

## **3. Aufgabenblatt**

**Ordnung bei Gruppenkommunikation**

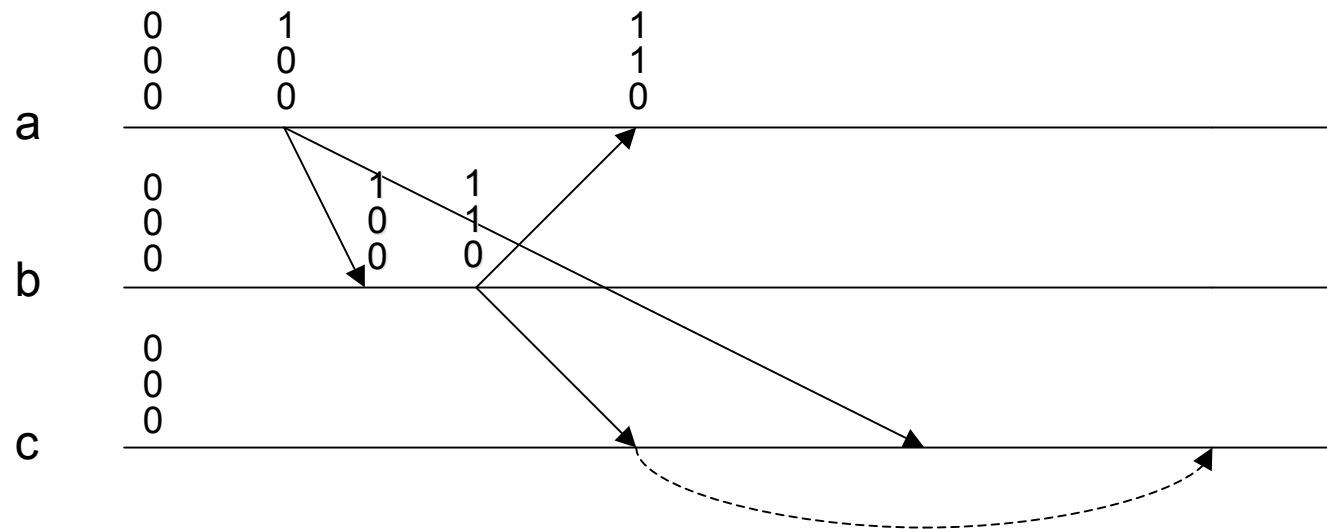
## Aufgabe 1: Ordnung von Rundnachrichten

- **ungeordnet:** keine Anforderung (nicht einmal FIFO)
- **FIFO:** Mehrere Nachrichten *eines Senders* kommen bei allen Empfängern in der Absende-Reihenfolge an.
- **Kausalordnung:** Für alle Nachrichten, die in einer Kausalitätskette gesendet wurden, gilt FIFO.
- **Totale Ordnung:** Alle Nachrichten treffen überall in der gleichen Reihenfolge ein.

