

Couche transport

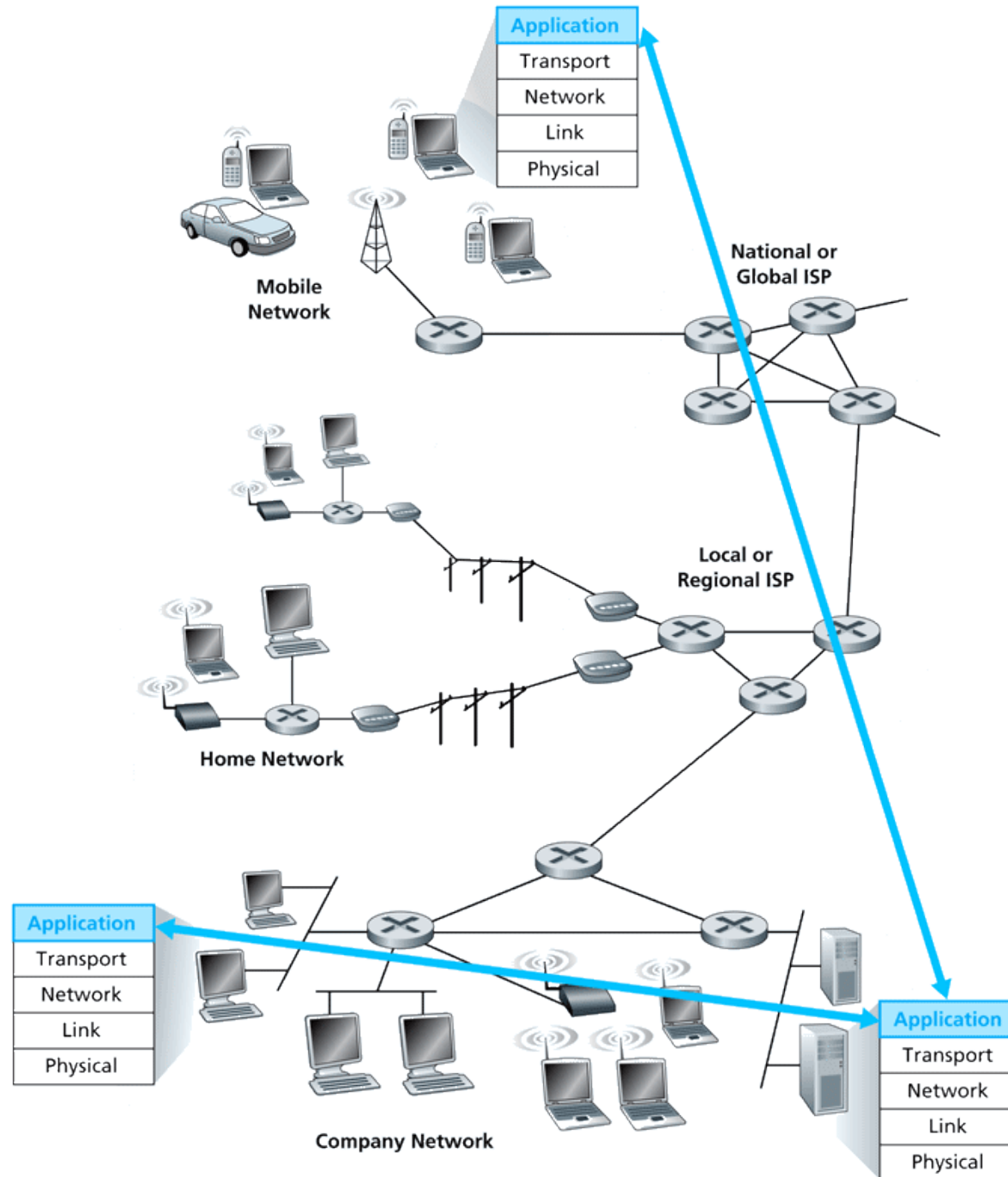
Principes et implantations

**Introduction aux
Réseaux**

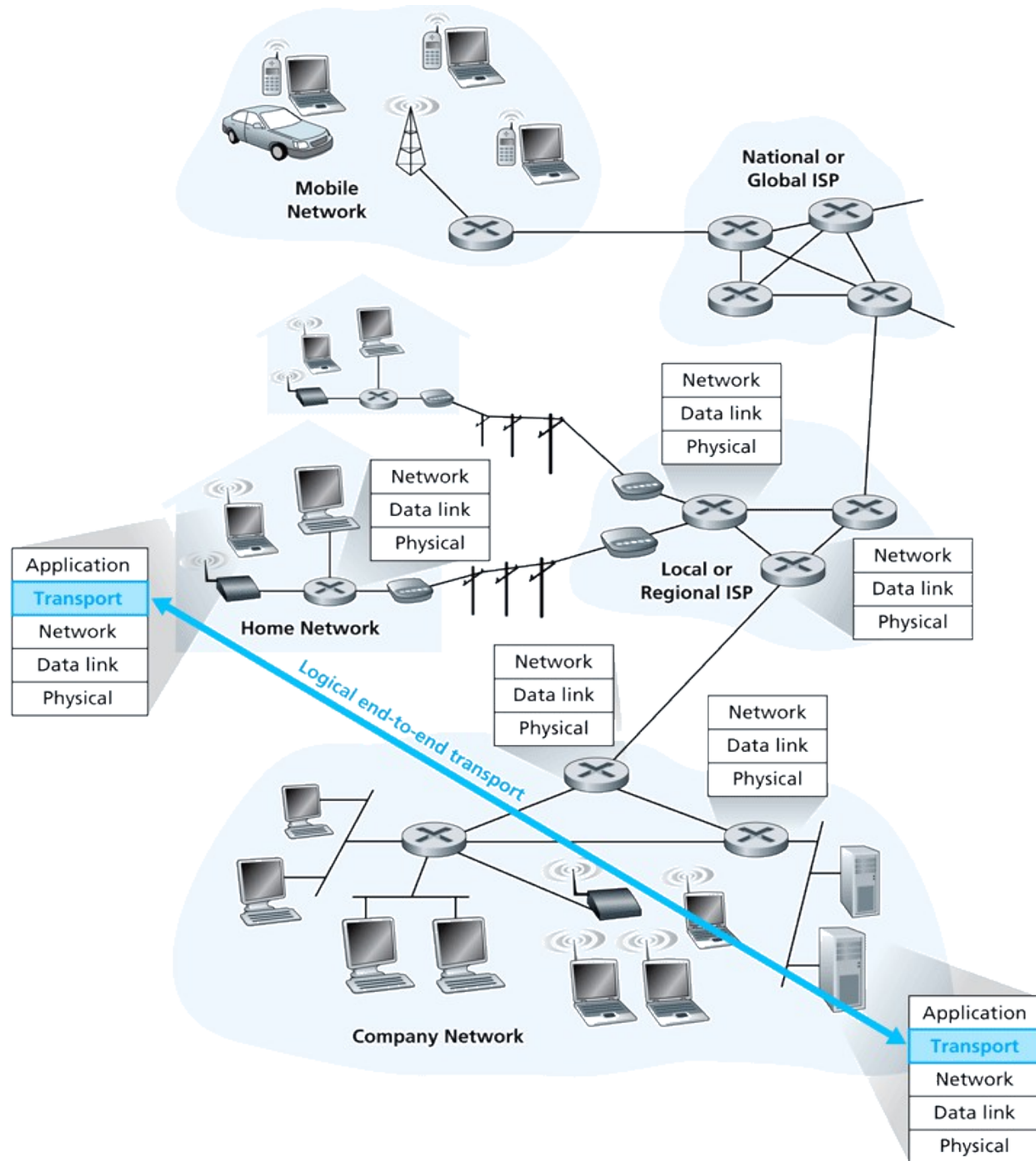
Modèles OSI et Internet



Objectif des applications réseaux



Couche transport



Plan

Objectifs et services

Notion de socket et de port

Multiplexage de la couche transport

Service de transmission non fiable UDP

Service de transmission fiable TCP

- Conception d'un service fiable
- Protocoles transport fiables

Couche transport

Objectifs

- Étendre le service de liaison entre hôtes de la couche réseau à un service de liaison entre applications (processus)

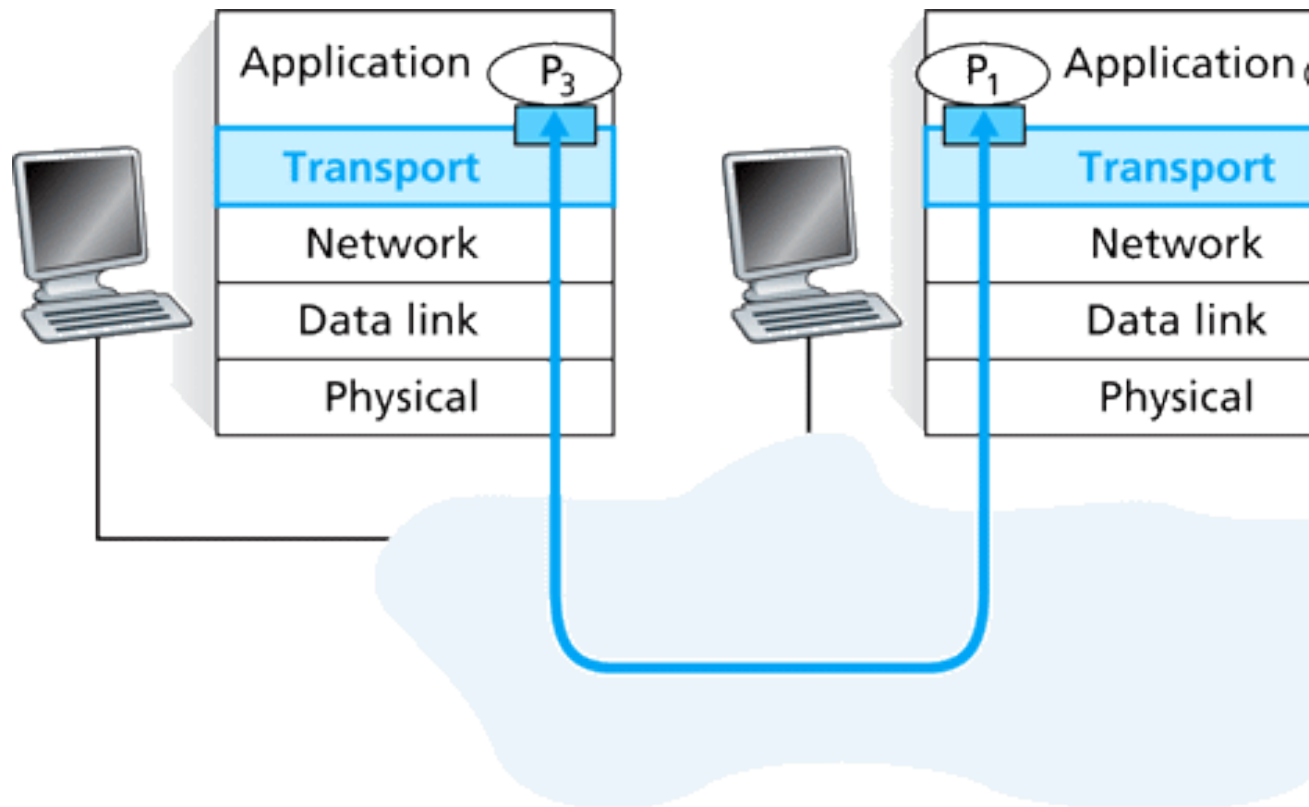
Services

- Multiplexage et démultiplexage de la couche transport
- Service avec et sans connexion
- Détection d'erreurs
- Acquittements

