

Module ASR4 ASR40

Réseaux informatiques

Chapitre 4

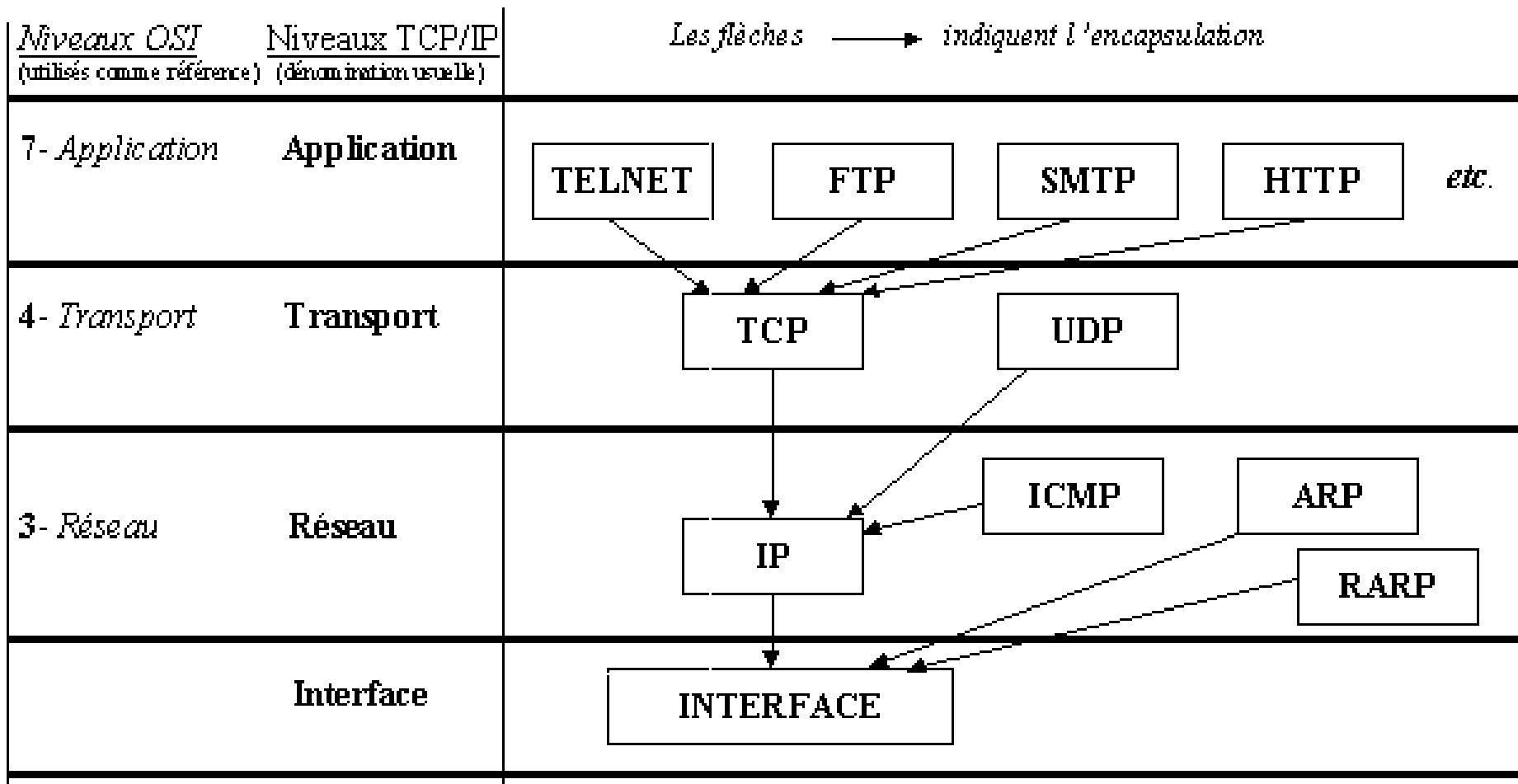
Protocoles UDP et TCP

Rappel architecture TCP/IP

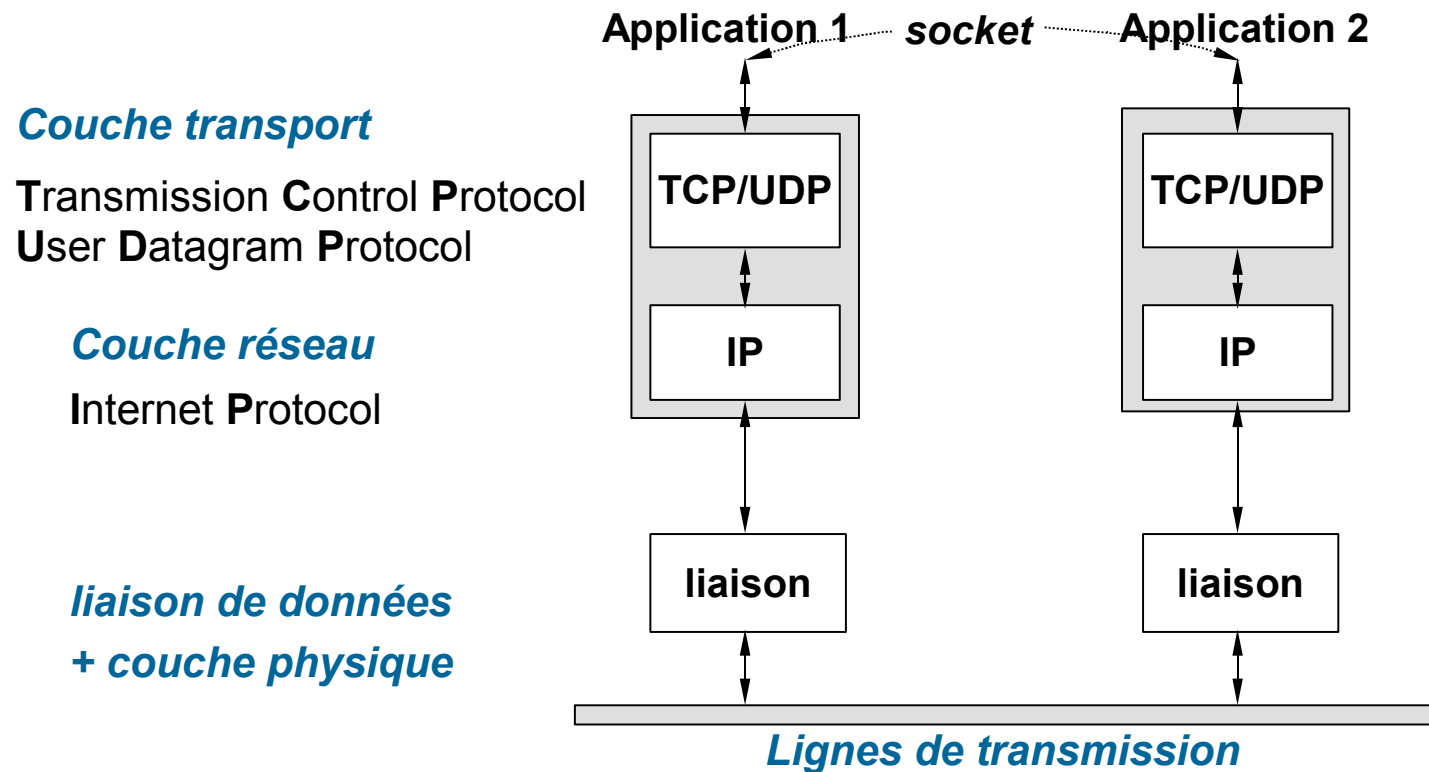
Niveau transport :

