Netwerken, Spring 2016 Bonus Assignment: Go-back-N ARQ implementation

Deadline: May 15, 2016.

The purpose of this assignment is to extend a Stop-and-wait ARQ implementation from sender A to receiver B, to a Go-back-N sliding window ARQ mechanism from A to B. Notice that in both cases we have a half-duplex connection, so data transfer is going only in a single direction from A to B. The parameters that control the Stop-and-wait protocol are (see header file `variables.h`):

- Packet delay A, in milliseconds (`PACKET_DELAY_TIME_A`).
- Packet delay B, in milliseconds (`PACKET_DELAY_TIME_B`).
- Packet loss ratio, in % (`PACKET_LOSS_PERC`).
- Packet damage ratio, in % (`PACKET_DAMAGED_PERC`).
- Expiration time, in milliseconds (`ARQ_TIMEOUT`).

For the implementation of Go-Back-N ARQ you will have to define an additional parameter `WINDOW_SIZE`, see also the hints below. Note that in the Stop-N-Wait implementation we have chosen to have the number of sent packets to be equal to 1000. This parameter is chosen such that it allows the achieved bandwidth to be "measured" (see `SR_N_PACKETS`). For the Go-Back-N implementation this parameter should not be changed. The other five parameters and the window size have to be varied.

As an outcome of this project we expect you to investigate the trade-offs of the different parameter settings. More specifically, we ask you to produce eight plots in which each time the bandwidth is plotted against the following parameters:

1. Packet delay A vs. packet delay B.
2. Packet loss ratio vs. packet damage ratio.
3. Window size vs. packet loss ratio.
4. Window size vs. packet damage ratio.
5. Window size vs. packet delay A.
6. Window size vs. packet delay B.
7. Expiration time vs. packet delay A.
8. Expiration time vs. packet delay B.

These plots should be 3D surface plots (the two parameters on the x and y-axis and the bandwidth on the z-axis). Use the following values for the parameters in your investigation:

- Packet delay A: 1, 2, 5, 10, 20 ms.
- Packet delay B: 1, 2, 5, 10, 20 ms.
- Packet loss ratio: 0.1, 1, 2, 5, 10 %.
- Packet damage ratio: 0.1, 0.2, 0.5, 0.8, 1.0 %.
- Expiration time: 200, 400, 600, 800, 1000 ms.
- Number of sequence numbers: 8, 64, 128, 256.

Submission

By May 15, we ask you to send a working version of your implementation written in C together with a PDF file containing the eight plots corresponding to the parameter settings and an explanation of what each plot shows. **The assignment must be done individually.** Ensure that you mention your name and student number in your source code and PDF file. Submit your work by e-mail to *krietvel (at) liacs (dot) nl*, make sure the subject **is equal to** "Networking 2016 bonus assignment" and include your name and student ID in the e-mail.

Upon successful completion of the assignment a bonus of up to 1.0 can be "earned" which will be added to the final grade of the final exam. In case of a non-passing grade for the final exam (grade ≤ 5) this addition will not occur. In case of full bonus and a grade of > 9 for the final exam, the minimum of bonus+grade and 10 will be taken.

Hints

- Note that the range of sequence numbers should be a power of two. However the window size should be less than the sequence number range.
- When debugging the code it is recommended that you set the packet delay times A and B to 500 milliseconds or more. Additionally, it is recommended to first get your implementation to work with 0% packet loss and 0% packet damage, representing a perfect connection. You can use the `PERFECT_CONNECTION` define in `variables.h` to enable this.
- When investigating parameter settings 7 and 8, the expiration time should not be smaller than the packet delay time.