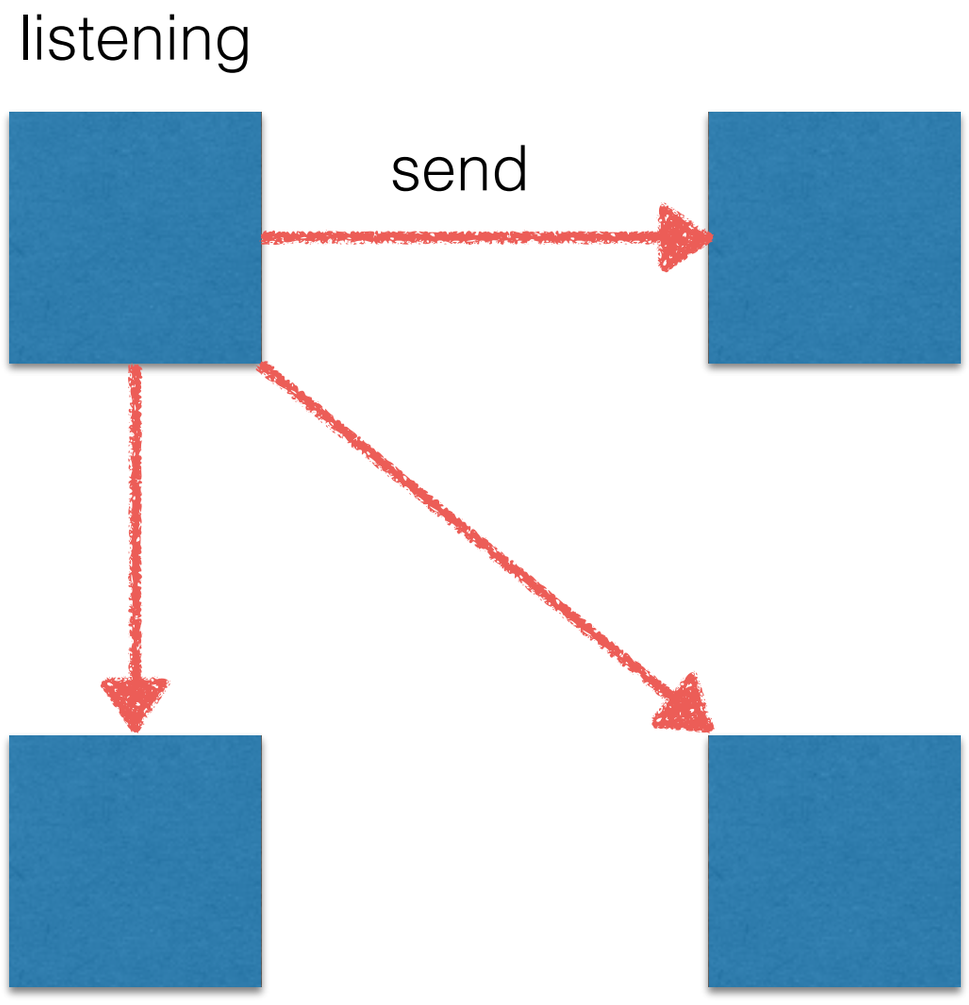
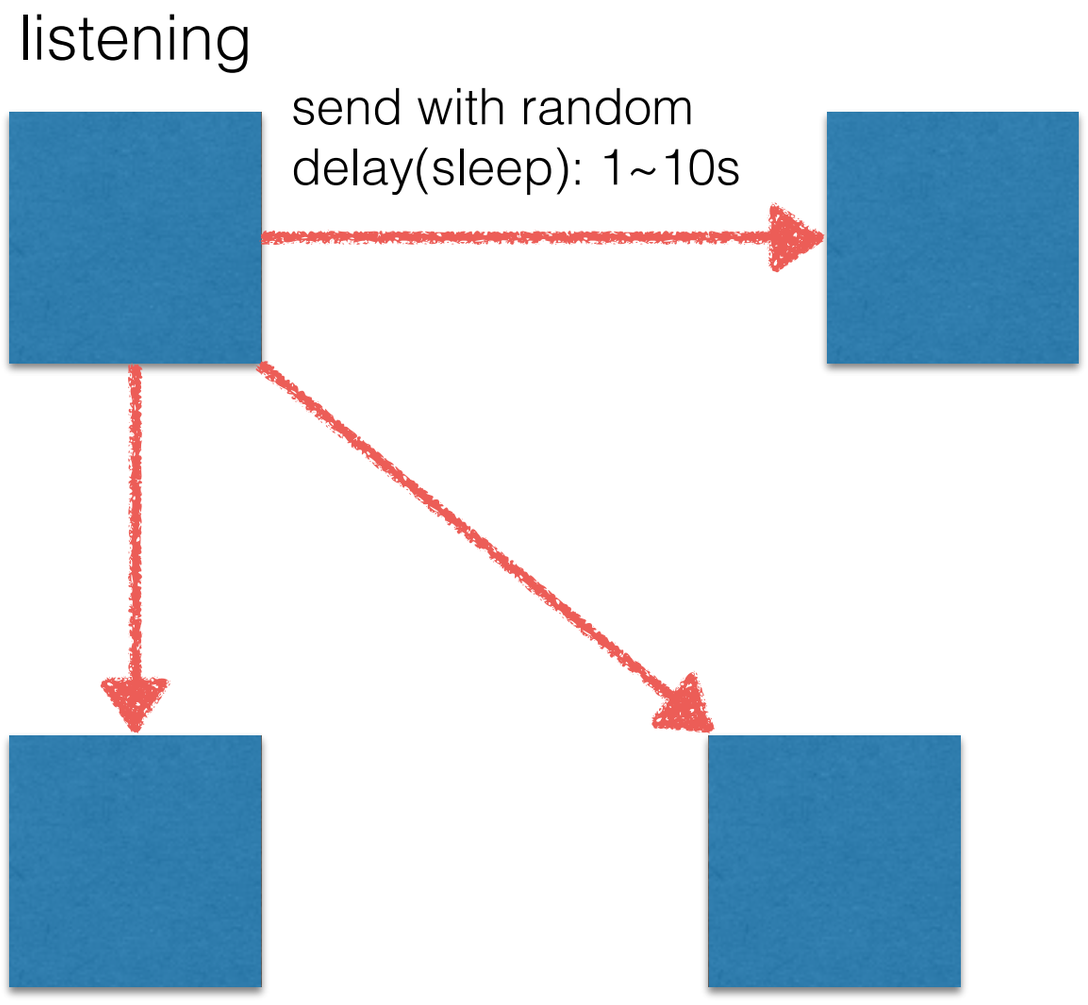
# UDP protocol

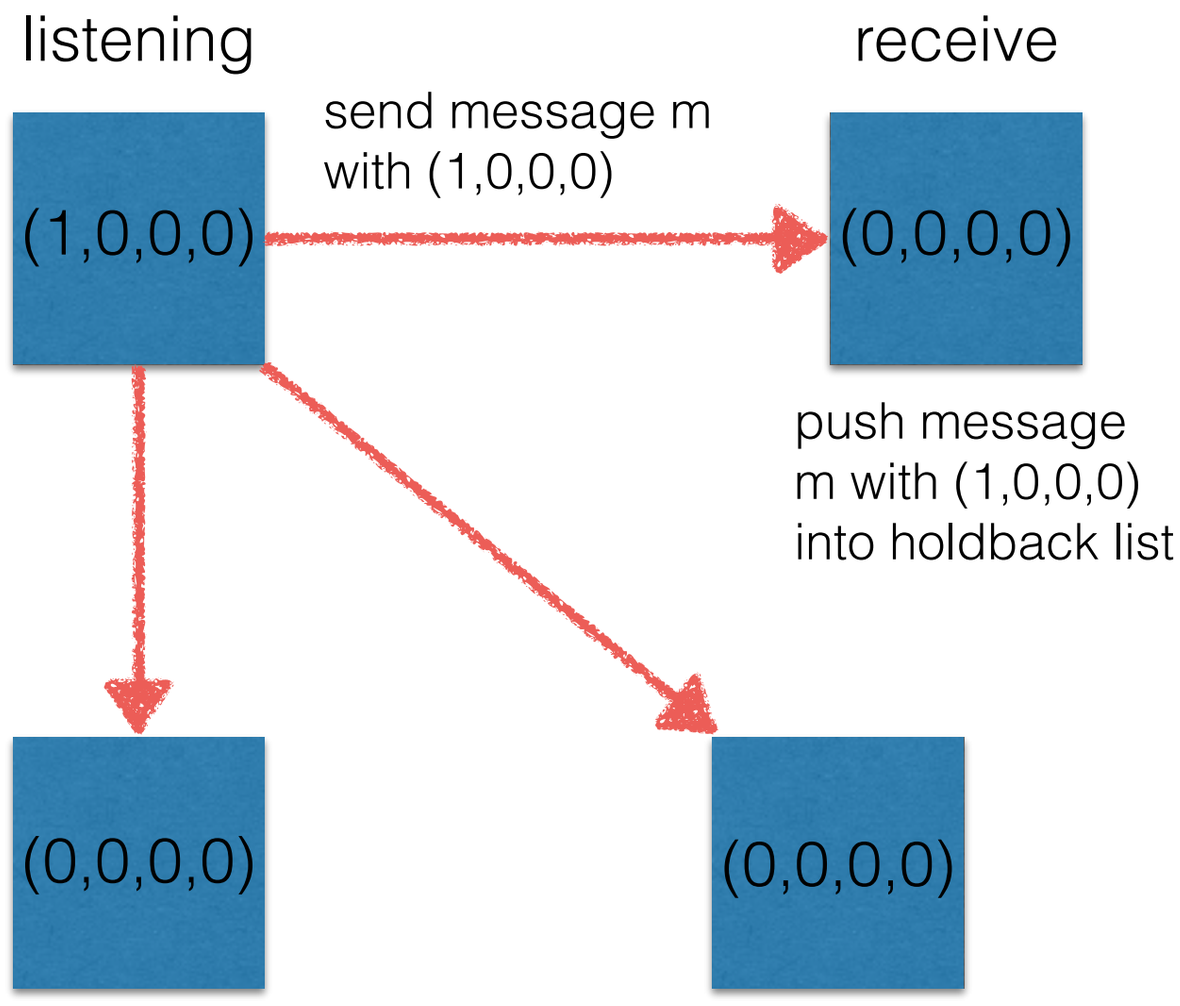


# Multicast with simulated delay



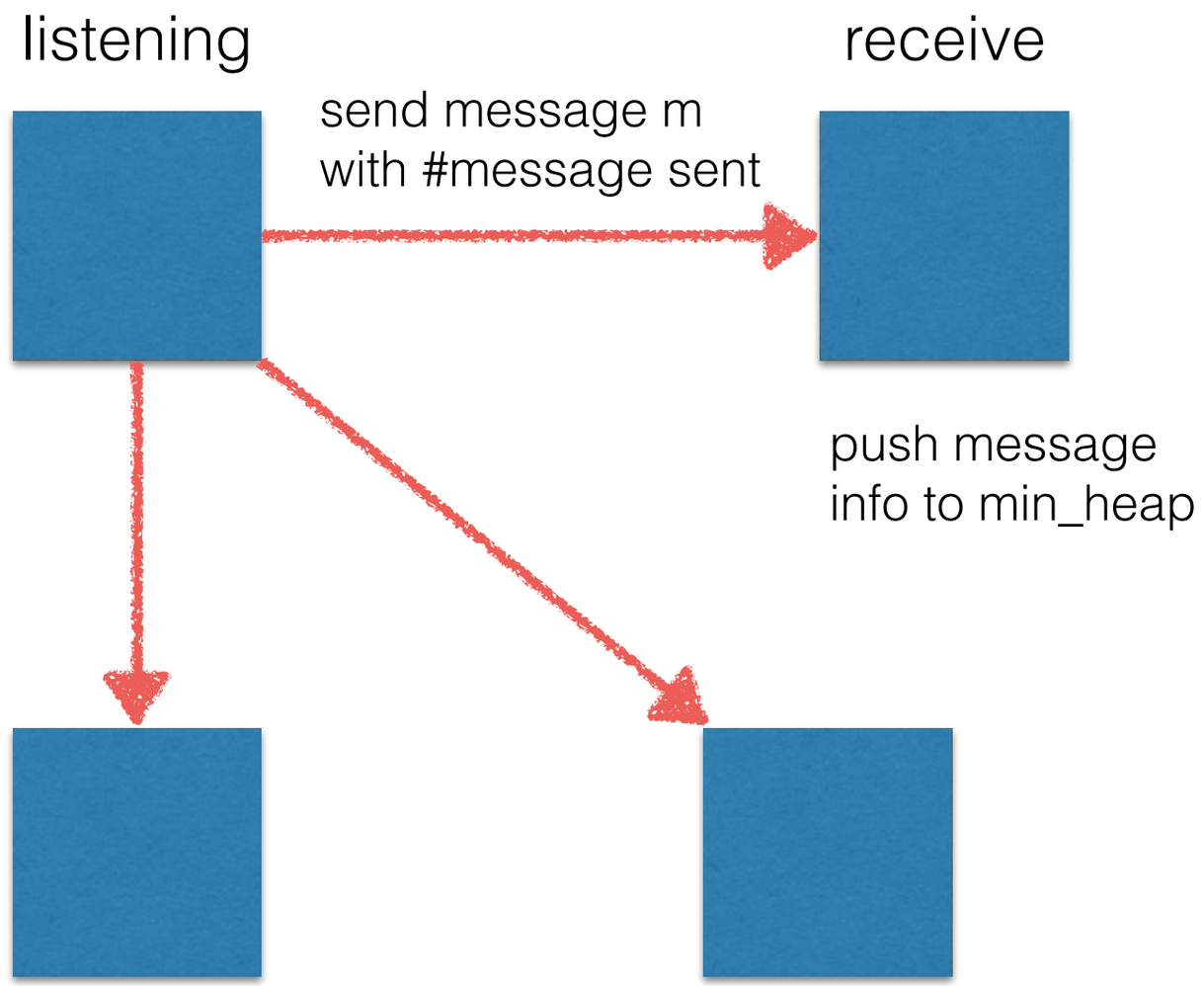
# Causal ordering multicast

- Init
  - vectorstamp (0,0,0,0)
  - #message sent = 0
- send
  - increment #message sent++
  - update vectorstamp
  - send message to each other proc with vectorstamp
  - sender pushes message along with vectorstamp into holdback list with pair(vector, message)
- receive
  - when a receiver receives a message, it pushes the message into its holdback list
- deliver
  - after sending a message, sender i then iterates the entire list, for each entry j of a vectorstamps in i and a vectorstamps in list, if V_j[j] = V_i[j] + 1 and V_j[k] <= V_i[k] (k! =j), then deliver this message, and update V_i[j] += 1. Repeat until no message is allowed to be delivered.
