Le protocole TCP

Info 0403

TCP : Transmission Control Protocol

transport fiable de la technologie TCP/IP.

- fiabilité = illusion assurée par le service
- transferts tamponés : découpage en segments
- connexions bidirectionnelles et simultanées

- service en mode connecté
- garantie de non perte de messages ainsi que de l'ordonnancement

TCP : La connexion

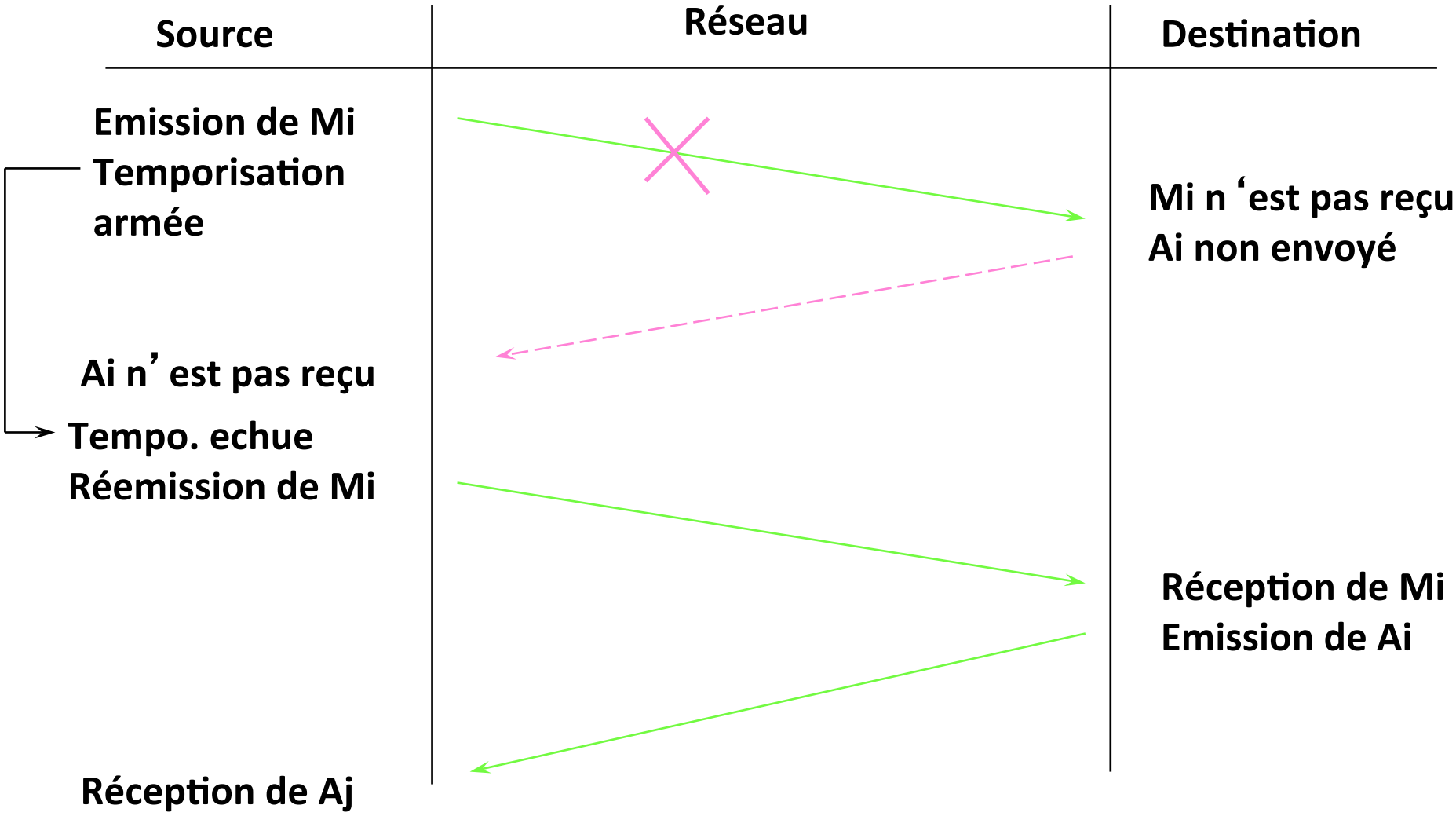
- une connexion de type circuit virtuel est établie avant que les données ne soient échangées : appel + négociation + transferts

- Une connexion = une paire d'extrémités de connexion
- Une extrémité de connexion = couple (adresse IP, port)
- Exemple de connexion : ((124.32.12.1, 1034), (19.24.67.2, 21))
- Une extrémité de connexion peut être partagée par plusieurs autres extrémités de connexions (multi-instanciation)
- La mise en oeuvre de la connexion se fait en deux étapes :
  - une application (extrémité) effectue une ouverture passive en indiquant qu'elle accepte une connexion entrante,
  - une autre application (extrémité) effectue une ouverture active pour demander l'établissement de la connexion.

