Physical and Logical Clocks

17 May 2026
Lecture 9

Some Slide Credits: Maarten van Steen

Topics for Today

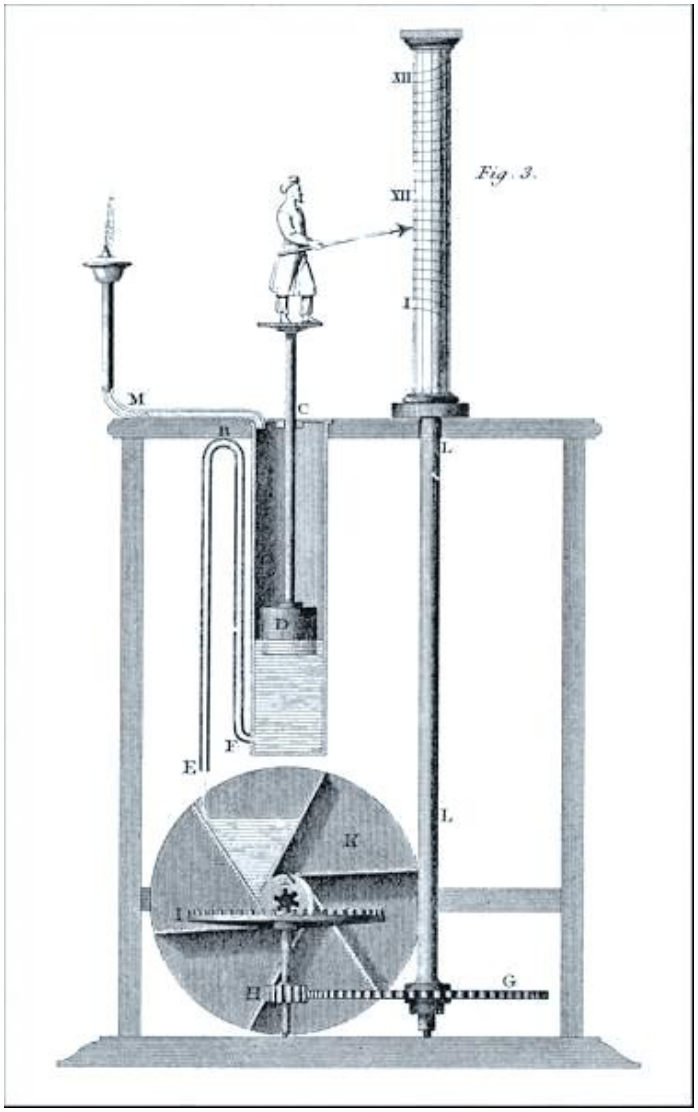
- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast

Source: TvS 6.1 - 6.2

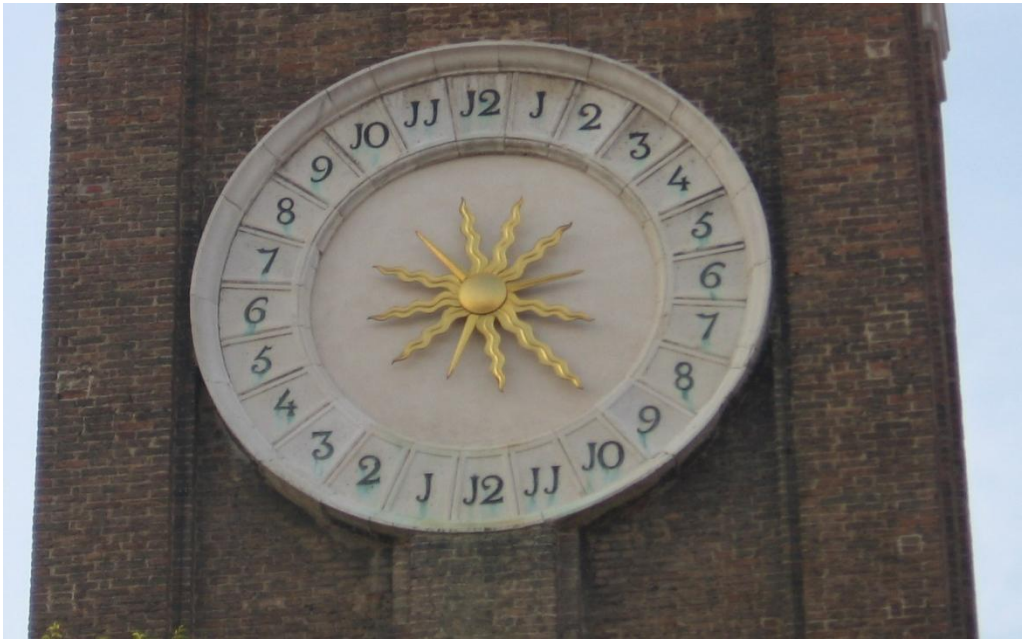
Sundials

Source: https://www.cornwalls.co.uk/photos/perranzabuloe-millennium-sundial-344.htm

Clocks



Source: By The illustrator was probably w:John Farey, Jr. (1791–1851). The principal engraver for the encyclopedia was Wilson Lowry (1762–1824).[1] [Public domain], via Wikimedia Commons



Physical Clocks (1/3)

Problem: Sometimes we simply need the exact time, not just an ordering.

Solution: Universal Coordinated Time (UTC):

- Based on the number of transitions per second of the cesium 133 atom (pretty accurate).
- At present, the real time is taken as the average of some 50 cesium-clocks around the world.
- Introduces a leap second from time to time to compensate that days are getting longer.
  - Decided in 2022 no more, decide by 2035 what to do

UTC is **broadcast** through short wave radio and satellite. Satellites can give an accuracy of about +/- 0.5 ms.

Atomic Clocks (USNO)

Image: By US Naval Observatory (http://tycho.usno.navy.mil/gif/clockvaults.jpg) [Public domain], via Wikimedia Commons

Israel Atomic Clock (HUJI)

Image source: http://www.davar1.co.il/15800/

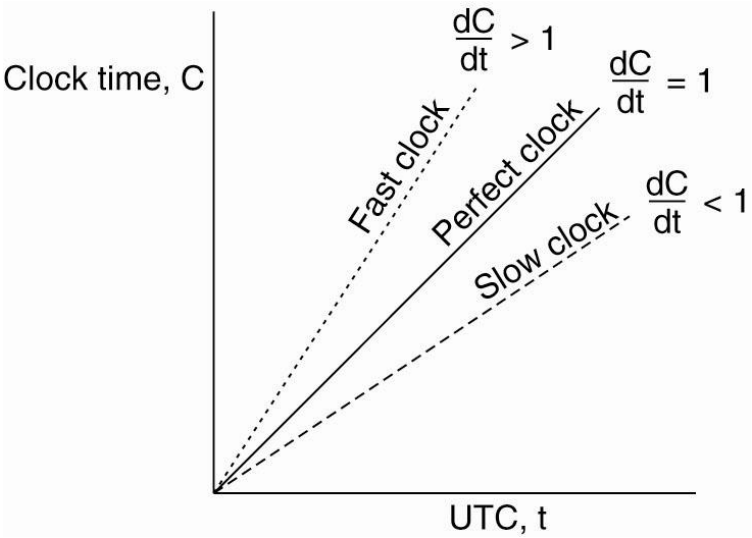
Physical Clocks (2/3)

Problem: Suppose we have a distributed system with a UTC-receiver somewhere in it → we still have to distribute its time to each machine.

Basic principle:

- Every machine has a timer that generates an interrupt H times per second.
- There is a clock in machine p that **ticks** on each timer interrupt. Denote the value of that clock by $\mathrm{C_p}(t)$, where t is UTC time.
- Ideally, we have that for each machine p, $C_p(t) = t$, or, in other words, $\frac{dC}{dt} = 1$.

Physical Clocks (3/3)



In practice: $1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$

Goal: Never let two clocks in any system differ by more than δ time units → synchronize at least every $\frac{\delta}{2\rho}$ seconds.

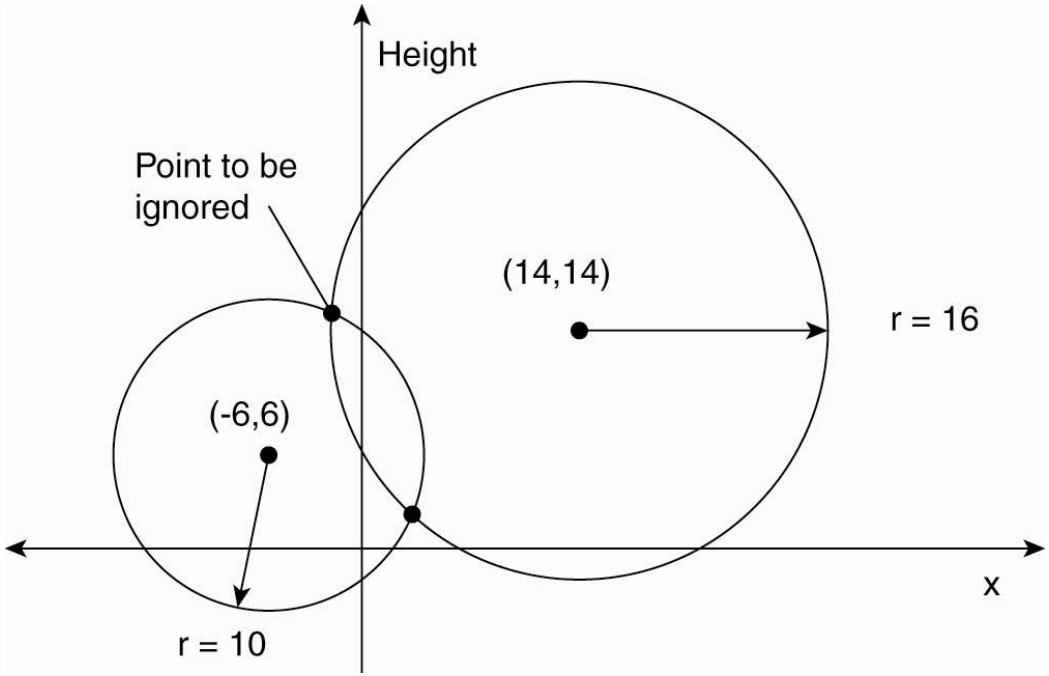
So Far

- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast

Triangulating location

Basic idea: You can get an accurate account of the time as a side-effect of GPS.

Principle:

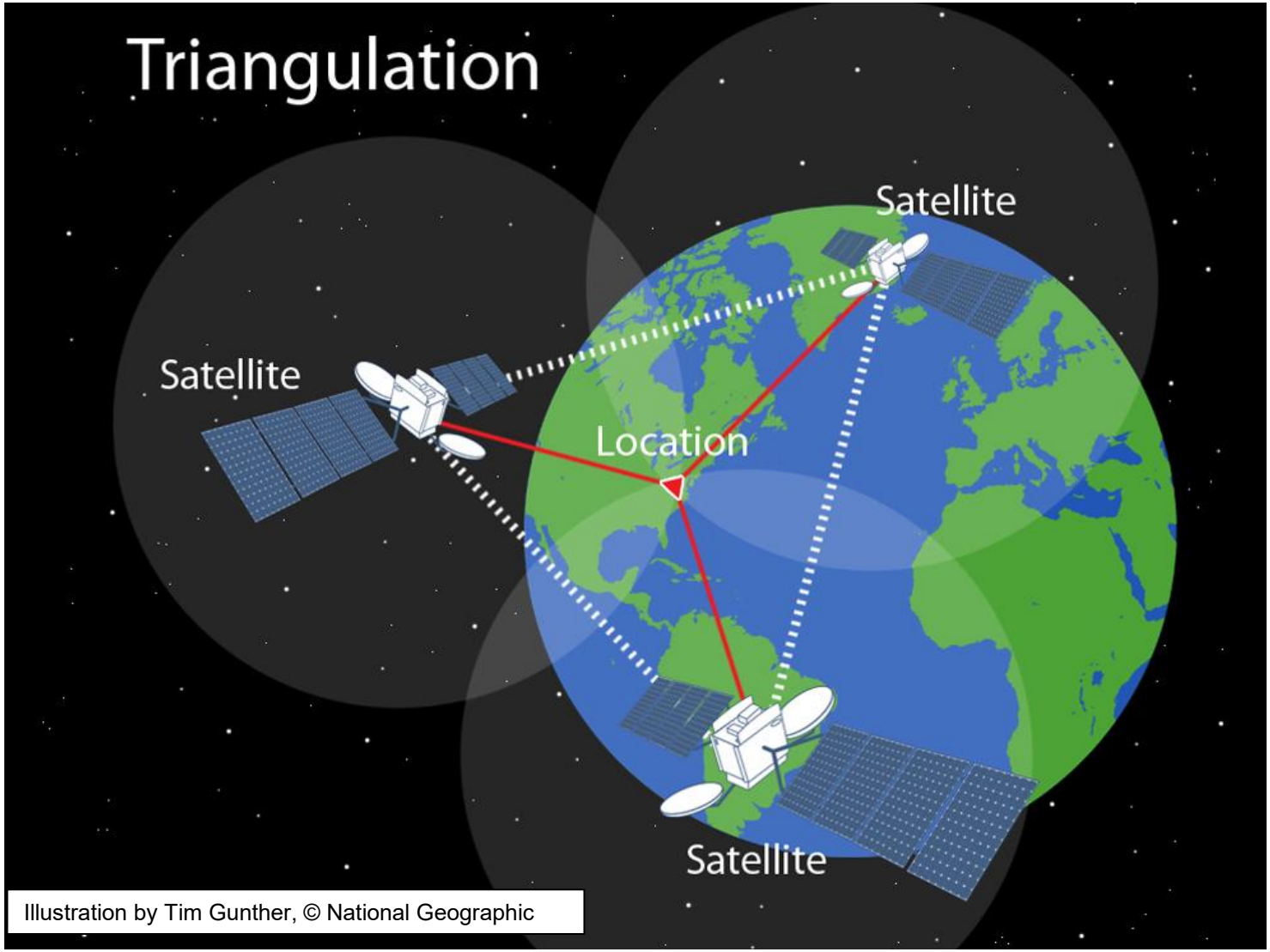


Problem:

Assuming that the clocks of the satellites are accurate and synchronized:

- It takes a while before a signal reaches the receiver
- The receiver's clock is definitely out of synch with the satellite

Can do it in space too



Getting time from location

- Δ_r : unknown deviation of the receiver's clock.
- x_r, y_r, z_r: unknown coordinates of the receiver.
- T_i is timestamp on a message from satellite i
- $\Delta_i = (T_{now} - T_i) + \Delta_r$: measured delay of the message sent by satellite i.
- Measured distance to satellite i: $c \times \Delta_i$ (c is speed of light)
- Real distance is

$$d_i = c\Delta_i - c\Delta_r = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$$

4 satellites → 4 equations in 4 unknowns (with Δ_r as one of them)

Positioning, Navigation and Timing (PNT)

Basis for global systems

Put satellites in space

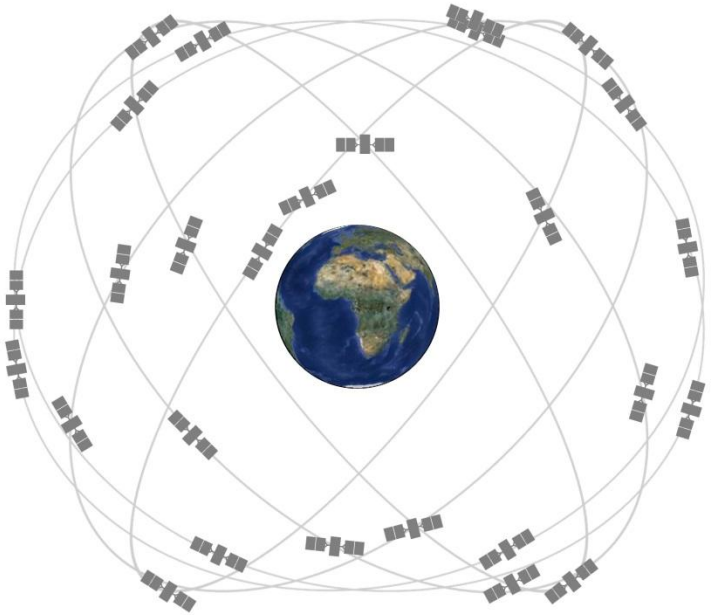
Put clocks on the satellites

Satellites broadcast their time and location to users

Photo by Fallon Michael on Unsplash

US Global Positioning System

US Space Force Mission Delta 31 manages



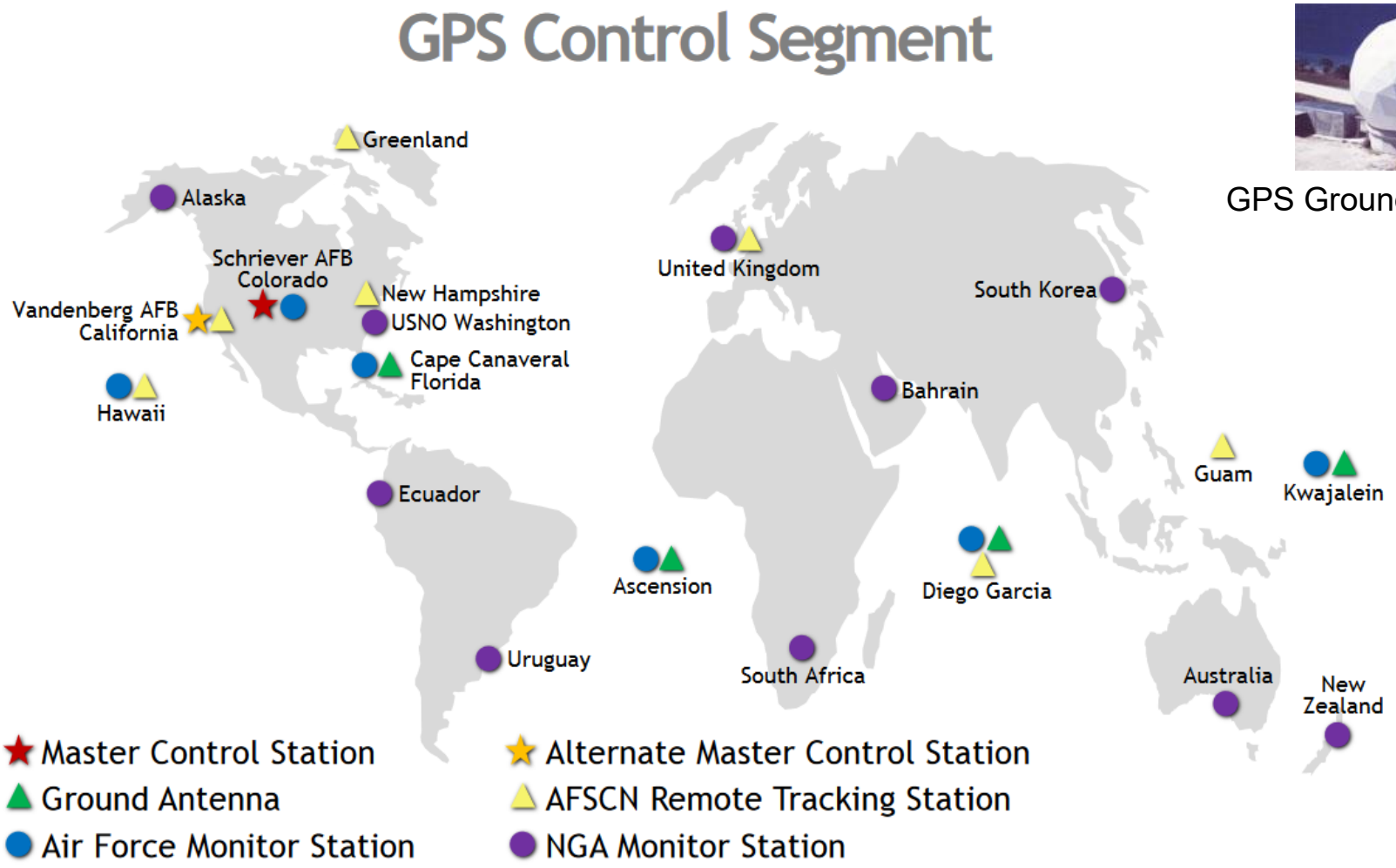
Source: gps.gov

Boeing GPS-12 Satellite (thermal test)
San Diego Air and Space Museum

- Satellites orbit at about 20,200 km (12,550 miles).
- Each satellite circles the Earth twice a day.
- Satellites kept in a 27-slot “expandable constellation”.
  - Until 2011, was 24 slots
- To ensure 95% uptime, there are 31 actual satellites
  - https://www.navcen.uscg.gov/gps-constellation

GPS Management?

