Power to the Protocolariat

A COMP3980 protocol

Team: Tim Bruecker, Keir Forster, John Tee, Alex Xia
Image credit:
http://www.stridentconservative.com/wp-content/uploads/2016/05/Power-to-the-Workers.jpg

State Diagram(s)

Figure 1. Revision 1. Retransmit sends ACK. Send cannot TOS, so that removed. Start state added to please Goran.

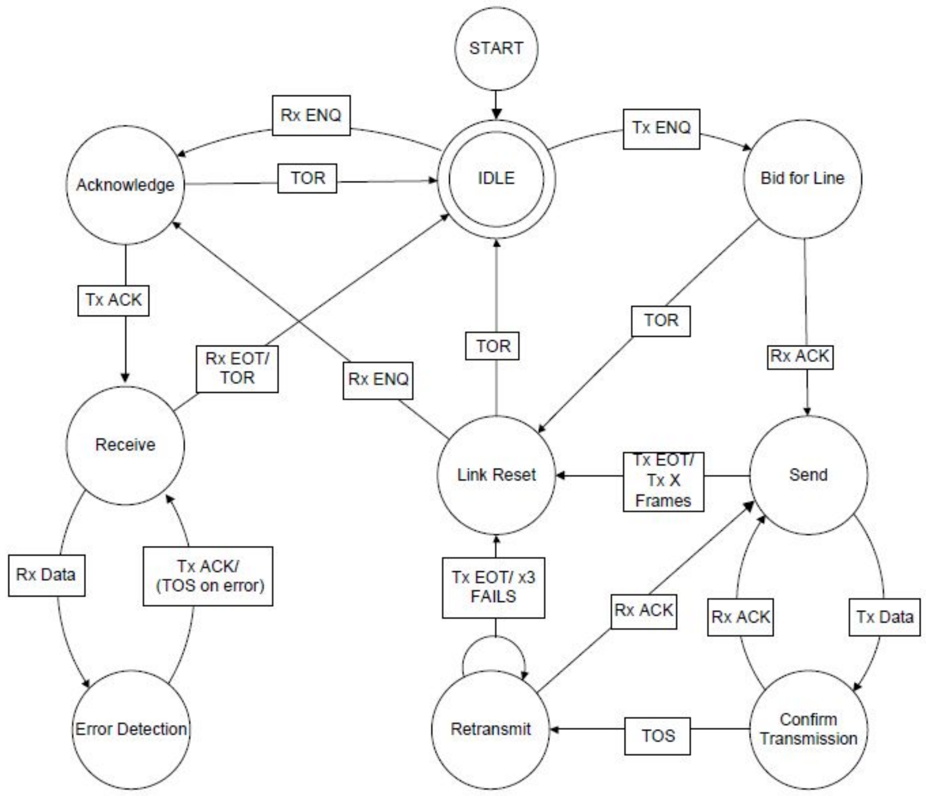


Figure 2. Our implementation of the Application Activity version 1

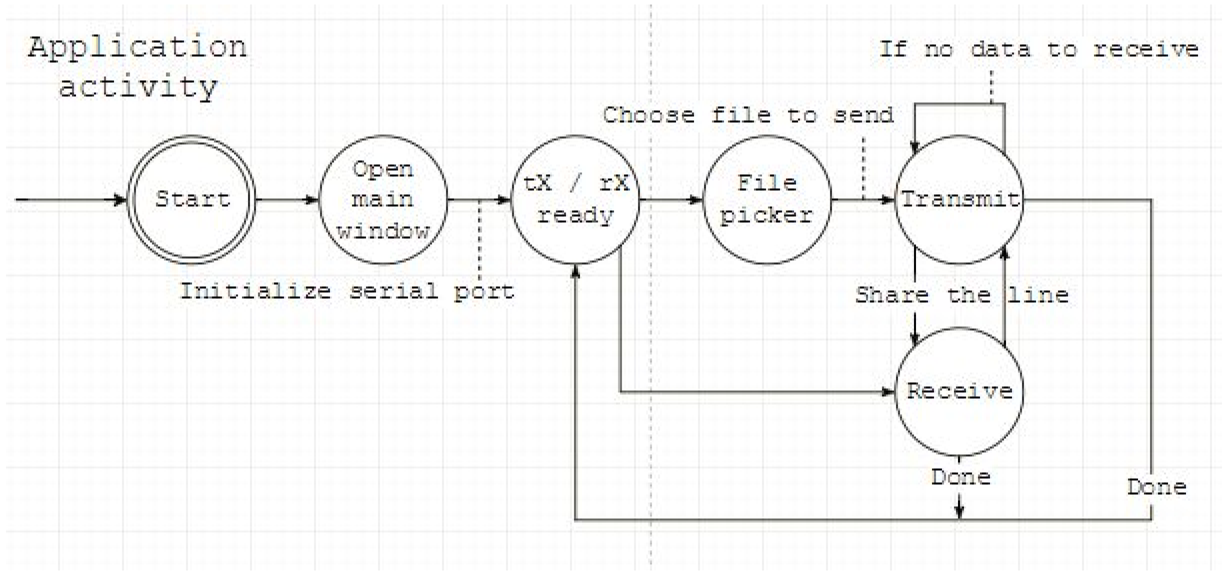