UDP : service de transport « simple » sans connexion

TCP : service de transport avec connexion (fiabilité des connexions)



Niveau transport dans l'architecture Internet



- Objectif : offrir un service **de bout en bout** entre les **processus** des applications, quelles que soient les caractéristiques du réseau

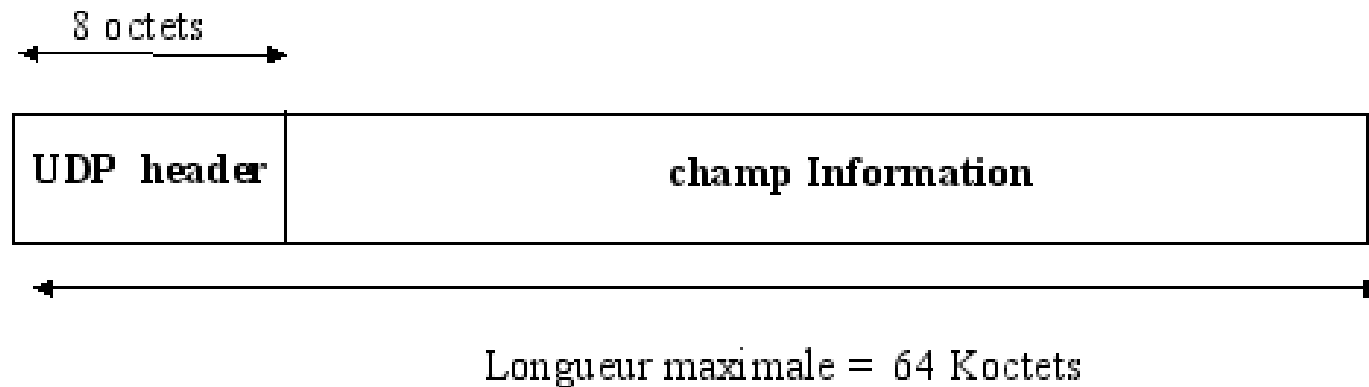
Fonctions du niveau transport

- **Transport des datagrammes ou des flots d'octets** entre un processus client et un processus serveur.
- **Nécessité d'identifier ces processus** lors de cet acheminement :
 - l'adresse IP ne suffit pas : plusieurs appli sur une même station
 - le numéro de processus système ne convient pas car le « PID » est local
 - > **numéro de port** : champ sur 16 bits de l'en-tête TCP ou UDP
- **Modèle client/serveur** :
 - toute communication est **initialisée par un client**.
 - le numéro de port et l'adresse IP du serveur doivent être connus du (des) client(s) avant tout communication.
 - les échanges de données sont ensuite **symétriques**.

Protocole UDP

- Principe : service de transport minimum et simple
- Services offerts :
 - Communication directe, dite « sans connexion ».
 - Encapsulation : dans un paquet IP (taille max = celle de IP).
 - Fragmentation/réassemblage : assuré pour les applications.
 - Multiplexage : possible par les n° de port.
 - **PAS** de contrôle d'erreur ni de flux.
- Exemples d'utilisation :
 - Protocoles « services » : NFS, DHCP
 - Applications : téléphonie, télémesure, jeux.