Poster: https://www.gps.gov/multimedia/poster/poster-web.pdf



GPS Ground Antenna

Source: gps.gov

GPS' Competitors: GLONASS

- Russian System
- Global coverage
- 26 Satellites
  - 24 operational
- Status
  - https://glonass-iac.ru/en/sostavOG/

Image © **Roscosmos**

GPS' Competitors: BeiDou

- Chinese System
- Global coverage as of 2020
- May 2015: Joint agreement with Russia to enable GLONASS use too
- 35 satellites
  - 3 Generations
  - 59 total launches
- More accurate than GPS over China
  - Military accuracy vs. civilian

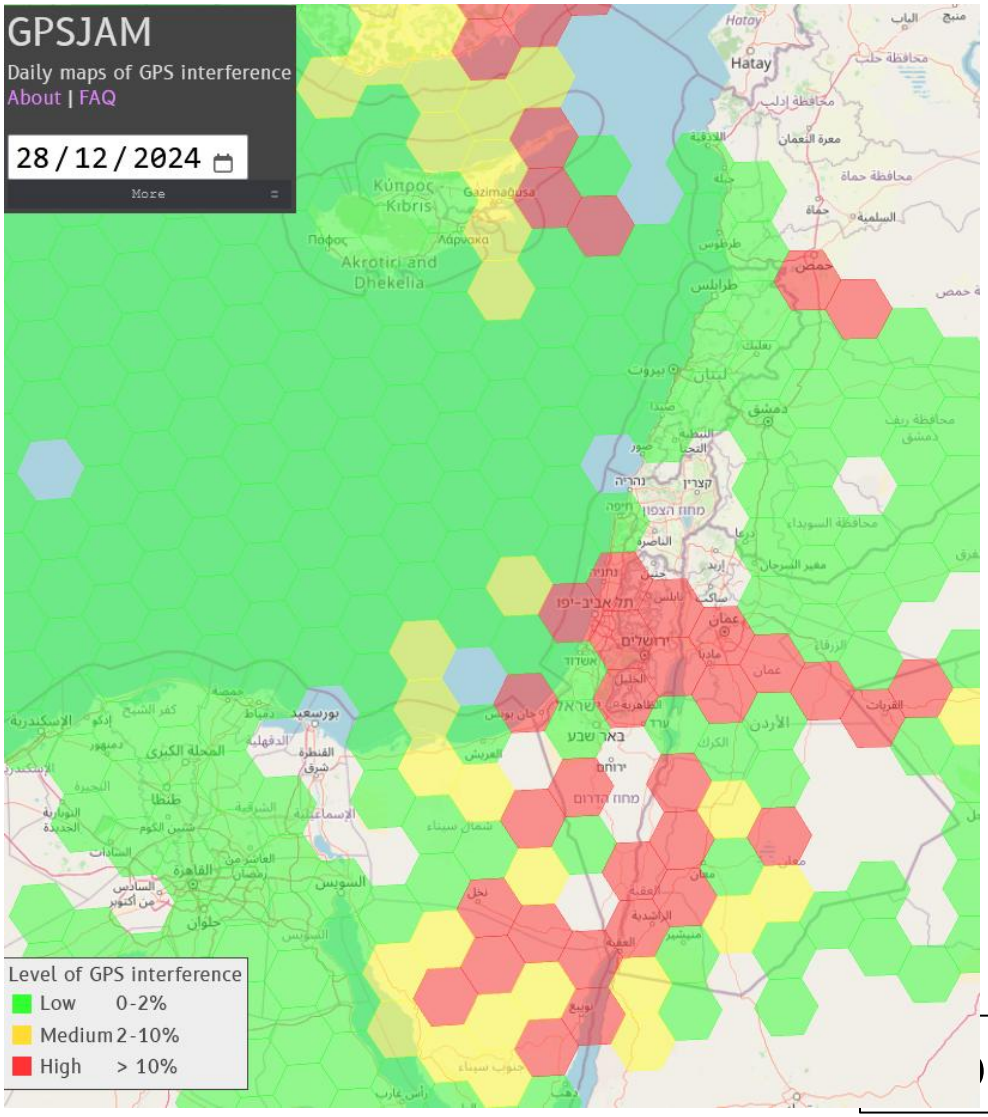
GPS' Competitors: Galileo

- EU System
- 32 Satellites total
  - 25 in use
  - 7 not usable
- Global coverage
- More accurate than GPS
- No encryption for military

GPS Satellite Locations

https://in-the-sky.org/satmap_worldmap.php

GPS Jamming / Spoofing



https://gpsjam.org/

So Far

- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast

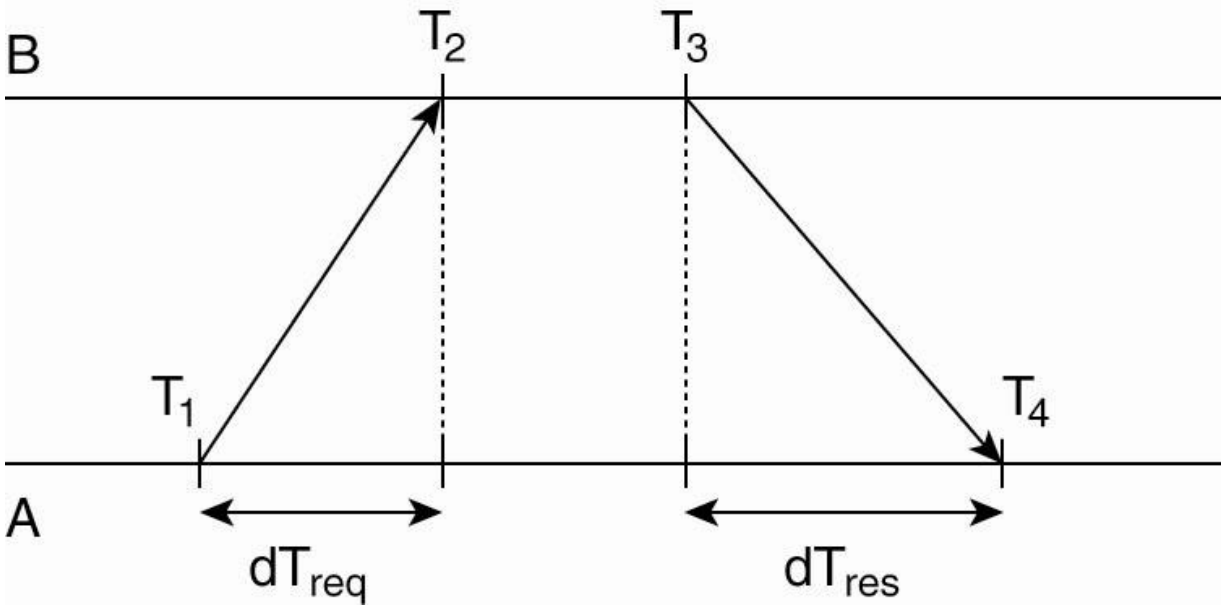
Clock Synchronization Methods

Method I: Every machine asks a **time server** for the accurate time at least once every $\frac{\delta}{2\rho}$ seconds (**Network Time Protocol**).

Okay, but you need an accurate measure of round trip delay, including interrupt handling and processing incoming messages.

Fundamental: You'll have to take into account that setting the time back is **never** allowed → smooth adjustments.

NTP Illustrated



$$\Theta = T_3 + \frac{(T_2 - T_1) + (T_4 - T_3)}{2} - T_4$$

$$= \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

NTP: Ask up, not down

- NTP Stratum – how far away are you from the time source

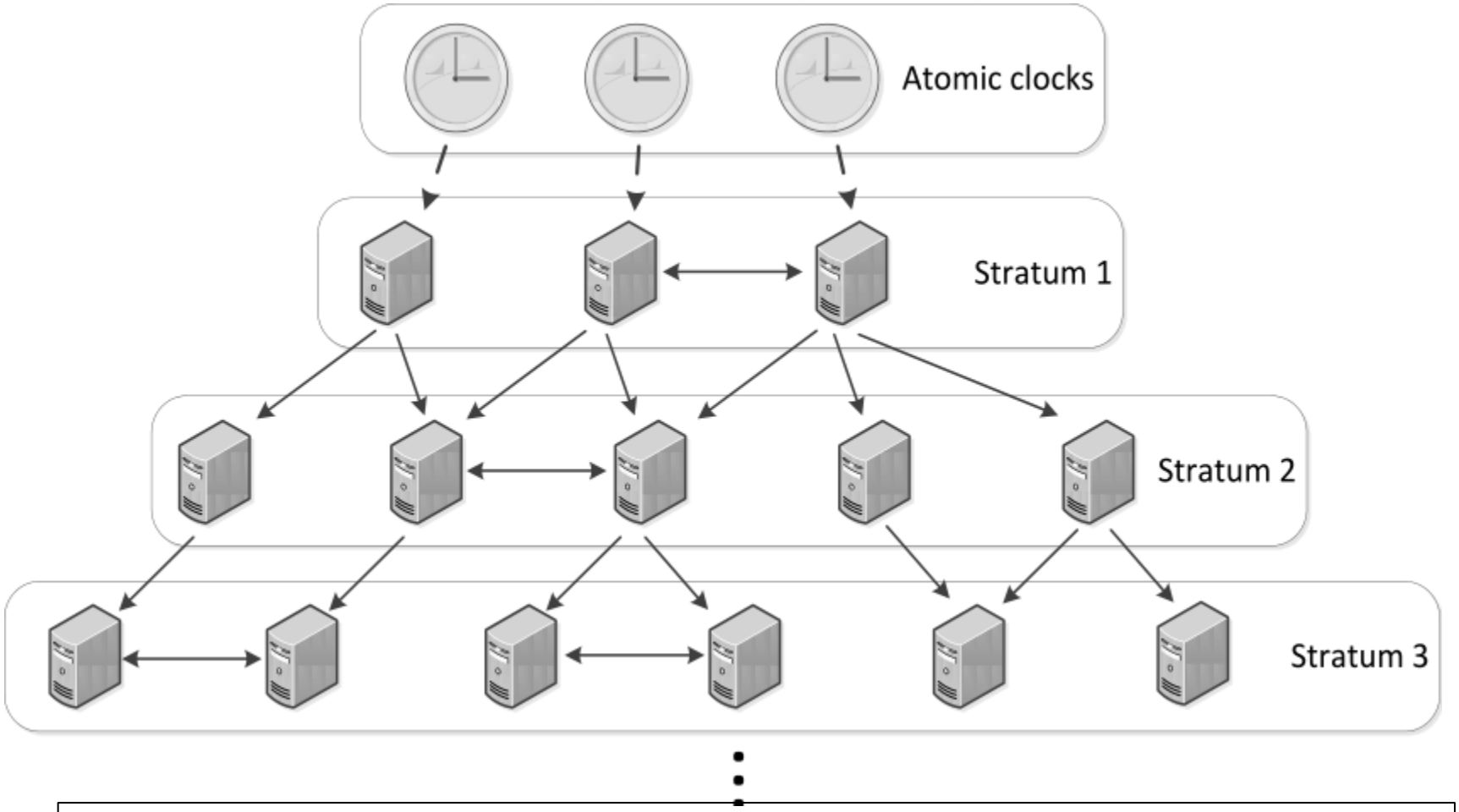


Image source: Bajer, Marcin. (2013). Synchronization of current and voltage measurements in a modular motor diagnostic system. Pomiary Automatyka Kontrola. 1080. https://www.researchgate.net/publication/259267101_Synchronization_of_current_and_voltage_measurements_in_a_modular_motor_diagnostic_system

https://www.isoc.org.il/technologies-and-infrastructure-services/iix/ntp-time-service

שעון ייחוס מדוייק- NTP

איגוד האינטרנט מעמיד לרשות החברות המחוברות למחלף האינטרנט הישראלי (IIX – Israeli Internet eXchange), שעון ייחוס מדויק בפרוטוקול NTP, ומטרתו לאפשר סנכרון של הנתבים ושעוני המערכת השונים (Network Time Protocol).

השירות מבוסס על ארבעה שעוני GPS, מתוכם שניים הממוקמים באיגוד האינטרנט, ושני שעונים נוספים הממוקמים באוניברסיטה העברית בירושלים.

מידע טכני לצורך התחברות לשעון הייחוס

התחברות לשרות השעון נגישה דרך timeserver.iix.net.il באמצעות NTP, והינה stratum-2.

לרוב מערכות ההפעלה המודרניות יש תמיכה מובנית בסנכרון מול שעון ייחוס מדויק באמצעות פרוטוקול NTP,
וברוב ציודי התקשורת, נתבים, מתגים וכדומה יש תמיכה מובנית בפרוטוקול זה לצורך סנכרון השעון.

בנוסף, ניתן להגדיר נתבי Cisco (וציוד אחר) לסנכרן את השעון שלהם באמצעות שירות ה-NTP ללא צורך בתוכנה נוספת. קיימים כלי NTP גם למערכות הפעלה אחרות.

לתשומת לב: מכיוון שהשירות מבוסס על מקורות זמן GPS, אין באפשרותנו להתחייב לפעולה תקינה של השעונים או לזמינותם. יחד עם זאת, ניתן לומר על סמך ניסיון השנים האחרונות שלא התרחשו תקלות משמעותיות במערכות stratum-1 עליה מבוססת המערכת, ועד כה השירות היה אמין ואיכותי.

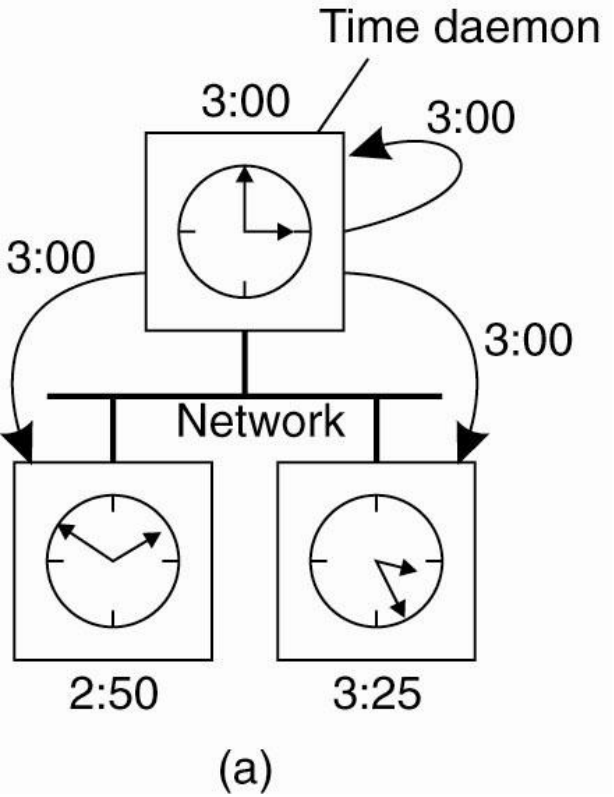
תודה: למעבדה הלאומית לפיזיקה באוניברסיטה העברית בירושלים על ארוח השירות בתחילת דרכו, באמצעות השעון האטומי במעבדה.

