Gilles.Roussel©univ-mlv.fr

TCP

TCP

- *Transmission Control Protocol*
  - Au-dessus d'une couche réseau non fiable en commutation par paquets
  - Mode connecté
  - Flot de données non structuré
  - Fiable
  - Full-duplex
  - Contrôle de flot et mise en tampon

TCP

- Notion de port comme UDP
- Implantation beaucoup plus complexe
- 90% du trafic de l'Internet
- Protocole « poli »
- Protocole réactif plutôt que prédictif

Évolution

- 1970 *Network Control Program*
- 1983 TCP BSD
- 1988 Tahoe : *slow start* et *congestion avoidance*
- **1990 Reno : *fast retransmission and recovery***
- 1994 Vegas : *congestion avoidance* (pas totalement adopté)
- 1996 SACK extension
- 1999 New-Reno : amélioration de Reno
- ... TCP WestWood, Fast TCP, etc.

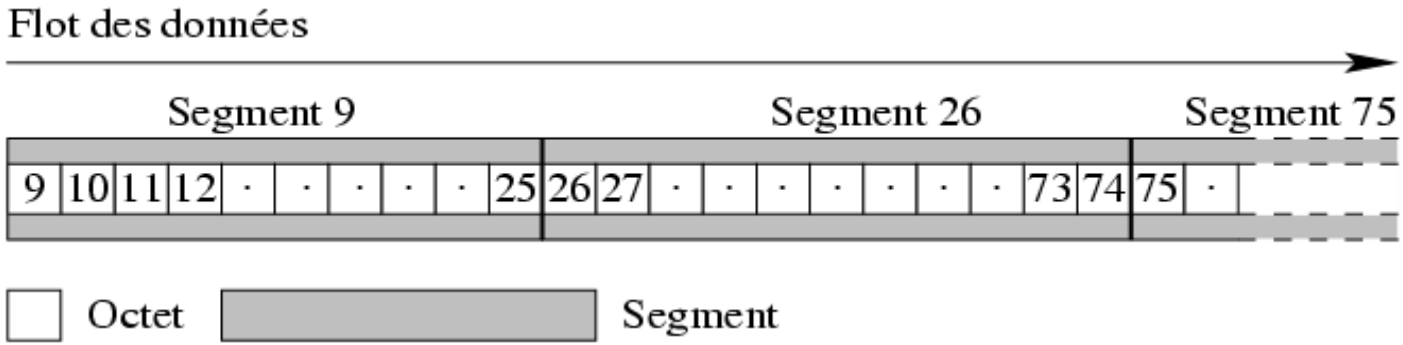
Le service de fiabilité

- Acquittement positif avec retransmission
- Émetteur démarre une alarme à chaque envoi de segment
  - si alarme expire avant l'arrivée d'un acquittement
    - retransmission des données du segment
  - sinon effacement du segment

Le service de fiabilité

- Chaque octet émis à un numéro de séquence
  - Numéro initial choisi à la connexion
- Chaque segment a le numéro du premier octet qu'il contient
- Acquittement contient numéro de séquence strictement supérieur à tous les octets déjà reçus
  - Acquitte tous les octets de numéro inférieur

