

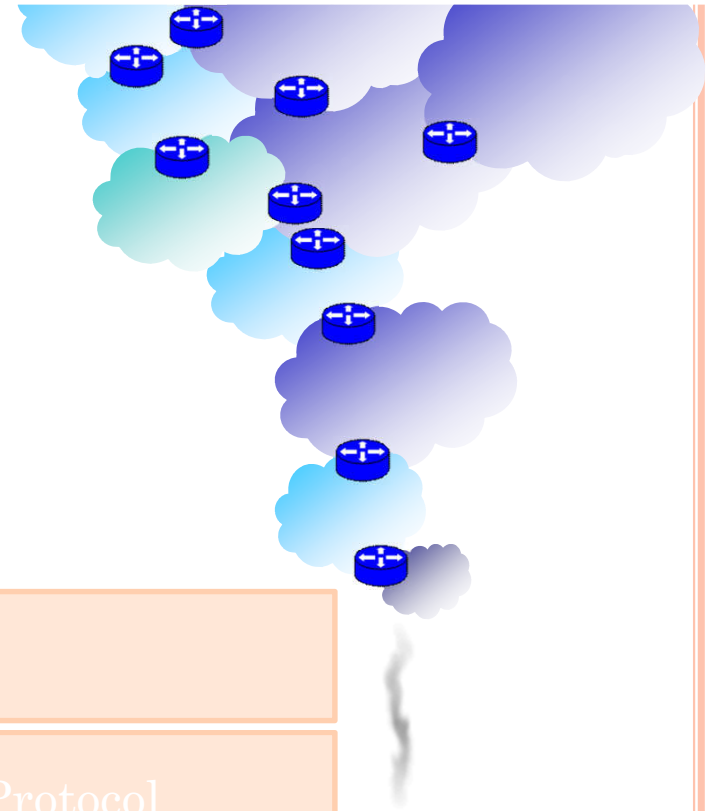
Chapitre I – Généralités

Chapitre II – Réseau : Internet Protocol

Chapitre III – Transport : TCP

Chapitre IV – Protocole de Routage : RIP

Chapitre V – De l'architecture d'Internet au
réseau domestique



OBJECTIF DE LA SÉANCE DE COURS

- Comprendre
 - Le principe d'un protocole à fenêtre
 - Celui proposé dans Internet: TCP
 - Protocole à fenêtre,
 - Connexion,
 - Contrôle de flux,
 - Reprise sur erreur,...
- En
 - Illustrant les mécanismes à fenêtre
 - Construisant l'en-tête TCP



PLAN CHAPITRE 3 – TRANSPORT: TCP

- Introduction : le besoin de transport
- 3.1 - De l'envoi d'un message au protocole à fenêtre
- 3.2 – TCP
 - Généralités
 - Construire l'en-tête
 - La numérotation
 - Notion de segment / Sequence number /Ack Number
 - Initial Sequence Number
 - La connexion
 - Ouverture /Fermeture
 - Le piggy backing
 - La fiabilisation
 - Le contrôle de flux

LE BESOIN DE TRANSPORT

DANS LE MONDE INTERNET

○ L'application

- A des caractéristiques variées
- Des besoins de transmission divers
- Mais souvent on peut retrouver les mêmes
 - Fiabilisation
 - Séquencement...

