

1 Couches Tcp/Ip

La pile TCP/IP contient quatre couches.

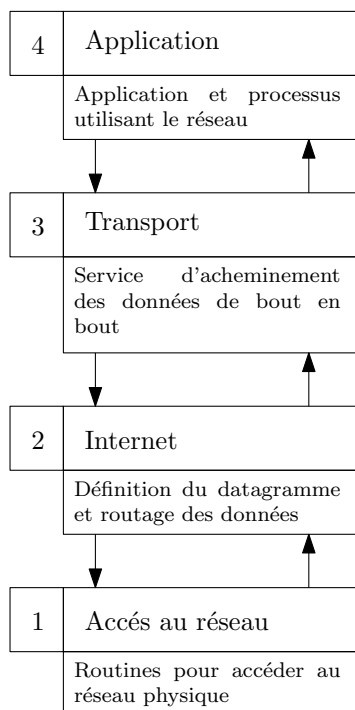


Figure 1: Couches Tcp/Ip

À titre d'exemple, la couche *Accès au réseau* contient les éléments physiques de la communication : liaison filaire (paires torsadées, fibre optique...) ou sans fil (WiFi...). À l'opposé, la couche *Application* contient entre autres ce qui concerne l'interrogation d'un serveur Web. On comprend bien l'intérêt du modèle :

Pour définir les modes d'interrogation d'un serveur Web, nul besoin de connaître les détails du type de connexion, filaire ou non : on fait abstraction des couches basses

2 Encapsulation

Lorsque des données issues de la couche application sont émises vers la couche transport, cette dernière conserve les données originales, et ajoute des informations propres à sa fonction au début des données initiales. On dit alors que les données de la couche application sont *encapsulées* dans une structure plus grande, qui sera émise par la couche transport. Il en est de même à tous les niveaux, si bien que chaque couche ajoute ses propres informations à l'information initiale, comme indiqué sur la figure 2.

Les différentes couches n'ont pas connaissance de l'encapsulation réalisée par les autres couches. Par exemple, lorsque la couche internet reçoit des informations, ces informations sont constituées de l'en-tête de la couche transport et des données initiales issues de la couche application. Mais pour la couche internet, il s'agit *juste* d'informations, qui peuvent contenir n'importe quoi, et auxquelles elle rajoute son propre en-tête. Lors de la remontée d'informations vers l'application, chaque couche se contente de retirer son propre en-tête (qui est bien sûr normalisé) avant de transmettre les données à la couche supérieure.

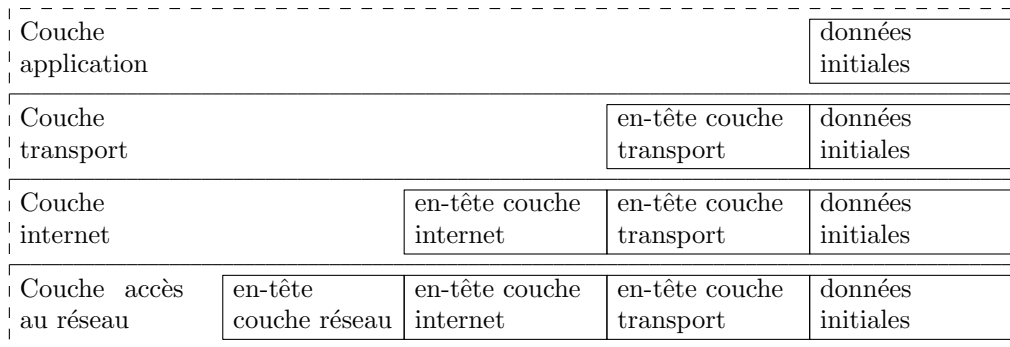


Figure 2: Encapsulation Tcp/Ip

3 Protocoles

Dans chaque couche vivent plusieurs protocoles (figure 3). Les plus familiers (et probablement les plus nombreux) étant naturellement ceux de la couche application.

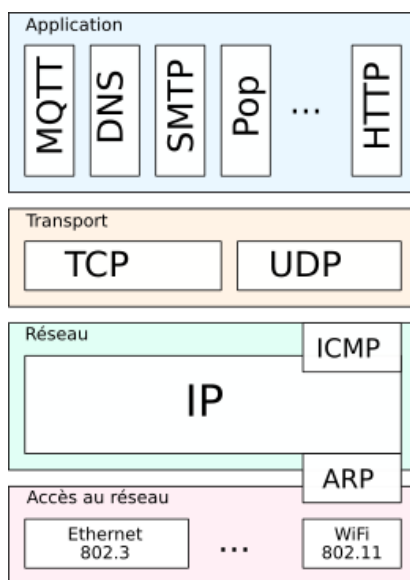


Figure 3: Quelques protocoles réseaux

4 Trame réseau

Une trame réseau est une séquence d’octets. La connaissance des protocoles, du format des entêtes etc. permet de décoder une trame :

```
98 9C 57 7C D0 2E B0 5A DA 3B 5A 7B 08 00 45 00 00 52 D3 B1 40 00 40 06 B5 F7 C0
A8 01 64 C2 FE 2B F2 CA 46 00 50 2C D8 86 52 13 5A 58 23 80 18 01 F6 B1 41 00 00
01 01 08 0A DC 97 FB A3 CC 0E CD 9C 47 45 54 20 2F 64 65 6D 6F 2F 70 61 67 65 2E
68 74 6D 6C 20 48 54 54 50 2F 31 2E 30 0D 0A
```

Voici comment la trame est interprétée par l’outil en ligne Hex packet decoder¹ (on pourrait aussi utiliser le logiciel Wireshark) :

¹<https://hpd.gasmi.net/>

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
98	9C	57	7C	D0	2E	B0	5A	DA	3B	5A	7B	08	00	45	00
00	52	D3	B1	40	00	40	06	B5	F7	C0	A8	01	64	C2	FE
2B	F2	CA	46	00	50	2C	D8	86	52	13	5A	58	23	80	18
01	F6	B1	41	00	00	01	01	08	0A	DC	97	FB	A3	CC	0E
CD	9C	47	45	54	20	2F	64	65	6D	6F	2F	70	61	67	65
2E	68	74	6D	6C	20	48	54	54	50	2F	31	2E	30	0D	0A

3 protocols in packet:



Frame 1: 96 bytes on wire (768 bits)

Encapsulation type: Ethernet (1)

Arrival Time: Feb 24, 2020 17:14:14.000000000 CET

Ethernet II

Destination: HuaweiTe_7c:d0:2e (98:9c:57:7c:d0:2e)

Source: HewlettP_3b:5a:7b (b0:5a:da:3b:5a:7b)

Type: IPv4 (0x0800)

Internet Protocol Version 4

0100 ---- = Version: 4

---- 0101 = Header Length: 20 bytes (5)

...

Time to live: 64

Protocol: TCP (6)

...

Source: 192.168.1.100

Destination: 194.254.43.242

Transmission Control Protocol

Source Port: 51782

Destination Port: 80

...

TCP segment data (30 bytes), HTTP :

GET /demo/page.html HTTP/1.0