TCP : Segmentation

- Segmentation, contrôle de flux
  - Les données transmises à TCP constituent un flot d'octets de longueur variable.
  - TCP divise ce flot de données en segments en utilisant un mécanisme de fenêtrage.
  - Un segment est émis dans un datagramme IP.
- Acquittement de messages
  - Contrairement à UDP, TCP garantit l'arrivée des messages, c'est à dire qu'en cas de perte, les deux extrémités sont prévenues.
  - Ce concept repose sur les techniques d'acquittement de message : lorsqu'une source S émet un message Mi vers une destination D, S attend un acquittement Ai de D avant d'émettre le message suivant Mi +1.
  - Si l'acquittement Ai ne parvient pas à S, S considère au bout d'un certain temps que le message est perdu et reémet Mi :

TCP : Acquittements

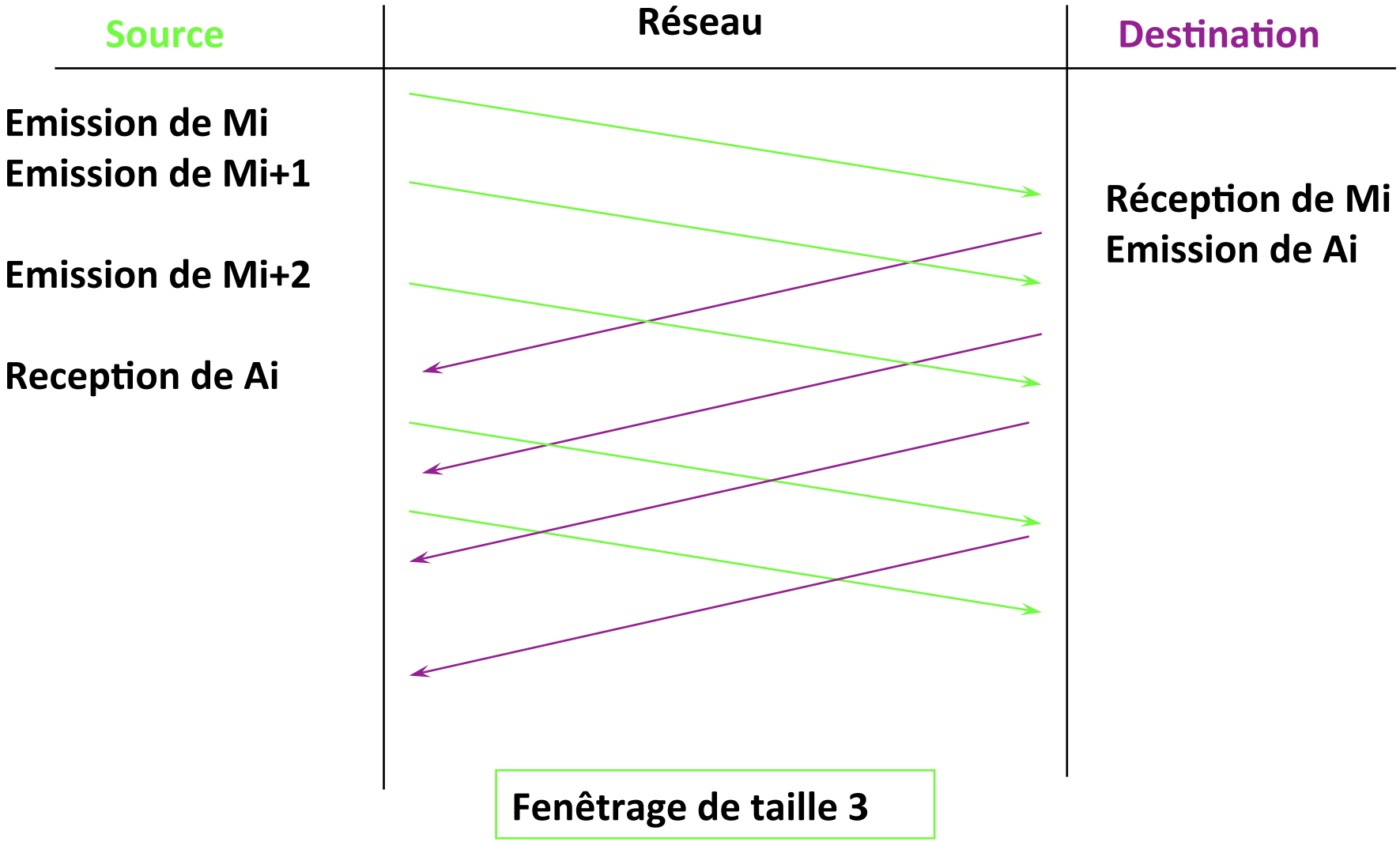


TCP : le fenêtrage

- La technique acquittement simple pénalise les performances puisqu'il faut attendre un acquittement avant d'émettre un nouveau message. Le fenêtrage améliore le rendement des réseaux.