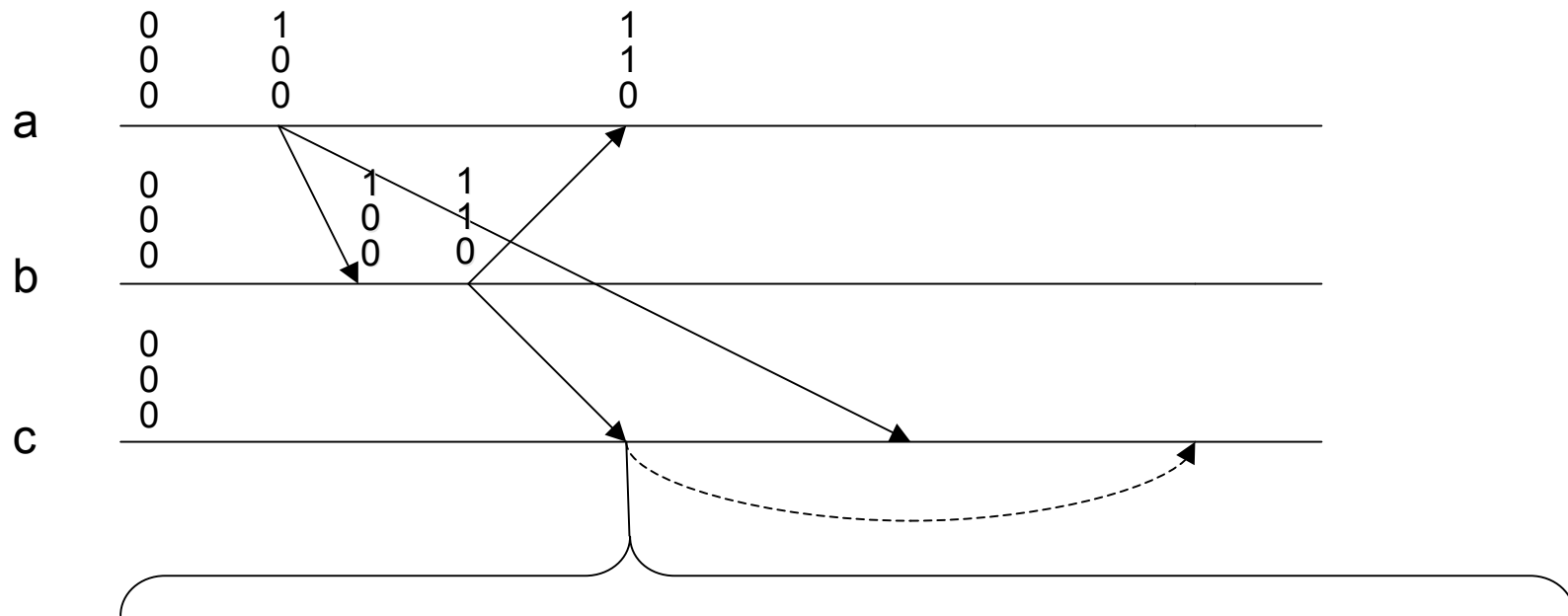
## Aufgabe 2: CBCAST

- bietet: kausal geordnete Rundmeldungen
- Vor.: zuverlässige Nachrichtenübertragung
- Vorgehen:
  - Prozess  $p$  erhöht in seiner Vektorzeit  $P$  vor dem Senden  $P_p$  um  $1$  und heftet dann  $P$  als Zeitstempel  $t$  an die Nachricht an.
  - Eine von  $p$  bei  $q$  eintreffende Nachricht mit Zeitstempel  $t$  wird vom Nachrichtensystem solange zurückgestellt (d.h. nicht abgeliefert), bis gilt:  
$$t_p = Q_p + 1 \quad (\text{garantiert FIFO-Ordnung}) \text{ und}$$
$$\forall i \neq p: t_i \leq Q_i \quad (\text{garantiert Kausalordnung: alle Nachrichten, die } p \text{ gesehen hat, hat auch } q \text{ gesehen})$$
  - Beim Abliefern einer Nachricht von  $p$  bei  $q$  wird  $Q_p$  um  $1$  erhöht.
- Rundsendungen hier nicht reflexiv

# CBCAST



# CBCAST



$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, t = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

