Freie Universität Berlin

Using Physical Clocks for Replication in MANETs

Manuel Scholz,
Frank Bregulla

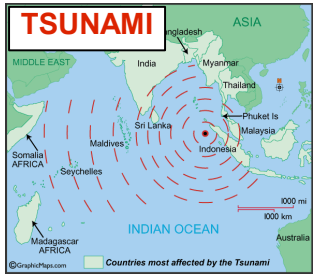
Freie Universität Berlin
Institute of Computer Science
14195 Berlin, Germany
{mscholz, bregulla}@inf.fu-berlin.de

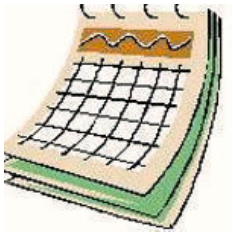
Annike Hinze

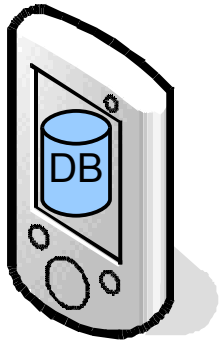
University of Waikato
Dept of Computer Science
Hamilton, NZ
hinze@cs.waikato.ac.nz

Motivation

- Why physical timestamps in mobile environments?
- Examples:







- Disaster areas: coordination of firefighters and helpers
- Mobile tourist information service

➔ Most recent information is needed

- Distributed calendar
- Distributed (mobile) database

➔ Resolve conflicting operations: abort the younger one (first come first serve)

Overview

- Introduction
- Scenario
- Synchronization Protocol
- Skew Vectors
- Grid Time
- Correctness
- Conclusion & Future Work

Introduction

- Goal: ordering of concurrent operations with physical clocks
  - Problem: imprecisely synchronized physical clocks
- Basic idea of our hybrid approach:
  - Physical clocks are used for temporal distant operations
  - Logical clocks are used for temporal close operations
  - Time grid
- Problems
  - What exactly is temporal close / distant?
  - Peers in MANET have to make same decision

Scenario

- Short term scenarios: one hour up to three days
  - Clock drift is disregarded
- 10 -100 participating peers forming a replication group
- Communication via WLAN (802.11)
- Update anywhere / multimaster replication
  - Operations: first executed locally and then propagated
  - Dissemination to all peers (sync. protocol)

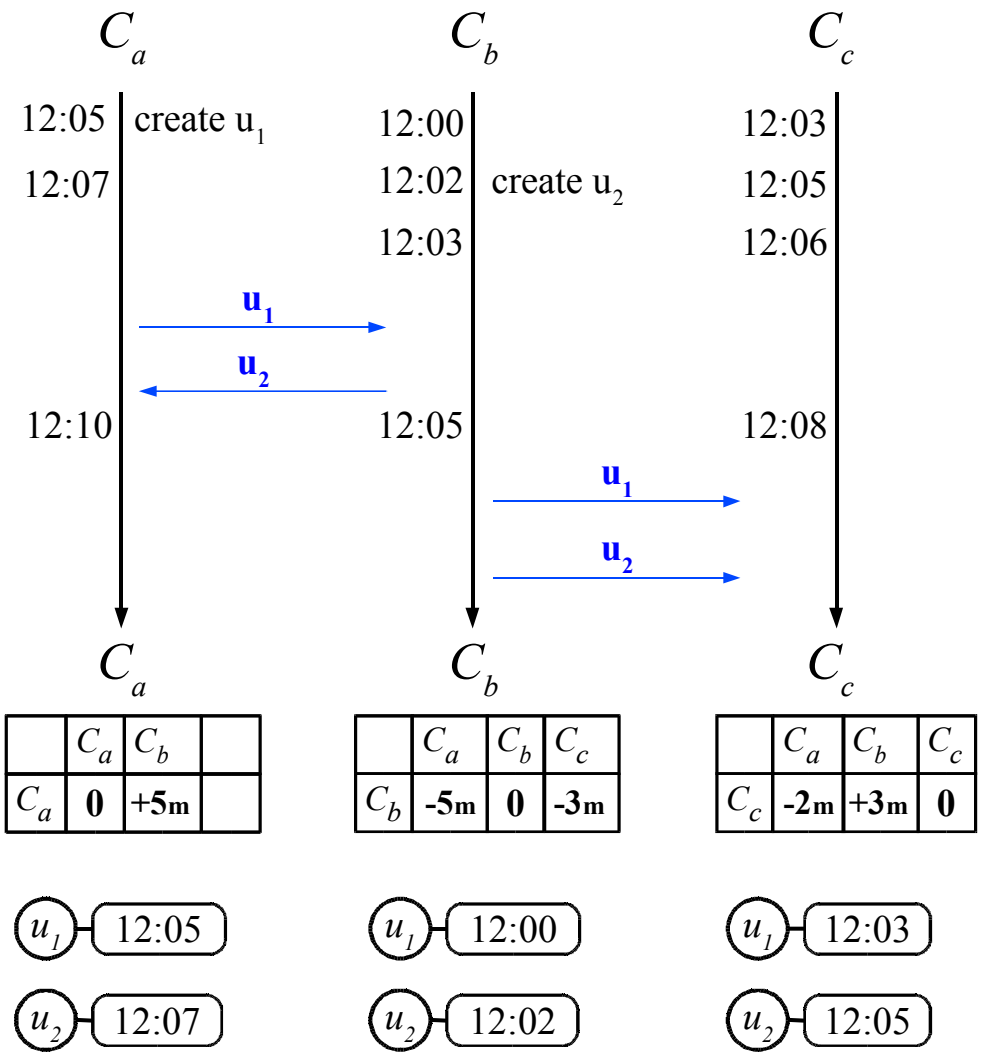
  ➔ Concurrent update operations
- Goal: global consistency – operations in same order