Sockets

Définition

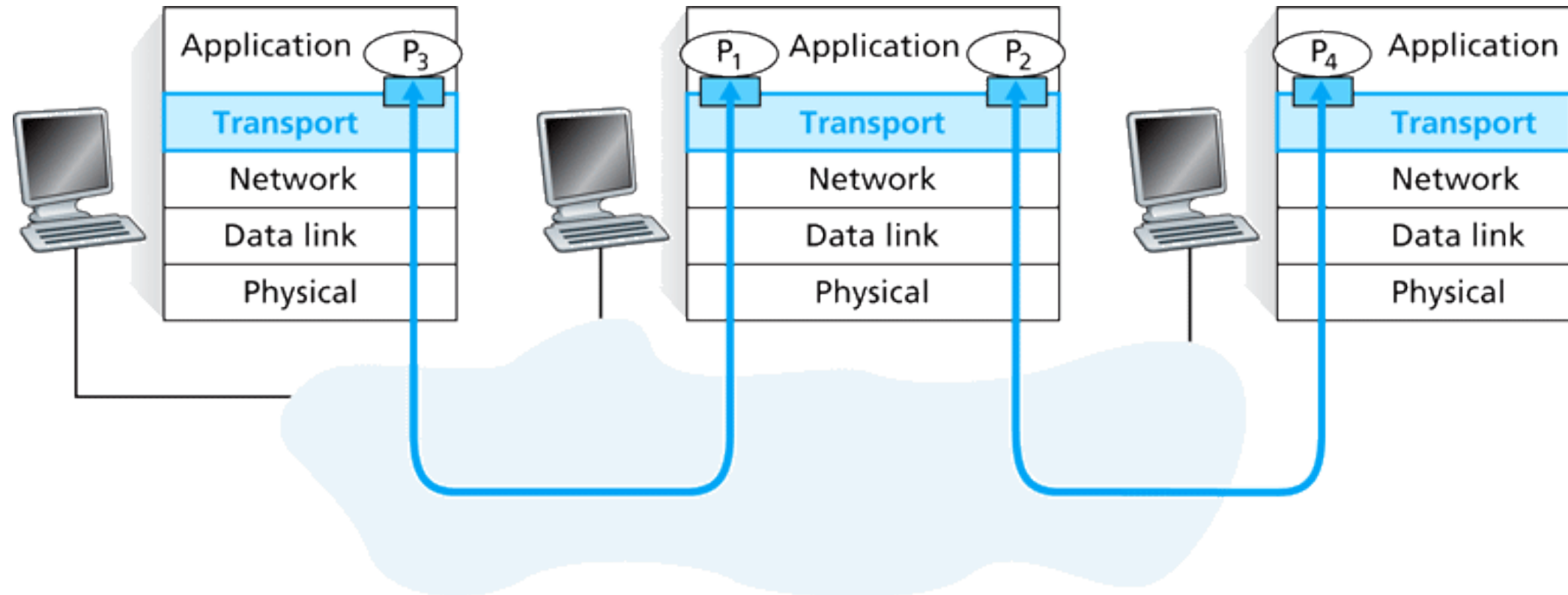
- Une socket est le point d'entrée à travers lequel l'information transite d'un processus (une application) vers le réseau et vice-versa
- Lien entre le processus est la couche transport



Notions de numéro de port

Problème

- Il y a plusieurs processus sur les hôtes, comment l'information est elle acheminée vers le bon processus ?



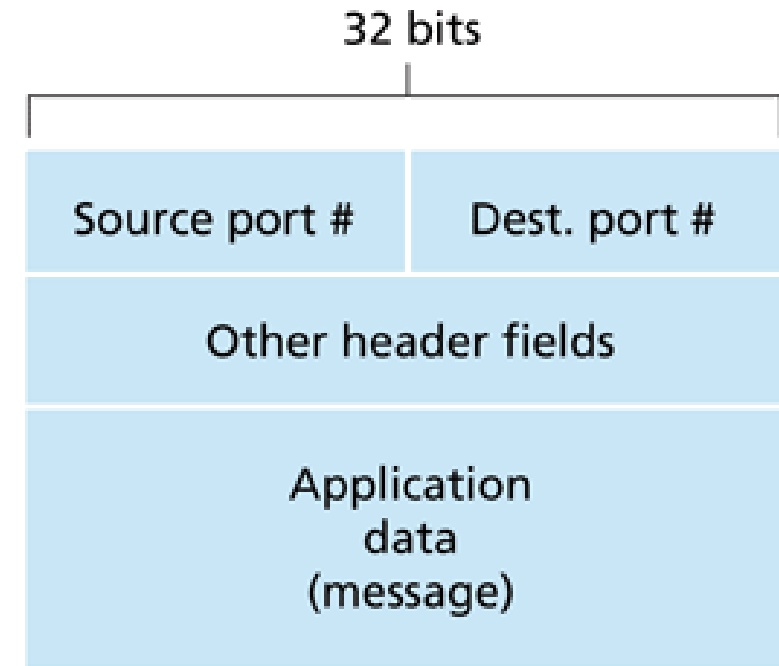
Numéro de port

Solution

- A chaque socket, on associe un identifiant unique (un port)
- Chaque segment contient un champ en entête qui indique la socket

Champs entête des segments

- Port destination et port source
- Port = 16 bits (0 à 65535)
 - 0 à 1023 ports réservés
 - 23 : telnet (shell distant)
 - 21 + 22 : ftp (transfert de fichiers)
 - 25 : smtp (mail)
 - 80 : http (web)