Figure 3. Our implementation of the Application Activity version 2

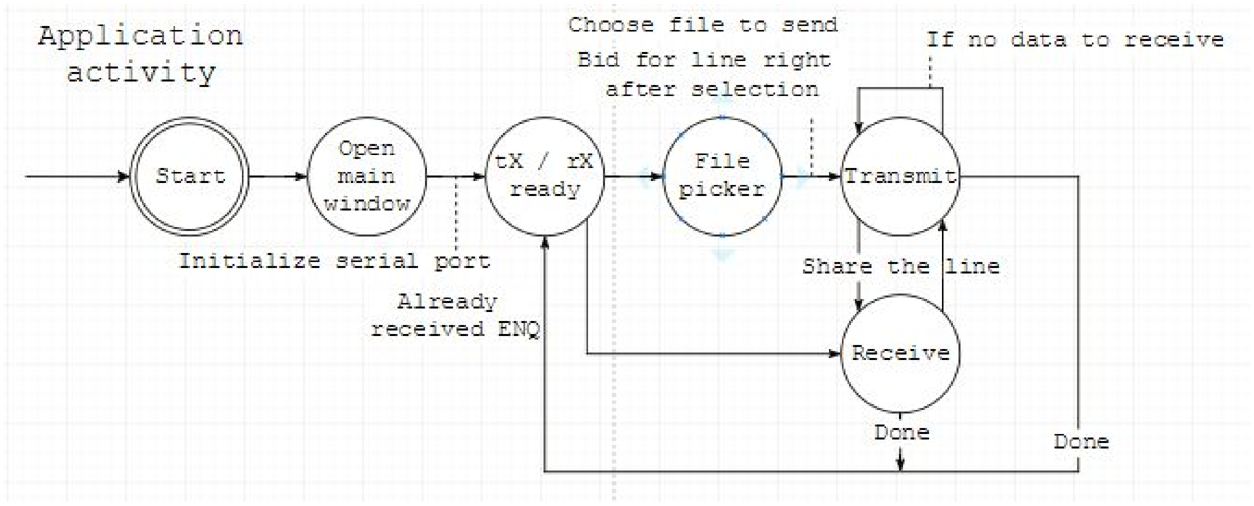


Figure 4. Our implementation of the Send and Receive Activities version 1

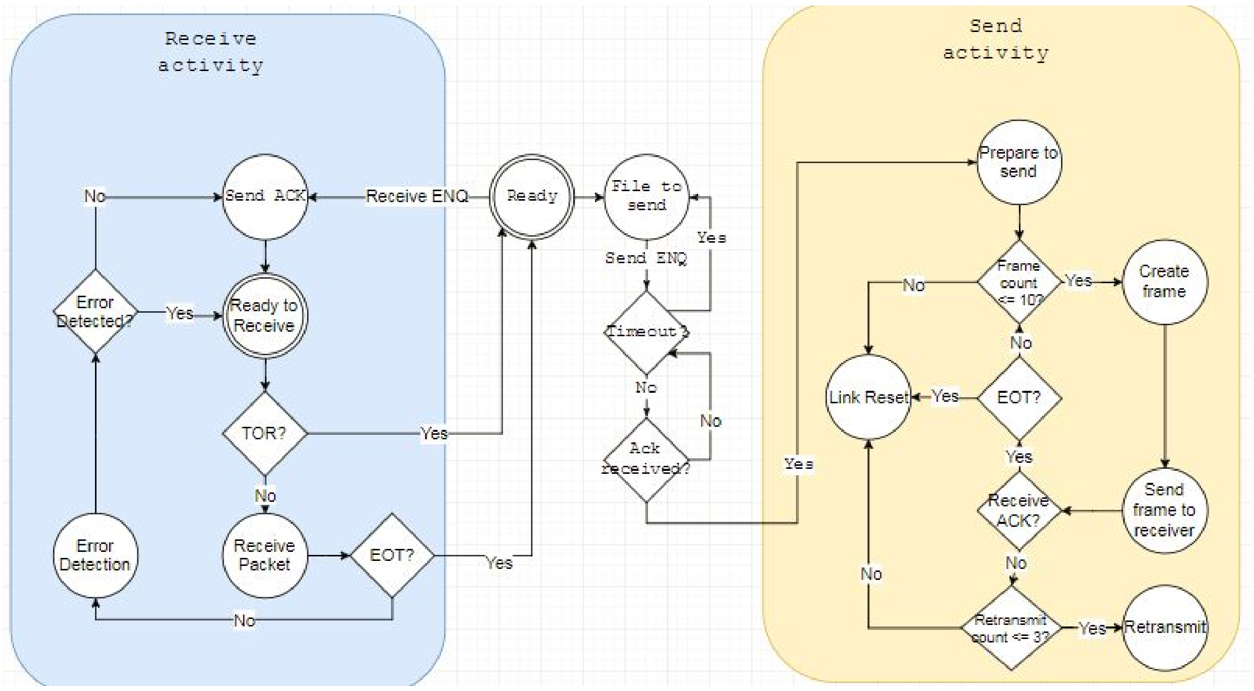
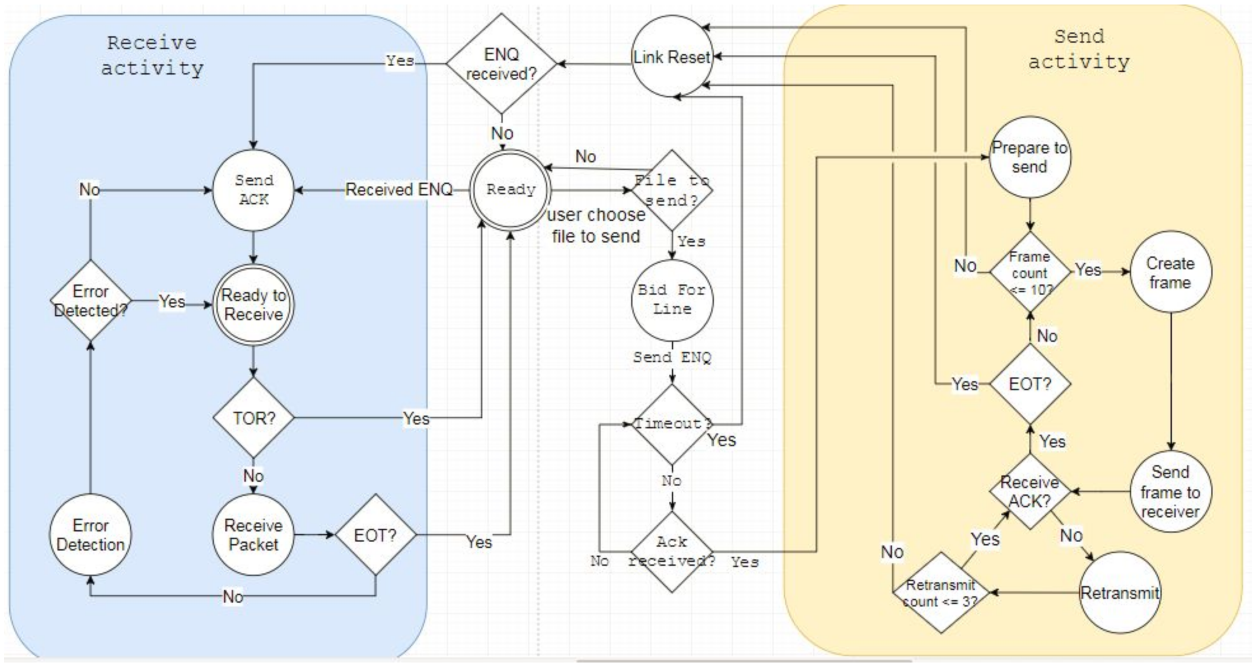