Le service de fiabilité



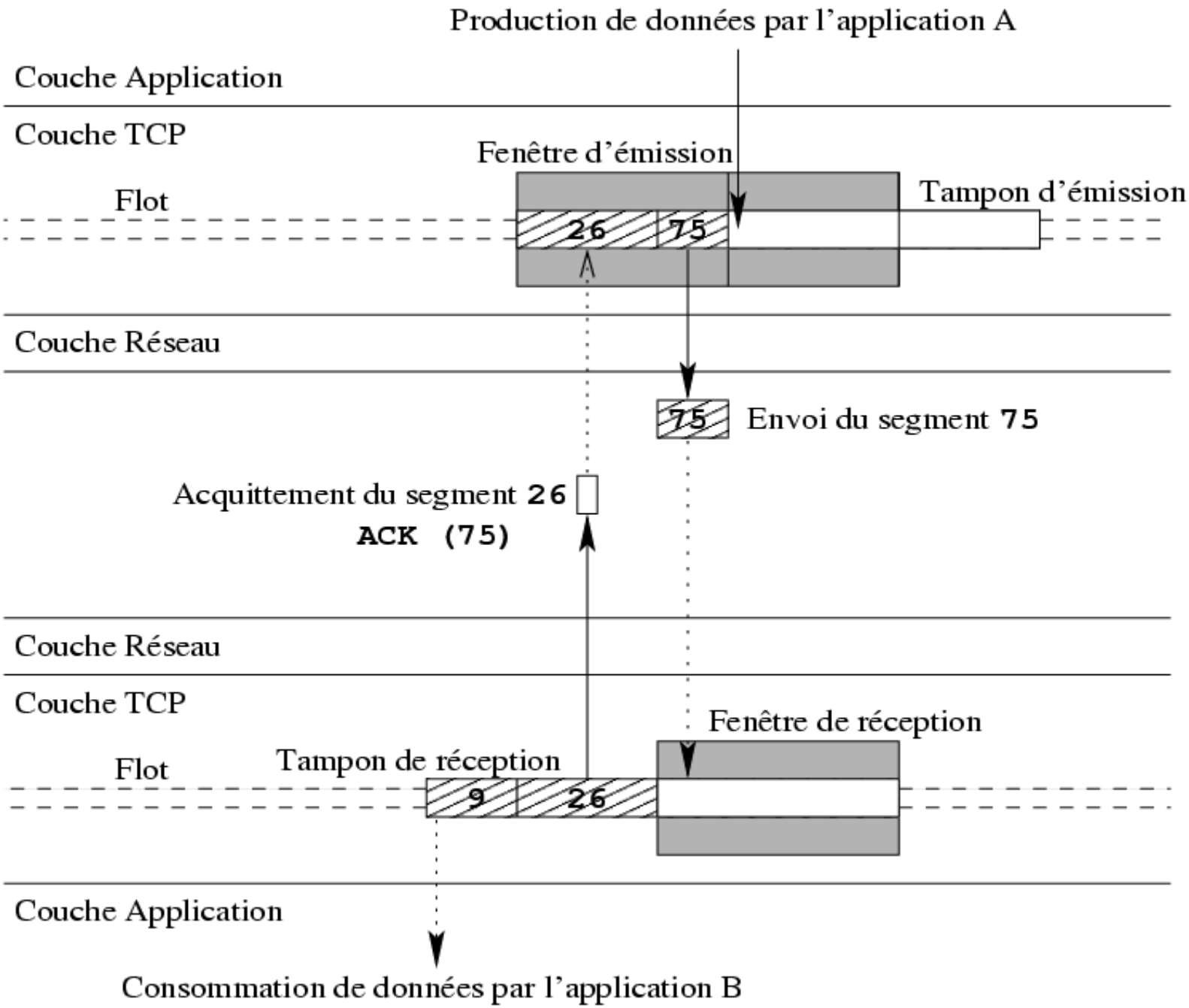
- Acquittement d'un segment allant de A vers B est véhiculé par un segment allant de B vers A (éventuellement sans données)
- Technique appelée *piggybacking* (porter sur le dos)

