

25 YEARS ANNIVERSARY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

CÁC HỆ THỐNG PHÂN TÁN VÀ ỨNG DỤNG



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# CHƯƠNG 4: ĐỒNG BỘ HÓA TRONG HPT

Based on the lectures of Assoc. Prof. Hà Quốc Trung

# Nội dung

- Đồng bộ hóa đồng hồ vật lý
- Đồng bộ hóa đồng hồ logic
- Các thuật toán loại trừ lẫn nhau
- Các thuật toán bầu chọn

# Mở đầu

- Các tiến trình thực hiện đồng bộ hóa ntn?
  - Nhiều tiến trình cần lần lượt vào sử dụng tài nguyên chia sẻ dùng chung tí 1 thi im TN ch phc v c 1 or 1 vài tin trình => cn c ch ln lt vào s dung TN chia s dung chung
  - Nhiều tiến trình cần thống nhất thứ tự các sự kiện
  - Vấn đề đối với ngũ cảnh HPT? (mì tin trình nm trên 1 máy tính c lp, các máy tính kt ni vi nhau qua 1 h tng mng)
- Đồng bộ hóa dựa trên giá trị thời gian thực
- Đồng bộ hóa dựa trên thứ tự các sự kiện

ng cnh HPT phúc tp hn nhu so vi ng cnh tắt c tin trình u thuc 1 máy tính: do ko có ng h dùng chung và ko có b nh chia s dung chung



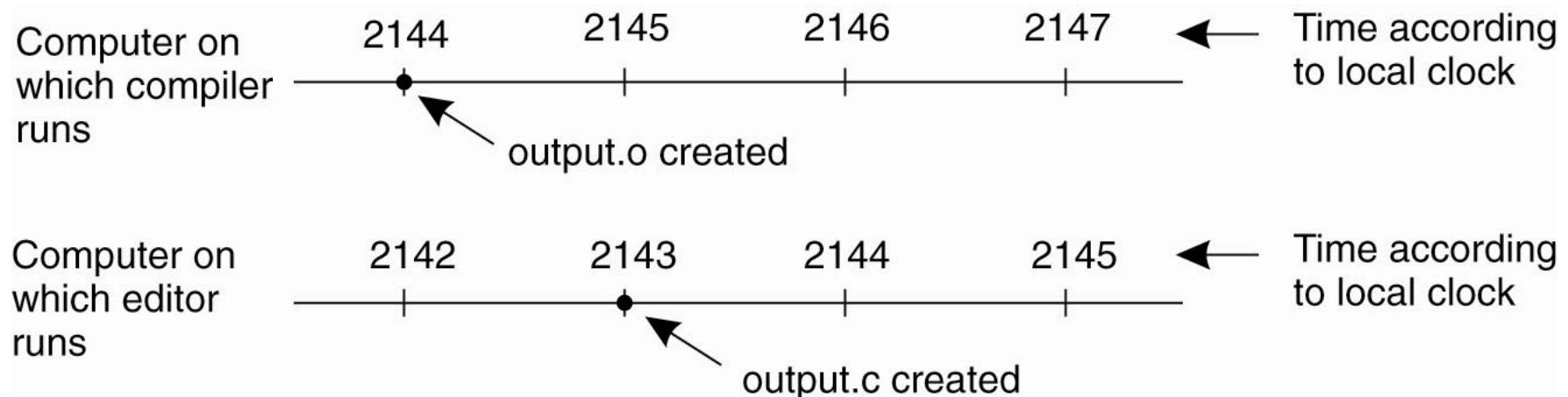
HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# 1. Đồng bộ hóa đồng hồ vật lý

# 1. ĐỒNG BỘ HÓA ĐỒNG HỒ VẬT LÝ

- Đồng hồ vật lý
- Các vấn đề khi không đồng bộ hóa đồng hồ vật lý
- Các thuật toán đồng bộ hóa đồng hồ vật lý

# Ví dụ 1: Lập trình trong HPT



Lập trình vi editor và compiler nm 2 máy tính khác nhau.. Khi dch 1 chng trình c thì compiler ch dch li nhng file .c mà nó có file .o có thi gian thay i din ra trc thi gian thay i ca file .c. Trong trng hp trên, khi LTV thay i file .c, do lnh nhau và ng h vt lý gia 2 máy tính => compiler hiu lm là file .o có thi gian (2144) sau thi gian thay i ca file .c(2143) => compiler ko dch li file .c này na. K vng ca LTV là phi dch li file .c này

