Computer Networks lecture notes - Selective Repeat

Aniruddha Jafa

September 25, 2018

1 Introduction

Before the Selective Repeat algorithm can be understood, it is necessary to have some basic knowledge about the GBN (Go-Back-N) Sliding Window Algorithm, as well as the Stop-and-Wait algorithm. Please refer to earlier lecture notes for this. In particular, it is worth re-examining the drawbacks of GBN.

:: Drawbacks of GBN

While GBN allows the pipeline to be filled and is more efficient than the Stop and Wait algorithm, it has some drawbacks:
1. Since the receiver's end has no buffer, out of order frames cannot be stored. While this simplifies the algorithm, it is wasteful.
2. When window-size and bandwidth delay product are large, many frames can be in the pipeline. However, a single timeout when the acknowledgement for a frame not received, will lead to a re-transmission of all frames in the window, which is unnecessary (Kurose and Ross 223).

:: Motivation behind Selective Repeat algorithm

The above drawbacks are what motivate the Selective Repeat algorithm. As the name suggests, unlike GBN, the sender in the Selective repeat algorithm re-transmits only those frames that the sender thinks have been lost or corrupted (Kurose and Ross 224). Unlike GBN where out of order frames are rejected, the Selective Repeat algorithm allows out of order packets to be buffered - the receiver side buffers frames based on the window size at the receiver's end. Hence, the Selective Repeat algorithm maintains 'sliding windows' on both the sender and, and the receiver's end. In the next section, we will look at this more closely.

2 Definitions are terminology

Before we have a closer look at the terms used to describe the working of the algorthim at Sender's and Receiver's end, it is helpful to have a general grasp of the following terms, which are also needed to understand GBN:

acknowledgement: When a receiver receives a frame successfully, it sends back an acknowledgment to the sender. Of course, this acknowledgment may be lost, in which case the receiver has a frame, but the sender does not know it. Such scenarios are handled through a timeout.

timeout: this is time period within which the sender must receive an acknowledgement for a frame in order for it to assume that it has been successfully transmitted to the receiver; if the timeout is exceeded, then the sender takes certain steps (the exact steps are based on the algorithm being used) to ensure that the frame is sent correctly. In Selective Repeat, each frame has its individual timer (Kurose and Ross 226).

window: A window is an interval within which determines whether or not certain operations are allowed (e.g. whether or not the sender is allowed to send a frame). As mentioned before, in the Selective Repeat algorithm both the sender and receiver maintain windows. In practice, a window is maintained by ensuring certain relationships between variables hold. This will be examined more closely in the following sections.

sequence numbers: Sequence numbers are used to index packets. The range of sequence numbers used is closely tied to the Sender's Window Size (SWS), which we will define below. Typically if SWS = N, N+1 sequence numbers 0,1,2...N are used to index frames (see lecture notes on GBN for a justification).

2.1 Terms related to Sender's end

Sender Window Size (SWS): This denotes the maximum number of unacknowledged frames the sender can send.

Last Frame Sent (LFS): This indicates the sequence number of the latest (last) frame that the sender has sent

Last Acknowledgement Received (LAR): This indicates the most recent frame that the sender knows that the receiver has acknowledged, such that all other frames preceding it have also been acknowledged.

:: Invariant for SWS:

The sender maintains a 'window' of frames it is allowed to send by ensuring the following holds: if SWS = N, exclusive-distance(LFS, LAR) $\leq$ N, where exclusive-distance denotes the number of frames between LFS and LAR (excluding LFS, including LAR). One way to think about why the distance excludes LFS is because if a frame has already been acknowledged, we are interested in what frames after this frame we are allowed to send. Hence, we exclude it from our window.

2.2 Terms related to Receiver's end

Receiver Window Size (RWS): This denotes the maximum number of out-of-order frames the receiver can buffer. Clearly, RWS > SWS is not useful since in practice the number of out-of-order frames the receiver needs to buffer (the RWS) can never exceed the number of unacknowledged frames the sender can send (the SWS). We will assume that RWS = SWS = N.

Earliest Expected Frame (EEF): It points to the earliest frame that has not been received, i.e. it has not been received, but everything before it has been received.

Largest Acceptable Frame (LAF): It denotes the largest frame that the receiver is ready to accept i.e. allowed to buffer.

:: Invariant for RWS:

At the receiver's end, the following invariant maintains the window size: inclusive-distance(LAF, EEF) $\leq$ N, where inclusive-distance denotes the number of frames between LAF and EEF (including LAF, and including EEF). The window for RWS is inclusive because EEF represents a frame that we expect but have not yet received, which means that it should be included in the receiver's window.

3 Sender's side of the algorithm

The sender is capable of the following actions (adapted from Kurose and Ross 226):

Send frame: If unsent frames fall within the sender's window, the sender starts sending them in successive intervals. Note that sender can only transmit frames that fall within the sender's window, and have either (i) not been transmitted yet, or (ii) have exceeded their timer and need to be re-transmitted.

Timeout and resend: Each frame has a timer. If the acknowledgement for the frame doesn't appear within a certain interval of time, the sender assumes that the receiver has not received the frame, and re-sends the frame.

Acknowledgement received: If the acknowledgement for a frame has been received, the sender keeps track of it.

Updating window: The sender checks if a new acknowledgement allows LAR to be updated; LAR is shifted to the most recent frame which has been acknowledged, but only if all frames before it have also been acknowledged. This has the effect of shifting the window.

4 Receiver's side of the algorithm

The receiver is capable of the following actions (adapted from Kurose and Ross 226):

Receive frame, send acknowledgment: If the frame falls within the receiver's window, the receiver will buffer it, and send an acknowledgement to the sender. If for some reasons the sender sends the receiver a frame that the receiver has already buffered (this could occur if the acknowledgement of the frame was lost), the receiver will still send an acknowledgement back to the sender. This is needed to ensure that the sender is not re-transmitting the same frame over and over, which will happen until it receives an acknowledgement for that frame.