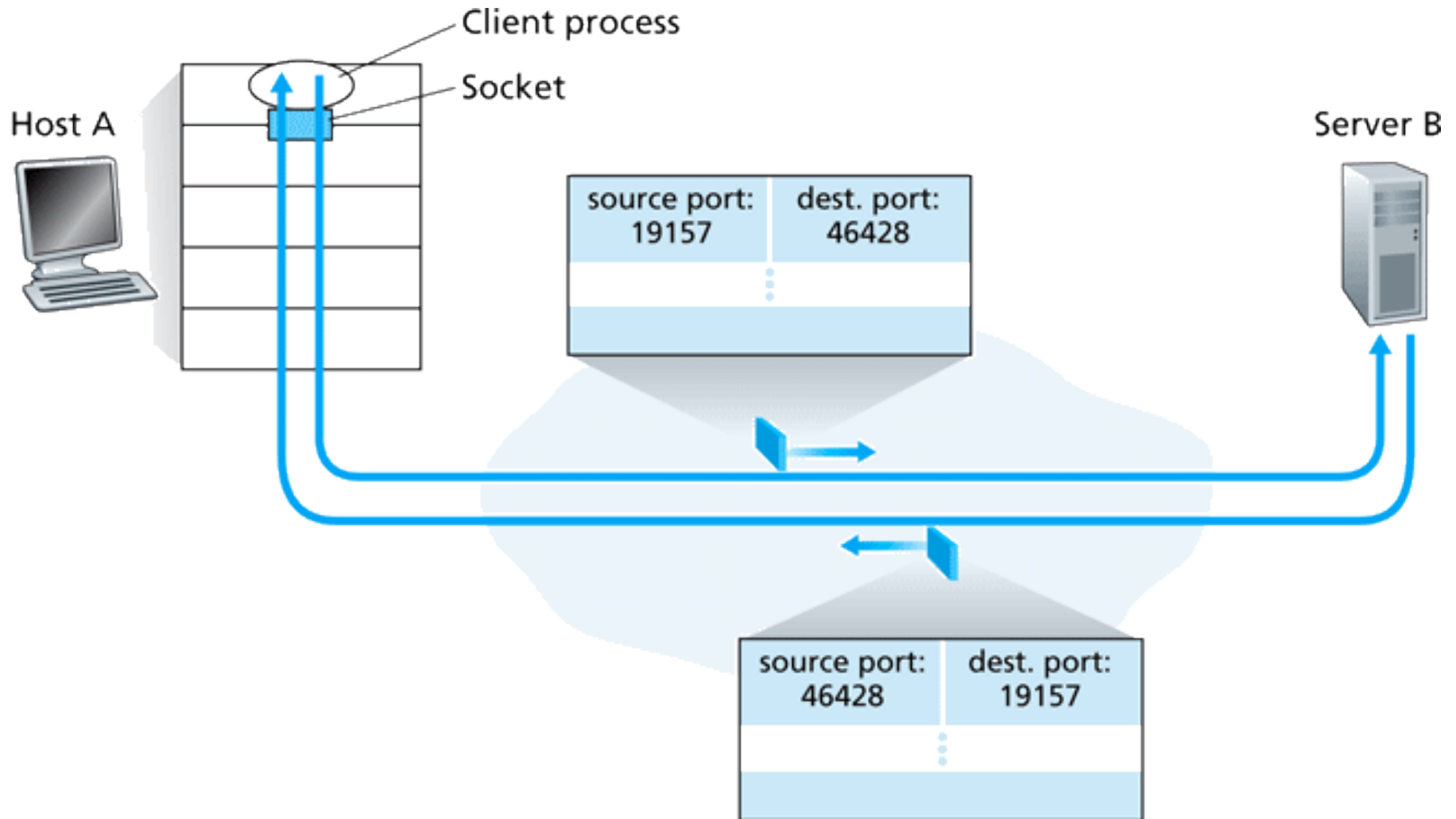


Segment de couche transport

Fonctionnement avec numéro de port

Inversement des ports source et destination

- En fonction du sens de la communication



Plan

Objectifs et services

Notion de socket et de port

Multiplexage de la couche transport

Service de transmission non fiable UDP

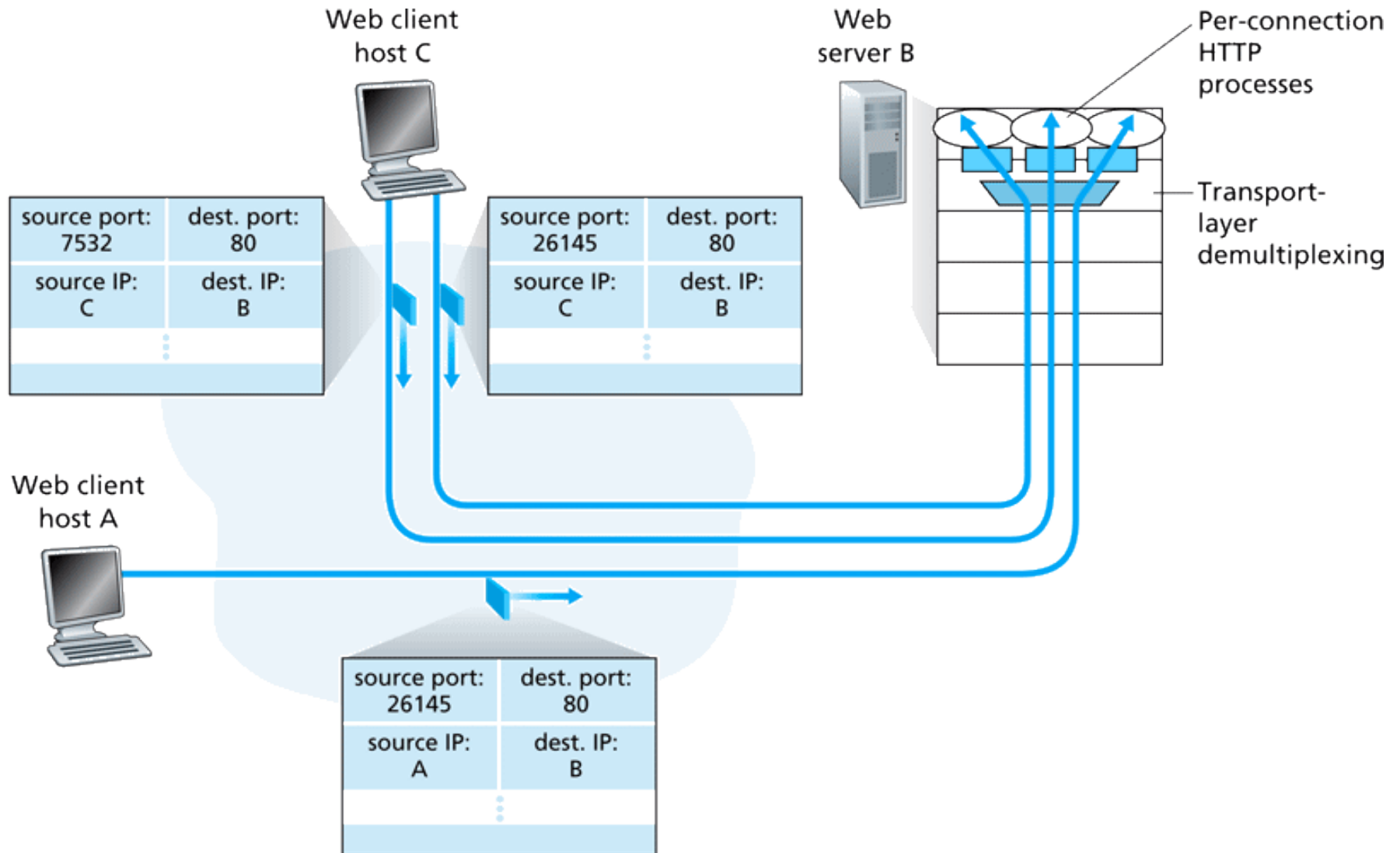
Service de transmission fiable TCP

- Conception d'un service fiable
- Protocoles transport fiables

Multiplexage de la couche transport

Définition

- Partager l'accès au réseau par plusieurs processus
- Acheminer le segment vers le bon processus



Plan

Objectifs et services

Notion de socket et de port

Multiplexage de la couche transport

Service de transmission non fiable UDP

Service de transmission fiable TCP

- Conception d'un service fiable
- Protocoles transport fiables

Le service sans connexion UDP

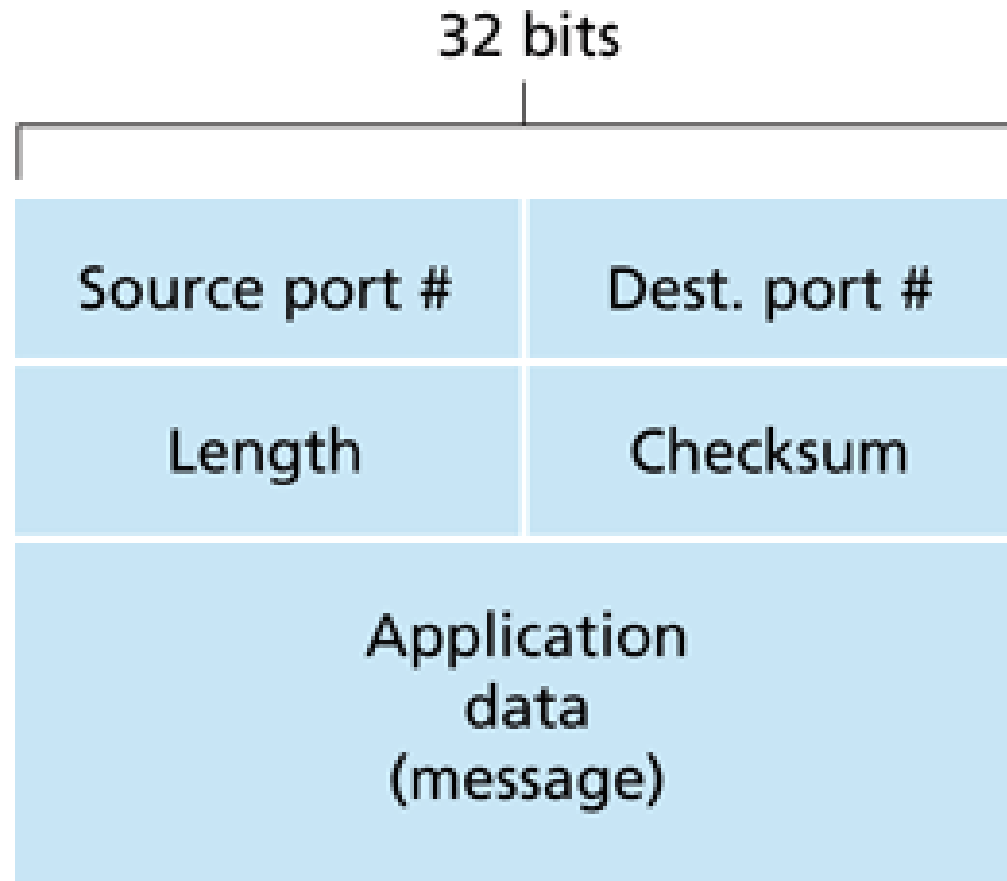
Protocole UDP (*User Datagram Protocol*)

- pas d'établissement de connexion (handshake)
- pas d'état de connexion
- faible volume des entêtes de paquets
- meilleur contrôle de l'application (envoi immédiat)

Le segment est envoyé sans garantie

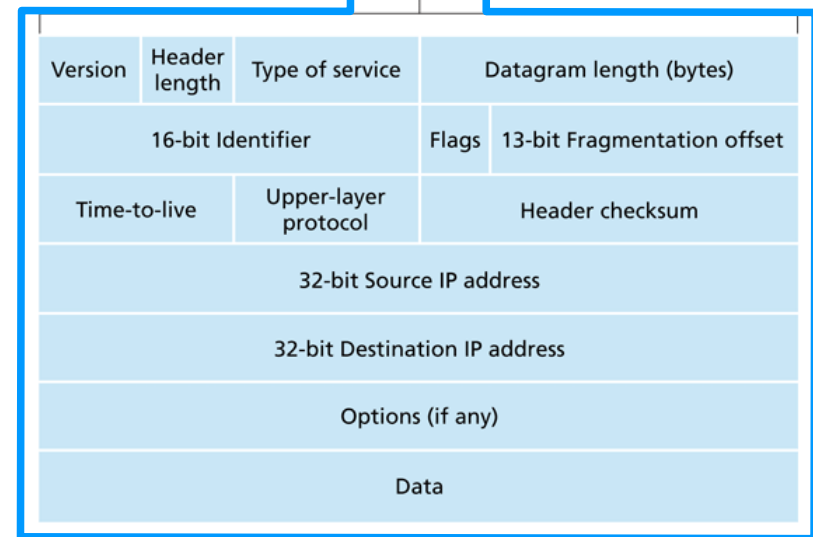
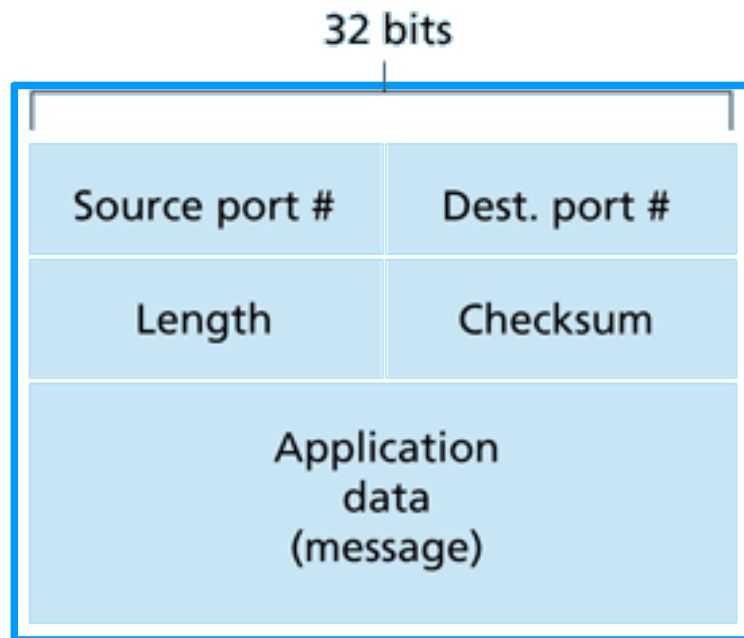
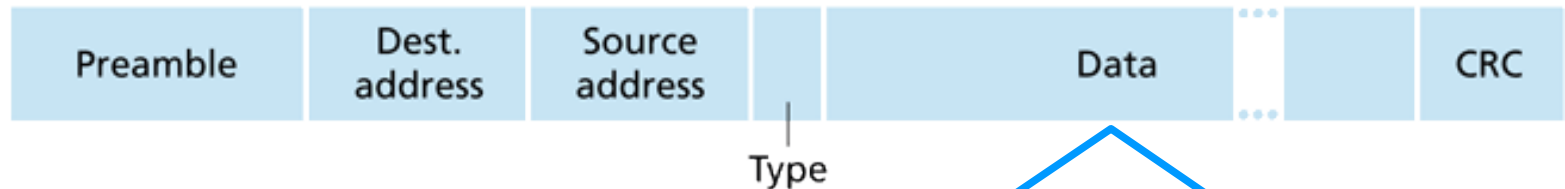
- champ d'utilisation limité aux applications tolérantes aux pertes et temps-réel
 - vidéo-conférence
 - téléphonie
 - multimédia

Segment UDP



Encapsulation des segments UDP

Trame Ethernet (couche liaison)



Datagramme IP (couche réseau)