Figure 5. Our implementation of the Send and Receive Activities version 2



Pseudo Code

//starts the program

//the only function in its file

```
Main
    Create local reference to Application class
    Create new Application class object
    For every “Create local/instance reference to...” below, unless
explicitly stated not to, also create the new instance there
    Exit
```

//In Application Activity Class

```
ApplicationClass Constructor
    Create instance variable reference to mainWindow UI object
    Initialize any mainWindow UI elements to start-state:
        Enable FilePicker button
        Disable disconnect button
        Disable send button
    Create listeners linking mainWindow UI elements to functions:
        On click FilePicker button
            Show FilePicker window
        On click disconnect button
            Call Disconnect function
        On click send button
            Call FileToSend function
    Start mainWindow instance on UI thread
    Call mainWindow’s display function
    Create instance variable reference to serial port:
        leave it uninitialized
```

//In FilePicker class
//FilePicker class Should be a UI element / UI element controller

```
FilePicker Constructor
    Create instance variable reference to Send Activity object
        Pass in reference to serial port as argument
    Set an instance variable to ApplicationClass using passed-in
    argument
    Initialize UI elements, ie confirm, cancel
    Set listener to confirm button
        On confirm button click:
            Set chosen file as instance reference in
            Application Activity
            //Call FileToSend function
```

//In Application class

```
FileToSend
    Call Application Class’s ConnectPort function
    Call Application Class’s BidForLine function
    Run BidForLine on background thread
```

```
          If returned bid success
               Set instance varibale is transmitting to true
               Create instance variable to user’s file
               Create instance variable reference to empty file
          buffer
               Parse file to buffer
                    Handle IO and File Not Found exceptions
               Call Send Activity’s PrepareToSend function
               Run PrepareToSend on background non-UI thread
                    Pass in user file buffer as argument
          //else do nothing, BidForLine would just timeout
```

//In Application Activity Class

```
ConnectPort
     Open serial port
          handle any null pointer and IO exceptions
     Set the global serial port reference to the opened serial port
     Disable Send button UI element
     Disable FilePicker UI element
     Enable Disconnect button UI element

     Create reference to Receive Activity object
     Set on RX ready listeners to serial port
          //frames received
          On RX ready:
               If instance variable isTransmitting is false
                    Call ReadData to handle it
                    Call Receive Activity’s RECEIVE function
                    //Receive Activity has no ref to application
     class
                    //so once receive finishes should call
     disconnect here
                    Call Disconnect function
               Else
                    Set instance variable has received ENQ to true
```

