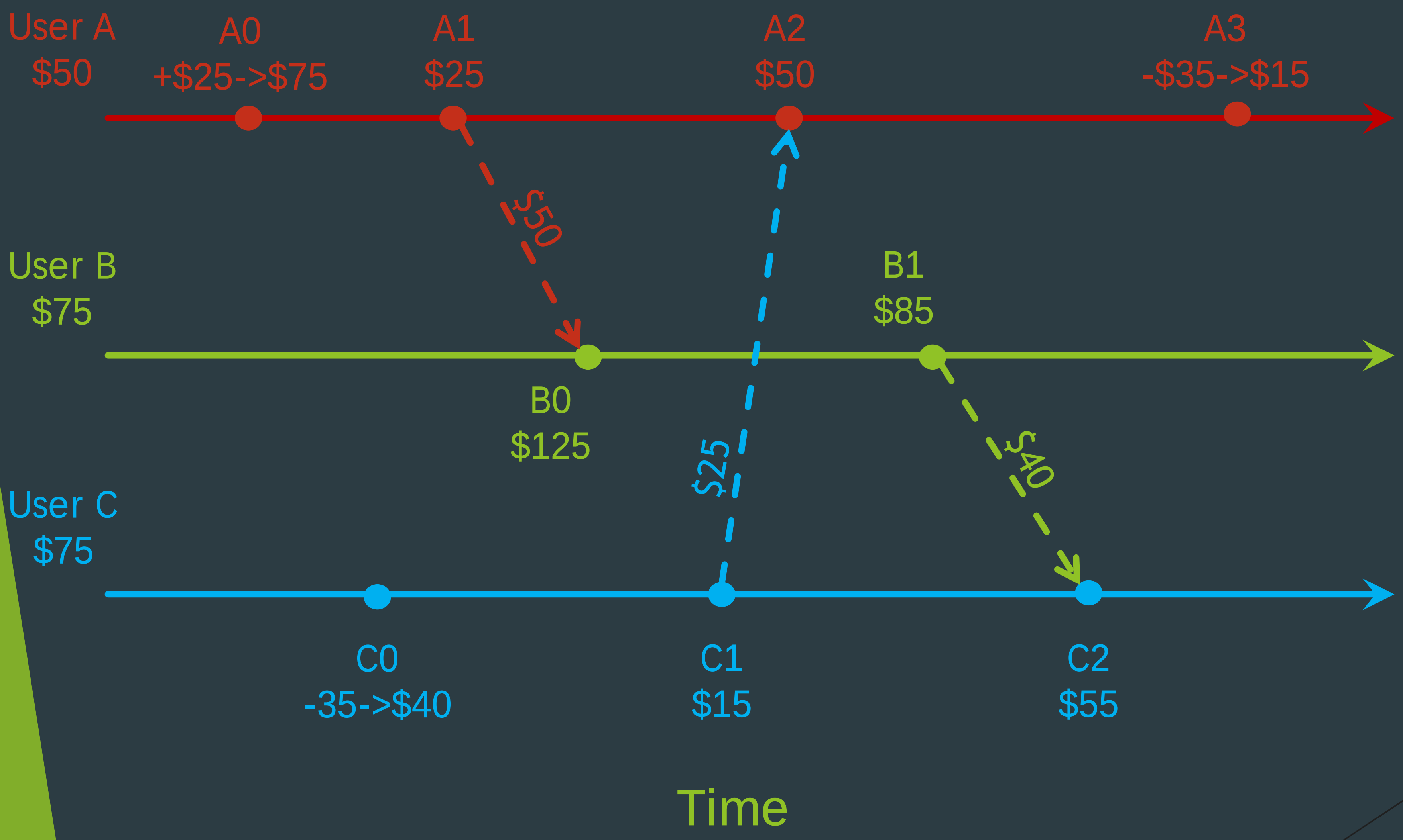
# Distributed Snapshots – Lecture 2

Temporal Accuracy

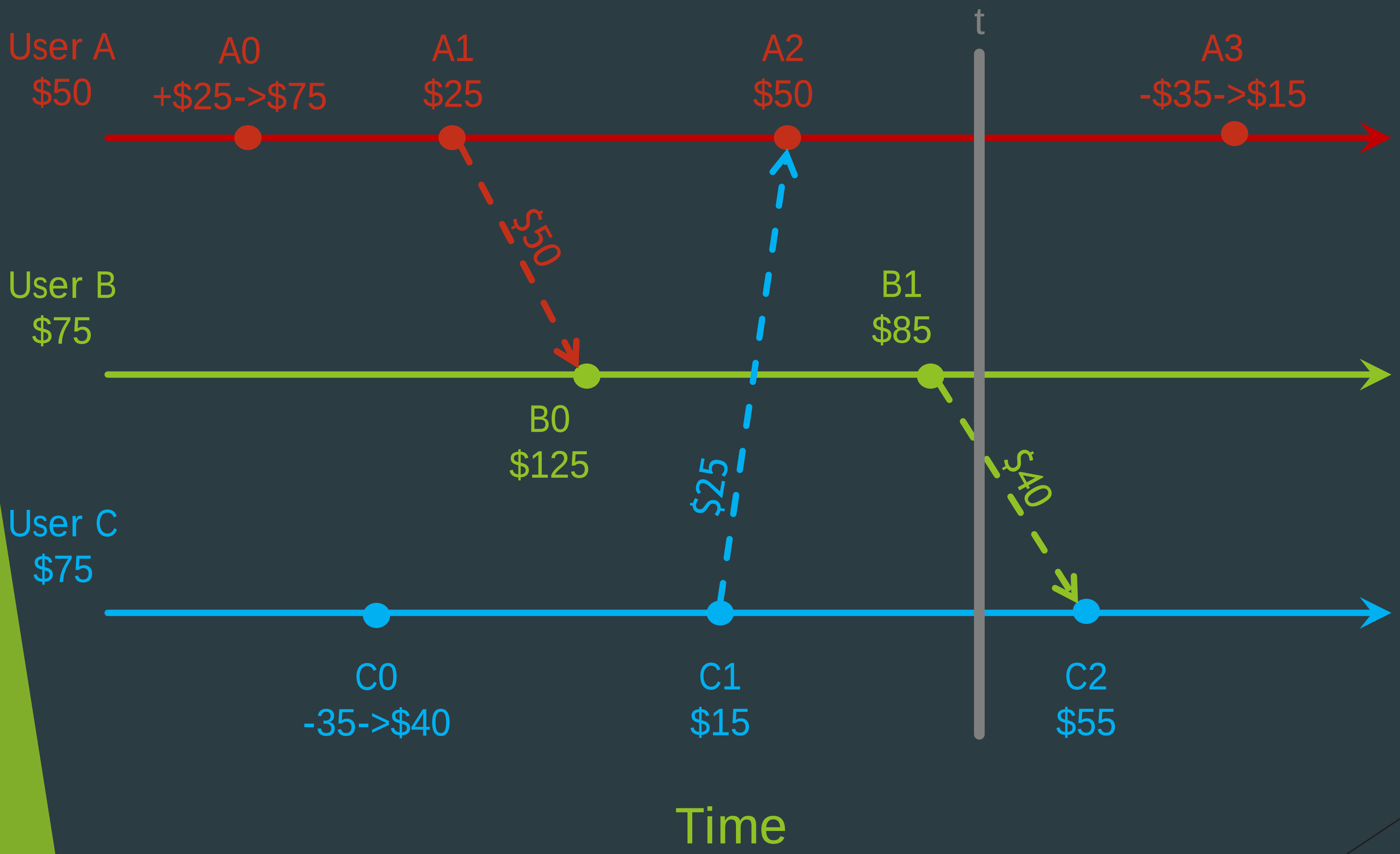
# Assumptions

- Failures
  - No failures.
- Network
  - Asynchronous -> A message might take arbitrary time to be delivered.
  - Reliable -> Messages cannot be lost or duplicated while in transit.
  - FIFO -> A network channel maintains the order of messages (e.g. If node A sends message 1 and 2 in that order to B, then B is going to receive them in the same order).
- Clock Synchronization
  - Adjusted to the presentation needs.

