La couche transport dans Internet (suite)

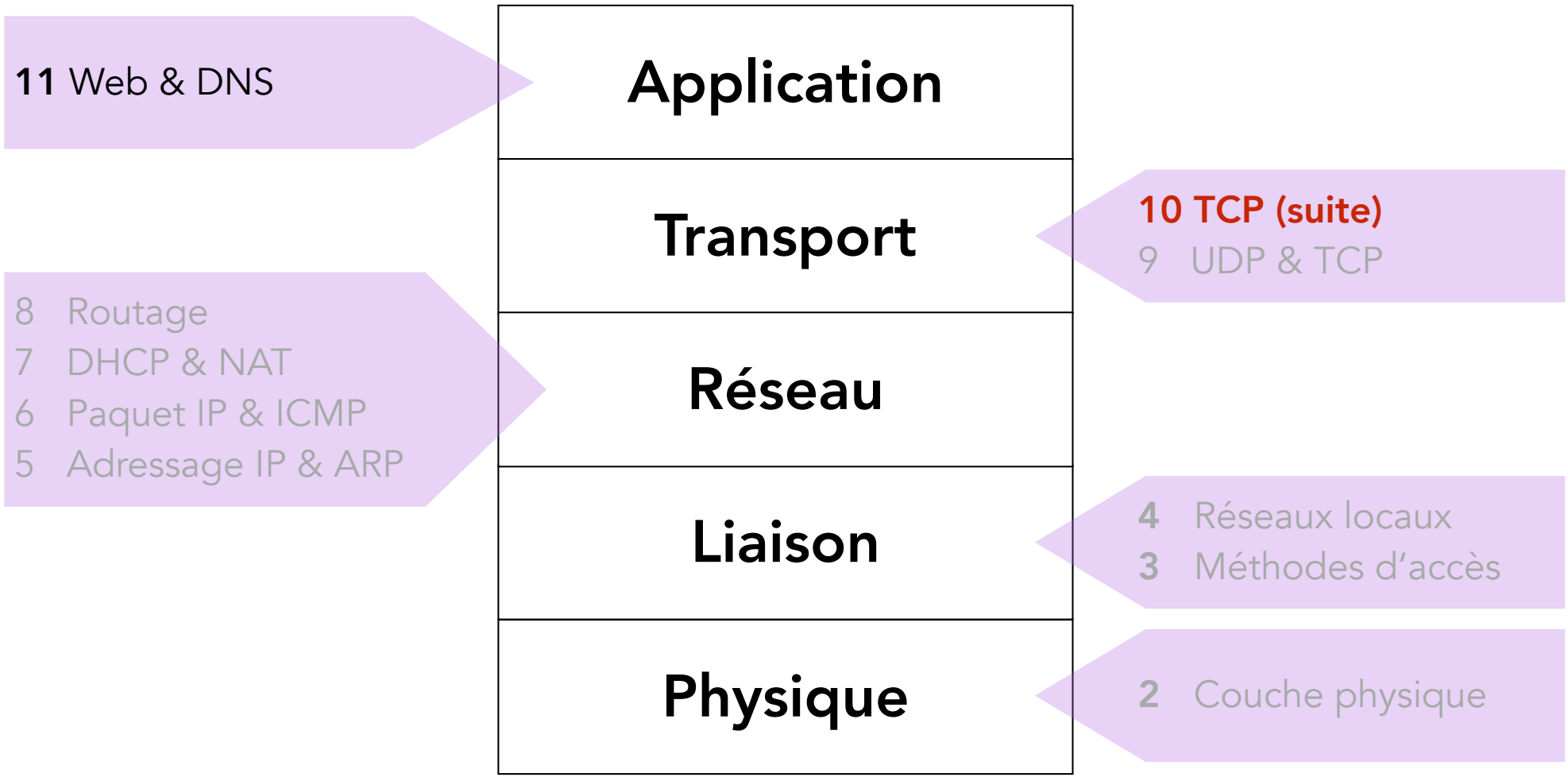
UE LU3IN033 Réseaux

2023-2024

Bruno Baynat

Bruno.Baynat@sorbonne-universite.fr

Programme de l'UE LU3IN033



Plan du cours

- Fiabilité et efficacité de TCP
- Transmission en mode flux d'octets
- Service en mode connecté
- Types et format des segments
- Numérotation des octets de données et accusés de réception
- Temporisation et retransmissions
- Contrôle de flux
- Octets de données urgentes et pushées
- Options TCP

TCP
Transmission Control Protocol

Transmission Control Protocol (TCP)

- Service en mode flux d'octets
  - L'écriture des octets à envoyer par l'application (dans le buffer d'émission) et l'envoi de segments par TCP ne sont pas corrélés
  - La lecture des octets reçus par l'application (dans le buffer de réception) et la réception de segments TCP ne sont pas corrélés
- Service orienté connexion
  - Un connexion TCP est constituée de 3 phases
    - phase d'établissement (ouverture) de la connexion
    - phase de transfert de données
    - phase de libération (fermeture) de la connexion
- Service de livraison fiable en séquence d'octets
  - Somme de contrôle (checksum)
    - pour détecter les octets en erreur
  - Numérotation en séquence des octets de données
    - pour détecter leur perte et les réordonner
  - Accusés de réception, temporisateurs et retransmissions
    - pour réparer les pertes ou les erreurs
- Mécanisme de contrôle de flux
  - Pour éviter l'engorgement des récepteurs
- Mécanisme de contrôle de congestion
  - Pour éviter l'engorgement du réseau

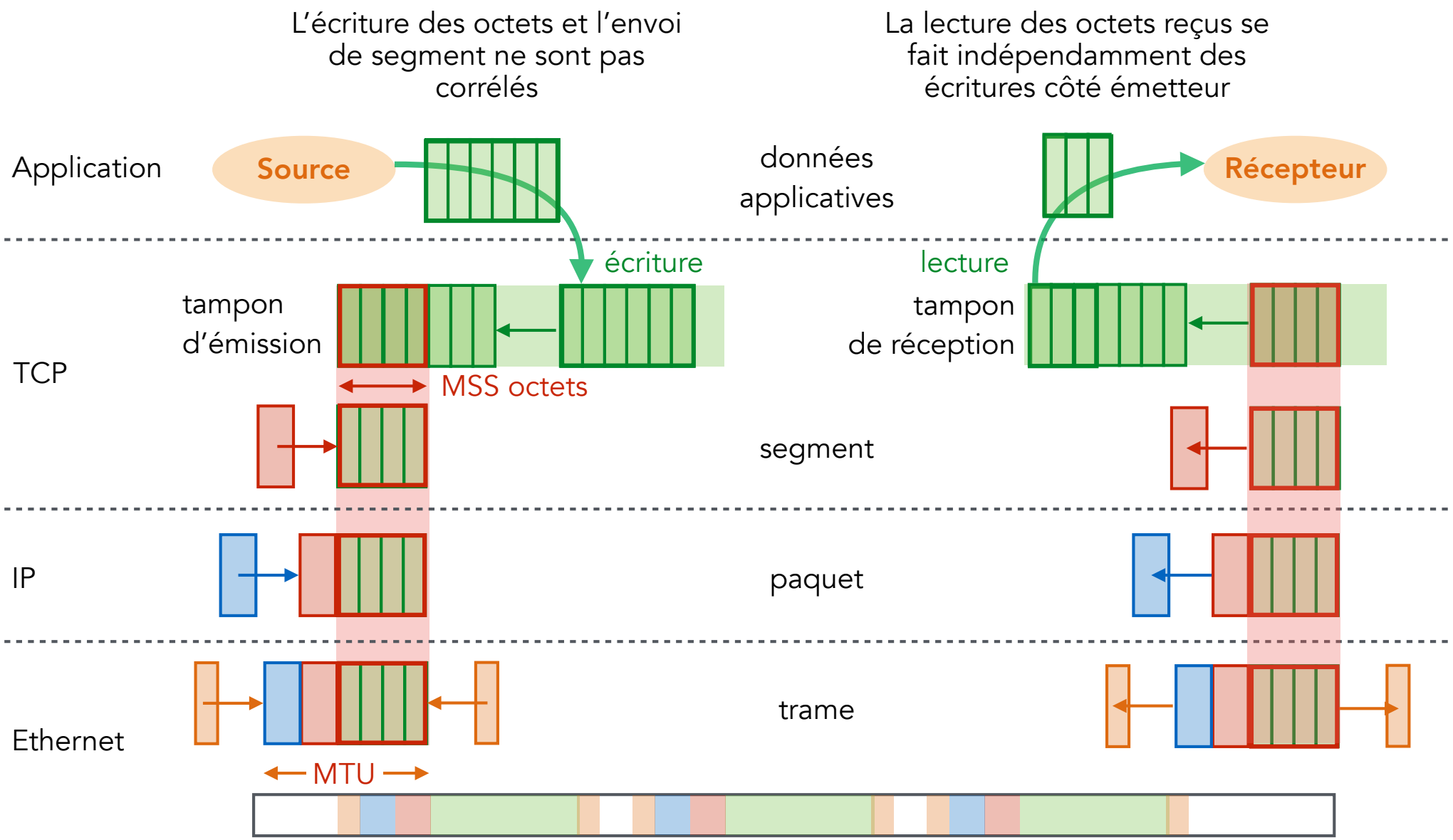
TCP : transfert fiable

- TCP (comme UDP) détecte les erreurs
  - somme de contrôle (checksum)
- Mais à l'encontre d'UDP…
- TCP détecte les pertes et les déséquencements
  - numéros de séquences
- TCP répare les erreurs et les pertes
  - buffer d'émission
  - acquittements des données
  - retransmissions
  - temporisateurs
- TCP réordonne les données déséquencées avant de les donner à l'application
  - buffer de réception

TCP : transfert efficace

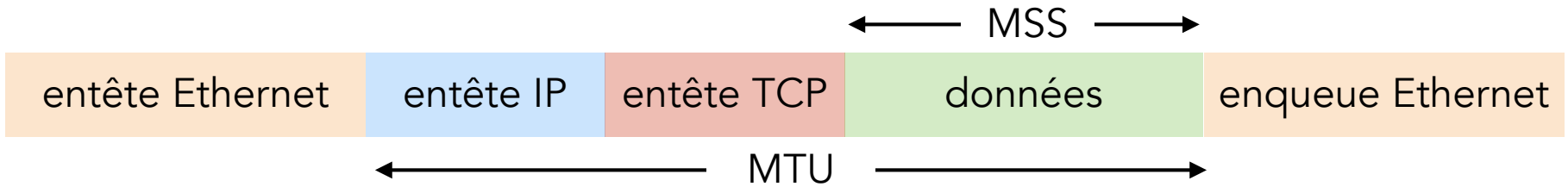
- TCP évite le gâchis de bande passante
  - en envoyant (dès que possible) des segments de taille maximum (MSS)
    - pour éviter l'envoi de trames à moitié vide
    - pour éviter la fragmentation IP
- TCP évite l'engorgement des buffers (des routeurs et des hôtes)
  - en limitant le débit d'émission de TCP
    - à la capacité d'absorption du récepteur : contrôle de flux
    - à la capacité de transfert du réseaux : contrôle de congestion
- TCP assure un partage équitable de la bande passante
  - en équilibrant de débit d'émission des différentes sources qui empruntent le même chemin
    - contrôle de congestion

TCP : Mode Flux d'octets



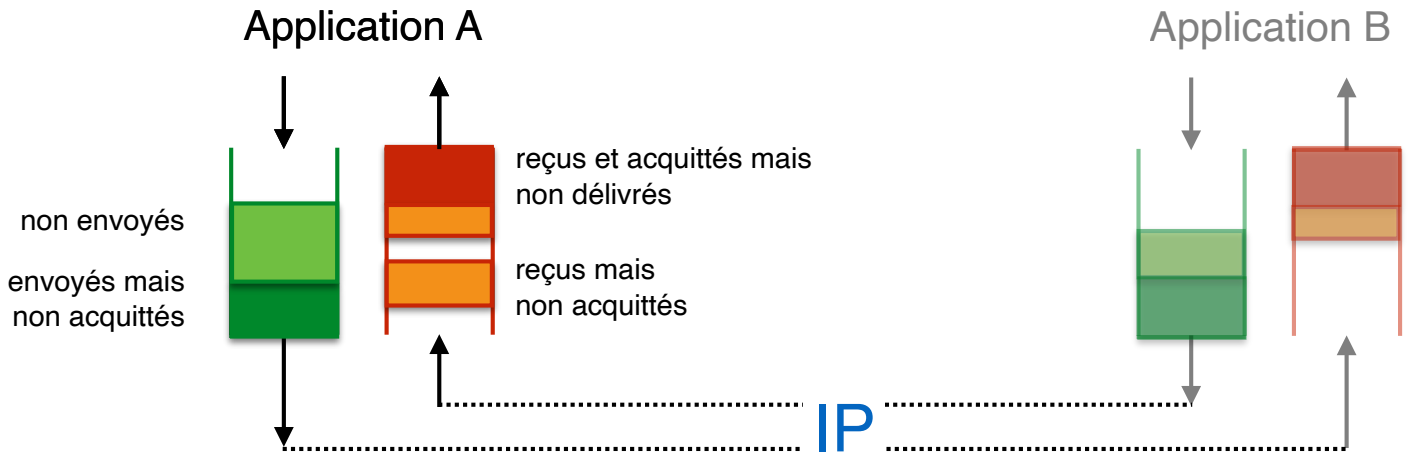
Constitution d'un segment TCP

- L'efficacité d'une transmission se mesure selon le taux de remplissage des trames
  - MTU (*Maximum Transmission Unit*) : taille max du champ données d'une trame
    - Ex : pour Ethernet MTU = 1500 octets
- TCP évite l'envoi de trames à moitié pleines
  - TCP attend que l'application ait écrit MSS octets dans le buffer d'émission
    - MSS : *Maximum Segment Size*
    - MSS = MTU – (taille en-tête IP + taille en-tête TCP)
    - MSS au-dessus d'Ethernet : 1460 octets si entêtes IP et TCP sans options
  - sauf si l'application demande explicitement à TCP d'envoyer un « petit » segment
    - fonction push
  - ou sur expiration d'un temporisateur
    - pour éviter d'attendre trop longtemps