○ Le réseau

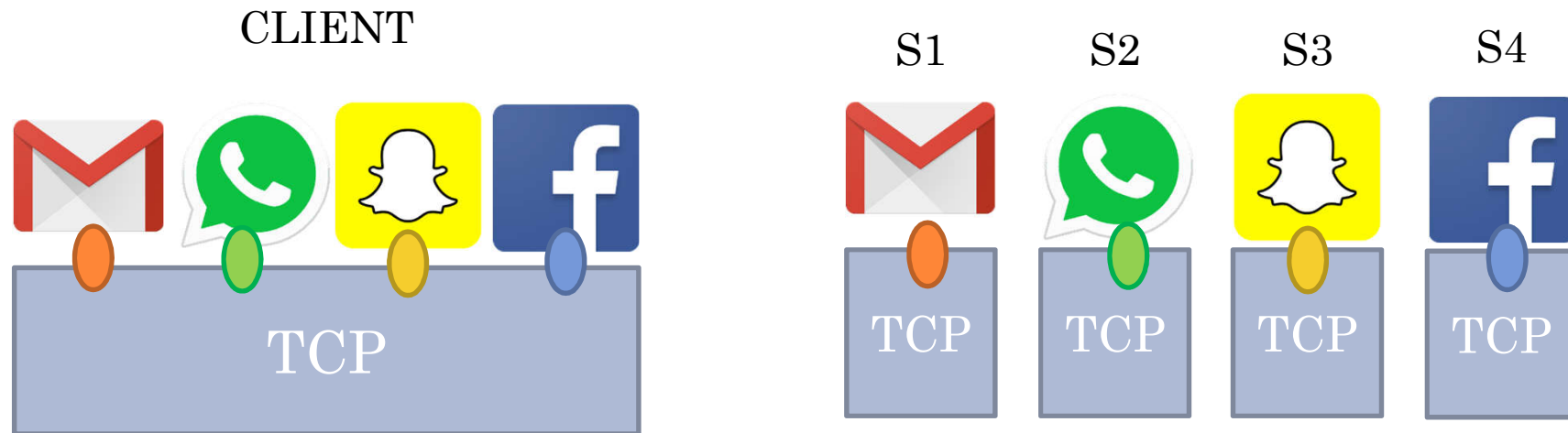
- Achemine des messages de A à B
- Pas parfaitement fiable
- Se veut simple et généraliste

○ Besoin d'un intermédiaire

- Mettre en place les concepts précédemment évoqués
 - Une fois pour toutes (ou presque)
- => Ne pas refaire tout pour chaque application

LE BESOIN DE TRANSPORT

LE LIEN AVEC L'APPLICATION



- Transport
 - TCP
 - UDP
- Multiplexage applicatif
 - Notion de port (2B): source et destination
 - Mode Client/Serveur
 - Socket : Adresse IP + Protocole de Transport + Port

3.1

DE L'ENVOI D'UN MESSAGE À SA BONNE RÉCEPTION

- Comment savoir si un message est bien reçu?

- Solution d'IP:
 - Ce n'est pas mon problème....
- La notion d'accusé de réception
 - ACK
 - ...