Plan

Objectifs et services

Notion de socket et de port

Multiplexage de la couche transport

Service de transmission non fiable UDP

Service de transmission fiable TCP

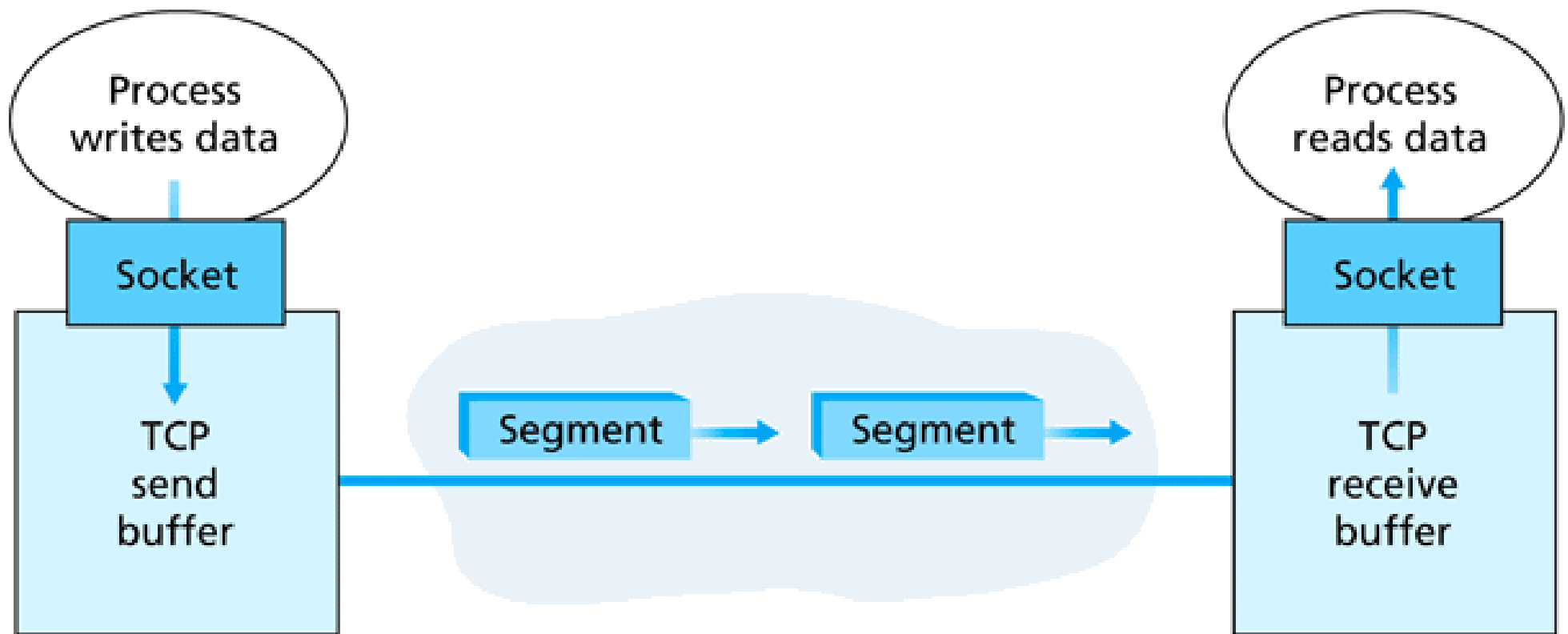
- Conception d'un service fiable
- Protocoles transport fiables

Le service avec connexion TCP

Protocole TCP (*Transport Control Protocol*)

- service fiable, sans perte et sans erreur
- service avec acquittement et retransmission
- orienté connexion avec « 3-way Handshake »
- instauration de connexion et allocation de ressources
- full-duplex
 - les deux processus sur les deux hôtes échangent dans les deux sens
- point-à-point
 - seuls deux processus en communication

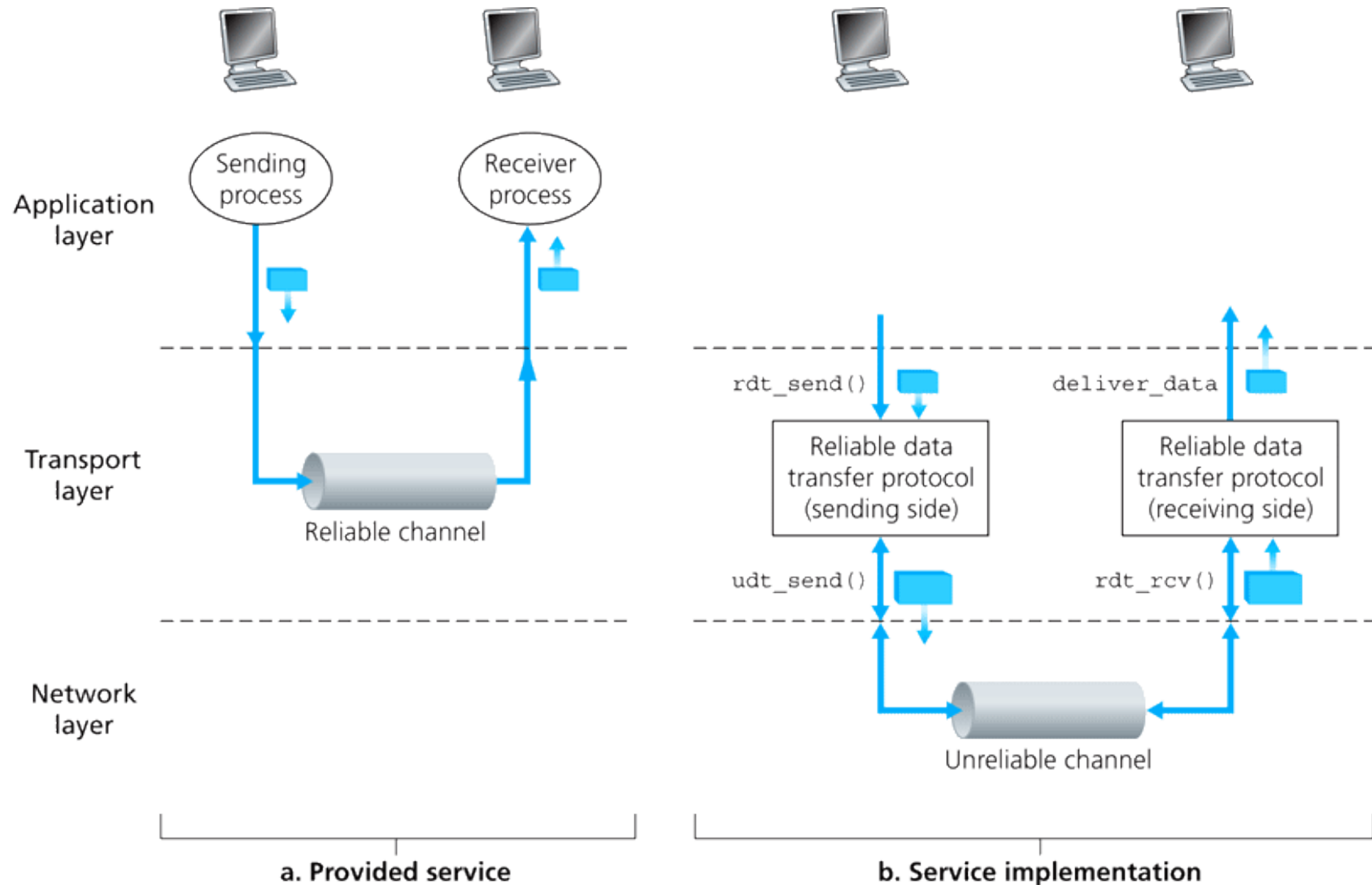
Le service avec connexion TCP



Transmissions fiables

Les données

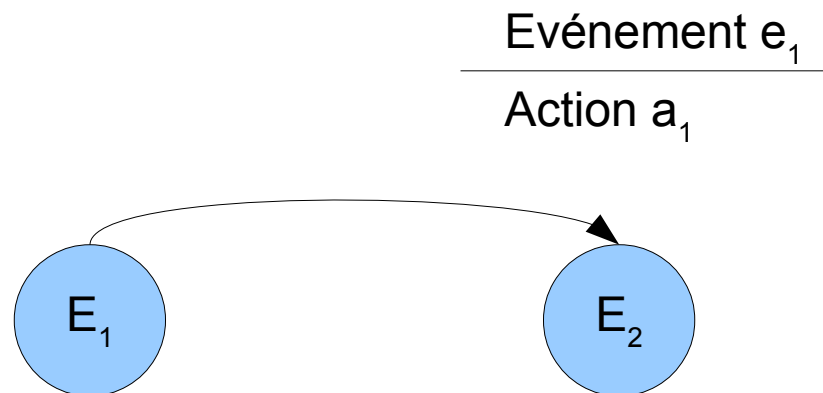
- ne doivent pas être altérées ou être perdues
- doivent être livrées dans l'ordre de leur envoi



Modélisation de protocole

Automates à états finis

- Un ensemble d'états d'une machine donnée, qui se déclenchent suite à des événements précis
- A chaque événement, l'automate exécute des actions et change d'état



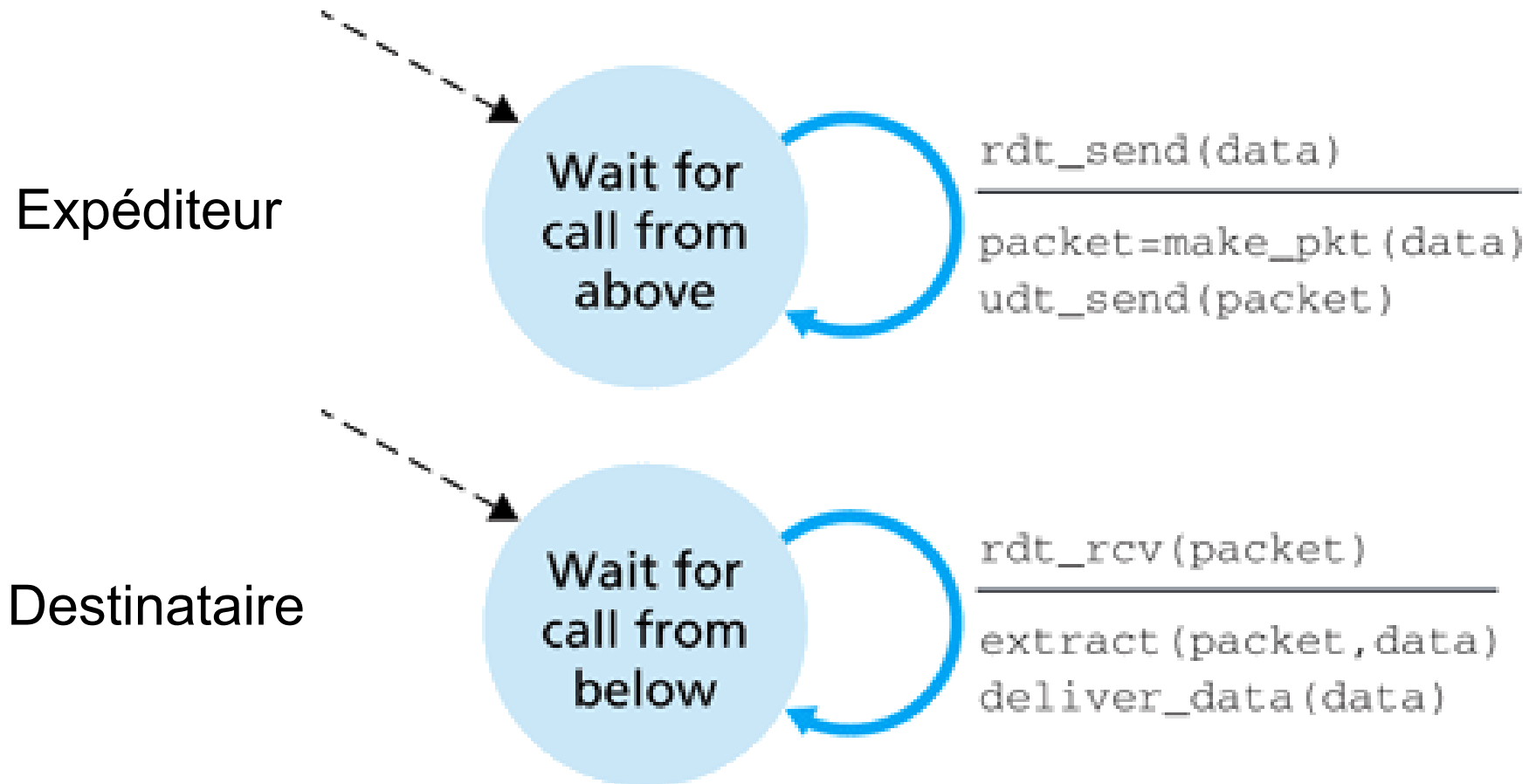
Exemple

- Suite à l'événement e_1 l'automate exécute l'action a_1 et change de l'état E_1 à l'état E_2

