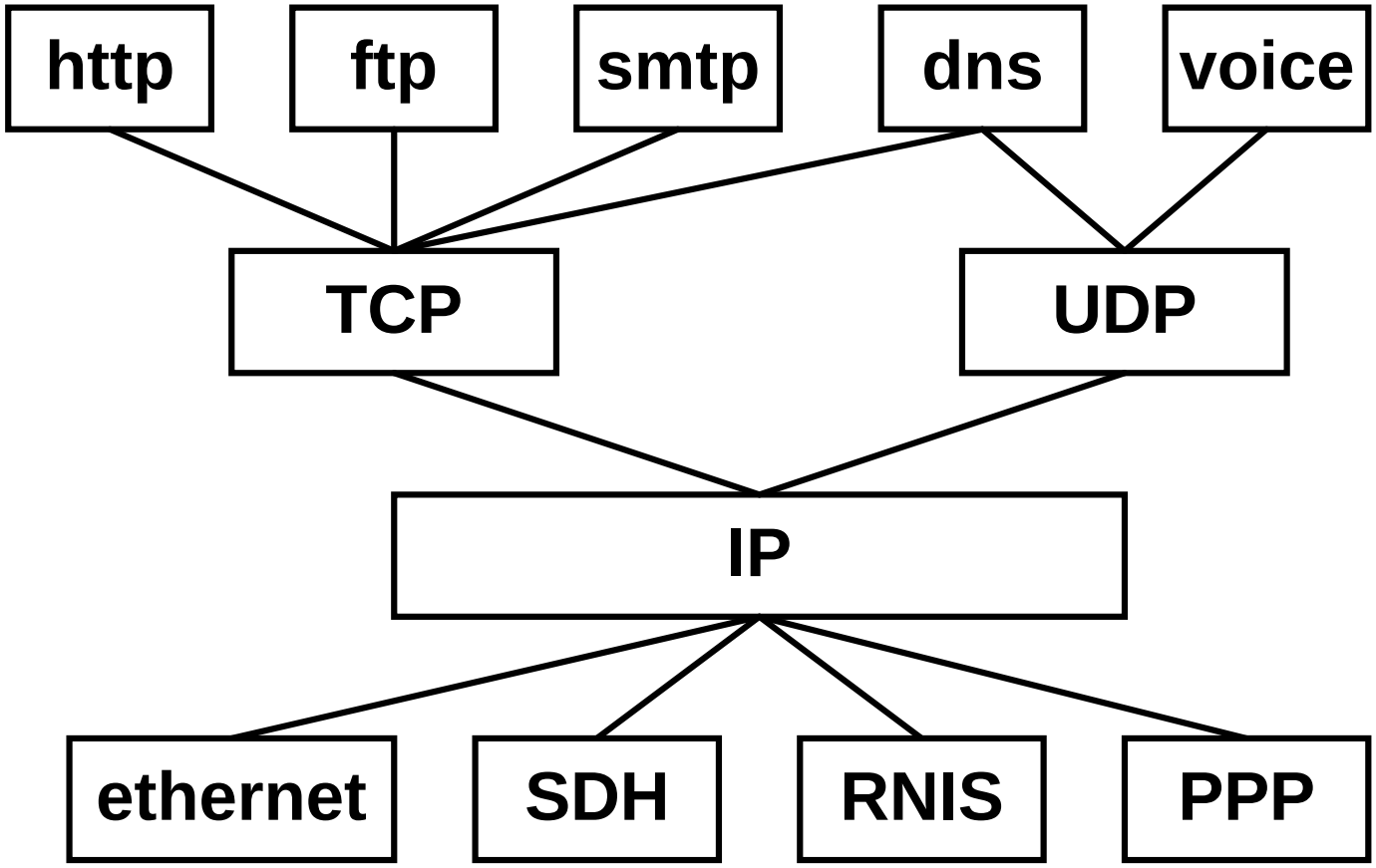
RESEAUX : Protocoles

- **Famille de protocoles TCP/IP**
- ***Internet Protocol* (IP)**
- ***Address Resolution Protocol* (ARP)**
- ***Ping* et *Internet Control Message Protocol* (ICMP)**
- ***Transmission Control Protocol* (TCP)**
- ***User Datagram Protocol* (UDP)**
- ***Domain Name System* (DNS)**
- **Applications *web & Protocol Stack***
- ***Loopback***
- **Labo B1 – B2 : analyse de protocoles**
- **Principales RFCs**
- **Livres & URLs**

© GL

Famille de protocoles TCP/IP



© GL

Introduction (1)

- **Ce chapitre va étudier la famille de protocoles TCP/IP issus du monde Unix et largement répandus**

- **Ils ont été développés (dès 1970) sur l'initiative du département américain de la défense (*DoD Department of Defense*) par des chercheurs travaillant autour de DARPA *(Defense Advanced Research Project Agency*)**

- **L'exemple le plus illustre en est le réseau *Internet* mis en service dans les années 80 et qui continue de s'étendre (plusieurs millions d'ordinateurs aujourd'hui)**

© GL

Introduction (2)

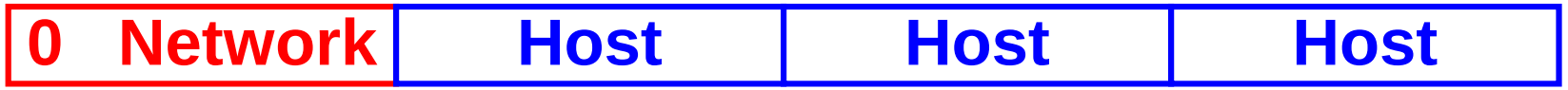
- **D'abord réservé au milieu académique, ce réseau mondial a vite intéressé le monde des affaires**
- **Son immense succès est aussi lié à la technologie WWW (*World Wide Web*) qui autorise un non spécialiste à *surfer on the net***
- **Avec l'*internet* commercial, ces protocoles sont livrés en standard dans les postes de travail actuels**
- **Les RFC (*Requests For Comment*), sortes de rapports techniques, décrivent ces différents protocoles**

© GL

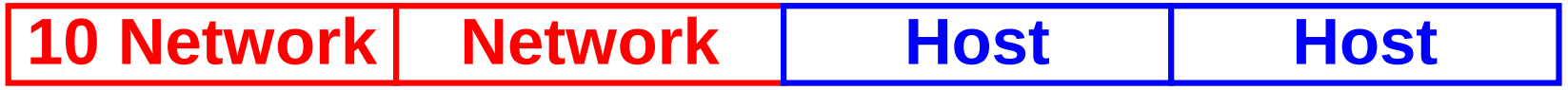
Adresse IP (1)

- **Adresse IP (32 bit) = network + host**

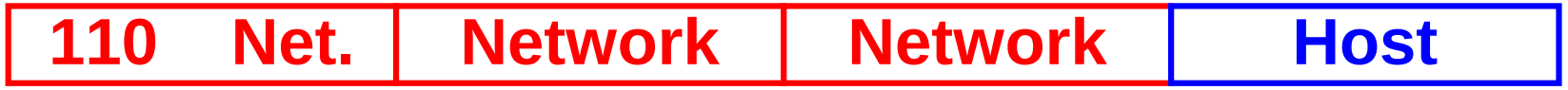
Classe A 1.H.H.H - 127.H.H.H



Classe B 128.N.H.H - 191.N.H.H



Classe C 192.N.N.H - 223.N.N.H



- **Adresses source et destination**

© GL

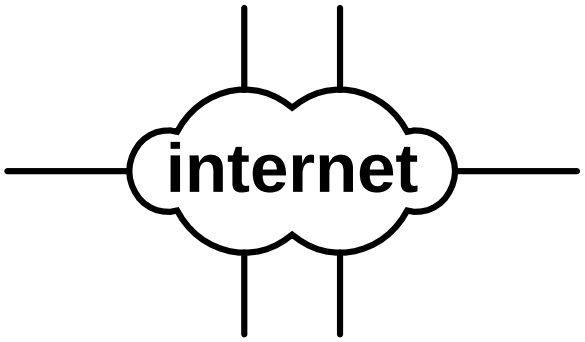
Adresse IP (2)

- **Adresse de classe A** — **1.H.H.H - 127.H.H.H**
  2^{24} ***hosts per network***
- **Adresse de classe B** — **128.N.H.H – 191.N.H.H**
  2^{16} ***hosts per network***
- **Adresse de classe C** — **192.N.N.H – 223.N.N.H**
  2^{8} ***hosts per network***
- **Adresse de classe D** — ***multicasting***

© GL

Adresse IP (3)

- **Adresse unique au niveau mondial**
  La partie network est attribuée par NIC *(Network Information Center)*

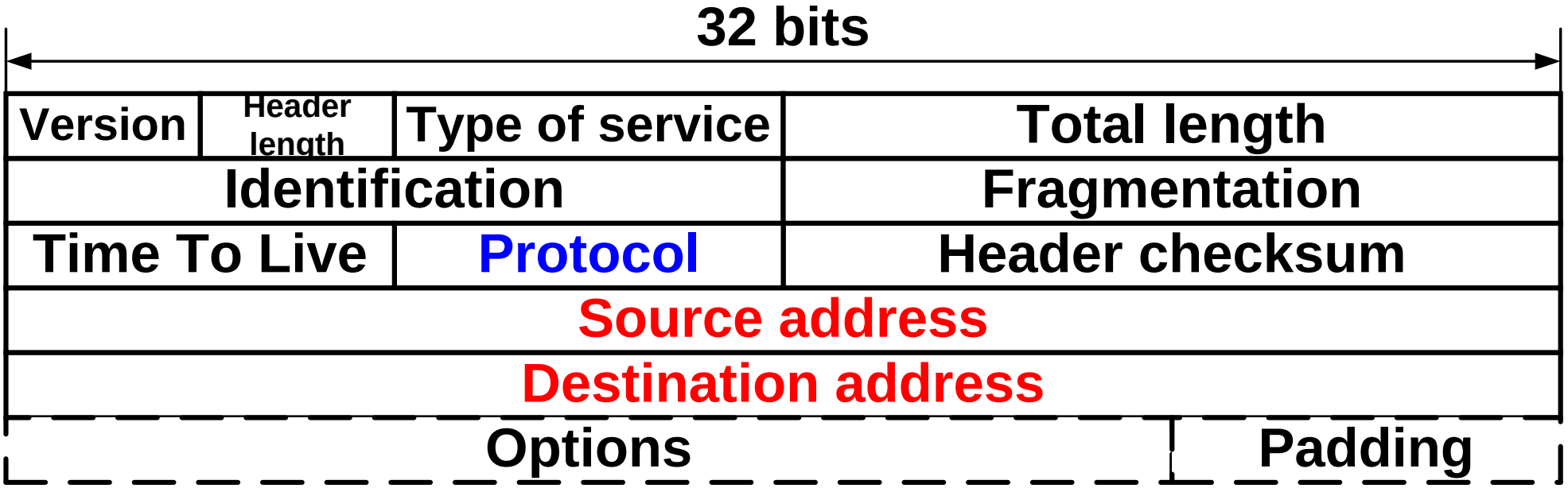


- **Administration manuelle – source de problème**
  Adresses incorrecte, dupliquée

- **DHCP *(Dynamic Host Configuration Protocol)***
  Lors du boot, le nœud demande une adresse IP au serveur DHCP

© GL

Champs IP (1)