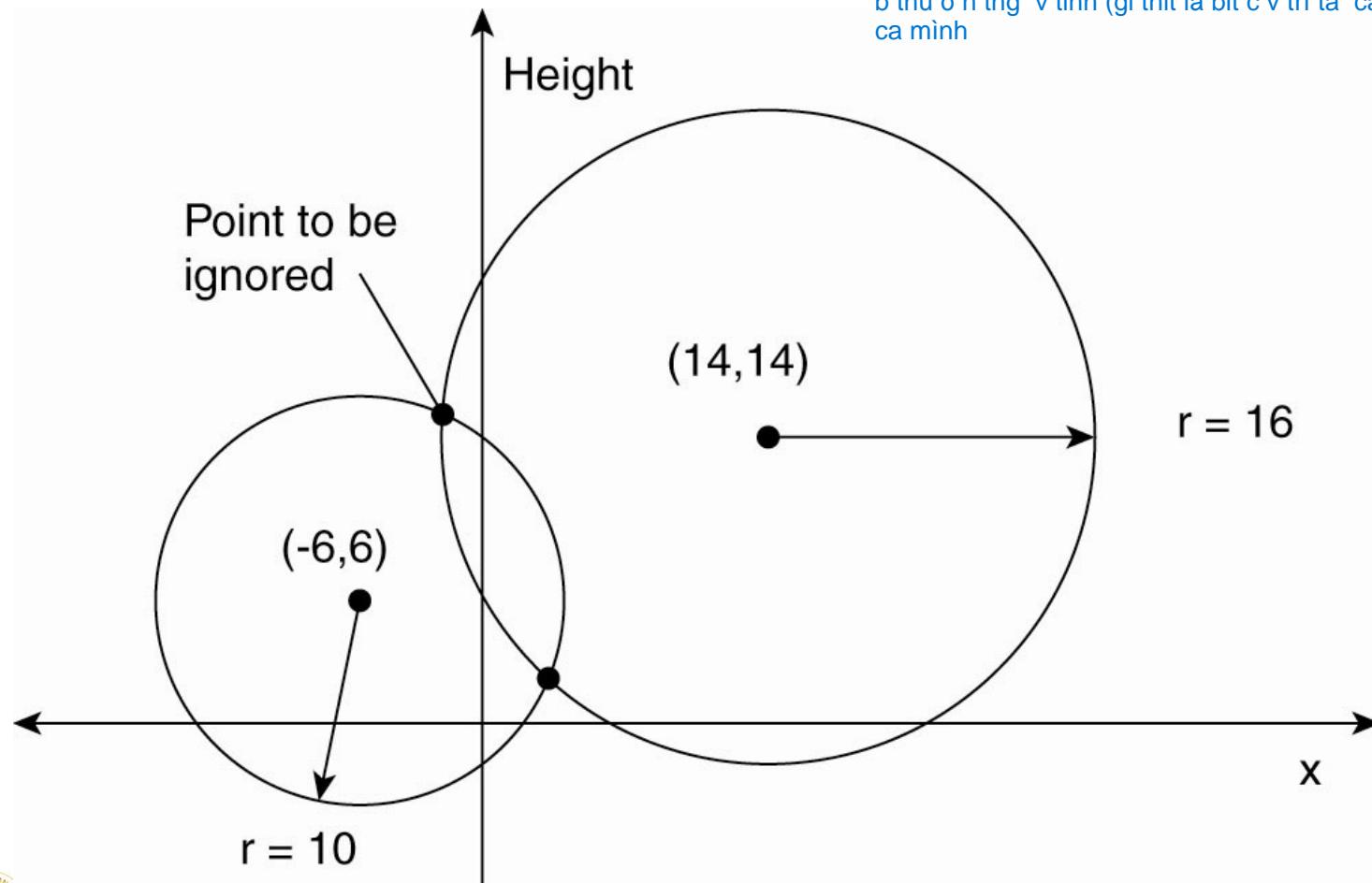
=>. ng h vaajyt lý trong HPT lch nhau -> nh hng n h ca h thng

# VD 2: Global Positioning System

(1)

GPS -h thông nh v toàn cu

C ch h: các b thu nhn tín hiu c gi i t các v tinh sau ó xác nh khong cách t b thu ó n tng v tinh (gi thit là bit c v trí ta ca các v tinh) -> b thu x c v trí ca mìn



