UPPSALA
UNIVERSITET

Homework 1: Reliable Communication

Computer Networks I

Oskar Tegby

September 2022

1 Exercise 1 - Go-Back-N

- The killing of Package 1 means that only Package 0 is acknowledged, which causes the window only to move one step until Packages 1-4 are resent and acknowledged, after which the window moves the four remaining steps.

  The reason for this behavior is that the packages have to arrive in order to the receiver in order for them to result in an acknowledgment. This does not happen after Package 0 since Package 1 is killed. Thus, Package 2, 3, and 4 all arrive out of order, and, consequently, do not result in an acknowledgment. This means that *only* Package 0 results in an acknowledgment, so the window only move one step.

  After a timeout has been reached, the unacknowledged packages are resent in order to try again. Then, they all result in acknowledgment since no package is killed on its way. Thus, the window moves the four remaining steps.

- Now, all packages result in acknowledgment since they arrive in order. This causes the window to move all five steps immediately.

2 Exercise 2 - Selective Repeat

- The killing of Package 1 results in all other packages to get acknowledged, but (of course) not Package 1 itself. It remains that way until it it resent and then is acknowledged. The window initially only moves one step, and moves the remaining four steps when Package 1 is acknowledged.

  The reason for this behavior is that only the lost packages result in the loss of an acknowledgment. This only happens for Package 1 here since only it is killed. Thus, Package 0, 2, 3, and 4 all are acknowledged. The window is only moved one step as it moves to the highest acknowledged package that only has acknowledged packages before it, which is Package 1 after the first time they are sent.

  After a timeout has been reached, Package 1 is resent in order to try again. Then, it results in an acknowledgment since it was not killed on its way. The window consequently moves the four remaining steps since this causes the highest acknowledged package (only having acknowledged packages before it) becomes Package 5.

- The same thing happens as in the previous part of this exercise because the acknowledgment for Package 1 again was not received in the first round. This causes the same sequence of events to happen for the same reasons except for the deletion time of the package being after instead before the reception by the receiver.

3 Exercise 3 - Sliding Window

- The base moves to the right when the last acknowledgment only with acknowledgments preceding it get one or more acknowledgments directly succeeding it.

- An empty slot becomes a part of the window when it moves forward in the scenario just described.

- In the figure in the instruction sheet, the window size, N, is ten. Thus, we need to be able to store greater than 8, which means that we need to be able to store numbers up to 16. That is, we need four bits since $\log_2(16) = \log_2(2^4) = 4$.

- The following answers are given.

  - **Go-Back-N**

    - `Good:` The receiver does not need to have a queue since the enumeration of the order of arrival to determine acknowledgment takes care of that.
    - `Bad:` It uses the transmission capacity inefficiently because of sending unnecessarily many acknowledgments (in comparison to e.g. Selective Repeat).

  - **Selective Repeat**

    - `Good:` It avoids unnecessary retransmissions since it only resends a package whenever an acknowledgment is not received before the timeout.
    - `Bad:` Has a high buffer storage since the messages in the receiver buffer continuously have to be reordered.