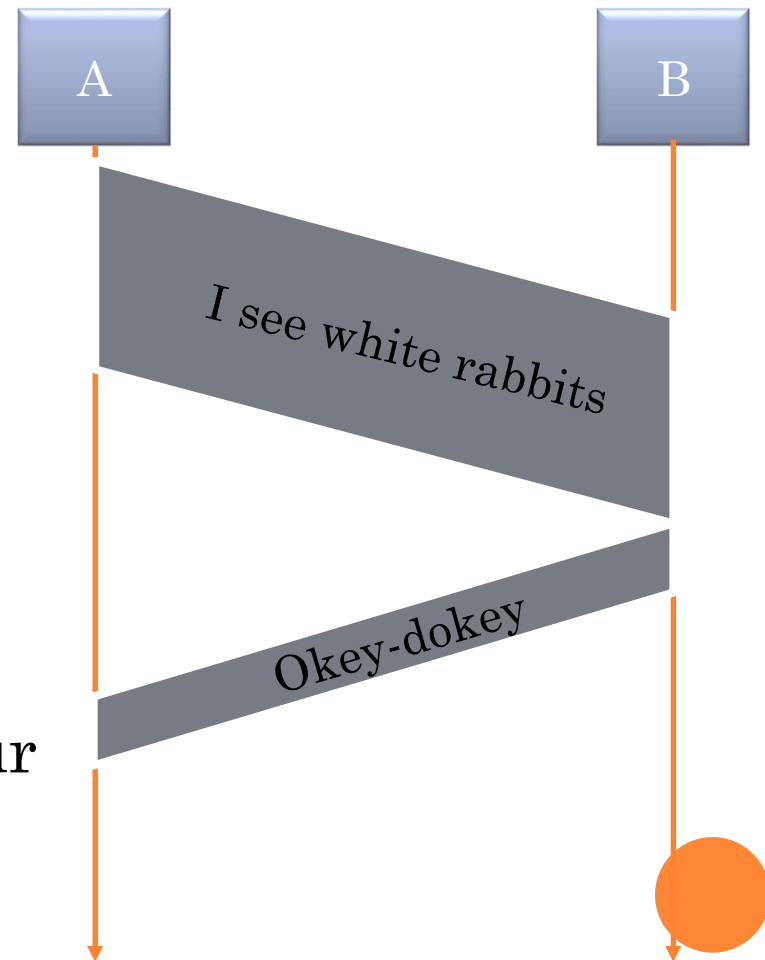
- Principe du SEND&WAIT

- Envoie d'un message
- Attente d'un ACK

- Mécanisme de reprise sur erreur

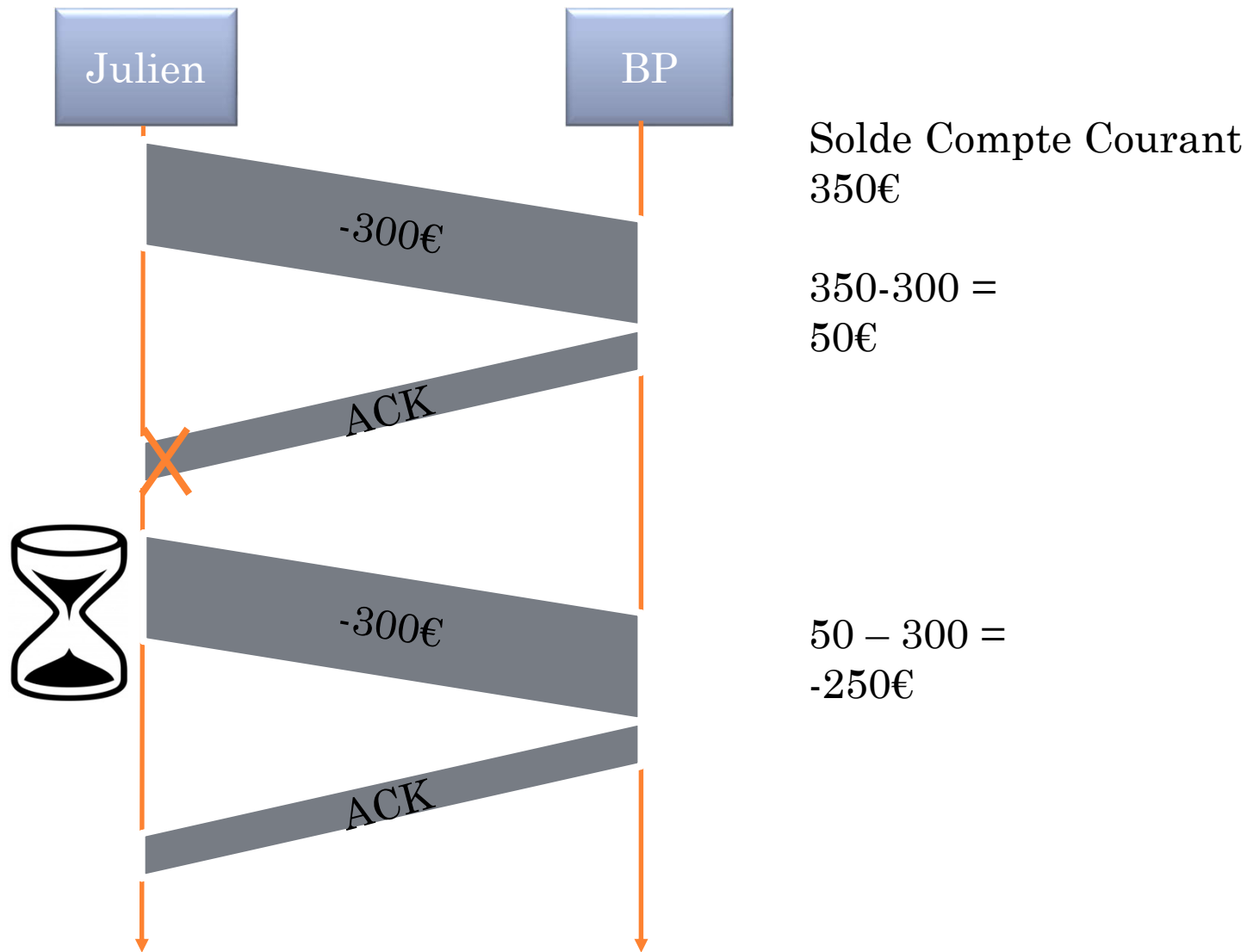
- Timer

- Mais n'y a-t-il pas des problèmes?



