

# Reliable Group Communication - Disposition

Lukas Peter Jørgensen, 201206057, DA4

26. marts 2014

## 1 Intro

Guarantee that messages are delivered to all members in a process group.

**Naive solution** → endless iteration.

## Issues

Simple case: simply have  $n$  point-to-point channels.

- May waste network bandwidth, might be an issue in large networks.
- Limited by number of channels.

## 2 Reliable multicasting

**Intuitively:** A message sent to a group, should be delivered to each member of that group.

**What if:** A process joins during communication? Or a (sending) process crashes during communication.

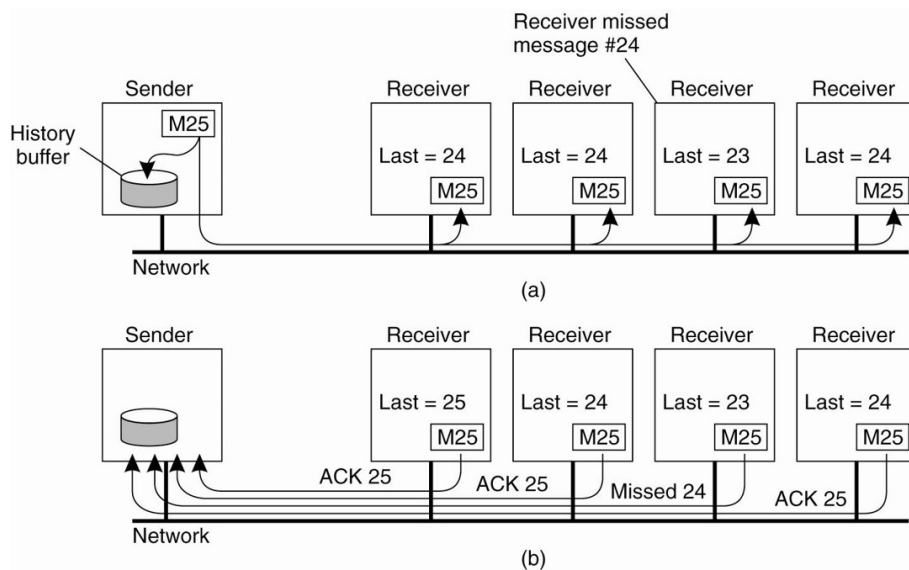
**Distinction** between two cases. Reliable communication in the presence of faulty processes, and reliable communication when processes are assumed to operate correctly.

In the first case, multicasting is considered to be reliable when all nonfaulty group members receive the message.

**First we will look at the second case, when we assume processes operate correctly.**

## 3 Simple solution

**Assumptions:** All receivers known and assumed not to fail.



## Problem

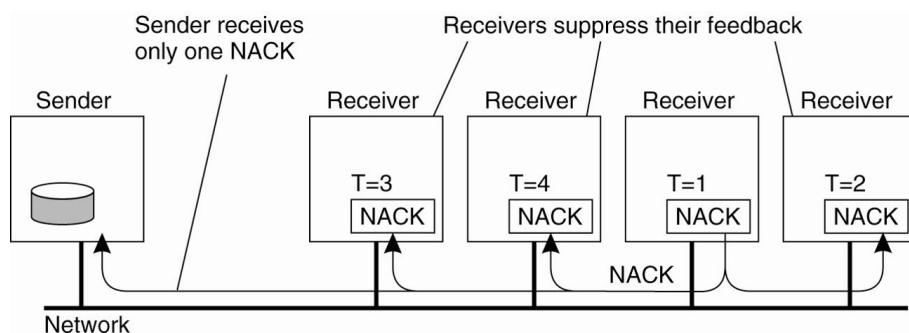
**Cannot support large number of receivers.** If there are  $N$  receivers, the sender must be prepared to accept at least  $N$  acknowledgements. Sender may be swamped by acknowledgements. (Feedback implosion)

## Possible solutions

**Only feedback when something goes wrong**, what if the message is lost before reaching any of the users? Feedback implosion might still happen, sender will be forced to keep messages in its buffer forever because it can never know if everyone has received the message.

## 4 Non-hierarchical

Scales reasonably well, has seen practical uses.



If a process realizes it is missing a package, multicast the retransmission request.

**But**, do it after a short random delay to avoid concurrent retransmission request, if a process needs a retransmission but has received a retransmission request from another receiver, it will simply suppress its own request.

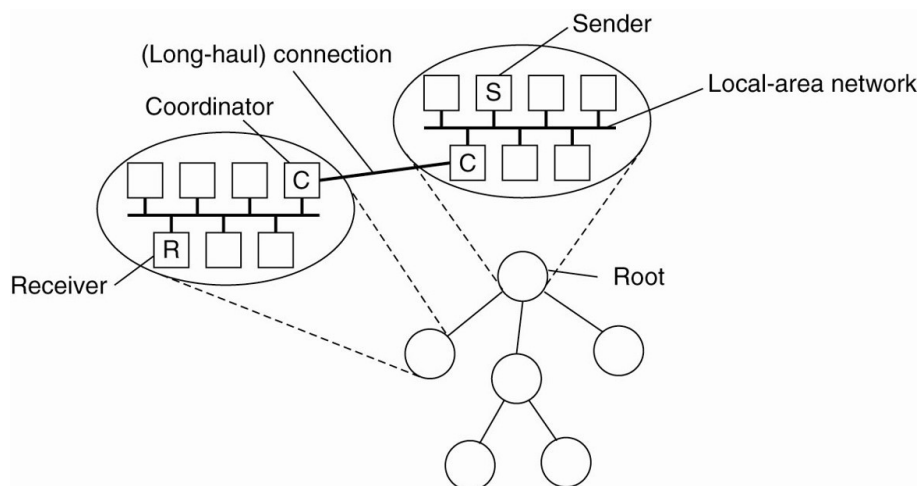
## Problems

**Receivers have to process useless messages.** Whenever a retransmission request or a retransmission occurs, all receivers have to process the request even when it has nothing to do with them.

**Ensuring that each feedback message is only sent once**, is a tricky issue as well?

## 5 Hierarchical

Scalability for large groups of receivers.



Partition receivers into a number of subgroups, which can be organized into a tree.

Each subgroup has a **coordinator** which is responsible for handling retransmission requests of receivers contained in its subgroup.

In a scheme based on acknowledgements, each coordinator can delete a message  $m$  from its buffer when it has received acknowledgements for message  $m$  from all members in its subgroup, as well as from its children.

**Note:** Doesn't have to get acknowledgements from all children before it can send acknowledgement to its parent.

## Problems

**Construction of the tree**, is an issue as many cases need the tree to be constructed dynamically. One approach is to use the multicast tree in the underlying network, if there is one.

**Principally**, the approach is to enhance each multicast router in the network layer such that it can act as a local coordinator. Unfortunately, as a practical matter such adaptations to existing networks are not easy to do, therefore application-level multicasting solutions have gained popularity.