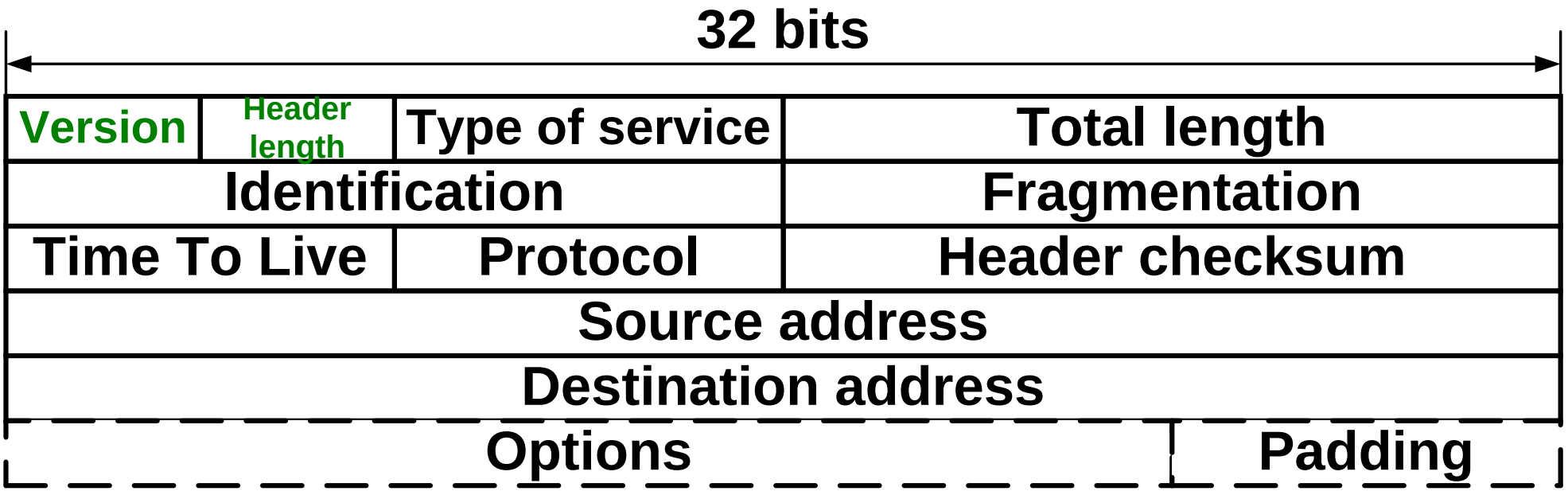


- Chaque datagramme IP contient l'adresse source (32 bits) et l'adresse destination (32 bits)

- Le champ *Protocol* identifie le protocole de couche supérieure (TCP : 6, UDP : 17, ICMP : 1, ...)

© GL

Champs IP (2)

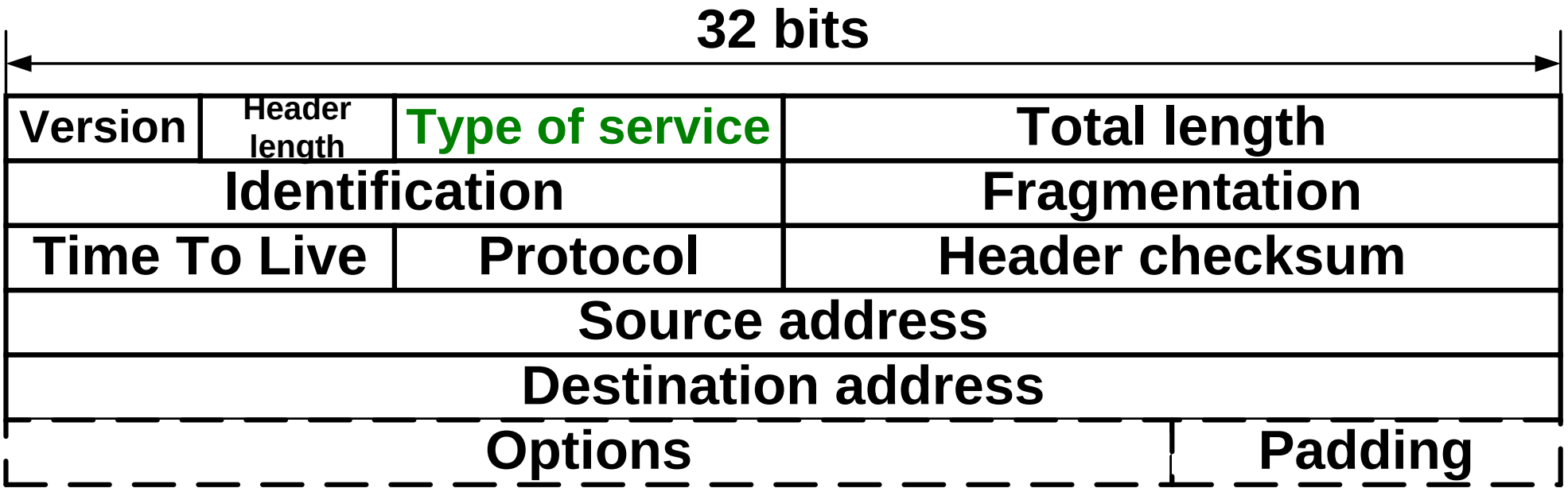


- ***Version*** **courante du protocole = version 4 (IPv4)**

- ***Header length*** **= nb de mots (32 bits) figurant dans l'en-tête.**
  Valeur habituelle = 20 octets signifiant l'absence d'option

© GL

Champs IP (3)

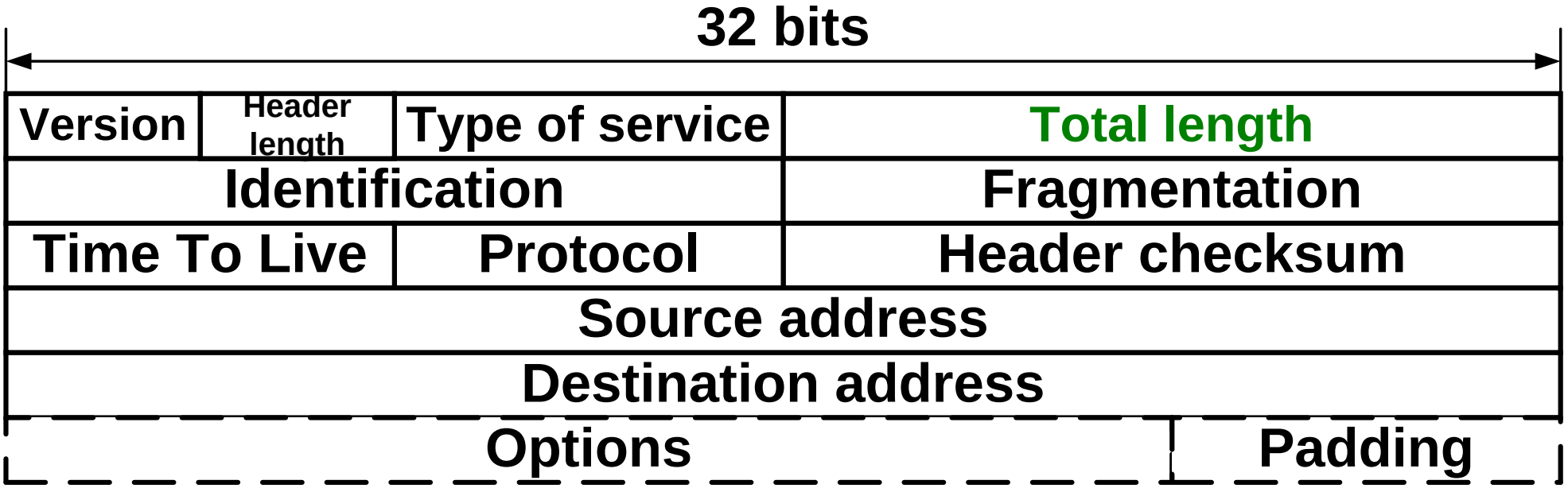


- ***Type of service*** **(4 bits) prévu (rfc 1340) pour minimiser le délai (telnet), maximaliser le débit (ftp), maximalise la fiabilité (smtp) et minimiser le coût monétaire (nntp)**

- **Voir *Quality of Service – Differentiated Services***

© GL

Champs IP (4)

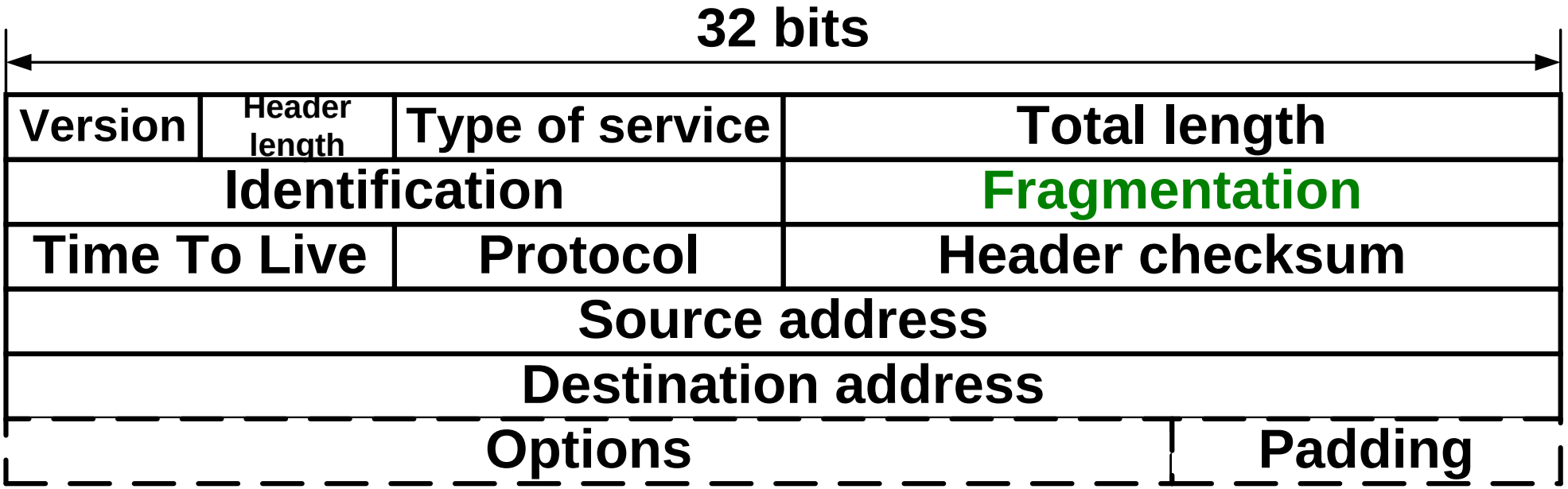


- *Total length* = taille totale du datagramme IP

- Champ de 16 bits → taille maximale d'un datagramme IP est donc de 65535 octets

© GL

Champs IP (5)

