Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Prelucrarea Imaginilor - Proiect

File transfer - Sliding window protocol

Echipa: 6
Beldiman Vladislav
Grupa 1305A
Hârțan Mihai-Silviu
Grupa 1305B
Timofti Gabriel
Grupa 1305B

November 1, 2020

Contents

1 Introduction

Sliding window protocols are used for reliable in-order delivery of packets is required, such as in the Transmission Control Protocol (TCP). They are also used to improve efficiency when the channel may include high latency. Our application goal is to reliably transfer a file over UDP using the Go-Back-N algorithm.

2 Algorithms

2.1 Sliding window protocol

The Sliding Window Algorithm

According to [1] the sliding window algorithm works as follows:

First, the sender assigns a sequence number, denoted **SeqNum**, to each frame. The sender maintains three variables: The send window size, denoted **SWS**, gives the upperbound on the number of outstanding (unacknowledged) frames that the sender can transmit; **LAR** denotes the sequence number of the last acknowledgment received; and **LFS** denotes the sequence number of the last frame sent. The sender also maintains the following invariant: LFS - LAR $\leq$ SWS.

When an acknowledgment arrives, the sender moves LAR to the right, thereby allowing the sender to transmit another frame. Also, the sender associates a timer with each frame it transmits, and it retransmits the frame should the timer expire before an ACK is received.

The receiver maintains the following three variables: The receive window size, denoted **RWS**, gives the upper bound on the number of out-of-order frames that the receiver is willing to accept; **LAF** denotes the sequence number of the largest acceptable frame; and **LFR** denotes the sequence number of the last frame received. The receiver also maintains the following invariant: LAF - LFR $\leq$ RWS.

When a frame with sequence number SeqNum arrives, the receiver takes the following action. If SeqNum $\leq$ LFR or SeqNum > LAF, then the frame is outside the receiver's window and it is discarded. If LFR < SeqNum $\leq$ LAF, then the frame is within the receiver's window and it is accepted. Now the receiver needs to decide whether or not to send an ACK. Let **SeqNumToAck** denote the largest sequence number not yet acknowledged, such that all frames with sequence numbers less than or equal to SeqNumToAck have been received. The receiver acknowledges the receipt of SeqNumToAck, even if higher numbered packets have

been received. This acknowledgment is said to be cumulative. It then sets LFR = SeqNumToAck and adjusts LAF = LFR + RWS.

The same notations as in [1] will be used in further sections. In addition, ***MaxSeqNum*** *will denote the number of available sequence numbers, and* ***NextSeqNum*** *will track the next packet to send.*

If the sender receives a duplicate ACK message, an untreated case in [1], it simply ignores the message.

2.2 Go-Back-N

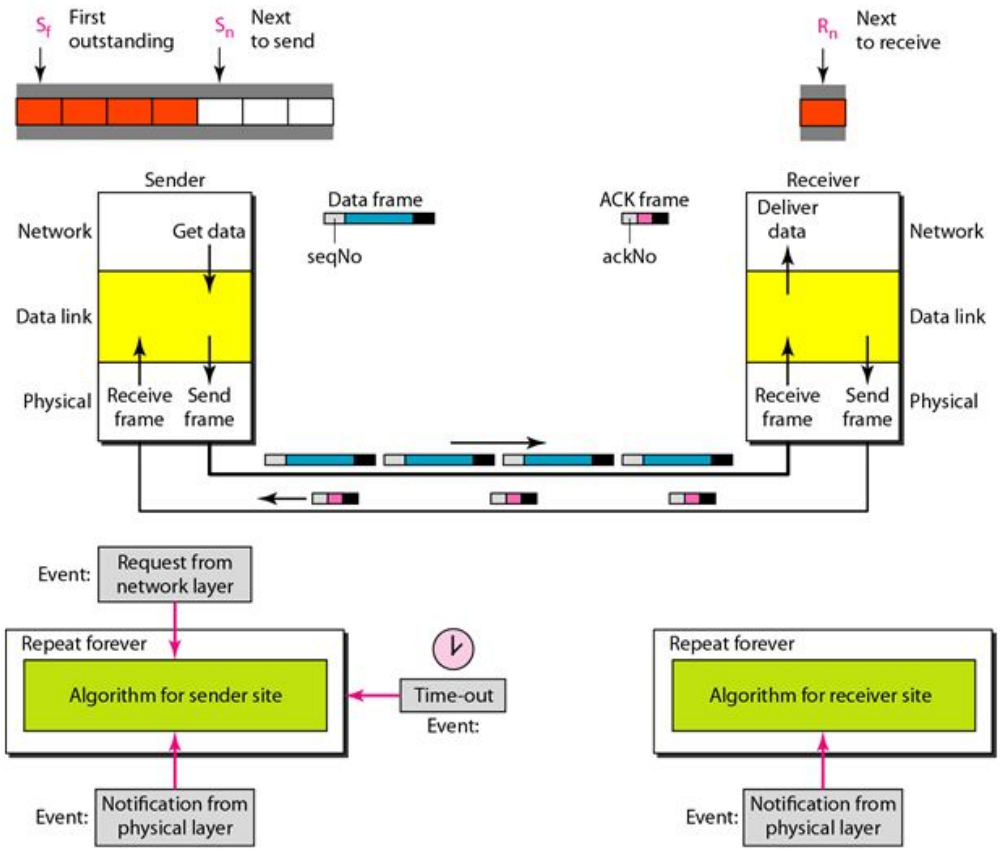
The Go-Back-N implementation of the sliding window protocol uses a SWS > 1, but has a fixed RWS = 1, thus the receiver refuses to accept any other packet but the next one in sequence. As RWS = 1, the sender only needs one timer for the entire window, and, when the timer expires it will resend the entier window. Furthermore, RWS = 1 means that MaxSeqNum $\geq$ SWS + 1 is sufficient. [1]

If a packet is lost in transit or arrives but is corrupted, all following packets are discarded until the missing packet is retransmitted which implies a minimum delay of a round-trip time and a timer timeout. Consequently, it is not efficient on connections that suffer frequent packet loss and/or from noise.

In the case that the receiver is sent a duplicate packet, it sends an ACK message of SeqNumToAck - 1.

2.3 Block diagrams

Figure 1: Block diagram of Go-Back-N algorithm. [Source]



2.4 Components

2.4.1 Sender

TODO

2.4.2 Receiver

TODO

2.4.3 Timer

TODO

2.4.4 Logger

TODO

2.4.5 Window

TODO

2.4.6 Sender Packet Handler

TODO

2.4.7 Reciever Packet Handler

TODO

2.4.8 Sender Acknowledgement Handler

TODO

2.4.9 Reciever Acknowledgement Handler

TODO

3 Git repository

https://github.com/gabitim/RC_Proiect

4 Reference

[1] Peterson L. L. and Davie B. S. Computer Networks a systems approach. (Fifth edition)