Format des datagrammes UDP



En-tête

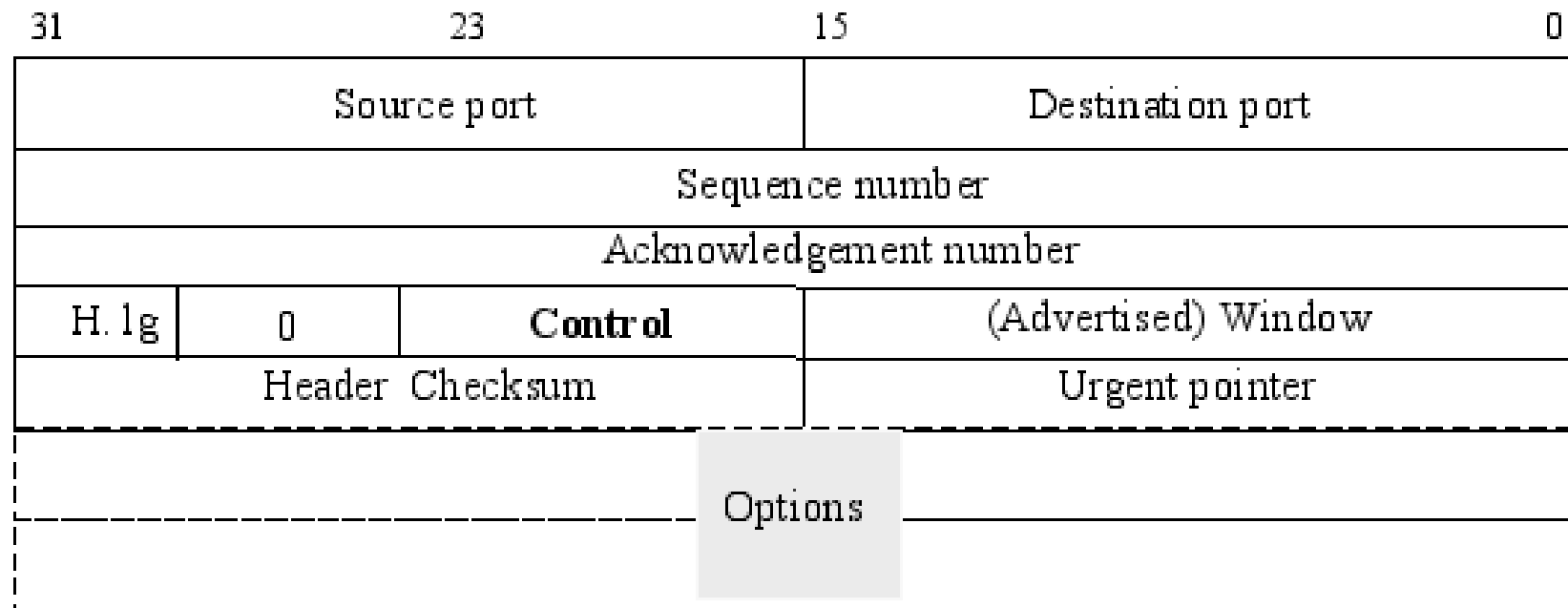
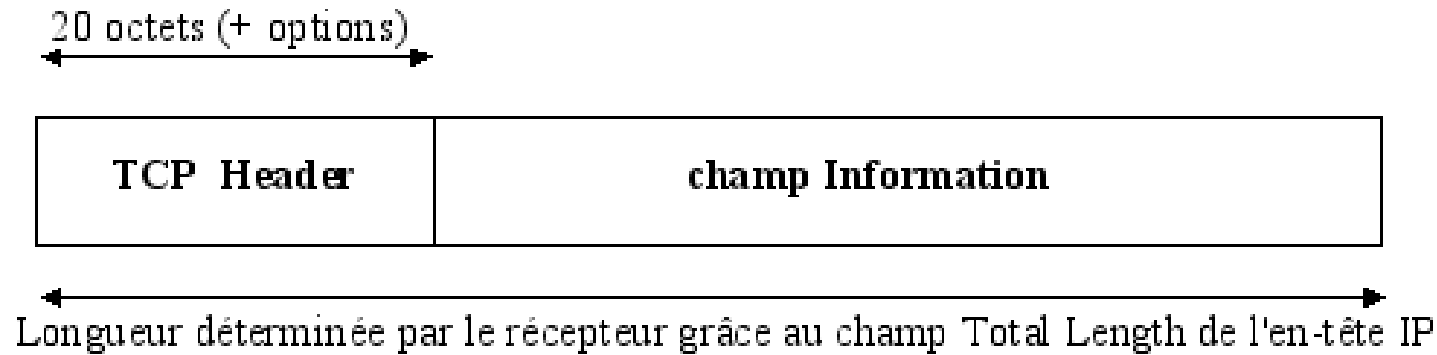
Remarque : le contrôle d'intégrité est facultatif et porte uniquement sur l'en-tête.

31	15	0
Source port	Destination port	
Total length (bytes)		Header Checksum

Protocole TCP

- Principes :
 - couche logicielle qui masque aux applications les contraintes du réseau (hors les délais induits!).
 - service de **transport fiable en mode flot d'octets**.
- Notion de **connexion TCP** associée au couple constitué du processus client et du processus serveur :
 - < **@IP serveur, N° port serveur, @IP client, N° port client** >
- Réalisation du **mode flot** :
 - Numérotation des octets à émettre.
 - Stockage dans des tampons.
 - Émission des octets indépendamment des dépôts.