  - receiver i uses the same procedure to deliver a message



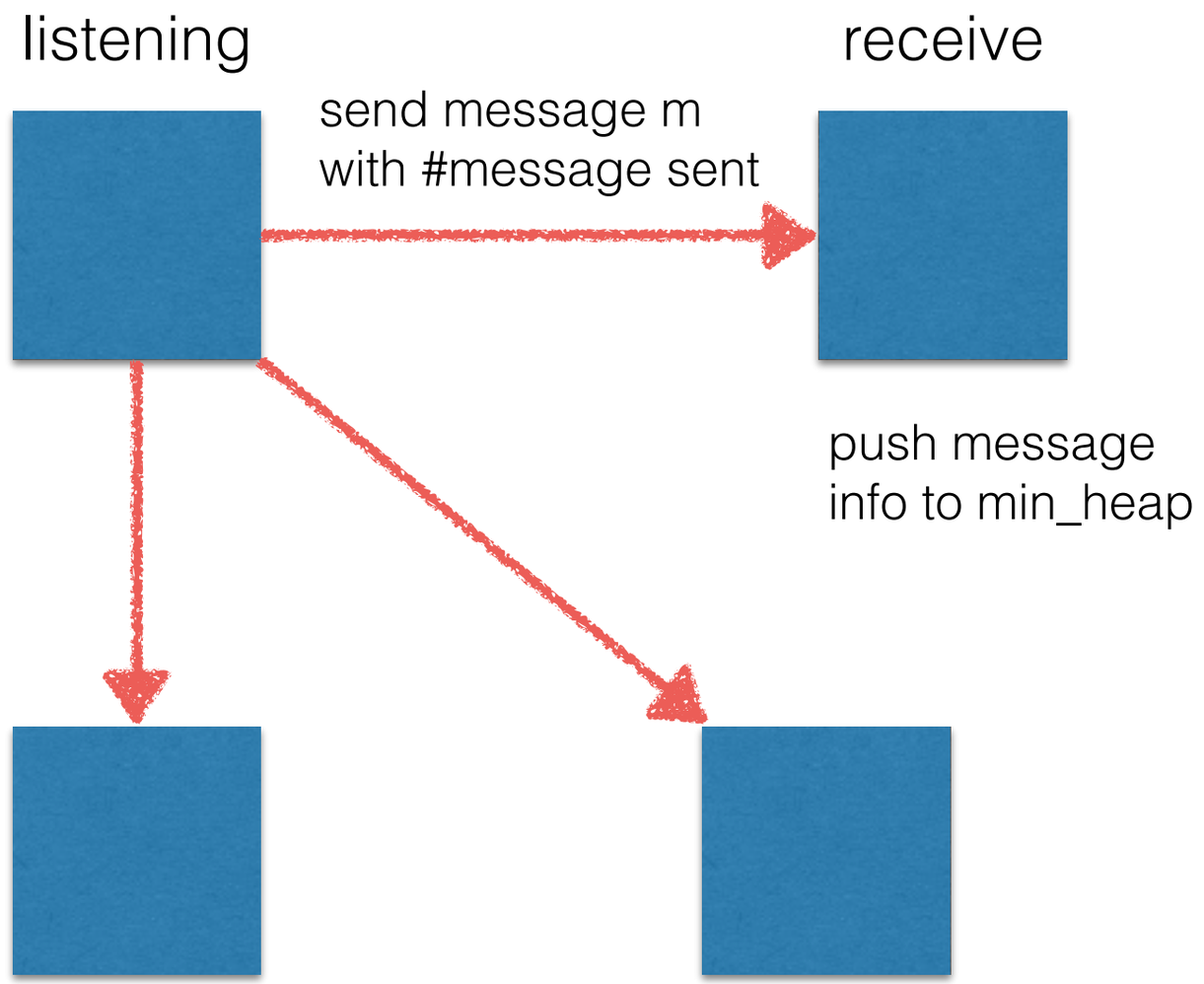
# Total ordering multicast

- Init
  - sequencer(proc1)
    - group order number = 1
    - #message sent = 0
  - processes
    - order number = 1
    - #message sent = 0
- send
  - increment #message sent++
  - sender (not sequencer) pushes message into an array hash maps, where each hash map corresponds to a proc. Each entry in hash map has key=#message sent from sender, and value=message
  - sequencer will deliver sent message in 10s



# Total ordering multicast

- receive
  - when a sequencer receives a message, it will deliver it in 10s
  - when a receiver receives a message, it pushes the message into a 2D array indexed by sender and #message sent
  - when a receiver receivers an order message, pushes the info of this this message into a min_heap, which is ordered by group order number and each entry is a pair(sender, #message sent)
- deliver
  - sequencer sleeps for 10s before delivering this message. Then it sends an order message with its group order number to other processes.