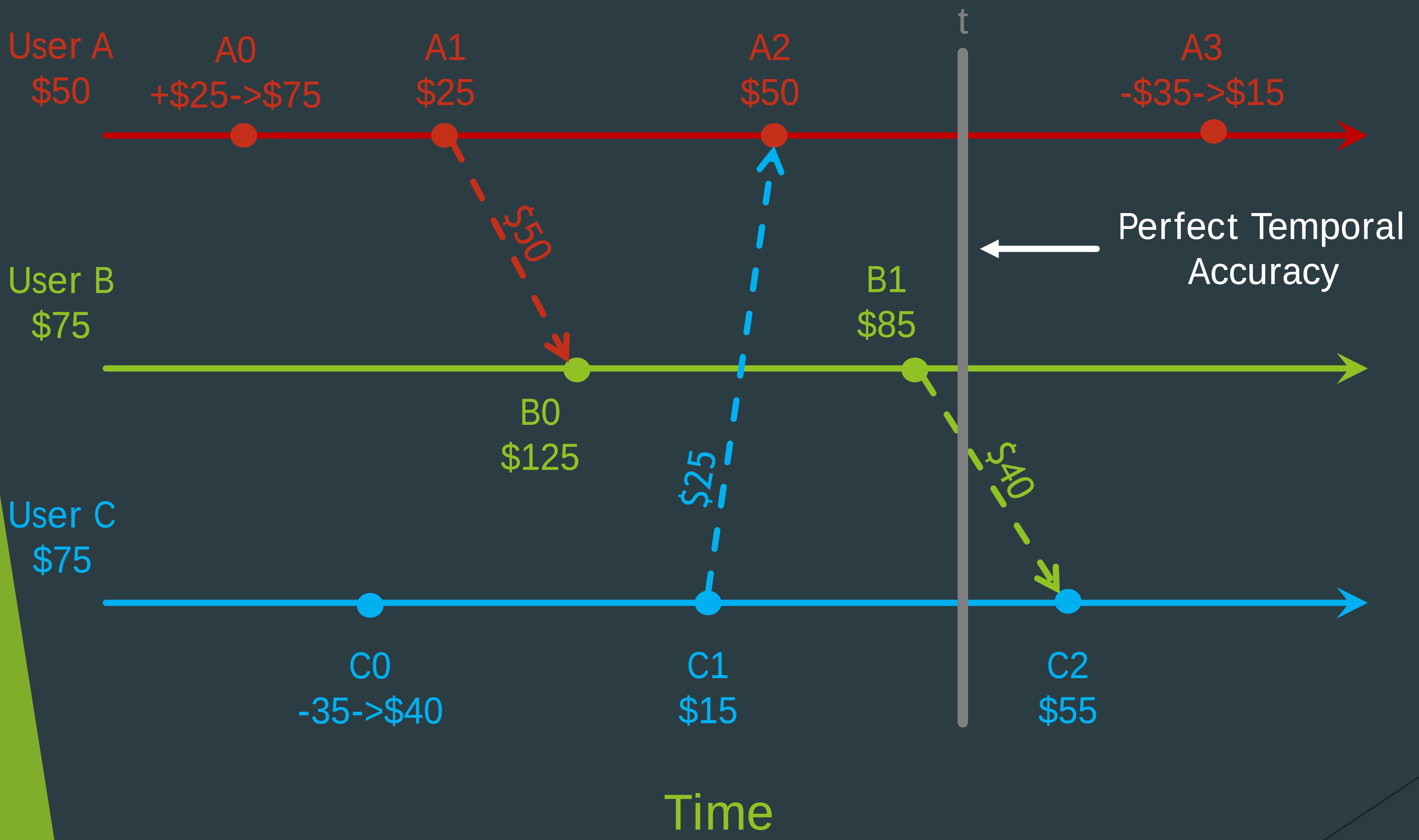
# Example – Bank Accounts



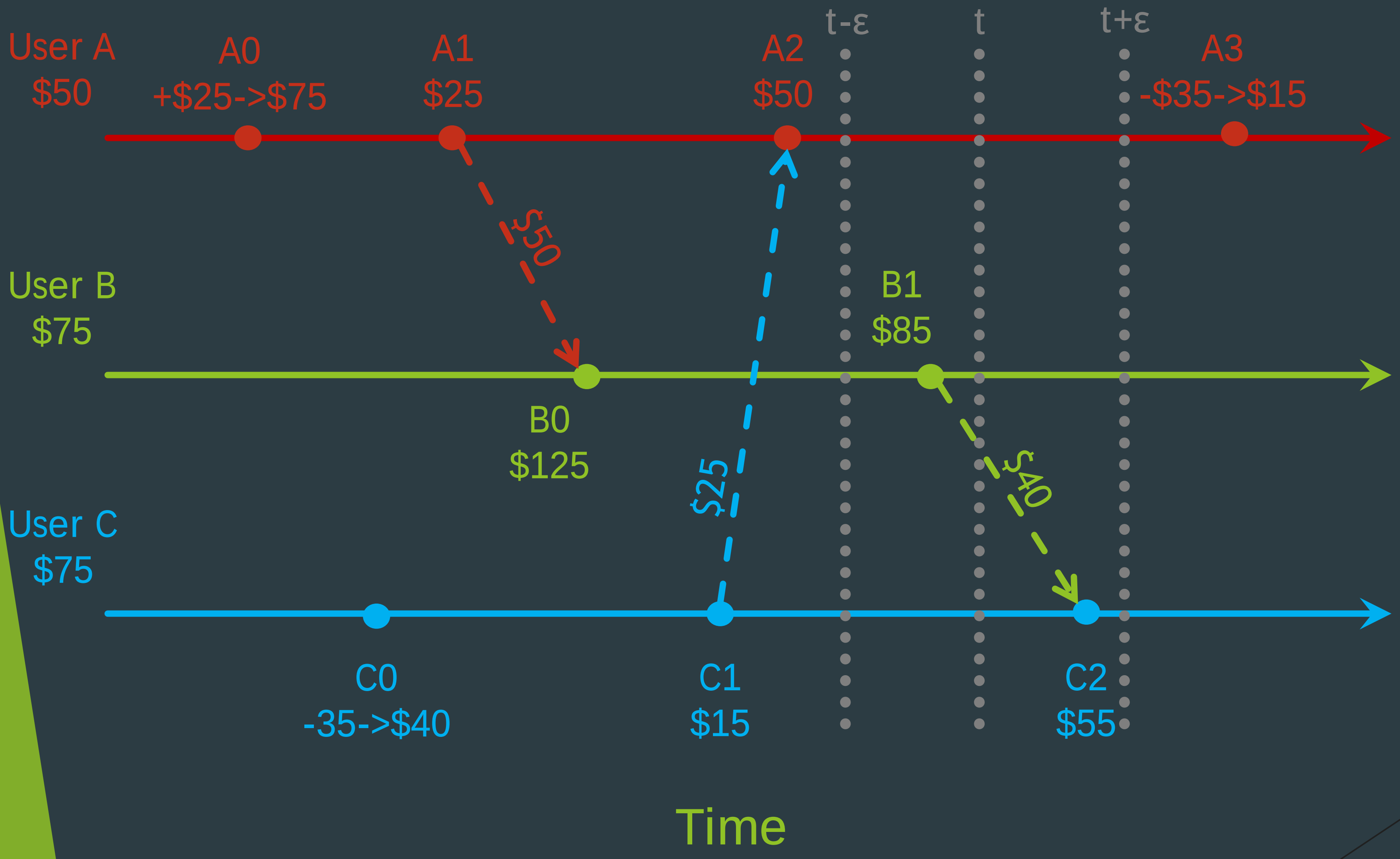
# Universal Time



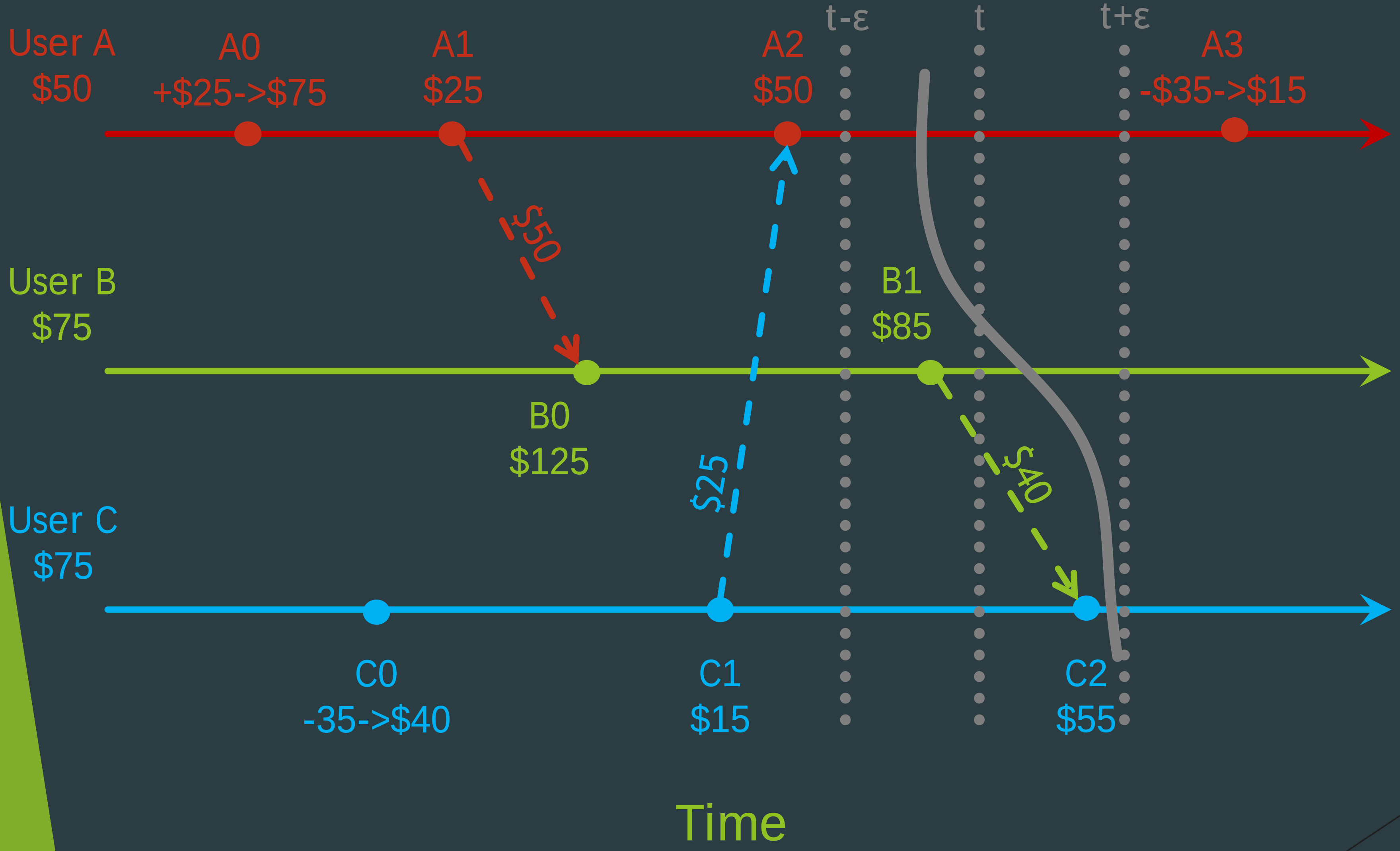
# Universal Time



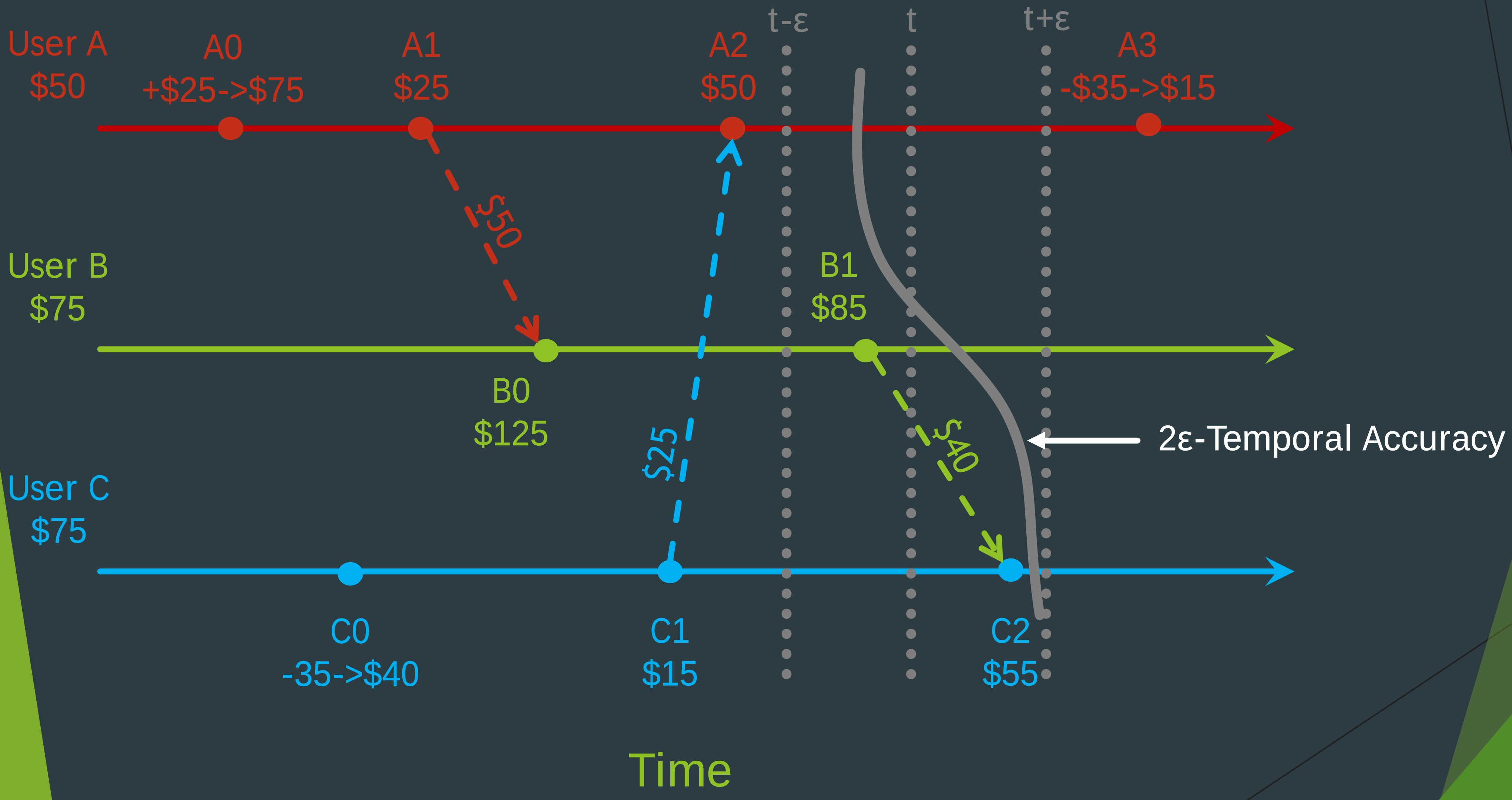
# Strongly-Synchronized Clocks



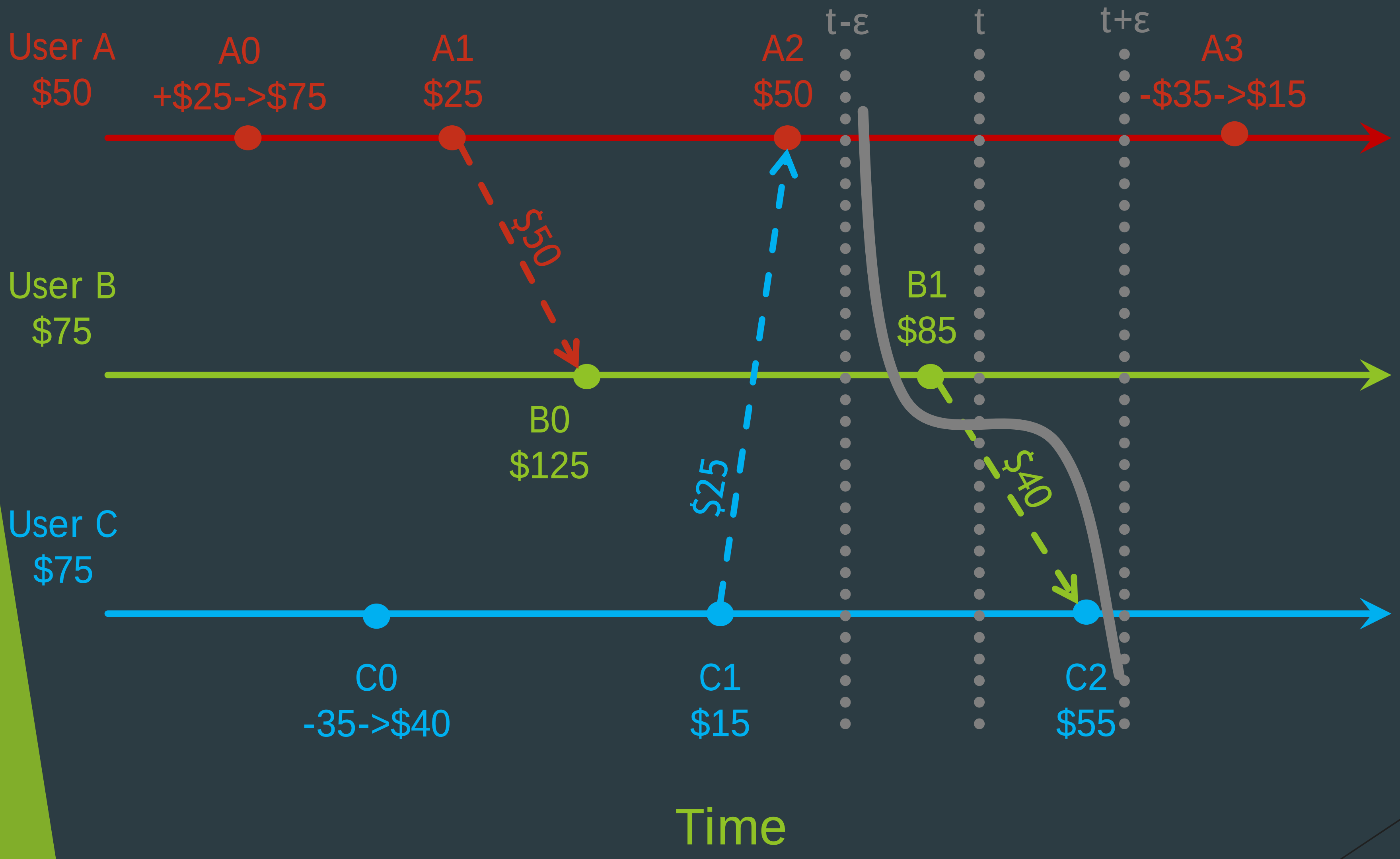