Format des paquets (segments) TCP



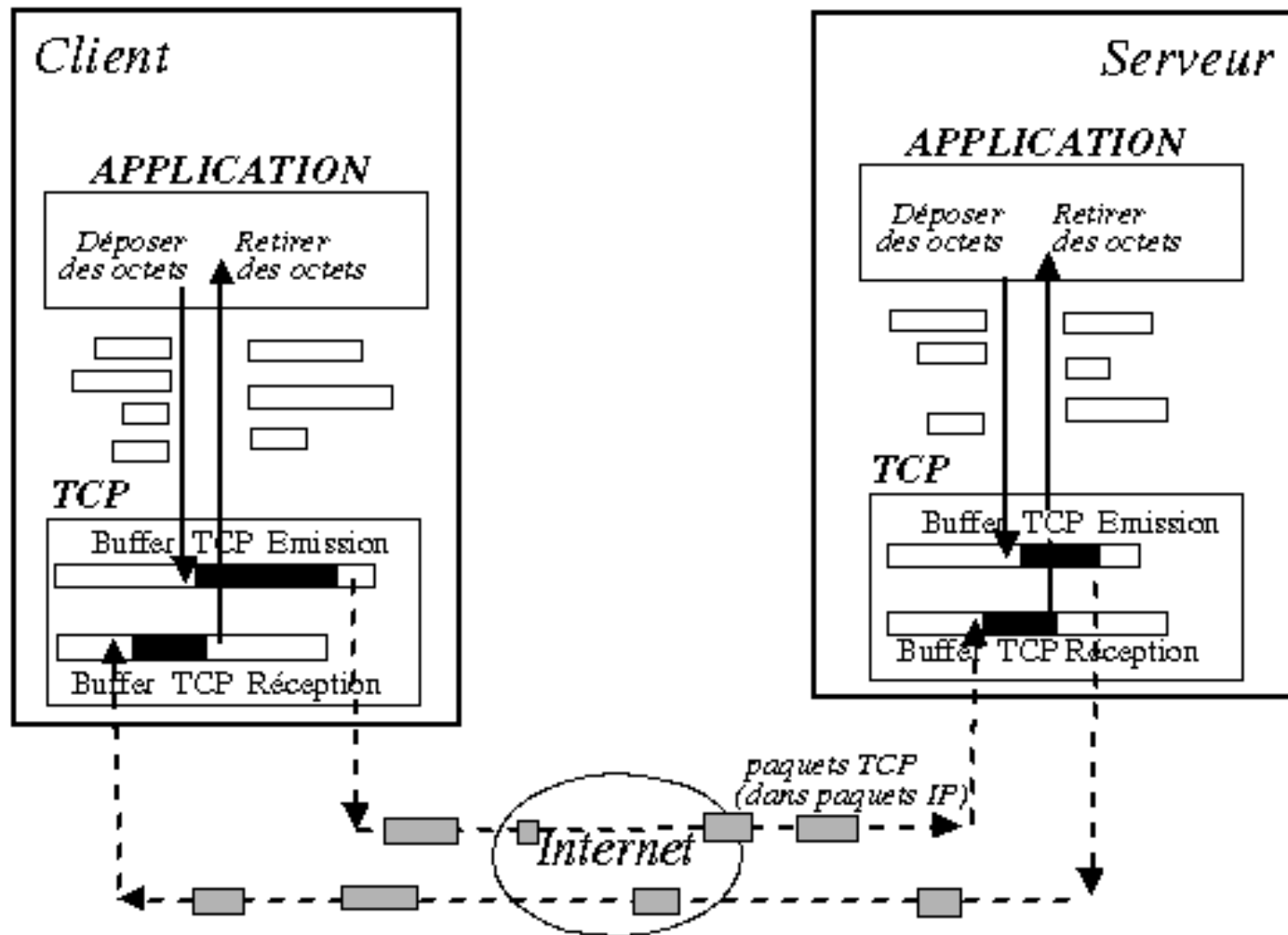
Champs de l'en-tête TCP

- **Sn : Sequence number** : n° du premier octet contenu dans le champ info (n° dans le flot sortant).
- **An : Acknowledge number** : n° du prochain octet attendu dans le flot entrant, acquitte les octets de n° inférieurs.
- **AW : Advertised Window** : fenêtre d'émission.
- **H.lg : Header Length** : longueur de l'en-tête, en mots de 32 bits.
- **Control : <0, 0, URG, ACK, PSH, RST, SYN, FIN>**
 - **SYN** : demande d'ouverture de connexion.
 - **FIN** : demande de fermeture de connexion.
 - **ACK** : le paquet acquitte des données ou une demande O/F.
 - **PSH** : les données peuvent être délivrées à l'application.

Fonctions du protocole TCP

- Service de transport en **mode connecté**.
- conservation de **l'ordre** des octets déposés par l'application.
- **Fragmentation** : découpage du flot en segments dont la taille est $\text{MIN}(N_a, \text{MSS}, W)$, $N_a : N_b$ d'octets « application » à émettre.
- **Contrôle d'erreur** : numérotation des octets et délai de garde T_{max} .
 - S_n : numéro du premier octet émis dans le segment.
 - A_n : numéro+1 du dernier octet acquitté.
- **Contrôle de flux** : envoi de W , la fenêtre d'émission.
- **Contrôle de congestion** : le délai T_{max} est augmenté s'il y a réémission, considérant que la cause est probablement due à une congestion dans les routeurs.

Tampons TCP



- Désynchronisation entre le dépôt et l'envoi d'une part et entre la réception et la délivrance des données d'autre part.
- Permet la mise en place des contrôles d'erreur et de flux.

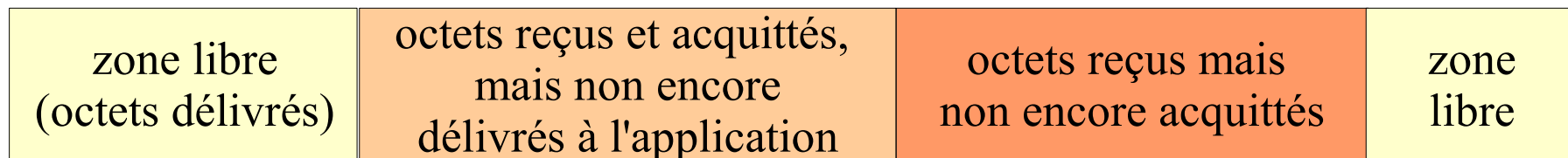
États des tampons émission et réception

Tampon émission



↑ Premier numéro
à émettre (S_n)

Tampon réception



↑ dernier numéro
à acquitter (A_{n-1})

Valeur de W à émettre = zone libre du tampon réception
= Longueur_tampon_reception – octets reçus et non délivrés

Champs de l'en-tête TCP pour le contrôle de flux

- **AW : Advertised Window**

- Permet de déterminer la fenêtre d'émission W :

$$W = AW \text{ ou } W = AW * 2^{WS}, \text{ avec } WS = \text{Window Scale (option)}$$

- **W indique le nombre d'octets que le récepteur est prêt à recevoir (et donc que l'émetteur peut envoyer).**
- Permet de réaliser le **contrôle de flux** puisque les tampons réception ne sont jamais en débordement.
- Avec AW et WS , les processus TCP peuvent ajuster la fenêtre d'émission pour **optimiser la connexion**.