# Global Positioning System (2)

- Những khó khăn khi triển khai GPS trong thực tế
  - 1. Tín hiệu đi qua các tầng khí quyển trước khi đến với thiết bị thu
  - 2. Đ Đồng hồ vật lý của thiết bị thu và vệ tinh không đồng bộ

(sai Ich v giá tr thi gian -> thoái gian tính khi tín hiu i t v tinh n b thu có sai Ich -> sai Ich trong vic tính khong cách -> sai Ich trong tính v trí ca b thu trong h thng GPS)

# Đồng hồ vật lý

- Timer
- Counter & Holding register
- Clock tick
- Ván đè trong HPT
  - Đồng bộ với giá trị thời gian vật lý thật
  - Đồng bộ các đồng hồ vật lý với nhau

B timer h dc trên s dao ng ca tinh th thch anh, giá tr khi u s c ghi vào thanh ghi counter và holding. Ban àu giá tr counter c gán bng giá tr holding, sau mì s dao ng ca tinh th thch anh s làm counter gim i 1 n v, n khi giá tr counter gim v 0 thì h thng sinh ra 1 ngt và counter c gán ngc tr l holding. Qung thi gian counter gim giá tr t holding v 0 c gi 1 clock tick, 60 clock tick = 1s



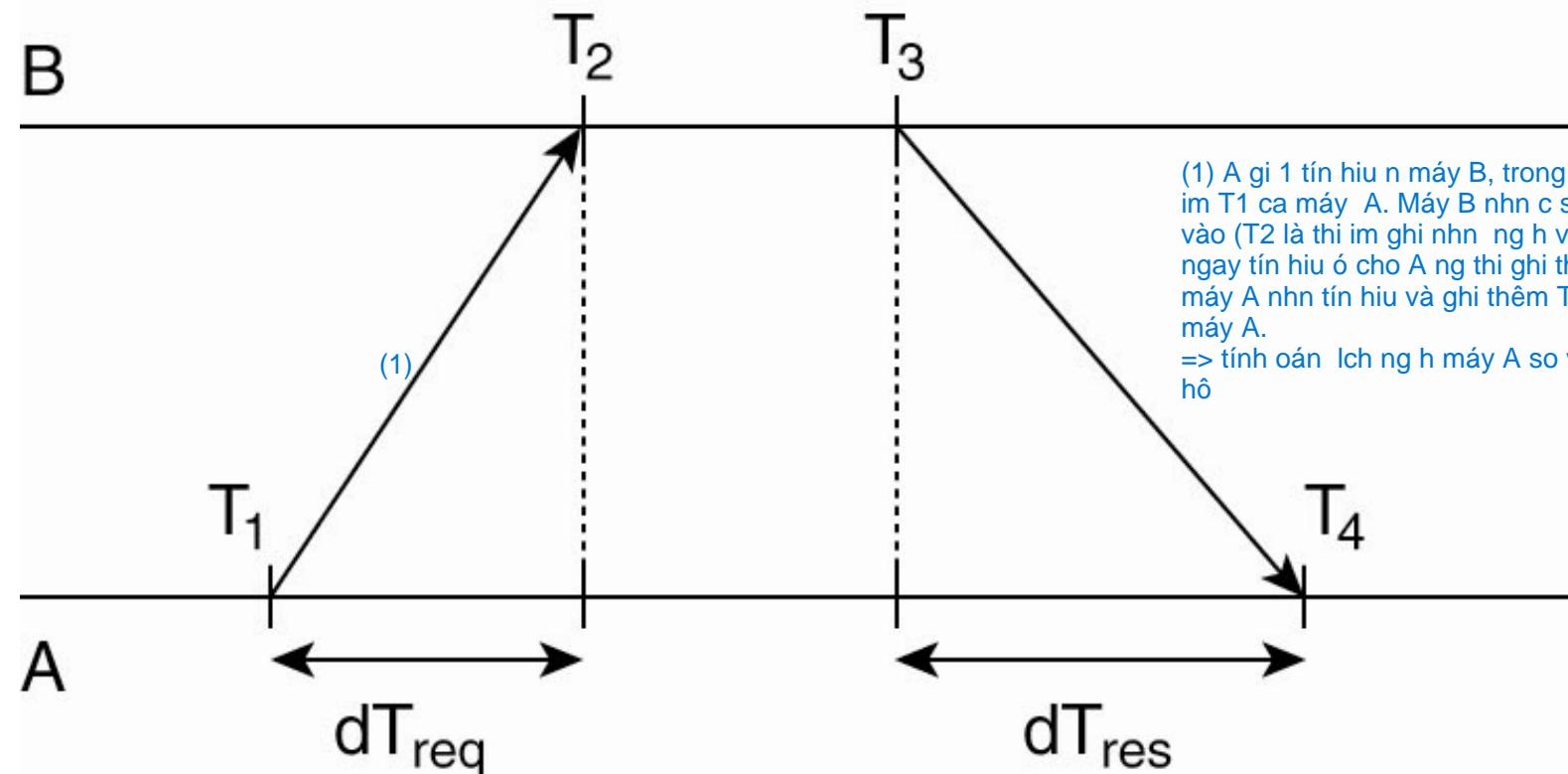
**RTC IC  
(Real Time Clock)**

trong b IC luôn có tinh th thch anh dao ng

# Các thuật toán đồng bộ hóa đồng hồ vật lý

- Network Time Protocol
- Thuật toán Berkeley
- Đồng bộ hóa đồng hồ vật lý trong các mạng không dây

# Network Time Protocol

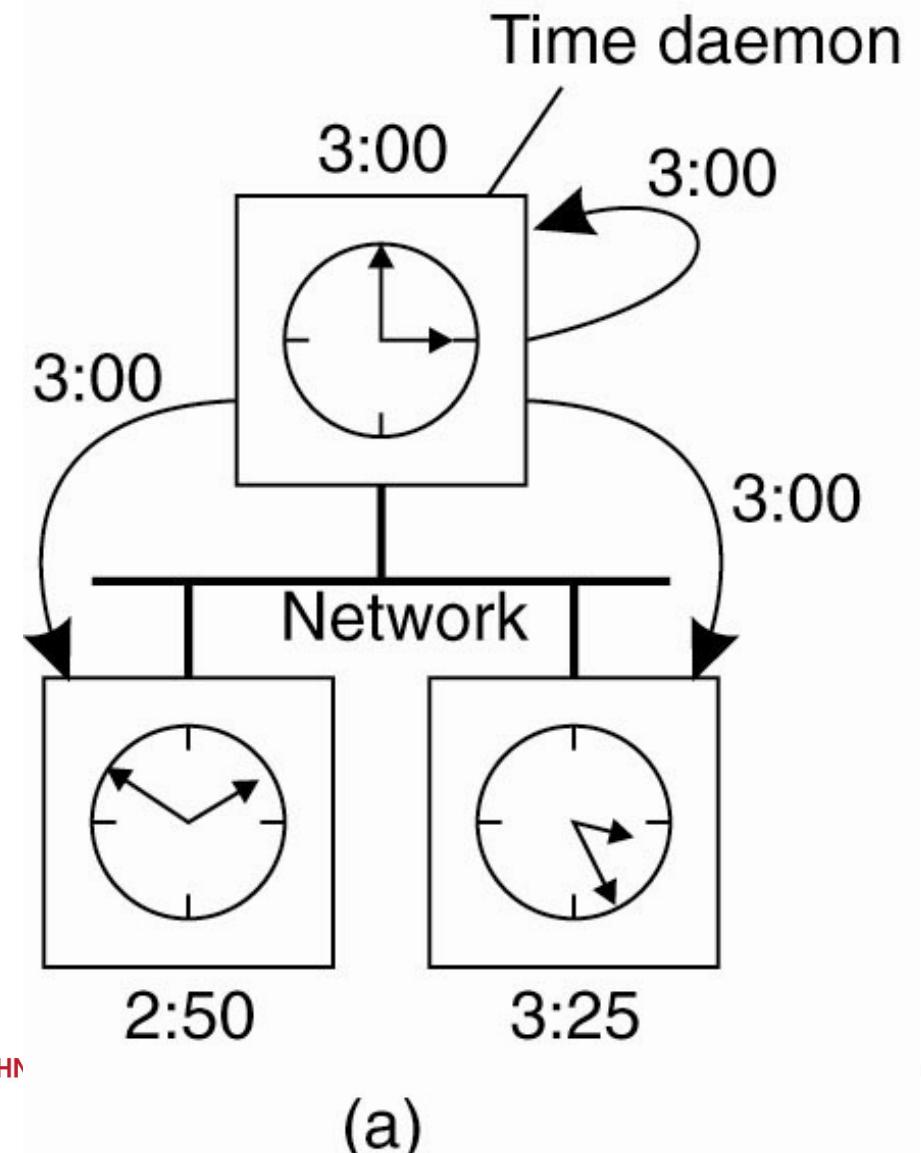


$$\theta = T_3 + \frac{(T_2 - T_1) + (T_4 - T_3)}{2} - T_4 = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