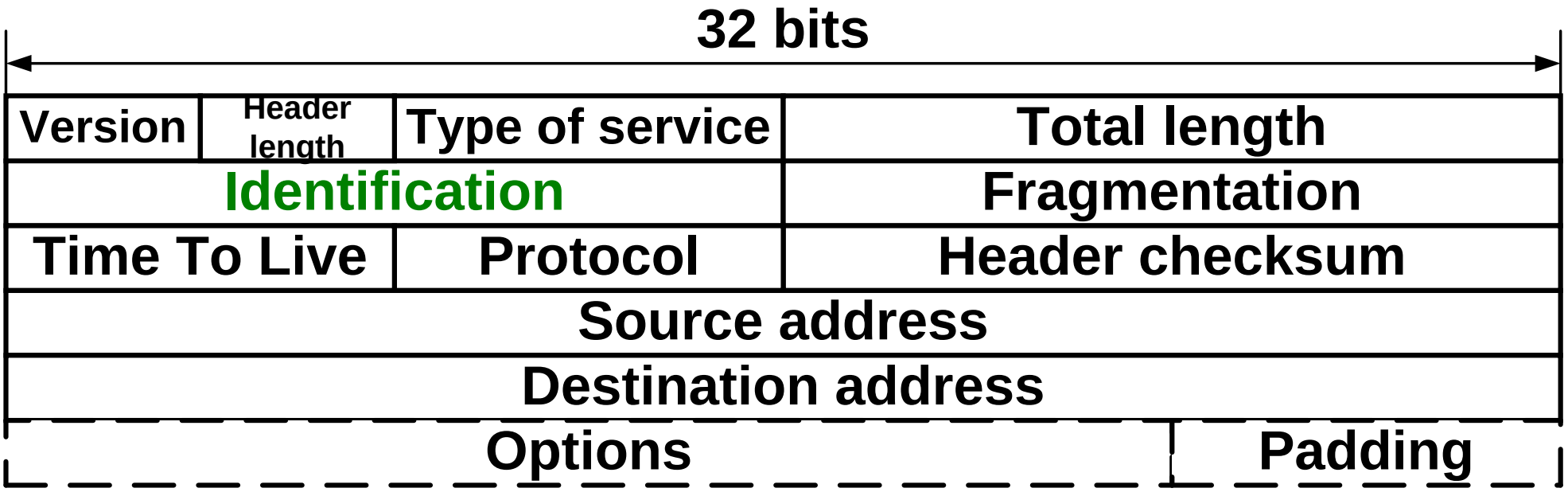


- Longueur max. = 1500 *(ethernet)* → ***Fragmentation***

- Bien que cette couche IP soit capable de fragmentation, la plupart des ordinateurs la pratique à un niveau supérieur

© GL

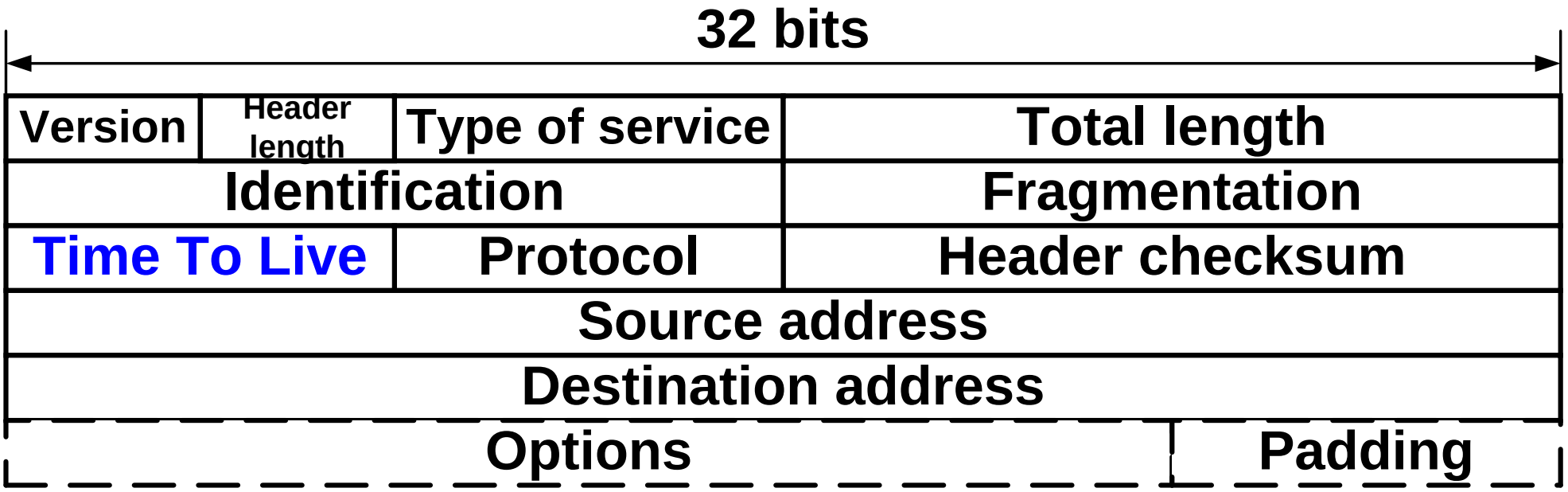
Champs IP (6)



- ***Identification*** **identifie de façon unique chaque datagramme envoyé par une machine qui l'incrémente simplement de un**

© GL

Champs IP (7)

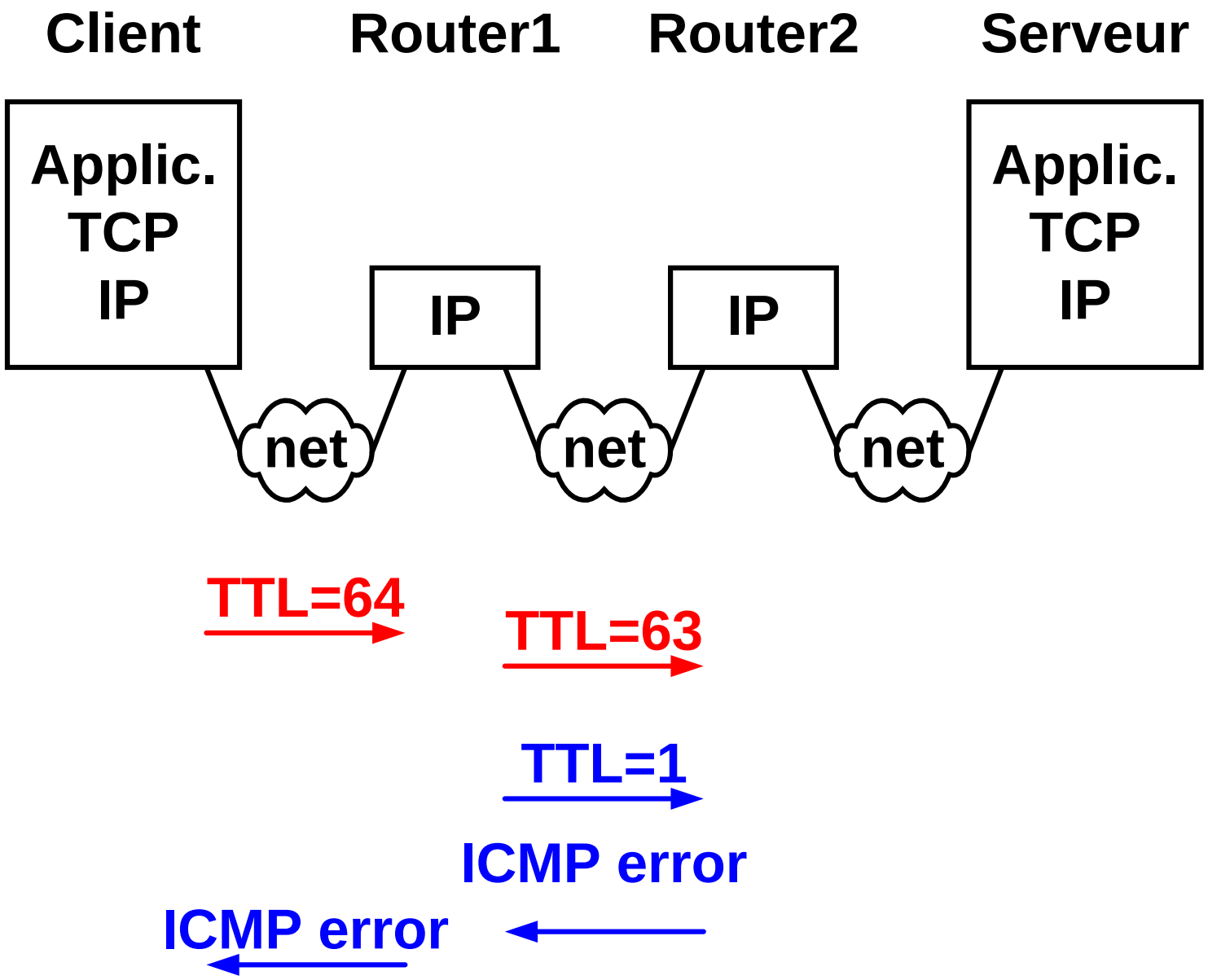


- *TTL (Time To Live)* donne une limite supérieure au nombre de routeurs qu'un datagramme peut traverser

- Limite ainsi la durée de vie du datagramme

© GL

Time To Live (TTL)



© GL

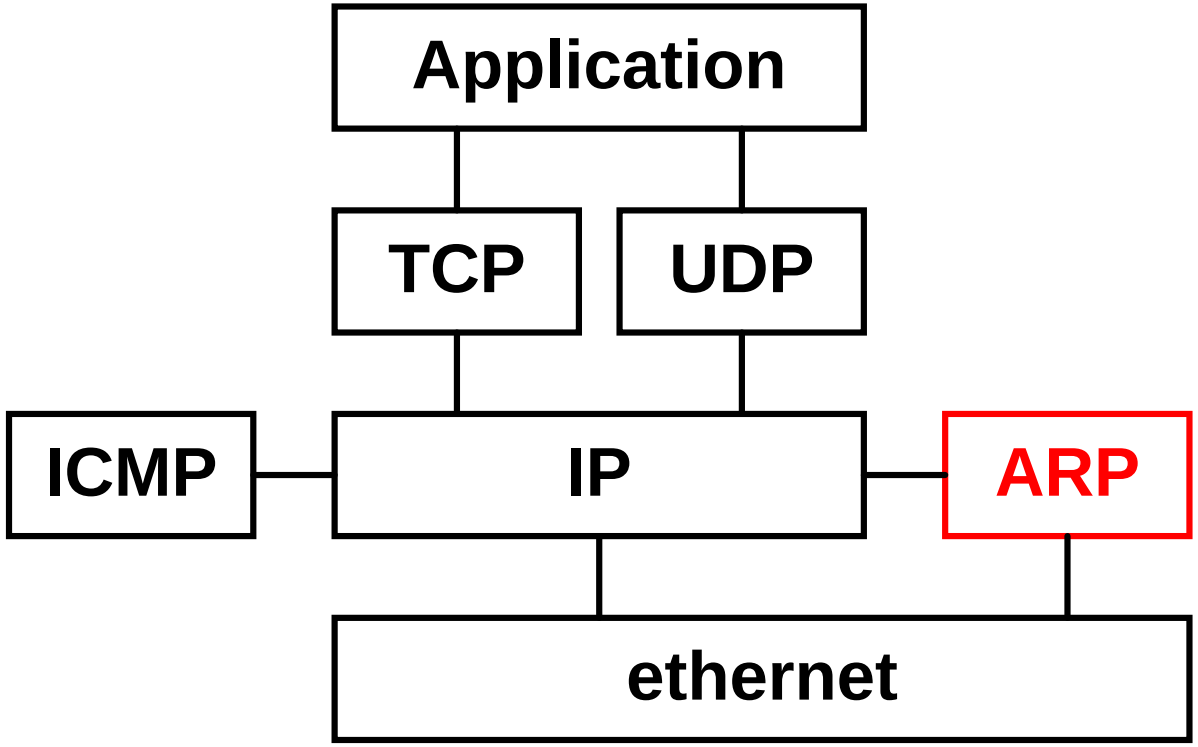
Analyse de protocole IP


```
IP:
   Version = 4, Header Length = 20 Bytes
   Type of Service = 0:
   Total IP length: 60
   ID: 0x0509
   Fragment = 0
   Time to live: 128
   PROTOCOL: 1 = ICMP
   Header checksum: 0xEFF1 (Good)
   IP Address Source = 10.1.2.1
   IP Address Destination = 129.194.184.2
```

