CHECKPOINT-1

- **Team Number and Names:**

Team Number: #21

Team Members:
- Agneev Ghosh
- Jayant Malani
- Heli Utpal Modi

- **What you've done so far?**

1. To begin with, we have thoroughly studied the MPTCP paper given in the syllabus to get the proper understanding of the concept and implementation.
2. Next, we downloaded the MPTCP source code for Linux Kernel, compiled it and installed it on one of our laptops.
3. Furthermore, we arranged for two Beaglebone black hardware, from a Robotics lab, which have support for both Ethernet and Wi-Fi. These are small Single Board Computer based on ARM architecture. We tried to compile the MPTCP on the Beagle board, it took like ~8 hours, but the board didn't boot with MPTCP enabled kernel. So, it's a work in progress.
4. Finally, we have also developed a simple File Transfer program between a client and server (using TCP) which transfers multiple file from client to server using socket programming. It also calculates the total throughput of sending the files from client.

- **What we plan to do for the next checkpoint?**

1. *Simulation of FTP using MPTCP:*

   We will simulate a File Transfer model of MPTCP in a single system (local host) as discussed in the MPTCP syllabus paper. We have planned to develop this simulated model in either Java or C++.

   Underlying Assumptions considered by us:

- Sockets will behave as physical adapters i.e. A socket "A" will correspond to an actual Ethernet port "A"
- To simulate different sub-flows, we will use parallel threading i.e. a single thread will correspond to single sub-flow.
- To introduce RTT delays, we will run separate waits in the threads themselves.

2. *Conduct Analysis of versions of MPTCP algorithms*

We will verify various versions of MPTCP congestion control algorithms such as EWTCP, COUPLED & SEMI COUPLED as described in the MPTCP syllabus paper. As we will be implementing MPTCP using multiple sockets and parallel threading, we are planning to tweak distinct protocol properties such as RTT, window size, etc. to analyze its impacts on the overall throughput achieved and time taken by these different algorithms in our File Transfer system.

3. *Hardware*

We will also try to demonstrate actual File Transfer between Client and Server on a hardware level (infrastructure created in the virtual box) using the MPTCP protocol. Both the Client and Server (instances) will reside on separate machine and would use separate Wifi and Ethernet channel for transmission of packets/data.

- **A description of the graph (or graphs) that we plan to generate to demonstrate what our project is attempting to do:**

1. *Throughput v/s no. of sub-flows for MPTCP*
   In this graph, we will be showing how number of sub-flows being used to implement MPTCP can affect the overall throughput i.e. whether there is a threshold value of number of sub-flows exceeding which the overall throughput of MPTCP degrades compared to normal TCP or not.

2. *Total Time v/s no. of sub-flows for MPTCP*
   In this graph, we will be showing how number of sub-flows being used to implement MPTCP can affect the overall time taken to transfer all the data to the server i.e. whether there is a threshold value of number of sub-flows exceeding which the overall time taken for transmission of packets using MPTCP degrades compared to normal TCP or not.

3. *Throughput v/s percentage of congestion in each sub-flow for MPTCP*
   In this graph, we will be showing different amount of congestion in individual sub-flows can affect the overall throughput of transferring all the data to the server i.e. we will simulate congestion scenarios in the sub-flows and check its impact.

4. *Throughput v/s time taken by TCP, MPTCP, EWTCP and COUPLED*
   As we are trying to analyze and validate different versions of MPTCP protocol, for this graph, we will be simulating all the above mentioned protocols and then we will show which protocol performs better with respect to Throughput and transmission time.

5. *Throughput v/s RTT and Window size of each sub-flow*
   In this graph, we will be showing the results of tweaking different properties, such as RTT and Window size, of the packet and sub-flow on the throughput of MPTCP. We will be monitoring the performance of MPTCP protocol under these changes.