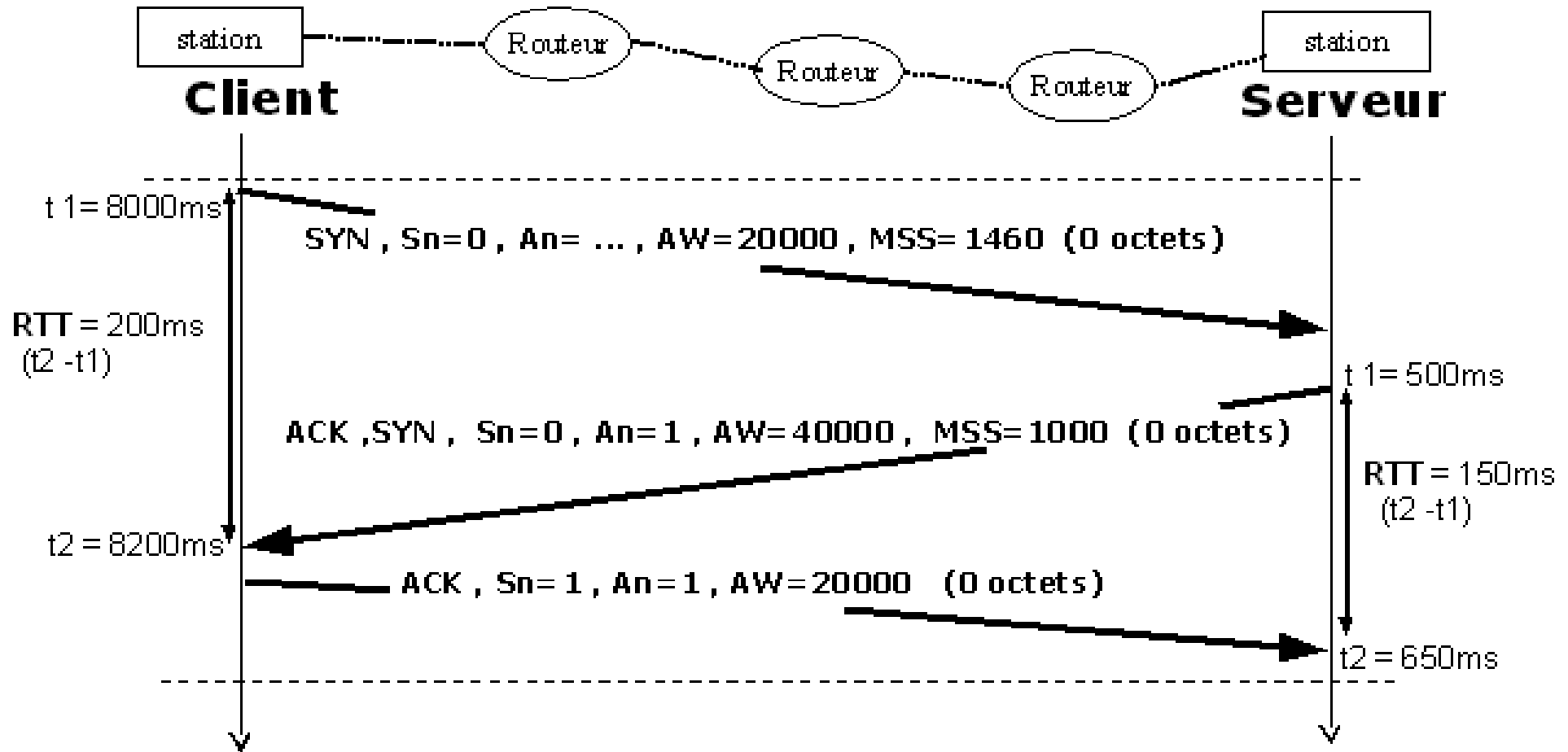
- **WS : Window Scale**

- Cette option TCP permet de définir des valeurs de fenêtre d'émission W supérieures à la valeur max de AW ($=2^{16}-1$).