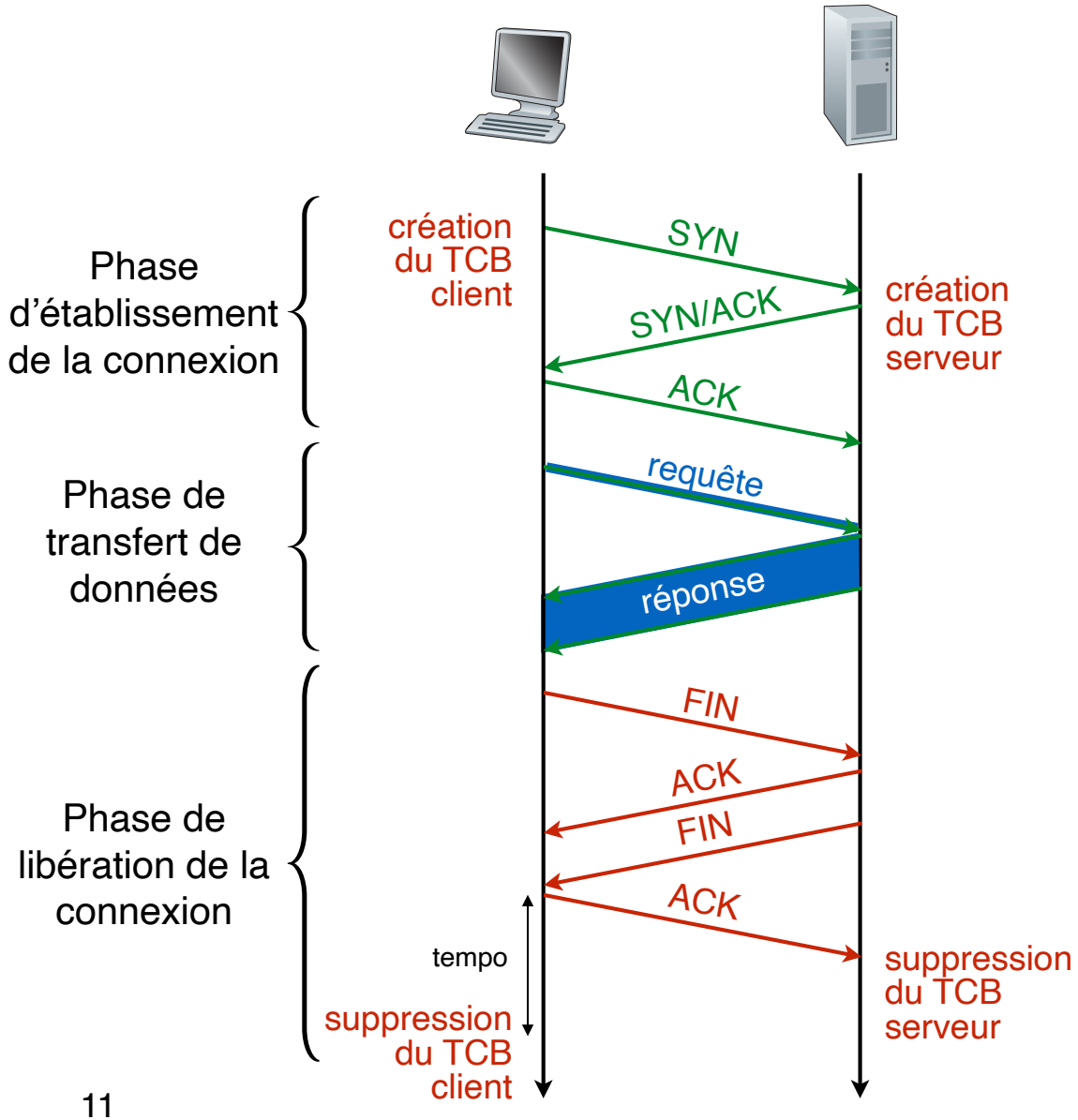
Buffers d'émission et de réception

- TCP offre aux deux applications en communication
  - un buffer d'émission où l'application considérée écrit les octets de données que TCP doit envoyer (de façon fiable) à l'application distante
  - un buffer de réception contenant les octets de données reçus de l'application distante, que l'application considérée vient lire quand elle le souhaite (dans l'ordre et sans erreur)

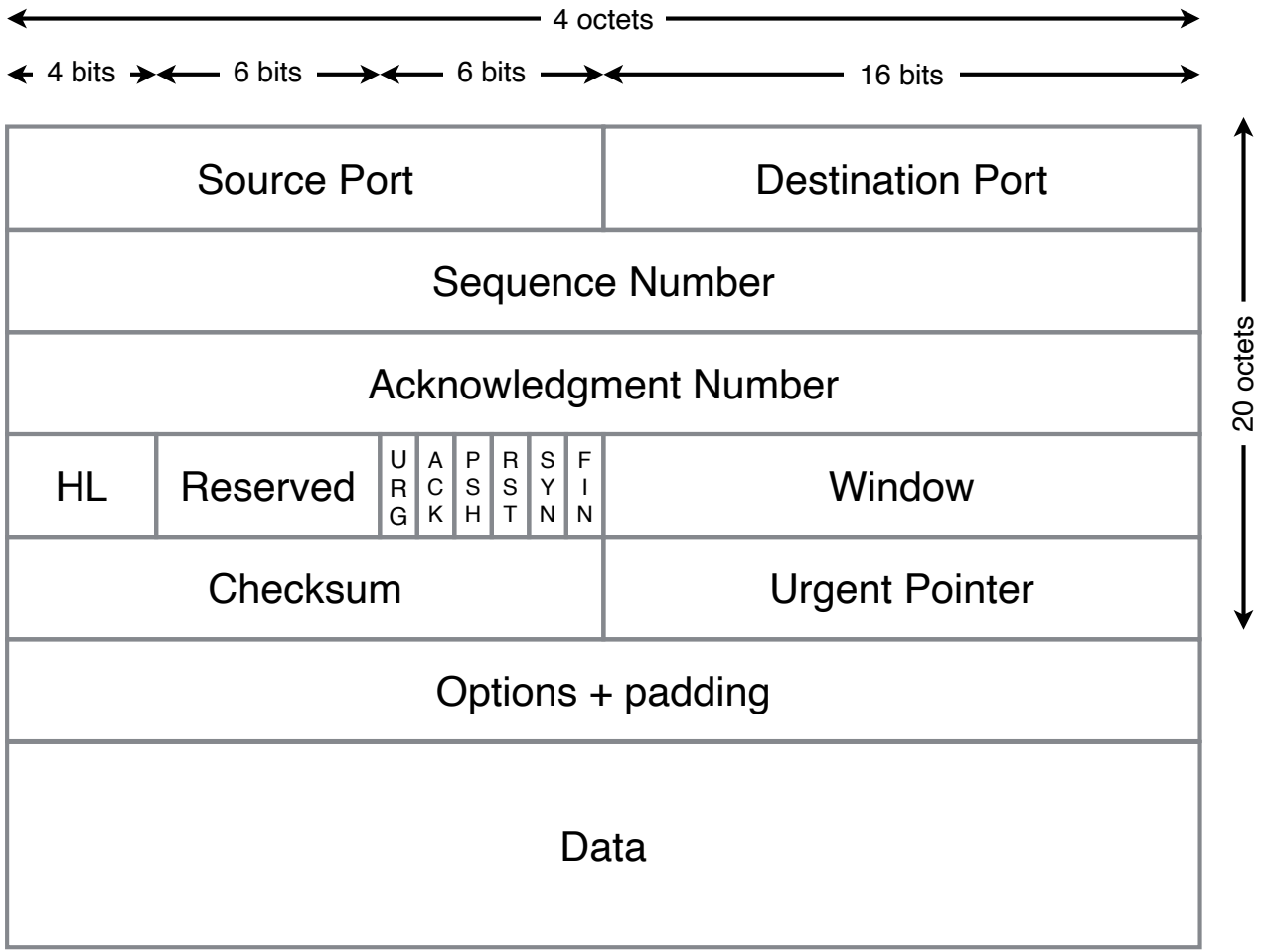


TCP : Mode connecté

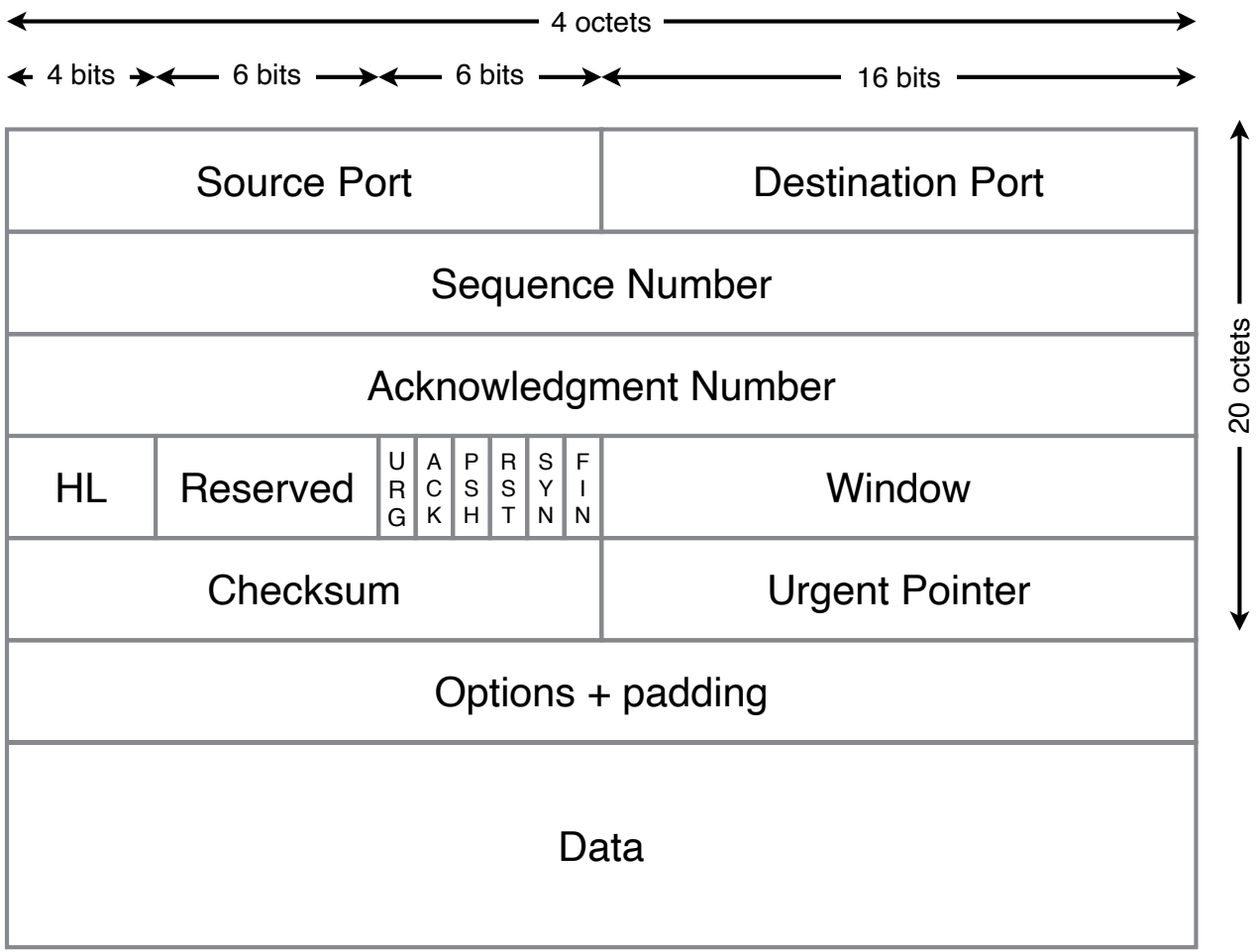
- Connexion TCP
  - 3 phases
  - ouverture : « *3-way handshake* »
    - segments SYN, SYN/ACK, ACK
  - fermeture : « *4-way handshake* »
    - segments FIN, ACK
- Les TCB (*Transport Control Block*) contiennent les informations d'état caractérisant l'échange
  - identifiants du processus (adresse IP, numéro de port), numéros de séquence des octets reçus en séquence et acquittés, valeurs des fenêtres, etc.
  - créés et initialisés à l'ouverture de la connexion
  - supprimés à la fermeture de la connexion
- Une connexion TCP est la combinaison des informations contenues dans les TCB client et TCB serveur