Fenêtre coulissante ou *Sliding Window*

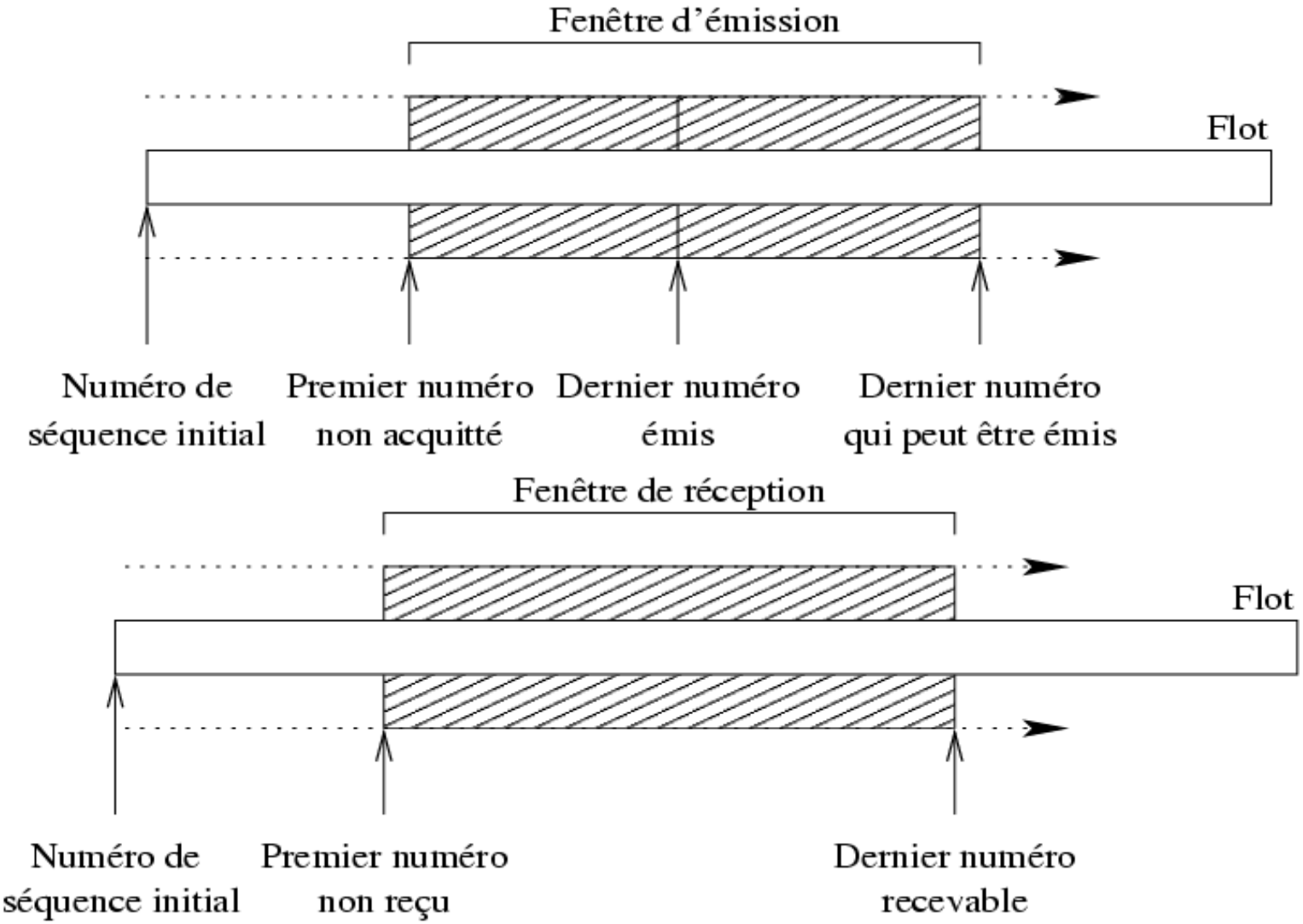
- Amélioration des performances
  - Ne pas attendre d'avoir reçu l'acquittement du segment émis pour envoyer le suivant
- Pour ne pas émettre des données pour rien = contrôle de flot
  - Connaître la place disponible dans le tampon du récepteur
    - Signalisation par *piggybacking*

Fenêtre coulissante

- Fenêtre d'émission déduite :
  - du numéro d'acquittement (prochain octet attendu)
  - de la taille de la fenêtre de réception
- Deux fenêtres coulissantes par sens de communication


Production de données par l'application A
Couche Application
Couche TCP
Fenêtre d'émission
Flot
Tampon d'émission
26
75
Couche Réseau
75
Envoi du segment 75
Acquittement du segment 26
ACK (75)
Couche Réseau
Couche TCP
Fenêtre de réception
Flot
Tampon de réception
9
26
Couche Application
Consommation de données par l'application B

