7.1 Worst-Case Latency of Eager Reliable Broadcast

In a worst-case scenario, all process that forward the message after they delivered it, will manage to reach only one process which has not delivered the message yet, before crashing. Furthermore we assume that the first and last process keep running.
Therefore we have:

We assume that $\mid \Pi \mid = n$:
Therefore, if we look at the previous discussed worst-case scenario, $O(n^2)$ messages have been sent, whereas $O(n)$ were delivered.

7.2 Uniform Reliable Broadcast in the Fail-Stop Model

(a) $\mathbb{P}$ does not satisfy its *strong completeness* property

The perfect failure detector does not fulfill the *strong completeness* property if it is not guaranteed that a process that is suspected has indeed crashed.

We assume that process p wants to send a mesagge m to process q which has crashed unnoticed.
In this case process p waits infinitely long for an answer from process q, but this will never happen because q has crashed, and therefore violates the validity property of p and is therefore a liveness issue.

(b) $\mathbb{P}$ does not satisfy its *strong accuracy* property

A failure detector without *stong accuracy* can suspect a process that has not crashed as faulty.

A process p broadcasts a message m to all other processes it is linked to whereas q (one of them) is suspected wrongfully has having crashed. Furthermore we assume that no other message is arriving at q.
All other correct and not suspected processes will eventually *urb*-deliver the message m after receiving the acknowledgement from each other that they *beb*-delivered m whereas q will not. If all these processes then crash, there are processes which *urb*-delivered m, yet the correct process q will never *urb*-deliver m, thereby violating the uniform agreement property which is a safety violation.

7.3 FIFO Broadcast from FIFO Links

Yes, this implement a FIFO-Order reliable broadcast because the usage of FIFO perfect links instead of regular perfect links ensures to have the properties RB1 through RG4.
We can now consider the following broadcasting procedure:
Process p broadcasts the message m_1 and subsequently the message m_2. If process q is delivering m_2 the usage of any FIFO perfect link with the FIFO property ensures that m_1 was delivered earlier.
With this knowledge, and assuming that the processes are single-threaded and event handlers are atomic, we can conclude property FRB5, because in the so modified algorithm it is guaranteed that m_1 is delivered before m_2.