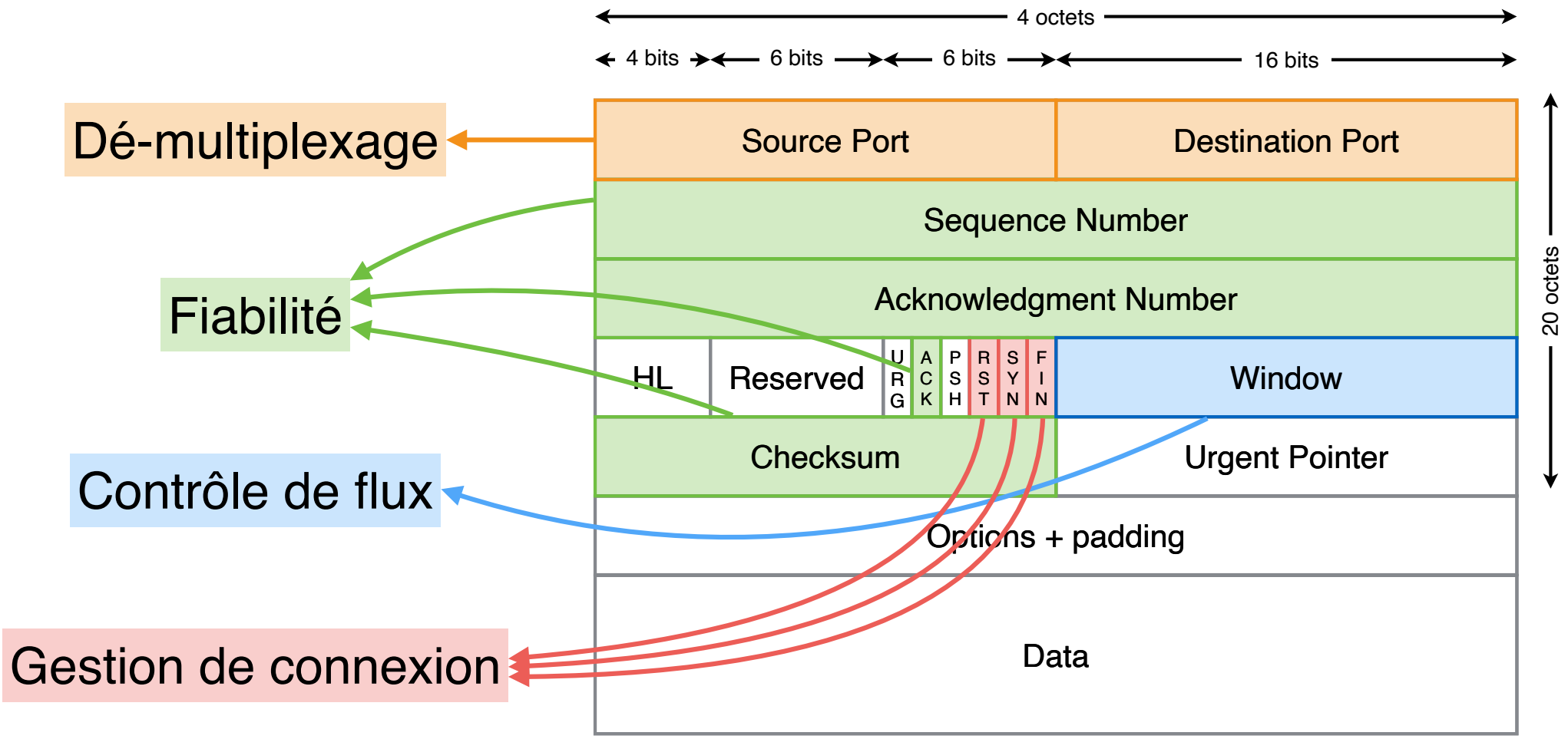
Entête TCP


4 octets
4 bits
6 bits
6 bits
16 bits
Source Port
Destination Port
Sequence Number
Acknowledgment Number
HL
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options + padding
Data
20 octets

Entête TCP


4 octets
4 bits
6 bits
6 bits
16 bits
Source Port
Destination Port
Sequence Number
Acknowledgment Number
HL
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options + padding
Data
20 octets

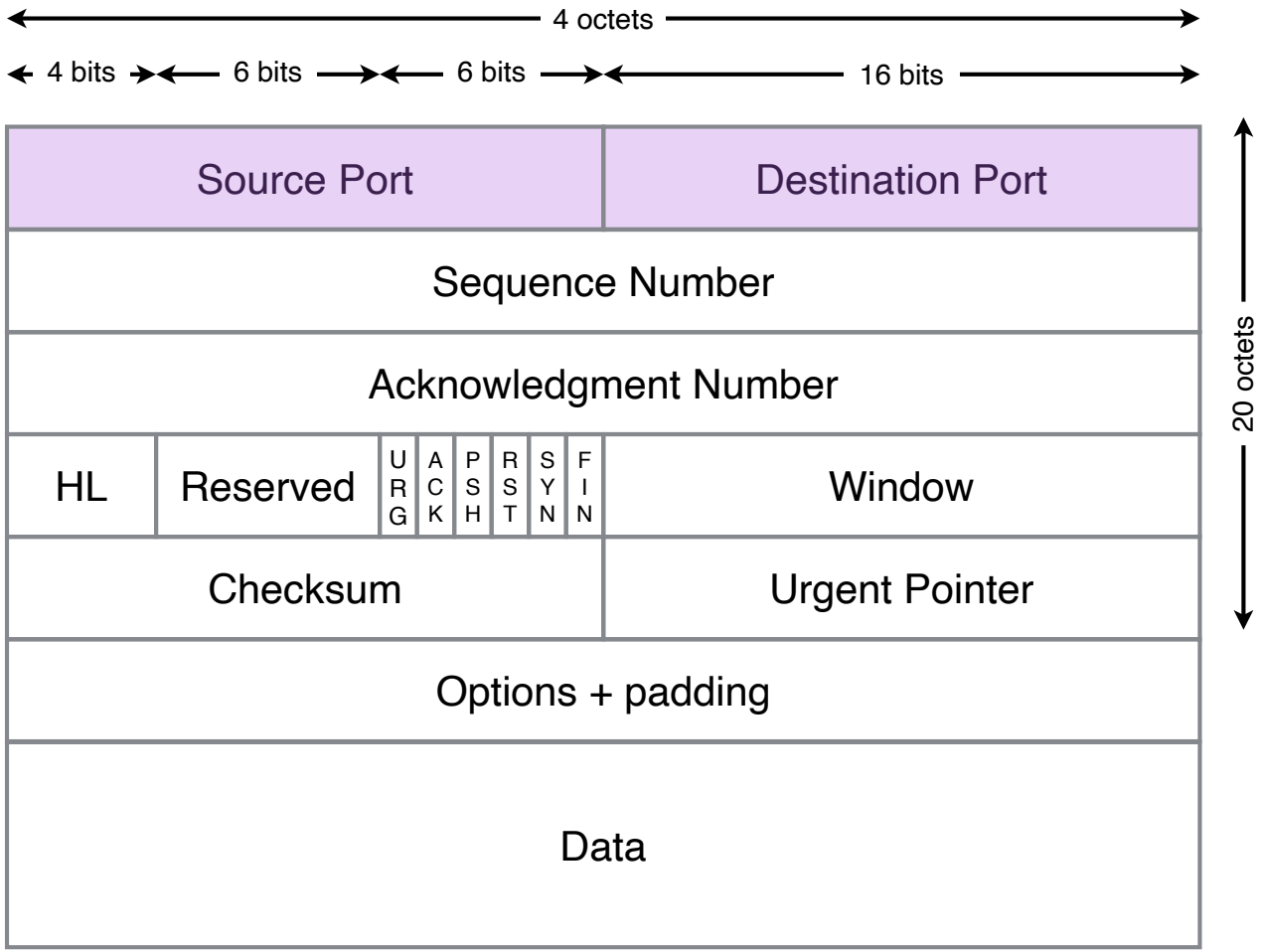
Entête TCP


4 octets
4 bits
6 bits
6 bits
16 bits
Dé-multiplexage
Source Port
Destination Port
Sequence Number
Fiabilité
Acknowledgment Number
HL
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Contrôle de flux
Options + padding
Data
Gestion de connexion
20 octets

Numéros de port

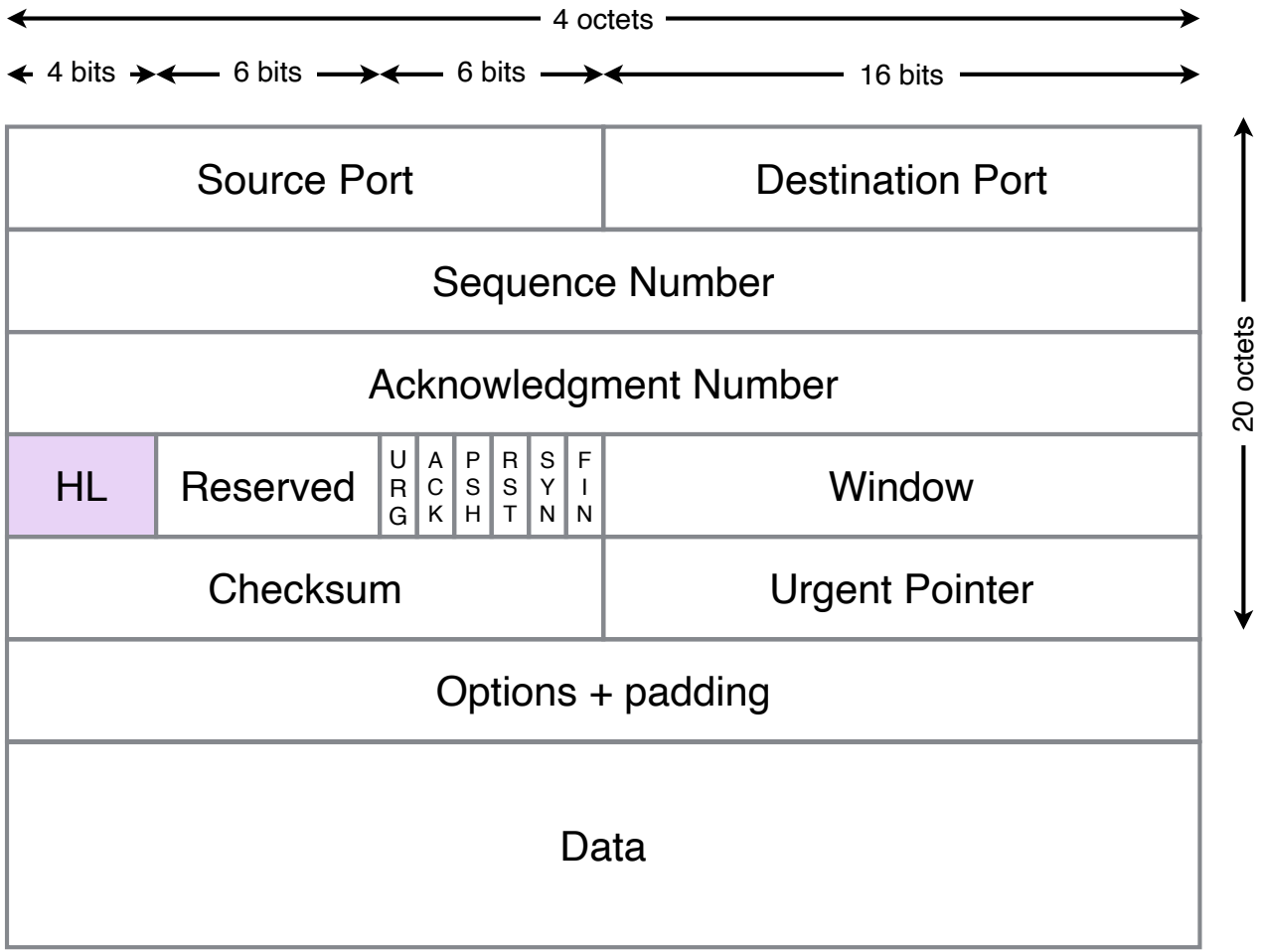
- Numéro de port source
  - identifie le processus qui émet le segment
- Numéro de port destination
  - identifie le processus à qui est destiné le segment

- Numéro de port client
  - valeur arbitraire ≥ 1024
    - laissé au choix de l'OS
- Numéro de port serveur
  - valeur connue < 1024
    - Ex : 80 pour un Serveur web



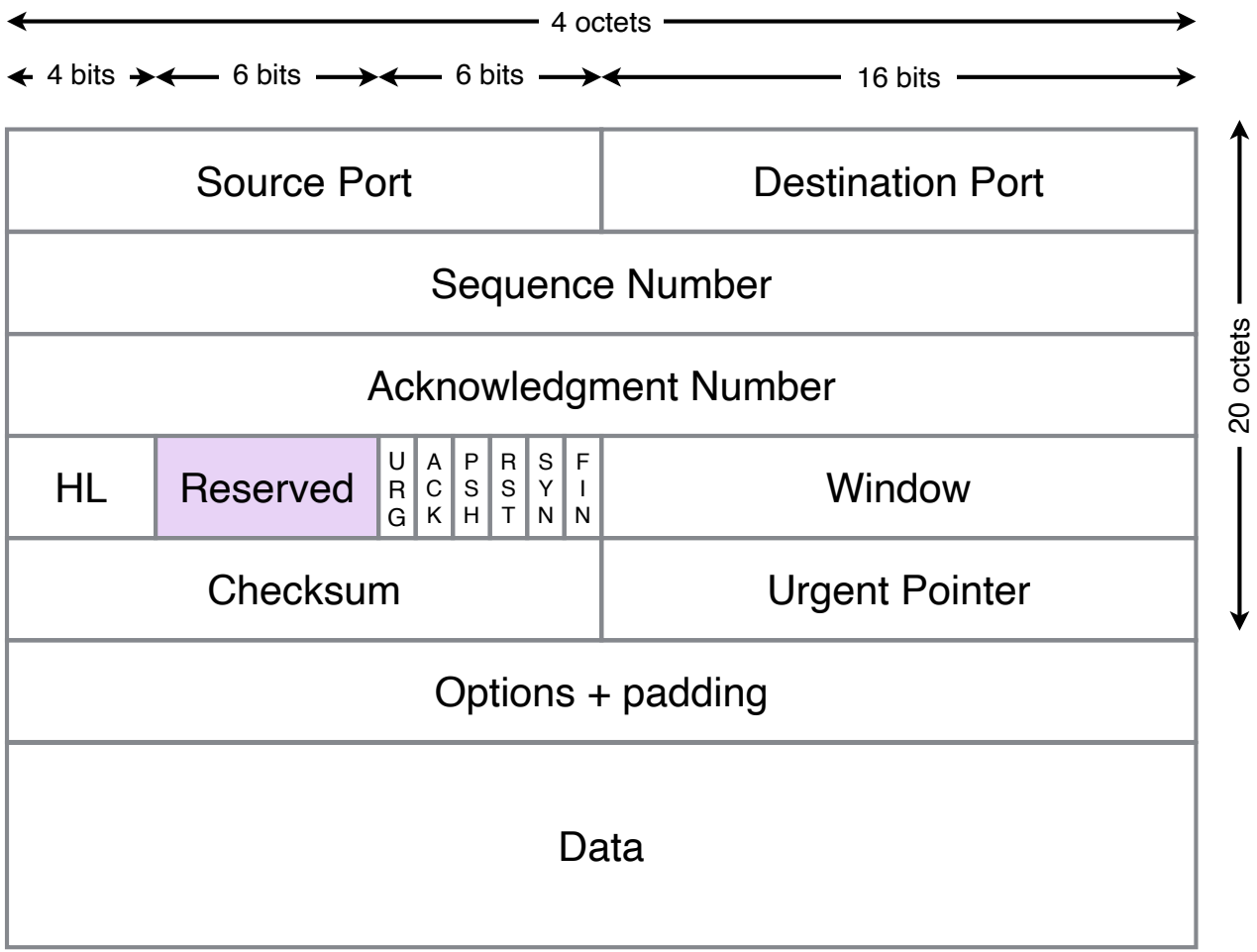
Champ THL

- HL (*Header Length*)
  - taille de l'entête exprimée en mots de 32 bits (4 octets)
- Taille min
  - HL = 0x5 (0101) : 5
    - entête de 20 octets
    - pas d'options TCP
- Taille max
  - HL = 0xF (1111) : 15
    - entête de 60 octets
    - 40 octets d'options TCP