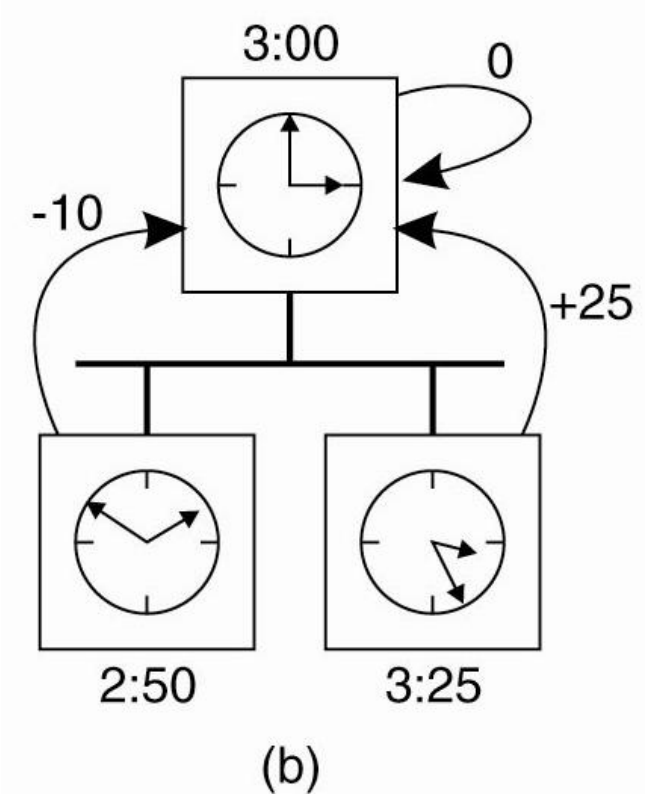
Clock Synchronization Methods

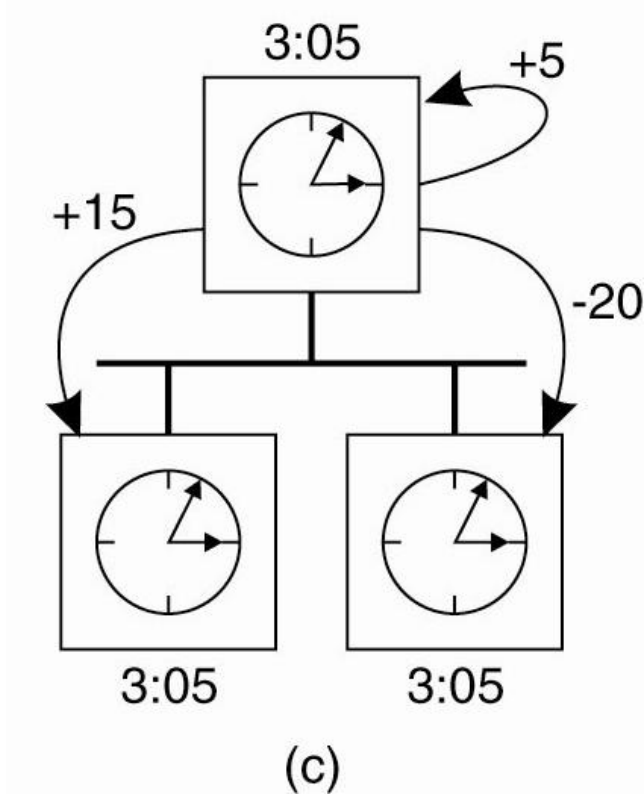
Method II: Let the time server scan all machines periodically, calculate an average, and inform each machine how it should adjust its time relative to its present time. (**Berkeley algorithm**)

Okay, you'll probably get every machine in sync. **Note**: you don't even need to propagate UTC time.

Berkeley Algorithm Illustrated







So Far

- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast

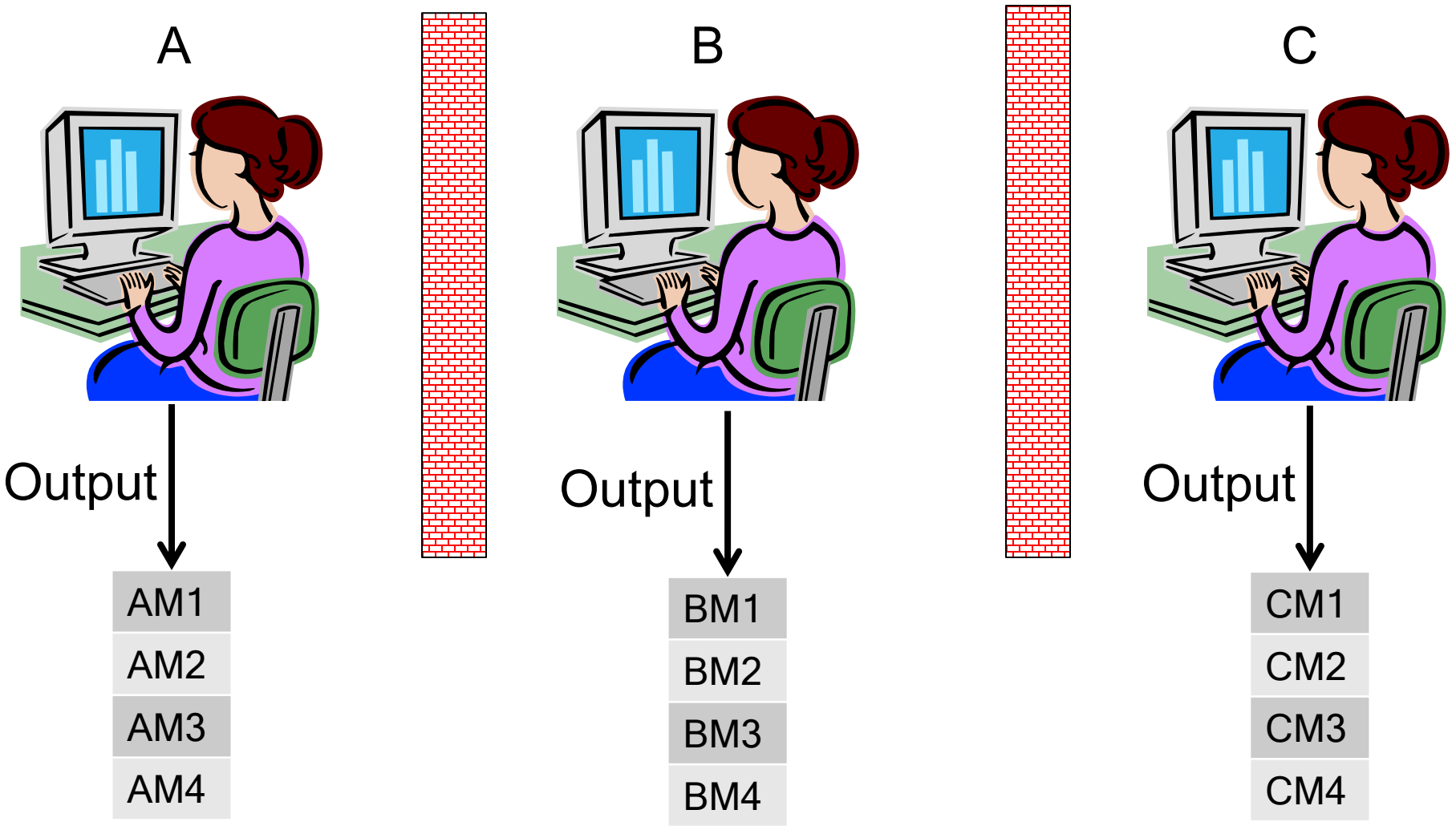
The Happened-Before Relationship

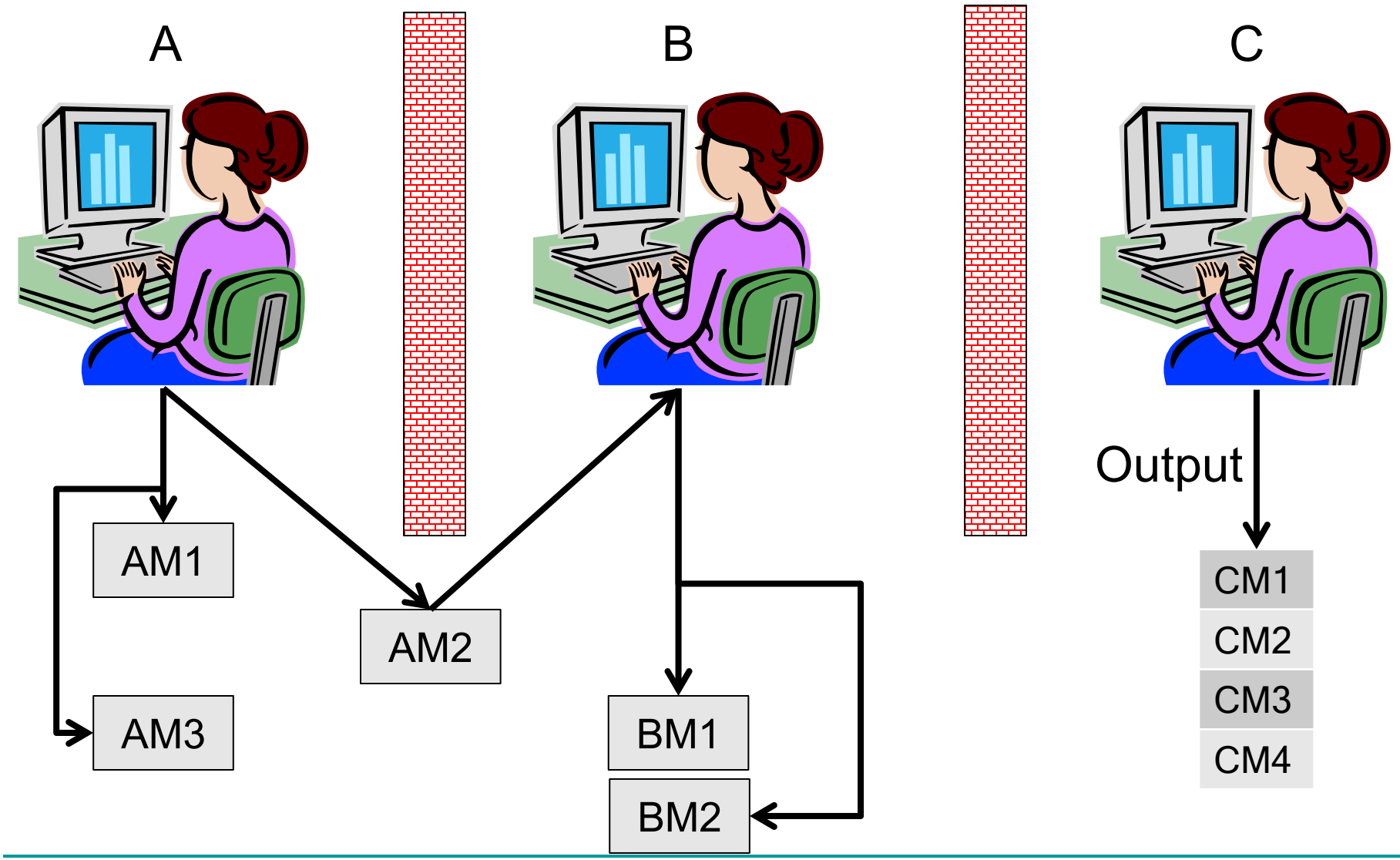
Problem: We first need to introduce a notion of order in before we can order anything.

The **happened-before** relation on the set of events in a distributed system:

- If a and b are two events in the same process, and a comes before b, then $a \rightarrow b$.
- If a is the sending of a message, and b is the receipt of that message, then $a \rightarrow b$
- If $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$

Note: this introduces a **partial ordering of events** in a system with concurrently operating processes.


A
B
C
Output
Output
Output
AM1
AM2
AM3
AM4
BM1
BM2
BM3
BM4
CM1
CM2
CM3
CM4


A
B
C
AM1
AM2
AM3
BM1
BM2
Output
CM1
CM2
CM3
CM4

Logical Clocks (1/2)

Problem: How do we maintain a global view on the system's behavior that is consistent with the happened-before relation?

Solution: attach a timestamp $C(e)$ to each event e, satisfying the following properties:

P1: If a and b are two events in the same process, and $a \rightarrow b$, then we demand that $C(a) < C(b)$.

P2: If a corresponds to sending a message m, and b to the receipt of that message, then also $C(a) < C(b)$.

Problem: How to attach a timestamp to an event when there's no global clock → maintain a **consistent** set of logical clocks, one per process

Logical Clocks (2/2)

Solution

Each process P_i maintains a **local** counter C_i and adjusts this counter according to the following rules:

1: For any two successive events that take place within P_i, C_i is incremented by 1.

2: Each time a message m is sent by process P_i, the message receives a timestamp $ts(m) = C_i$.

3: Whenever a message m is received by a process P_j, P_j adjusts its local counter C_j to $\max\{C_j, ts(m)\}$; then executes step 1 before passing m to the application.

Property **P1** is satisfied by (1);

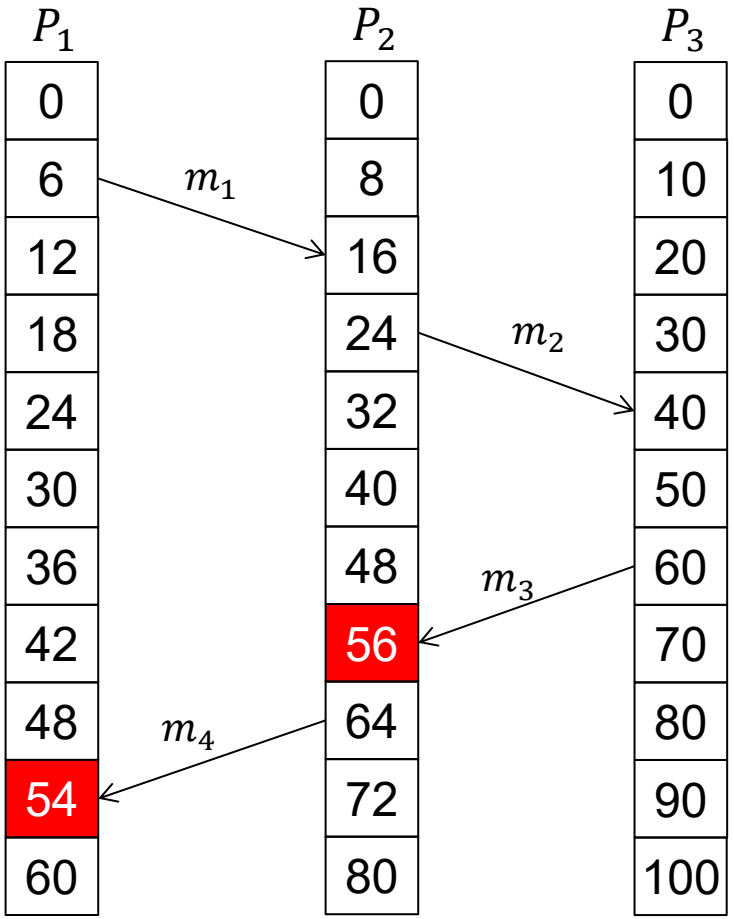
Property **P2** by (2) and (3).

Note

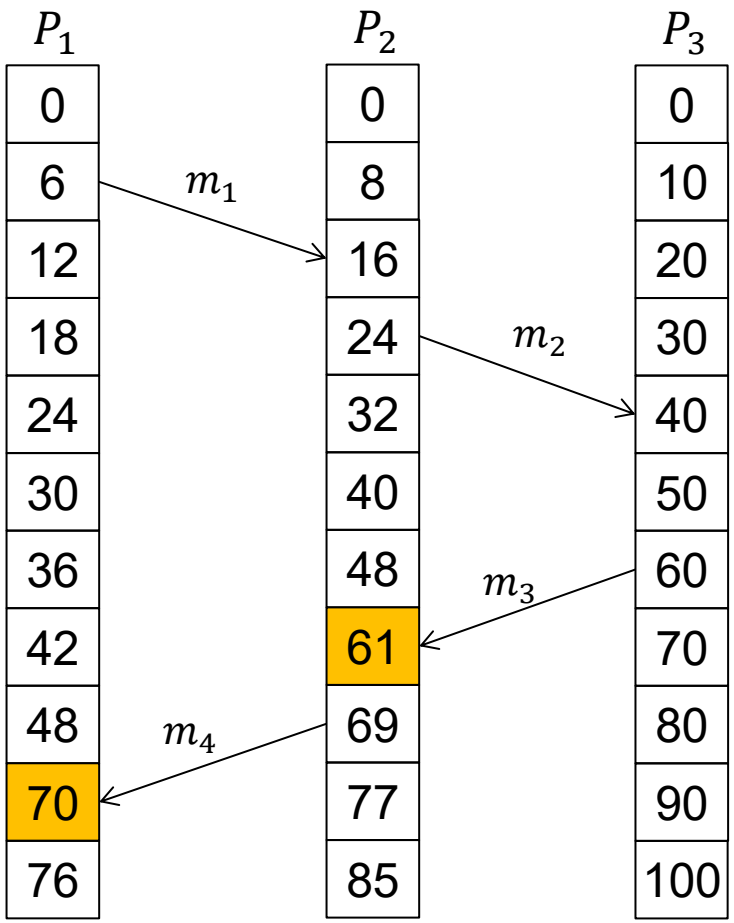
It is still possible for two events to happen at the same time. Avoid this by breaking ties through process IDs.