© GL

Address Resolution Protocol (ARP)

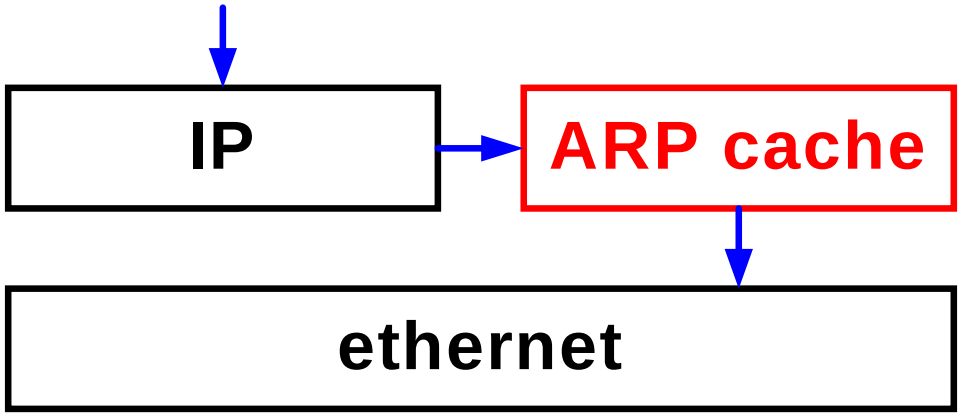
- **Le protocole ARP (*Address Resolution Protocol*) gère la relation entre adresse logique (IP) et adresse physique (*ethernet*) :**



© GL

Cache ARP

1 **Couche IP reçoit une requête d'une des interfaces supérieures**



2 **Adr. *ethernet* correspondante présente dans le cache ?**

Si non

Requête ARP ARP request

vers toutes les stations (diffusion)

Réponse de la station ARP response

Mise à jour du cache

3 **Envoyer la trame**

© GL

Analyse de protocoles Ethernet + ARP (1)


```
Ethernet:
  Destination Address:          FF:FF:FF:FF:FF:FF
  Source Address:               00:D0:59:A1:42:02
  Type:                         0x0806 ARP
Address Resolution Protocol
  ...
  Operation: 1 (ARP Request)
  Source Hardware address:          00:D0:59:A1:42:02
  Source IP address:                10.1.2.1
  Destination Hardware address:     00:00:00:00:00:00
  Destination IP address:           10.1.1.10
```

© GL

Analyse de protocoles Ethernet + ARP (2)

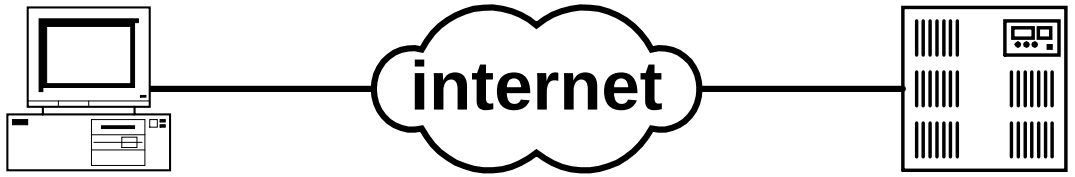

```
Ethernet:
  Destination Address:          00:D0:59:A1:42:02
  Source Address:               00:04:76:9C:7C:76
  Type:                         0x0806 ARP
Address Resolution Protocol
  ...
  Operation: 2 (ARP Response)
  Source Hardware address:        00:04:76:9C:7C:76
  Source IP address:              10.1.1.10
  Destination Hardware address:   00:D0:59:A1:42:02
  Destination IP address:         10.1.2.1
```

© GL

Ping (1)

- **Cette application permet de vérifier que la communication avec une autre machine est possible**

ping ftp.luth.se



ICMP echo request

ICMP echo reply

- ***Internet Control Message Protocol*** **(ICMP)**

© GL

Ping (2)

Commande

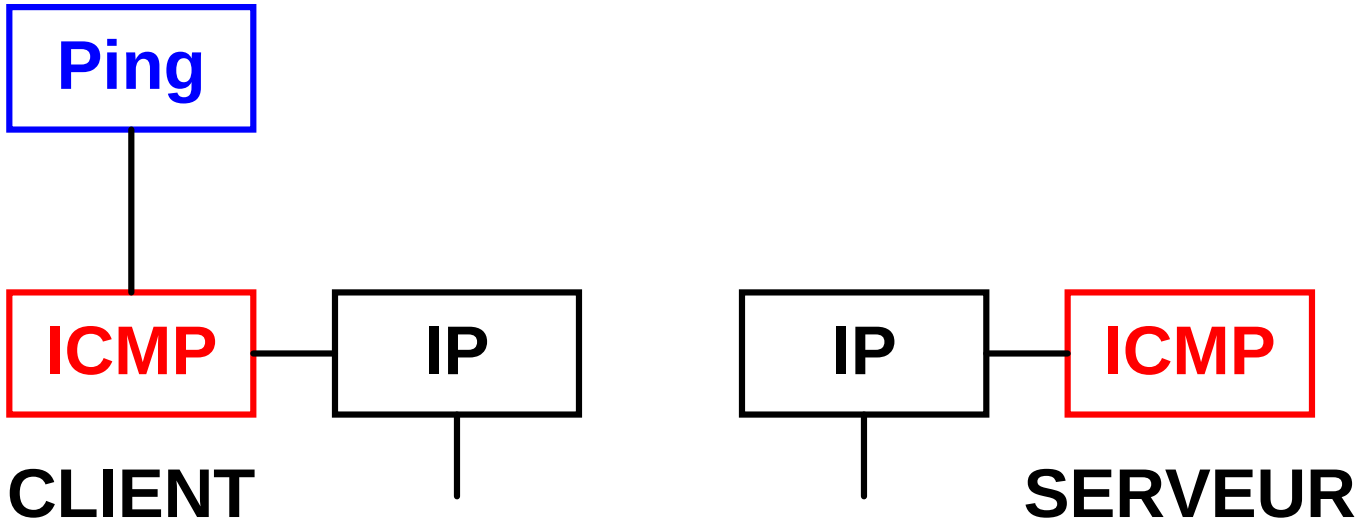
```
Ping Host 130.240.16.39 (ftp.luth.se)
Number      : 4
Timeout     : 15 [s]
Delay       : 1000 [ms]
Packet Size : 700 [byte]
```

Mesure du temps aller et retour (RTT : *Round-Trip Time*)

```
#    Address          Response Time
1    130.240.16.39    221 ms
2    130.240.16.39    147 ms
3    130.240.16.39    204 ms
4    130.240.16.39    112 ms
Statistics:    Out 4, in 4, loss 0%
times (min/avg/max) 112/171/221
```

© GL

Ping (3)



- **La plupart des implémentations de TCP/IP supporte le serveur ICMP directement dans le noyau**

© GL

Analyse de protocole ICMP


```
Internet Control Message Protocol:
  Type: 8 = Echo Request
  Checksum: 0x2F5C
  Identifier: 0x0200
  Sequence Number: 0x1C00
  Optional Data: 32 bytes

Internet Control Message Protocol:
  Type: 0 = Echo Reply
  Checksum: 0x375C
  Identifier: 0x0200
  Sequence Number: 0x1C00
  Optional Data: 32 bytes
```

© GL

Transmission Control Protocol (TCP)

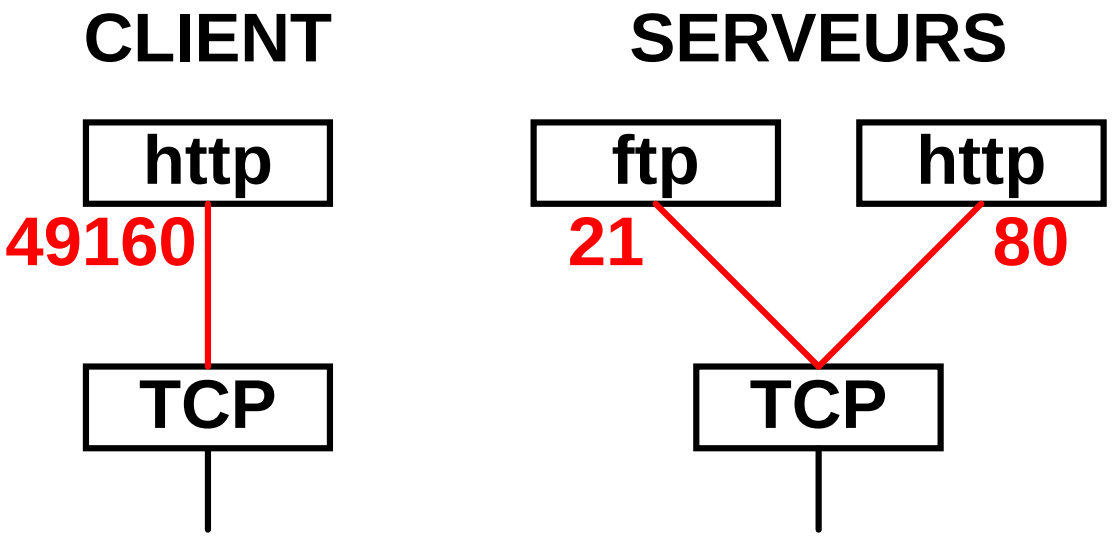
- **Le service offert par cette couche est orienté connexion**
- **Tout échange va donc commencer par une phase d'établissement**
- **Le flux de données sécurisé dispose donc de fonctions de contrôles d'erreur et de flux (numéro de séquence, accusé de réception,...)**
- **L'unité d'échange est appelé segment**

© GL

Ports TCP (1)

- **Ces champs identifient le protocole de couche supérieure**

  Source port + Destination port



- **La norme définit le *socket* = adresse IP + port**

© GL

Ports TCP (2)

- **Des valeurs normalisées *(Well Known Ports)* comprises entre 0 et 1023 donnent accès aux serveurs :**