Champ Reserved

- Champs réservé à l'origine pour une utilisation future
  - aujourd'hui non utilisé
  - doit être laissé à zéro (000000)



Champ Checksum

paquet IP

4	IHL	TOS	Total Length
Identifier		R DF MF	Fragment offset
TTL	Protocol	Header checksum	
Source IP address			
Destination IP address			

pseudo entête

Source IP address		
Destination IP address		
0	Protocol	TCP length

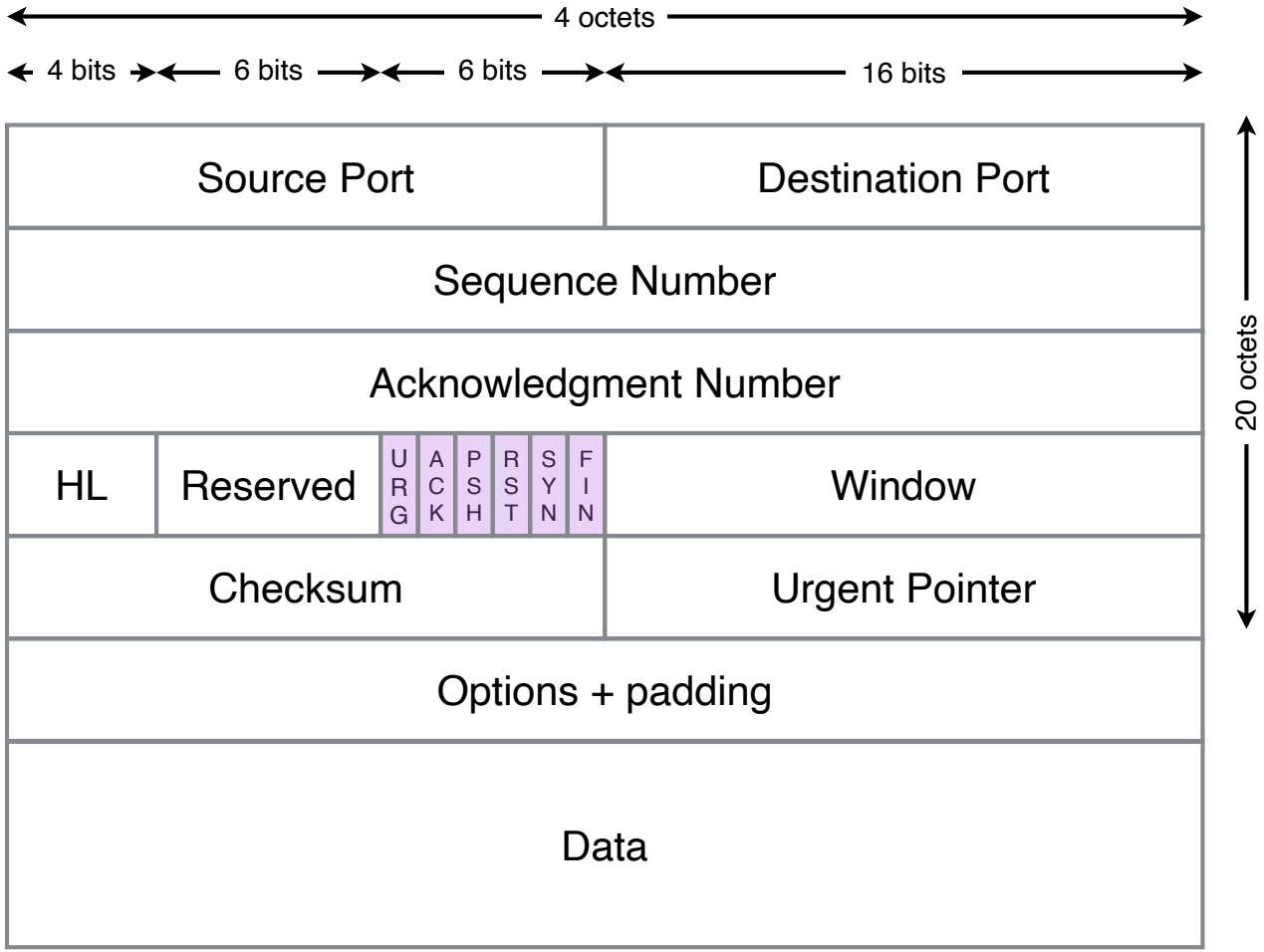
segment TCP

Source Port								Destination Port
Sequence Number								
Acknowledgment Number								
HL	Reserved	URG	ACK	PSH	RST	SYN	FIN	Window
Checksum								Urgent Pointer
Options + padding								
Data								

- *Checksum* : somme de contrôle portant
  - sur l'en-tête du segment
  - sur les données transportées
  - sur un « pseudo-entête »

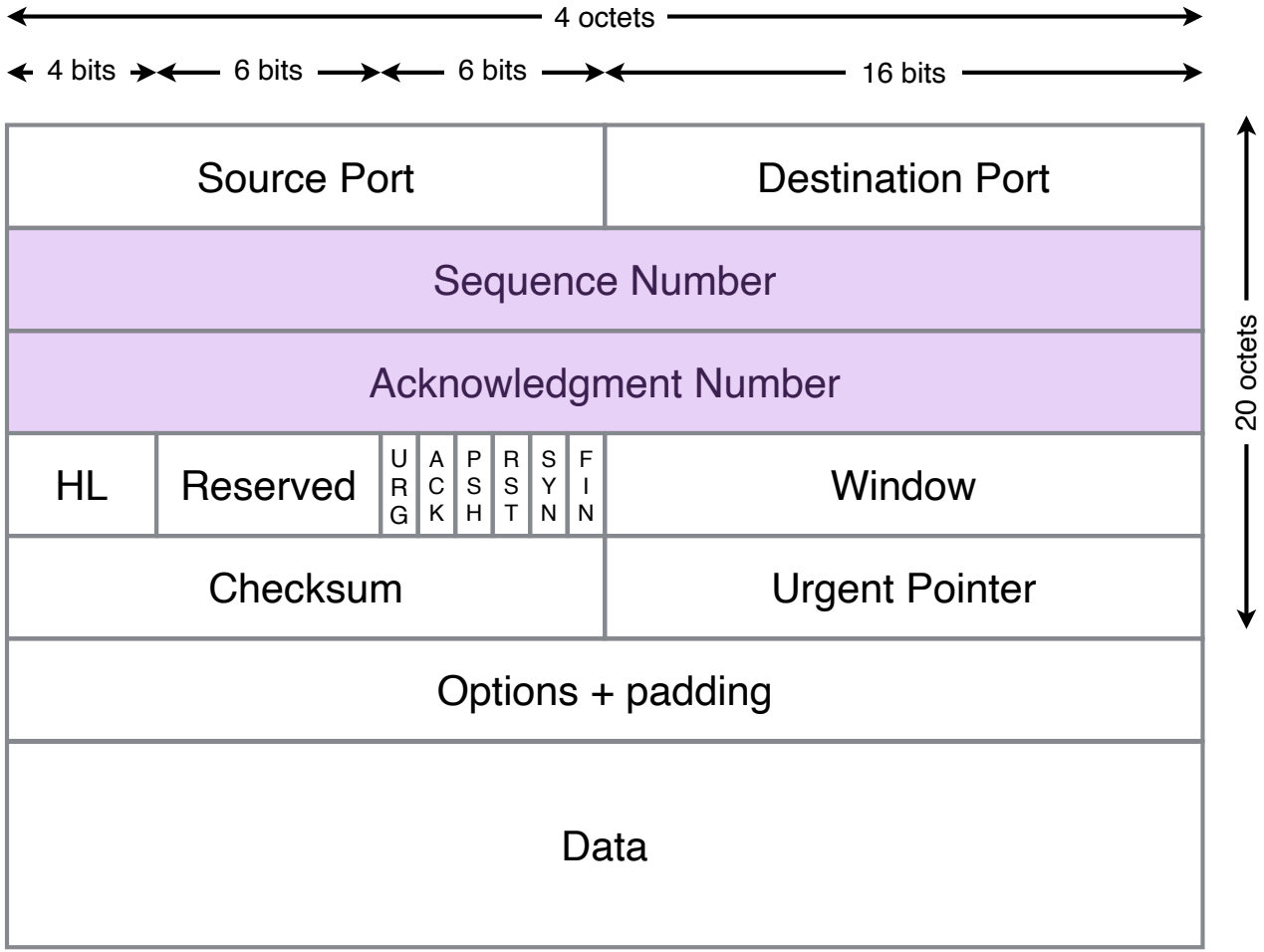
Drapeaux TCP

- SYN, FIN et RST
  - identifient 3 types de segment
  - les segments de données n'ont pas de drapeau dédié
- SYN
  - ouverture de connexion
- FIN
  - fermeture de connexion
- RST
  - réinitialisation de la connexion
- ACK
  - le champ AN est valide
- PSH
  - les données doivent être lues au plus vite par l'application côté récepteur
- URG
  - le segment contient des données urgentes



Numérotation des segments

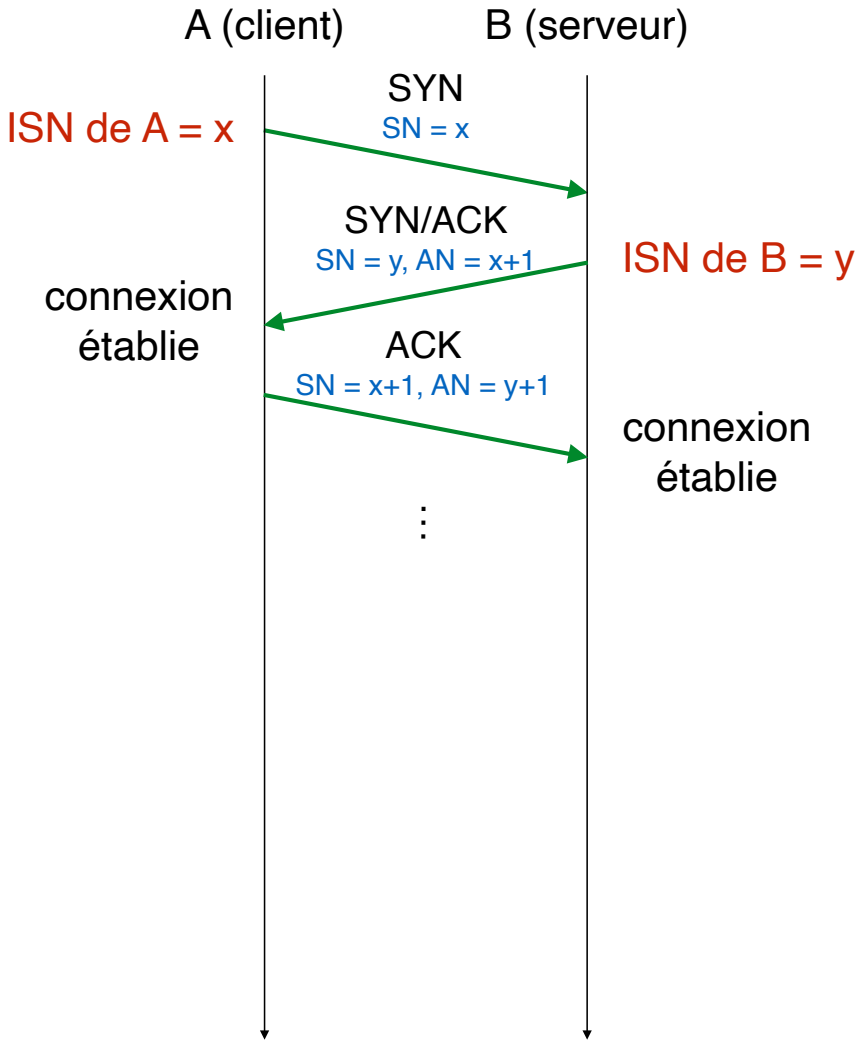
- Numéro de séquence
  - pour un segment SYN : ISN
  - pour un segment de données : numéro du premier octet transporté par le segment
  - pour un segment FIN : numéro du dernier octet transporté + 1
- Numéro d'acquittement
  - valide uniquement lorsque le drapeau ACK est positionné
  - accuse la réception des segments SYN et FIN
  - accuse la réception des octets de données envoyés par l'application distante