Logical Clocks - Example

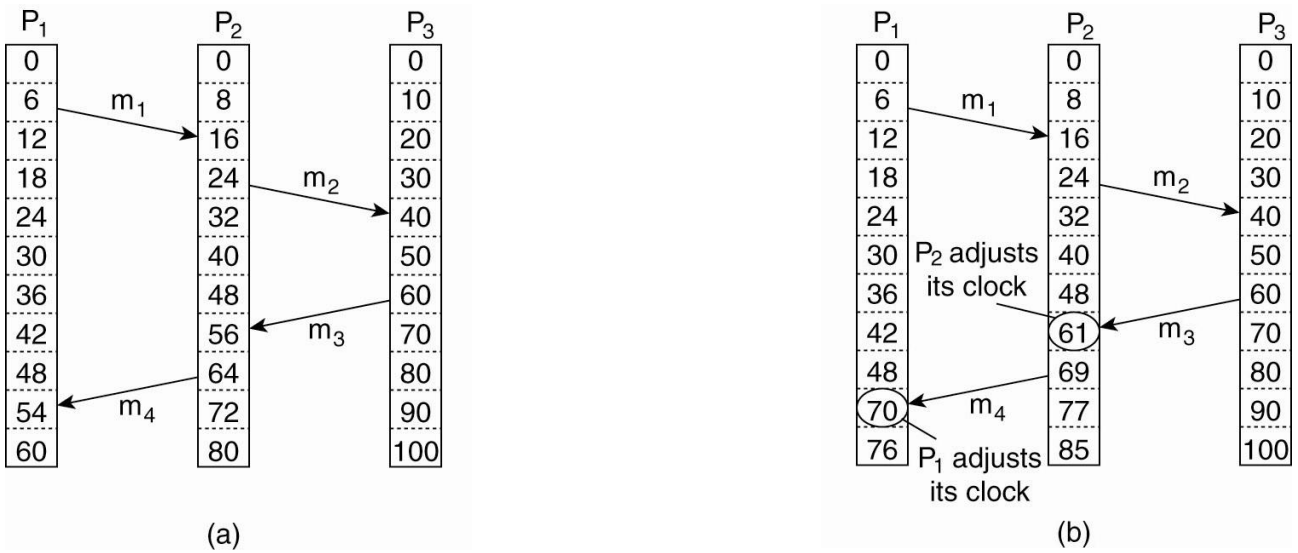
No Clock Adjustment



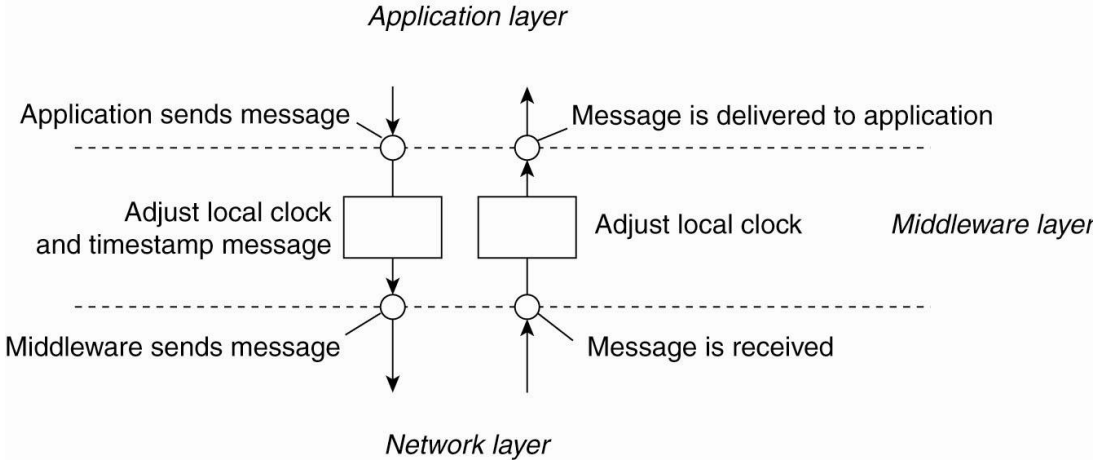
Clock Adjustment



Logical Clocks - Example



Note: Adjustments take place in the middleware layer:



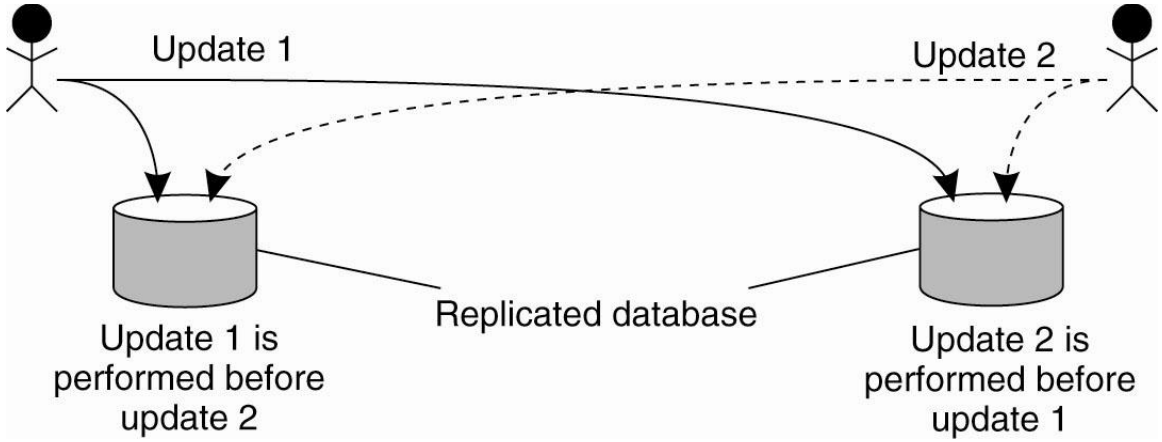
So Far

- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast

Example: Totally Ordered Multicast (1/2)

Problem: We sometimes need to guarantee that concurrent updates on a replicated database are seen in the same order everywhere:

- P_1 adds \$100 to an account (initial value: \$1000)
- P_2 increments account by 1%
- There are two replicas



Result: in absence of proper synchronization:
replica #1 ← \$1111, while replica #2 ← \$1110.

Example: Totally Ordered Multicast (2/2)

Solution:

- Process P_i sends timestamped message msg_i to all others. The message itself is put in a local queue $queue_i$.
- Any incoming message at P_j is queued in $queue_j$, according to its timestamp, and acknowledged to every other process.

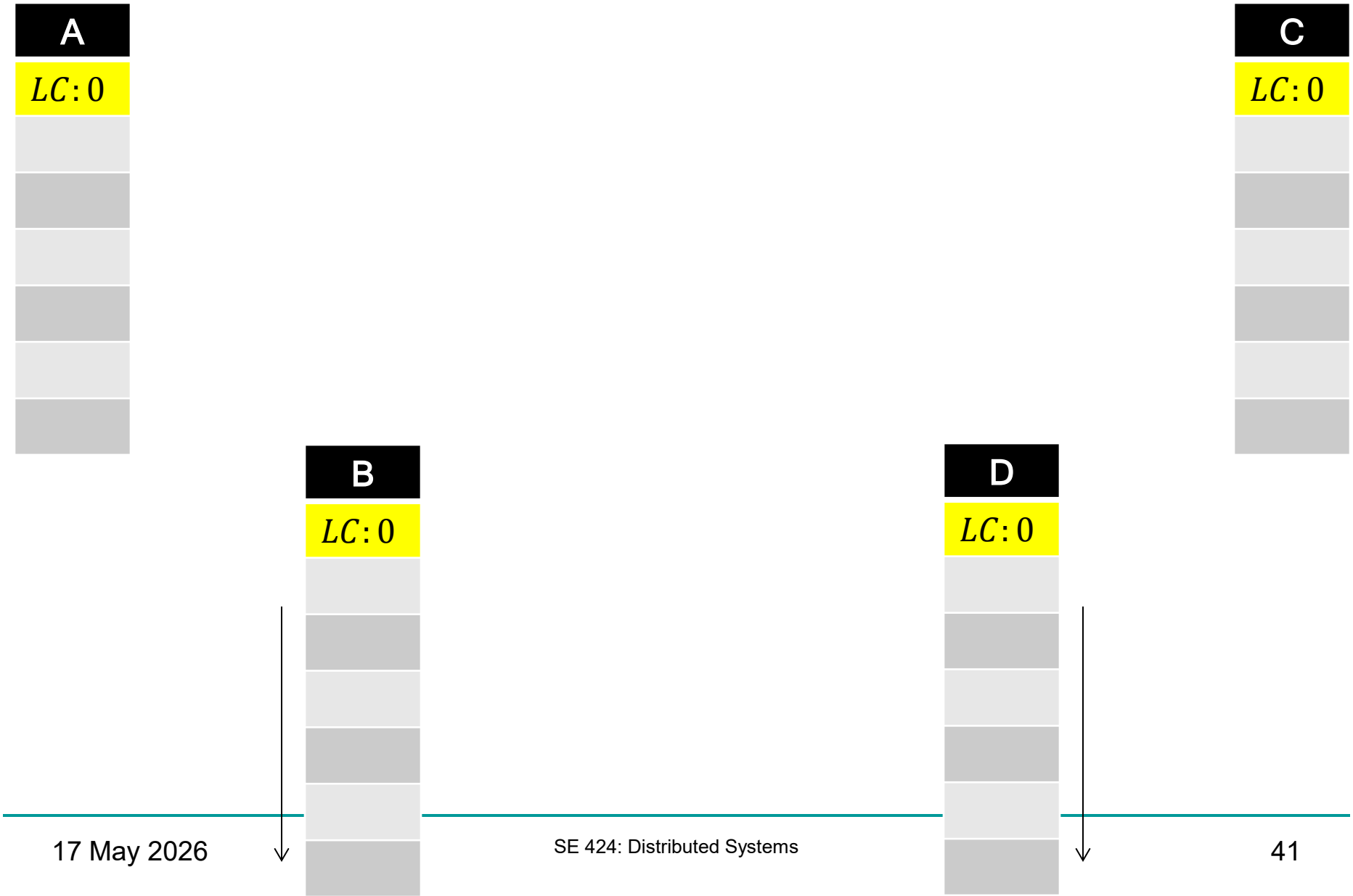
P_j passes a message msg_i to its application if:

(1) msg_i is at the head of $queue_j$

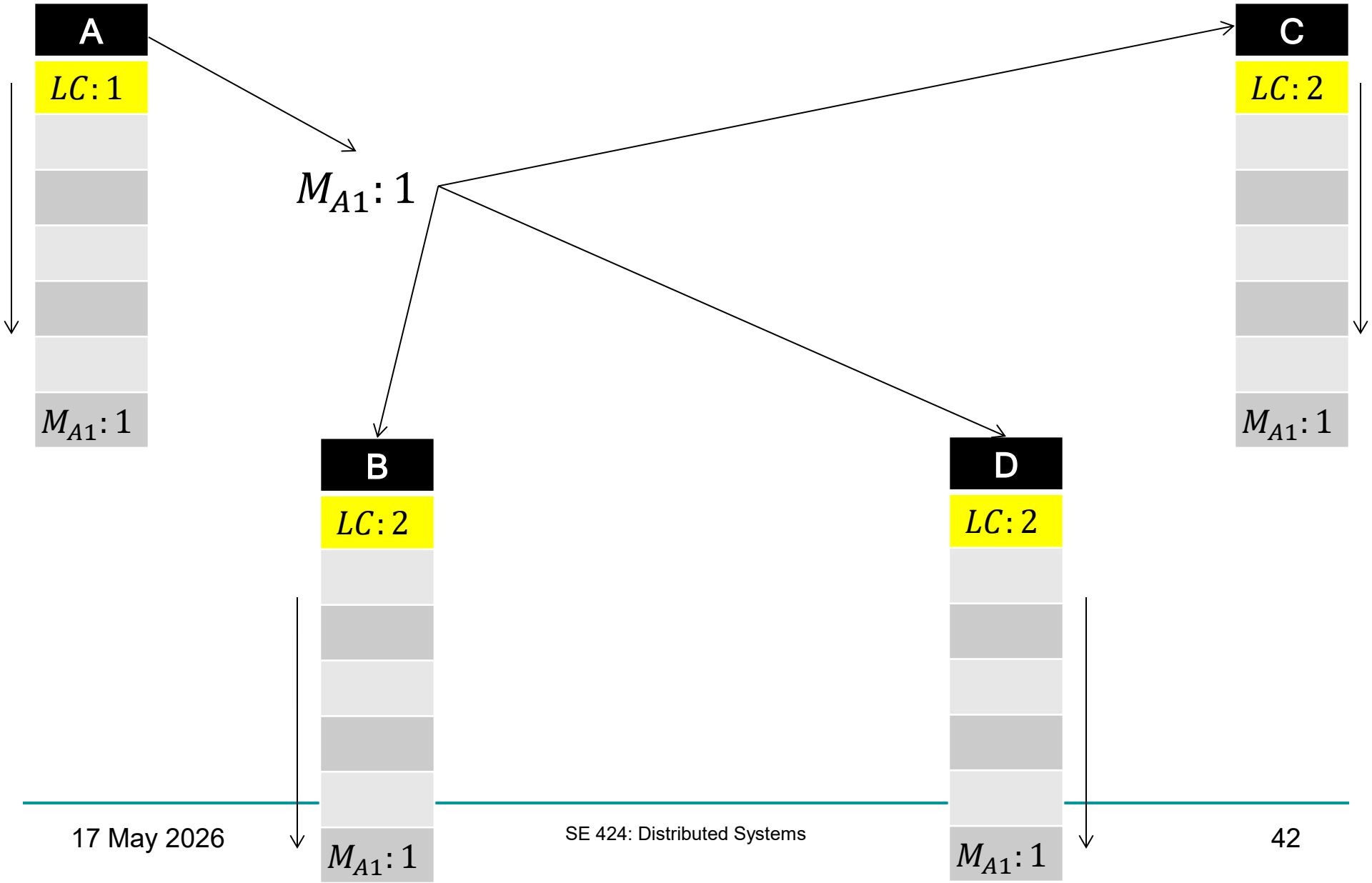
(2) for each process P_k, there is a message msg_k in $queue_j$ with a larger timestamp.

Note: We are assuming that communication is reliable and FIFO ordered (i.e. messages from a single sender arrive in the order sent)