Options TCP

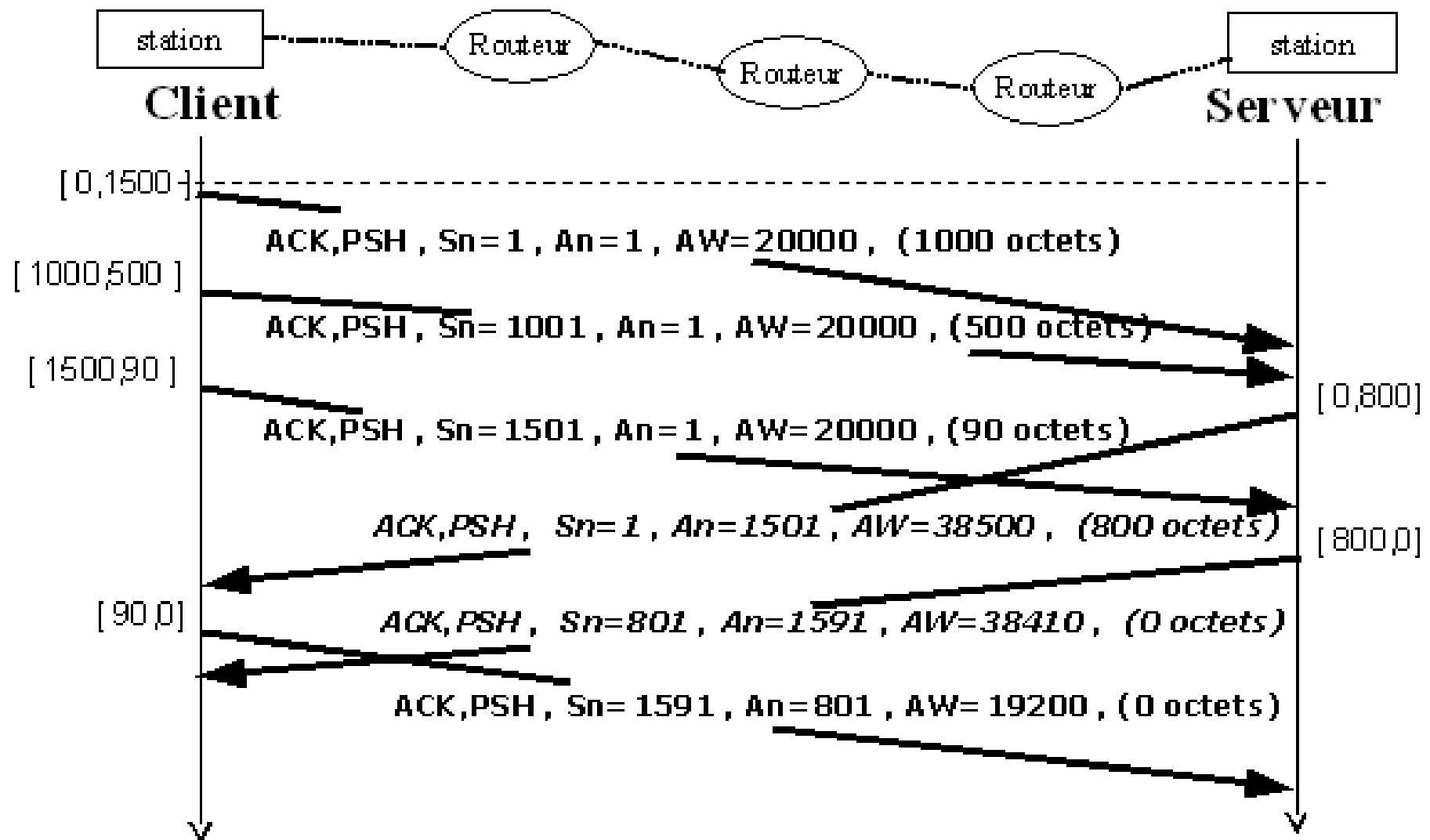
- Aux champs fixes de l'en-tête TCP, on peut ajouter des options pour :
 - **Le contrôle de flux** : WS.
 - **La fragmentation** : MSS (Maximum Segment Size) établie à la valeur du champ information de la trame moins les en-têtes TCP et IP.
 - sur un réseau Ethernet : $MSS = 1460$ octets
 - **Le contrôle d'erreur** :
 - TS (Time Stamps) : pour mettre à jour la valeur du délai de réémission RTT (Round Trip Time).
 - SACK (Selective ACK Permission /Indication) pour mettre en œuvre le service d'acquiescement sélectif.

Échanges à l'ouverture d'une connexion



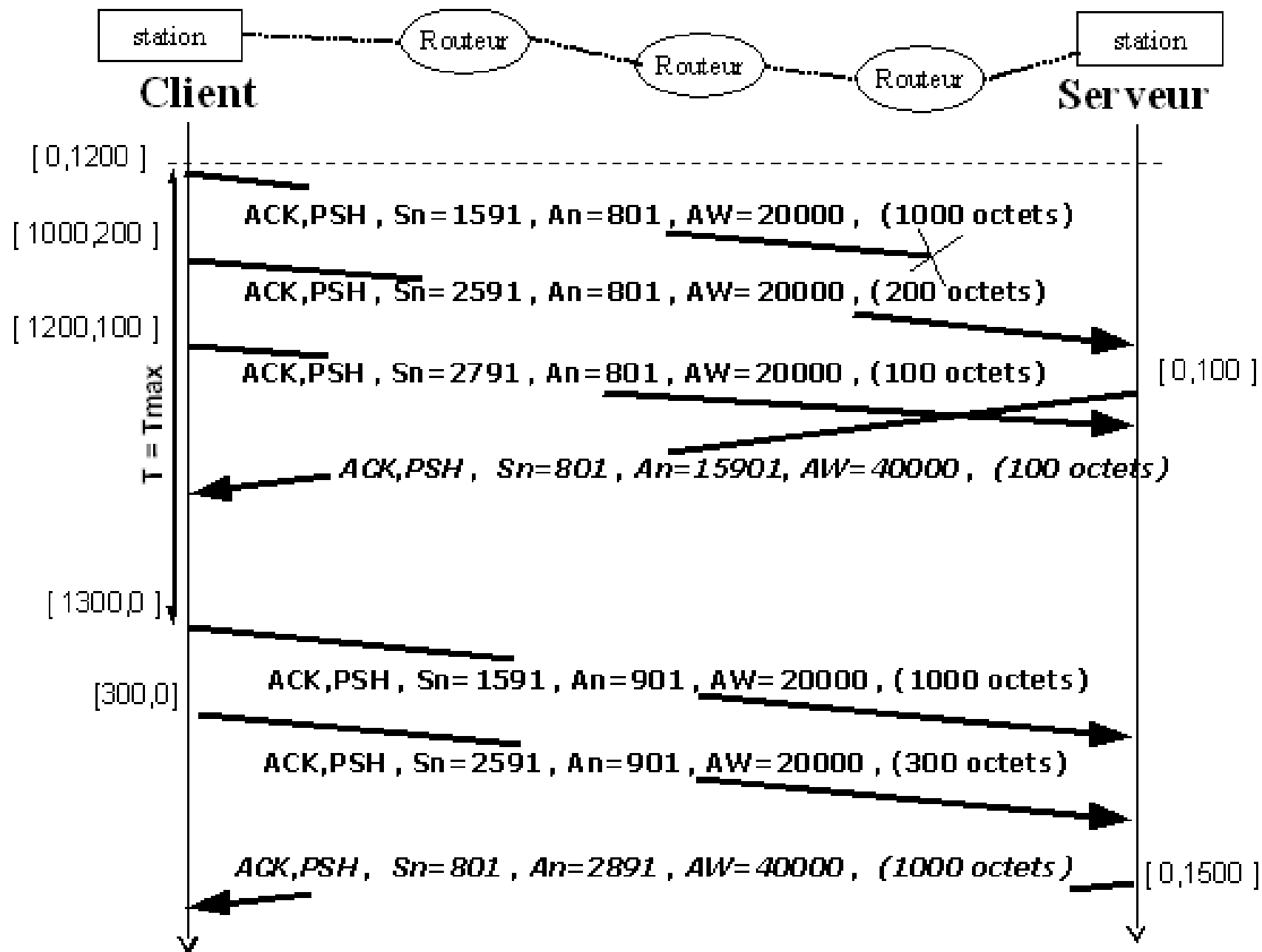
- Initialisation de N_s , A_n , MSS , AW , $T_{max} = 2 * RTT$