- La technique du fenêtrage : une fenêtre de taille T, permet l'émission d'au plus T messages "*non acquittés*" avant de ne plus pouvoir émettre :

TCP : le Fenêtrage



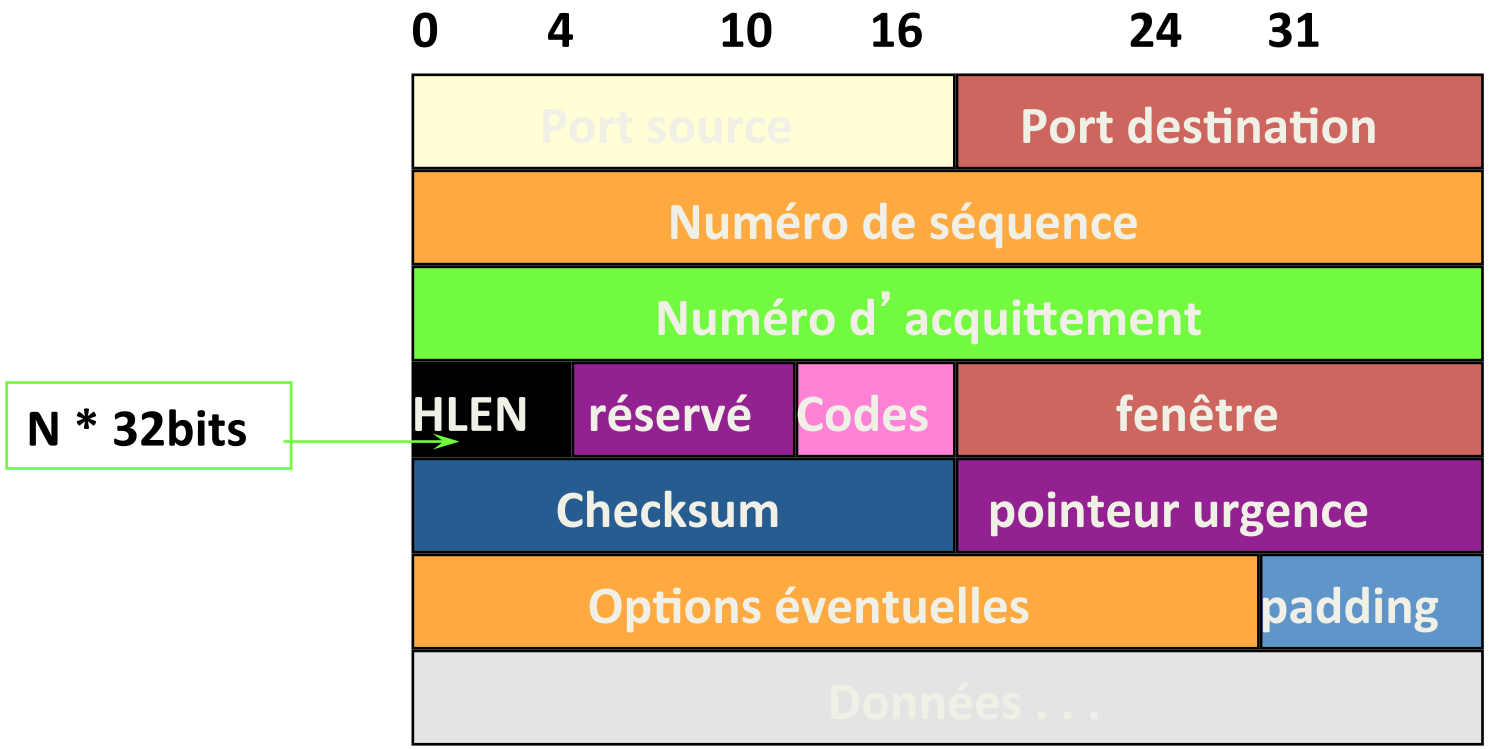
TCP : Technique de fenêtrage

- fenêtrage glissante permettant d'optimiser la bande passante
- permet également au destinatairede faire diminuer le débit de l'émetteur donc de gérer le contrôle de flux.
- Le mécanisme de fenêtrage mis en oeuvre dans TCP opère au niveau de l'octet et non pas au niveau du segment; il repose sur :

  – la numérotation séquentielle des octets de données,
  – la gestion de trois pointeurs par fenêtrage :

TCP : Segments

- Segment : unité de transfert du protocole TCP.
  - échangés pour établir les connexions,
  - transférer les données,
  - émettre des acquittements,
  - fermer les connexions;



TCP : format du segment

- Numéro de séquence : le numéro de séquence du premier octet (NSP) de ce segment. Généralement à la suite d'octets O1, O2, ..., On (données du message) est associée la suite de numéro de séquence NSP, NSP+1, ..., NSP+n.

  Il existe deux exceptions à cette règle :