Performance



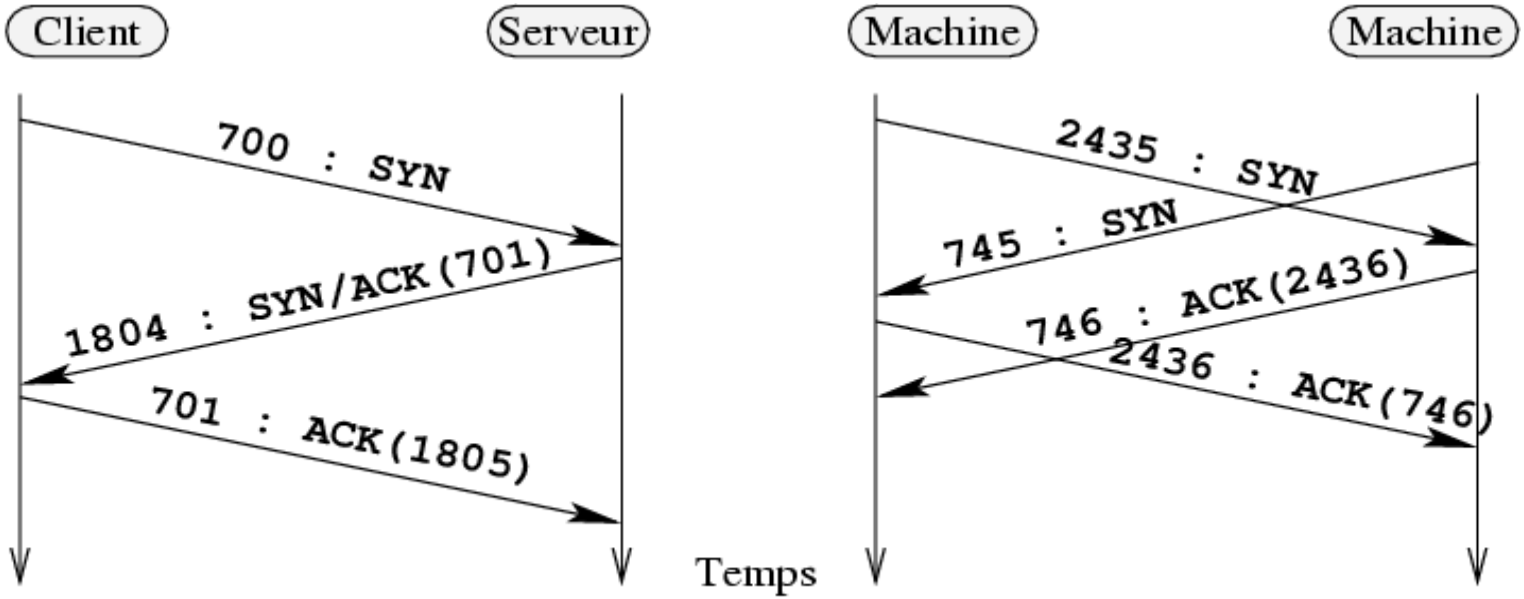
Fenêtre coulissante

- Comportement normal
  - **SEND_UNA**: coulisse en fonction des acquittements qui arrivent et **SEND_WNDW** diminue
  - **SEND_NEXT**: coulisse en fonction des segments émis
  - **RECV_NEXT**: coulisse en fonction des segments reçus et **RECV_WNDW** diminue
  - **RECV_WNDW**: augmente lorsque l'application récupère les données reçues
  - **SEND_WNDW**: augmente quand l'émetteur est informé que **RECV_WNDW** a augmentée

Fenêtre coulissante

- Contrôle de flot :
  - Si SEND_NEXT - SEND_UNA == SEND_WNDW alors émission stoppée
- Taille du tampon d'émission n'est pas corrélée à la taille de la fenêtre d'émission

