

Guide TCP Stack

1. Introduction.

Ce document explique comment développer un stack TCP.

2. Variables de gestion des séquences.

2.1 Variables d'émission.

Vous trouverez ci-dessous la liste des variables d'émission :

- snd_nxt : Le prochain numéro de séquence à envoyer.
- snd_una : Le plus ancien octet non acquitté. C'est le début de la fenêtre d'envoi.
- snd_wnd : La fenêtre d'envoi annoncée par l'hôte distant. Elle indique combien d'octets tu peux envoyer.

Vous trouverez ci-dessous la liste des variables d'émission :

- rcv_nxt : Le prochain numéro de séquence attendu.
- rcv_wnd : Ta fenêtre de réception. Elle indique combien d'octets tu peux encore recevoir dans ton buffer.

2.2 Algorithme de traitement.

SI Réception :

1. Vérifier que Seq == rcv_nxt (sinon ignorer ou bufferiser)
2. Traiter les flags :
 - SI SYN : rcv_nxt = Seq + 1
 - SI PSH : rcv_nxt = Seq + L
 - SI FIN : rcv_nxt = Seq + 1
3. Mettre à jour rcv_wnd (espace restant dans buffer)
4. Décider si on doit envoyer un ACK

SI Émission :

1. Construire le segment :
 - Seq = snd_nxt
 - Ack = rcv_nxt
 - Window = rcv_wnd
2. Ajouter les flags appropriés (SYN, ACK, PSH, FIN)
3. Mettre à jour snd_nxt :
 - SI SYN : snd_nxt += 1
 - SI données : snd_nxt += L
 - SI FIN : snd_nxt += 1
4. Sauvegarder le segment pour retransmission si nécessaire
5. Démarrer un timer de retransmission

SI Réception d'un ACK :

1. SI Ack > snd_una :
 - snd_una = Ack (données acquittées)
 - Supprimer segments acquittés du buffer de retransmission
 - Annuler les timers correspondants
2. Mettre à jour snd_wnd (fenêtre annoncée par l'hôte distant)

2.3 Fonctionnement en mode Serveur.

#	Sens	Flags	Seq	Ack	Action	Vérification	rcv_wnd	snd_wnd
1	Réception	SYN	X	0	rcv_nxt = seq + 1		RCV_WND_INIT	WIN_CLIENT
2	Émission	SYN ACK	Y	X + 1	snd_nxt = ISS seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT	WIN_CLIENT
3	Réception	ACK	X + 1	Y + 1	snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT	WIN_CLIENT
4	Réception	PSH ACK	X + 1	Y + 1	rcv_nxt = seq + L1 snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L1	WIN_CLIENT
5	Émission	ACK	Y + 1	X + 1 + L1	seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT - L1	WIN_CLIENT
6	Émission	PSH ACK	Y + 1	X + 1 + L1	seq = snd_nxt ack = rcv_nxt snd_nxt += L2		RCV_WND_INIT - L1	WIN_CLIENT
7	Réception	ACK	X + 1 + L1	Y + 1 + L2	snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L1	WIN_CLIENT
8	Réception	FIN ACK	X + 1 + L1	Y + 1 + L2	rcv_nxt = seq + 1 snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L1	WIN_CLIENT - L2
9	Émission	ACK	Y + 1 + L2	X + 2 + L1	seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT - L1	WIN_CLIENT - L2
10	Émission	FIN ACK	Y + 1 + L2	X + 2 + L1	seq = snd_nxt ack = rcv_nxt snd_nxt += 1		RCV_WND_INIT - L1	WIN_CLIENT - L2
11	Réception	ACK	X + 2 + L1	Y + 2 + L2	snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L1	WIN_CLIENT - L2

2.3 Fonctionnement en mode Client.

#	Sens	Flags	Seq	Ack	Action	Vérification	rcv_wnd	snd_wnd
1	Émission	SYN	X	0	snd_nxt = ISS seq = snd_nxt snd_nxt += 1		RCV_WND_INIT	
2	Réception	SYN ACK	Y	X + 1	rcv_nxt = seq + 1 snd_una = ack	ack == snd_nxt	RCV_WND_INIT	WIN_SERVER
3	Émission	ACK	X + 1	Y + 1	seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT	WIN_SERVER
4	Émission	PSH ACK	X + 1	Y + 1	seq = snd_nxt ack = rcv_nxt snd_nxt += L1		RCV_WND_INIT	WIN_SERVER
5	Réception	ACK	Y + 1	X + 1 + L1	snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT	WIN_SERVER - L1
6	Réception	PSH ACK	Y + 1	X + 1 + L1	rcv_nxt = seq + L2 snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L2	WIN_SERVER - L1
7	Émission	ACK	X + 1 + L1	Y + 1 + L2	seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT - L2	WIN_SERVER - L1
8	Émission	FIN ACK	X + 1 + L1	Y + 1 + L2	seq = snd_nxt ack = rcv_nxt snd_nxt += 1		RCV_WND_INIT - L2	WIN_SERVER - L1
9	Réception	ACK	Y + 1 + L2	X + 2 + L1	snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L2	WIN_SERVER - L1
10	Réception	FIN ACK	Y + 1 + L2	X + 2 + L1	rcv_nxt = seq + 1 snd_una = ack	seq == rcv_nxt ack == snd_nxt	RCV_WND_INIT - L2	WIN_SERVER - L1
11	Émission	ACK	X + 2 + L1	Y + 2 + L2	seq = snd_nxt ack = rcv_nxt		RCV_WND_INIT - L2	WIN_SERVER - L1