Exercise 2, FoDS

Yannick Martin
Anastasiya Merkushova

Task 1.) Algorithm: Initialization Barrier

Implements:
Initialization Barrier, instance ib.

Uses:
BestEffortBroadcast, instance beb.

upon event <ib, Init> do
processes := Π;

upon event <p, Init> do
trigger <beb, Broadcast>;

upon event <beb, Deliver | p, m>
processes := processes \ {p};
if processes = ∅ then
trigger <ib, Ready>;

Task 2: Eager

(I.)

Sender correct — all Π correct

Sender correct — not all Π correct

Sender is not correct — all Π correct

Sender is not correct (but had time to send) — not all Π correct

(II.) RB4.) Agreement: If a message m is delivered by some correct process, then m is eventually delivered by every correct process.

If a message m is rb-delivered by some correct process, we know that m was added to delivered-set.
Since message m was rb-delivered, we know from BEB3 that m was beb-broadcasted by sender s.
Therefore m was rb-broadcasted before.

1 case: If m was beb-broadcasted by correct process then we know from BEB1 that every correct process eventually delivers m.

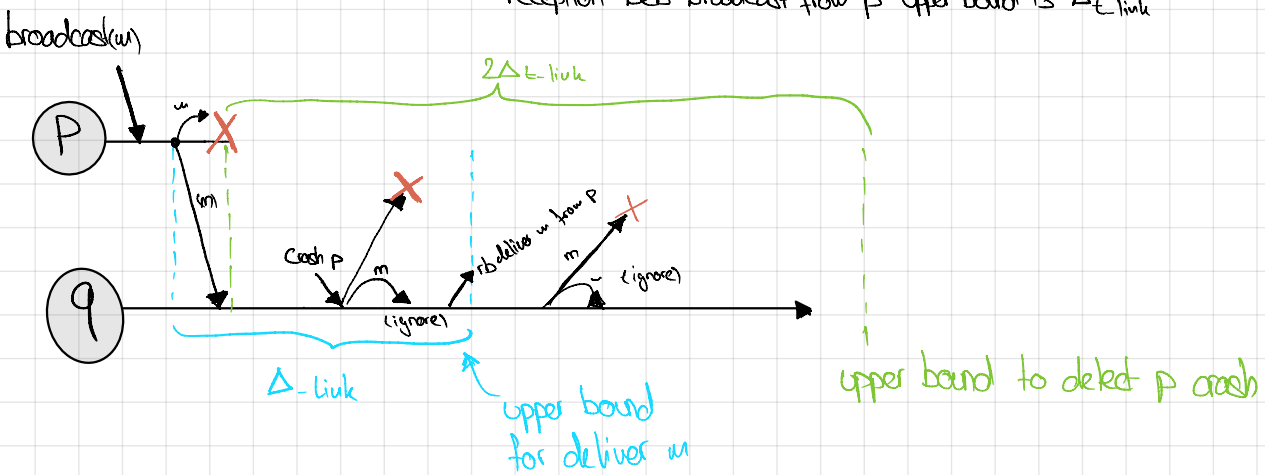
2 case: For the case that beb-broadcast process was faulty, not all processes received m.
But since one message was rb-delivered, we beb-broadcast again the message m with a new sender
If this process was correct then from beb1 follows again that every correct process delivers m. If this new sender is again a faulty sender but at least one will receives it and we are again at the beginning of this case 2 till all Π are reached.

Task 3 :

Synchronous System: There exists an upper bound on the time it takes to deliver and process message m after m has been sent on a link.

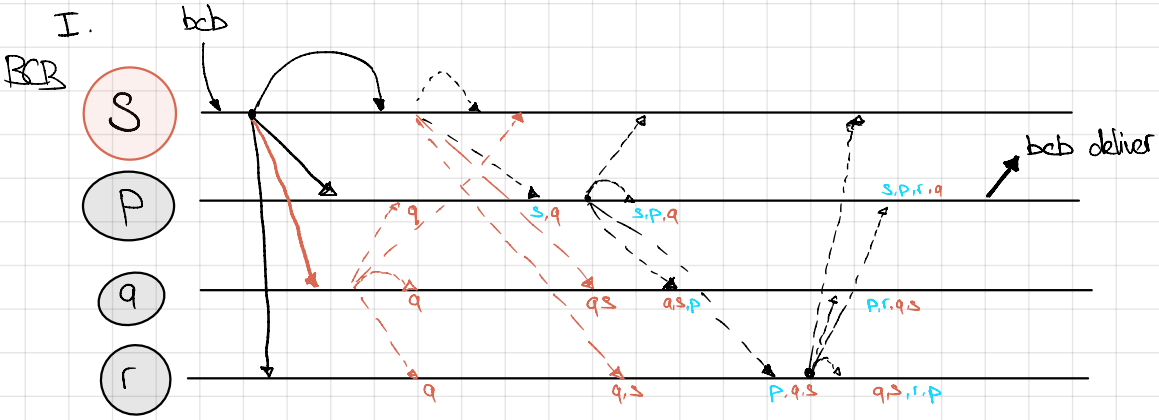
In our case detection by process $p = 2 \cdot \Delta t_{link}$

reception beb-broadcast from p upper bound is Δt_{link}



Task 4)

$N = 4 \quad f = 1$



It is not possible, because there would be an 2nd echo therefore it wouldn't be the same message exchange.