```
type mkind = enum(SEND, CREATE_CHAN, CHAN_DONE);
bool writing = false;      # status of network write
other variables for the write queue and transmission buffers;

proc netWrite(int dest; mkind kind; byte data[]) {
  acquire buffer; format message and store it in the buffer;
  if (writing)
    insert the message buffer on the write queue;
  else {
    writing = true;
    start transmitting the message on the network;
  }
}

netWrite_handler: {  # entered with interrupts inhibited
  save state of executing;
  free the current transmission buffer;
  writing = false;
  if (write queue not empty) { # start another write
    remove first buffer from the queue; writing = true;
    start transmitting the message on the network;
  }
  dispatcher();
}

netRead_handler: {   # entered with interrupts inhibited
  save state of executing;
  acquire new buffer; prepare network controller for next read;
  unpack first field of input message to determine kind;
  if (kind == SEND)
    remoteSend(channel name, buffer);
  else if (kind == CREATE_CHAN)
    remoteCreate(rest of message);
  else  # kind == CHAN_DONE
    chanDone(rest of message);
}
```

Figure 10.3 Network interface routines.