Connexion



Connexion client–serveur

Connexion simultanée

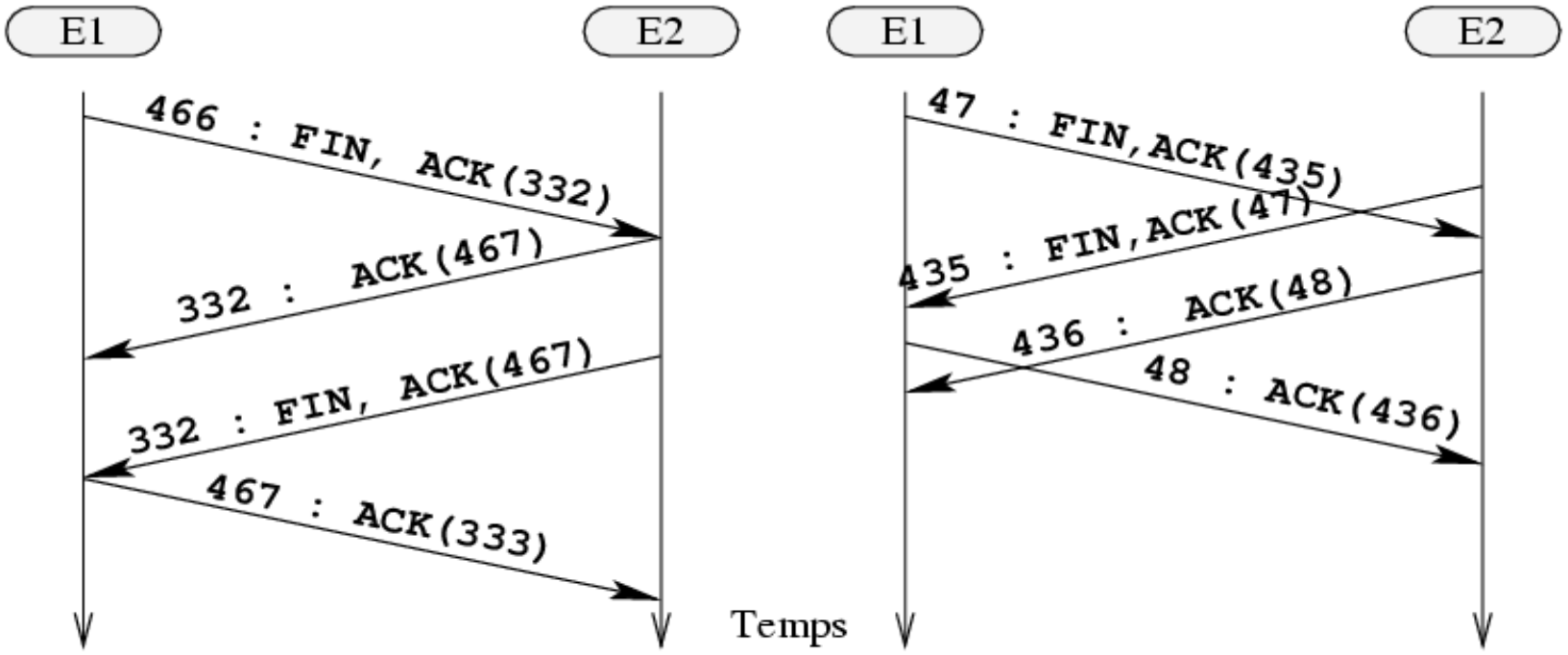
Connexion : extensions

- Échange des valeurs de MTU
  - Permet de déterminer les valeurs initiales du *Maximum Segment Size* (MSS) de chaque sens de communication
    - MSS modifié par *Path MTU Discovery*
- Négociation de l'horodatage des segments
  - Utilisé pour la mesure du temps d'aller-retour (*Round Trip Time* Mesure - RTTM) et la protection contre le rebouclage des numéros de séquences (Protection Against Wrapping Sequence - PAWS)