Échanges de données sur une connexion

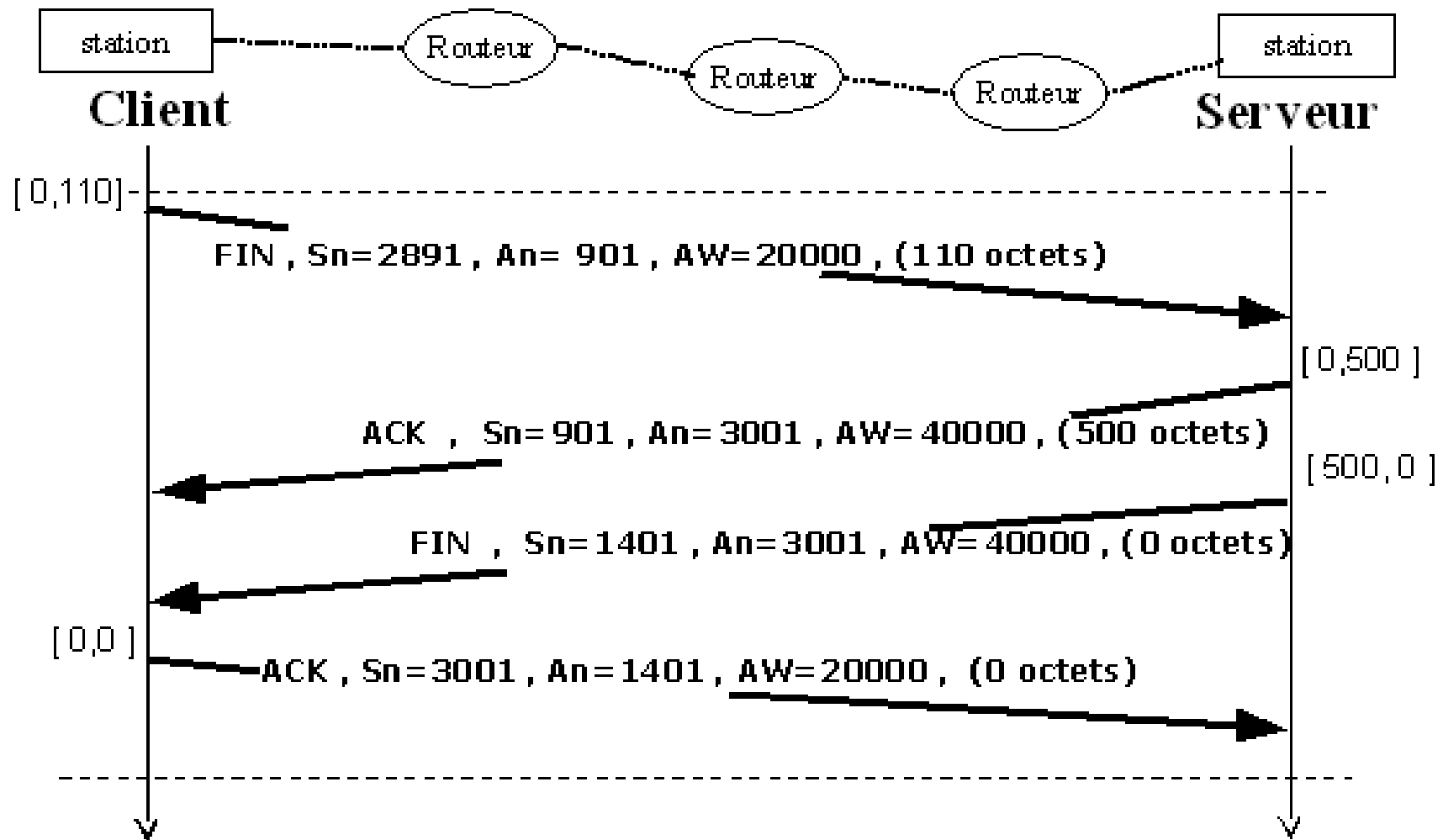


- **Acquittement** systématique de tout paquet reçu au bout d'un délai fixe.
- AW évolue en fonction de la **place** dans le tampon réception.

