# TDMA Implementation using GNU Radio and USRP

**Avinash Vishnampet Chandrasekaran**

**Priyaraj Banerjee**

**EEC 687 Mobile computing**
**Final Presentation**

Cleveland State University

April 6th, 2009

## Outline

- TDMA
- Project Goal
- Hardware setup
- Software setup
- TDMA Algorithm
- Python script
- Future work
- References
- Questions

Cleveland State University

April 6th, 2009

# TDMA

- **Time division multiple access** is a channel access method for shared medium networks.
- The transmitter transmits in rapid succession, one after the other, each using its own time slot.
- Thus in TDMA protocol a single frequency can support multiple simultaneous data channels.



# TDMA frame Structure

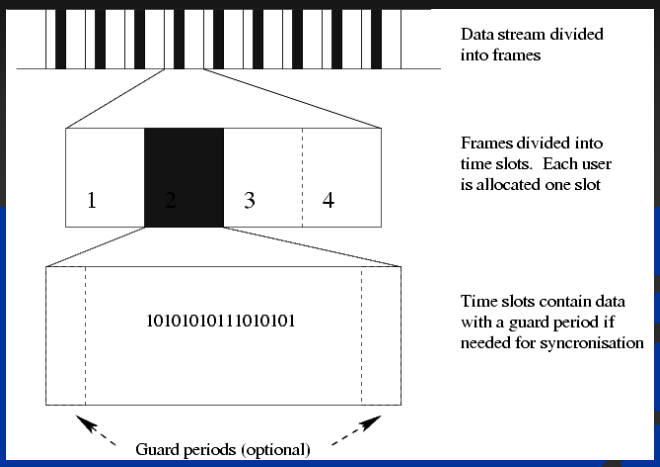


Diagram Source - www.sonoma.edu/users/b/bloomal/graphics/tdma.png

## Project Goal

- To implement a simple Time Division Multiple Access (TDMA) protocol in GNU Radio using USRP.
- To perform a simple file transfer in a 2-node network consisting of two USRP boards connected to two PCs.



## Hardware setup

- We are using two PC's (PC1 and PC2) connected to two USRP's (USRP1 and USRP2) respectively.
- USRP1 consist of two Tx daughter boards acting as node A and node B.
- USRP2 consist of a single Rx daughter board acting as the receiving node.

## Software setup

- We are using the files Benchmark_Tx.py and Benchmark_Rx.py for transmission and reception.
- Benchmark_Tx.py module sends data packets to the connected daughter board without acknowledgement of reception.
- Benchmark_Rx.py module is always on listening mode. Whenever some data is received through USRP it is checked for errors. Firstly, the packet header and PN code are checked and then CRC of the payload is verified.
- rx_callback is called on receiving a packet and if the CRC check is successfully completed, it displays an output message "true", otherwise "false" is displayed with the packet number.



## TDMA Algorithm

- Node 1 starts sending at time t=0sec.
- At time t=10sec the node 1 stops sending and at time t=12sec the Node 2 starts sending .
- The delay of 2 sec is used for synchronization.
- Then at time t= 22 secs Node 2 stops sending and after a delay of 2 secs Node 1 resumes sending the packets and this process goes on.

## Python script

- We can use the time. time( ) or time. clock( )functions for getting the current time ,start= time. time( ).
- Elapsed time = time. time( ) - start , this will give you the elapsed time which is used for switching between the two transmitters.
- The two nodes are assigned Node A and Node B in the data so that the packets at the receiver end can be differentiated.



## Python script

- We use self.subdev = (self.u.db[0][0], self.u.db[1][0]) for selecting both the daughterboard slots.

```
self.u = usrp.sink_c(nchan=2)          # say we want two channels


# we're using both daughterboard slots, thus subdev is a 2-tuple
self.subdev = (self.u.db[0][0], self.u.db[1][0])
print "Using TX d'board %s" % (self.subdev[0].side_and_name(),)
print "Using TX d'board %s" % (self.subdev[1].side_and_name(),)

# set up the Tx mux so that

self.subdev[0].set_gain(self.subdev[0].gain_range()[1])    # set max Tx gain
self.subdev[1].set_gain(self.subdev[1].gain_range()[1])    # set max Tx gain

self.set_freq(0, freq0)
self.set_freq(1, freq1)
self.subdev[0].set_enable(True)             # enable transmitter
self.subdev[1].set_enable(True)             # enable transmitter
```

## Future work

- Implementing the TDMA protocol with more number of nodes.
- Improving the protocol so that when a node has nothing to send the other node can use its time slot.
- Improving the synchronization using timers.



## References

[1] www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html

[2] Rahul Dhar, Gesly George, Amit Malani - Supporting Integrated MAC and PHY Software Development for the USRP SDR.

[3] Naveen manicka -GNU Radio Test Bed , Masters Thesis 2006.

[4] GNU Radio Architectural Changes – BBN technologies Corp

# Thank You

# Questions

Cleveland State University