Glossaire

Événements	rdt_send(data)	Envoi de data par l'application
	rdt_rcv(rcvpkt)	Réception de rcvpkt par la couche réseau
	corrupt(rcvpkt)	Vrai si rcvpkt est corrompu
	notcorrupt(rcvpkt)	Vrai si rcvpkt n'est pas corrompu
	isACK(rcvpkt)	Vrai si rcvpkt est un ACK
	isNAK(rcvpkt)	Vrai si rcvpkt est un NAK
	has_seq0(rcvpkt)	Vrai si rcvpkt a le numéro de séquence 0
	has_seq1(rcvpkt)	Vrai si rcvpkt a le numéro de séquence 1
	timeout	Fin du temporisateur
	udt_send(sndpkt)	Envoi de sndpkt à la couche réseau
	pkt=(data, checksum)	Construire un segment pkt avec data et calculer le checksum
Actions	extract(rcvpkt, data)	Extraire data du segment rcvpkt
	deliver_data(data)	Envoi de data à l'application
	start_timer	Démarrer le temporisateur
	stop_timer	Arrêter le temporisateur
	\wedge	Ne rien faire

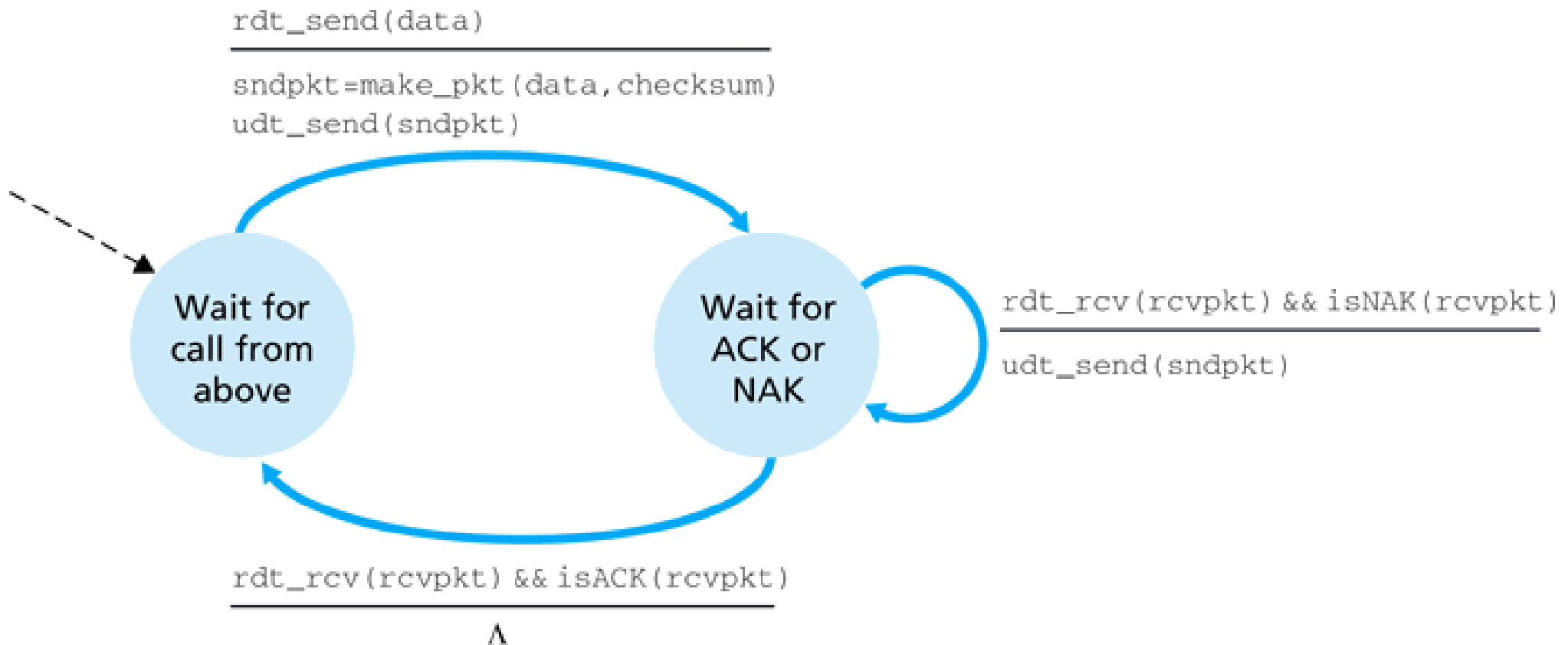
Cas 1 : canal sans erreur



Cas 2 : canal avec erreurs (expéditeur)

Utilisation d'acquittements

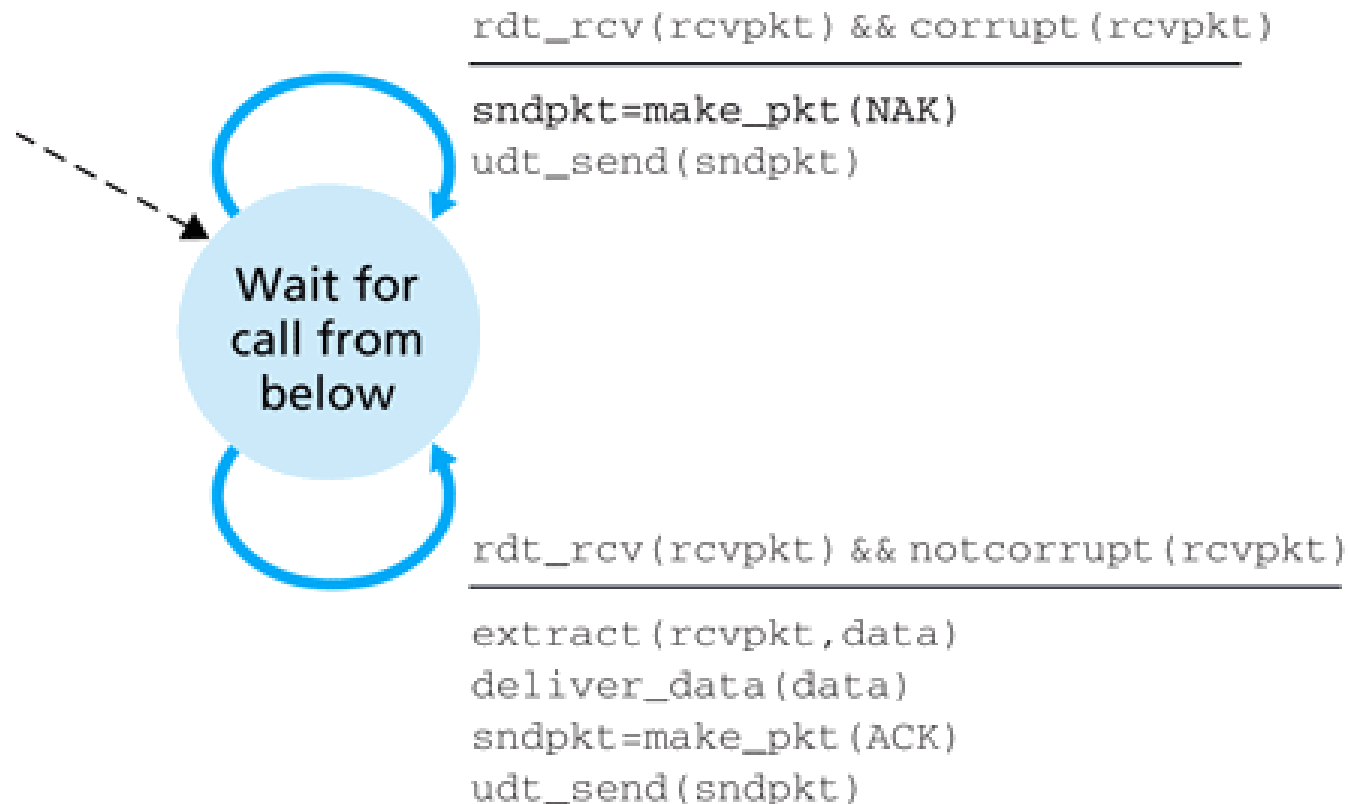
- positifs ACK et négatifs NAK
- détection d'erreurs + retransmission



Cas 2 : canal avec erreurs (destinataire)