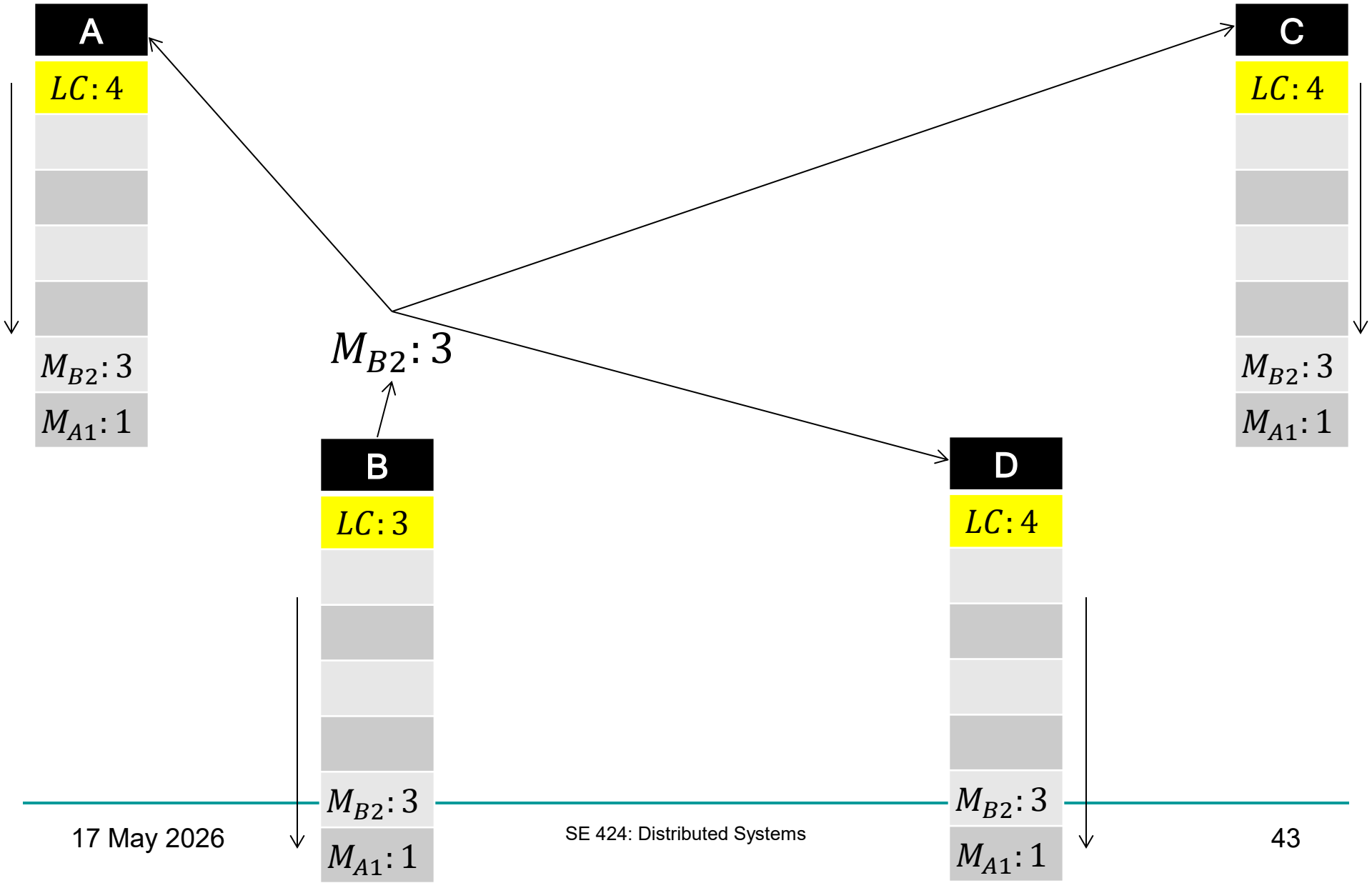
TOM Illustrated 1



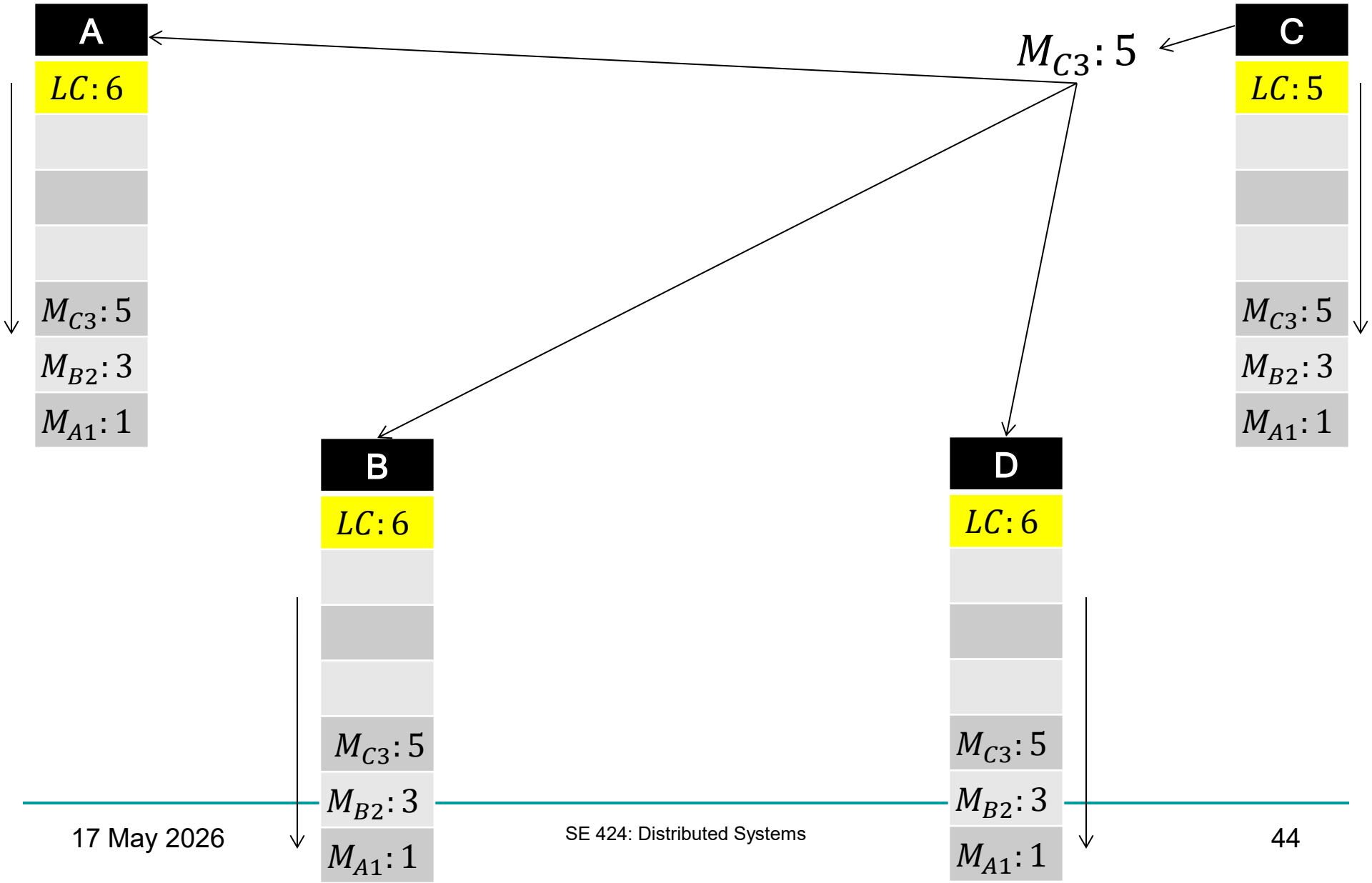
TOM Illustrated 2



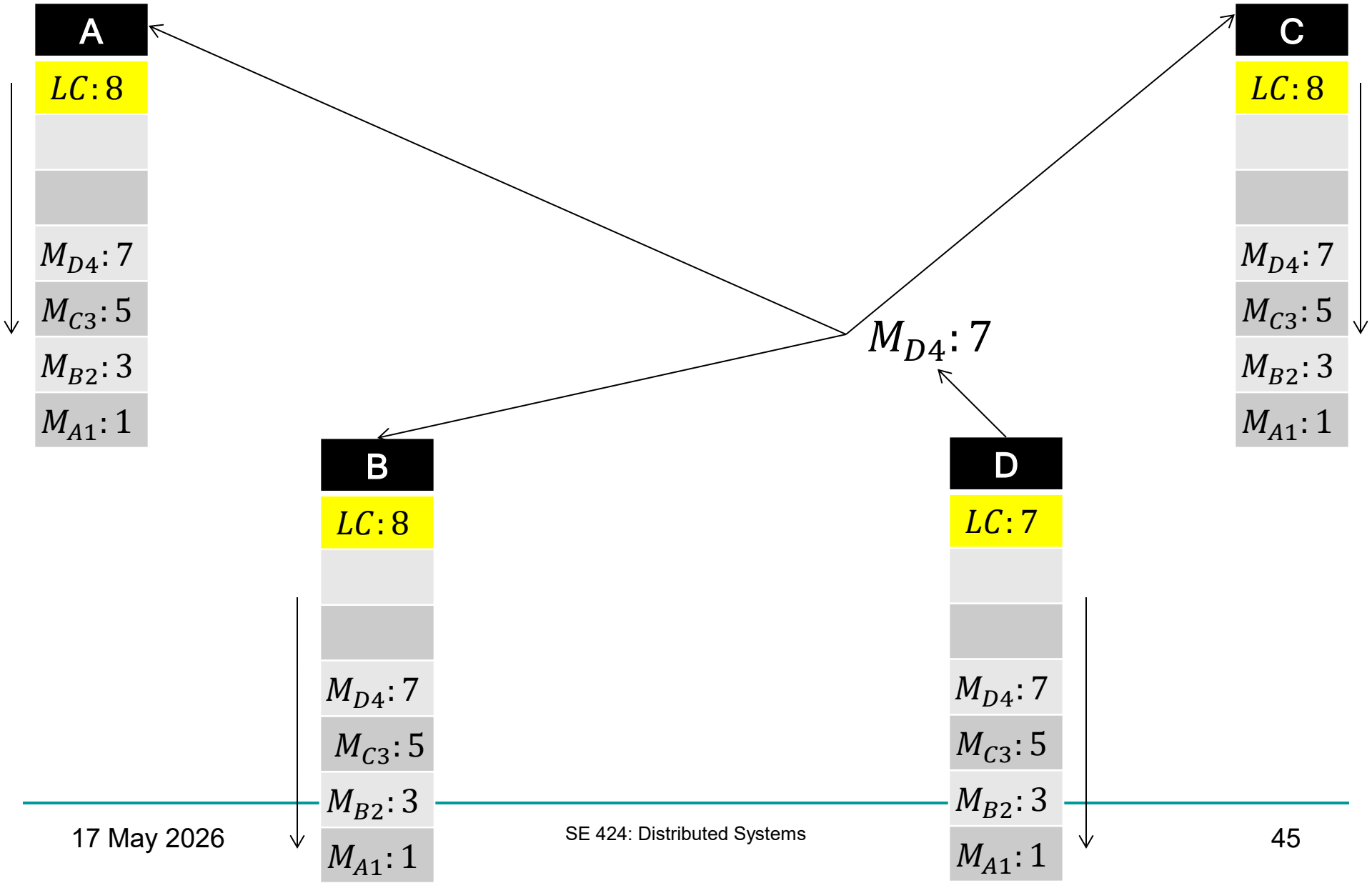
TOM Illustrated 3



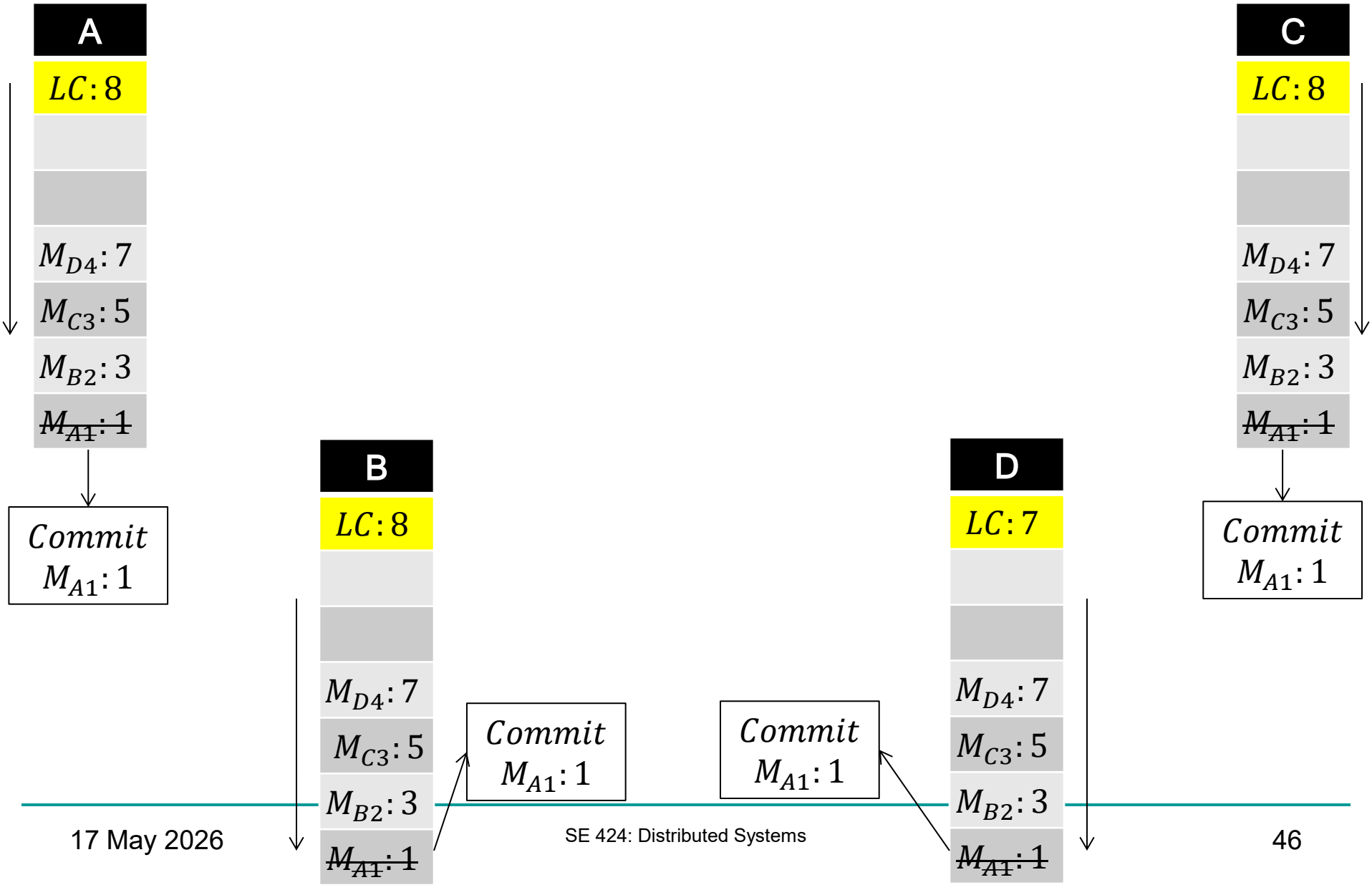
TOM Illustrated 4



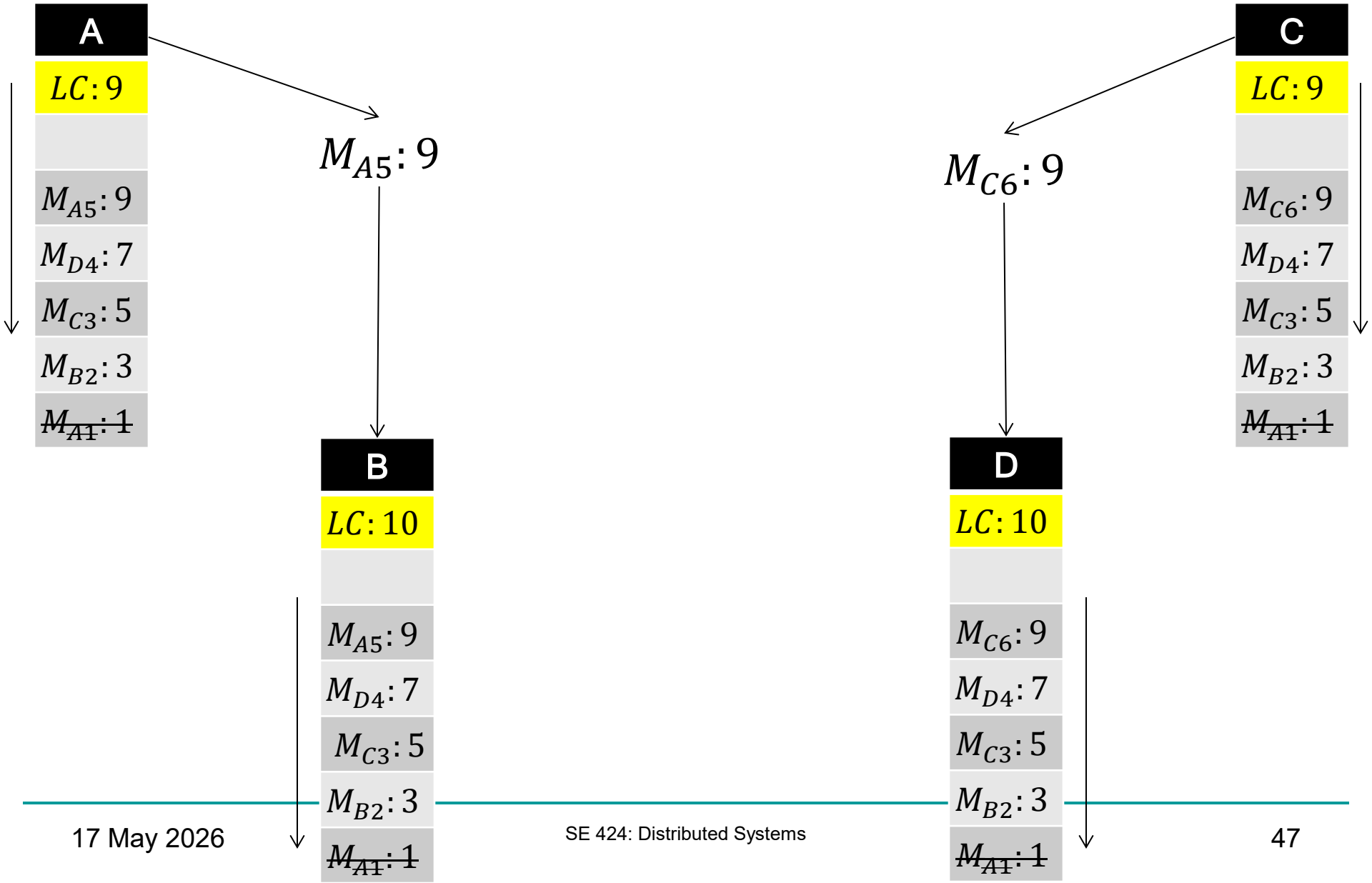
TOM Illustrated 5



