Go-Back-N Protocol

Mayank Gupta - 2015MT10603
Ridam Maheshwari - 2015MT10610

Things Achieved

- First implemented simple client and server pair to test the sending and receiving of packets.
- Implemented go-back-n protocol with sending data and acknowledgements in one direction only in the existing client-server.
- Made two separate network entities to support bidirectional transfer. Each entity acts as both server and client.

Things Achieved

- Making the **connection full-duplex** was achieved by **threading**. Every entity contains two threads, one each for server and client.
- **Length of the dataframe has been chosen randomly** between 512 and 2024 bits. Length of ackframe is constant at 256 bits.
- Packets and acks have been dropped with probabilities specified beforehand.
- The **window size can be modified** for experimentation.

Generic Implementation Details

- All code has been written in Python 2 and socket programming has been used to make network entities, sockets, etc.
- The **two kind of frames (Dataframe and Ackframe)** have been defined in two separate classes.
- “threading” module has been used instead of “thread”; the former having higher-level interfaces.
- We have **printed details like no. of packets sent/received** etc. at regular intervals to get a look of the network’s performance in real time.

Implementation Details - Client & Server

- Both the client and server socket has been created of the type UDP.
- Server is bounded to a fixed port (9999 in our case). Server keeps listening and the client keeps sending data to this (IP,PORT) pair.
- Whenever a packet has to sent (data or ack), a **class object is constructed of appropriate** type; pickling is done to send the object as bytes over the network.
- Standard go-back-n algorithm as studied in the class has been implemented.

Implementation Details - Entity

- Making a separate network entity was necessary to send the data **bidirectionally**.
- **Entity** consists of both the server & client. Two servers have been constructed in the two entities and and have been bound to different port (9998 & 9999).
- **Full-duplex(ity)** of the network has been achieved by threading. Two threads have been made having the target as client and server respectively and then started simultaneously.

Implementation Details - Transmission

- A **transmit function** has been written which takes care of the **error/drop probability** of the sent packets and acks.
- **Throughput is being calculated** by counting the number of packets received and their respective size, by the entity per second.

Thank You!