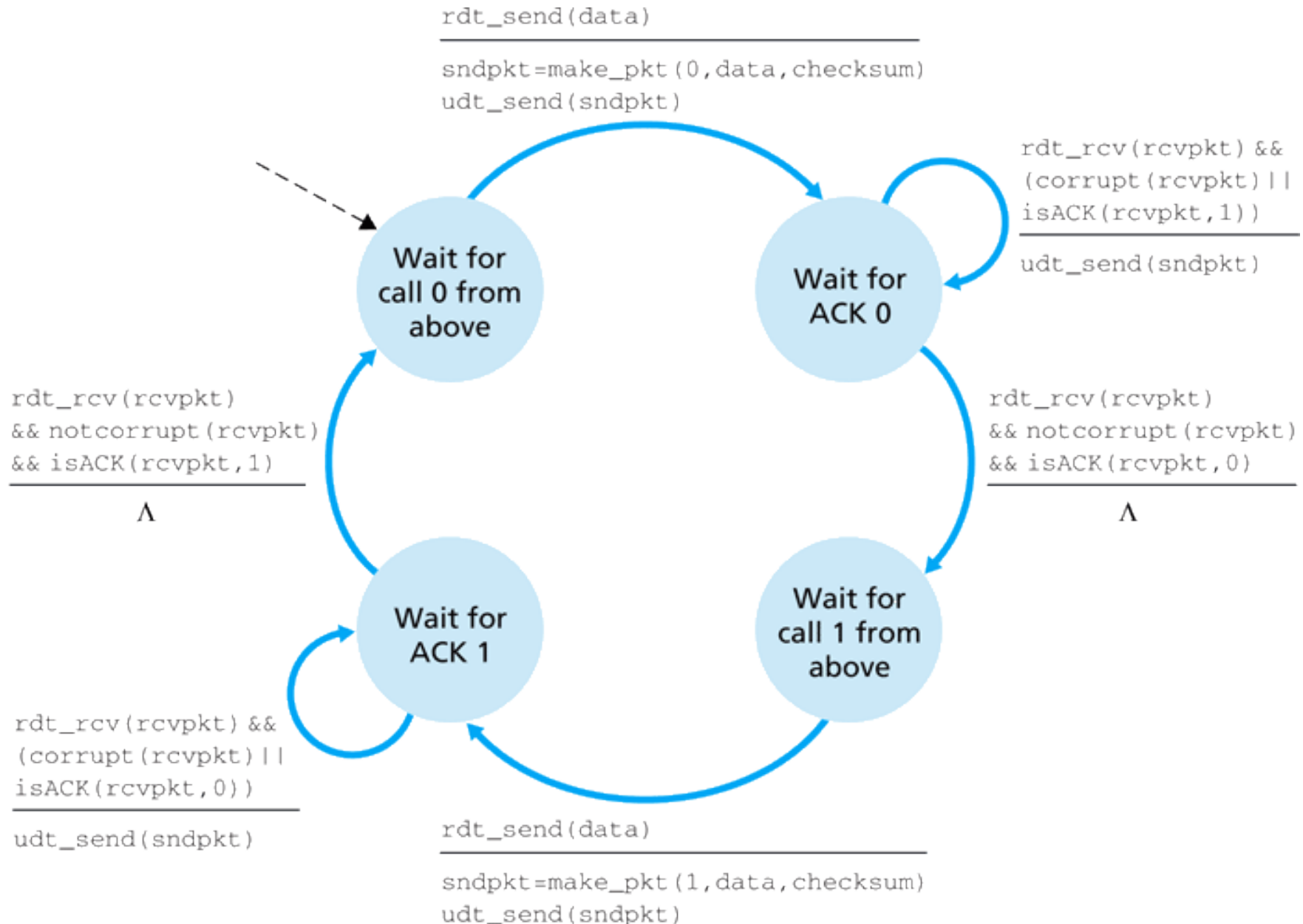
Utilisation d'acquittements

- positifs ACK et négatifs NAK
- détection d'erreurs + retransmission



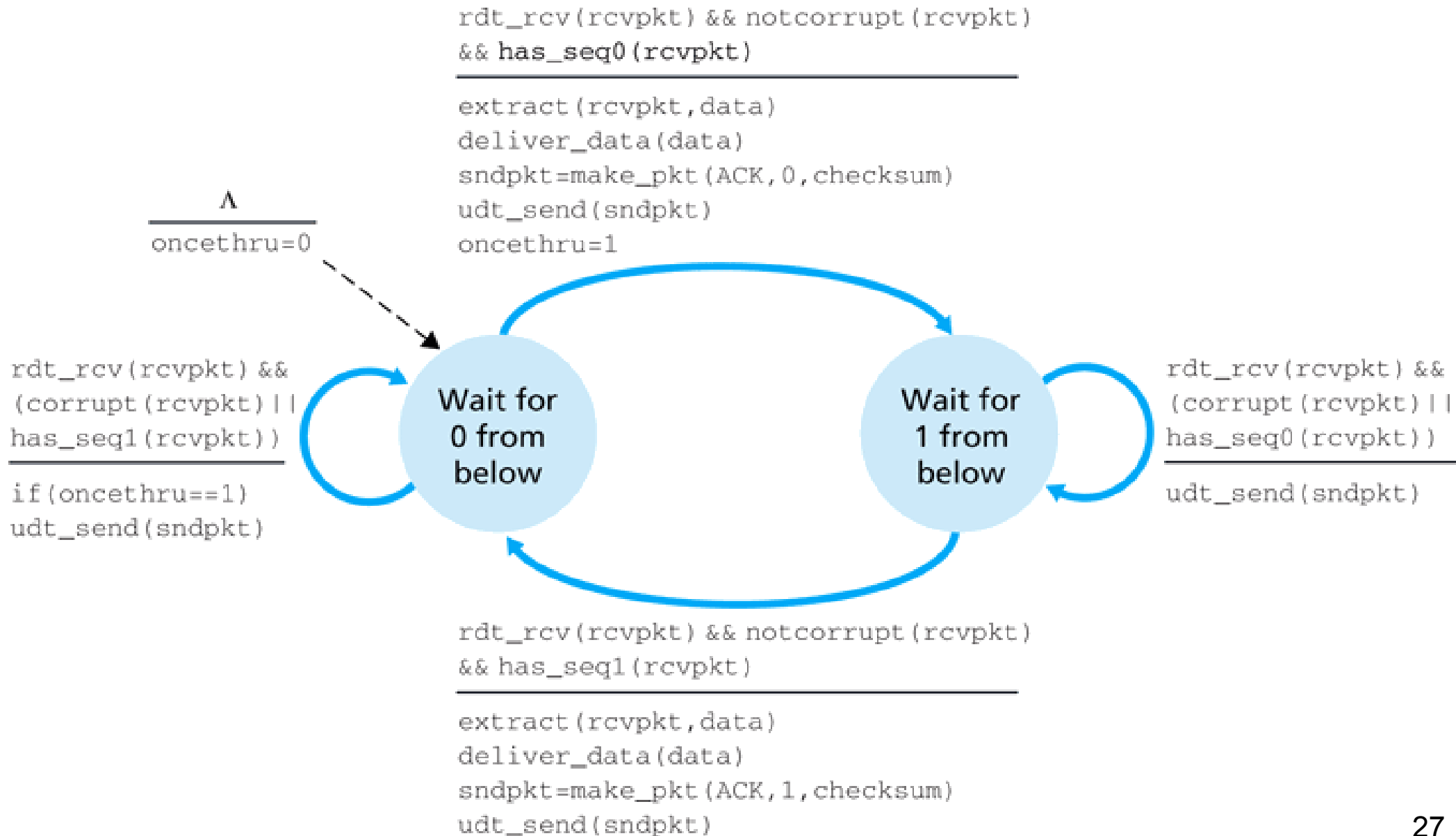
Cas 3 : canal avec erreurs (expéditeur)

Tenir compte de la corruption des acquittements



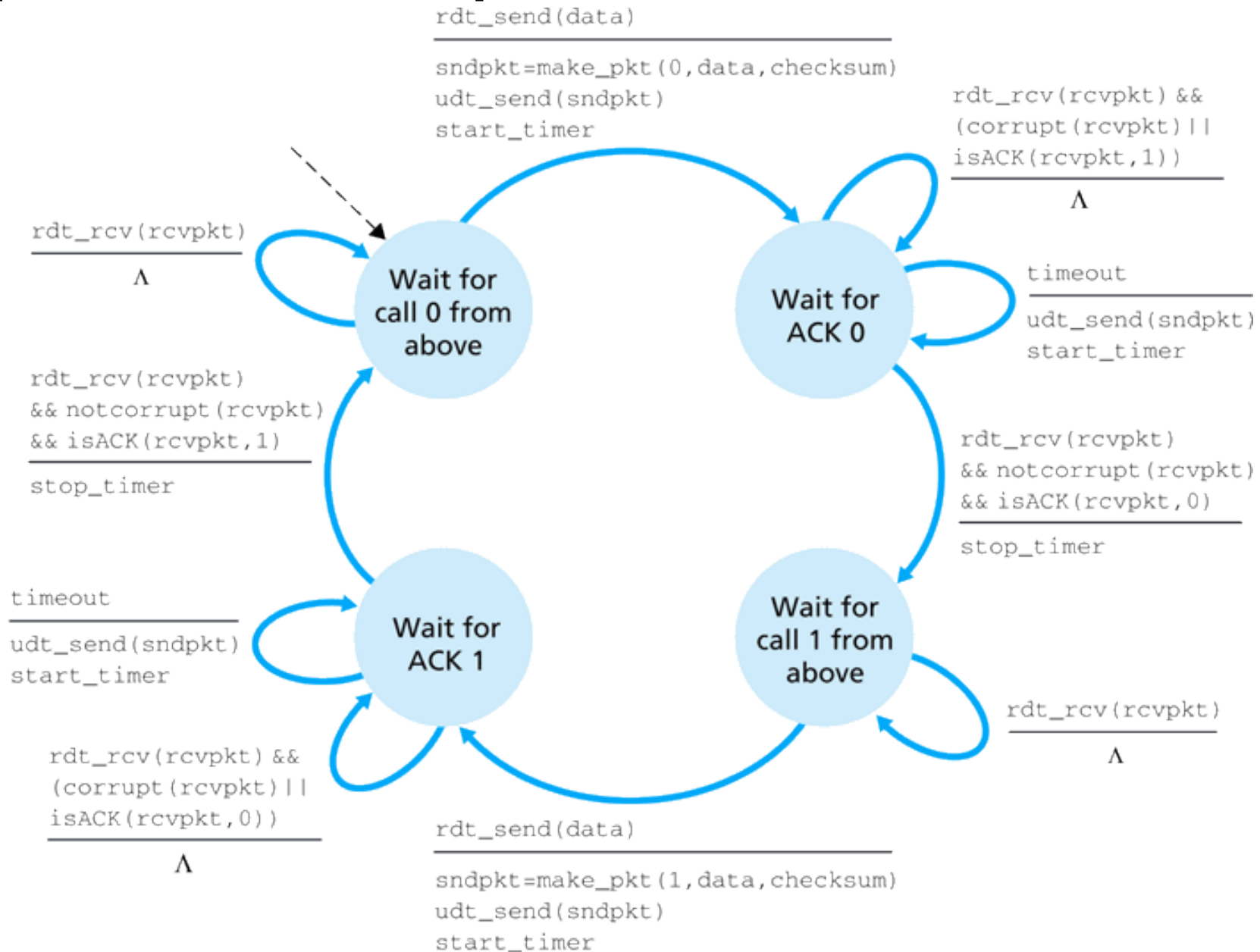
Cas 3 : canal avec erreurs (destinataire)