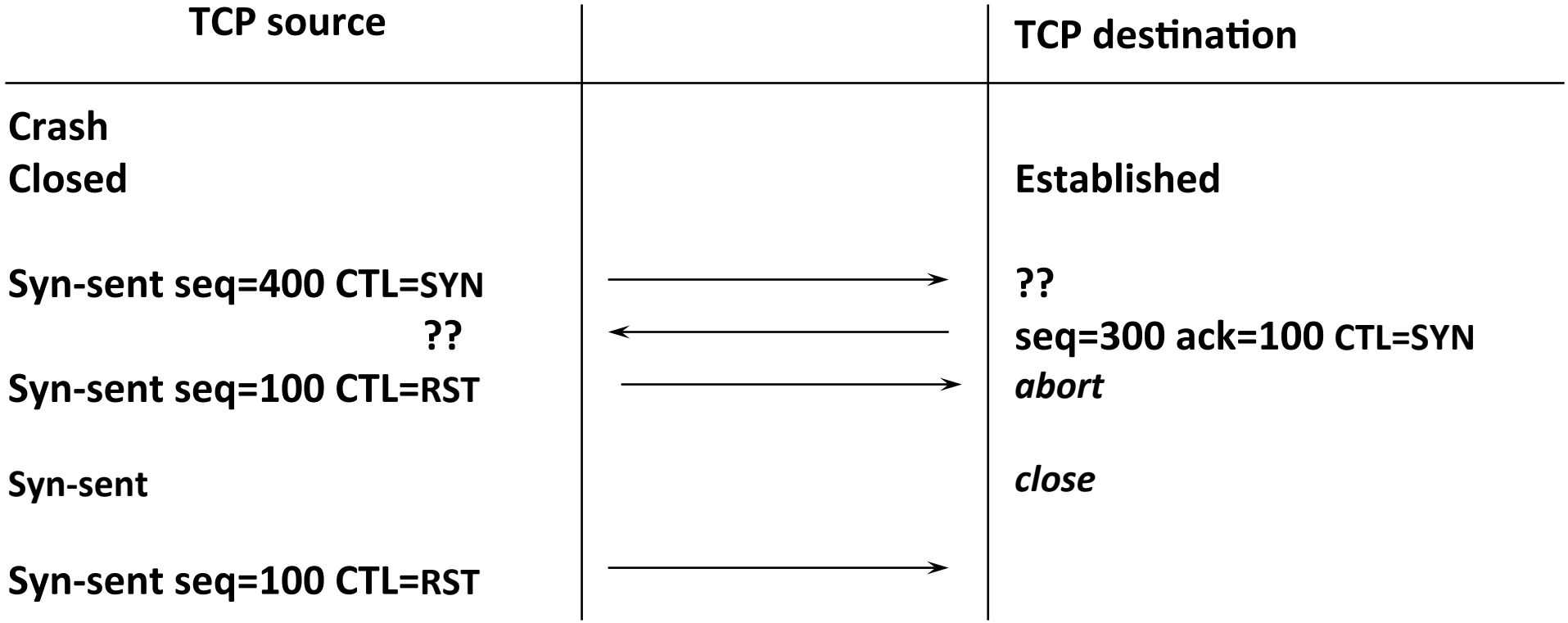
  - lorsque le bit SYN (voir CODE BITS) est positionné, le NSP représente cette donnée de contrôle et par conséquent la suite NSP, NSP+1, NSP+2, ..., NSP+n +1, associe la suite de données SYN, O1, O2, ..., On.
  - lorsque le bit FIN (voir CODE BITS) est positionné, le NSP+n représente cette donnée de contrôle et par conséquent la suite NSP, NSP+1, NSP+2, ..., NSP+n, associe la suite de données O1, O2, ..., On, FIN.
- Numéro d'acquittement : le prochain numéro de séquence NS attendu par l'émetteur de cet acquittement. Acquitte implicitement les octets NS-1, NS-2, etc.
- Fenêtre: la quantité de données que l'émetteur de ce segment est capable de recevoir; ceci est mentionné dans chaque segment (données ou acquittement).

TCP : Format du segment

- CODE BITS (URG,ACK,PSH,RST,SYN,FIN): indique la nature du segment
  - URG : le pointeur de données urgentes est valide (exemple : interrupt en remote login), les données sont émises sans délai, les données reçues sont remises sans délai.
  - SYN : utilisé à l'initialisation de la connexion pour indiquer où la numérotation séquentielle commence. Syn occupe lui-même un numéro de séquence bien que ne figurant pas dans le champ de données. Le Numéro de séquence inscrit dans le datagramme (correspondant à SYN) est alors un *Initial Sequence Number* (ISN) produit par un générateur garantissant l'unicité de l'ISN sur le réseau (indispensable pour identifier les duplications).
  - FIN : utilisé lors de la libération de la connexion;
  - PSH : fonction push. Normalement, en émission, TCP reçoit les données depuis l'applicatif , les transforme en segments à sa guise puis transfère les segments sur le réseau; un récepteur TCP décodant le bit PSH, transmet à l'application réceptrice, les données correspondantes sans attendre plus de données de l'émetteur. Exemple : émulation terminal, pour envoyer chaque caractère entré au clavier (mode caractère asynchrone).