//In Application Activity Class

```
TXRXReadyAgain
     //allow user to select UI elements to allow for TX again
     Disable mainWindow’s disconnect button
     Enable mainWindow’s Send button UI element
     Enable mainWindow’s FilePicker UI element
```

//In ApplicationClass
```
Disconnect
    Close serial port
        Handle IO and File TX Interruption Exceptions
    //allow user to be able to start new connection
    TXRXReadyAgain
```

//In Application Activity Class
```
BidForLine
    Create local reference to new timer
    Loop forever
        Start timer
            On timeout
                Call Application Class's linkReset function
                Run linkReset on non-UI thread
                Exit loop
        Create new 2-Byte control frame, EnqFrame
        Set EnqFrame's header field to ENQ Ascii char
        Send EnqFrame to serial port
        Set RX listener to serial port
            On get frame:
                Stop timer
                Isolate frame's header
                //check if control frame
                If frame size = 2 Byte
                //check if control frame = ACK
                AND header char = ACK Ascii char
                    Return Bid success
                    Exit loop
                Else
                    Restart loop and timer
```

//In SendActivity class
```
SendActivity Constructor
    Passed in reference to serial port, set it as global variable
```

//In SendActivity class
//Passed in buffer holding user's chosen file
```
PrepareToSend
    Create instance reference to file buffer holding user file
    Create local variable frameCount
    Initialize frameCount to 0
    Create local reference to new timer
    While frameCount <= 10
    Do
         Create new 518-byte empty data frame, sentFrame
         Set new frame header field to STX Ascii char
         Fill new frame data field with bytes from user file buffer
         Generate new CRC from frame data
         Set new frame CRC field to new CRC
         Pass new frame to serial port
         Set RX listener to serial port
              On get frame:
                   Stop timer
                   Isolate frame’s header
                   //check if control frame
                   If frame size = 2 Byte
                   //check if control frame = ACK
                   AND header char = ACK Ascii char
                        If last byte added to sentFrame=EOT
                             //finished receiving
                             Call ApplicationClass’s linkReset
                   func
                             Run linkReset on current thread
                        Else
                             Increment frameCount
                             Continue loop
         Start timer
              On timeout
              Call Retransmit and pass in created frame as argument
                   //retransmit good, send again
                   If returned success
                        Increment frameCount
                        Continue loop
                   //retransmit 3 times failed
                   If returned fail
                        Call ApplicationClass’s linkReset func

    Endwhile
    //used up all 10 frames
    Call ApplicationClass’s linkReset func
```

//In SendActivity class

```
Retransmit
    Initialize attemptCount to 0
```

```
     Create local reference to passed in frame
     Create local reference to new timer
     While attemptCount <= 3
     Do
          Start timer
               On timeout
                    increment attemptCount
                    Continue loop
          Pass the passed-in frame to serial port
          Set RX listener to serial port
               On get frame:
                    Isolate frame's header
                    //check if control frame
                    If frame size = 2 Byte
                    //check if control frame = ACK
                    AND header char = ACK Ascii char
                         If last byte added to sentFrame=EOT
                              //retransmit success, prepare to send
                              //next frame
                              //only stop timer on ACK, nothing
                         else
                              Stop timer
                              Return success
                         //else dont do anything
                         //let timer run down until get next frame
     Endwhile
     //used up all 3 attempts
     Return fail
```

//In Application Activity Class

```
linkReset
     //check for instance variable has received ENQ is true
     If has received ENQ = true
          Call Receive Activity class' RECEIVE function
     If has received ENQ = false
          //application was in middle of TX
          If is transmitting = true
               //cant directly call PrepareToSend
               //need to bid for line again
               Call FilePicker's FileToSend

          //application finished TX
          If is transmitting = false
               Call TXRXReadyAgain function
```