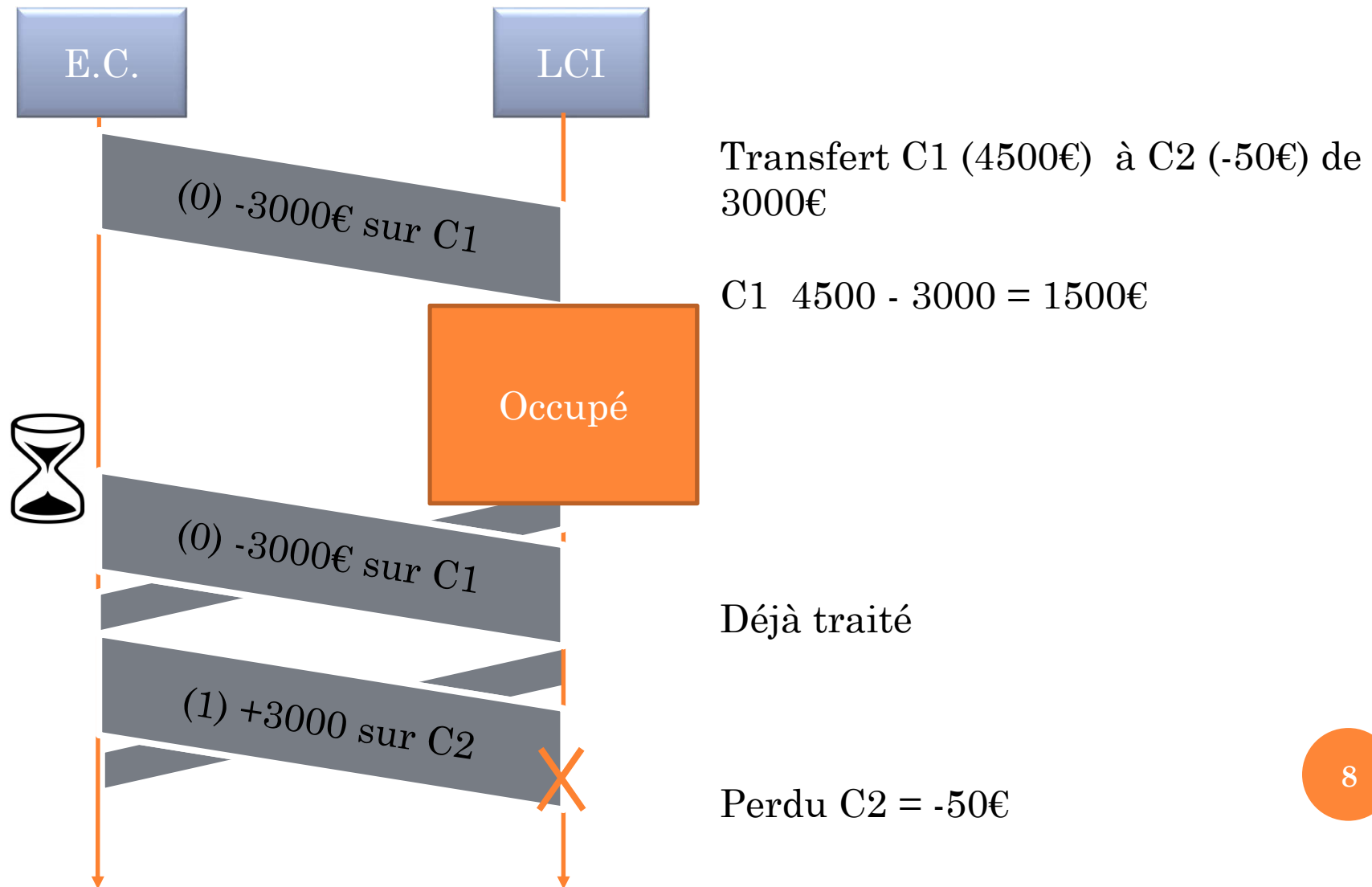
3.1 SEND & WAIT

POURQUOI NUMÉROTER LES MESSAGES?



