

Descriptif des fichiers de tests

Note : Lors des phases où les requêtes de connexions (CR) ont déjà été initié, les entités Initiator et Responder initialisent chacun leur numéros de séquence à 1 comme indiqué dans la spécification, dans la mesure où il s'agit du premier échange de la transaction.

Fichier correct_CR_IDIS:

Le diagramme MSC de ce fichier couvre l'ensemble des messages échangés lors d'une demande de déconnexion de l'utilisateur coté Responder après avoir reçu une requête demande de connexion CR de la part de Initiator.

Le schéma est le suivant :

- Initiator reçoit une demande de connexion de son utilisateur.
- Initiator envoi à Responder une requête de connexion CR.
- Le Responder reçoit la requête CR de Initiator à travers le médium.
- Le Responder indique la requête de connexion à son Utilisateur.
- Celui ci répond par une directive IDISreq à Responder.
- Le responder envoi alors à Initiator une requête DR et retourne en mode deconnexion. (représenté par l'état idle).
- Initiator, sur réception de la requête DR passe en mode déconnexion. (représenté par l'état idle).

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_CR:

Le diagramme MSC associé à ce fichier couvre l'ensemble des messages échangés décrivant la procédure de connexion, qui conduit les deux interlocuteurs dans le mode data_transfert, où ils sont prêts à envoyer et acquitter.

Le schéma est le suivant:

- Initiator reçoit une demande de connexion de son utilisateur.
- Initiator envoi à Responder une requête de connexion CR.
- Le Responder reçoit la requête CR de Initiator à travers le médium.
- Le Responder indique la requête de connexion à son Utilisateur.
- Celui ci répond par une directive ICONresp acceptant la demande de connexion.
- Responder envoi alors à Initiator une requête CC signifiant qu'il est prêts à recevoir et arrive dans l'état "data_transfert".
- Initiator en reception de la requête CC envoi une requête ICONconf à son Utilisateur signifiant la confirmation de l'échange, puis entre dans l'état data_transfert, dans l'attente d'une requête de transfert venant de son Utilisateur.

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_DT1:

Le diagramme MSC associé à ce fichier tient compte du fait que les deux entités Initiator et Responder se trouvent déjà en mode data_transfert. L'ensemble de tests se fait sur 2 échanges (un échange étant un envoi d'un DT(i) et d'un acquittement AK(i) avec $i \in [0;1]$)

Le schéma est le suivant:

- Initiator reçoit une requête de transfert de la part de son Utilisateur, contenant la donnée à transmettre.
- Initiator encapsule la données dans une requête de type DT, avec le bon numéro de séquences puis l'envoi à Responder.
- Responder reçoit la requête DT de Initiator à travers le medium, vérifie que le numéro de séquence soit correct. Il l'est. Responder récupère la donnée contenu dans le message et l'envoi à son Utilisateur.
- Responder renvoi une requête AK avec le même numéro de séquence du DT précédemment reçu.
- Initiator reçoit l'acquittement AK avec le bon numéro de séquence.

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_DT2:

Le diagramme MSC associé à ce fichier tient compte du fait que les deux entités Initiator et Responder se trouvent déjà en mode "data_transfert". Dans ce tests ici, le schéma est similaire au cas de tests du fichier ~_DT1, mais étend le tests à 4 échanges (un échange étant un envoi d'un DT(i) et d'un acquittement AK(i) avec $i \in [0;1]$) afin de s'assurer que les numéros de séquences alternent correctement au delà de 3 échanges.

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_Retransmit_4TIMES_CR:

Le diagramme MSC de ce fichier couvre l'ensemble des messages échangés lors d'une ré-émission de demande de connexion de la part de l'entité Initiator à l'entité Responder.

- Initiator reçoit une demande de connexion de son utilisateur.
- Initiator envoie à Responder une requête de connexion CR et active son timer pour une durée de 5 secondes.
- Initiator attend la fin du timer. Puis Initiator retransmet une requête CR et incrémente le nombre de tentatives de 1.
- Lorsque le nombre de tentative atteint 4, Initiator passe en mode déconnexion.

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_Retransmit_4TIMES_DT:

Le diagramme MSC de ce fichier couvre l'ensemble des messages échangés lors d'une ré-émission de transfert de données en l'absence de réception d'un acquittement, de la part de l'entité Initiator à l'entité Responder.

- Initiator reçoit une requête de transfert de la part de son Utilisateur, contenant la donnée à transmettre.
- Initiator encapsule la données dans une requête de type DT, avec le bon numéro de séquences puis l'envoi à Responder et active son timer pour une durée de 5 secondes.
- Initiator attend la fin du timer. Puis Initiator retransmet une requête DT avec le même numéro de séquence que la tentative précédente et incrémente le nombre de tentatives de 1.
- Lorsque le nombre de tentative atteint 4, Initiator passe en mode déconnexion.

Le système doit renvoyer "PASS" sur ce cas de test.

Note : Les fichiers pour les timers sont absents mais les mscs peuvent être facilement décrites. Supposons que les fichiers soient présents:

Fichier correct_Timer_CR:

Le diagramme MSC de ce fichier couvre l'ensemble des messages échangés lors d'une ré-émission de demande de connexion de la part de l'entité Initiator à l'entité Responder.

- Initiator reçoit une demande de connexion de son utilisateur.
- Initiator envoie à Responder une requête de connexion CR et active son timer pour une durée de 5 secondes.
- Le timer de Initiator se déclenche 5 secondes plus tard.
- Initiator effectue alors l'opération de retransmission.

Le système doit renvoyer "PASS" sur ce cas de test.

Fichier correct_Timer_DT:

Le diagramme MSC de ce fichier couvre l'ensemble des messages échangés lors d'une ré-émission de transfert de données en l'absence de réception d'un acquittement, de la part de l'entité Initiator à l'entité Responder.

- Initiator reçoit une requête de transfert de la part de son Utilisateur, contenant la donnée à transmettre.
- Initiator encapsule la données dans une requête de type DT, avec le bon numéro de séquences puis l'envoi à Responder et active son timer pour une durée de 5 secondes.
- Le timer de Initiator se déclenche 5 secondes plus tard.
- Initiator effectue alors l'opération de retransmission.

Le système doit renvoyer "PASS" sur ce cas de test.