Numéros de séquence et d'acquittement

Phase d'établissement de la connexion

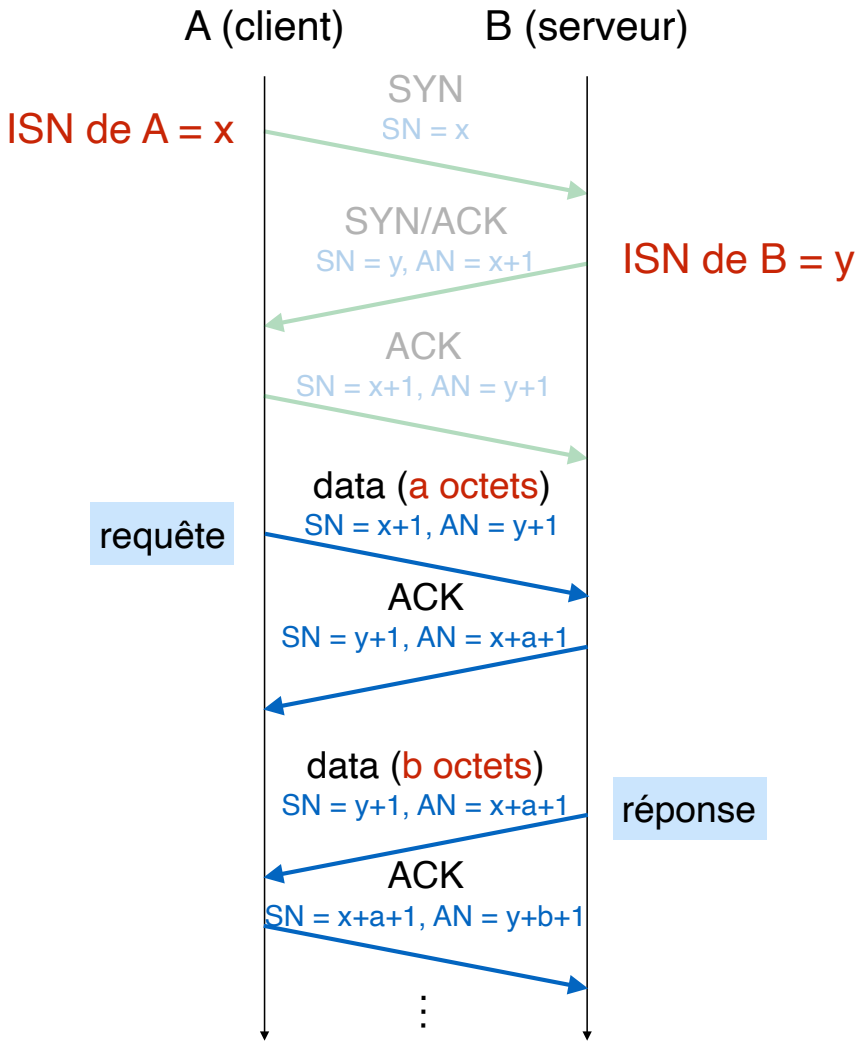
- Segments SYN
  - Ouverture bilatérale d'une connexion
    - le client envoie un segment SYN auquel le serveur répond par un segment SYN/ACK
    - la connexion TCP est établie une fois que le SYN/ACK du serveur est acquitté par le client
  - Synchronisation des numéros de séquence
    - le champ SN (*Sequence Number*) contient la valeur de l'ISN (*Initial Sequence Number*)
    - choisie aléatoirement
  - Paramétrage de la connexion (options TCP de l'entête du SYN et du SYN/ACK)
- Segments ACK
  - accusent la bonne réception des SYN
    - le champ AN (*Ackowledgment Number*) contient le SN du SYN incrémenté de 1



Numéros de séquence et d'acquittement

Phase de transfert de données

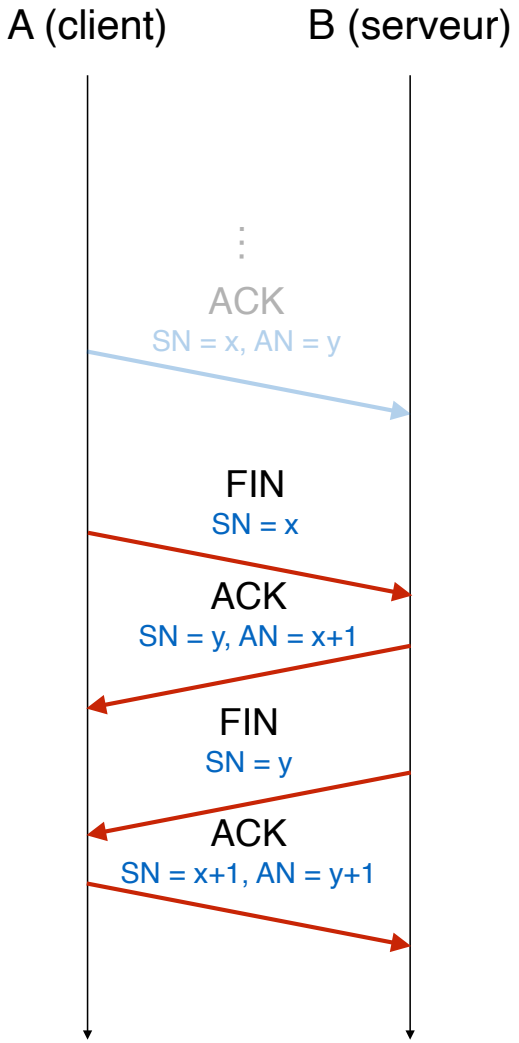
- Segments de données
  - Drapeaux
    - SYN, FIN et RST à 0
    - ACK à 1
    - PUSH et URG à 0 ou à 1
  - le champ SN (*Sequence Number*) contient le numéro de séquence du premier octet de données transporté dans le segment
  - le champ AN (*Ackowledgment Number*) contient le numéro de séquence du prochain octet attendu de l'application distante
    - acquittement dans les données : « piggybacking »
- Segments ACK
  - ne transportent pas de données
  - accusent les octets correctement reçu
    - le champ AN contient le numéro de séquence du prochain octet attendu de l'application distante



Numéros de séquence et d'acquittement

Phase de libération de la connexion

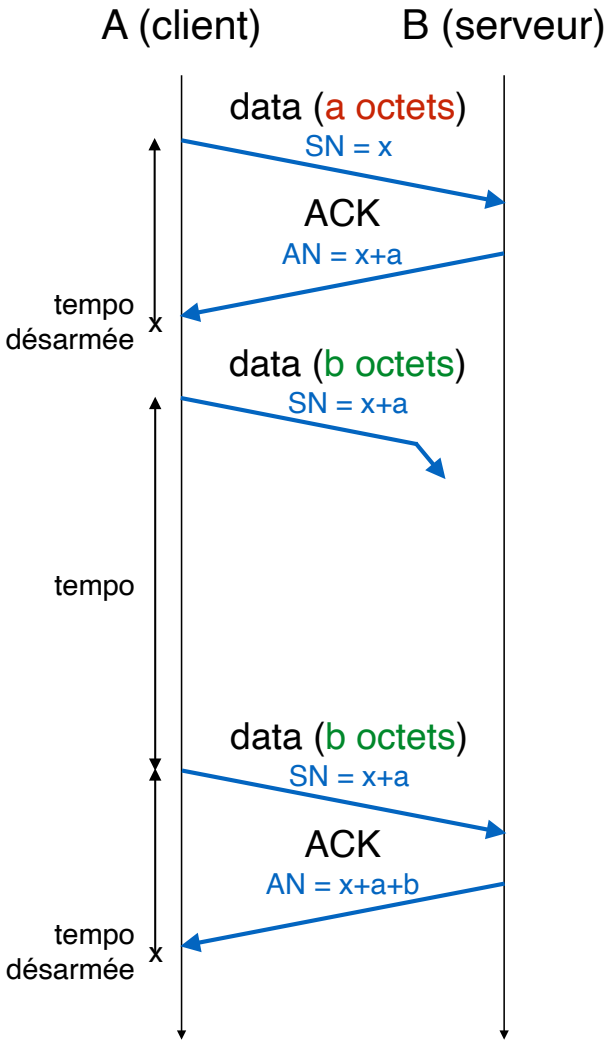
- Segment FIN
  - Fermeture bilatérale de connexion
    - serveur et client envoient un segment FIN
    - la connexion est fermée une fois les deux FIN acquittés
- Segments ACK
  - accusent la bonne réception des FIN
    - en incrémentant de 1 le SN du FIN



Retransmissions

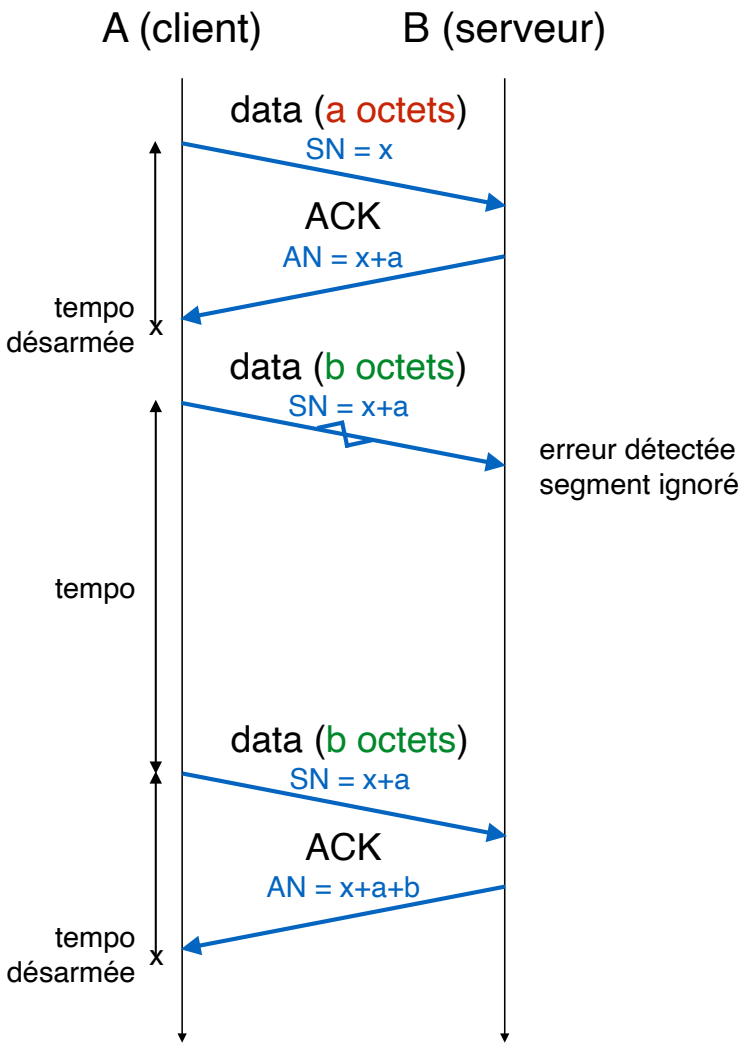
En cas de pertes d'un segment de données

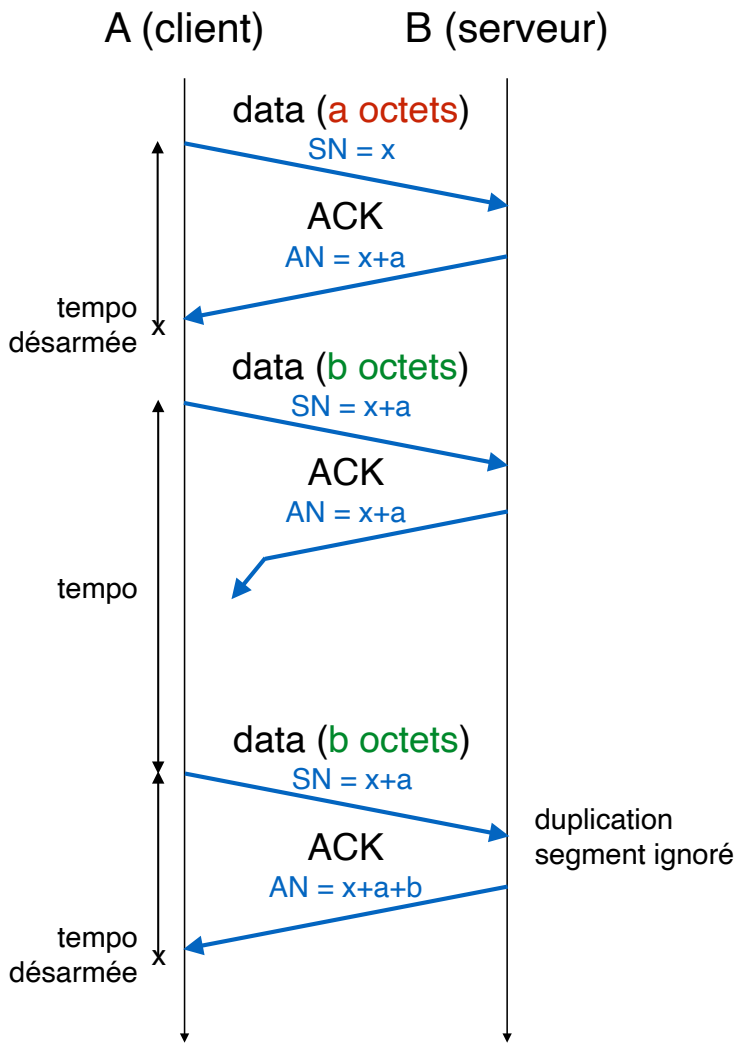
- Pour chaque segments de données envoyé une temporisation est déclenchée
  - si un acquittement couvrant les octets transportés dans le segment revient avant la fin de la tempo
    - la tempo est désarmée
  - sinon
    - un segment contenant les mêmes octets de données est retransmis



Retransmissions

En cas d'erreur ou de perte d'un ACK

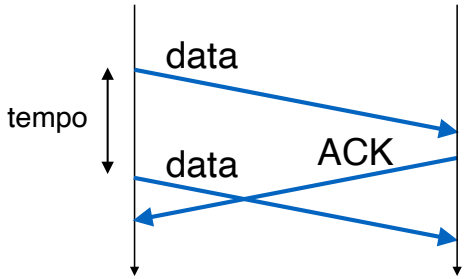




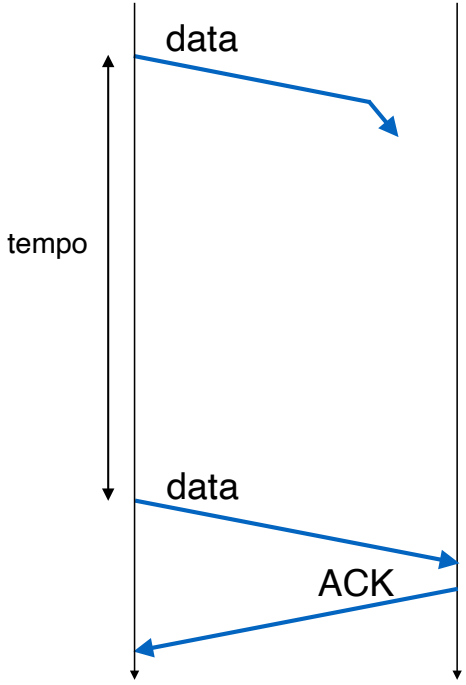
Temporisation de retransmission

Comment dimensionner la durée de la temporisation ?