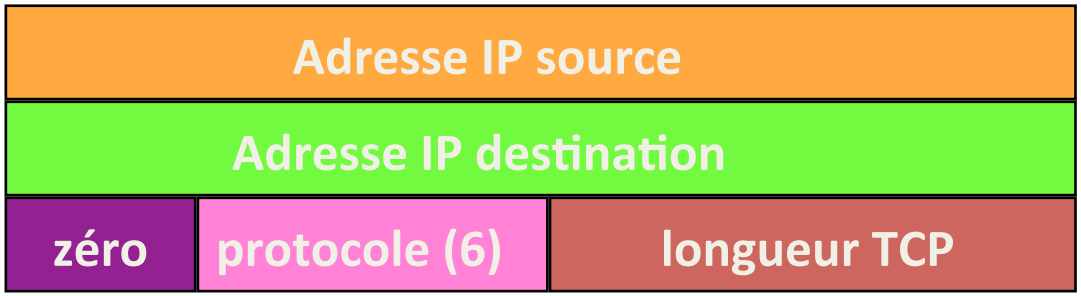
TCP : format du segment

- RST : utilisé par une extrémité pour indiquer à l'autre extrémité qu'elle doit réinitialiser la connexion. Ceci est utilisé lorsque les extrémités sont désynchronisées. Exemple :



TCP format du segment

- CHECKSUM : calcul du champ de contrôle : utilise un pseudo-en-tête et s'applique à la totalité du segment obtenu (PROTO =6) :



TCP : format du header

OPTIONS

- Permet de négocier la taille maximale des segments échangés. Cette option n'est présente que dans les segments d'initialisation de connexion (avec bit SYN).
- TCP calcule une taille maximale de segment de manière à ce que le datagramme IP résultant corresponde au MTU du réseau. La recommandation est de 536 octets.
- La taille optimale du segment correspond au cas où le datagramme IP n'est pas fragmenté mais :
  - il n'existe pas de mécanisme pour connaître le MTU,
  - le routage peut entraîner des variations de MTU,
  - la taille optimale dépend de la taille des en-têtes (options).

TCP : acquittements

Acquittements et retransmissions

- Le mécanisme d'acquittement de TCP est cumulatif :
  - il indique le numéro de séquence du prochain octet attendu : tous les octets précédents cumulés sont implicitement acquittés
  - Si un segment a un numéro de séquence supérieur au numéro de séquence attendu (bien que dans la fenêtre), le segment est conservé mais l'acquittement référence toujours le numéro de séquence attendu(-->).

