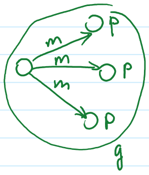
Project Notes

Monday, May 6, 2019 1:40 PM

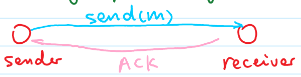
Basic / Reliable Multicast

Basic Multicast — B-delivery.

Guaranteed Delivery, unless multicast process crashes.



for p∈g, performing send (p.m)



problem: ACK received on sender, at almost same time.
if buff overflew, cause of dropping of ACKs
retransmit would create more ACKs

Reliable Multicast — {R- multicast, R- delivery.

Properties: a correct proc p delivers a msg at most once → no duplication / retransmission
if a correct proc p multicast a msg. it would eventually deliver msg. (validity).
if a correct (on receiving) proc p delivers msg m, then every other proc in group g would deliver msg m as well (equality)

For proc p to R-multicast to group g:

```
init:   recv queue: {}
send:   for q in group(g):
            B-multicast (g.m) + (p itself)
recv:   if {m ∉ recv queue → recv queue +={m}  (non-duplication). → msg class + isEqual.
           | then reply p with ACK (Agreement), send m to any other proc in g, if (q≠g).
           | pass msg m to processing
```

Sequencer TO

* for proc in group g:

```
sending:   B-multicast <msg, i> to gU{sequencer(g)}        → id-proc
receiving: msg <m, i> from peer:   hold-back Q += <m, i>
           seq msg <order, i, S> from sequencer, wait until S == rg. → deliver m in holdback Q (TO-deliver)
           rg = S+1
```

* for sequencer of g:

```
init:      Sg = 0
receiving: <m, i> → B-multicast <order, i, Sg>.
           Sg ++.
```

msg + id, class. direction ↓
sequencer2, gs