Échanges de données avec perte

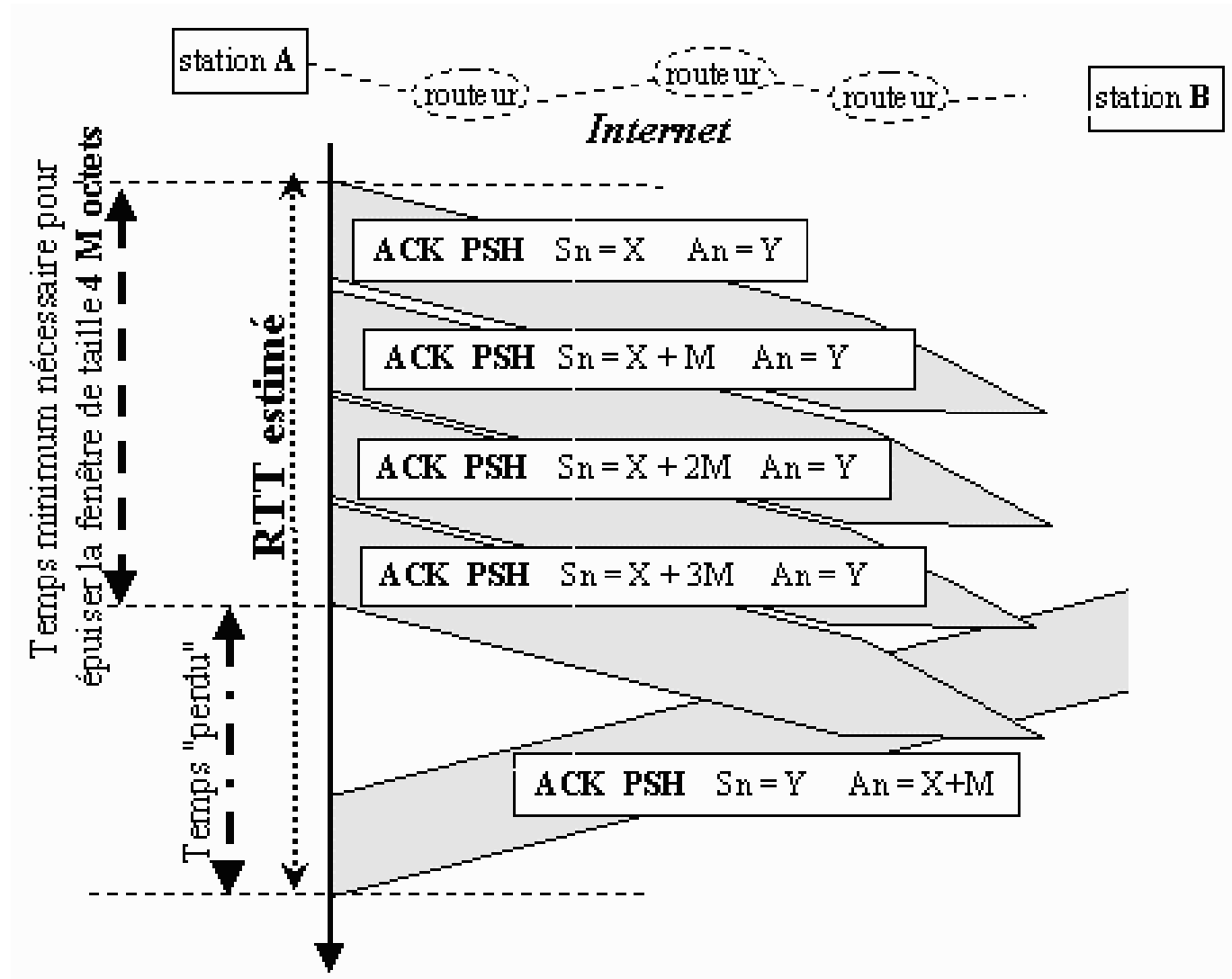


Échanges à la fermeture d'une connexion



- Après une demande FIN, les tampons sont vidés, puis la connexion arrêtée.
- De nombreux cas sont à considérer (double envoi de FIN...).

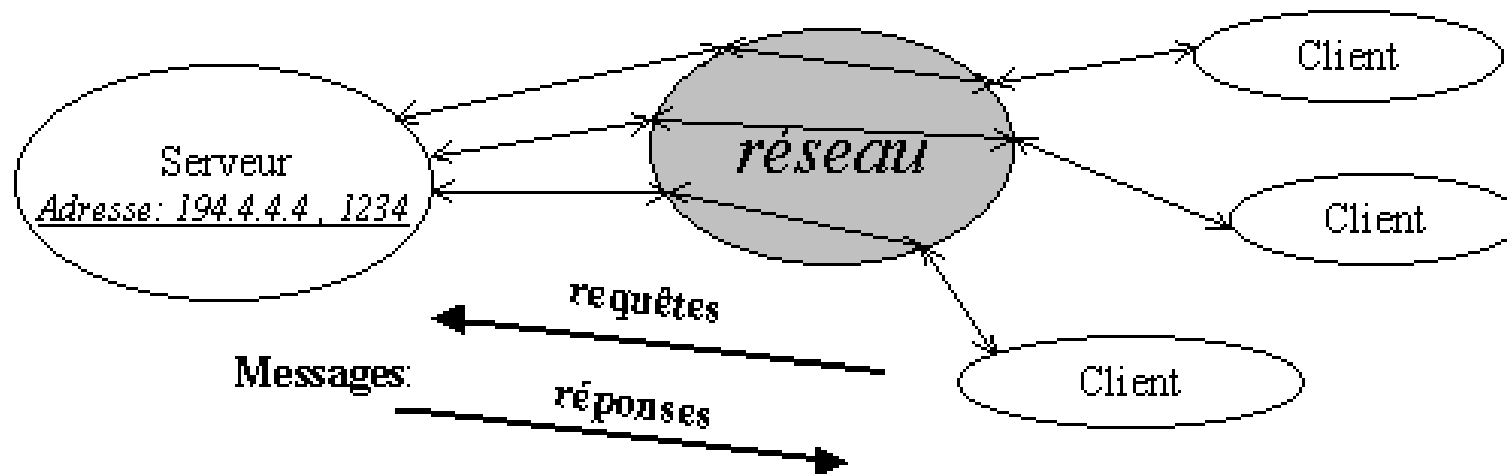
Évaluation optimale de W



- $W_{\text{optimal}} = \text{nb bits en transit} = D_{\text{tcp}} * \text{RTT}$

Interface socket

- Interface de programmation pour la mise en œuvre de programmes d'application au-dessus de la couche TCP ou UDP
- Modèle client /serveur :



Glossaire

- ACK
- An
- AW
- connexion TCP
- datagramme UDP
- mode flot
- MSS
- PSH
- RTT
- socket
- SYN
- Tampon
- Tmax
- W
- WS