# Cristian's Algorithm

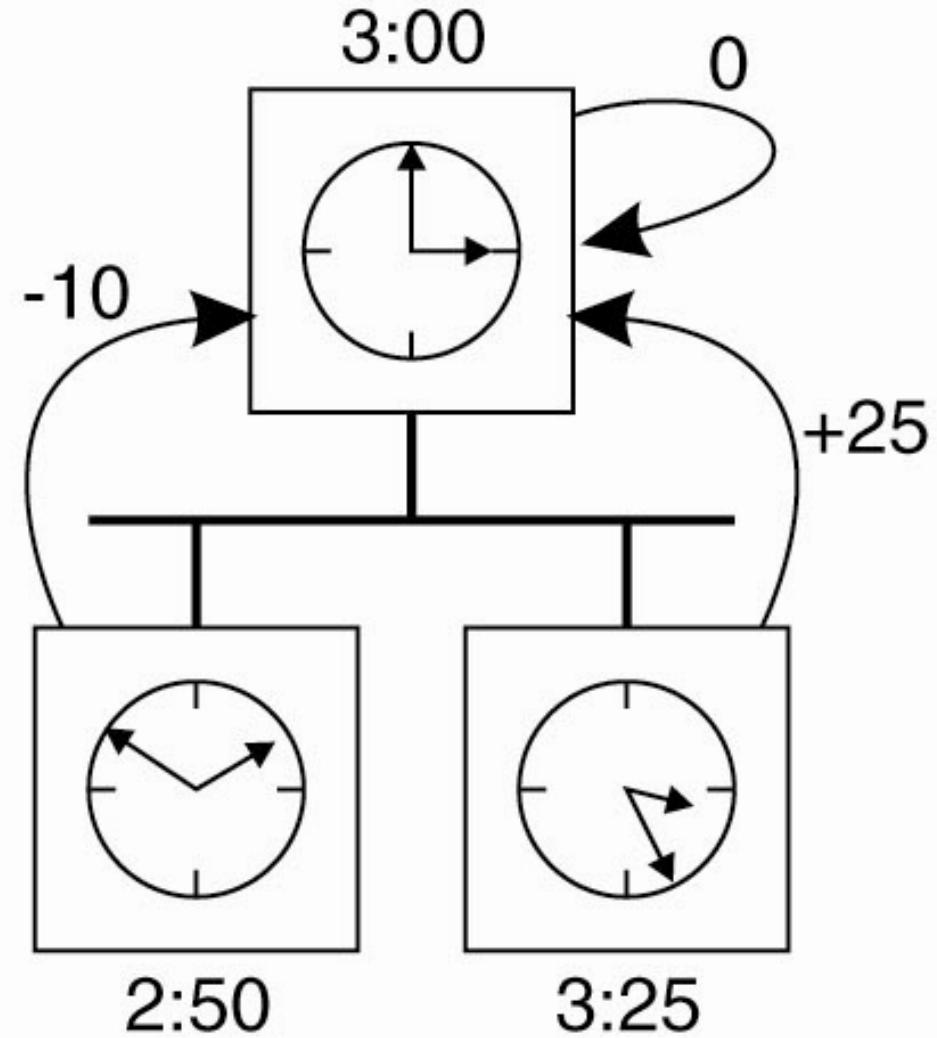
# The Berkeley Algorithm (1)