Time Synchronization Protocol

- External synchronization protocols (e.g. NTP) not suitable for MANETs
- Executed when peers meet for the first time
- Needs a predefined upper bound for round-trip of sync. message δ_{mrt}
- Simple and lightweight protocol:
  - Peers exchange their local clock values
  - When round-trip time of sync. message < δ_{mrt} protocol is finished
  - If not: protocol is repeated
  - Protocol also determines skew

Skew Vectors

- Every peer stores time skew to all other peers
- Different local time on all peers
- But: same order on all peers
- Example:
  - Updates (u_1 ,u_2) have diff. times on diff. peers
  - u_2 always 2 min. after u_1

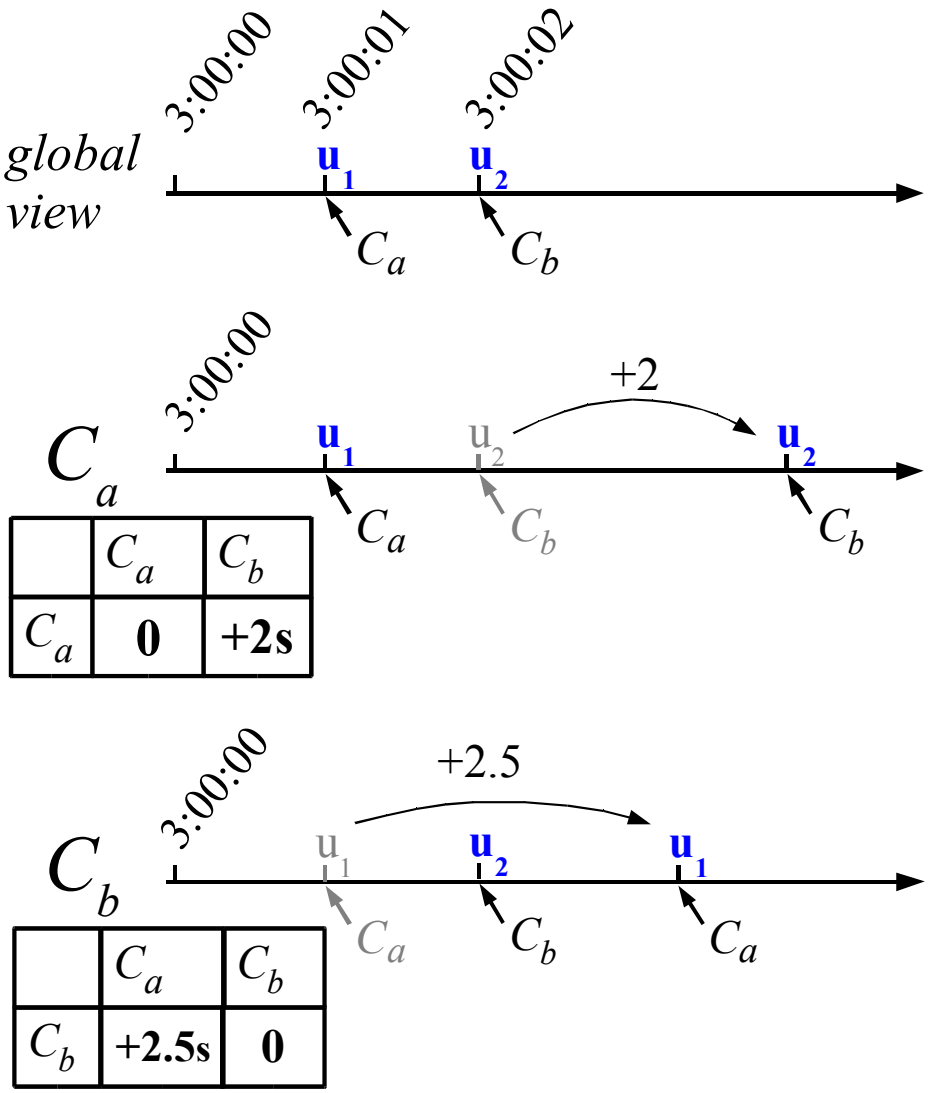


Message Delay Problem

- Message delay can cause different order

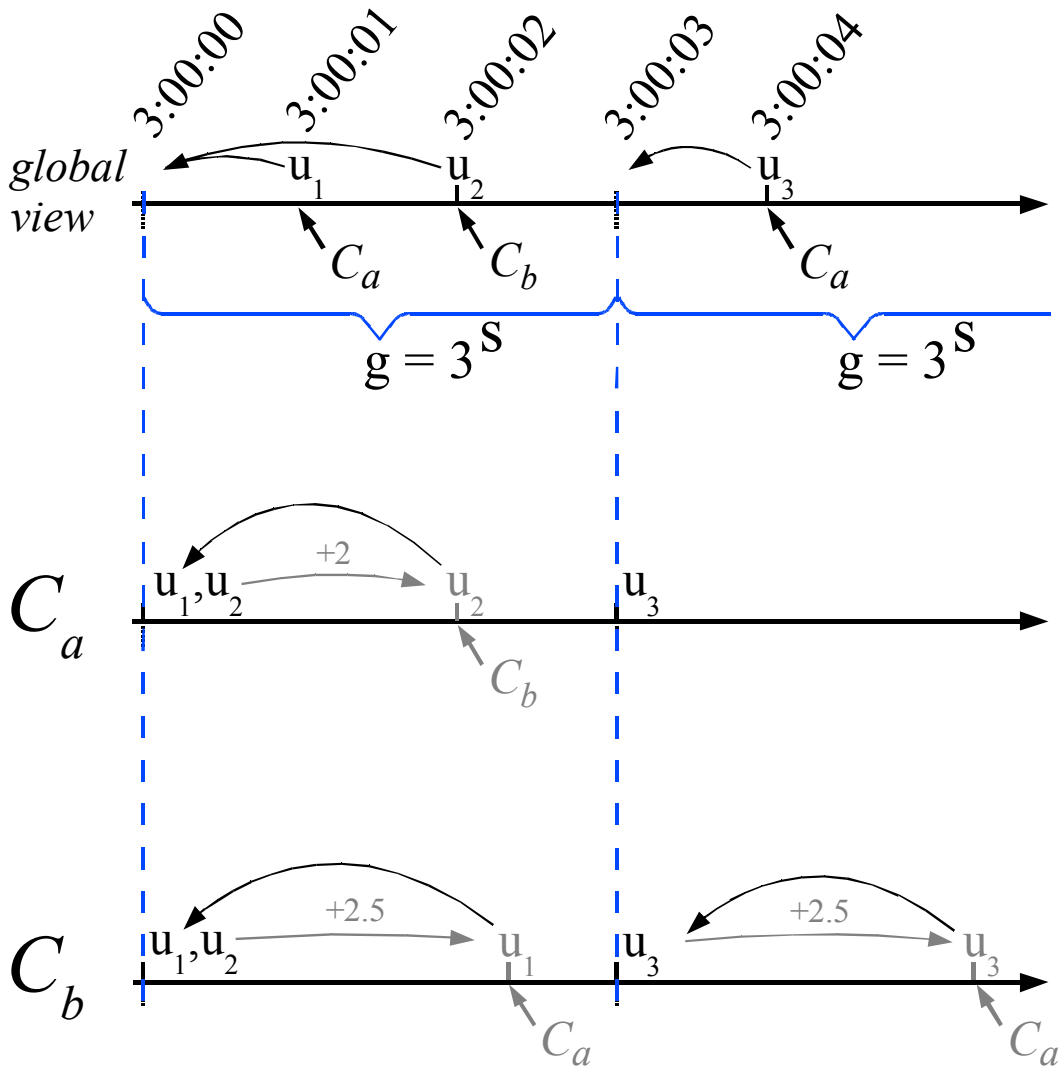
- Example:
  - Only 1 sec. between u_1 and u_2
  - Delay 2 and 2.5 sec