3.1 SEND & WAIT

POURQUOI NUMÉROTER LES ACKs?



3.1 SEND & WAIT

EFFICACITÉ

- Quel est le problème d'efficacité?
 - Illustration satellite (cf cours 1)
- Comment le résoudre?
 - Augmenter le débit
 - Non, on augmente que le débit !
 - Raccourcir la distance ?
 - Bof... ce n'est pas souvent possible
 - Augmenter la taille des messages
 - Compromis, limite
 - Être plus malin ?
 - Notion de protocole à fenêtre

3.1 PROTOCOLE À FENÊTRE

- Idée de base
 - Anticiper les ACK en se donnant une « fenêtre » de taille n
 - Pour toujours fournir un service fiable
- Objectif
 - Arriver à émettre en permanence (efficacité de 1)
 - Calcul de la taille de fenêtre idéale

3.1 PROTOCOLE À FENÊTRE

DU CÔTÉ ÉMETTEUR

- Émetteur peut envoyer jusqu'à n messages sans ACK
 - De $[0 - n]$
 - Besoin de les mettre en mémoire = fenêtre d'émission
 - Comment fixer N ?

3.1 PROTOCOLE À FENÊTRE

DU CÔTÉ RÉCEPTEUR

- Que fait le récepteur?
 - Il reçoit les messages et il renvoie les ACKS
- Et s'il n'est pas près à traiter l'info?
 - Contrôle de flux
 - Il a besoin d'une mémoire = fenêtre de réception
 - Négociation d'une taille (n)
- Le plus simple : Contrôle de flux explicite
 - STOP (comme le RNR d'HDLC)
 - Changer n

