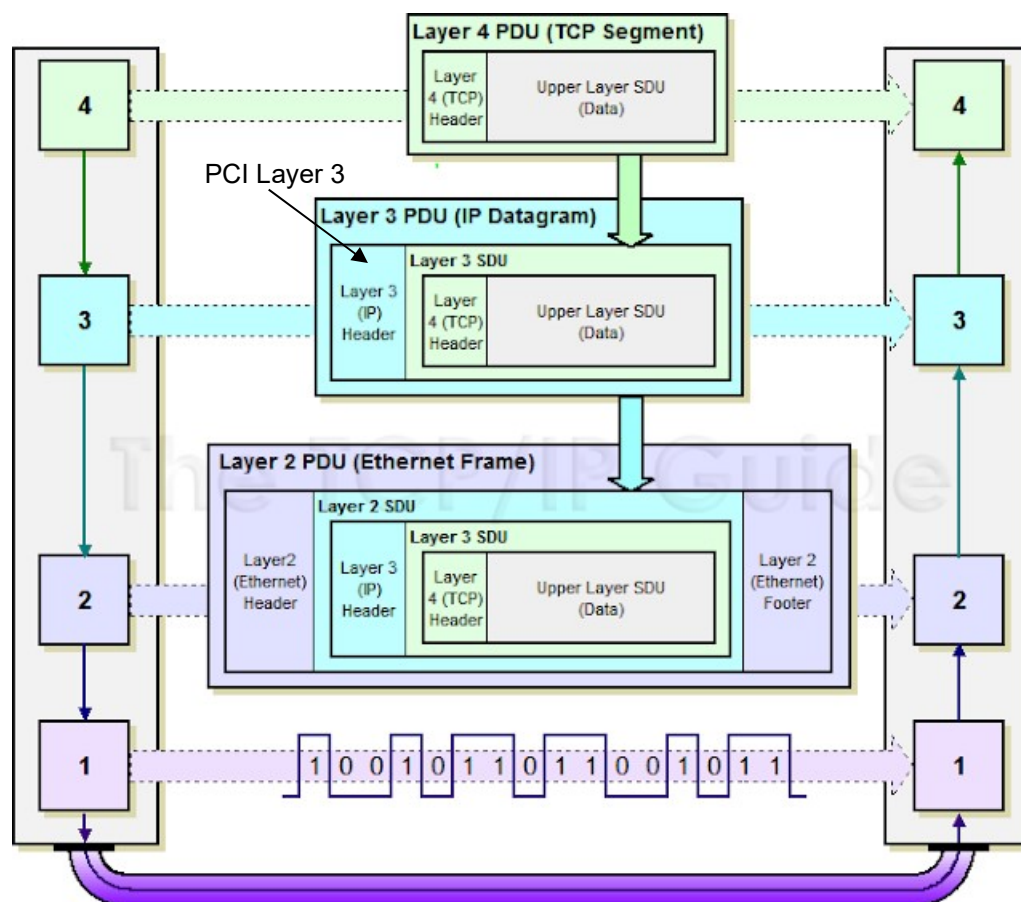
Avec la métaphore des poupées matriochka : 7 poupées emboîtées pour l'OSI (1 à 7) et 5 poupées emboîtées pour l'architecture Internet (1 à 4 + 7) Dans le modèle Internet la couche 7 est un peu fourre-tout entre l'applicatif, et les protocoles spécialisés comme http, SMTP, FTP... Dans le modèle ISO, l'applicatif est en couche 8 non formalisé dans le modèle ISO, mais formalisé dans une autre norme qui lui est dédiée complètement et qui s'appelle RM-ODP, Reference Model Open Distributed Processing.

Dans tous les cas, la poupée la plus au cœur représente les données à vocation applicative.



sources : <https://www.chez-les-enfants.fr/shop/product/tob2307-poupees-russes-matriochkas-en-bois-720> et <https://www.cdiscount.com/juniors/poupees-poupes/tempse-5pcs-poupees-russes-en-bois-fraise-motif-en/f-12064-tem6296623031291.html>

L'encapsulation donne la figure suivante pour les 3 couches Liaison, Réseau, Transport de l'Internet :



source : <https://blog.3g4g.co.uk/2009/03/difference-between-sdu-and-pdu.htm> (consultée le 22/09/2020)



**Merci pour votre attention !!!!!**

---

<sup>4</sup> Troupe d'élite dans BoomBeach de SuperCell, bombardier lanceur de pastèques explosives, [https://boombeach.fandom.com/wiki/Melon\\_Bombardier](https://boombeach.fandom.com/wiki/Melon_Bombardier) (29/08/2021)