- Si elle est trop courte, elle risque de se déclencher avant le retour de l'acquittement
  - Retransmissions inutiles



- Si elle est trop longue, elle génère une attente trop longue avant la réparation d'une perte
  - Retards inutiles



Temporisation de retransmission

- La temporisation de retransmission doit être dimensionnée de façon à laisser à l'acquittement le temps de revenir
  - RTO (*Retransmission Time-Out*) > RTT (*Round-Trip Time*)
- Le RTT sur une connexion TCP
  - dépend de la distance entre la source et la destination
  - dépend de la bande passante du chemin emprunté
  - dépend de la charge du réseau
  - varie d'un segment à l'autre
- Le RTT doit être estimé par une moyenne sur des valeurs mesurées
  - adaptée à la connexion
  - variable dans le temps
- TCP calcule une moyenne glissante du RTT appelée le SRTT (*Smoothed Round-Trip Time*)
  - à chaque nouvel envoi d'un segment, TCP mesure le délai nécessaire au retour de l'acquittement correspondant : RTT
  - TCP calcule alors le SRTT :

$$SRTT = \alpha * SRTT + (1 - \alpha) * RTT$$

où α est un facteur de lissage (Ex : $\alpha = 0{,}9$)

  - et la valeur courante du RTO :

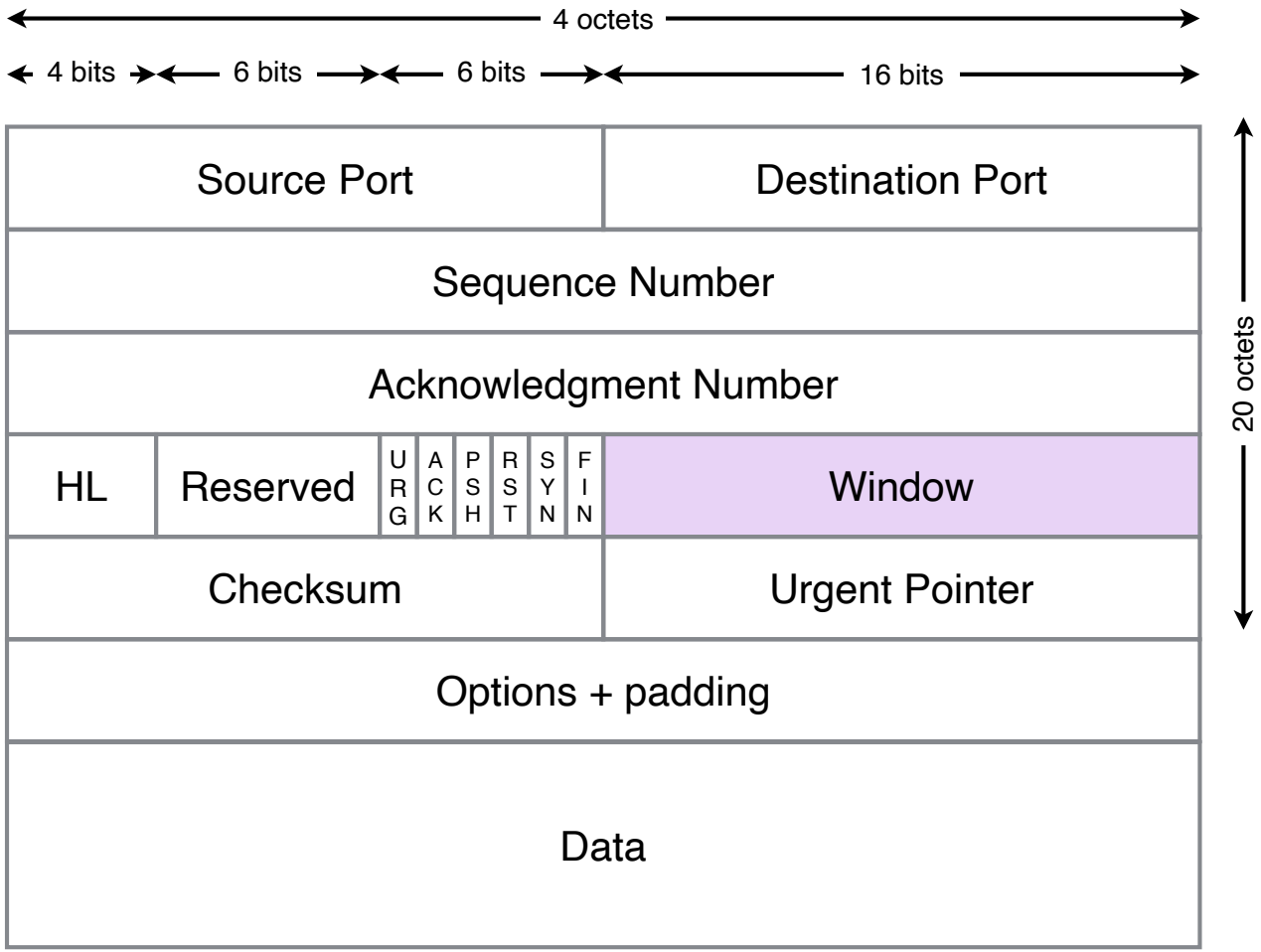
$$RTO = \beta * SRTT$$

où β est un facteur de pondération (Ex : $\beta = 2$)

  - valeur initiale du RTO = 1 seconde

Champ Window

- Champ *Window*
  - utilisé pour le contrôle de flux
  - indique le nombre d'octets prêts à être reçu à partir du numéro AN



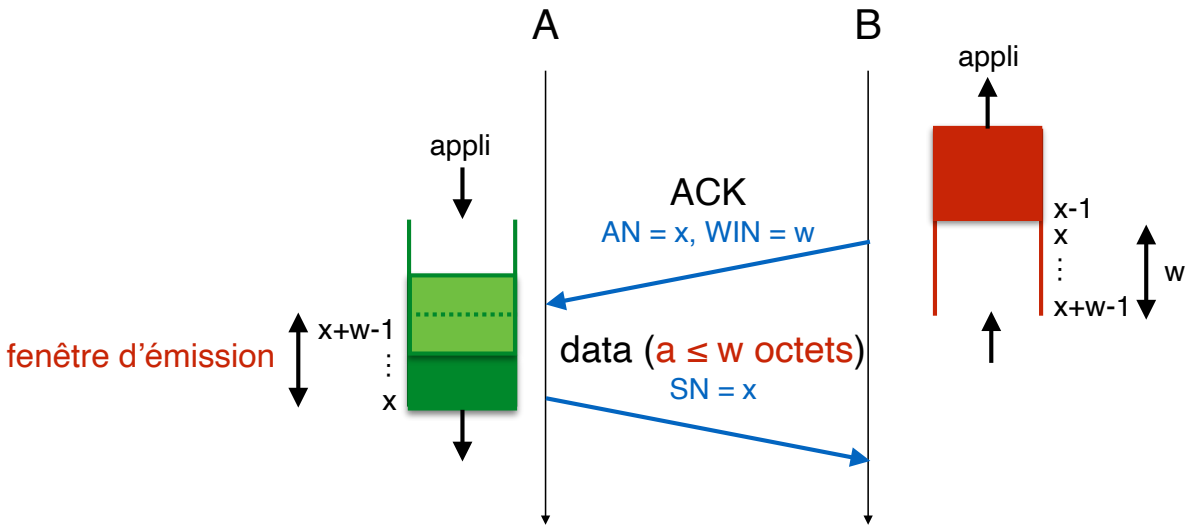
Contrôle de flux

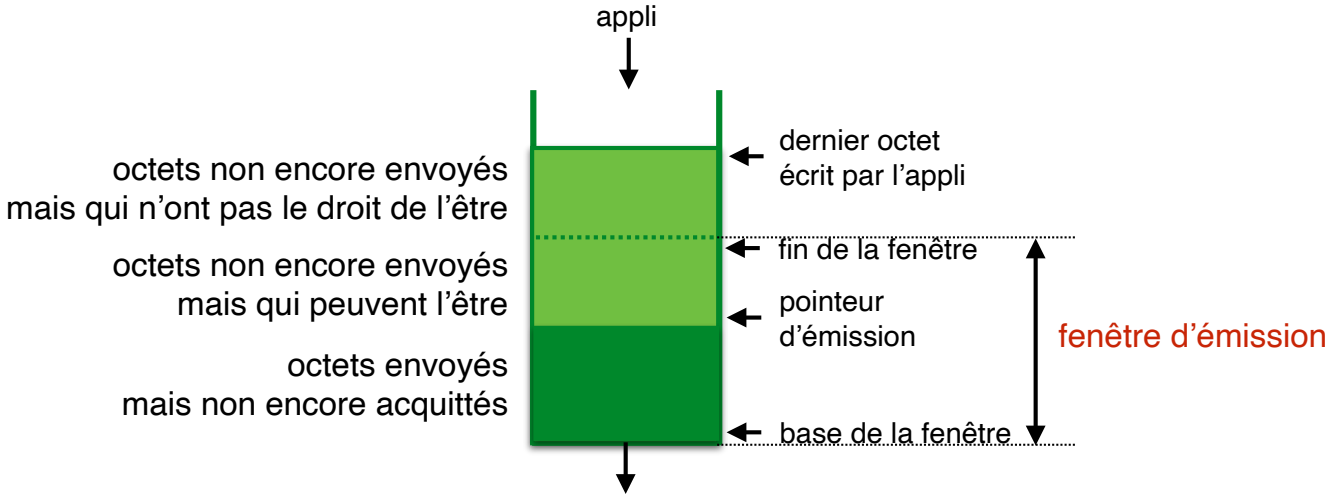
- TCP offre un mécanisme contrôle de flux
  - permettant à l'émetteur d'adapter sa vitesse d'émission à la capacité de traitement du récepteur
  - au travers d'une fenêtre d'émission appelée « fenêtre de contrôle de flux » (*Flow Control Window*)
    - plage d'octets que l'émetteur à le droit d'émettre
    - afin ne pas saturer le récepteur
  - corrélée à la place libre dans le buffer de réception du récepteur

Contrôle de flux

Fenêtre d'émission

- Lorsque B envoie à A un ACK avec
  - AN = x
  - WIN = w
- B donne le droit à A d'envoyer
  - w octets de données
  - dont les numéros de séquence sont compris entre x et x+w-1