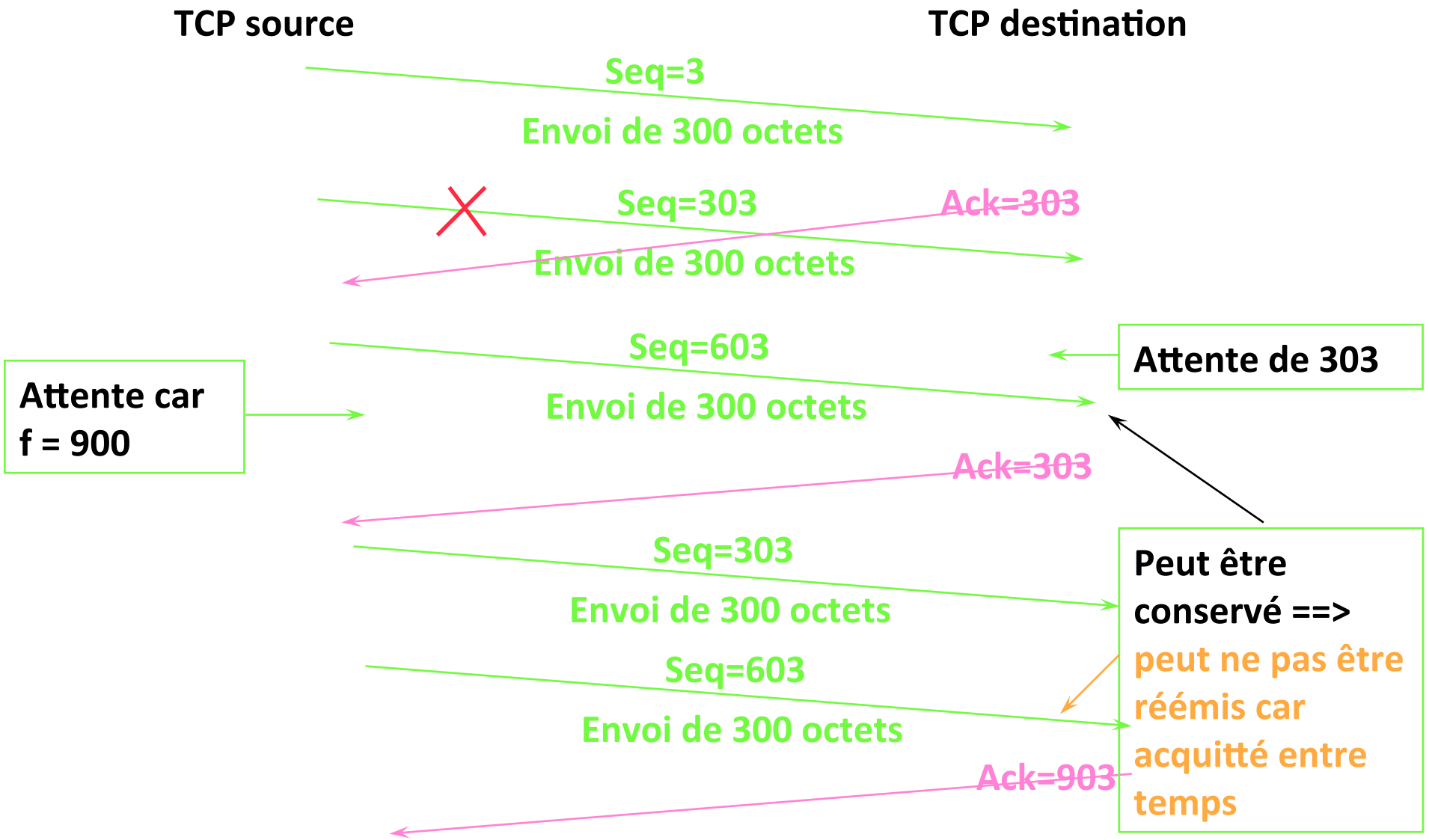
TCP : acquittements

- Pour tout segment émis, TCP s'attend à recevoir un acquittement
  - Si le segment n'est pas' acquitté, le segment est considéré comme perdu et TCP le retransmet.
  - Or un réseau d'interconnexion offre des temps de transit variables nécessitant le réglage des temporisations;
  - TCP gère des temporisations variables pour chaque connexion en utilisant un algorithme de retransmission adaptative

TCP : Acquittements

Fenêtre=900

Segment=300



TCP : retransmissions

<u>algorithme de retransmission adaptative</u>

- enregistre la date d'émission d'un segment,
- enregistre la date de réception de l'acquittement correspondant,
- calcule l'échantillon de temps de boucle A/R écoulé,
- détermine le temps de boucle moyen RTT (Round Trip Time) :

  RTT = (a * anc_RTT) + ((1-a) * NOU_RTT))

  avec 0<= a < 1

  a proche de 1 : RTT insensible aux variations brèves,

  a proche de 0 : RTT très sensible aux variations rapides,
- calcule la valeur du temporisateur en fonction de RTT.
- Les premières implémentations de TCP ont choisi un coefficient constant B pour déterminer cette valeur : Temporisation = B * RTT avec B >1 (généralement B=2).
- Aujourd'hui de nouvelles techniques sont appliquées pour affiner la mesure du RTT : l'algorithme de Karn.

TCP : retransmissions

L'algorithme de Karn repose sur les constatations suivantes :

- en cas de retransmission d'un segment, l'émetteur ne peut savoir si l'acquittement s'adresse au segment initial ou retransmis (ambiguïté des acquittements), =>l'échantillon RTT ne peut donc être calculé correctement,
- => TCP ne doit pas mettre à jour le RTT pour les segments retransmis.

- L'algorithme de Karn combine les retransmissions avec l'augmentation des temporisations associées (*timer backoff*):
  - une valeur initiale de temporisation est calculée
  - si une retransmission est effectuée, la temporisation est augmentée (généralement le double de la précédente, jusqu'à une valeur plafond).
- Cet algorithme fonctionne bien même avec des réseaux qui perdent des paquets.

TCP : la congestion

Gestion de la congestion

- TCP gère le contrôle de flux de bout en bout mais également les problèmes de congestion liés à l'interconnexion.

- La congestion correspond à la saturation de noeud(s) dans le réseau provoquant des délais d'acheminement de datagrammes jusqu'a leur pertes éventuelles.

- Les extrémité ignorent tout de la congestion sauf les délais. Habituellement, les protocoles retransmettent les segments ce qui agrave encore le phénomène.

- Dans la technologie TCP/IP, les passerelles (niveau IP) utilisent la réduction du débit de la source mais TCP participe également à la gestion de la congestion en diminuant le débit lorsque les délais s'allongent :

TCP : la congestion

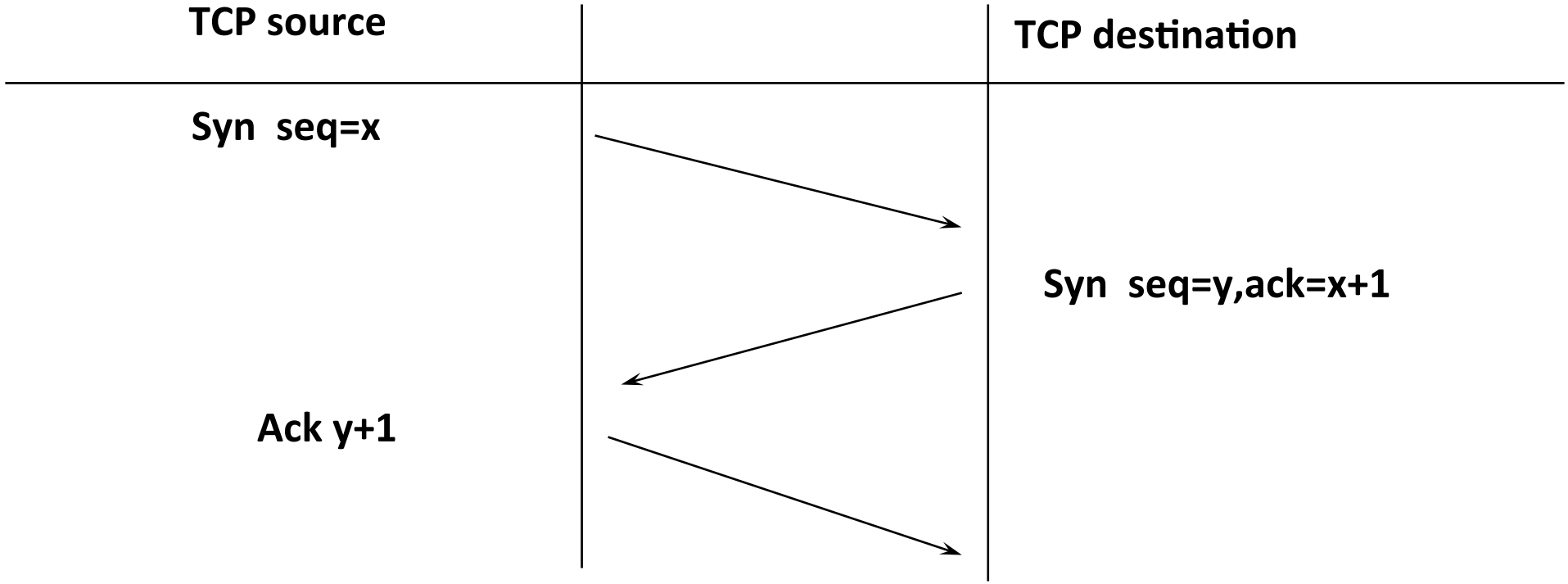
- TCP maintient une fenêtre virtuelle de congestion
- TCP applique la fenêtre d'émission suivante:
  - fenêtre_autorisée = min (fenêtre_récepteur, fenêtre_congestion).