Reject frame: If the sender sends a frame that is outside the receiver's window, the receiver will reject it.

Update window: The receiver will check if a new frame received allows it to update the window i.e. update EEF. At any stage, the receiver has to make sure that EEF is the frame such that all frames before it have been received, but the frame itself hasn't yet been received. If EEF is updated, this has the effect of shifting the window.

5 Examples of the algorithm, with commentary

The following run-throughs of the algorithm will clarify many of the notions presented above. However, as a simplifying assumption we will consider the sender and receiver separately, i.e. we will have 2 examples to illustrate how the sender and receiver work, but the 2 examples will be completely independent of each other.

5.1 Example of Sender's side

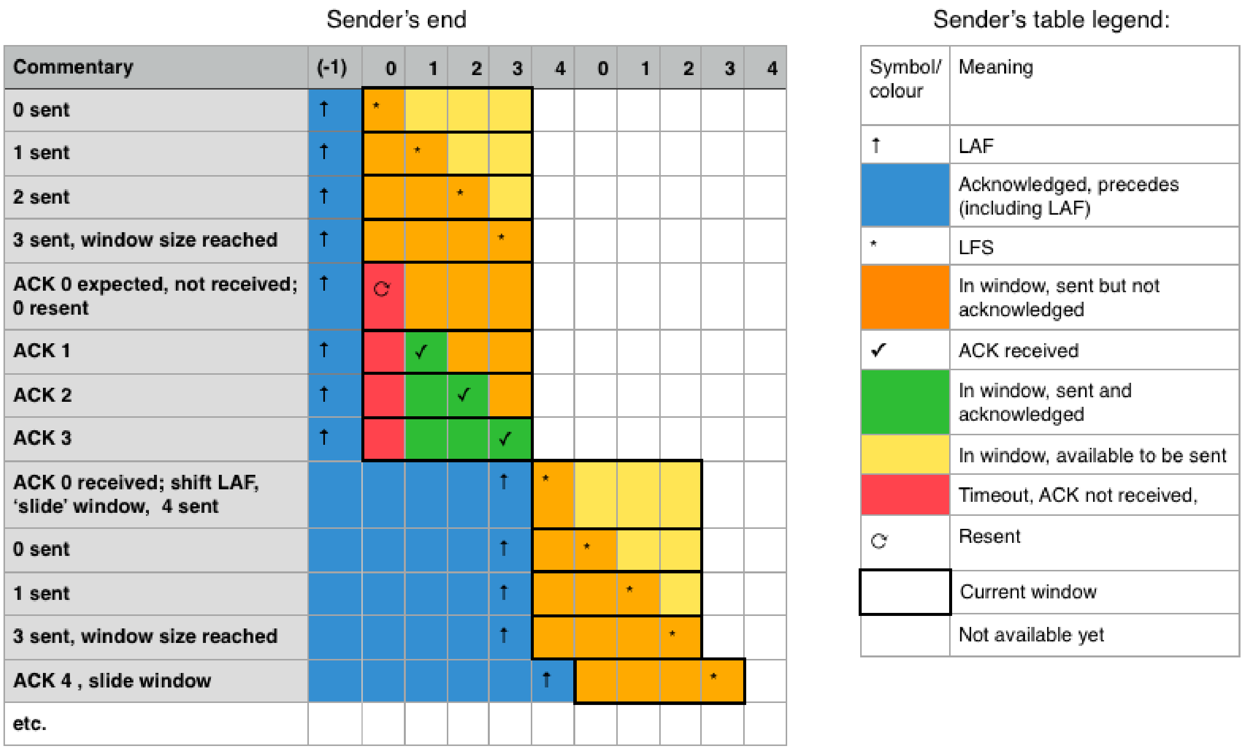

Figure 1: Sender's side of the Selective Repeat Algorithm

5.2 Example of Receiver's side

Receiver's end

Commentary	0	1	2	3	4	0	1	2	3	4				
Initialisation	↑			*										
0 received		↑			*									
1 expected, not received		↑			*									
2 received		↑	✓		*									
3 received		↑		✓	*									
4 received		↑			✓									
Assume the second 0 is received; it will be rejected since it lies outside the window		↑			*									
1 received, update EEF, slide window						↑			*					
etc.														

Receiver's table legend:

Symbol/ colour	Meaning
↑	EEF
*	LAF
✓	Received
	Current window
	Not available for receiving yet
	Received, precedes EEF (excluding EEF)

Figure 2: Receiver's side of the Selective Repeat Algorithm

6 How to choose sequence numbers for frames?

It is important to note here that in Selective Repeat, the sender and receiver may have different pictures of what has been received or not, which can lead to synchronization issues (Kurose and Ross 226). An example is as follows. Assume SWS = RWS = 4, and frames are sequenced 0, 1, 2, 3, 4. Say frames 0, 1, 2, 3, 4 are received successfully and buffered, but all of their acknowledgements get lost. This means that the sender will start re-sending these frames, starting with frame 0. When the receiver gets frame 0, since the window at the receiver's end includes the new 0 (this is the current EEF), it will assume that this old copy of 0 is in fact the new 0. This is clearly problematic.

How to solve this problem? The solution is to increase sequence numbers: if SWS = RWS = N, we need at least 2N distinct sequence numbers.

7 Additional resources

An applet for GBN and Selective Reat can be found here: `http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/`

8 References

Computer Networking, A Top-Down Approach. Kurose and Ross, 6th edition
Computer Networks, A Systems Approach. Peterson and Davie, 3rd edition.