  - a proc in group tries to deliver a message every 10s, by comparing the root of min_heap to its order number, if matched, then try to find the message and deliver it using info in min_heap



# causal ordering

```
cur process is 1
causal ordering
calling listening ...
calling unicast_receive ...
msend m1
Sent "1 0 0 0 1 m1" to process 2, system time is 1456614552
Sent "1 0 0 0 1 m1" to process 3, system time is 1456614552
Sent "1 0 0 0 1 m1" to process 4, system time is 1456614552
Delivered "m1" from process 1, message count 1, system time is 1456614552
msend m2
Sent "2 0 0 0 1 m2" to process 2, system time is 1456614553
Sent "2 0 0 0 1 m2" to process 3, system time is 1456614553
Sent "2 0 0 0 1 m2" to process 4, system time is 1456614553
Delivered "m2" from process 1, message count 2, system time is 1456614553
msend m3
Sent "3 0 0 0 1 m3" to process 2, system time is 1456614554
Sent "3 0 0 0 1 m3" to process 3, system time is 1456614554
Sent "3 0 0 0 1 m3" to process 4, system time is 1456614554
Delivered "m3" from process 1, message count 3, system time is 1456614554
Received message "m2" from process 2, message count 2, system time is 1456614560
calling unicast_receive ...
Received message "m1" from process 2, message count 1, system time is 1456614564
Delivered "m1" from process 2, message count 1, system time is 1456614564
Delivered "m2" from process 2, message count 2, system time is 1456614564
calling unicast_receive ...
Received message "m3" from process 2, message count 3, system time is 1456614564
Delivered "m3" from process 2, message count 3, system time is 1456614564
calling unicast_receive ...
```

```
cur process is 2
causal ordering
calling listening ...
calling unicast_receive ...
Received message "m2" from process 1, message count 2, system time is 1456614555
calling unicast_receive ...
msenReceived message "m1" from process 1, message count 1, system time is 1456614556
Delivered "m1" from process 1, message count 1, system time is 1456614556
Delivered "m2" from process 1, message count 2, system time is 1456614556
calling unicast_receive ...
d m1
Sent "2 1 0 0 2 m1" to process 1, system time is 1456614557
Sent "2 1 0 0 2 m1" to process 3, system time is 1456614557
Sent "2 1 0 0 2 m1" to process 4, system time is 1456614557
Delivered "m1" from process 2, message count 1, system time is 1456614557
msend m2
Sent "2 2 0 0 2 m2" to process 1, system time is 1456614558
Sent "2 2 0 0 2 m2" to process 3, system time is 1456614558
Sent "2 2 0 0 2 m2" to process 4, system time is 1456614558
Delivered "m2" from process 2, message count 2, system time is 1456614558
msend m3
Sent "2 3 0 0 2 m3" to process 1, system time is 1456614559
Sent "2 3 0 0 2 m3" to process 3, system time is 1456614559
Sent "2 3 0 0 2 m3" to process 4, system time is 1456614559
Delivered "m3" from process 2, message count 3, system time is 1456614559
Received message "m3" from process 1, message count 3, system time is 1456614572
Delivered "m3" from process 1, message count 3, system time is 1456614572
calling unicast_receive ...
```

```
cur process is 3
causal ordering
calling listening ...
calling unicast_receive ...
Received message "m1" from process 2, message count 1, system time is 1456614560
calling unicast_receive ...
Received message "m3" from process 2, message count 3, system time is 1456614562
calling unicast_receive ...
Received message "m1" from process 1, message count 1, system time is 1456614564
Delivered "m1" from process 1, message count 1, system time is 1456614564
calling unicast_receive ...
Received message "m2" from process 2, message count 2, system time is 1456614564
calling unicast_receive ...
Received message "m2" from process 1, message count 2, system time is 1456614565
Delivered "m2" from process 1, message count 2, system time is 1456614565
Delivered "m1" from process 2, message count 1, system time is 1456614565
Delivered "m2" from process 2, message count 2, system time is 1456614565
Delivered "m3" from process 2, message count 3, system time is 1456614565
calling unicast_receive ...
Received message "m3" from process 1, message count 3, system time is 1456614575
Delivered "m3" from process 1, message count 3, system time is 1456614575
calling unicast_receive ...
```