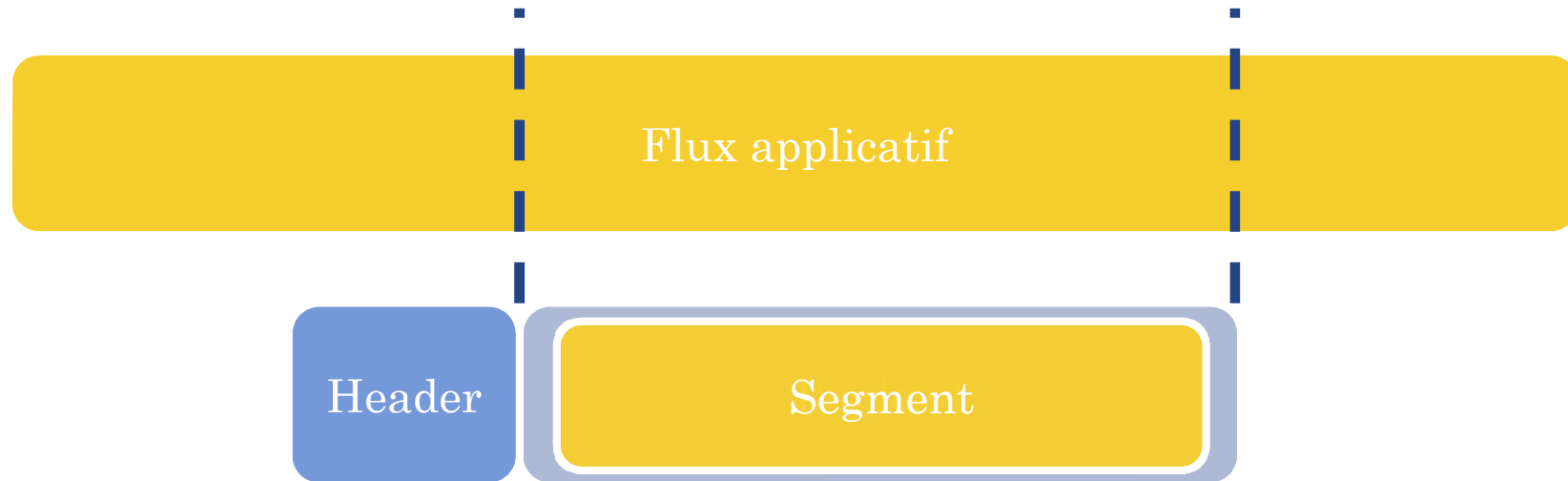
3.1 - PROTOCOLE À FENÊTRE

EN CAS DE PERTES

- Que se passe-t-il si le message 1 est perdu?
 - Non événement: on ne fait rien
 - On a de la mémoire
 - Le 2 et le 3 peuvent être stockés mais ils sont inutilisables
 - On attend le timer de retransmission
 - Quand on reçoit 2 que fait on?
 - Demander de retransmettre?
 - Ne rien faire de nouveau?
 - Emettre ACK de 2?
 - NON!
 - Quel intérêt?
 - Non utilité d'un message hors séquence
 - Pb de fenêtre et de place.
 - Idée importante d'ordre!

3.2 – LE FORMAT DU MESSAGE TCP

- Principe de l'encapsulation



- Format d'un message TCP

- Signalisation

- En début de message: En-tête / header
 - En fin de message :
 - Souvent du contrôle
 - Pas dans TCP

- Données

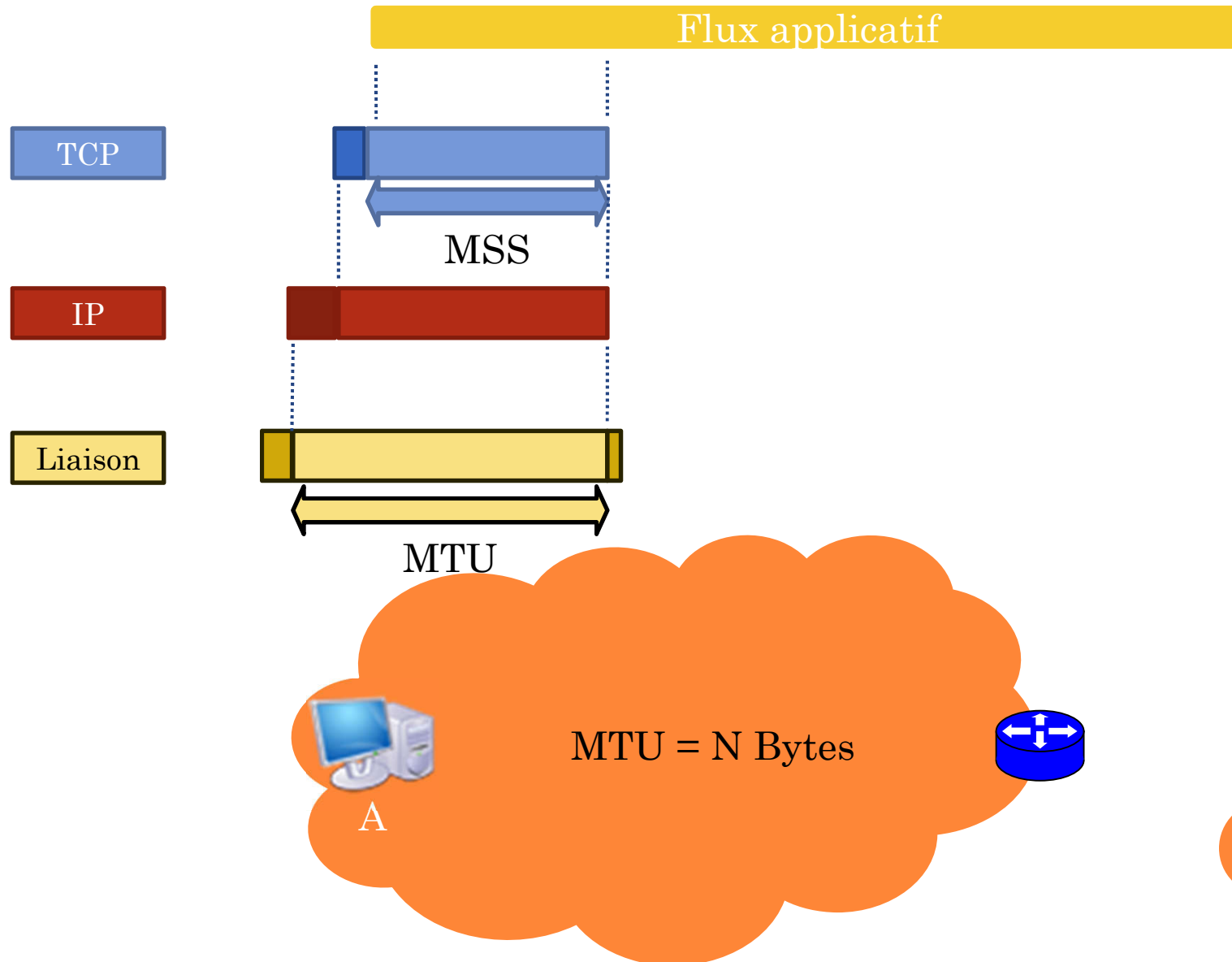
- Ici segment du flux applicatif
 - Données / Corps / Charge utile / Data / Payload