# Strongly-Synchronized Clocks



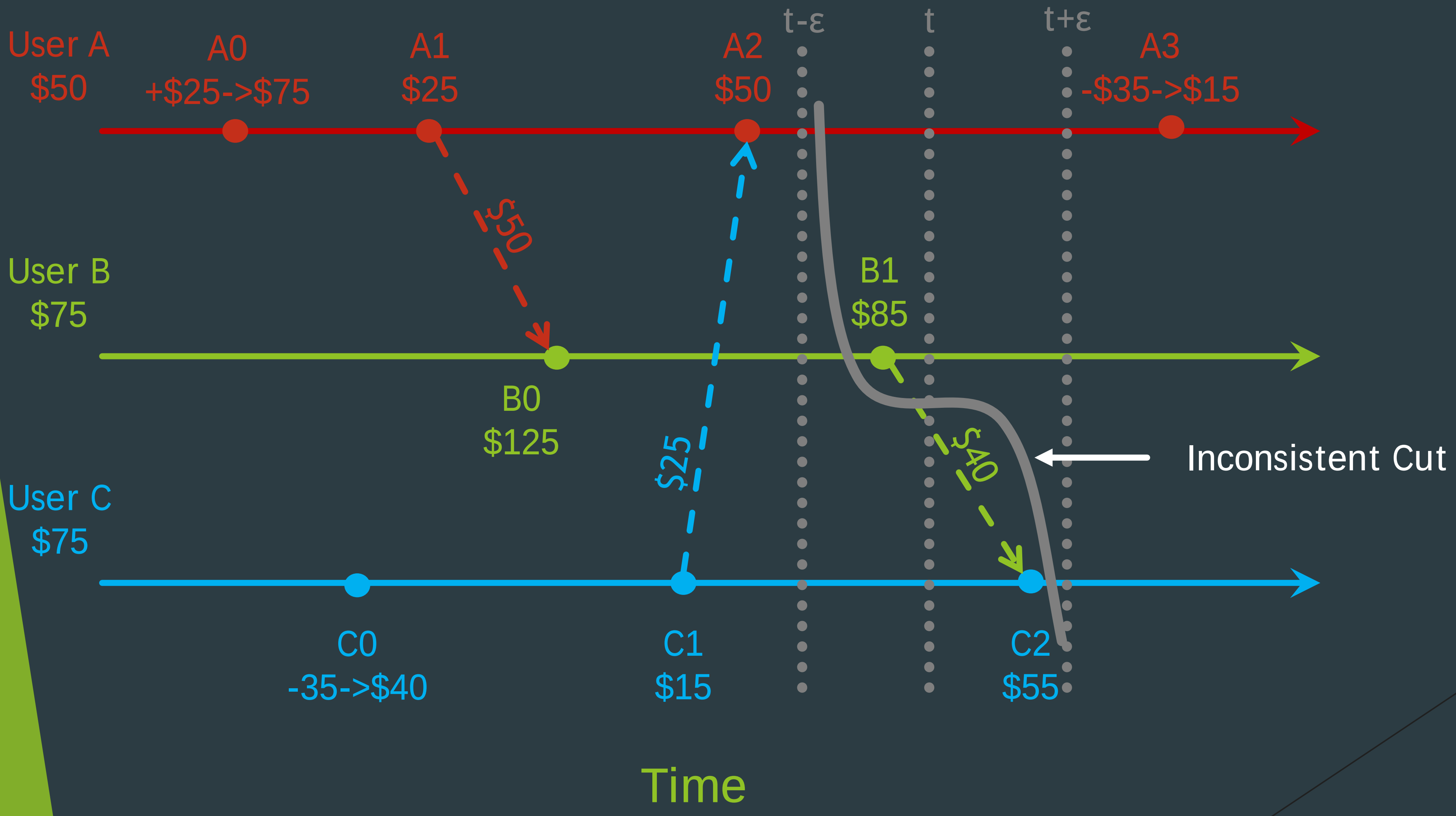
# Strongly-Synchronized Clocks



# Strongly-Synchronized Clocks



# Strongly-Synchronized Clocks



# Requirements

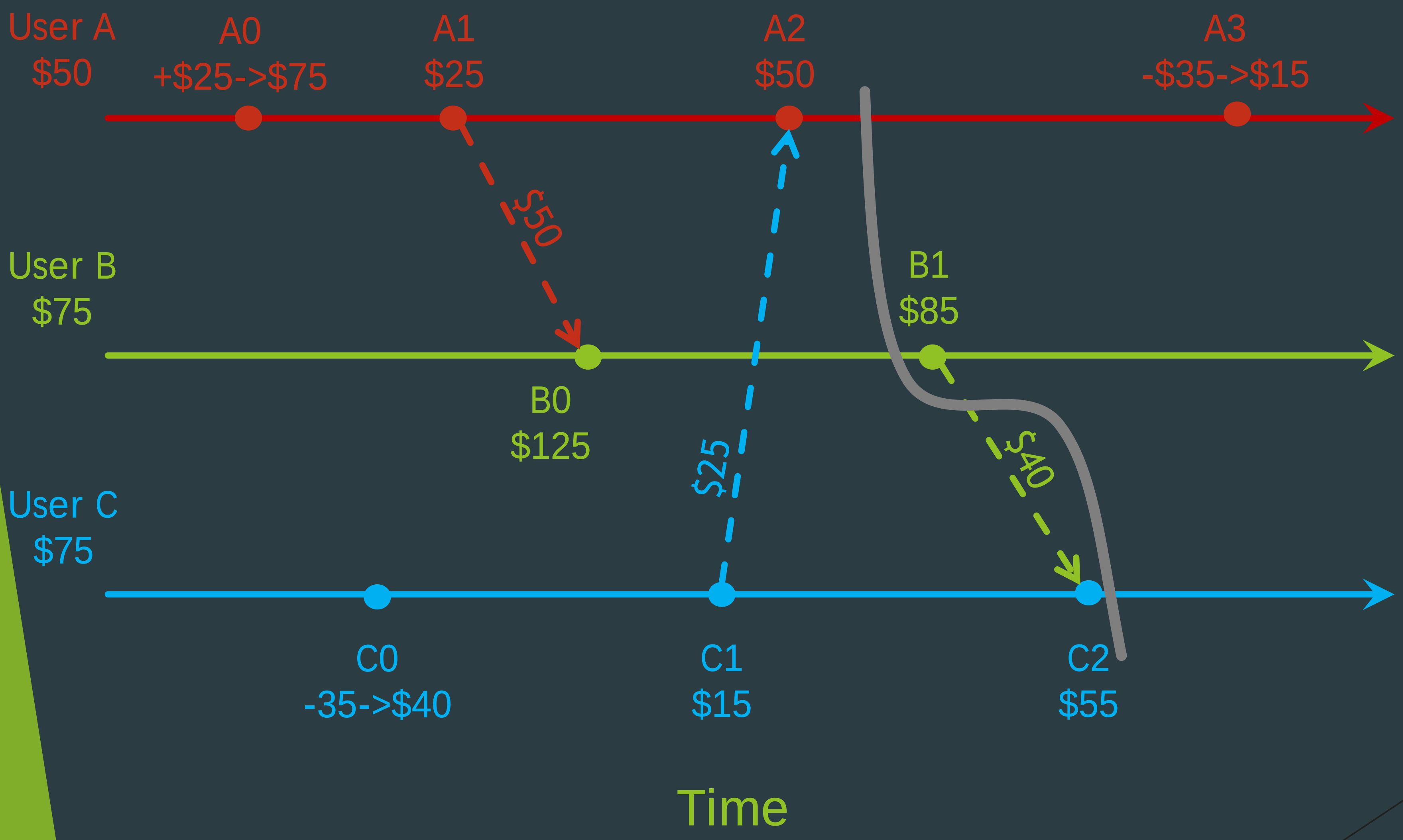
1. Causal Consistency
2. Temporal Accuracy

# Logical Clock - Revisited

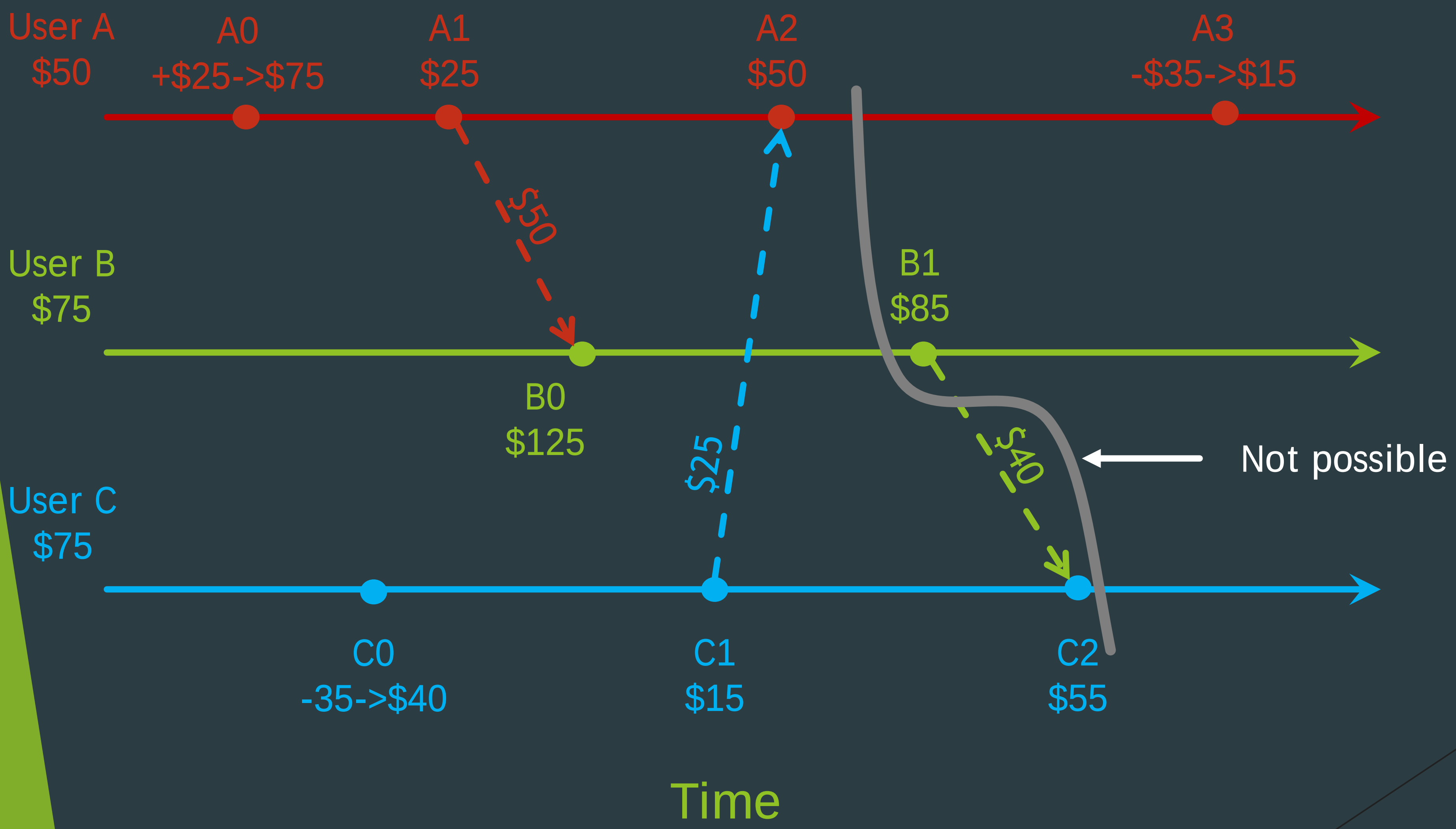
At process $P$:

1. On local event $e$:
$$LC(e) = \max\{LC^P + 1, PC^P\}$$
2. When sending a message $m$ to another process $Q$ ($e$):
$$LC(e) = \max\{LC^P + 1,\ PC^P\}$$
Send message $m, LC(e)$ to process $Q$.
3. When receiving a message $m, LC^m$ from process $Q$ ($e$):
$$LC(e) = \max\{LC^P + 1, LC^m + 1, PC^P\}$$
4. We always set $LC^P$ to $LC(e)$ after we finish executing the events.

# Logical Clocks – Is this possible?



# Logical Clocks – Is this possible?



# Causal Consistency – Logical Clocks

We know that:

$$e' \rightarrow e \Rightarrow LC(e') \leq LC(e)$$

Proof as exercise.

