

Project 1: Mobile TCP Proxy  
CS 425 – Networking (Dr. Zhang)  
Eric Evans - Songzhe Zhu  
October 23, 2017

### Objective

Using TCP, telnet, and C, create a server-client proxy network that can reconnect automatically when the client changes IP addresses.

### Packet Format

Files: mymessages.c and mymessages.h

The header is 20-bytes in length and consists of 5 fields, each 4-bytes long (interpreted at integers): type, ackID, sessionID, paylen, and SeqID. The payload is simply the contents of the original message.

- *Type* – defines the type of message: HEARTBEAT, DATA, ACK (macros)
- *AckID* – for retransmission (not fully implemented); this field is also used as a heartbeat ID.
- *SessionID* – contains the unique identifier of the sproxy-cproxy session
- *PayLen* – the length of the payload
- *SeqID* – for retransmission (not fully implemented)

### Protocol Actions

After a successful connection between cproxy and sproxy:

- A random integer is generated as the Session ID by cproxy and is sent to sproxy.
- Heartbeat messages are sent to each other every 1 second.
- Incoming messages from cproxy/sproxy have their header fields and payload extracted by `parse_msg()`.
- Outgoing messages to cproxy/sproxy have their header fields and payload filled by `make_msg()`.
- On timeout, each proxy closes their ports to each other, and attempts to reconnect.
  - Cproxy sets up a socket and repeatedly tries to connect to sproxy.
  - Sproxy listens for and accepts cproxy.

- On telnet termination (logout, exit, ^C, etc.), all ports are closed and the startup sequence is reattempted. This means returning from run() to main(), then calling run() again.

#### Retransmission – Extra Credit

We did not fully complete the retransmission implementation. Messages are being correctly enqueued and dequeued, however dequeued messages are never retransmitted. There may still be data loss due to a timeout.