CS490 – Networking

Andrew West

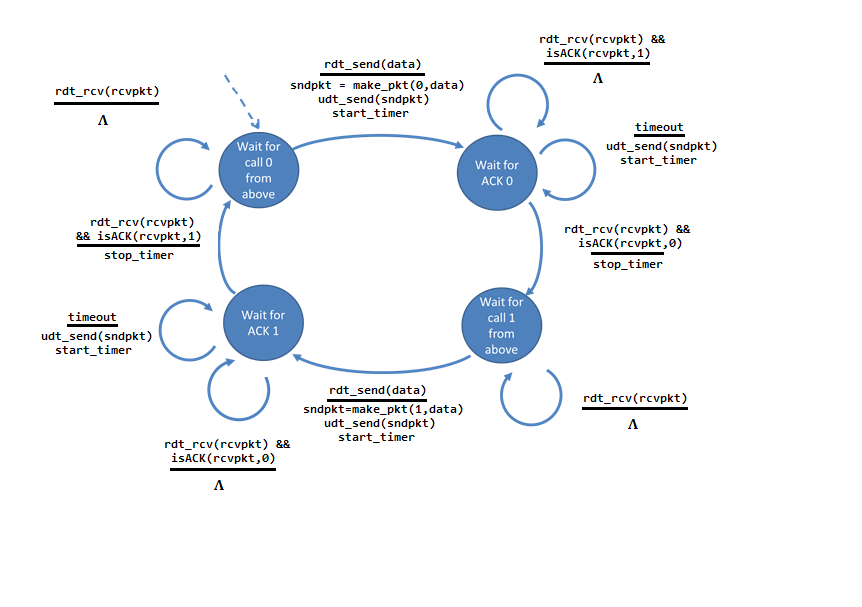
Tomasz Klejmont

**System Documentation**

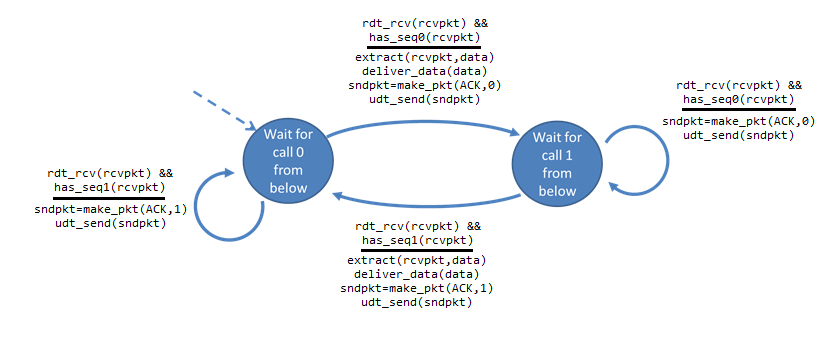
To keep track of files, both on the server and client side, we implemented a file directory class that uses a simple array list to keep track of its contents. On the client side, we’re utilizing the java File class and its methods to load all filenames from a specified directory path on the system to our file directory.

The only important algorithm that was used was the one implementing our RDT sender and receiver class based on the following diagrams:

RDT Sender:



RDT Receiver:



And here’s the logic of the two algorithms used to implement them:

RDT Sender --------------------------------------------------------------------------------------------------------------------------

int state **=** 0**;**

**while(**sending**){**

// Wait for call 0 from above

**if(**state **==** 0**)** **{**

sndpkt **=** make\_pkt**(**0**,** data**)**

udt\_send**(**sndpkt**)**

start\_timer

state **=** 1

**}**

// Wait for ACK0

**if(**state **==** 1**)** **{**

**if(**timeout**)** **{**

udt\_send**(**sendpkt**)**

start\_timer

**}**

// received correct ACK

**if(**rdt\_rcv**(**rcvpkt**)** **&&** isACK**(**rcvpkt**,**0**))** **{**

stop\_timer

state **=** 2

**}**

**}**

// Wait for call 1 from above

**if(**state **==** 2**)** **{**

sndpkt **=** make\_pkt**(**1**,** data**)**

udt\_send**(**sndpkt**)**

start\_timer

state **=** 3

**}**

**if(**state **==** 3**)** **{**

**if(**timeout**)** **{**

udt\_send**(**sendpkt**)**

start\_timer

**}**

// received correct ACK

**if(**rdt\_rcv**(**rcvpkt**)** **&&** isACK**(**rcvpkt**,**1**))** **{**

stop\_timer

state **=** 0

**}**

**}**

**}**

RDT Receiver ---------------------------------------------------------

int state **=** 0**;**

**while(**receiving**){**

// Wait for 0 from below

**if(**state **==** 0**)** **{**

**if(**rdt\_rcv**(**rcvpkt**)** **&&** has\_seq1**(**rcvpkt**))** **{**

sndpkt **=** make\_pkt**(**ACK**,** 1**)**

udt\_send**(**sndpkt**)**

**}**

**if(**rdt\_rcv**(**rcvpkt**)** **&&** has\_seq0**(**rcvpkt**)){**

extract**(**rcvpkt**,**data**)**

deliver\_data**(**data**)**

sendpkt **=** make\_pkt**(**ACK**,** 0**)**

udt\_send**(**sendpkt**)**

state **=** 1

**}**

**}**

// Wait for 1 from below

**if(**state **==** 1**)** **{**

**if(**rdt\_rcv**(**rcvpkt**)** **&&** has\_seq0**(**rcvpkt**))** **{**

sndpkt **=** make\_pkt**(**ACK**,** 0**)**

udt\_send**(**sndpkt**)**

**}**

**if(**rdt\_rcv**(**rcvpkt**)** **&&** has\_seq1**(**rcvpkt**)){**

extract**(**rcvpkt**,**data**)**

deliver\_data**(**data**)**

sendpkt **=** make\_pkt**(**ACK**,** 1**)**

udt\_send**(**sendpkt**)**

state **=** 0

**}**

**}**

**}**

We continuously evaluate the state for any changes and proceed accordingly.