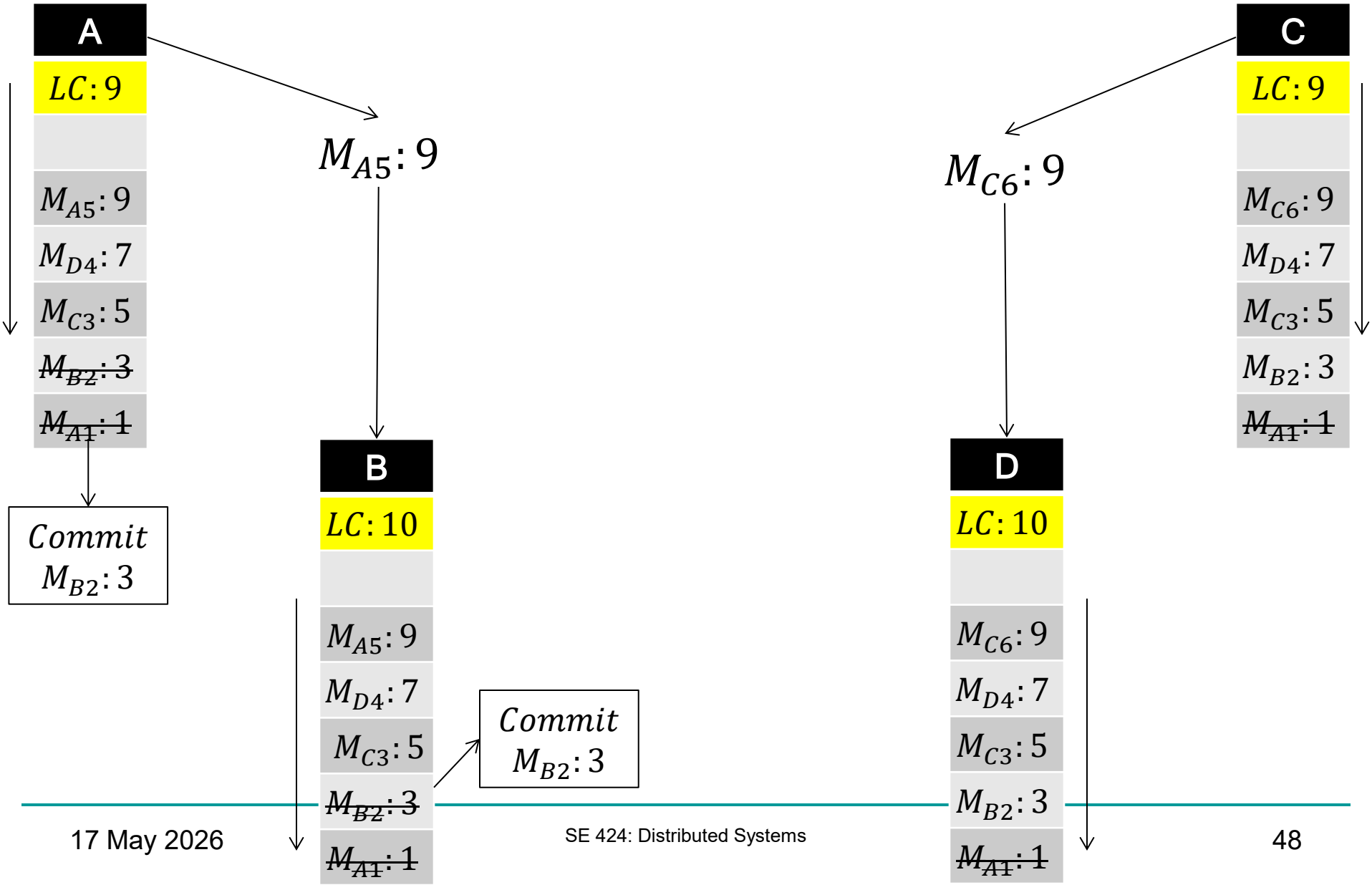
TOM Illustrated 6



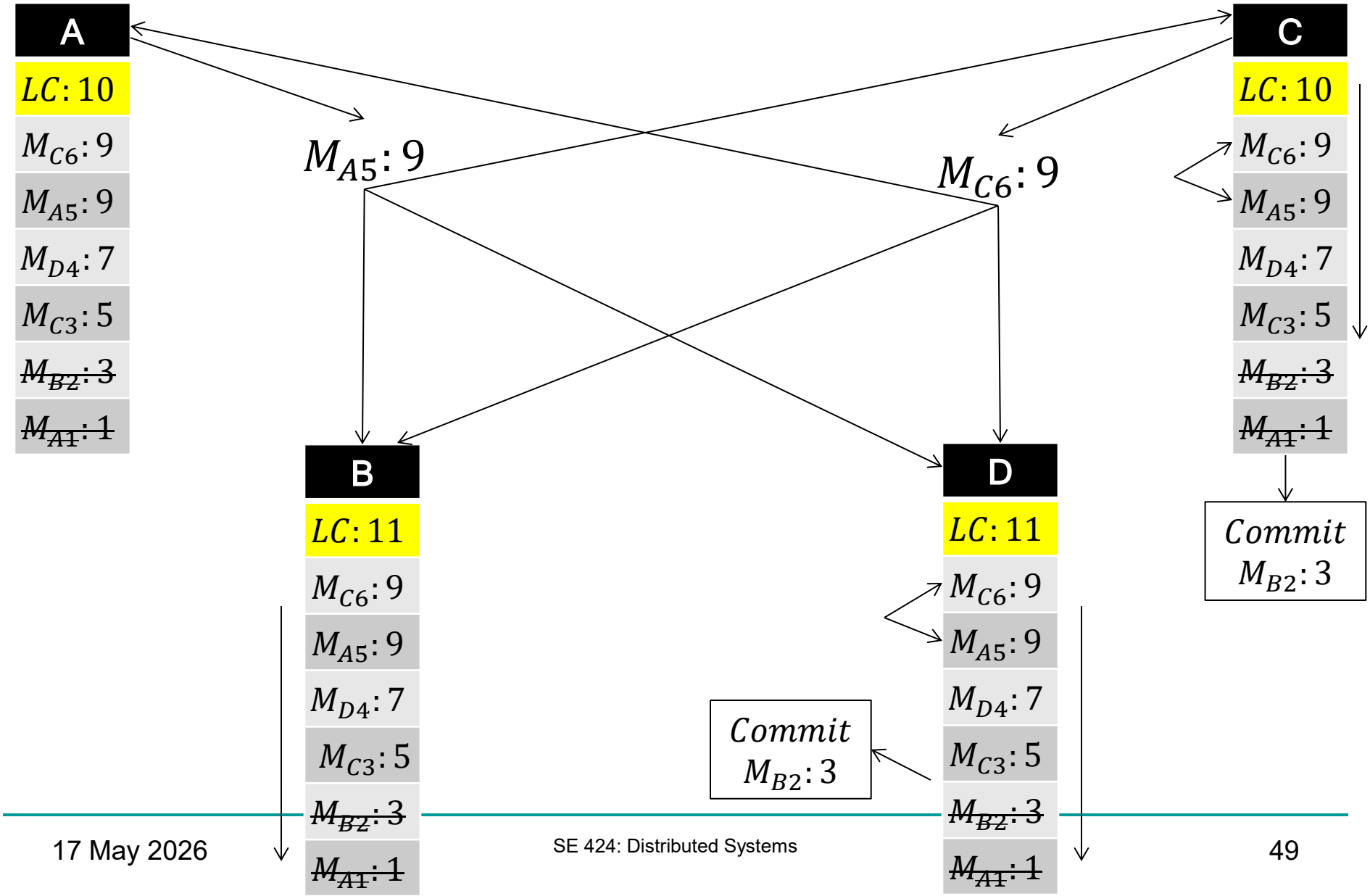
TOM Illustrated 7



TOM Illustrated 8



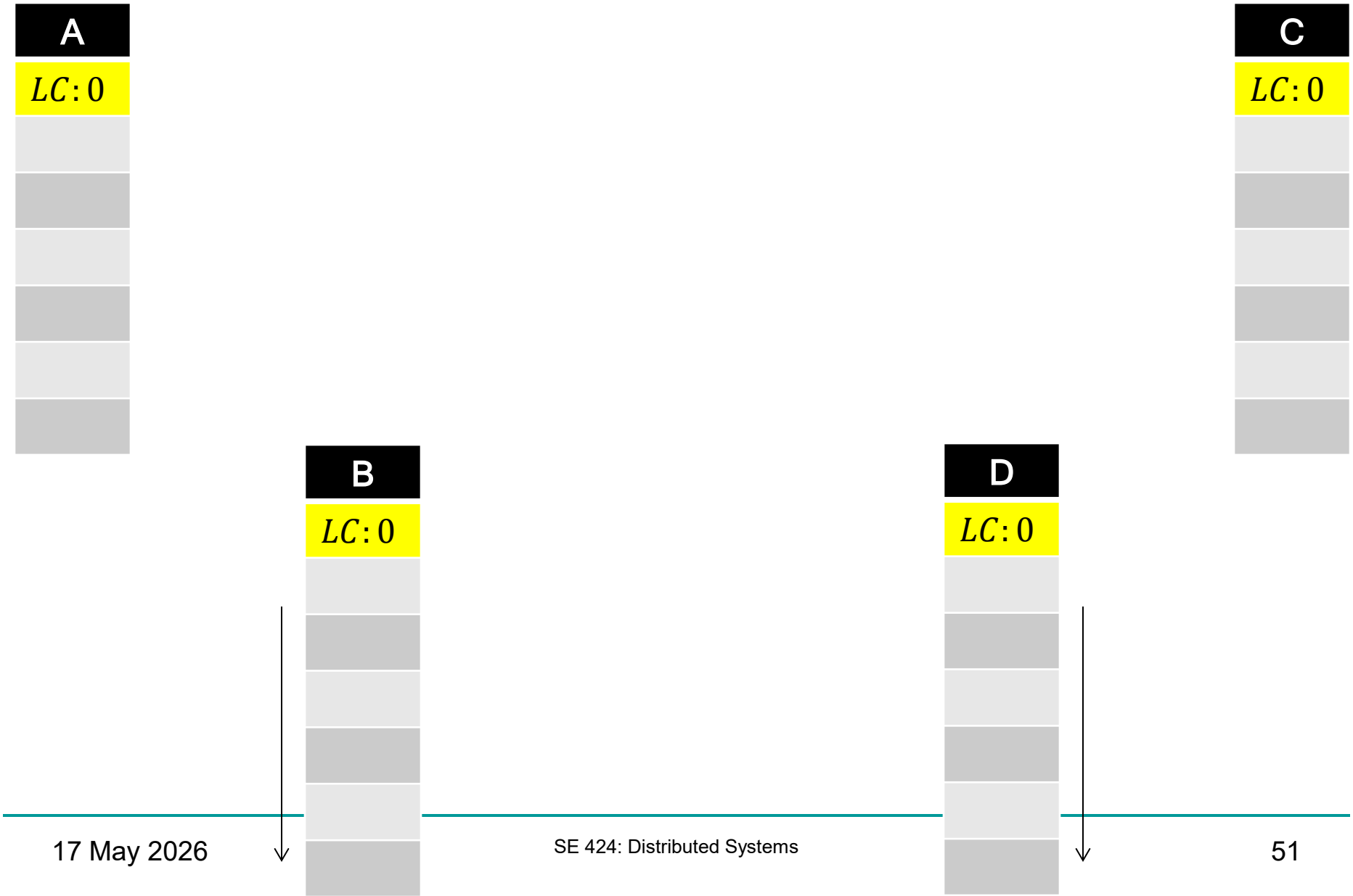
TOM Illustrated 9



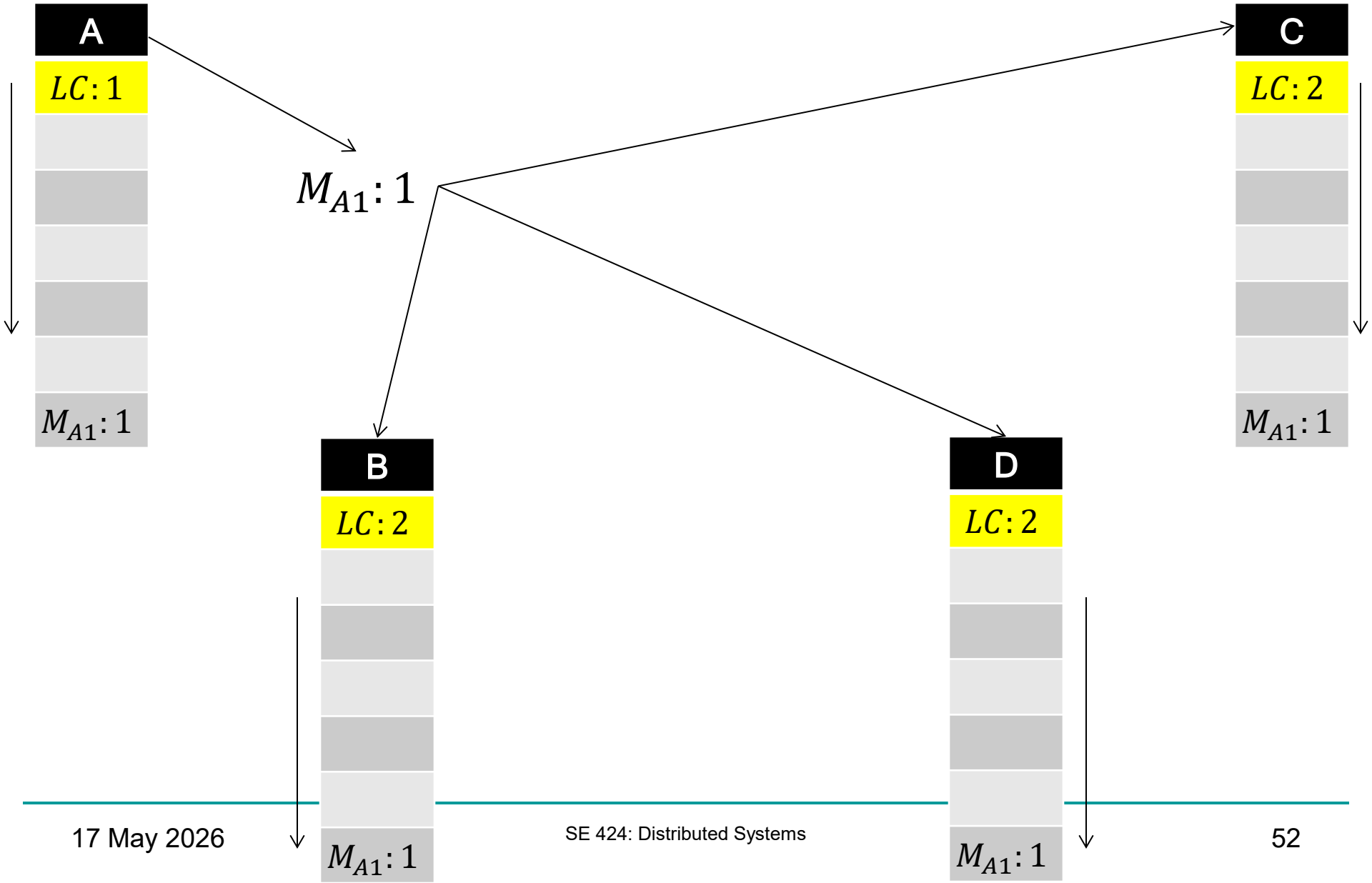
TOM with ACKs

- For systems with slower message sending, we can use ACKs to make TOM work better