- Dans une situation de non congestion:
  - fenêtre_récepteur = fenêtre_congestion.

- En cas de congestion, TCP applique une diminution dichotomique :
  - à chaque segment perdu, la fenêtre de congestion est diminuée par 2 (minimum 1 segment)
  - la temporisation de retransmission est augmentée exponentiellement.

TCP : connexion (3-way handshake)

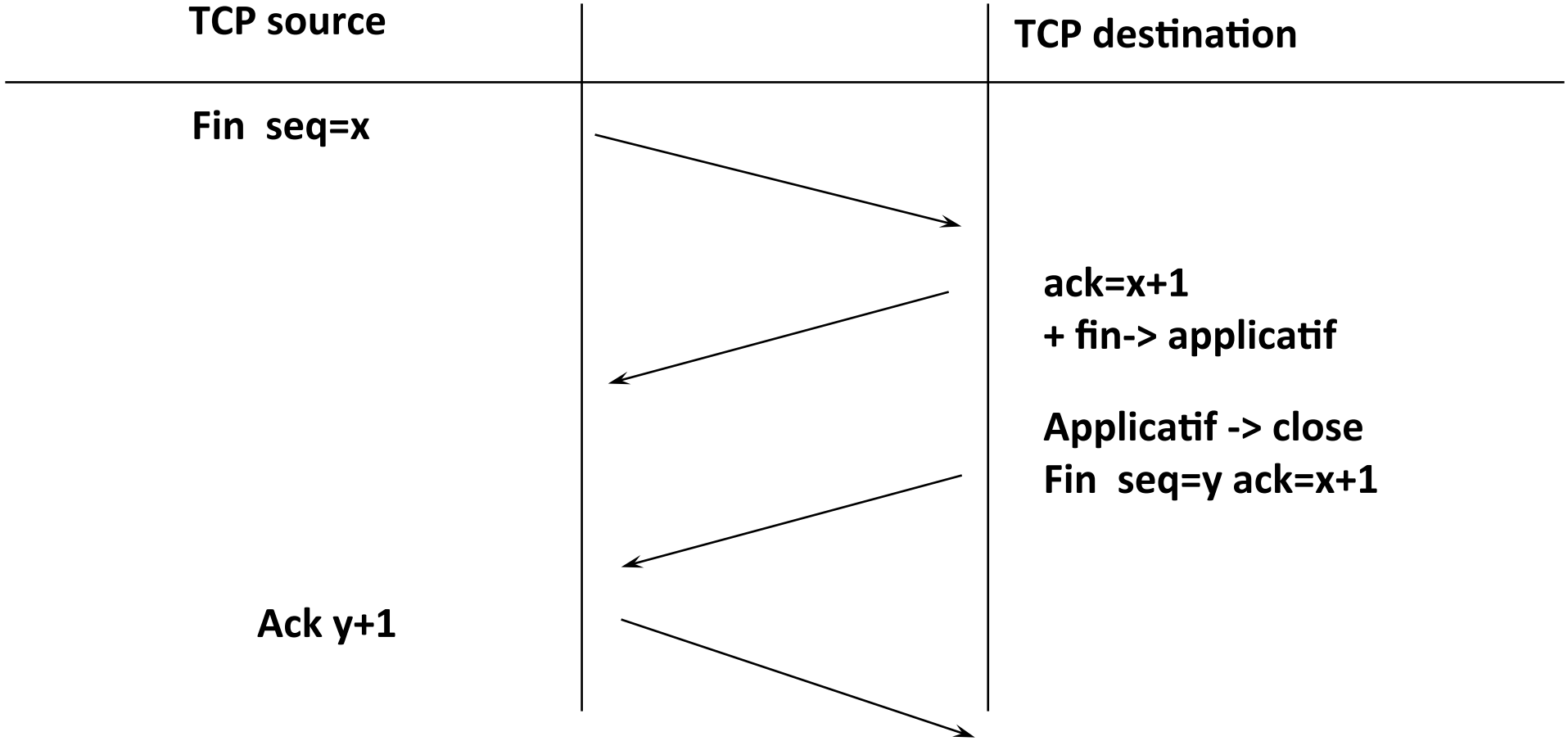
Une connexion TCP est établie en trois temps de manière à assurer la synchronisation nécessaire entre les extrémités :