- Que mettre dans l'en-tête?

- Message TCP = Segment TCP

3.2 - MAXIMUM SEGMENT SIZE

DISCUSSION SUR LA SEGMENTATION



3.2 – LA NUMÉROTATION

- Notion de Segment
 - Messages applicatifs considérés comme un flux
 - TCP peut
 - les découper
 - les assembler
 - dans un même segment TCP
- Numérotation des segments
 - Octet par octet
 - Sequence Number (4B)
- Numérotation des accusés de réception
 - Ack Number (4B)

3.2 – LA CONNEXION

- Principe d'une connexion
 - Quels intérêts?
 - Que négocier?
 - Taille de la fenêtre d'émission/réception (Advertised Window, awnd, 2B)
 - Initial Sequence Number (ISN)
- Principaux états
 - Ouverture
 - En trois messages = triple handshake
 - SYN, SYN ACK, ACK
 - Etablie
 - On communique
 - Fermeture
 - Indépendance des deux clôture de connexion (FIN)

3.2 – LE PIGGYBACKING

- Signification
 - “to carry someone on your back”
- Objectif
 - Profiter de la communication de données B -> A
 - Pour envoyer les accusés de réceptions de la communication A -> B
- Comment?
 - Champ ACK toujours présent dans un segment TCP
 - Utilisation de timers

3.2 – DÉTECTION D'UN MESSAGE ERRONÉ

- Utilisation d'un champ de contrôle dans l'entête
 - Checksum
 - Principe du bit de parité, XOR, etc...
 - 2 Bytes
- Que faire si le Checksum et son calcul ne correspondent pas?

3.2 – LA FIABILISATION

- Utilisation d'un timer de retransmission
 - Retransmission Time Out (RTO)
- Comment l'obtenir?
 - Basé sur le temps d'aller – retour
 - Round Trip Time
 - **Comment le calculer?**
- Que faire quand il est écoulé?
 - Retransmission du premier de la fenêtre
 - A la réception d'un ACK, on considère que ce qui reste dans la fenêtre d'émission a été perdu

3.2 – LE CONTRÔLE DE FLUX

- Comment faire quand le récepteur ne peut pas consommer les données?
- Manipulation de la taille de la fenêtre de réception annoncée (awnd)
 - Réduction progressive à zéro

CONCLUSION

LES CHAMPS DE L'EN-TÊTE TCP

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Source Port | | | | | | | | | | | | | | | | Destination Port | | | | | | | | | | | | | | | |
| Sequence Number | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ack Number | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Header Length | | Reserved | | | | | | | | U | A | P | R | S | F | Window Size | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | | Urgent Pointer | | | | | | | | | | | | | | | |
| Option ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

NOTIONS À RETENIR SUR CHAPITRE 3

TCP

- Transport dans Internet
 - Encapsulation
 - Port (multiplexage applicatif)
- TCP
 - Numérotation en octets (Segments, Acks et MSS)
 - Orienté connexion (SYN, FIN)
 - Mécanisme à fenêtre glissante (awnd, RTT, RTO)
 - Fiabilisation
 - Contrôle de flux
 - Piggybacking
- **TD : chronogramme TCP**