- Time deamon quảng bá giá trị đồng hồ vật lý của mình



# The Berkeley Algorithm (2)

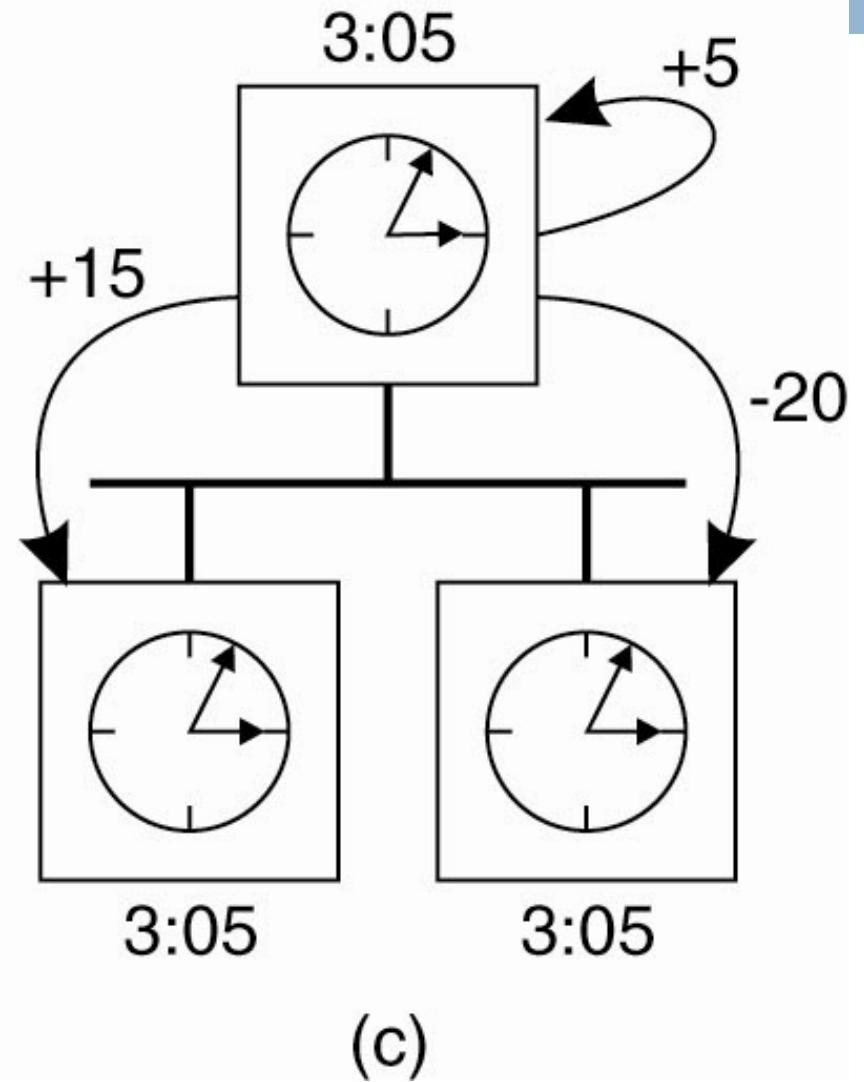
- ❑ các máy khác trả lời



(b)

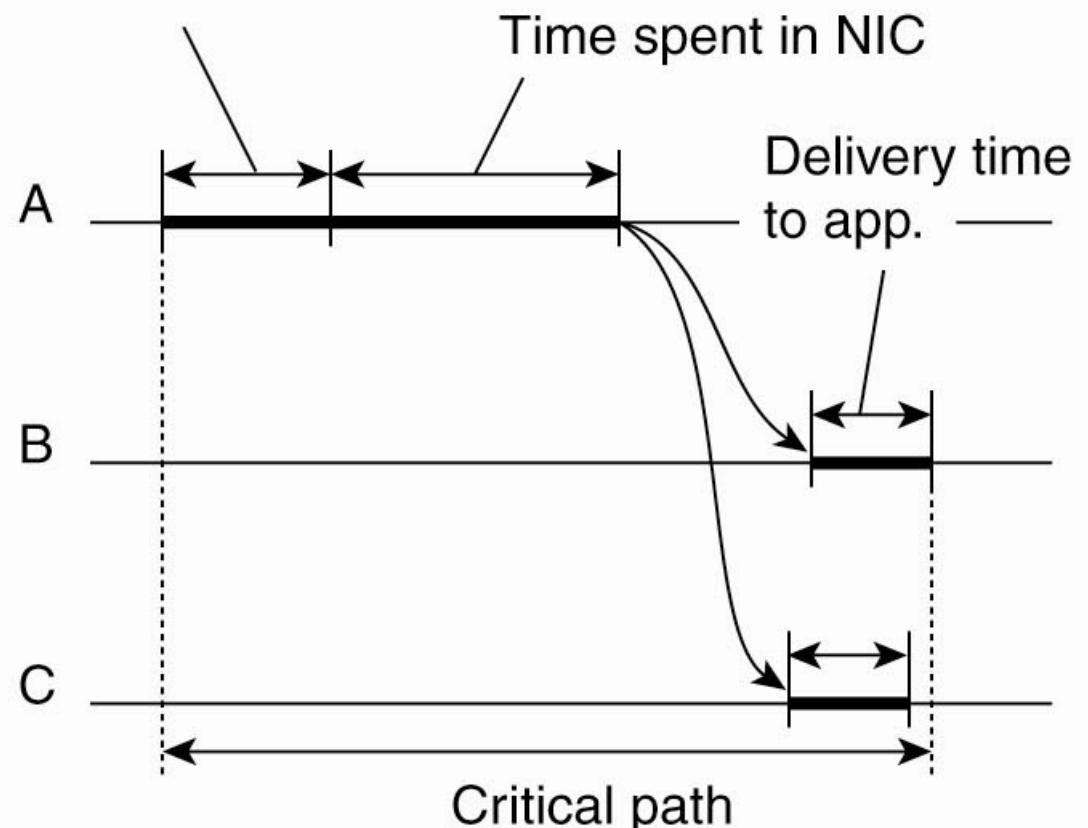
# The Berkeley Algorithm (3)