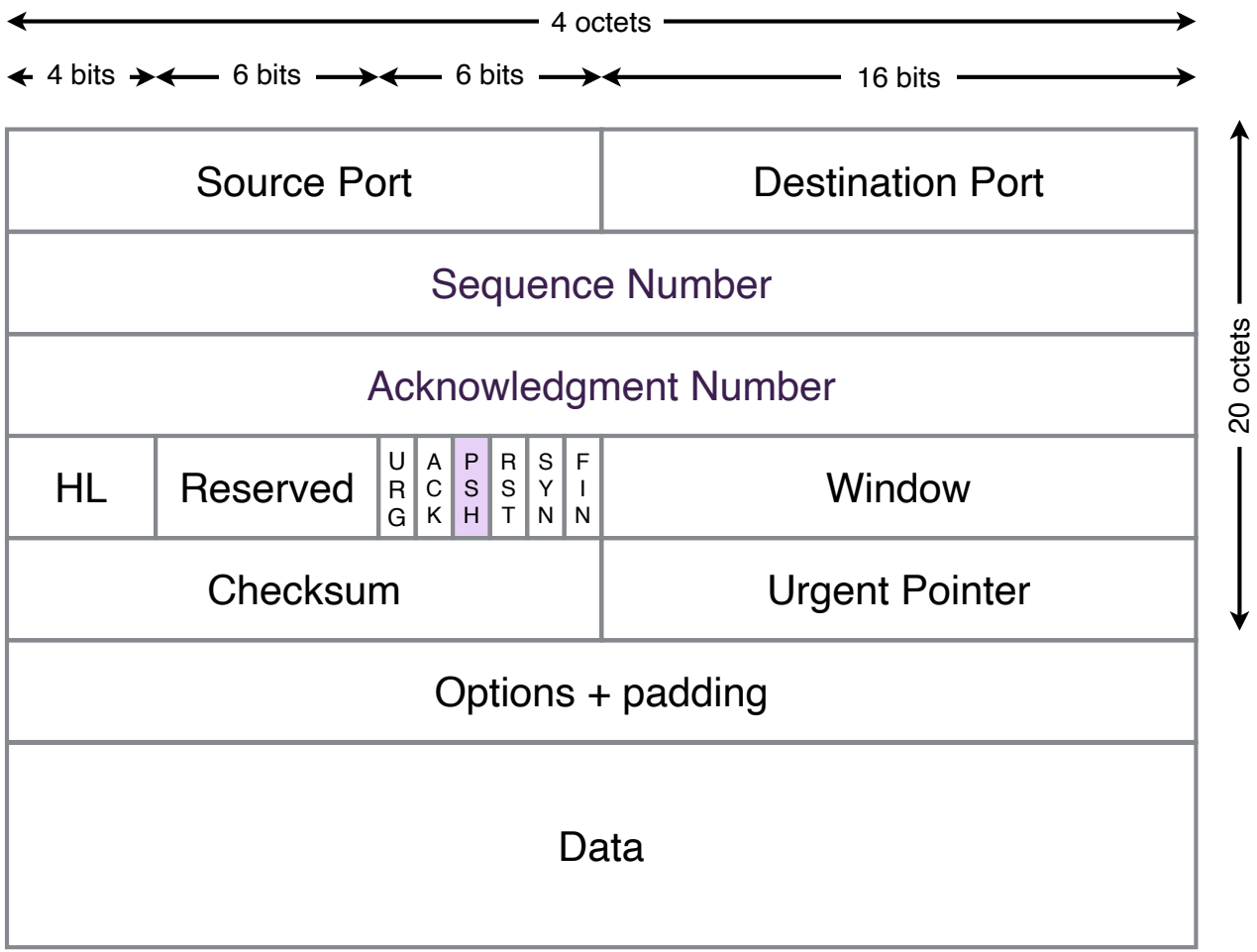


Contrôle de congestion

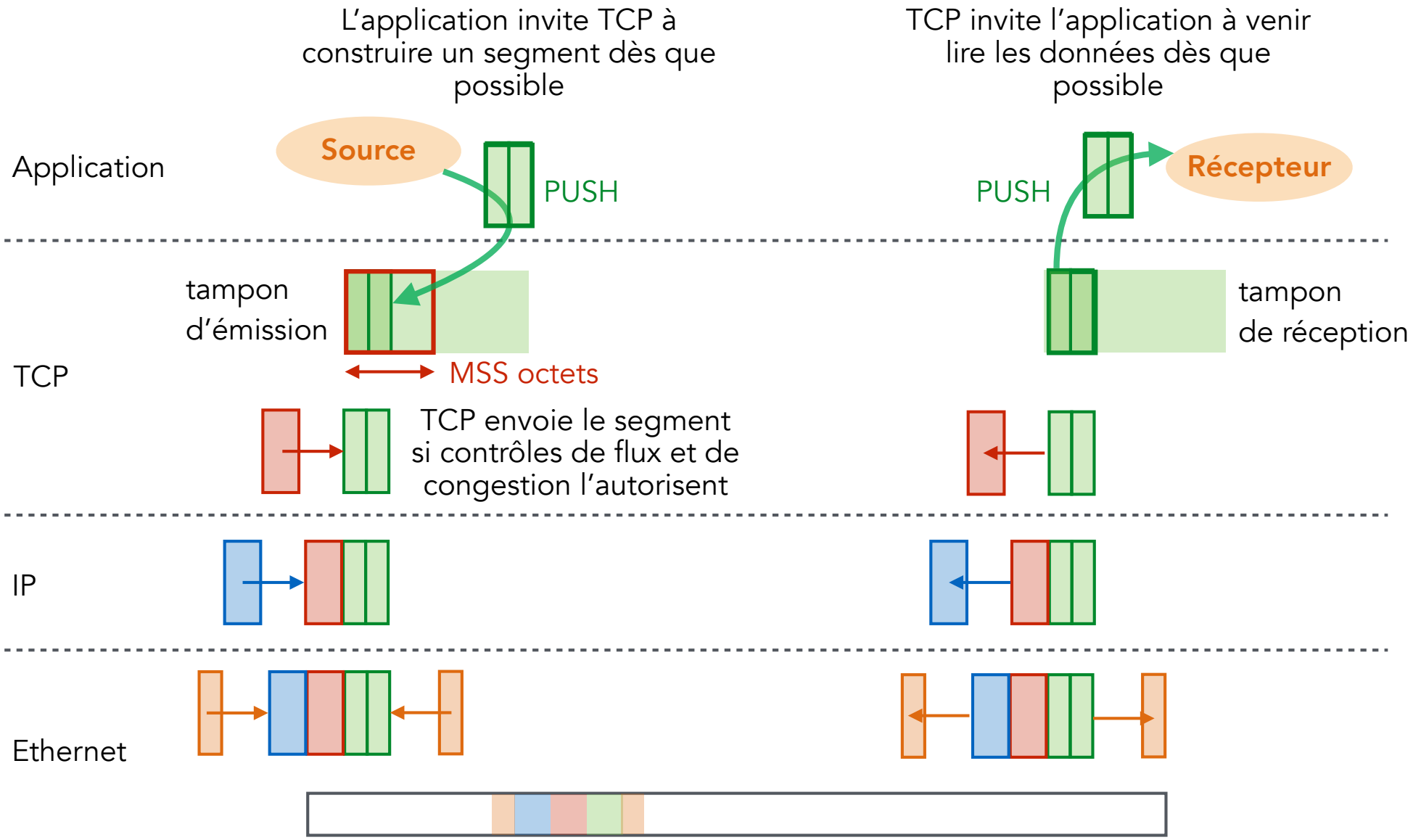
- TCP offre un mécanisme contrôle de congestion
  - permettant à l'émetteur d'adapter sa vitesse d'émission à la capacité de transfert du réseau
  - au travers d'une deuxième fenêtre d'émission appelée « fenêtre de congestion » (Congestion Window)
    - plage d'octets que l'émetteur à le droit d'émettre
    - afin ne pas saturer le réseau
  - calculée en fonction des segments de données envoyés et des segments perdus (sans ACK)
- La fenêtre d'émission utilisée par TCP est en réalité l'intersection des deux fenêtres
  - fenêtre d'émission = fenêtre de contrôle de flux ∩ fenêtre de contrôle de congestion

Drapeau PUSH

- Drapeau PUSH
  - côté émetteur
    - l'application invite TCP à construire un segment sans attendre d'octets supplémentaires
  - côté récepteur
    - l'application est invitée à lire les octets reçus dès que possible