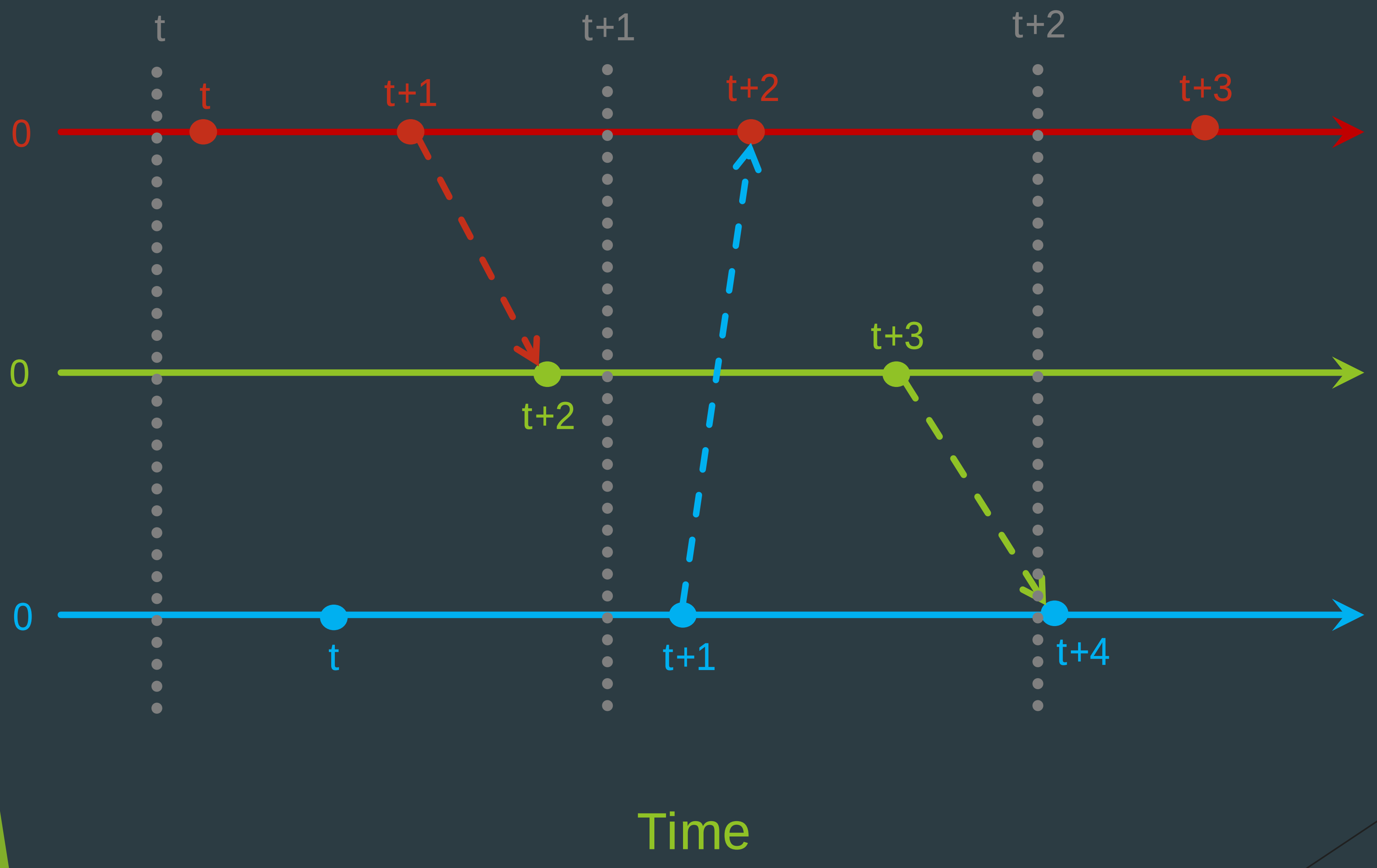
We take a snapshot $C$ where we include every event $e$ such that $LC(e) \leq t$. For all events $e' \rightarrow e$ such that $e \in C$, we have

$$LC(e') \leq LC(e) \Rightarrow$$
$$LC(e') \leq t \Rightarrow$$
$$e' \in C$$

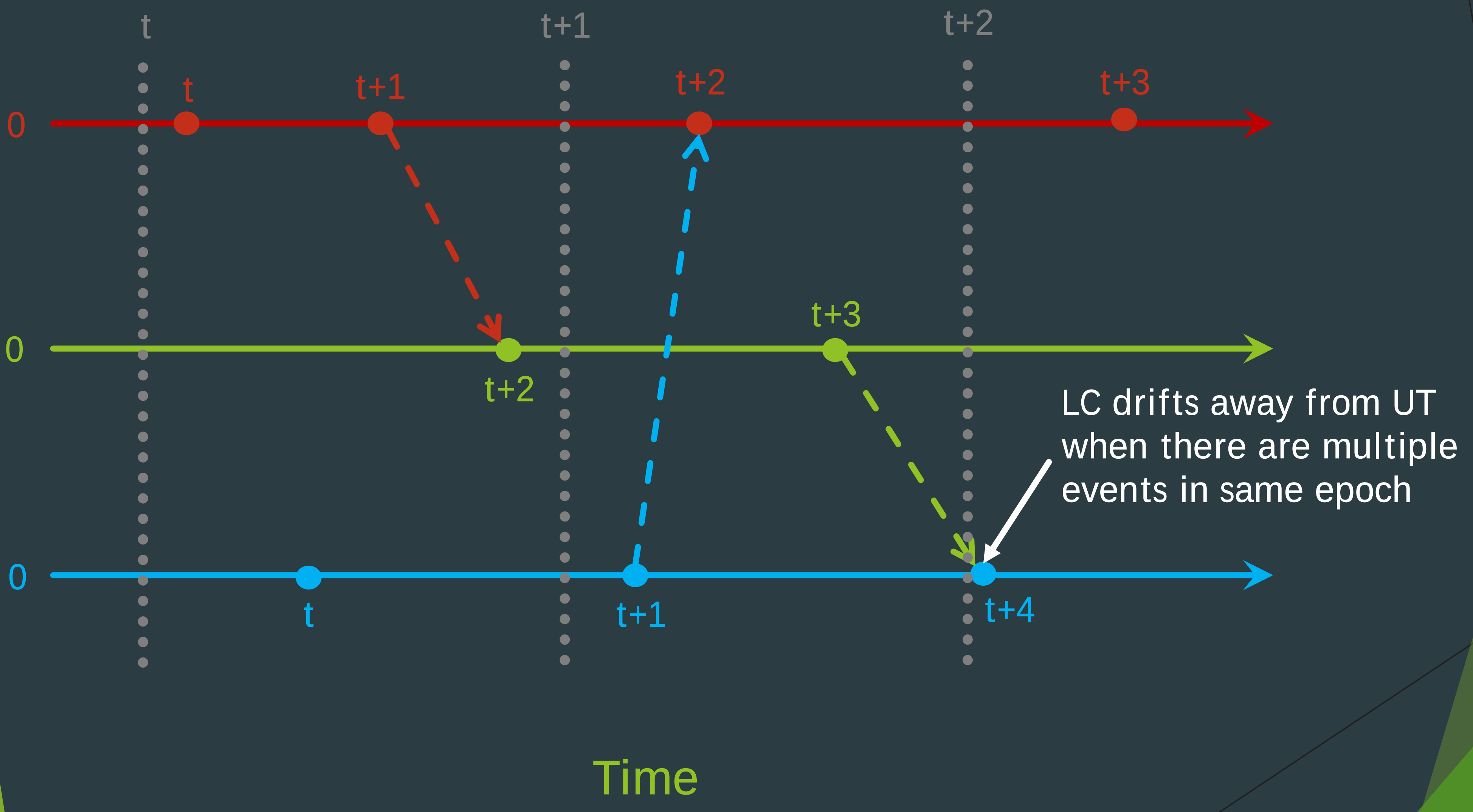
# Temporal Accuracy – Logical Clocks

Do we have Temporal Accuracy by using this scheme?