- Time deamon tính toán ra giá trị trung bình và gửi cho các máy giá trị cần hiệu chỉnh



# Đồng bộ hóa đhvl trong các mạng không dây (1)

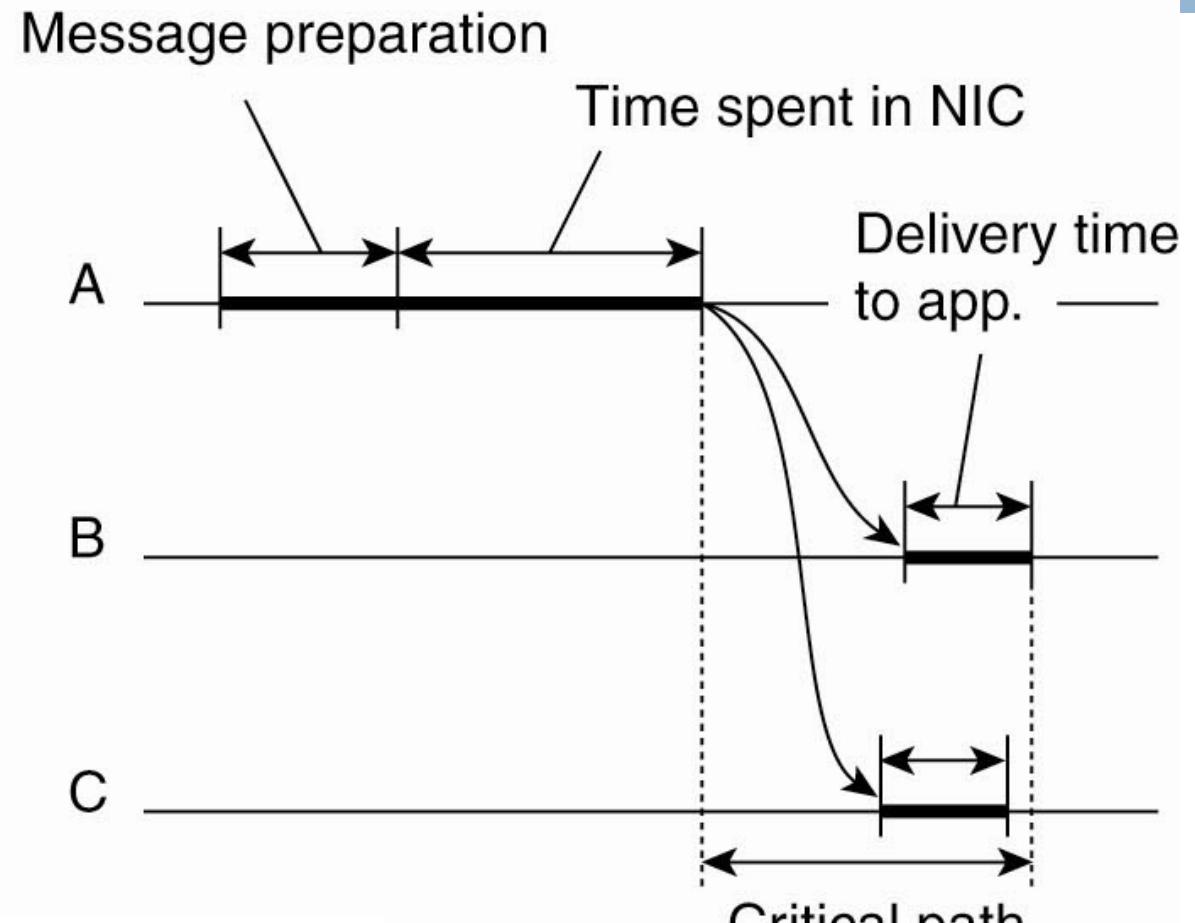
Message preparation



(a)