1. Bedingung

$$t_p = Q_p + 1$$

$$\Rightarrow t_2 = C_2 + 1$$

✓ erfüllt

2. Bedingung

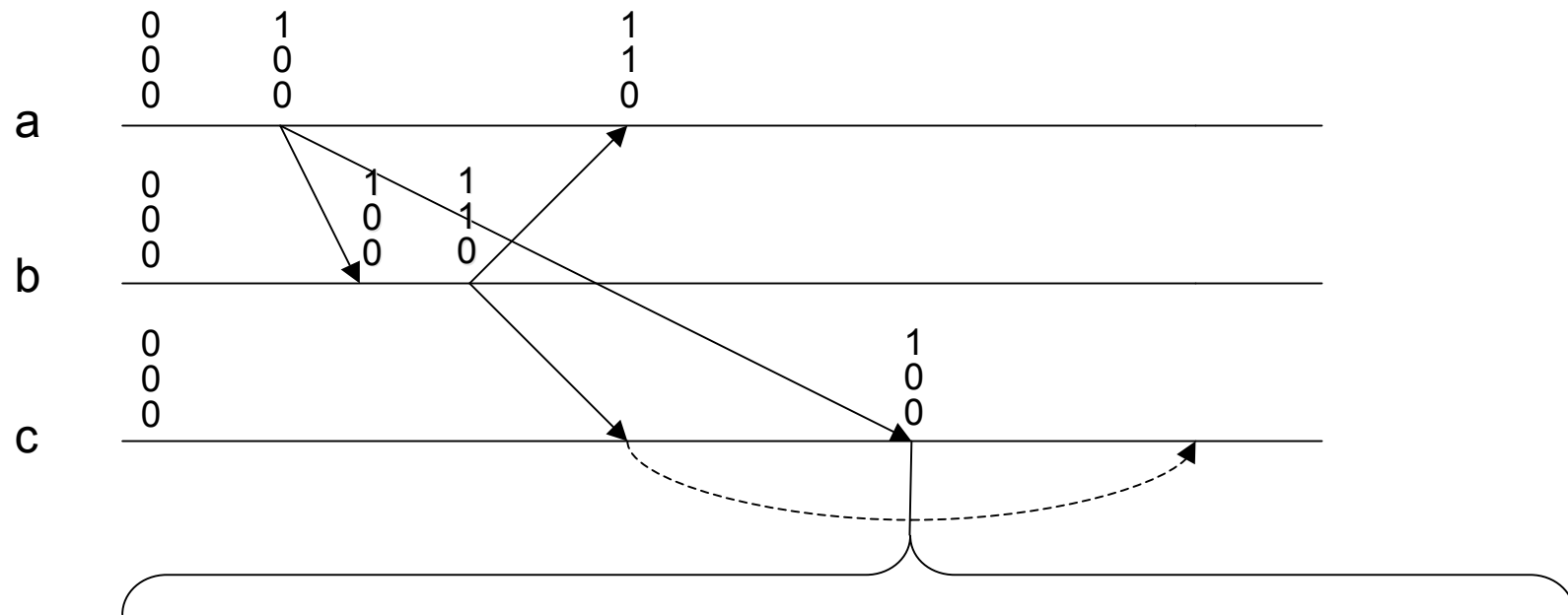
$$\forall i \neq p : t_i \leq Q_i$$

$$\Rightarrow \forall i \neq 2 : t_i \leq C_i$$

aber :  $t_1 > C_1$  ( $1 > 0$ )

⚡ Nachricht „aus der Zukunft“  
 $\Rightarrow$  verzögern

# CBCAST



$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, t = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