  23 → telnet

  21 → ftp

  25 → smtp

  80 → http

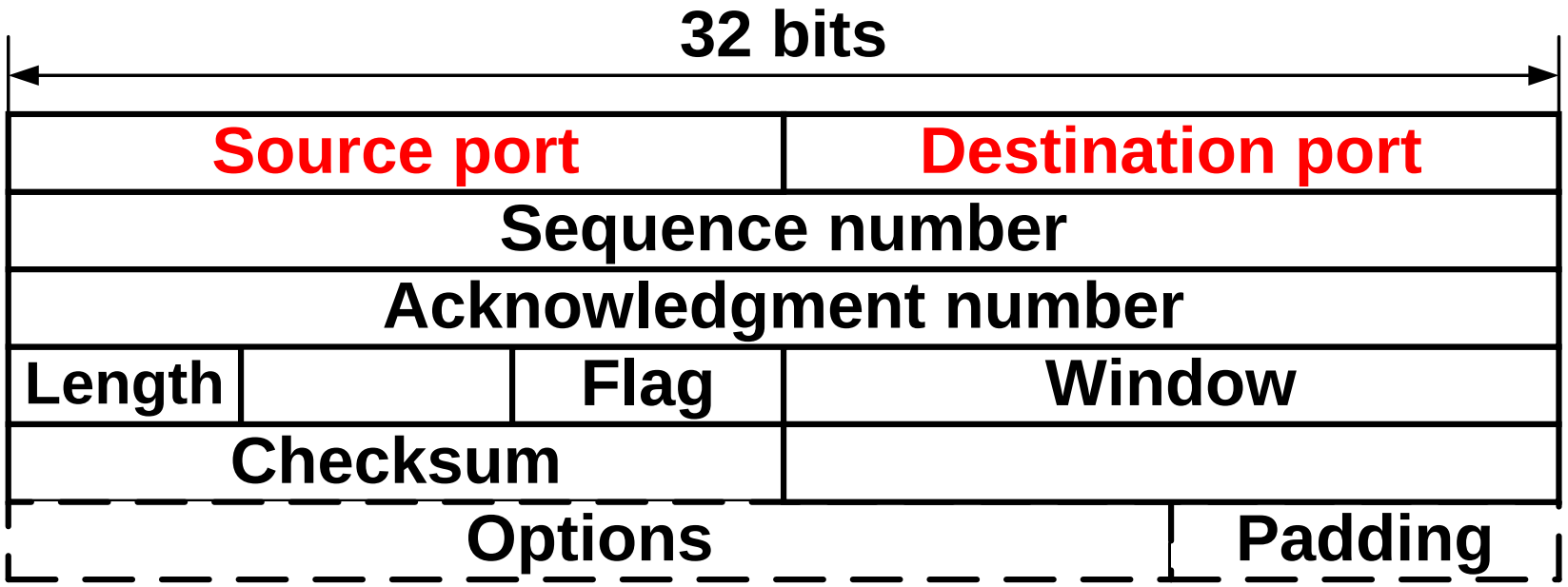
- **Le client attribue dynamiquement les numéros de port dans l'espace 49152 et 65535 *(Dynamic Ports)***

- **Autre espace défini : *Registered Ports* = 1024 .. 49151**

© GL

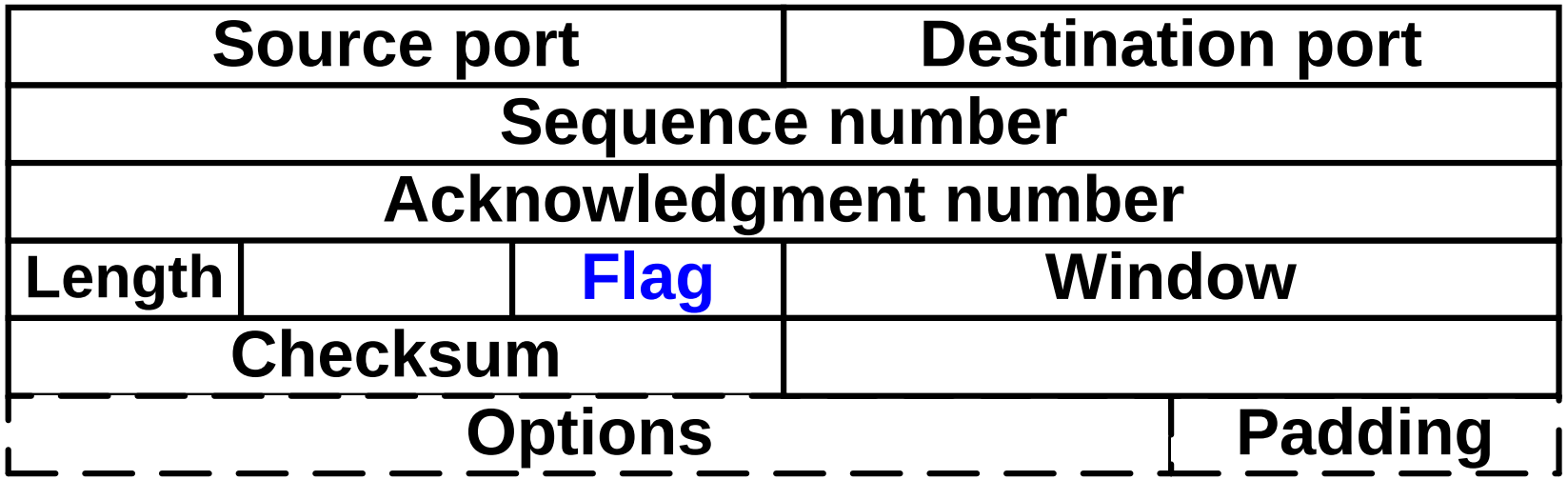
Champs TCP

- La longueur habituelle (sans option) est de 20 octets :



© GL

Champ Flag



- SYN synchronise les numéros de séquence lors de l'établissement
- FIN libération
- PSH (*push*) envoi des données utiles
- ACK accusé de réception
- RST *reset* de la connexion

© GL

TCP : Etablissement

- Ce protocole orienté connexion comporte 3 phases :

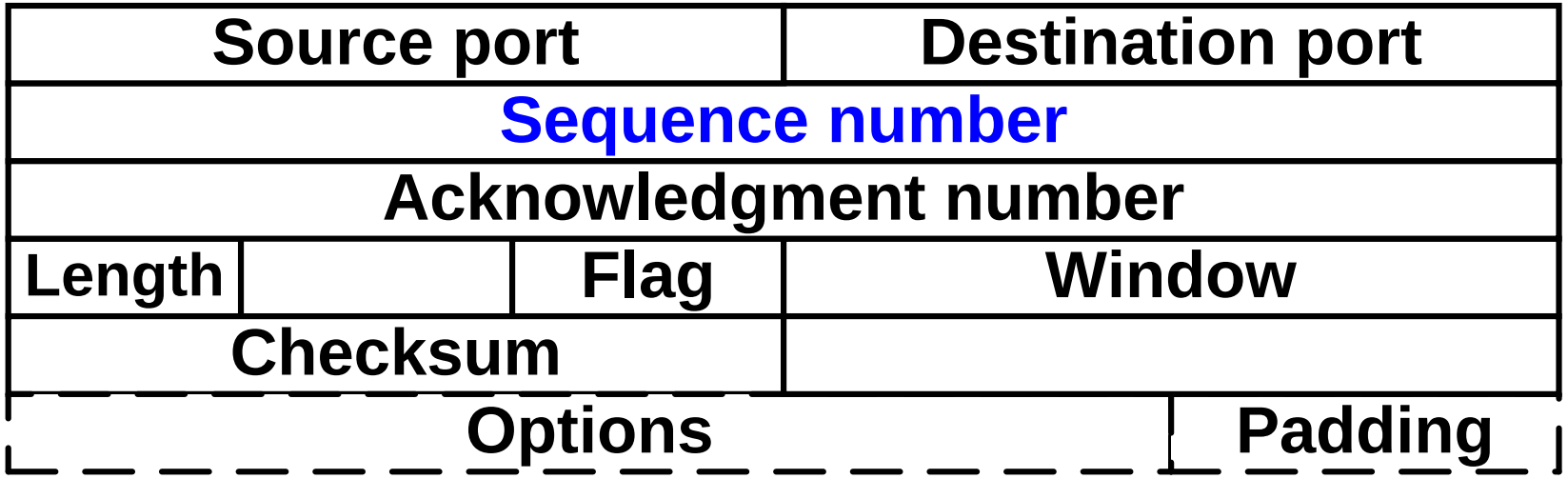
Client		Serveur
	→	SYN
	←	SYN, ACK
	→	ACK

- Etablissement symétrique

© GL

TCP : Transfert

- Il est orienté octet (*byte stream service*)
- Le numéro de séquence (*sequence number*) marque la position du premier octet du segment envoyé



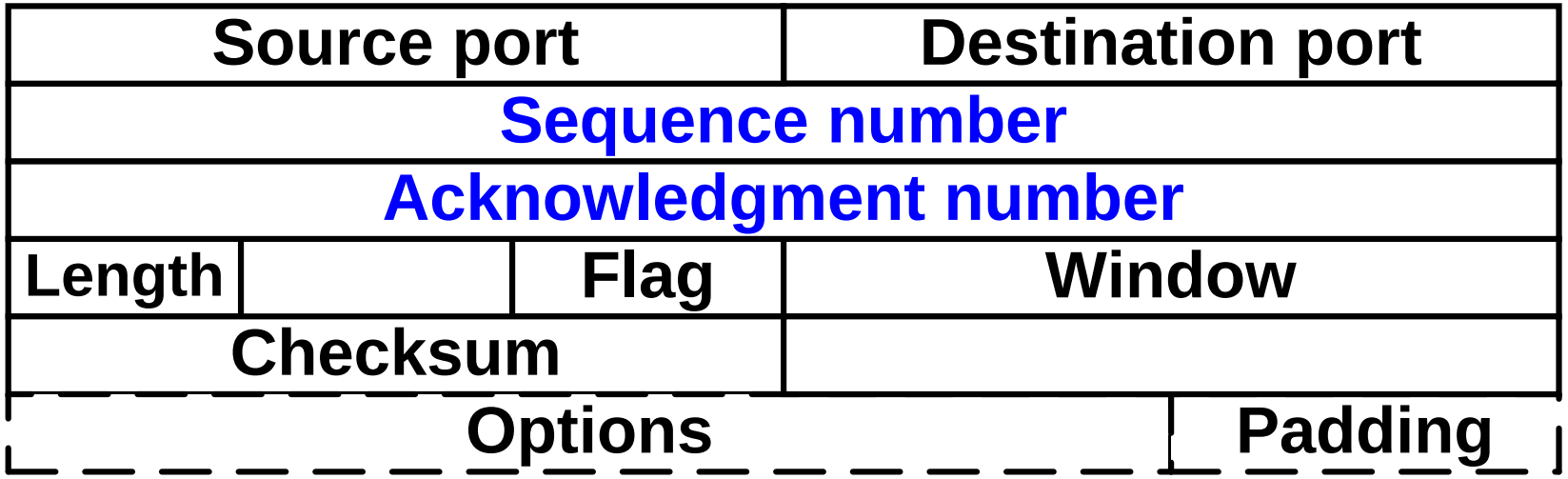
- Valeur "aléatoire" fixée à l'établissement

  clock-based sequence numbers (RFC793)

© GL

TCP : Contrôle d'erreur (1)

- Numéro de séquence à l'émission (Producteur)
- Accusé de réception retourné par le Consommateur



© GL

TCP : Contrôle d'erreur (2)

Producteur → Consommateur

1000 bytes → PSH seq = 1

← ACK ack = 1000

1000 bytes → PSH seq = 1001

1000 bytes → PSH seq = 2001

← ACK ack = 3000

← PSH Producteur

© GL

TCP : Contrôle d'erreur (3)

