Program B

- Fixed length packets (1000 bytes)
- Variable window size, \( W=W' \) = Maximum
- Send 1000 packets (same as Program A)
  - NEW packet header (See packet.h)

Server/Receiver
- Accept Policy: In-window
- ACK Policy: Your choice

Client/Sender
- RTT Estimation: Van Jacobson/Karel algorithm
  - Option: \( R(n), R'(n) \)
- Retransmit Policy: Your choice
- ETR (Kbps) = \( (8M)/(\text{Elapsed Time}) \)
- Statistics

netsim

- netsim acts like a relay node
  - Configurable: Delay, Drop, Reorder via Xdelay
  - Forward and Reverse channel parameters

netsim Connect Protocol

Program B

- Fixed length packets (1000 bytes)
- Variable window size, \( W=W' \) = Maximum
- Send 1000 packets (same as Program A)
  - NEW packet header (See packet.h)

Server/Receiver
- Accept Policy: In-window
- ACK Policy: Your choice

Client/Sender
- RTT Estimation: Van Jacobson/Karel algorithm
  - Option: \( R(n), R'(n) \)
- Retransmit Policy: Your choice
- ETR (Kbps) = \( (8M)/(\text{Elapsed Time}) \)
- Statistics

netsim

- netsim acts like a relay node
  - Configurable: Delay, Drop, Reorder via Xdelay
  - Forward and Reverse channel parameters

netsim Connect Protocol

Program B

- Fixed length packets (1000 bytes)
- Variable window size, \( W=W' \) = Maximum
- Send 1000 packets (same as Program A)
  - NEW packet header (See packet.h)

Server/Receiver
- Accept Policy: In-window
- ACK Policy: Your choice

Client/Sender
- RTT Estimation: Van Jacobson/Karel algorithm
  - Option: \( R(n), R'(n) \)
- Retransmit Policy: Your choice
- ETR (Kbps) = \( (8M)/(\text{Elapsed Time}) \)
- Statistics

netsim

- netsim acts like a relay node
  - Configurable: Delay, Drop, Reorder via Xdelay
  - Forward and Reverse channel parameters

netsim Connect Protocol
netsim Connect Protocol and Usage

netsim
» A concurrent server (forks child for each connection)
» Acts like a relay node (client and server)
» Running your own netsim
» "-p Port" (XPORT) is normally 2002
» "-t Sec" triggers SIGALRM after Sec seconds
» "-d" turns on debug output

```
 netsim [-p Port] [-t Sec] [-d] >& Log &
netsim –p 2002 –t 300 # Defaults
```

netsim Reply Messages
» BANNER: ASCII string
» CONFIG: ASCII string "220 5 active, netsim v2.0"

netsim as a Packet Relay Node

```
Client
 write() HEADER
 write() DATA
 xread() ACK HEADER
```
```
 Child
 xread() HEADER
 xread() DATA
```
```
Server
 write() HEADER
 xread() DATA
 write() ACK HEADER
```

The netsim Packet Relay Protocol

```
Communication is actually through a TCP connection
netsim expects the stream of header-body pairs
Use xfgets() to read newline-terminated strings
Packet Header
» See packet.h
```
The netsim Model

**Sender-Receiver Direction**

- **Packet**  \(1-p_1\)  **Drop**  \(p_1\)  **Full Queue?**  Yes  **Drop**  No  **Delay \(d+X\)**

- **\(d\)** = Minimum delay
- **\(X\)** = Extra delay with probability \(p_2\) and 0 otherwise
  - i.e., It is a random variable
  - Can cause packets to get out of order
- **Queue size is 128 packets in each direction**
- **Receiver-sender direction looks the same except uses \(p_3\) and \(p_4\)**

**Strategies**

- **Mini-experiment**
  - Telnet to netsim to see if you can connect
- **Start with:**
  - No packet loss and no additional delay
  - Small number of packets
- **Read the FAQ**
- **Work from a firm code base**
- **Do things incrementally**
- **Measure**
- **For debugging, start by outputing just state evolution**
  - Add additional output as needed

**Debugging Strategies (1)**

- **Hand computation**
  - Compute \(R'(n)\) for error-free stop-and-wait with 4 pkts
- **Determine performance parameters for some cases**
  - No drops
  - Fixed window sizes: 1, 8, 64
  - RTT distribution for \(W=1\)
  - Compare with theory: i.e., Effect of pipelining
- **Run test cases using script(s) or Makefile**
  - Activate only primary debug messages
- **Debug messages**
  - Make it easy to understand output (e.g., Use tabs)
  - Label output lines (e.g., server, mod name)

**Debugging Strategies (2)**

- **Use your own netsim**
  - Turn on netsim debugging (-d)
- **Gather supporting information**
  - Number of retransmits
  - Number of timeouts
  - Perhaps by "window rounds"