Connexion : extensions

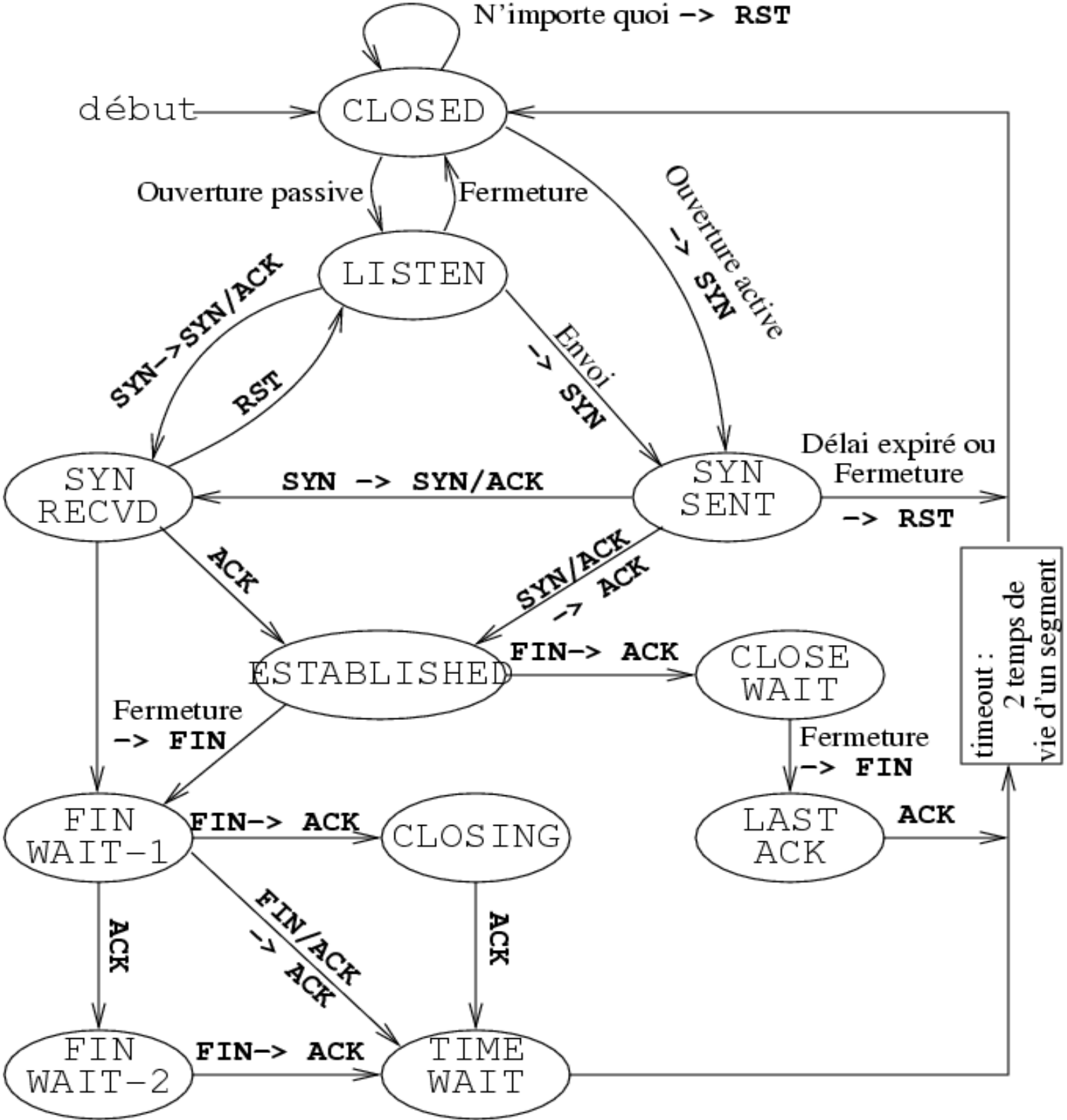
- Négociation de l'utilisation d'un facteur multiplicatif pour les tailles de fenêtre (de 2^0 à 2^{16})
  - Permet de passer de fenêtre de tailles maximale 2^{16} à des fenêtre de taille maximale 2^{30}
- Négociation de l'utilisation d'acquittements sélectifs (SACK)

Déconnexion



Déconnexion demandée par E1 puis par E2

Déconnexion demandée par E1 et E2


N'importe quoi -> RST
début
CLOSED
Ouverture passive
Fermeture
Ouverture active
-> SYN
LISTEN
SYN->SYN/ACK
RST
Envoi
-> SYN
SYN
RECVD
SYN -> SYN/ACK
SYN
SENT
Délai expiré ou
Fermeture
-> RST
ACK
SYN/ACK
-> ACK
ESTABLISHED
FIN-> ACK
CLOSE
WAIT
timeout :
2 temps de
vie d'un segment
Fermeture
-> FIN
Fermeture
-> FIN
FIN
WAIT-1
FIN-> ACK
CLOSING
LAST
ACK
ACK
ACK
FIN/ACK
-> ACK
ACK
FIN
WAIT-2
FIN-> ACK
TIME
WAIT