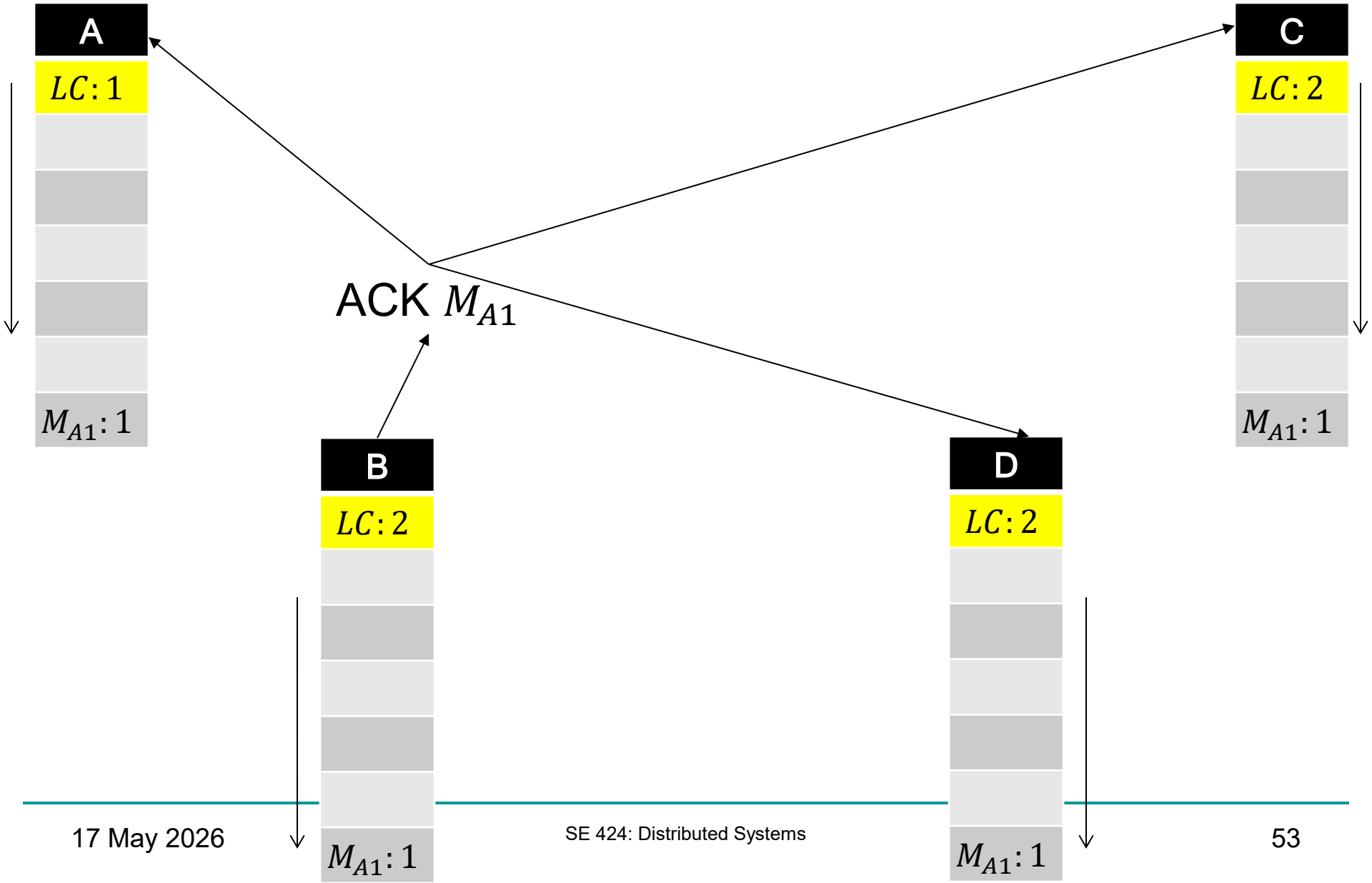
TOM ACKs Illustrated 1



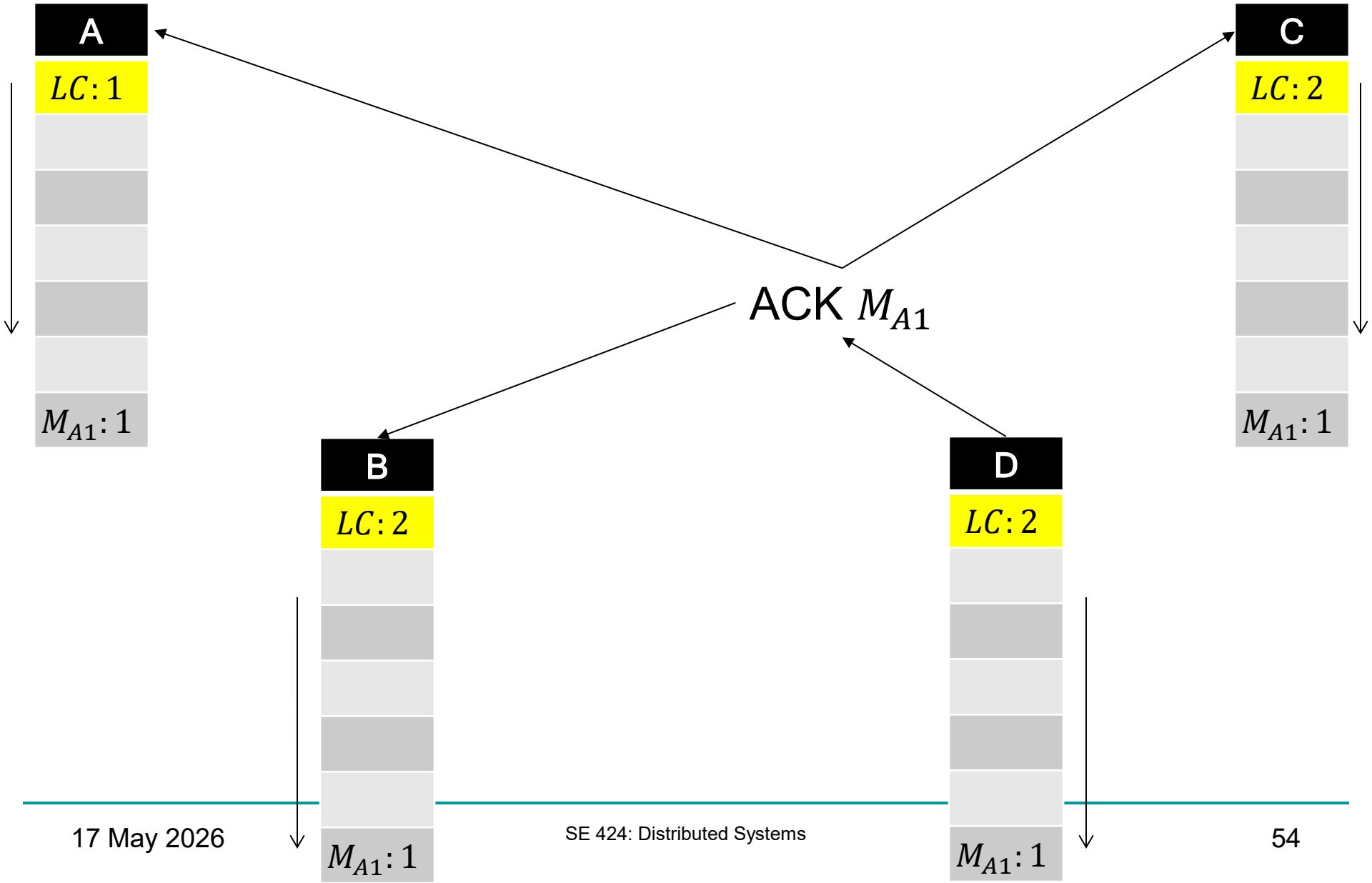
TOM ACKs Illustrated 2



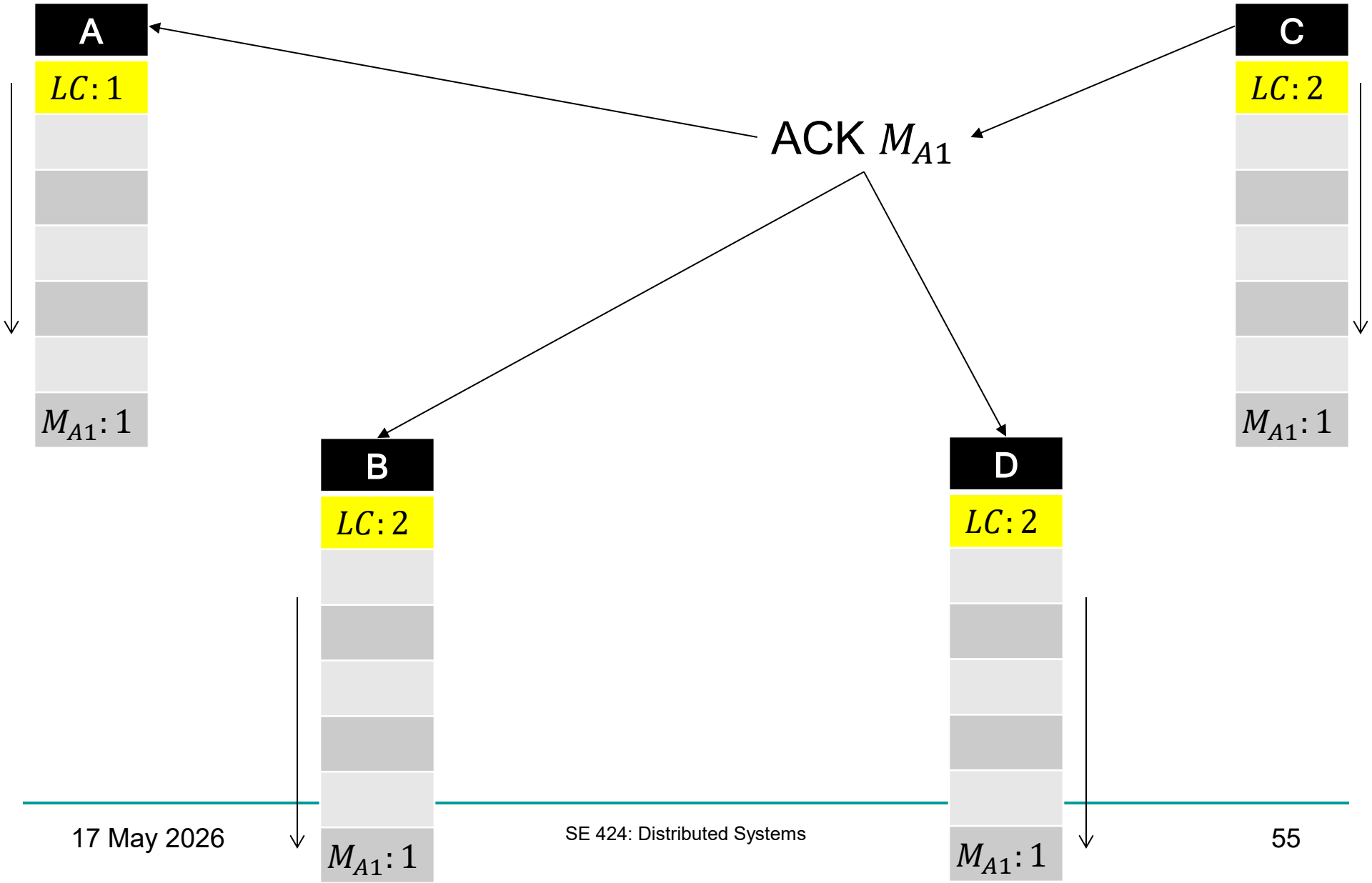
TOM ACKs Illustrated 3



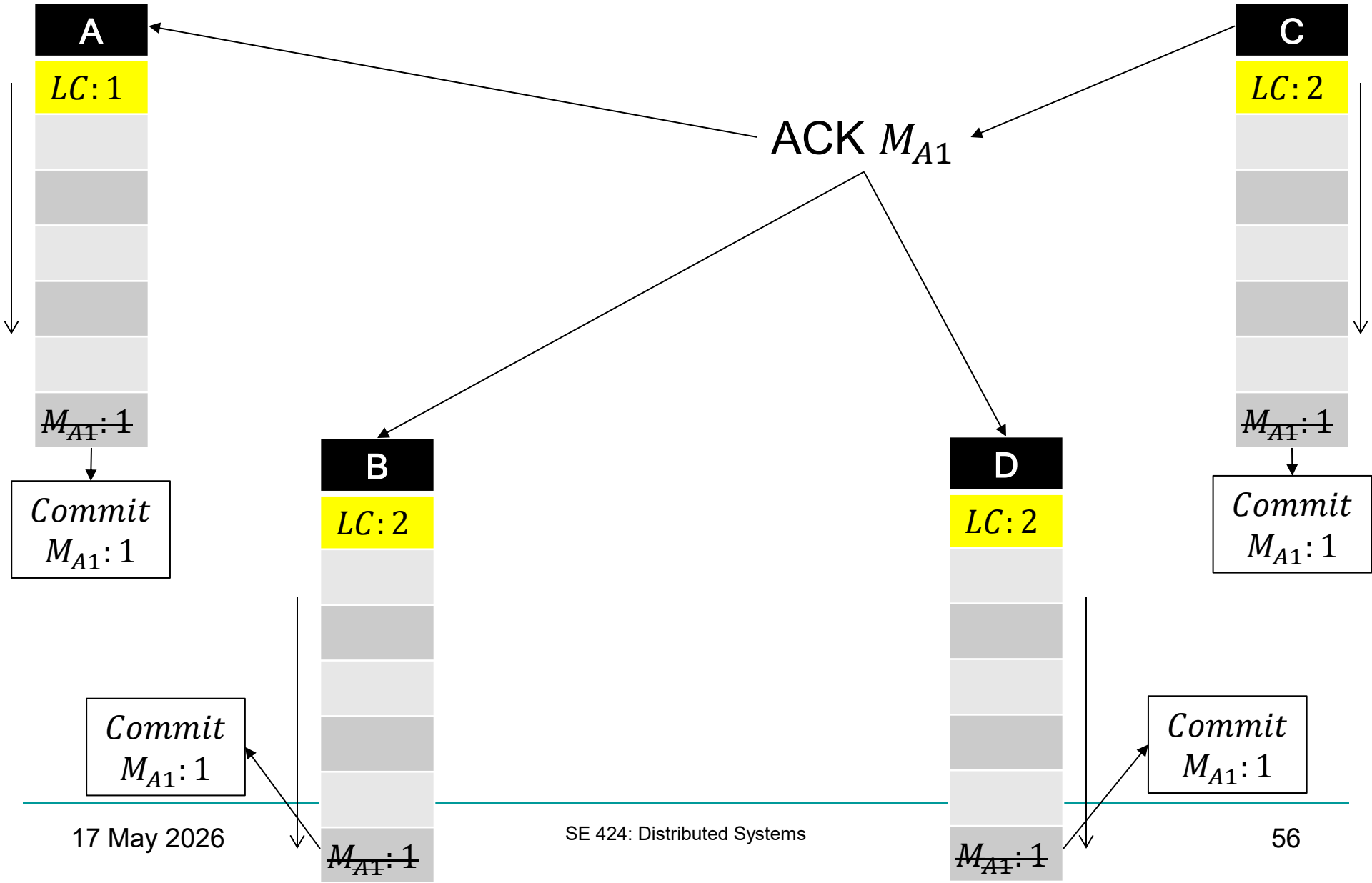
TOM ACKs Illustrated 4



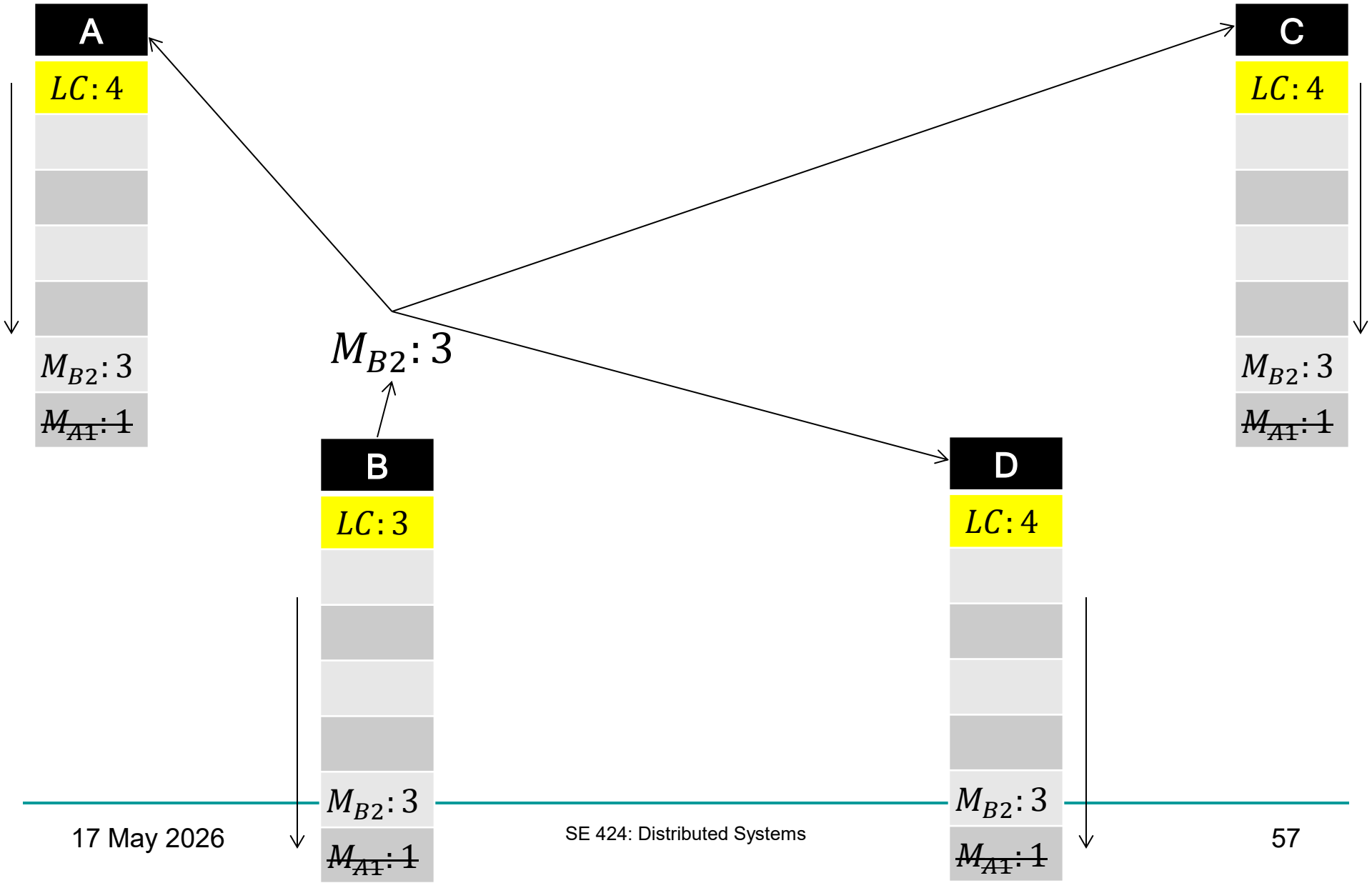
TOM ACKs Illustrated 5



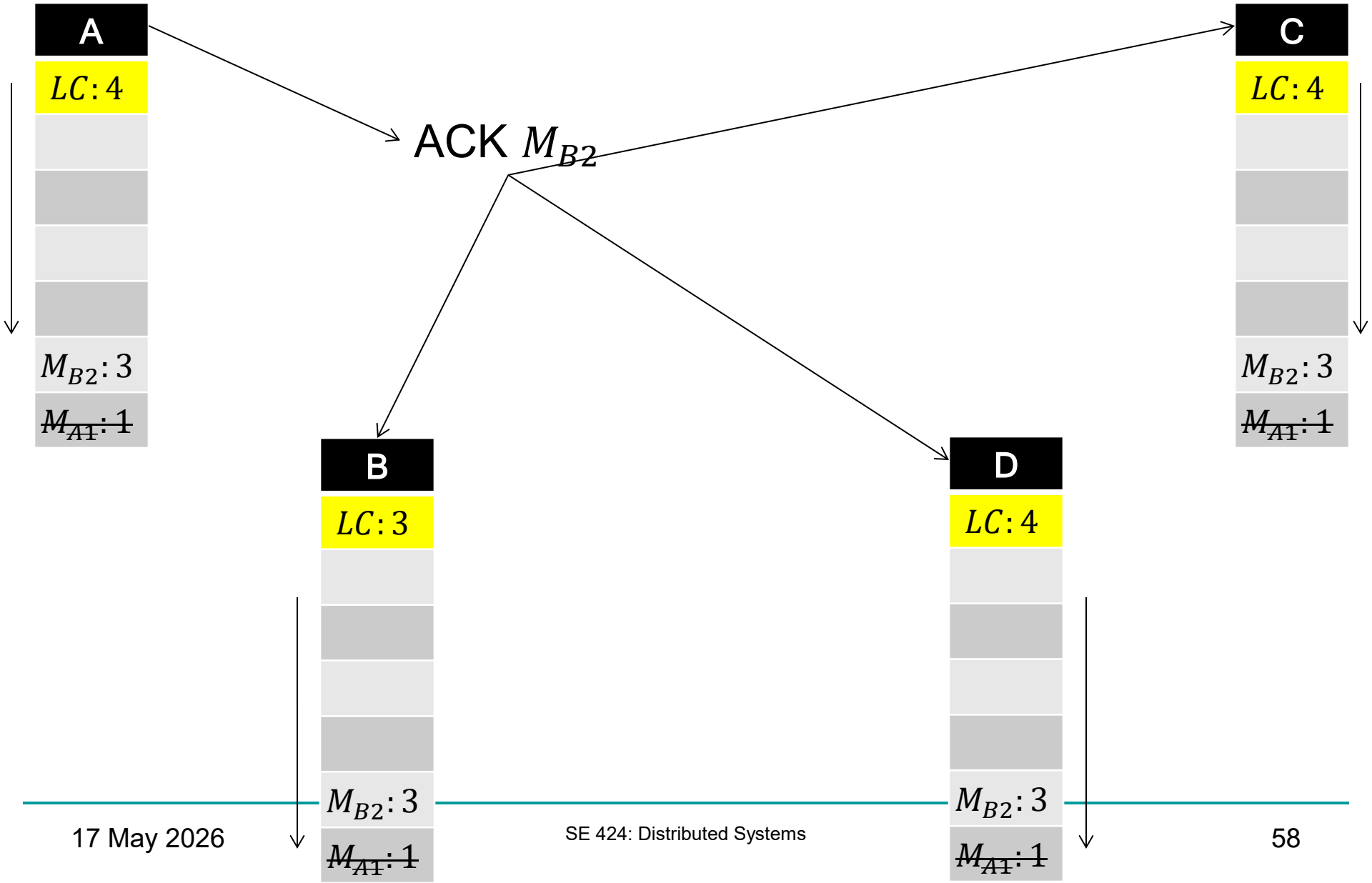
TOM ACKs Illustrated 6



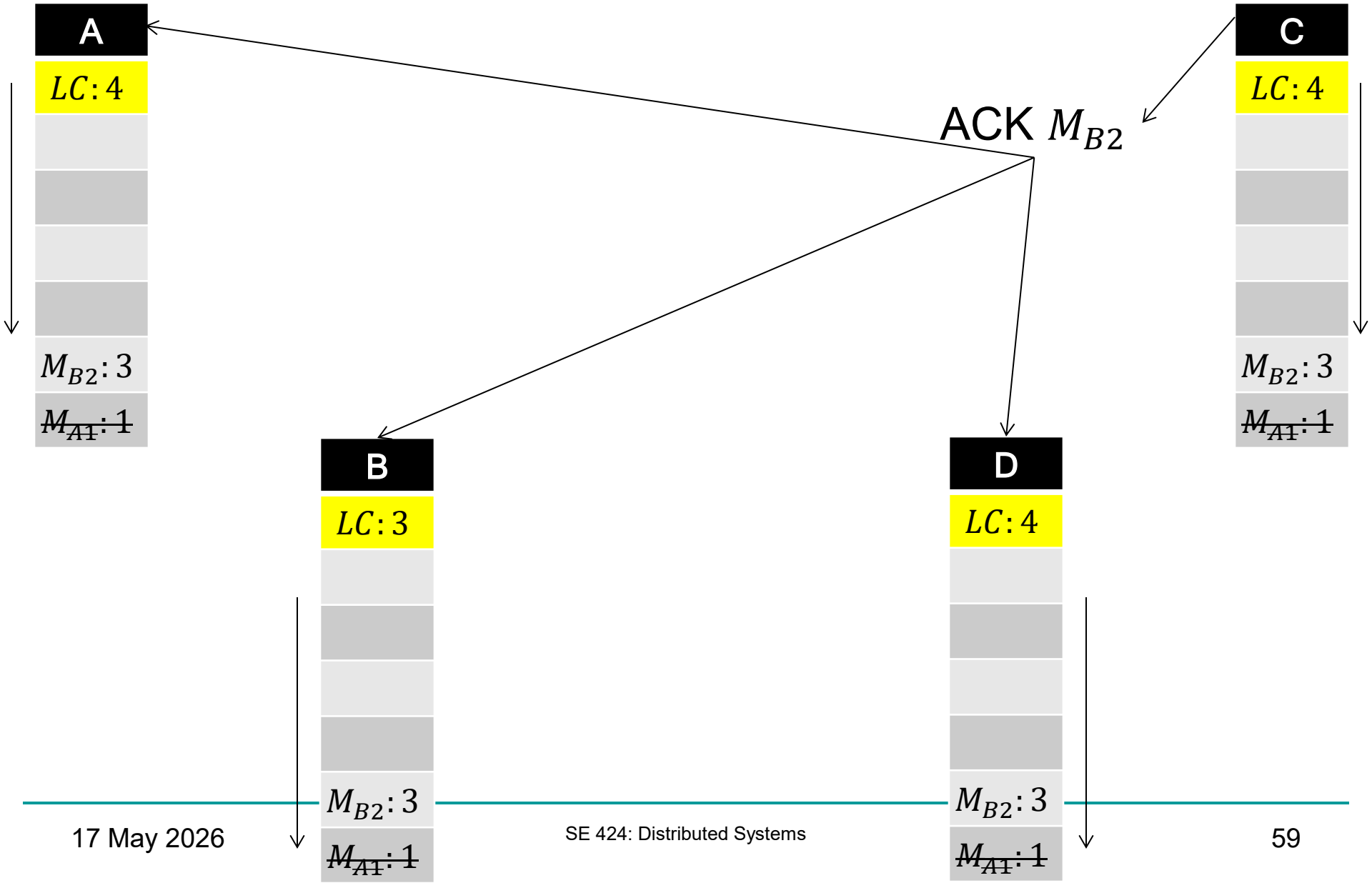
TOM ACKs Illustrated 7