Producteur → Consommateur

Buffer vide

→ PSH seq = 1

1000 bytes

← ACK ack = 1000

Buffer vide

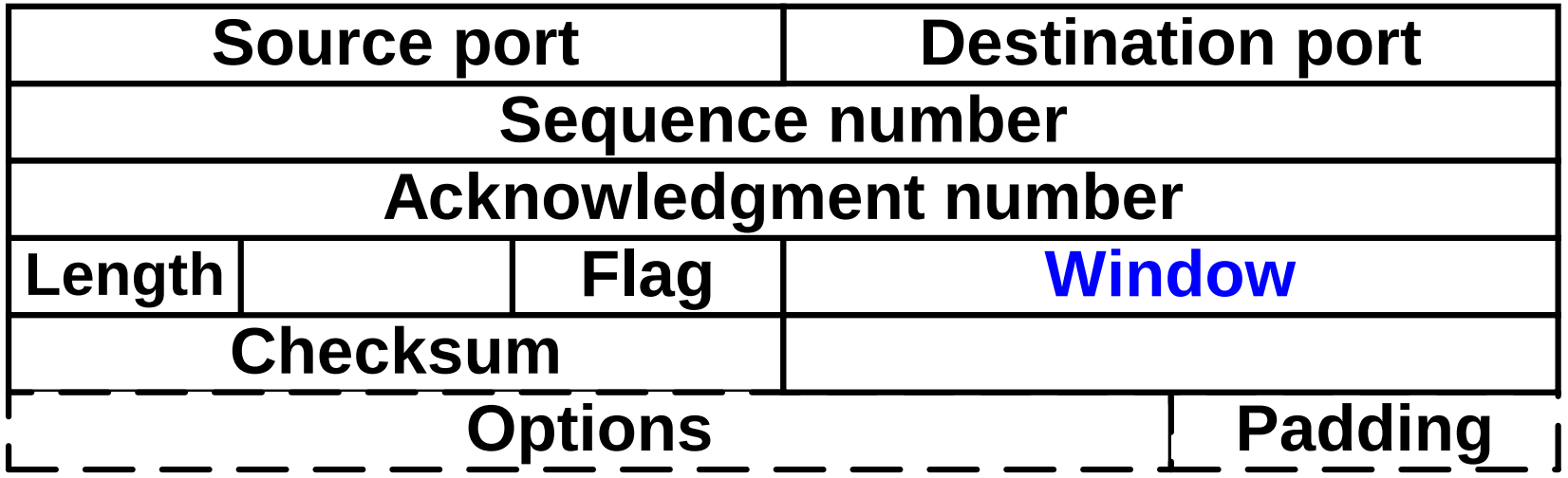
- **Le Producteur (TCP) doit conserver une copie dans l'attente d'une éventuelle retransmission**

- **Buffers de contrôle d'erreur situés du côté Producteur**

© GL

TCP : Contrôle de flux (1)

- **Champ *Window* indique le nombre d'octets que la station peut recevoir :**



© GL

TCP : Contrôle de flux (2)

Producteur → **Consommateur**

Buffer de 6000 bytes

← **ACK win = 6000**

1000 bytes → **PSH seq = 1001**

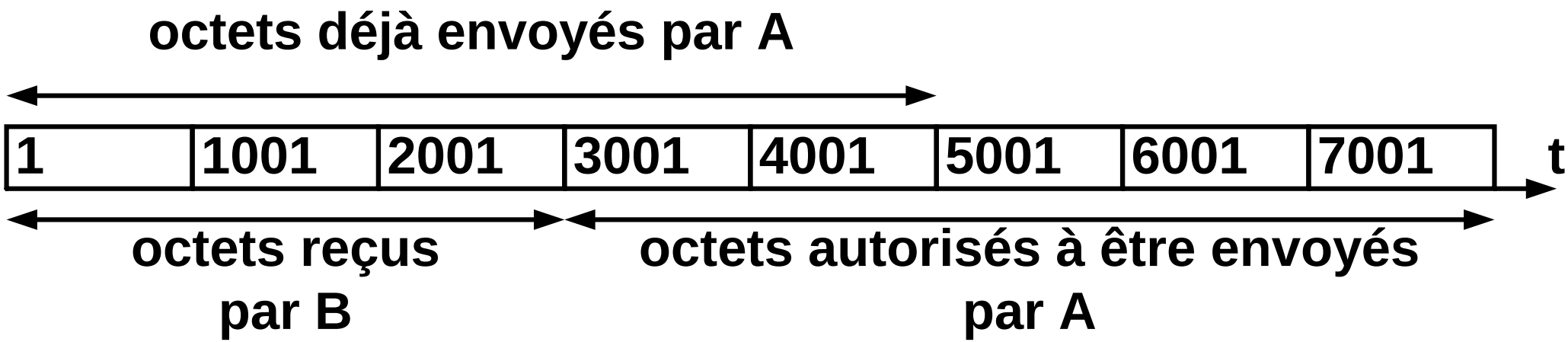
1000 bytes → **PSH seq = 2001**

- **Le Consommateur dispose d'un buffer de 6000 bytes**
- **Buffers de contrôle de flux situés du côté Consommateur**

© GL

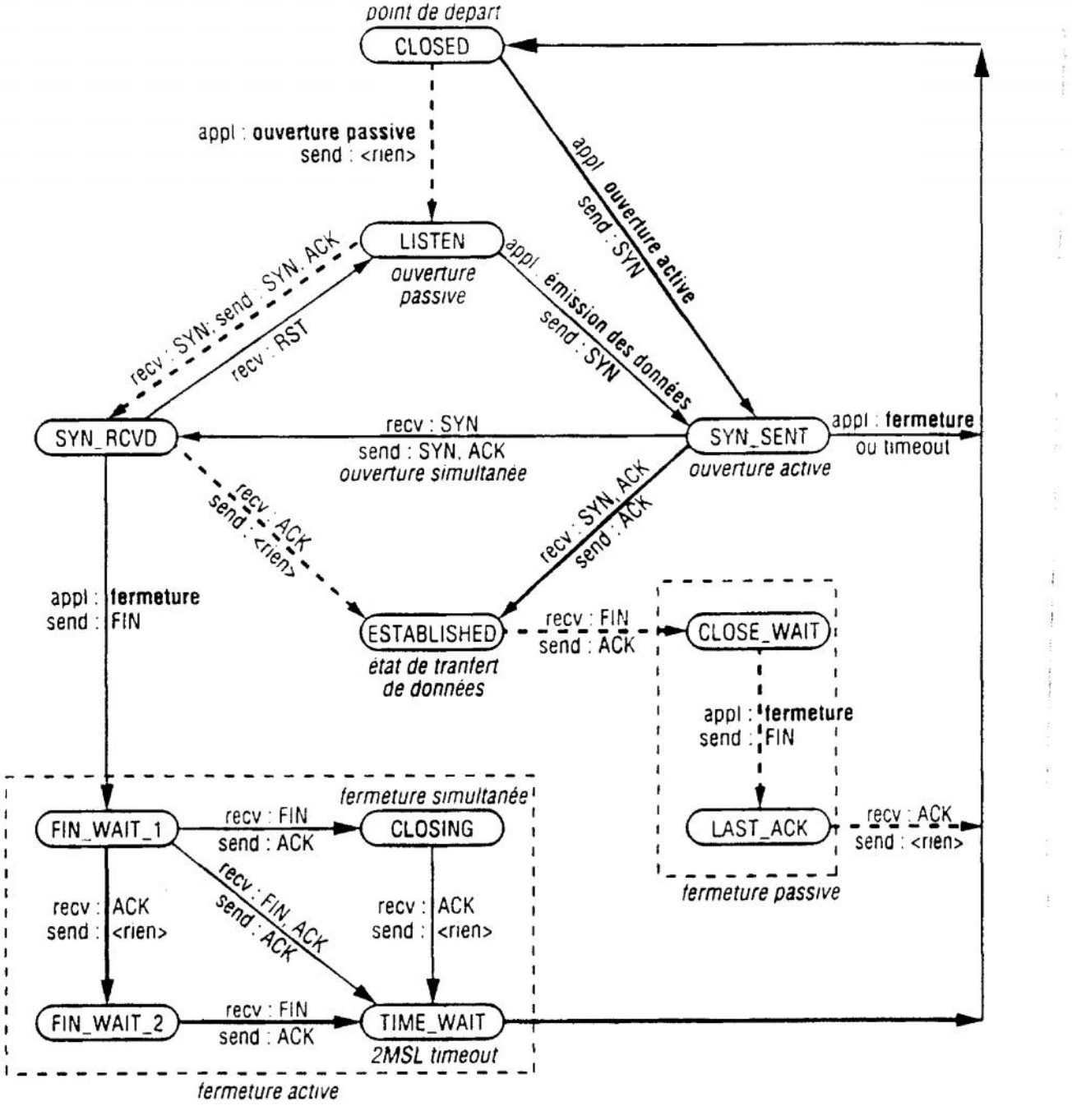
TCP : Contrôles d'erreur et de flux

Producteur	→	Consommateur
→ PSH seq=1	ack=1	win=2000
→ PSH seq=1001	ack=1	win=2000
→ PSH seq=2001	ack=1	win=2000
→ PSH seq=3001	ack=1	win=2000
← ACK	ack=3000	win=5000
→ PSH seq=4001	ack=1	win=2000



© GL

TCP : State Diag.



→ *indique les transitions normales pour le client*

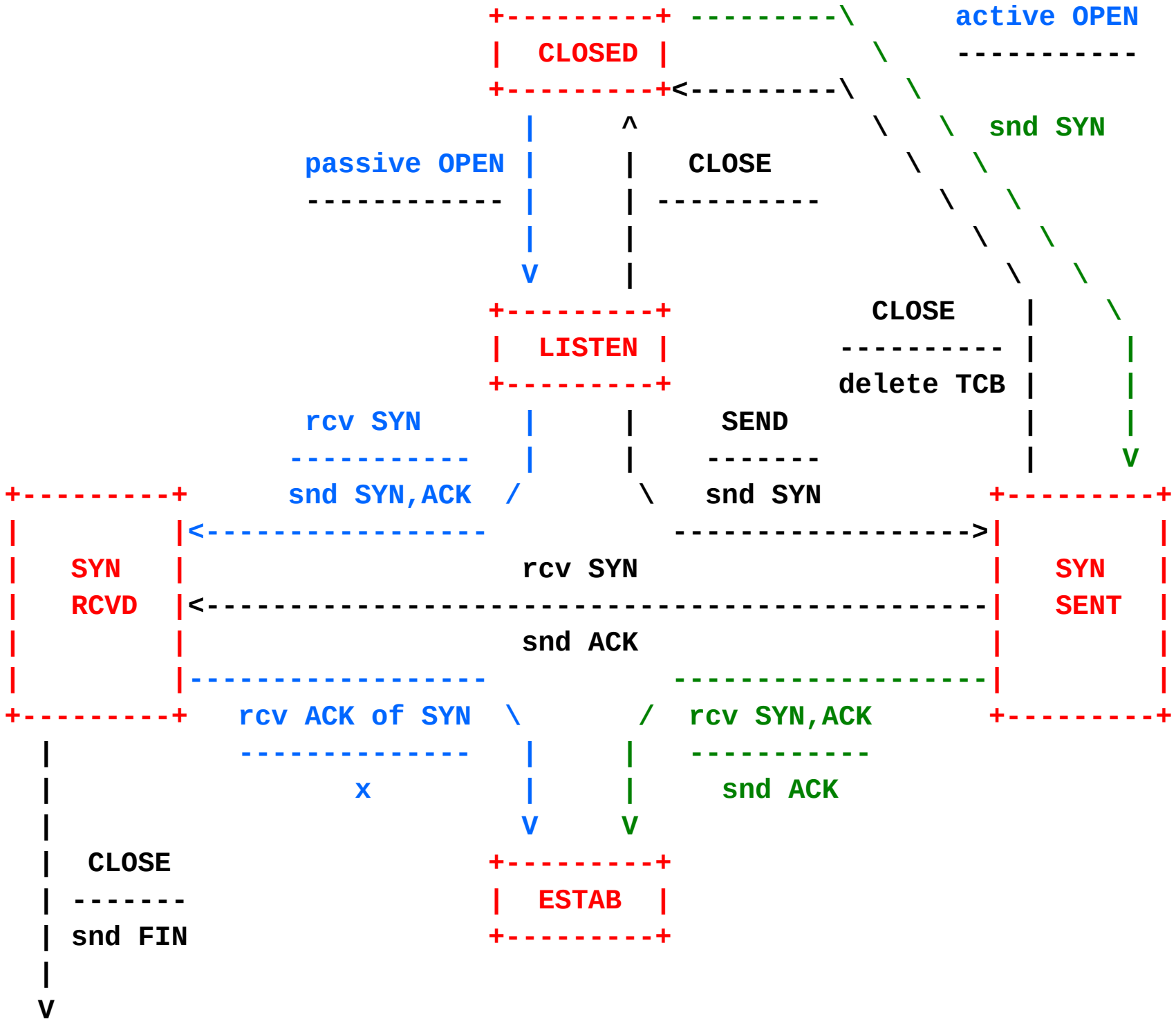
→ *indique les transitions normales pour le serveur*

