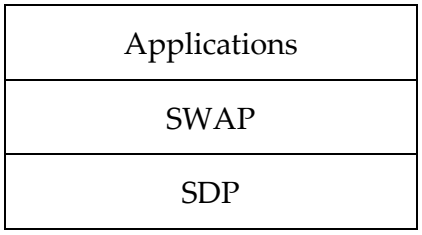
Project Part 3: Background Information

Read each of the sections below before you start your project. You will need this information to complete Part 3 of the project. Consult your Open Learning Faculty Member if you have any questions.

SWAP: A Simple Data Link Layer Protocol

In Project Part 3, you will design and implement a reliable block transfer protocol, called Stop-and-Wait ARQ Protocol (SWAP), which uses another protocol called Simple Datalink Protocol (SDP). As SDP is a non-reliable block transfer protocol, you will also implement an application, called File Transfer Application (FTA), to send a file to another computer using SWAP, as shown below:



SDP supports the following Application Programming Interfaces (APIs), and SDP is already completely implemented in the file sdp.c posted in the Project section of your course. These APIs are used by SWAP.

int sdp_send (int *sd*, char **buf*, int *length*): It writes data of *length*, which is stored in *buf*, to the session *sd*. *sd* is the value returned from swap_open() or swap_wait(). The maximum size that it can send is 256 bytes. It returns the number of bytes written to the session *sd* if there is no error, otherwise -1.

int sdp_receive (int *sd*, char **buf*): It reads data and stores it in *buf*, from the session *sd*, which is returned from swap_open() or swap_wait(). The maximum size that it can receive is 256 bytes. And it returns the number of bytes read from the session *sd* if there is no error, -2 if the session is disconnected, and -1 for other general errors.

int sdp_receive_with_timer (int *sd*, char **buf*, unsigned int *expiration*): It reads data and stores it in *buf*, from the session *sd*, which is returned from swap_open() or swap_wait(). *expiration* is used to set a timer and the unit is a millisecond. The maximum size that it can receive is 256 bytes. It returns the number of bytes read from the session *sd* if there is no error -3 if the timer expires, -2 if the session is disconnected, and -1 for other general errors.

SWAP supports the following APIs, and SWAP is implemented in two files swap_client.c and swap_server.c posted in the Project section of your course. These APIs are used by applications.

int swap_wait (unsigned short *server_port*): It is used by a SWAP server to wait for a reliable session connection request from a SWAP client. The port number *server_port* is in the network byte order. It returns an identifier, called session descriptor, used for data exchange if there is no error, otherwise -1.

int swap_open (unsigned int *destination_address*, unsigned short *server_port*): It is used by a SWAP client to open a reliable session to a SWAP server of the destination IP address *destination_address* in the network byte order with the port number.