



Redes de Computadores

Grado en Ingeniería Informática.

Curso 2019/2020

PROBLEMA PROPUESTO

Sea un protocolo de comunicación bidireccional en la capa n de una arquitectura de red. Se trata de un protocolo de intercambio de datos orientado a conexión. Su esquema general de funcionamiento se basa en el establecimiento de una conexión previa al envío de datos de emisor a receptor, y liberación de la conexión después de la transmisión.

El elemento emisor del protocolo se encuentra inicialmente a la espera de solicitudes de envío de datos por parte del nivel $n+1$. Si el nivel $n+1$ solicita el envío de un bloque de información, el emisor envía una petición de conexión al receptor pasando a esperar una respuesta. Si la respuesta es un rechazo de conexión, el emisor informa al nivel $n+1$ de un error en la conexión y pasa a esperar nuevas solicitudes de envío de bloques de información del nivel superior. Si la respuesta es una aceptación de la conexión, el emisor envía el bloque de datos y espera una confirmación de la recepción correcta. Al enviar el bloque de datos, el emisor también activa un temporizador de espera de la confirmación. Si expira el temporizador sin haber llegado la confirmación, el emisor reenvía el bloque de datos transmitido. Si antes de expirar el temporizador llega la confirmación del bloque de datos enviado, el emisor informa al nivel $n+1$ del éxito en la transmisión del bloque y envía una solicitud de desconexión. Cuando el emisor recibe una confirmación de la desconexión por parte del receptor, vuelve al estado inicial de espera de solicitudes del nivel $n+1$.

El elemento receptor del protocolo se encuentra inicialmente a la espera de peticiones de conexión. Cuando recibe una petición de conexión, informa al nivel $n+1$ y espera una respuesta del mismo. Si el nivel $n+1$ no acepta la conexión, se envía un rechazo de conexión al emisor y se vuelve al estado inicial. Si el nivel $n+1$ acepta la conexión, se envía una aceptación de la conexión al emisor y se pasa a esperar un bloque de datos. Cuando se recibe el bloque de datos, se envía una confirmación de la recepción, se envían los datos recibidos al nivel $n+1$ y se pasa a esperar una petición de fin de conexión. Si la confirmación no llega al emisor, el receptor recibirá datos nuevamente del emisor. En ese caso, el receptor rechaza los datos y vuelve a enviar la confirmación, pasando a esperar una petición de fin de conexión. Cuando la petición de fin de conexión llega al receptor, se confirma, volviendo al estado inicial.

Determina los estados, eventos de entrada y salida, y la MEF que describe el funcionamiento del elemento RECEPTOR del protocolo.

Estados Receptor

EC → Receptor espera petición de conexión.

ERNS → Receptor espera respuesta del nivel superior ($n+1$) a una petición de conexión.

ED → Receptor espera un bloque de datos.

EF → Receptor espera una petición de desconexión.

Eventos de entrada

PET_CON_IN → Receptor recibe una petición de conexión.
 NO_CON → El nivel n+1 rechaza la petición de conexión.
 SI_CON → El nivel n+1 acepta la petición de conexión.
 DATA_IN → El receptor recibe un bloque de datos.
 FIN_IN → El receptor recibe una petición de desconexión.

Eventos de salida

CON_NS → El receptor informa al nivel superior (n+1) de una petición de conexión.
 ACK_CON_OUT → El receptor envía una aceptación de la conexión.
 REJ_CON_OUT → El receptor envía un rechazo de la conexión.
 ACK_OUT → El receptor envía una confirmación de bloque de datos.
 REJ_DATA → El receptor rechaza unos datos duplicados.
 DATA_NS → El receptor envía el bloque de datos al nivel superior (n+1).
 ACK_FIN_OUT → El receptor envía una confirmación de desconexión.

MEF