1. Bedingung

$$t_p = Q_p + 1$$

$$t_1 = C_1 + 1$$

✓ erfüllt

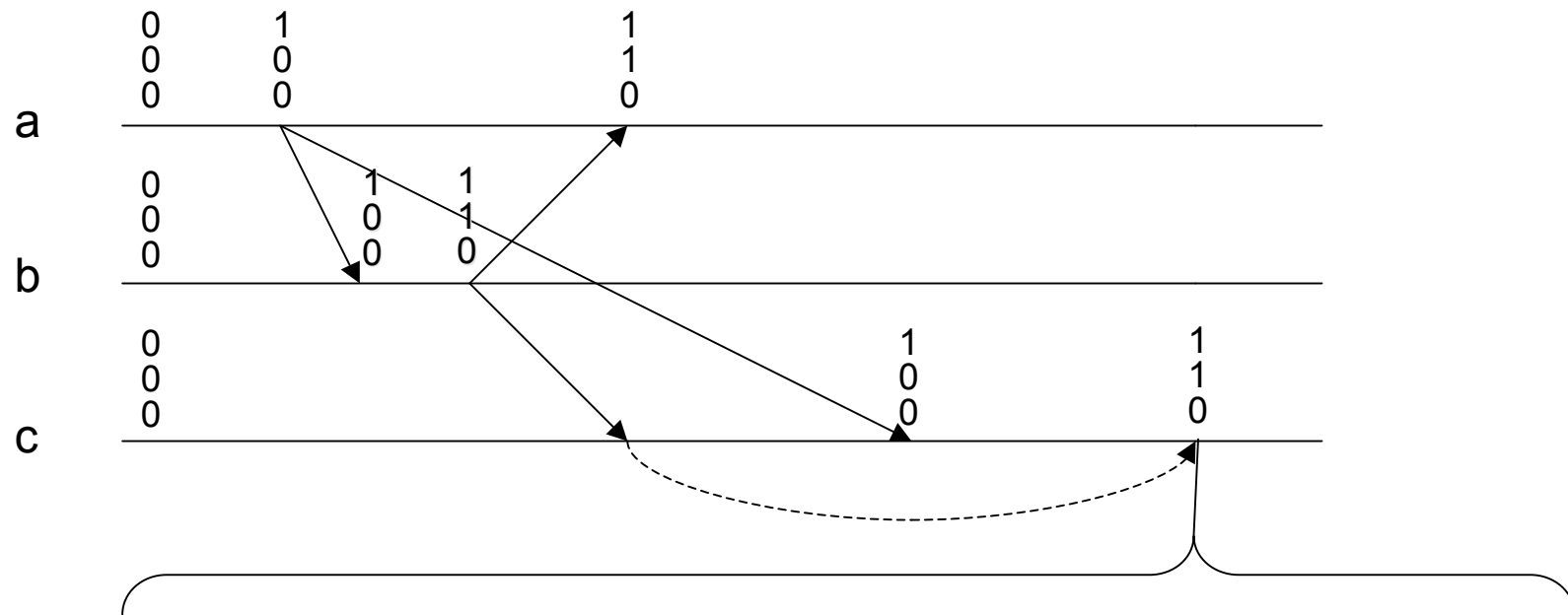
2. Bedingung

$$\forall i \neq p : t_i \leq Q_i$$

$$\forall i \neq 1 : t_i \leq C_i$$

✓ erfüllt  $\Rightarrow$  zustellen,  
Ausstehende prüfen

# CBCAST



$$C = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, t = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

1. Bedingung

$$t_p = Q_p + 1$$

$$t_2 = C_2 + 1$$

✓ erfüllt

2. Bedingung

$$\forall i \neq p : t_i \leq Q_i$$

$$\forall i \neq 2 : t_i \leq Q_i$$

✓ erfüllt  $\Rightarrow$  zustellen

## Sicherstellen einer Ordnung

- theoretische Betrachtung von Ereignissen vs. Sicherstellen in realem System
- kausale Ordnung bei Rundmeldungen
  - Vektorzeit mit der Relation " $<$ " ist isomorph zur kausalen Ordnung
  - Sicherstellen in CBCAST *ähnlich* Vektorzeit (kein lokales Hochzählen, Broad- statt Unicast, keine Maximumbildung)
- kausale Ordnung beim Eintritt in kritischen Abschnitt
  - Skalarzeit (Lamport) mit " $<$ " ist zwar nicht isomorph zur kausalen Ordnung, genügt aber zum Sicherstellen



## Rundsendungen in JPVM

```
public class jpvmEnvironment {  
  
    public void pvm_send(jpvmBuffer buf,  
                        jpvmTaskId tid,  
                        int tag) {...}  
  
    public void pvm_mcast(jpvmBuffer buf,  
                        jpvmTaskId tids[],  
                        int ntids,  
                        int tag) {...}  
  
}
```

## Rundsendungen in JPVM

```
public class jpvmbuffer {  
  
    public void pack(jpvmtaskid d[],  
                    int n,  
                    int stride) {...}  
  
    public void unpack(jpvmtaskid d[],  
                       int n,  
                       int stride) {...}  
  
}
```

- für die Übung aber: "manueller" Multicast

## Beispiel: Pseudocode

```
monitor Station (Station succ)
    boolean electing := false;
    Station coordinator := nil;

proc startElection()
    electing := true;
    succ.election(this);
    await coordinator  $\neq$  nil;

port election (Station s)
    if s > this then succ.election(s);
    else if s = this then succ.elected(this);
    else if !electing then succ.election(this);
    electing := true;

port elected (Station s)
    if s  $\neq$  this then succ.elected(s);
    coordinator := s;
    electing := false;
```