➔ u_1 and u_2 are ordered differently



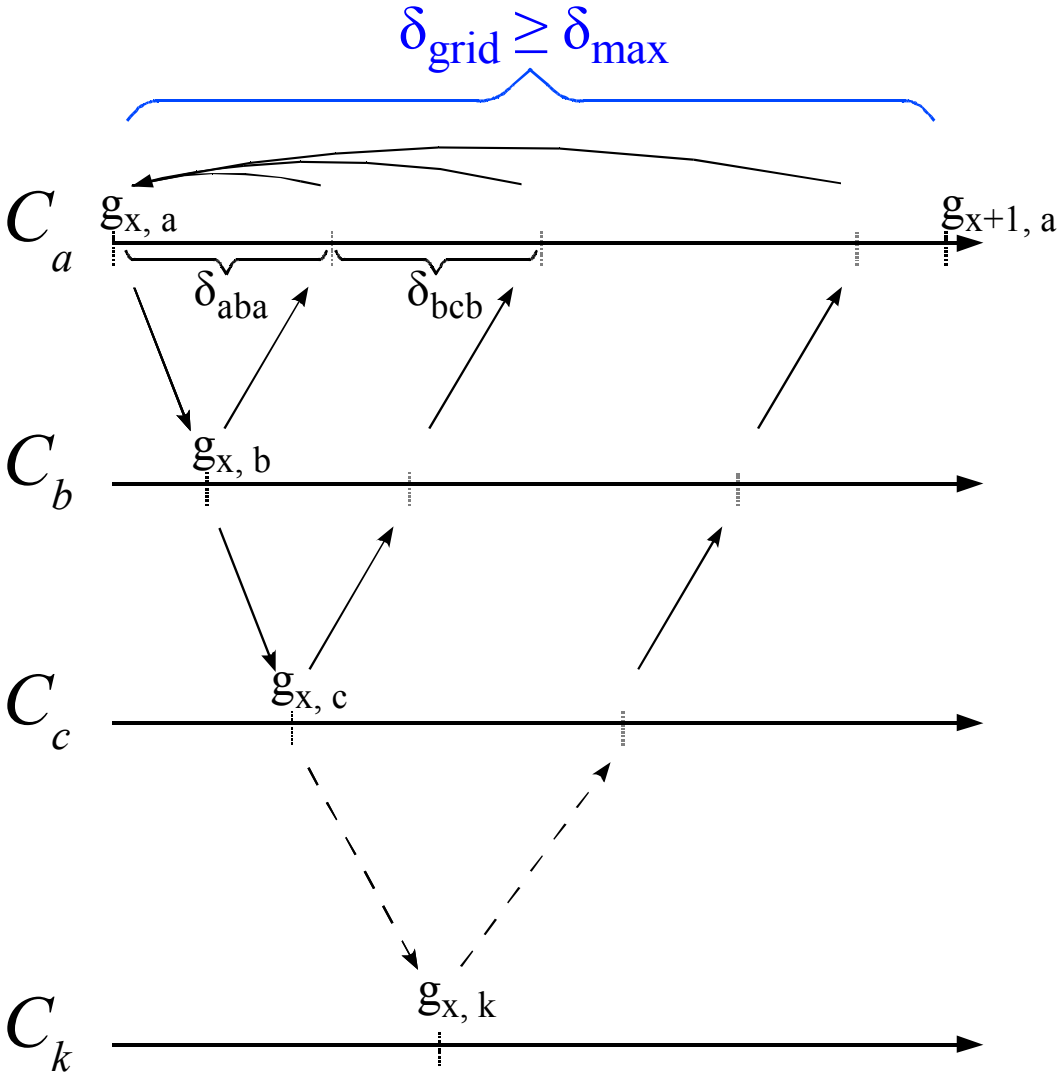
Grid Time

- Overlay time grid is used
- Timestamps are assigned to a grid time-slot
- Example:
  - u_1 and u_2 are in the same slot
  - u_3 is in the next slot



Grid Width

- Grid width must be greater than max delay
- Max delay: single delay $\times$ max hops
- Single delay = δ_{mrt}
- Max hops = (no peers) - 1



Correctness

To prove correctness of method we have to show:

1. All operations are in the same order on all peers

   ➔All grid values of a peer are assigned to the corresponding grid values on all other peers.