appl : *indique les états de transitions pris lorsque l'application fait une opération*

recv : *indique les états de transitions pris lorsque le segment est reçu*

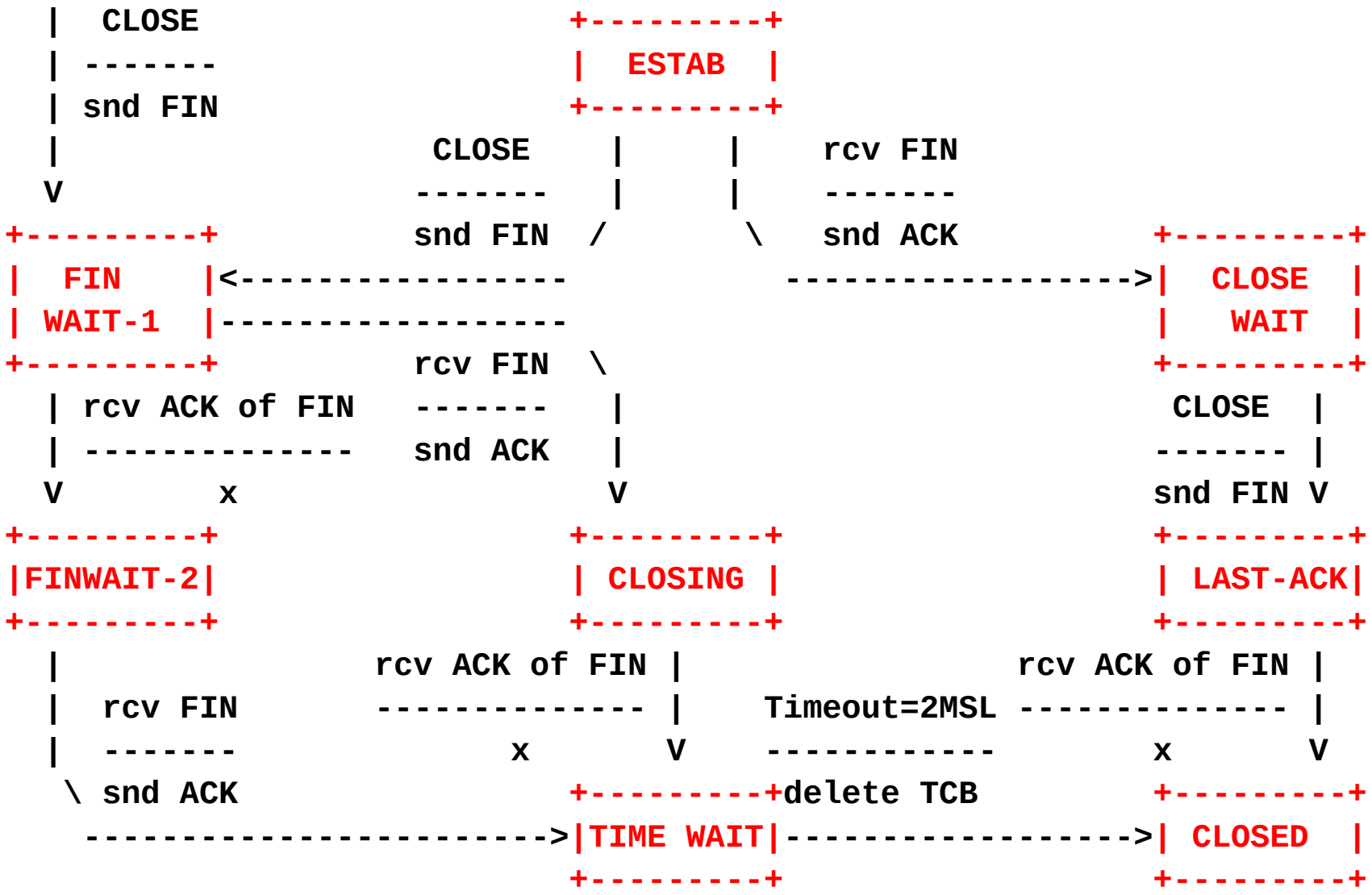
© GL

TCP : State Diag.



© GL

TCP : State Diag



© GL

TCP : Temporisateurs (1)

- **Ce protocole, chargé de sécuriser l'échange, présente une gestion complexe des temporisateurs : 4 temporisateurs différents étant nécessaires par connexion**

- **Le protocole TCP surveille la réception des accusés de réception grâce à son temporisateur de retransmission qui doit s'adapter au temps aller-retour de l'échange**

- **Un temporisateur persistant est activé par le producteur dont le flux est bloqué (contrôle de flux) par le récepteur opposé (fenêtre fermée)**

© GL

TCP : Temporisateurs (2)

- **Un temporisateur *keepalive* optionnel permet à une extrémité de sonder l'autre côté alors qu'aucune donnée ne circule depuis 2 h**

  En effet, aucune donnée ne circulera si la couche application reste muette. Certains estiment que l'application doit posséder ses propres temporisateurs; d'autres préfèrent utiliser ce temporisateur

- **Un temporisateur 2MSL mesure le temps pendant lequel une connexion a été dans l'état TIME_WAIT**

  Chaque implémentation doit choisir une durée de vie maximum du segment *(Maximum Segment Lifetime)*

© GL

Analyse de protocole TCP (1)


```
Transmission Control Protocol:
  Source Port :        1308                    Client
  Destination Port :   80                      Server web
  Sequence number:     611644602
  Acknowledgement:     3855596529
  Header Length:        20 bytes
  TCP flags:
     ........ ..0..... : Urgent pointer field not significant
     ........ ...1.... : Acknowledgment field significant
     ........ ....1... : Push function ON
     ........ .....0.. : Reset the connection OFF
     ........ ......0. : Synchronize sequence numbers OFF
     ........ .......0 : Expect more data from sender
  Window:              65535
  Checksum:            0xC20E (Good)
```

© GL

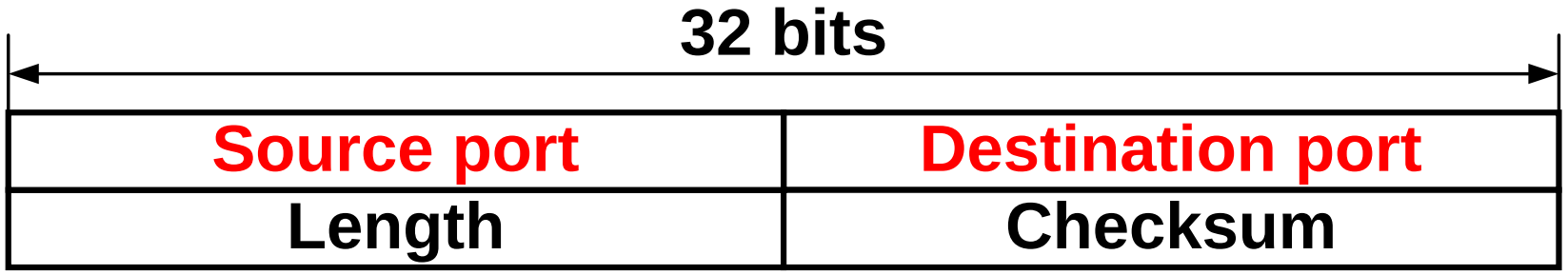
Analyse de protocole TCP (2)


```
Transmission Control Protocol:
  Source Port :          80                    Server web
  Destination Port :     1308                  Client
  Sequence number:       3855596529
  Acknowledgement:       611644953
  Header length:         20 bytes
  TCP flags:
     ........ ..0..... : Urgent pointer field not significant
     ........ ...1.... : Acknowledgment field significant
     ........ ....1... : Push function ON
     ........ .....0.. : Reset the connection OFF
     ........ ......0. : Synchronize sequence numbers OFF
     ........ .......0 : Expect more data from sender
  Window:                17520
  Checksum:              0x6E6B (Good)
```

© GL

User Datagram Protocol (UDP)

- **UDP est un protocole simple orienté datagramme**
- **Il fait ainsi partie des protocoles sans connexion**
- **A l'inverse de TCP, il n'offre aucune garantie de fiabilité**
- **Champs de l'en-tête UDP :**



- **UDP utilise les mêmes valeurs de ports que TCP**

© GL

Analyse de protocole UDP

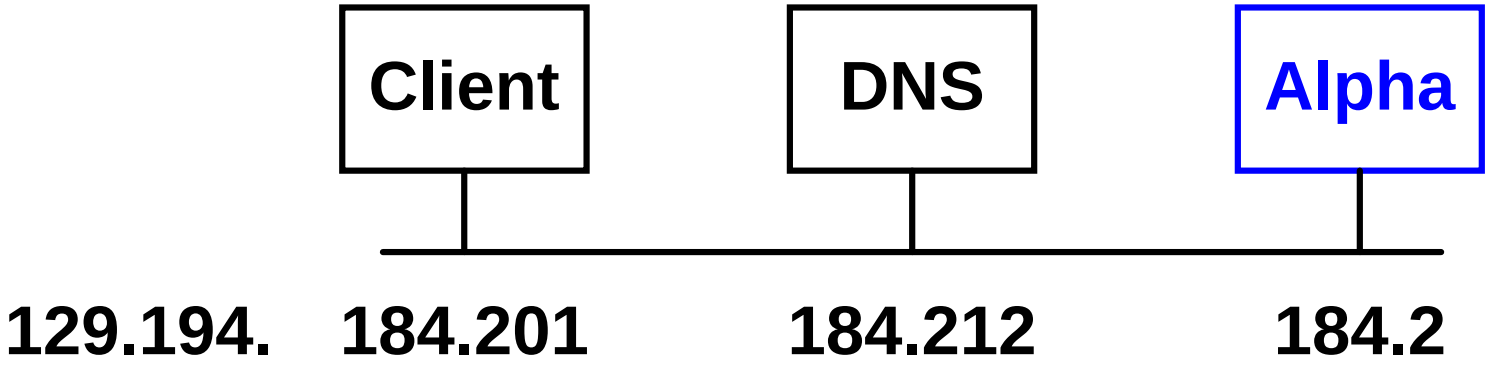

```
User Datagram Protocol:
  Source port:          1311
  Destination port:     53          DNS
  UDP length:           40
  Checksum:             0xEFD5 (Good)

User Datagram Protocol:
  Source port:          53          DNS
  Destination port:     1311
  UDP length:           115
  Checksum:             0xD12C (Good)
```

© GL

Domain Name System (DNS)

ping eig.unige.ch



DNS eig.unige.ch = ?