Tenir compte de la corruption des acquittements



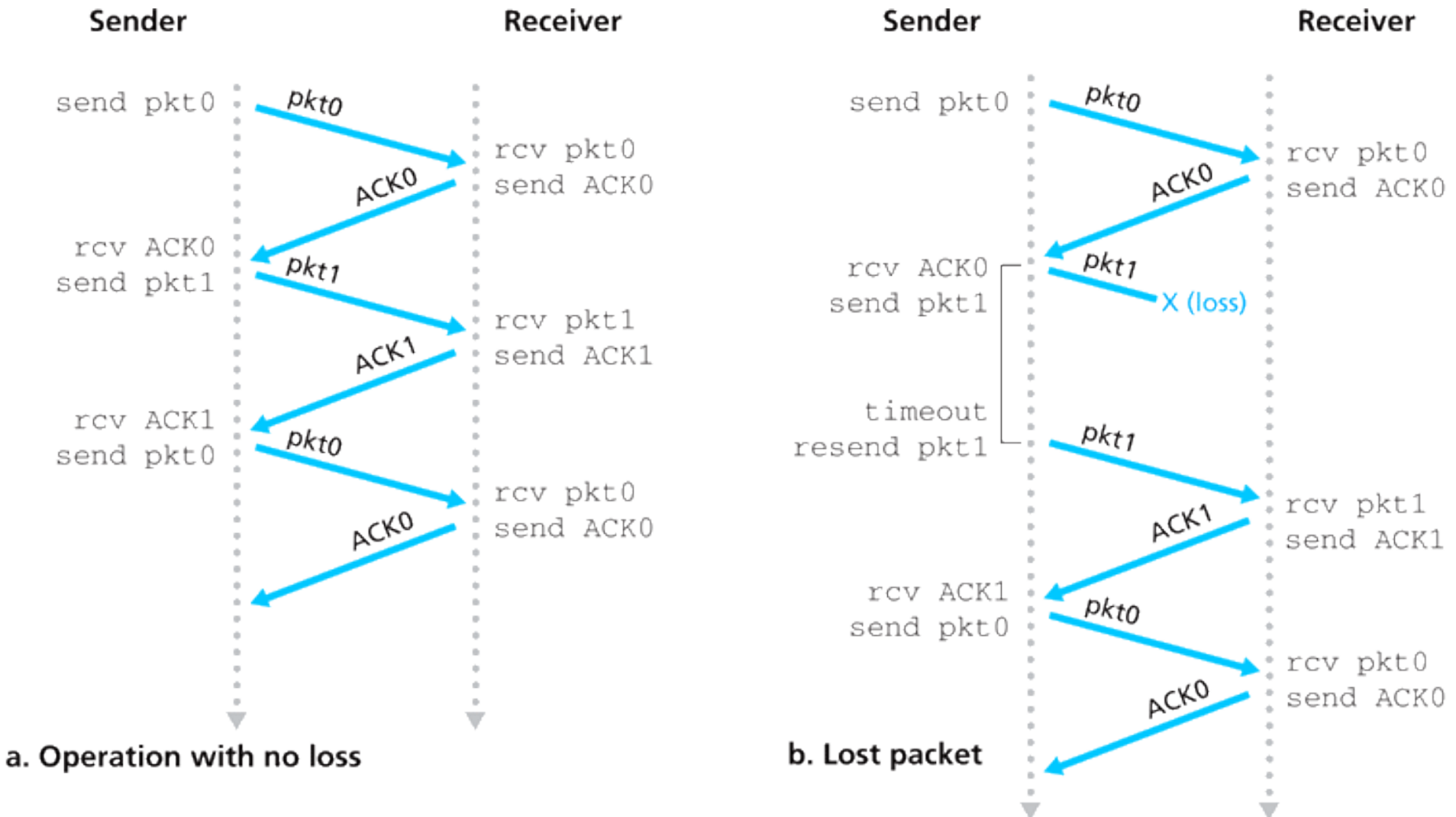
Cas 4 : canal avec perte de paquets

Expéditeur avec temporisation



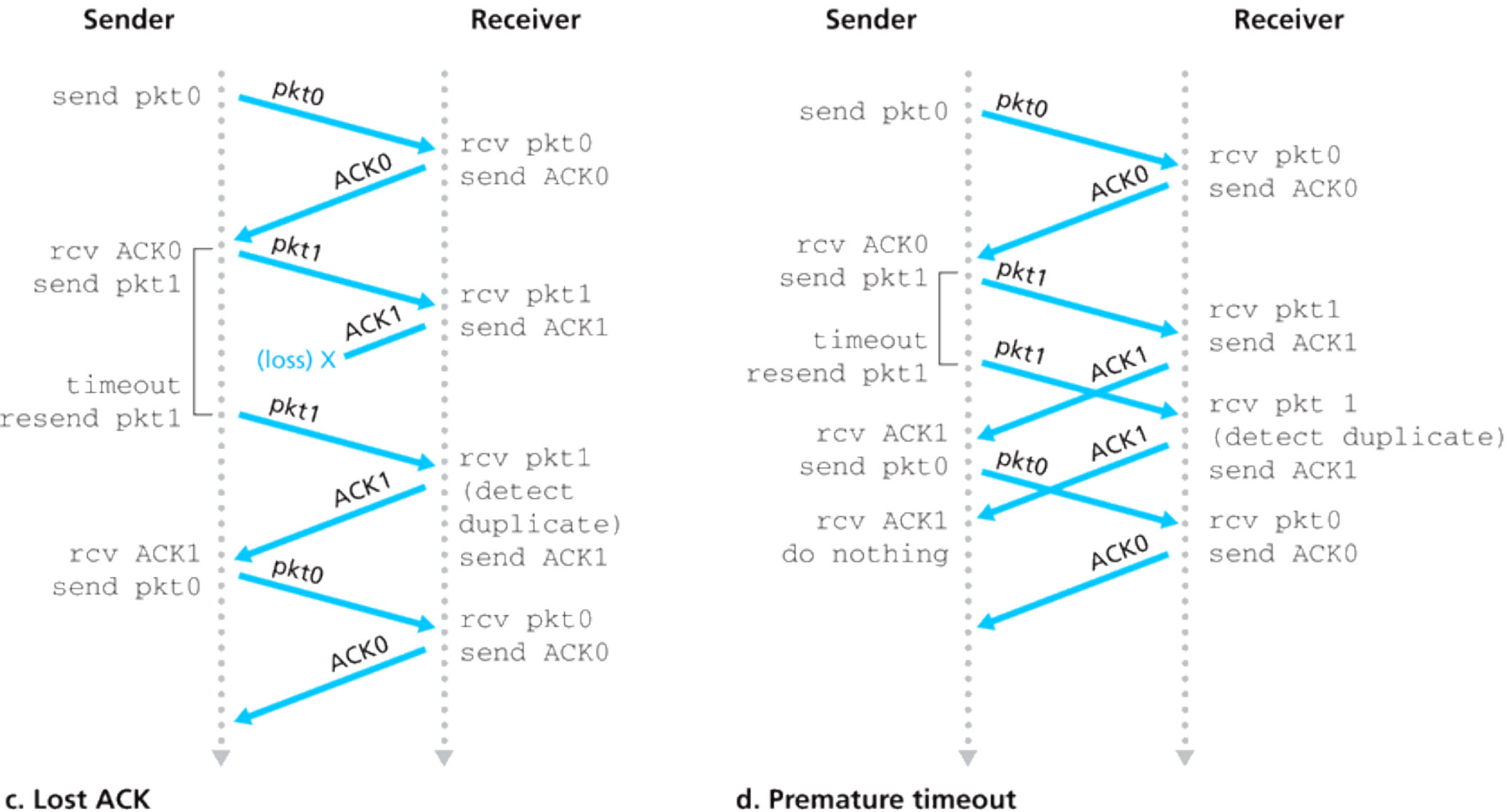
Cas 4 : canal avec perte des paquets

Déroulement : perte des données

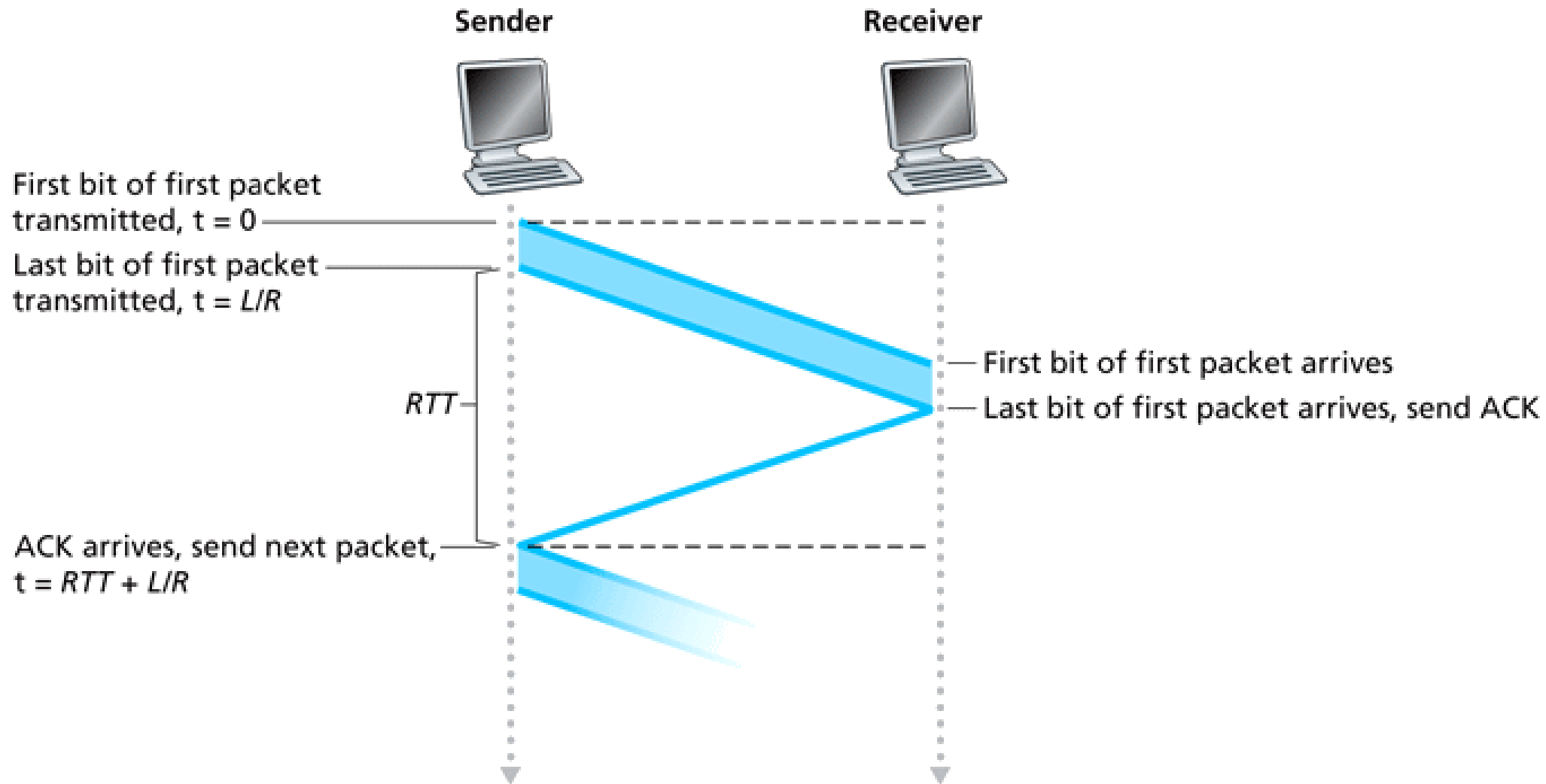


Cas 4 : canal avec perte des paquets

Déroulement : perte des acquittements ou *timeout*

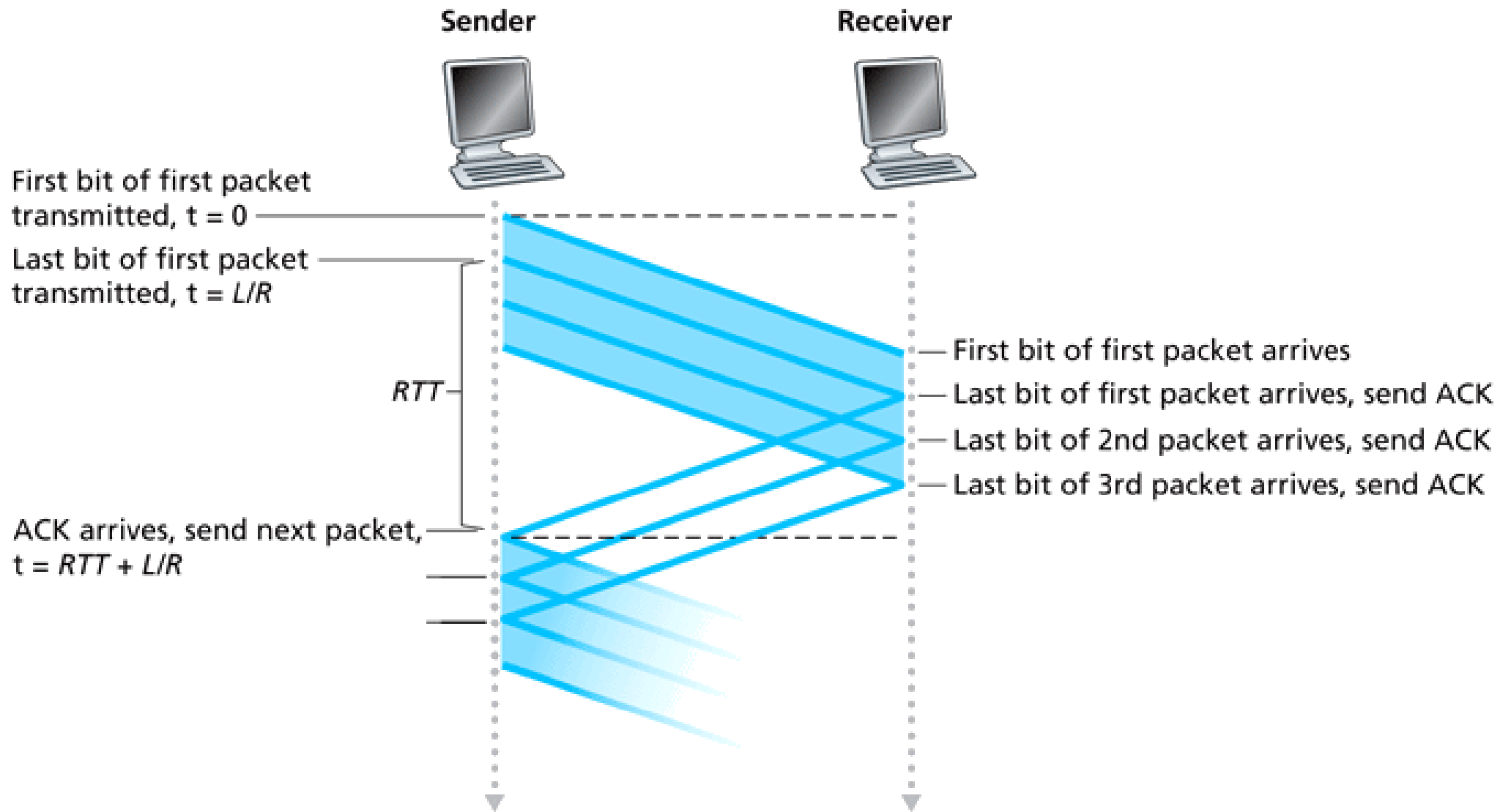


Principe du protocole *stop and wait*



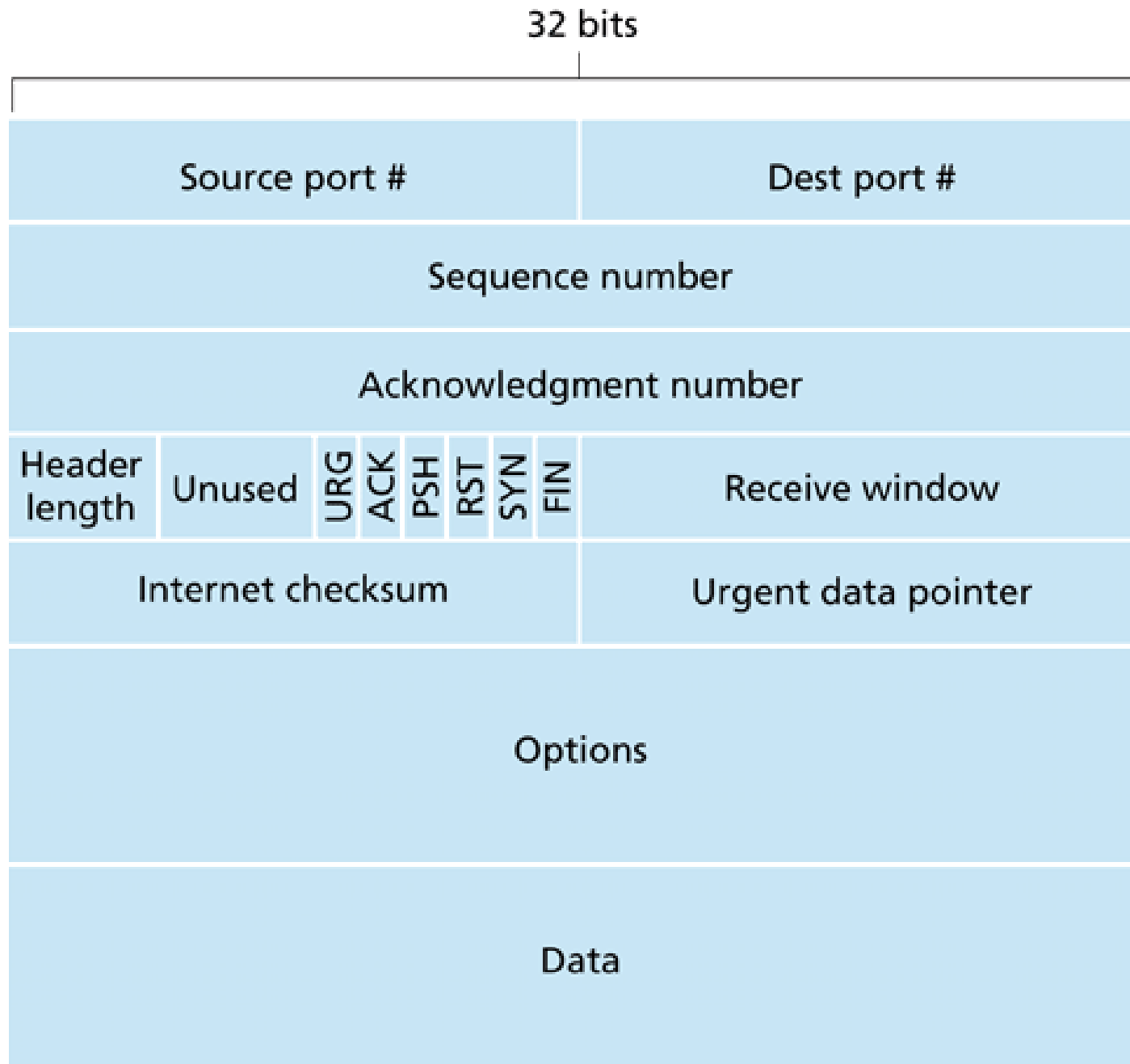
a. Stop-and-wait operation

Traitement en chaîne (*Pipelining*)



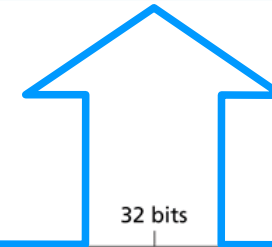
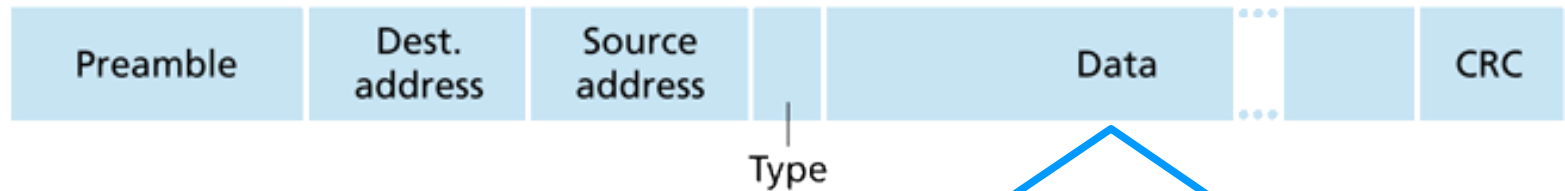