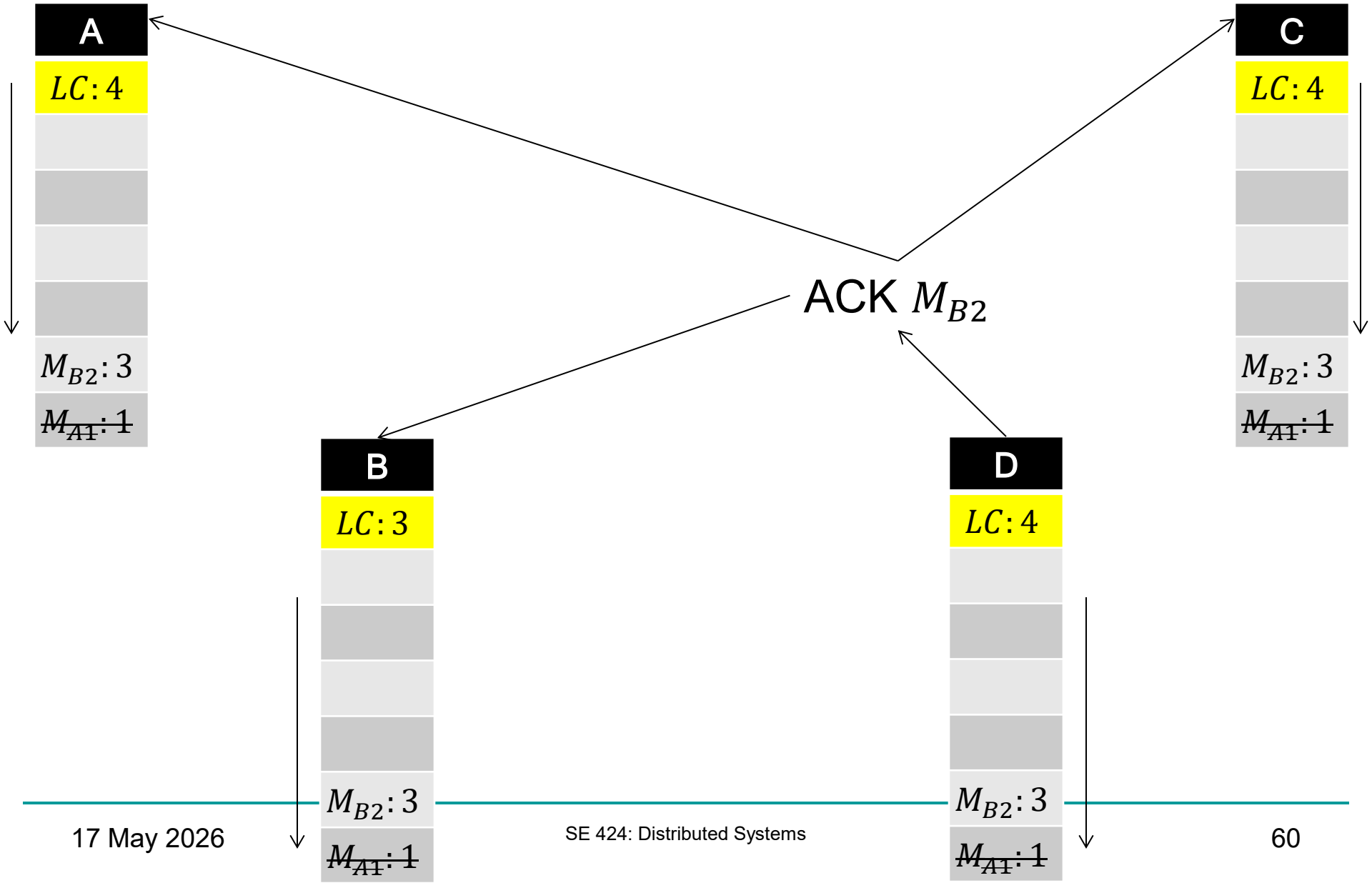
TOM ACKs Illustrated 8



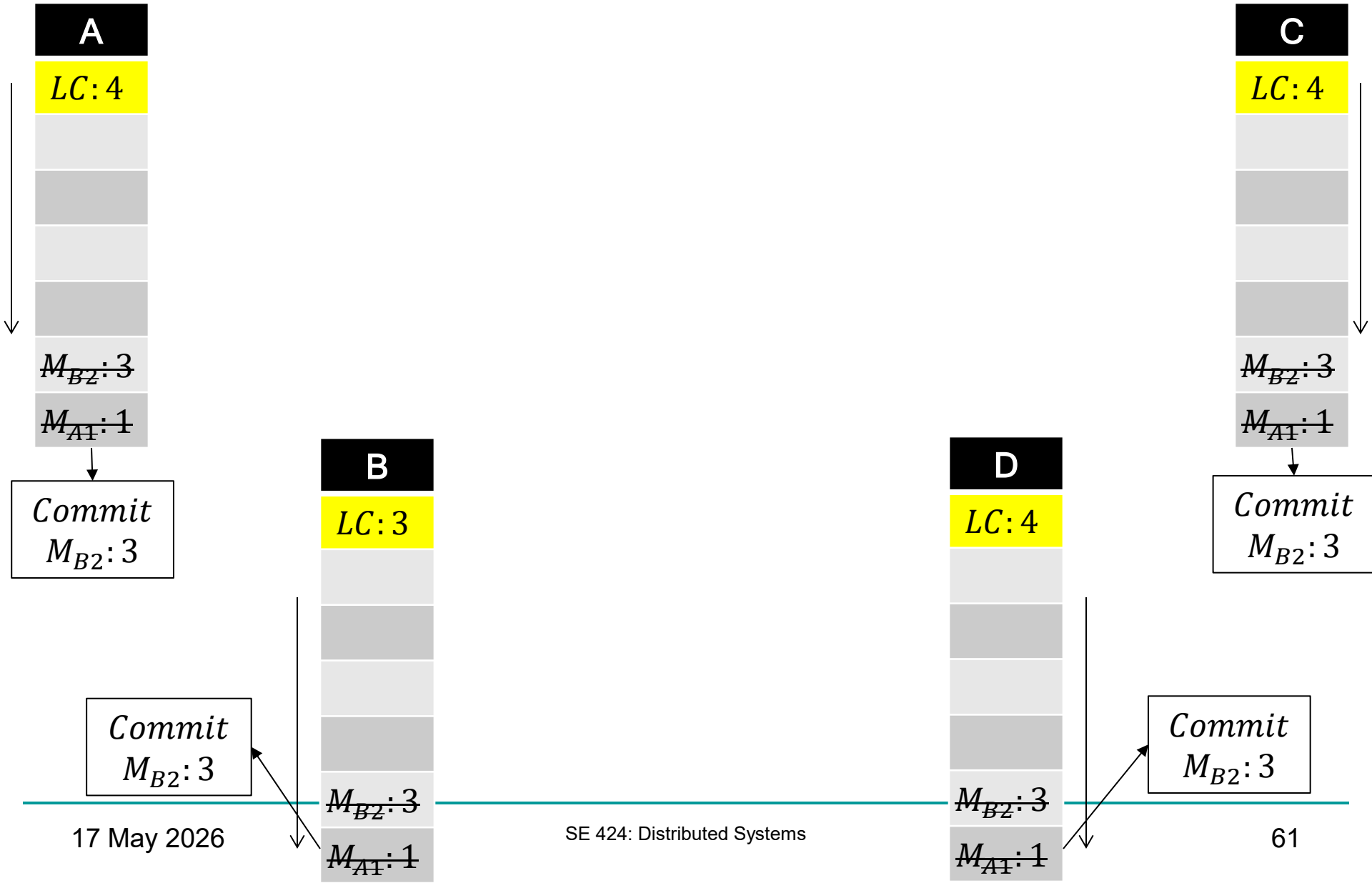
TOM ACKs Illustrated 9



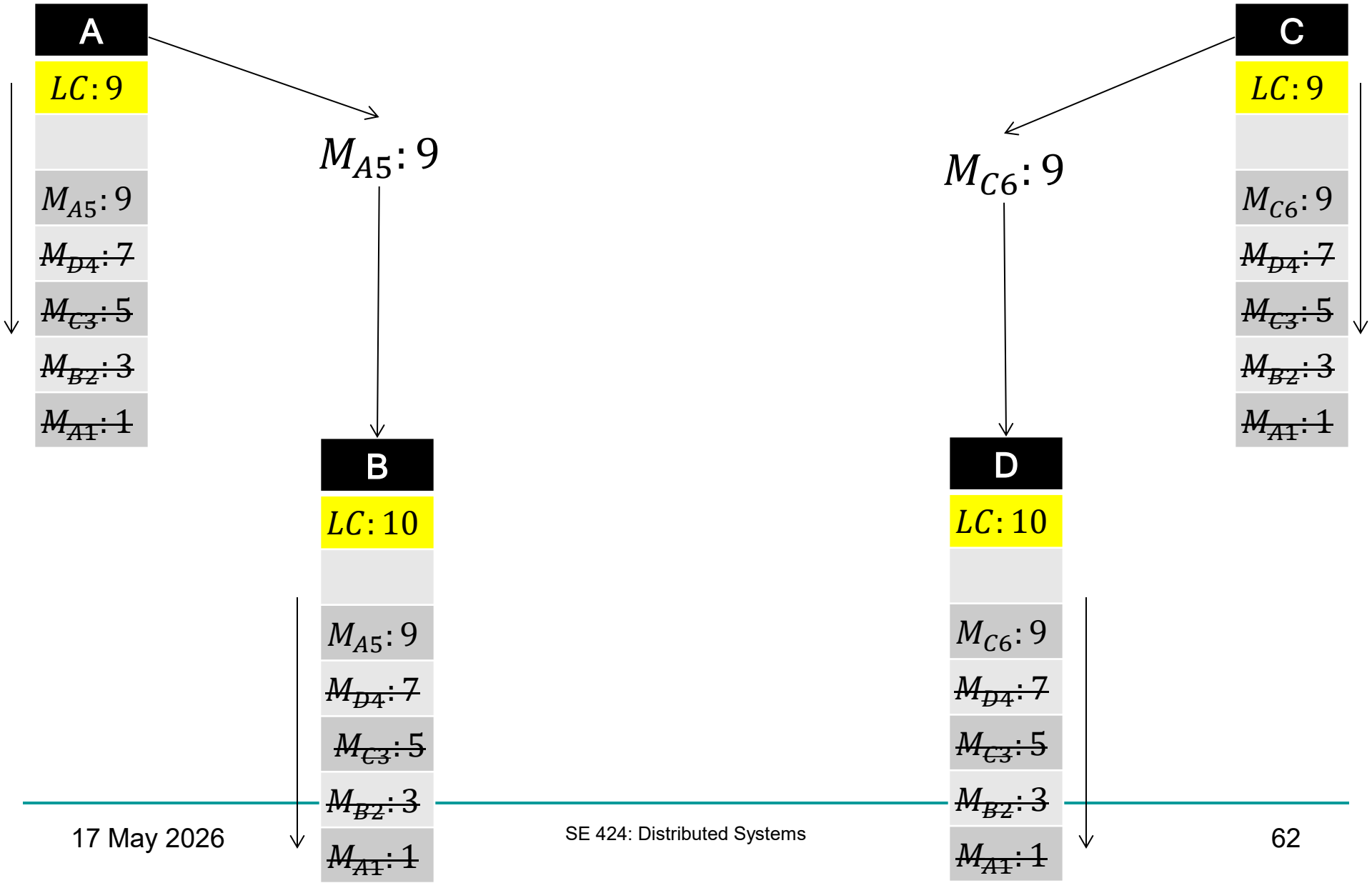
TOM ACKs Illustrated 10



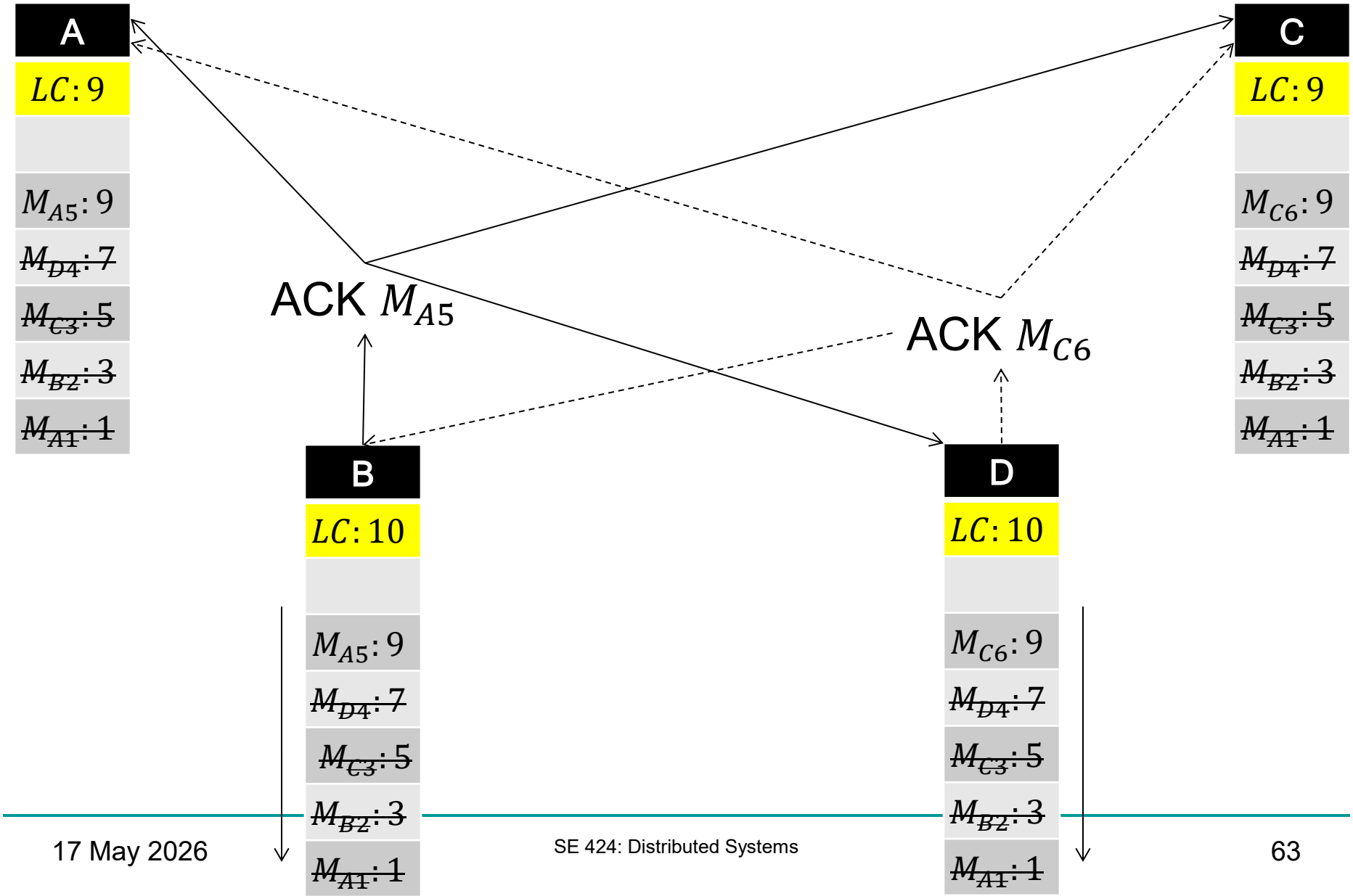
TOM ACKs Illustrated 11



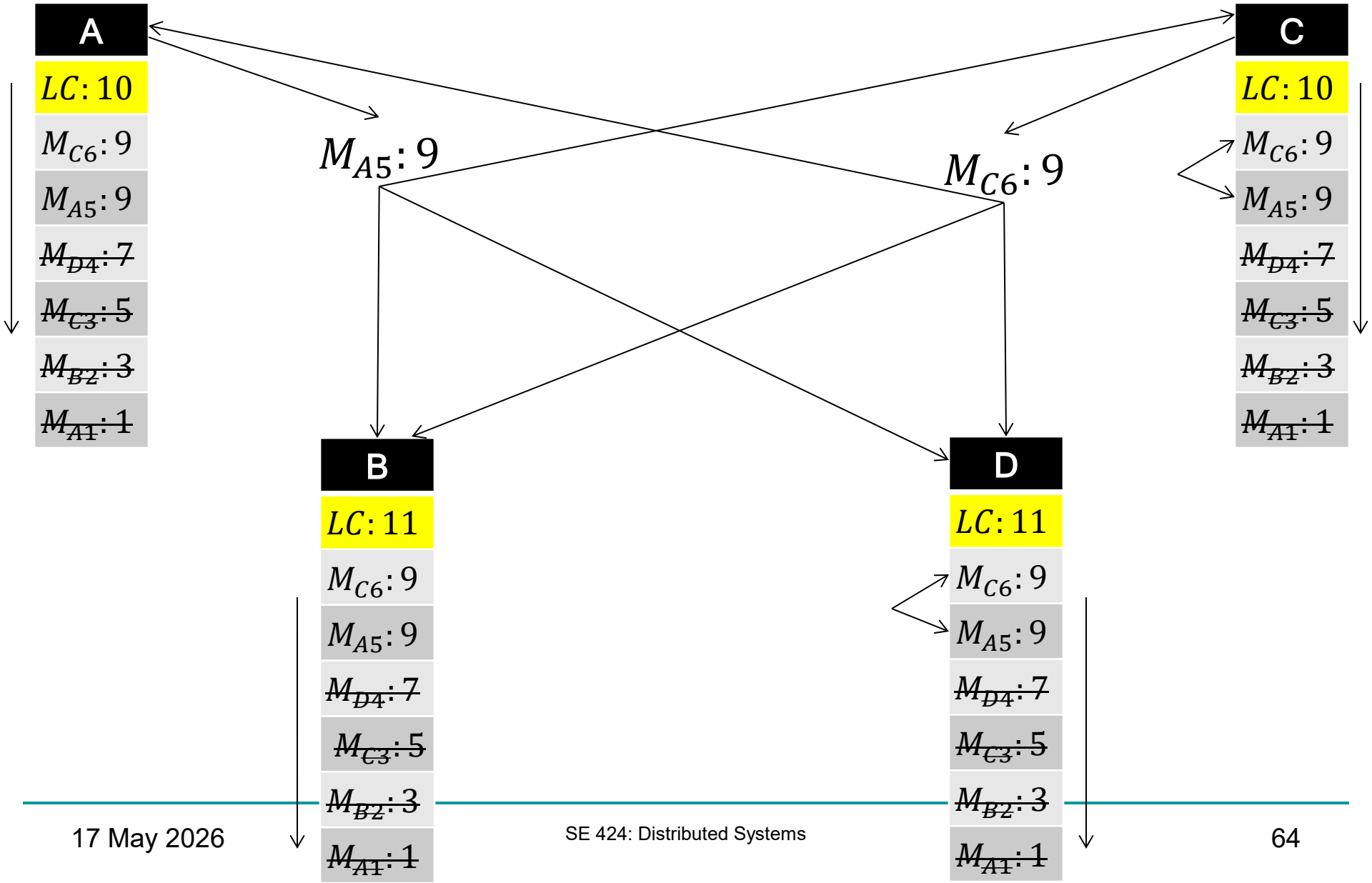
TOM ACKs Illustrated 12



TOM ACKs Illustrated 12



TOM ACKs Illustrated 13



Conclusion

- Physical Clocks
  - GPS
  - Synchronization
- Logical Clocks
  - Lamport
  - Totally Ordered Multicast