//In Receive Activity class

```
ReceiveActivty Constructor
     Passed in reference to serial port, set it as global variable
```

//In Receive Activity class

```
RECEIVE
     Create new 2-Byte control frame, AckFrame
     Set AckFrame's header field to ACK Ascii char
     Pass AckFrame to serial port

     Set RX listener to serial port
          On get frame:
               Stop timer
                    Call ReceivePacket function
                         Pass in frame as argument
                    Restart timer
          Create local reference to new timer
          Start timer
               On timeout
                    //go back in stack to main
                    return
```

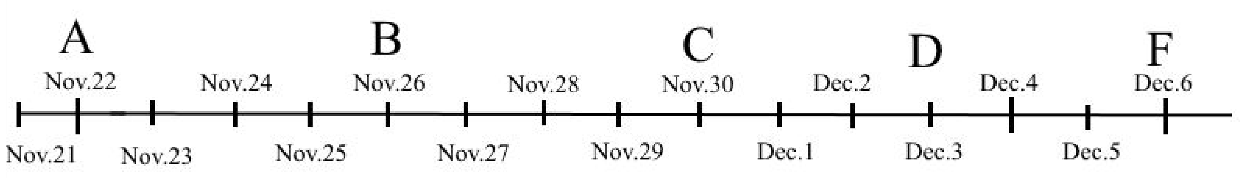
//In Receive Activity class

```
ReceivePacket
     If list of received frames has not been created yet
          Create instance reference to empty list of received frames
     Passed in a frame as argument
     If passed in frame's last byte = EOT ascii char
          //TODO Goto main ready?
     Else
          Call ErrorDetection function
               Pass in frame as argument
          If success
               Add frame to list of received frames
               Create new 2-Byte control frame, AckFrame
               Set AckFrame's header field to ACK Ascii char
               Pass AckFrame to serial port
          //else, do nothing and wait for retransmit on sender side
```

//In Receive Activity class
//would return success if no error found

```
ErrorDetection
    Passed in frame as argument
    Retrieve value in passed in frame’s CRC field
    Perform CRC on value retrieved
    //if no error found
    If CRC result = 0
        Return success
    Else
        Return fail
```

Project Timelines and Deadline



Milestone	Time	Details
A	Nov. 22nd 9:00AM	Design deadline - All design work (this document) is due - Experiment with
B	Nov. 26th	Tentative End of experimentation - Should have already figured out details of func calls - Start serious implementation
C	Nov. 30th	Tentative end of coding - Should have most code in place - Start testing and report documentation
D	Dec 4th 9:00AM	Coding deadline, Demo in-class - 3 more days to test
F	Dec 6th 9:30AM	Final deadline, - all work due in Share-in

Task breakdown

Team Member	**Task**	**Deadline**	**Dependencies**
Alex	Figure out c++ (preferably Qt) API for breaking down filestream into frames	Nov.26.2017	N/A
Keir	Find TCP/IP-like protocol implementation in C++	Nov.26.2017	N/A
JC	GitLab/Hub repo is set up, and everyone join	Nov.26.2017	N/A
Tim	Redesign GUI from assignment 1	Nov.26.2017	Assignment 1 code on Git repo
Alex	Help implementing sender-side functions:	Nov.30.2017	Assignment 1 code on Git repo and API figured out
Keir	Help implementing Client-side functions	Nov.30.2017	Assignment 1 code on Git repo and API figured out
JC	Connect functions to GUI elements	Dec 4.2017	Sender & receiver functions implemented
Tim	Comment function headers	Dec 6.2017	Coding finished
Alex	Create and fill out Technical Report (doubles as test case doc)	Dec 6.2017	Program compiles

Keir	Help conduct testing for technical report	Dec 6.2017	Program compiles
JC	User Manual	Dec 6.2017	Program is finished and documented
Tim	Screenshots for user manual	Dec 6.2017	Program is finished and documented