Alarme

- Déterminer le valeur de l'alarme
- Utilisation du temps d'aller-retour moyen
  - Jacobson : calcul de proche en proche s'il n'y a pas d'expiration
    - SampleRTT mesuré
    - EstimateRTT = (1-α) EstimateRTT + α SampleRTT
    - DevRTT = (1-β) DevRTT + β |SampleRTT -EstimateRTT|
- Timeout = EstimateRTT + 4 DevRTT

Gestion des « pertes »

- Temps d'aller-retour pas pris en compte
  - Algorithme de Karn : alarme augmentée par facteur multiplicatif en cas d'expiration
    - Augmentation exponentielle de l'alarme
  - Reprise du comportement normal à l'arrivée d'un acquittement

Contrôle de congestion

- Éviter les congestions au sein du réseau
- Utilisation d'une fenêtre de congestion (cwnd) qui contraint la fenêtre d'émission
- Au démarrage pour ne pas submerger le réseau
  - Utilisation de l'algorithme *Slow Start*

Algorithme *Slow Start*

- cwnd initialisée à 1 MSS
  - Éventuellement 2 à 4
- Accroissement de 1 MSS à chaque acquittement reçu
  - Taille de cwnd double à chaque RTT
    - Accroissement exponentiel

Contrôle de congestion

- En cas de perte de segment
- Détectée par :
  - Expiration d'une alarme
    - Perte d'un segment ou plusieurs sans suivant
  - Réception de plus de deux acquittements (acquittement négatif)
    - Perte d'un segment intermédiaire et réception d'un segment suivant

Expiration d'une alarme

- Redémarre en *Slow Start*
- Mise en place d'un seuil = moitié de la taille de la fenêtre de réception au moment de la congestion
- Évitement de congestion quand seuil atteint
  - Accroissement de 1 MSS à chaque RTT
    - Accroissement linéaire

Perte d'un segment isolé

- Retransmission rapide avant expiration
- *Fast recovery*
  - cwnd = seuil + 3 MSS
  - Redémarrage en évitement de congestion
    - Pas de *Slow Start*

Acquittement sélectif

- Éviter d'émettre les paquets déjà reçus en cas de perte de paquets isolés
- Options SACK-permitted dans segment SYN
- Option SACK permet d'acquitter les segments déjà reçus

Améliorer les performances

- Favoriser les pertes de paquet isolé pour réduire le flux sans le stopper
- Mise en place de *Random Early Detection*
  - Élimination aléatoire de segment si encombrement du tampon atteint un certain seuil
    - Plus le tampon est plein plus la probabilité de perte est forte

Marquer plutôt que supprimer

- Éviter la suppression de segments par RED
- Extension *Explicit Congestion Notification*
  - ECT bit (IP) indique que l'émetteur est compatible ECN
  - CE bit (IP) utilisé par le routeur pour marquer les segments en cas de congestion
  - Drapeau ECN-Echo (TCP) placé par le récepteur
  - Drapeau CWR (TCP) placé par l'émetteur pour indiqué qu'il a mis en place une fenêtre de congestion

Gestion des sessions interactives

- Transmettre les données dès qu'elles sont disponibles
  - Multiplication de petits paquets
- Algorithme de Nagle
  - L'émetteur peut transmettre un segment uniquement quand :
    - il a atteint la MSS
    - tous les segments précédents ont été acquittés
  - Entraîne des variations de délai inter-trames (gigue)

Format

0	*4*	*8*	*16*	*24*
Port source			Port destination	
Numéro de séquence				
Numéro d'acquittement				
T. entête	Rés./ECN	Drapeaux	Taille fenêtre réception	
Somme de contrôle			Pointeur urgent	
Options				Bourrage
Données				
...				

Format

- Drapeaux:
  - 0x20 `URG` contient un message urgent
  - 0x10 `ACK` contient un acquittement (tous sauf SYN)
  - 0x08 `PSH` indication que la transmission a été forcée (session interactive)
  - 0x04 `RST` indicateur de réinitialisation de la connexion
  - 0x02 `SYN` indication de début de communication
  - 0x01 `FIN` indicateur de fin d'envoi de l'émetteur

Données urgentes

- Drapeau URG permet d'indiquer que des données urgentes sont en attente
- Pointeur urgent donne le numéro de séquence du premier octet urgent