S3F Sockets and Blocking Behavior

Blocking vs Non-blocking

- Refers to behavior of socket calls such as recv()
- Blocking – waits until data is available to send or receive; stops process/thread execution until unblocked
- Non-blocking – if no data can be sent/received, indicate this to the process/thread and continue execution; never block execution
- S3fnet's original (blocking) socket was adapted from ssfnet's non-blocking socket
  - 'non-blocking' was actually a misnomer!
  - The 'block_till' call suspends tcp/udp app execution until a callback is triggered – exactly simulating blocking behavior!
  - Ssfnet's 'blocking' socket actually simulated process interleaving so callbacks were unnecessary and blocking naturally occurred

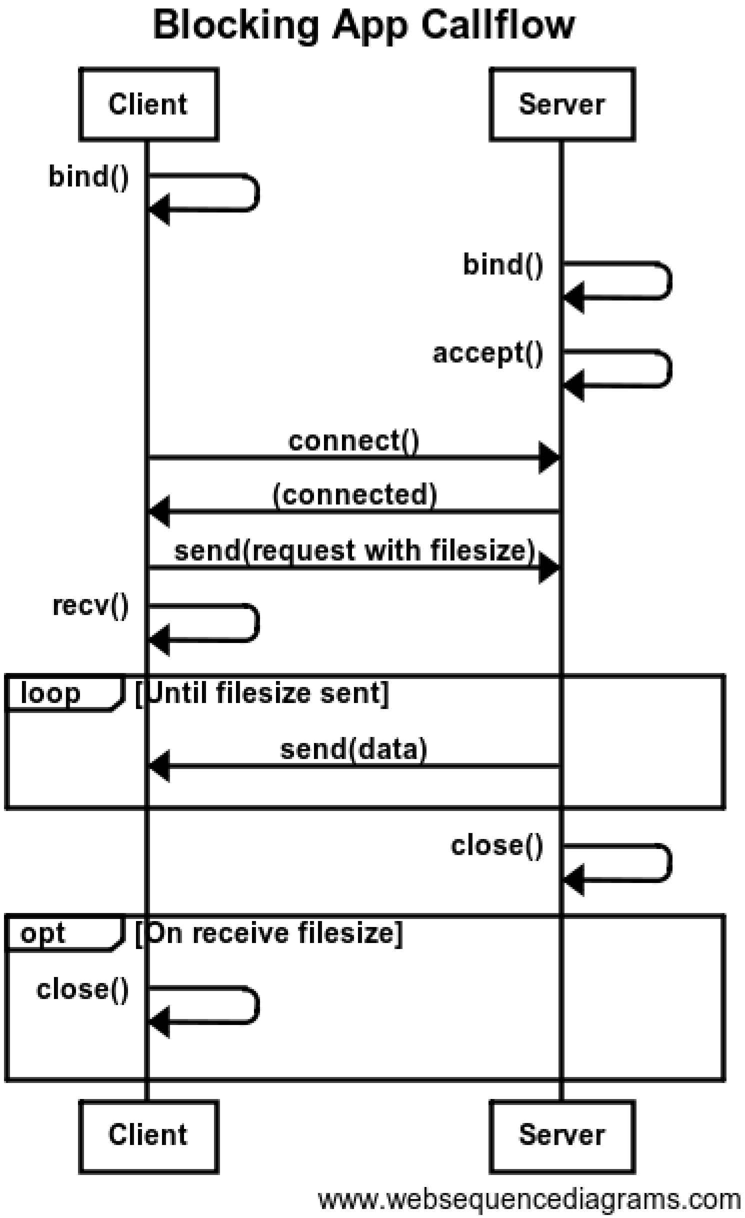
Socket Implemenations

- Implementations are SIMULATION MODELS ONLY!
  - Both use continuations to *model* blocking/non-blocking behaviors
- Blocking
  - Suspends application execution until woken up
  - Does NOT allow context-switching on the process level
    - Callbacks behave atomically for any specific (simulation) timestamp
- Non-blocking
  - Returns EWOULDBLOCK if an operation would suspend execution
  - Continues execution of parent application
  - Socket functions are reentrant
    - Maintains (limited) additional state for resuming

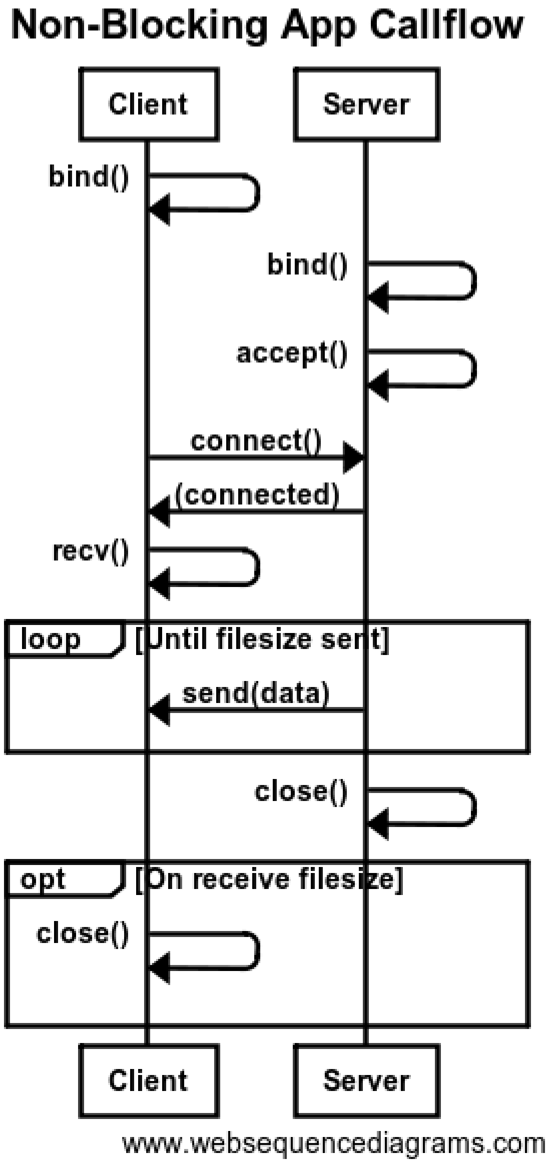
Applications

- Blocking:
  - udp_client.cc
  - udp_server.cc
  - tcp_client.cc
  - tcp_server.cc
  - blocking_tcp_client.cc
    - Is blocking, but follows non-blocking callflow
- Non-blocking:
  - nb_tcp_client.cc
  - nb_tcp_server.cc
    - Hardcoded filesize – client MUST have the same size set in DML to behave correctly; a mismatch may prevent the sockets from closing correctly until timeout
- Detailed (but summarized) callflows for the udp client/server and tcp's accept() are included in socketcallflows.tar.gz

Blocking Callflow



Non-blocking callflow



- NOTE: DOES NOT SEND FILESIZE REQUEST

Included Test Cases

- udp_oneflow
  - Blocking client and server; blocking callflow
- tcp_oneflow
  - Blocking client and server; blocking callflow
- tcp_oneflow_2timeline
  - Blocking client and server; blocking callflow
- nb_tcp_oneflow
  - Blocking client with non-blocking server; non-blocking callflow
- full_nb_tcp_oneflow
  - Non-blocking client and sever; non-blocking callflow