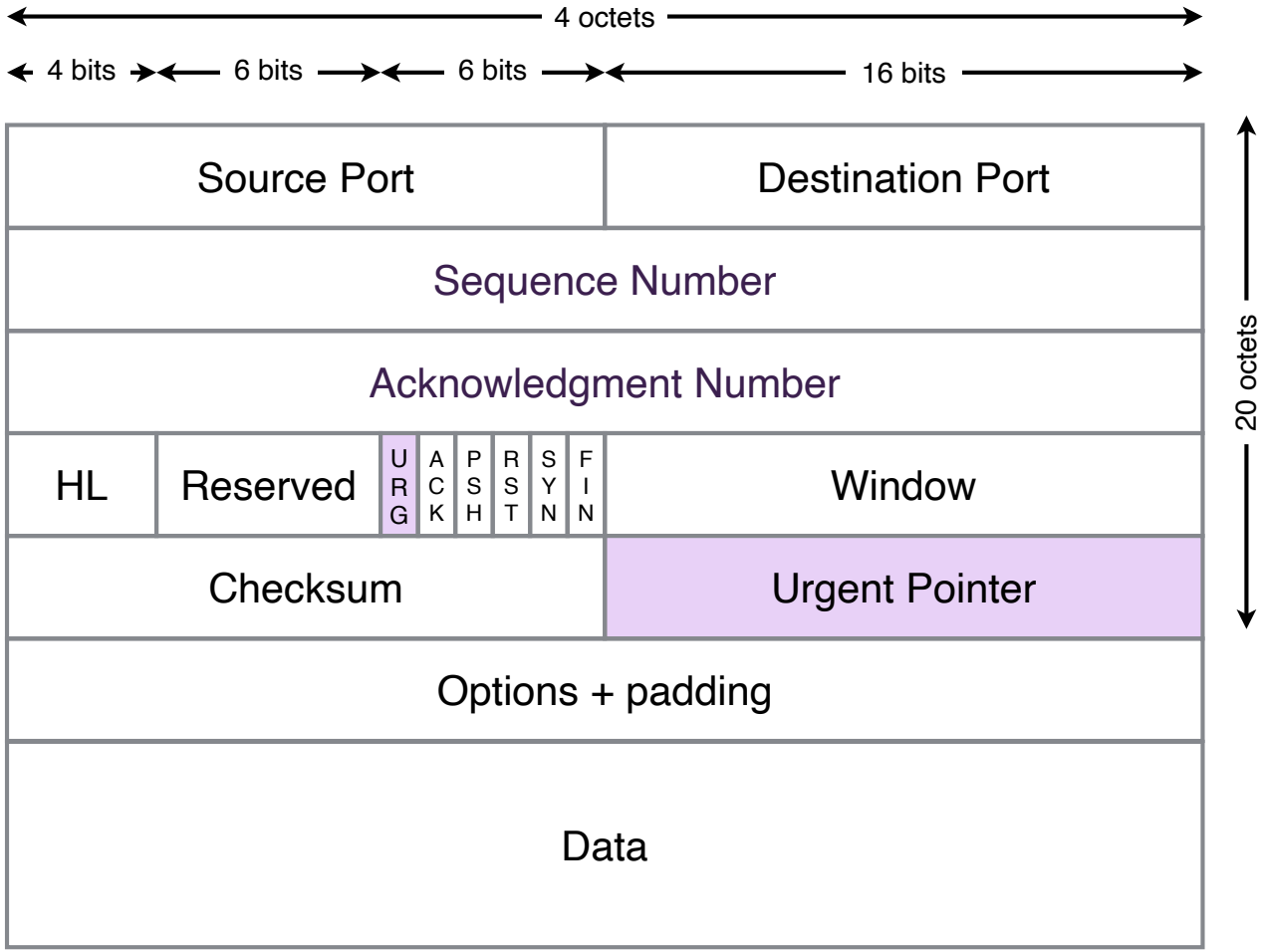


Drapeau PUSH

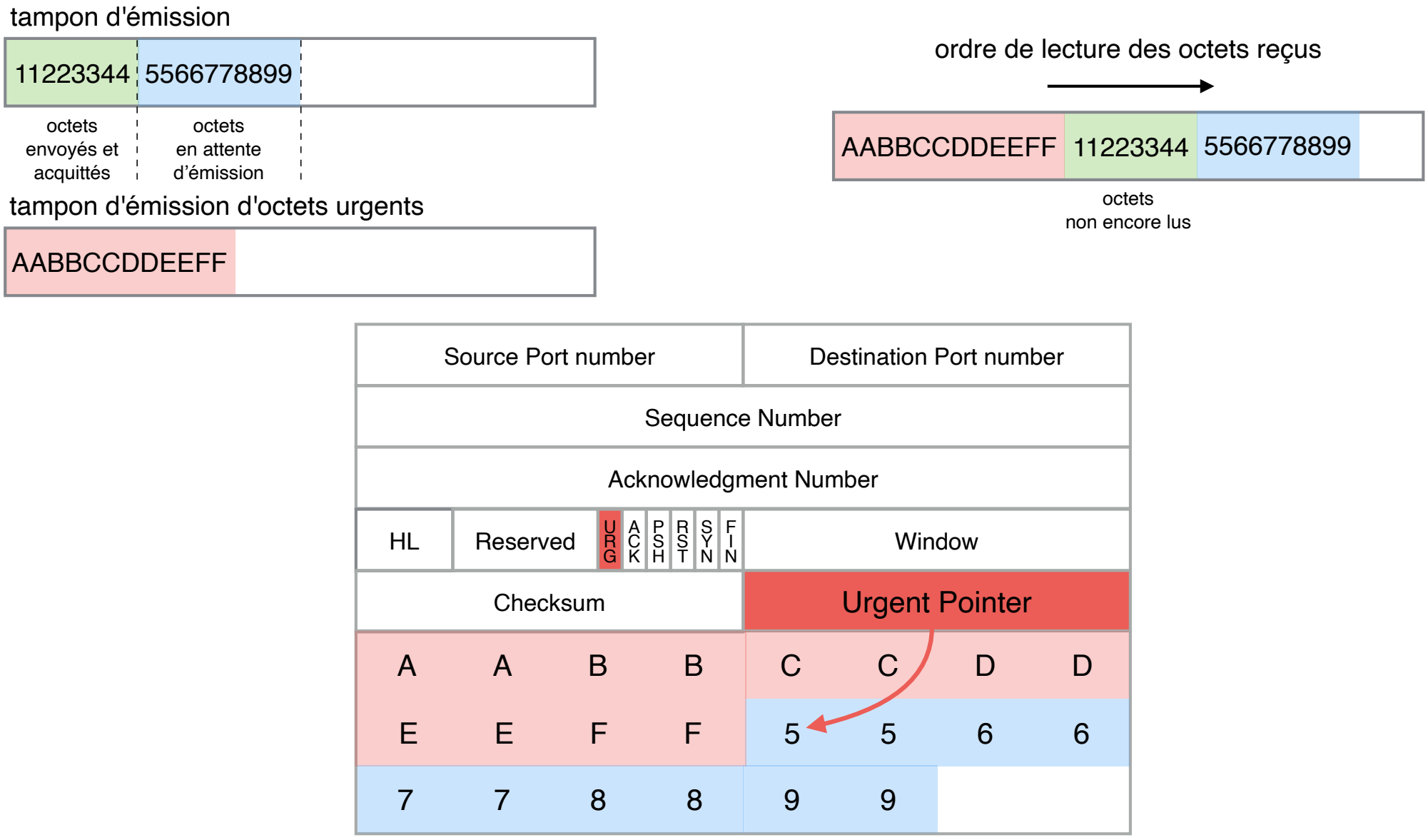


Drapeau URG

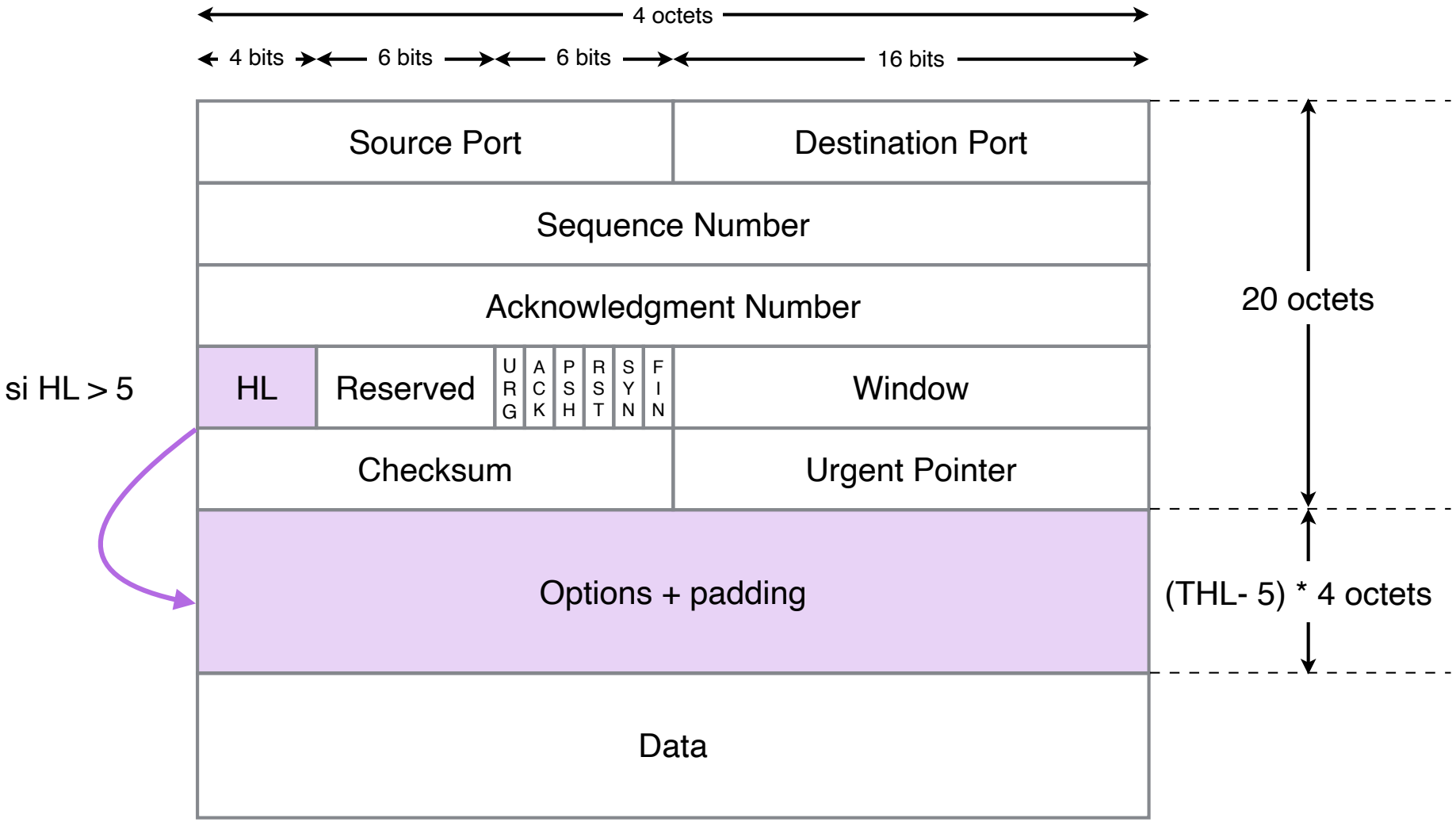
- Drapeau URG
  - indique que le segment contient des données urgentes (placées au début du champ de données) nécessitant un traitement prioritaire
- Champ *Urgent Pointer* (16 bits)
  - indique la fin des données urgentes
  - ces données sont lus en priorité par l'application réceptrice…
  - … sans attendre que les octets précédemment reçus ne soient lus



Champ Urgent Pointer



Options TCP

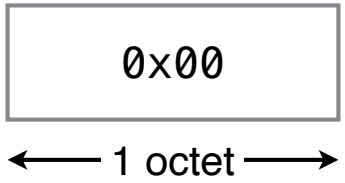

4 octets
4 bits
6 bits
6 bits
16 bits
Source Port
Destination Port
Sequence Number
Acknowledgment Number
HL
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options + padding
Data
si HL > 5
20 octets
(THL- 5) * 4 octets

Options TCP

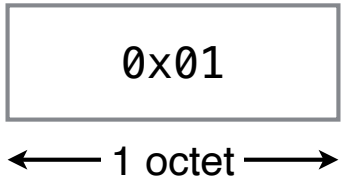
Kind	Length	Meaning	Reference
0x00	–	End of Option List	RFC 793
0x01	–	No-Operation	RFC 793
0x02	0x04	Maximum Segment Size	RFC 793
0x03	0x03	WSOPT - Window Scale	RFC 1323
0x04	0x02	SACK Permitted	RFC 2018
0x05	N	SACK (Selective ACK)	RFC 2018
0x06	0x06	Echo (obsoleted by option 8)	RFC 1072
0x07	0x06	Echo Reply (obsoleted by option 8)	RFC 1072
0x08	0x0A	TSOPT - Time Stamp Option	RFC 1323
0x09	0x02	Partial Order Connection Permitted	RFC 1693
0x0A	0x03	Partial Order Service Profile	RFC 1693
0x0B	–	CC	RFC 1644
0x0C	–	CC.NEW	RFC 1644
0x0D	–	CC.ECHO	RFC 1644
0x0E	0x03	TCP Alternate Checksum Request	RFC 1146
0x0F	N	TCP Alternate Checksum Data	RFC 1146

Options TCP

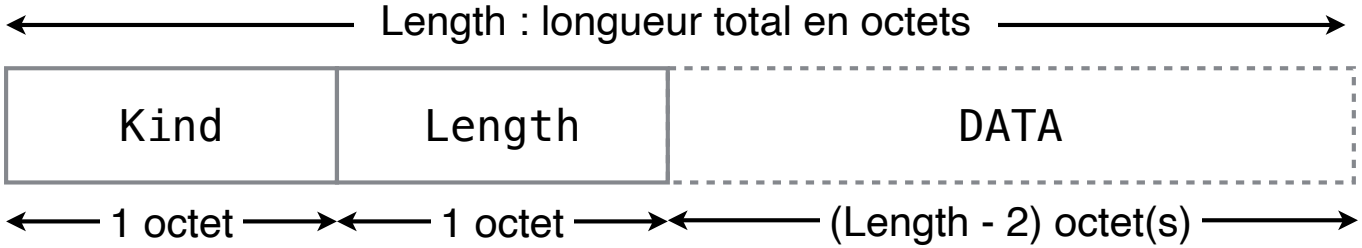
Format de l'option *End of Options List* (EOL) : type = 0



Format de l'option *No OPeration* (NOP) : type = 1

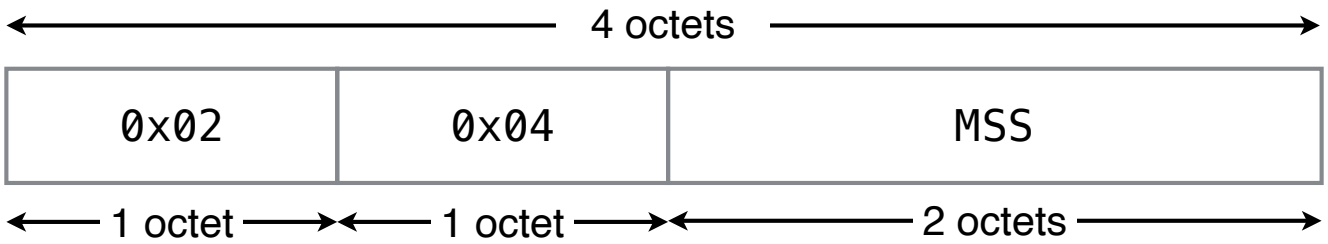


Format des options de type > 1

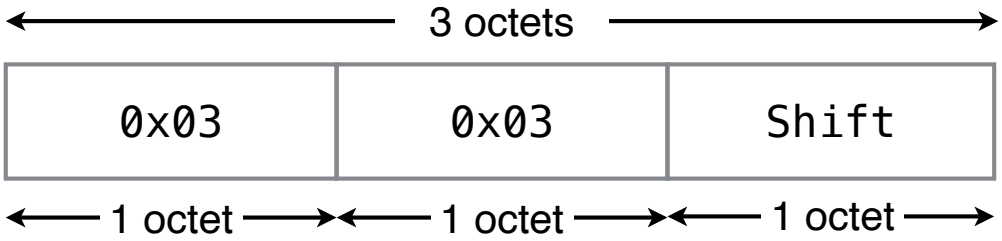


Options TCP

Maximum Segment Size (MSS) : type = 2

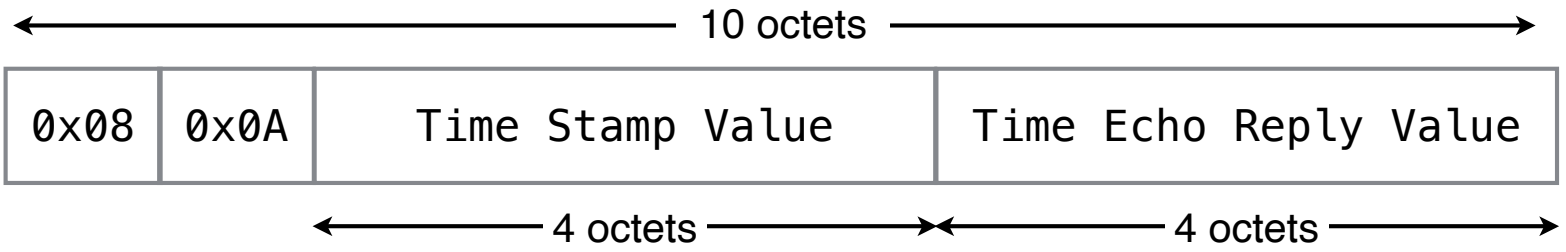


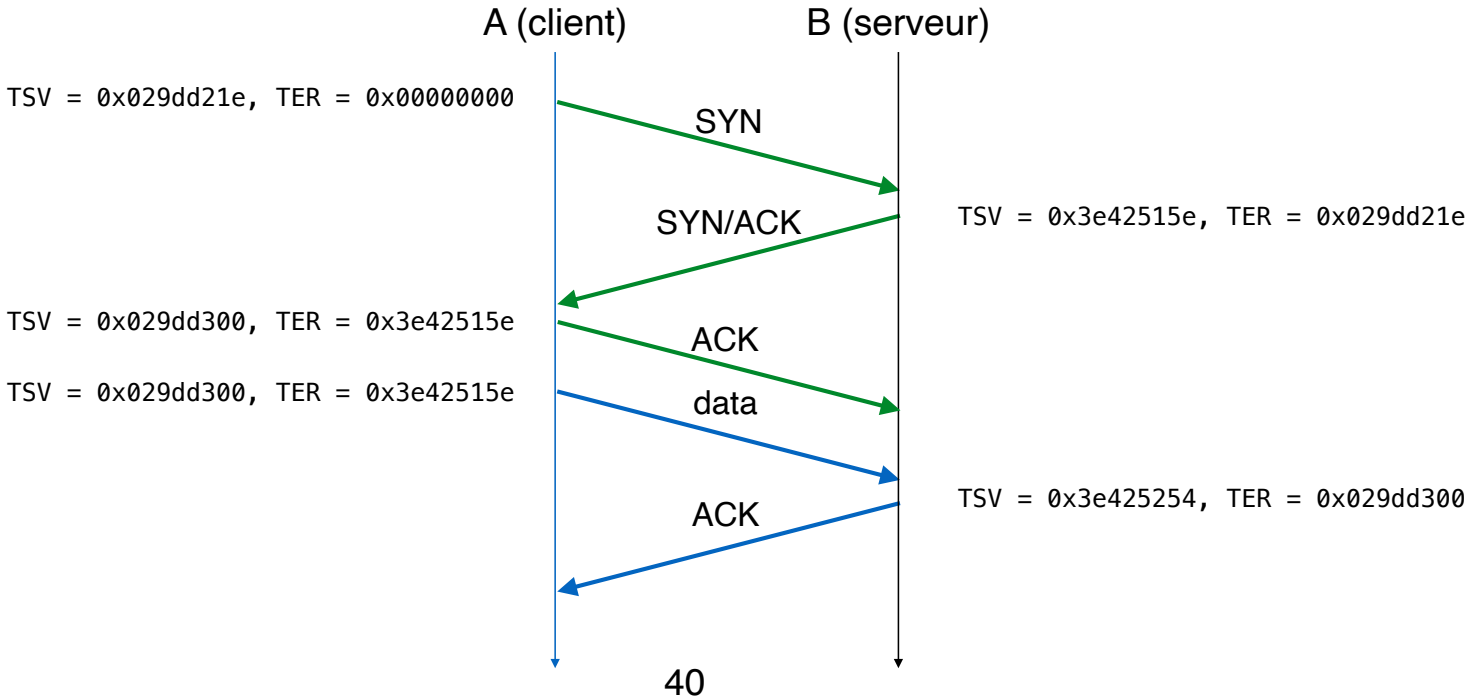
Windows Scale (WSopt) : type = 3



Options TCP

Time Stamp (TS) : type = 8





Conclusion

- TCP comme UDP
  - Multiplexage et démultiplexage
  - Détection d'erreur
    - somme de contrôle
- TCP à l'encontre d'UDP
  - Connexion et états
  - Réparation des pertes et correction d'erreur
    - numéros de séquence
    - estimation du RTT
    - expiration de temporisateur et retransmission
  - Contrôle de flux
    - fenêtre d'émission
  - Contrôle de congestion

A faire

- Cours 10
  - à relire attentivement
- Devoir 10 sur Moodle
  - date de rendu : dimanche 26 novembre