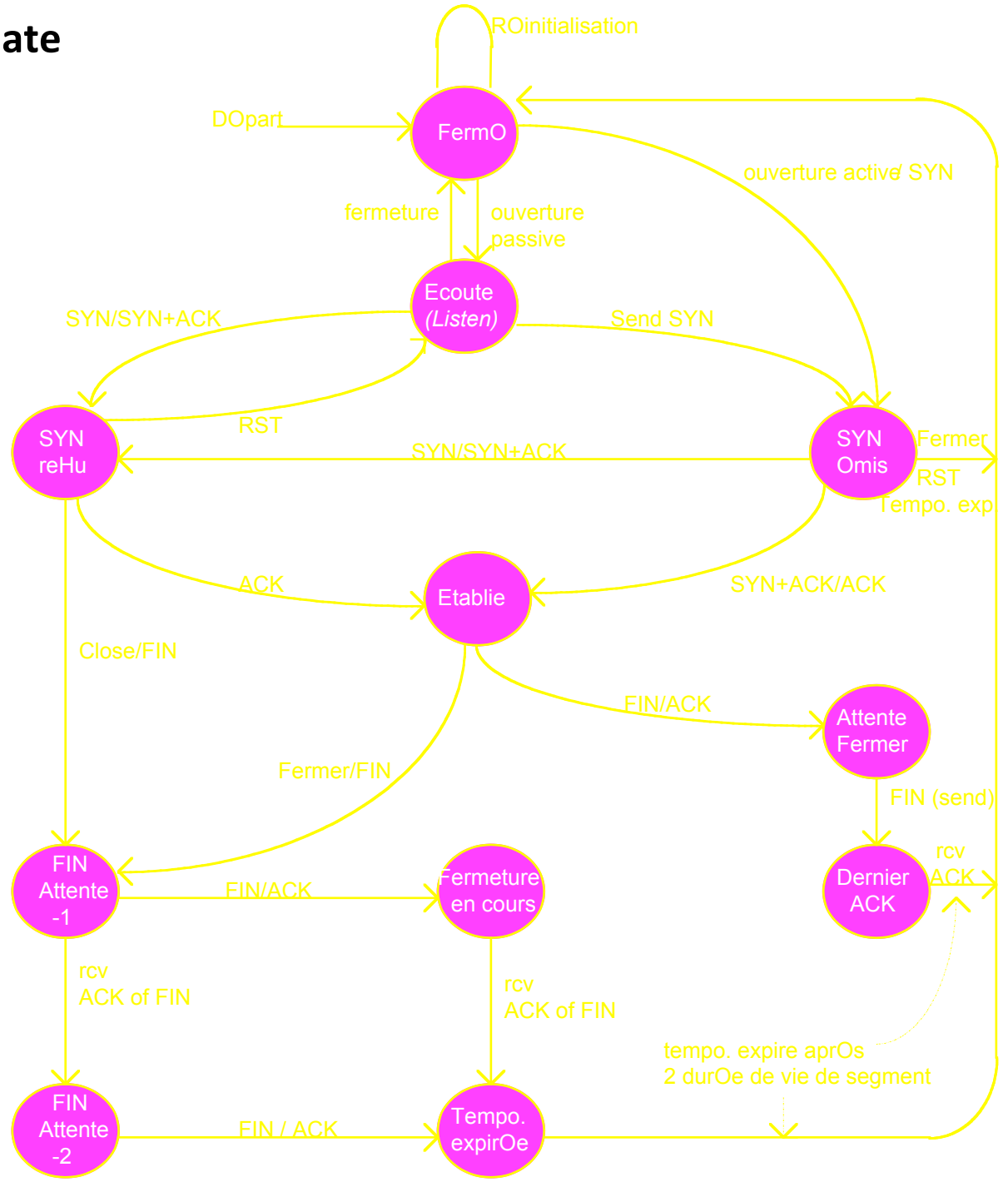
Ce schéma fonctionne lorsque les deux extrémités effectuent une demande d'établissement simultanément. TCP ignore toute demande de connexion, si cette connexion est déjà établie.

TCP : déconnexion

- Une connexion TCP est libérée en un processus dit ”3 way handshake”modifiée:



TCP : L'automate



TCP : ports standards

No port	Mot-clé	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80	HTTP	WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call

Bibliographie

Internetworking with TCP/IP, Douglas E. Comer
TCP/IP Protocoles de base, F. Playe (centraleWeb)
TCP/IP network administration, Craig Hunt
TCP/IP Illustré, Richard Stevens