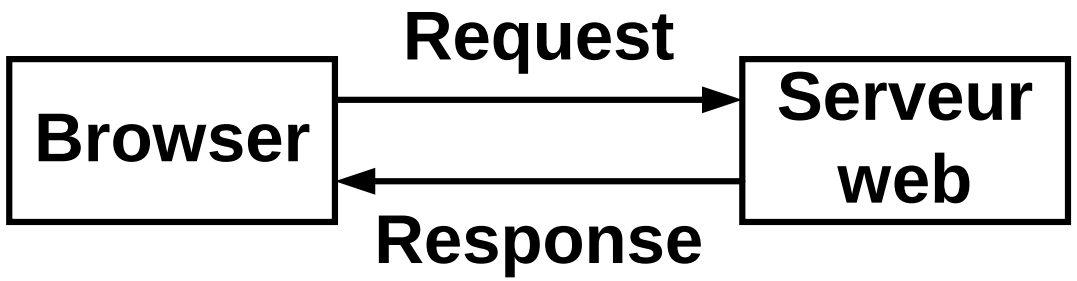
DNS eig.unige.ch = 129.194.184.2

© GL

Application web

- **Relation de type client – serveur**

- **Protocole HTTP *(HyperText Transfer Protocol)* utilisé entre navigateur *(browser)* et serveur *web* :**



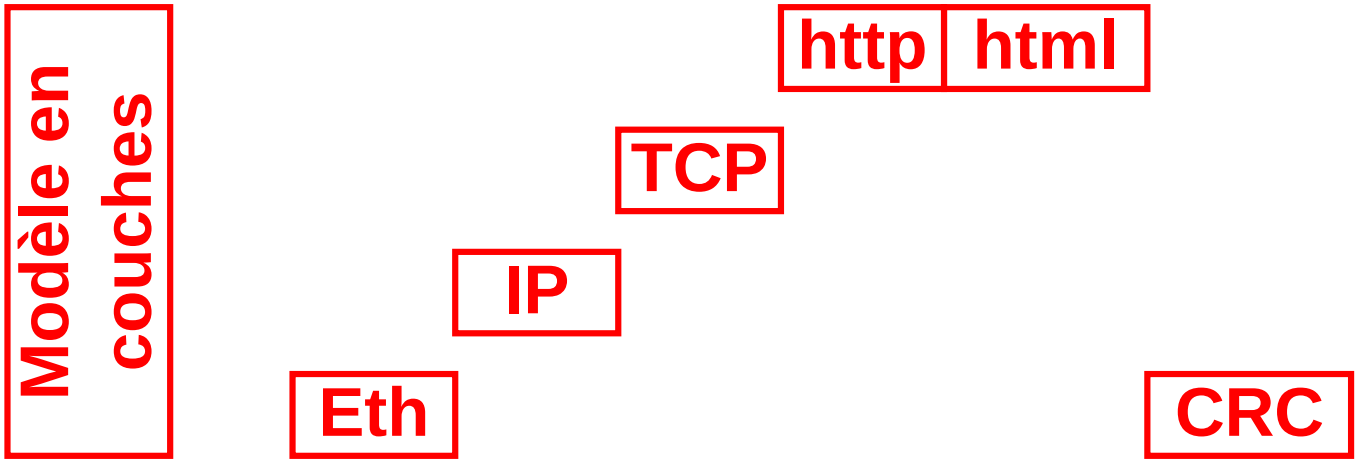
→ **Get http://www.td.unige.ch/default.html**

← **Response**

© GL

Protocol Stack

- **Modèle en couches avec empilement des protocoles *(protocol stack)***

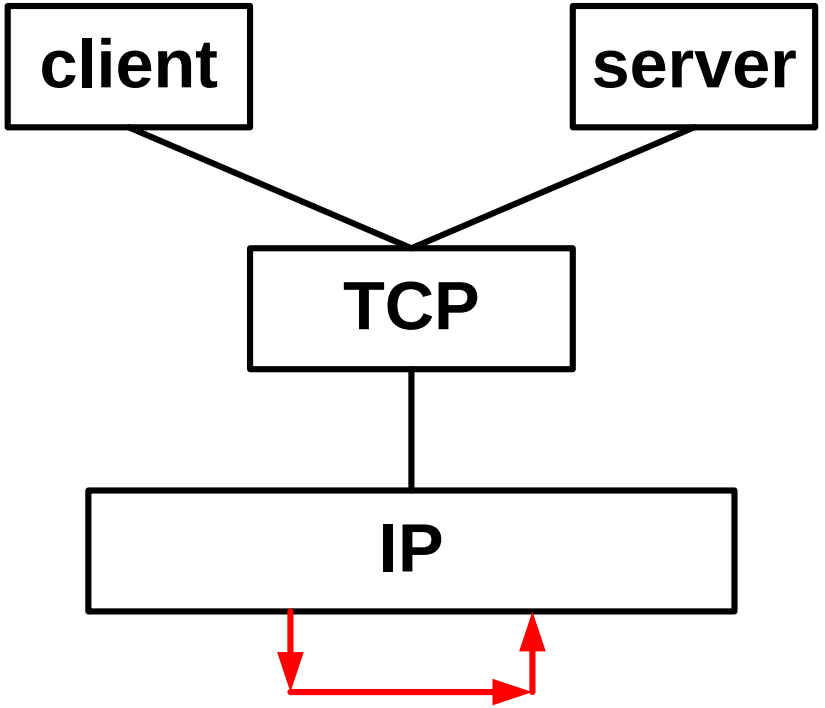


- **Format du paquet sur le réseau**

© GL

Loopback

- **Interface de bouclage permettant à client et serveur de communiquer**




```
C:\WINNT\system32\drivers\etc\hosts
127.0.0.1  localhost
```

© GL

Labo B1 : Analyse de protocoles

- **Commandes arp et ping**
- **Logiciel Observer en mode analyseur de protocole avec filtrage d'adresse**
- **Analyse de l'empilement ethernet – IP – TCP – http**
- **Complément théorique arp, icmp, IP et TCP**

© GL

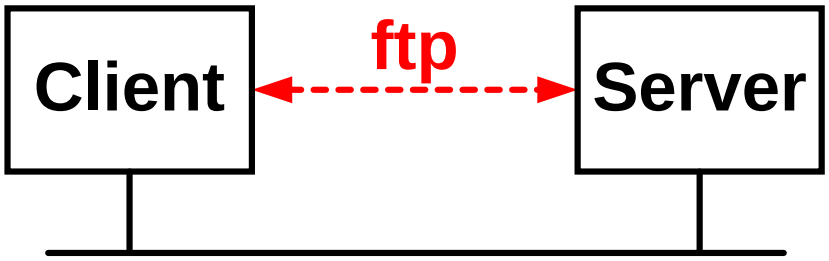
Labo B2 : Analyse du protocole FTP

- **File Transfer Protocol**

ftp.tdeig

10.1.1.2

CuteFTP (GUI)
ftp (CLI)
ftp depuis IE



Analyses de protocole
connexions TCP ?
ports utilisés ?
commandes ftp ?
réponses ftp ?
mode passif

© GL

Principales RFCs

ARP	826	DHCP	1531
DNS	1034, 1035	ICMP	792
FTP	959	HTTP	2616
IP	791, 919, 922, 950	SMTP	821, 822
TCP	793	TELNET	854, 855
UDP	768		
IP over ethernet	894		
Assigned numbers	1700		

http://www.switch.ch/ Internet Standards

sunsite.cnlab-switch.ch/cgi-bin/search/standard/nph-findstd

© GL

Livre & URLs

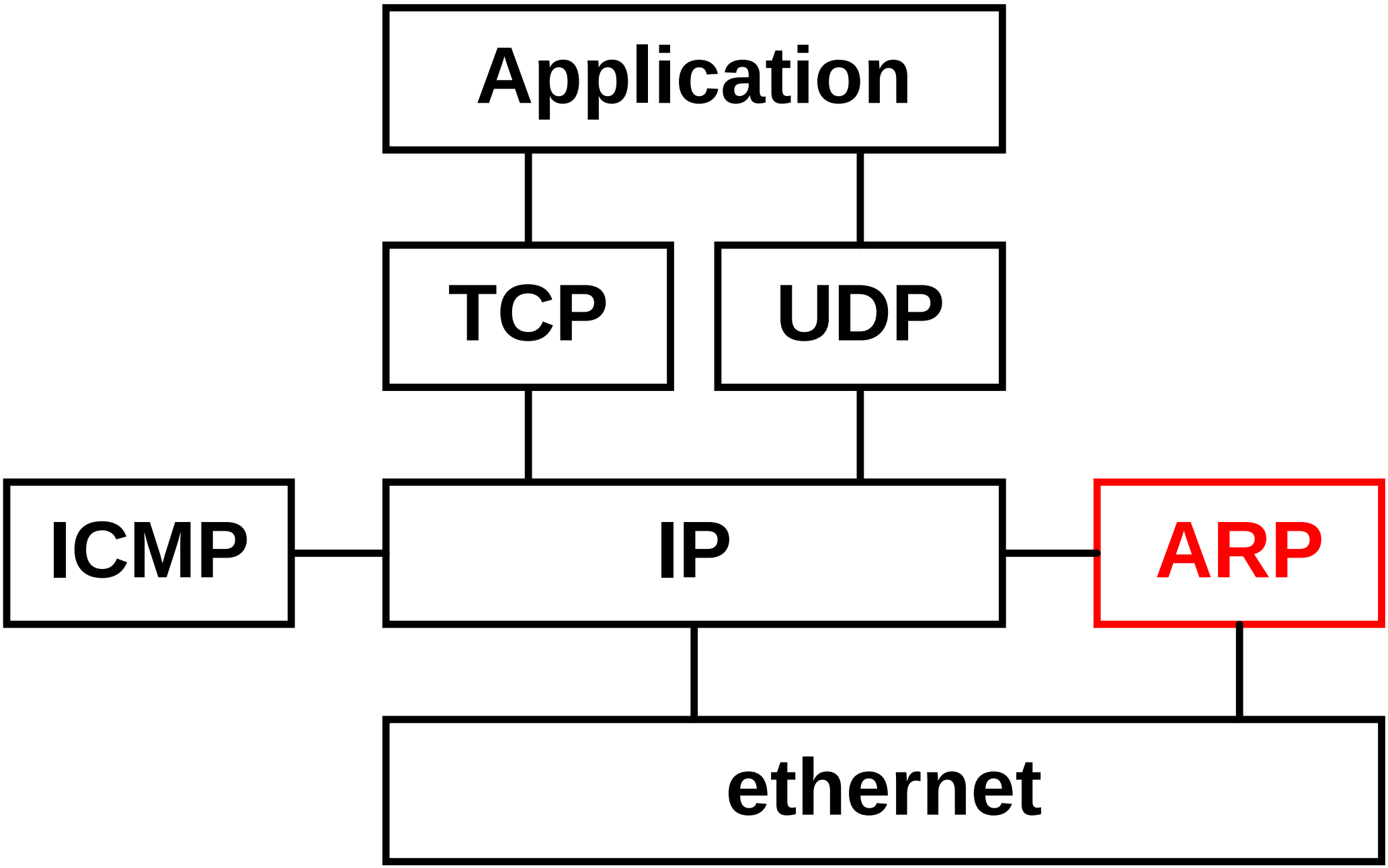
- **TCP/IP illustré – Les protocoles W. Richard Stevens**
  Versions anglaise et française

- **http://www.unige.ch/dinf/jfl/elem/etherhom.htm**

- **http://www.cs.rpi.edu/~hollingd/netprog2001/**

- **Analyseur ethereal http://www.ethereal.com**
  Observer

© GL



© GL