# LC – Perfectly Synchronized Clocks



# LC – Perfectly Synchronized Clocks



# Temporal Accuracy – Logical Clocks

Do we have ε-Temporal Accuracy by using this scheme?

**NO!**

# Hybrid Logical Clock

We know that:

$$HLC = (r, l)$$
$$HLC > HLC' \Leftrightarrow (r > r') \vee ((r = r') \wedge (l > l'))$$

At process $P$:

1. On local event $e$:
$$HLC(e) = \max\{HLC^P, (PC^P, -1)\} + (0,1)$$
2. When sending a message $m$ to another process $Q$ ($e$):
$$HLC(e) = \max\{HLC^P, (PC^P, -1)\} + (0,1)$$
Send message $m, HLC(e)$ to process $Q$.
3. When receiving a message $m, HLC^m$ from process $Q$ ($e$):
$$HLC(e) = \max\{HLC^P, HLC^m, (PC^P, -1)\} + (0,1)$$
4. We always set $HLC^P$ to $HLC(e)$ after we finish executing the events.

# Causal Consistency – HLC

We know that:

$$e' \to e \Rightarrow HLC(e') \leq HLC(e)$$

Proof as exercise.

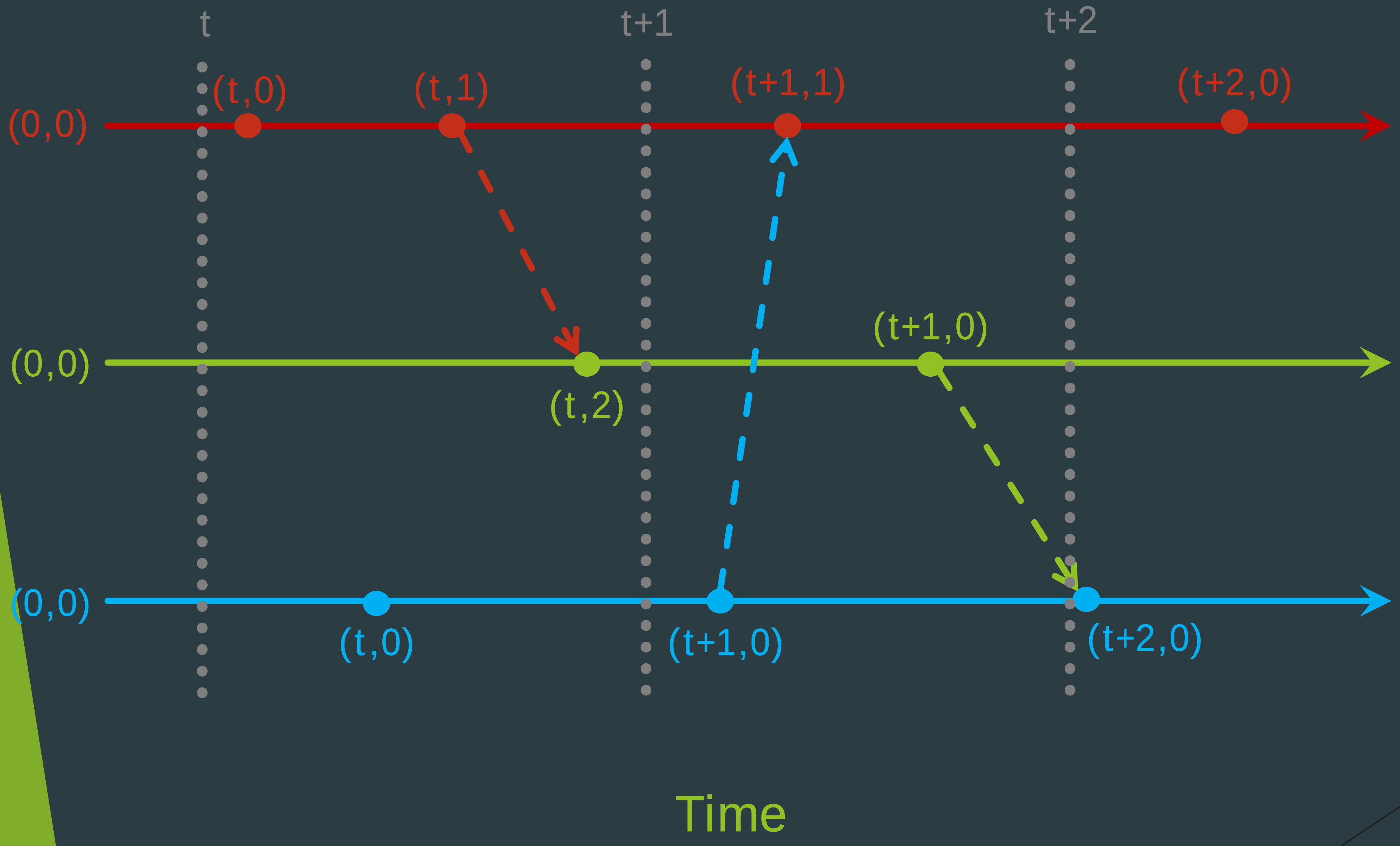
We take a snapshot $C$ where we include every event $e$ such that $HLC(e) < (t+1,0)$. For all events $e' \to e$ such that $e \in C$, we have

$$HLC(e') \leq HLC(e) \Rightarrow$$
$$HLC(e') \leq t \Rightarrow$$
$$e' \in C$$

# Temporal Accuracy – HLC

Do we have ε-Temporal Accuracy by using $HLC$?

# HLC – Perfectly Synchronized Clocks



# Temporal Accuracy – HLC

Do we have ε-Temporal Accuracy by using $HLC$?

It turns out, that if there is an unknown bound $\varepsilon$ (weakly-synchronized clocks) such that:

$$\forall P. |PC^P(t) - t| \leq \varepsilon$$

then we know that $HLC$ provide $\varepsilon$-Temporal Accuracy. In particular, if we take a snapshot $C$ at time $(t + 1,0)$ (exclusively):

1. For any event $e$ that happens before $\mathrm{t} - \varepsilon$, we have $\mathrm{e} \in C$
2. For any event $e$ that happens after $\mathrm{t} + \varepsilon$, we have $\mathrm{e} \notin C$

# Hybrid Logical Clock

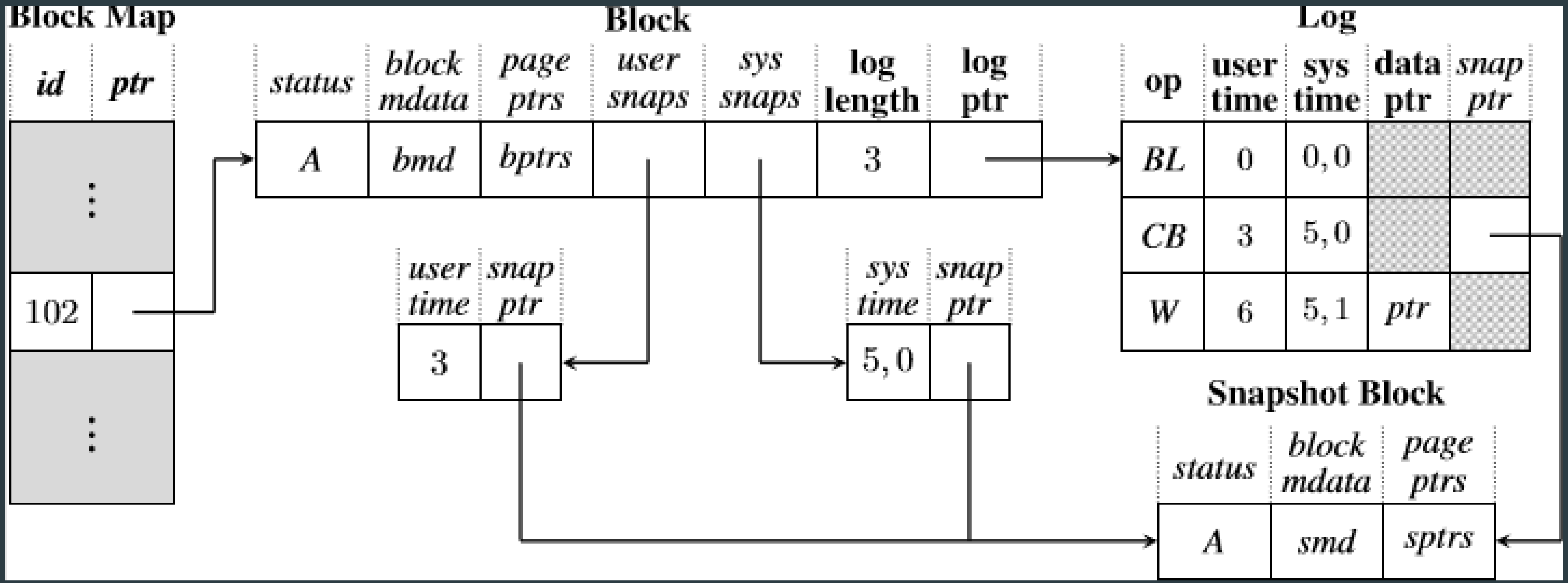
We still do not have the following property:

$$HLC(e) < HLC(e') \Rightarrow e \rightarrow e'$$

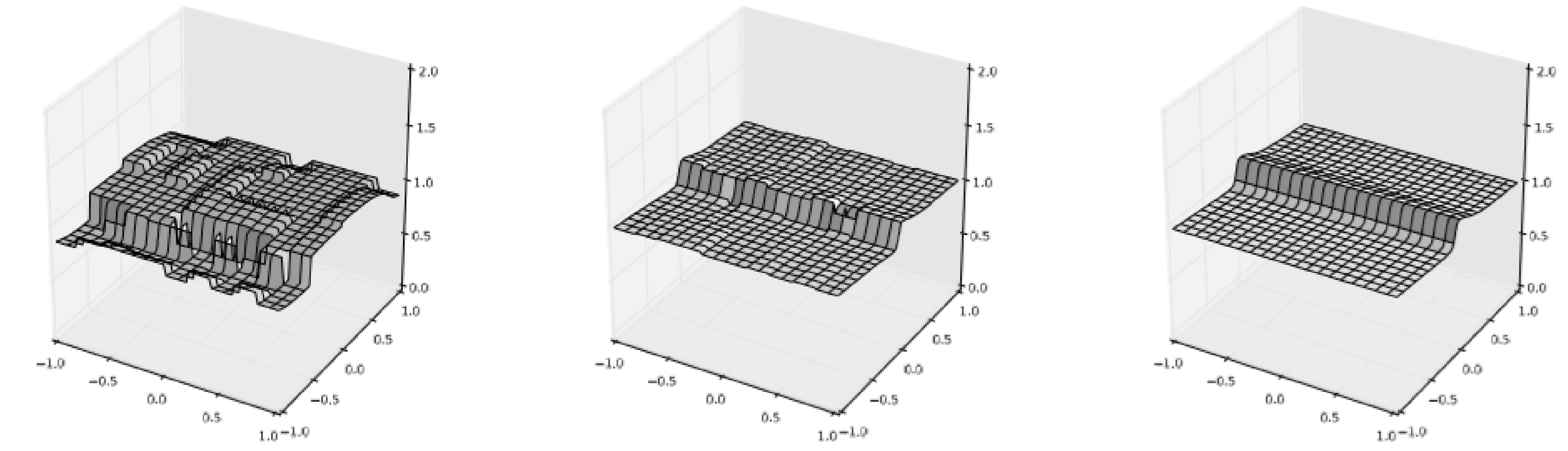
How we design Hybrid Vector Clocks?

Exercise

# FFFS

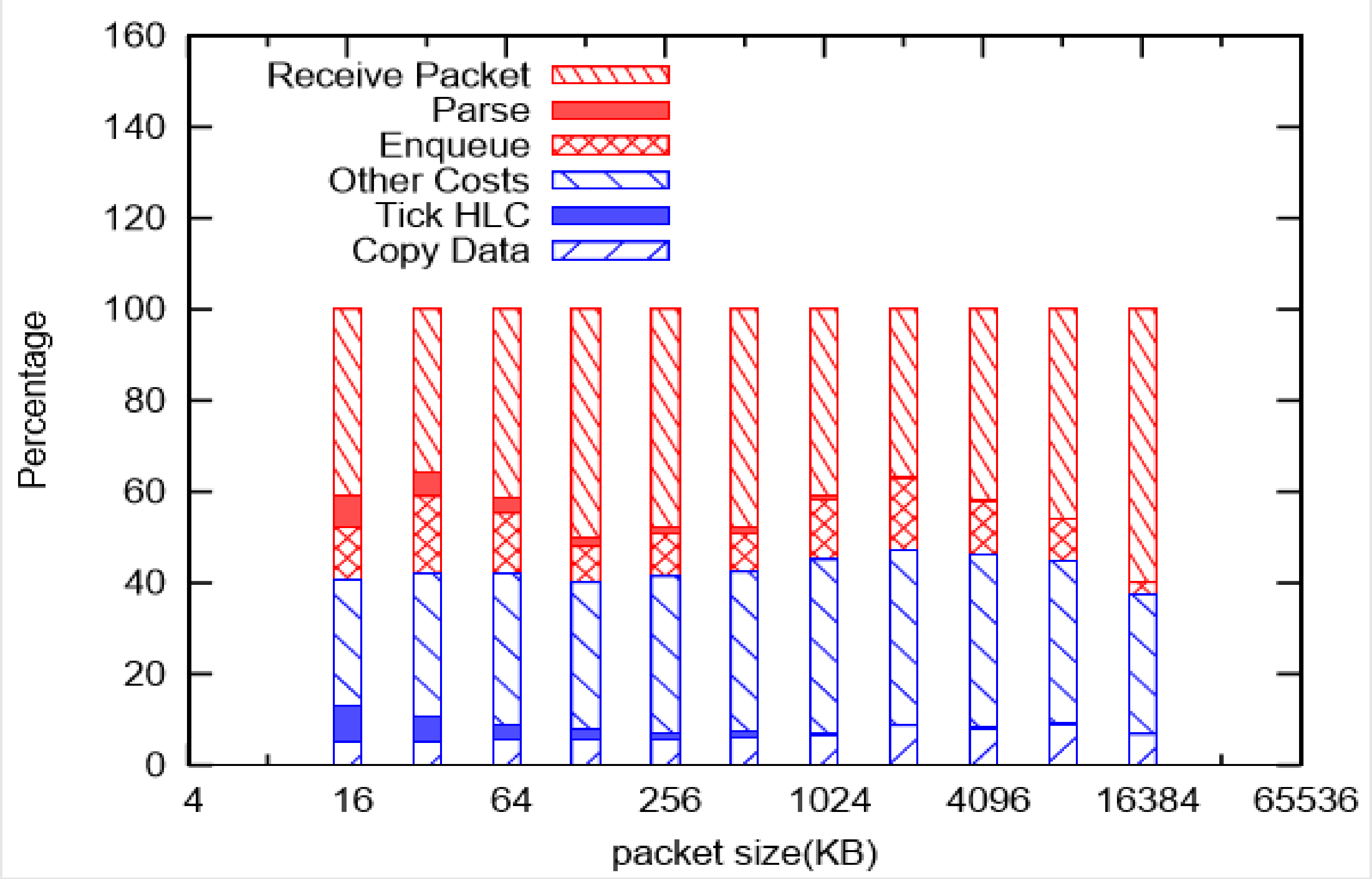

Block Map
id
ptr
102
Block
status
block mdata
page ptrs
user snaps
sys snaps
log length
log ptr
A
bmd
bptrs
3
user time
snap ptr
3
sys time
snap ptr
5, 0
Log
op
user time
sys time
data ptr
snap ptr
BL
0
0, 0
CB
3
5, 0
W
6
5, 1
ptr
Snapshot Block
status
block mdata
page ptrs
A
smd
sptrs

# FFFS – Temporal Accuracy




(a) HDFS (b) FFFS (server time) (c) FFFS (sensor time)

# FFFS – Overhead



# Questions?