2. All operations with a temporal distance greater than a given value ($2\delta_{grid}$) are ordered according to their physical clocks

Correctness (Same Order on All Peers)

- Def. grid function:

$$\mathrm{grid}_a(t) = \left\lfloor \frac{t - \mathrm{offset}_a}{\delta_{\mathrm{grid}}} \right\rfloor \cdot \delta_{\mathrm{grid}} + \mathrm{offset}_a$$

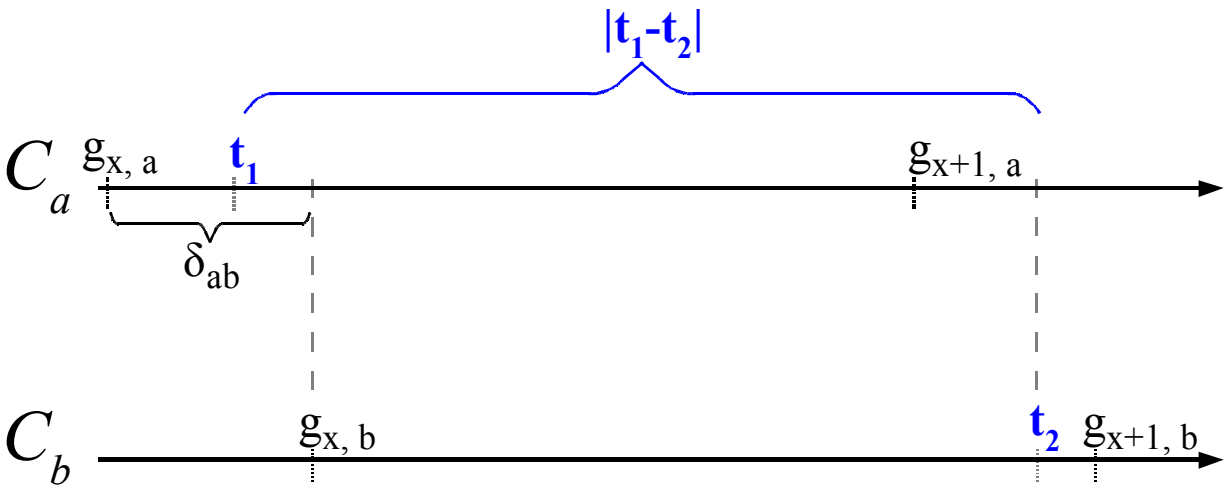
- Proof:

$$\begin{aligned} g_{x,a} &= \mathrm{grid}_a(g_{x,b} + \text{m-skew}_{ba}) \\ &= \mathrm{grid}_a(g_{x,a} + \text{m-skew}_{ab} + \text{m-skew}_{ba}) \\ &= \mathrm{grid}_a(g_{x,a} + \mathrm{skew}_{ab} + \delta_{ab} + \mathrm{skew}_{ba} + \delta_{ba}) \\ &= \mathrm{grid}_a(g_{x,a} + \delta_{ab} + \delta_{ba}) \\ &= \mathrm{grid}_a(g_{x,a}) \end{aligned}$$

$$\begin{aligned} g_{x,b} &= g_{x,a} + \text{m-skew}_{ab} \\ \text{m-skew}_{ab} &= \mathrm{skew}_{ab} + \delta_{ab} \\ (\mathrm{skew}_{ab} &= -\mathrm{skew}_{ba}) \\ (\delta_{ab} + \delta_{ba} &< \delta_{\mathrm{grid}}) \end{aligned}$$

□

Correctness (Physical Clock Order)



- In which case are t_1 and t_2 ordered according to their physical clocks?
- Without message delay: minimal distance δ_{grid}
- With message delay: worst case grid slots are shifted by δ_{grid} because grid values cannot overlap
- $|t_1-t_2| > 2\delta_{grid}$ ➔ physical clock ordering

Conclusion & Future Work

- Hybrid timestamping mechanism (time grid):
  - Physical clocks are used for temporal distant operations (different grid slots)
  - Logical clocks are used for temporal close operations (same grid slot)
- Local skew vectors are used to store the skew among the peers
- Grid size is determined by single round trip time δ_{mrt} and max number of hops needed for initialization
- Next steps:
  - Refinement (time drift, bounded max hops, etc.)
  - Further tests of implementation (emulation ,repl. system)

Thank You!