b. Pipelined operation

Segment TCP



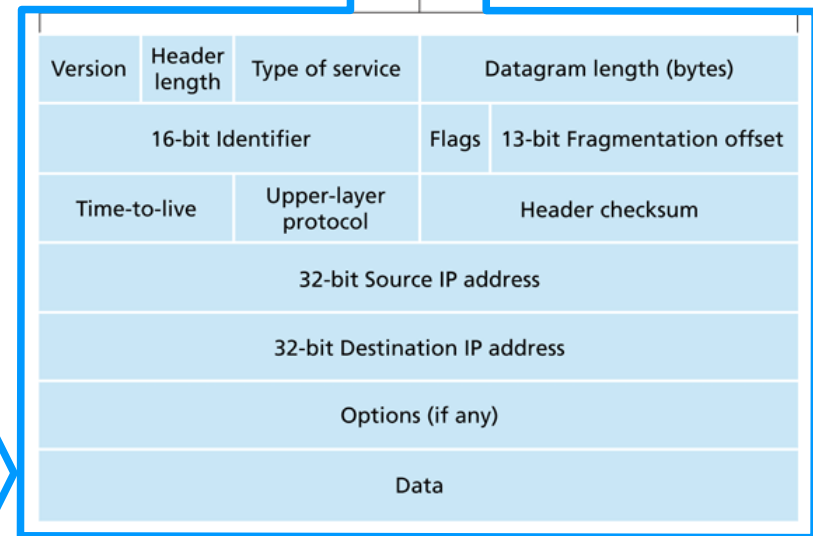
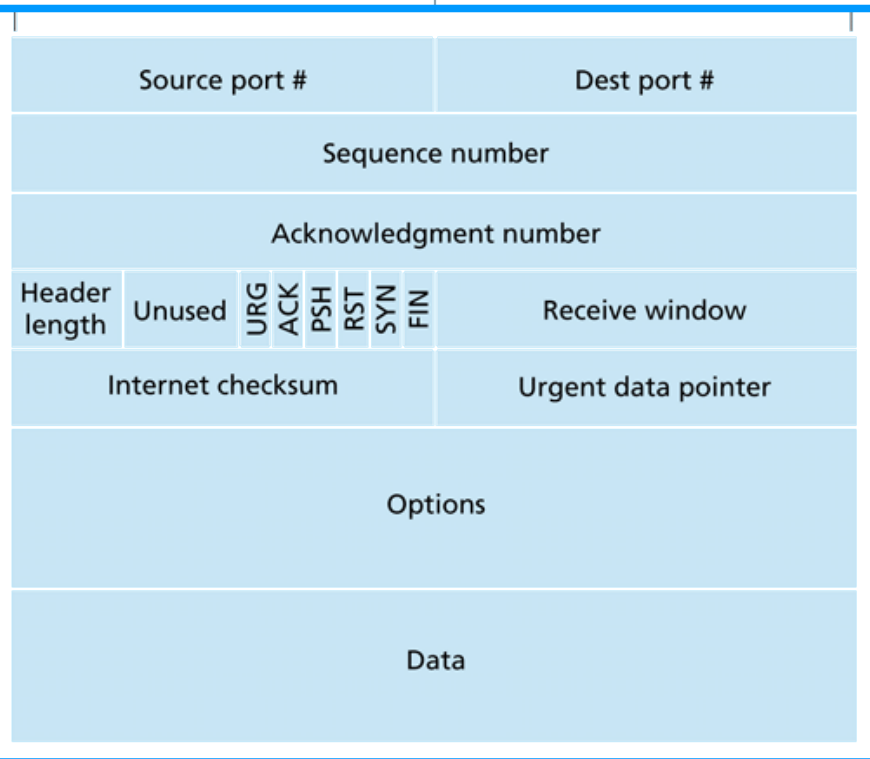
Encapsulation des segments TCP

Trame Ethernet (couche liaison)



32 bits

32 bits



Datagramme IP (couche réseau)

Conclusions

Couche transport

- Service liaison entre processus

Service simple et rapide mais non fiable UDP

- Conditionner par la qualité de service des couches inférieures

Service plus coûteux mais fiable TCP

- même si le réseaux sous-jacent n'est pas fiable

Fiabilité de TCP repose sur

- Checksum
- Temporisateurs
- Numéro de séquence
- Acquittements
- Pipelining

Fourni une API pour le développement d'applications !