# Đồng bộ hóa đhvl trong các mạng không dây (2)

- **RBS** (Reference Broadcast Synchronization)



$$Offset [p,q] = \frac{\sum_{k=1}^M (T_{p,k} - T_{q,k})}{M}$$

(b)



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

## 2. Đồng bộ hóa đồng hồ logic

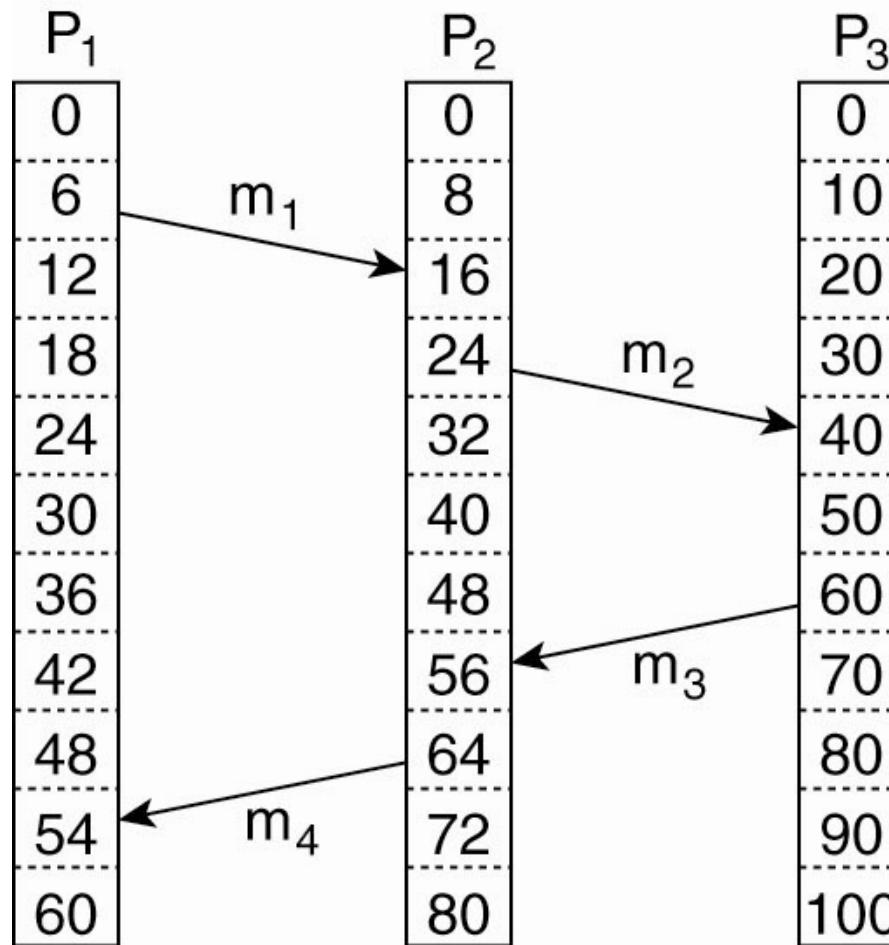
## 2. Đồng bộ hóa đồng hồ logic

- Cơ chế đồng bộ hóa đồng hồ logic của Lamport
- Vector clocks

## 2.1. Cơ chế đbh đồng hồ logic của Lamport (1)

- Mỗi quan hệ “xảy ra trước” (happens-before) → diễn ra với 2 ngũ cảnh:
  1. Nếu  $a$  và  $b$  là các sự kiện của cùng 1 tiến trình, và  $a$  xảy ra trước  $b$ , thì  $a \rightarrow b$  là đúng.
  2. Nếu  $a$  là sự kiện gửi một thông điệp từ một tiến trình,  $b$  là sự kiện nhận của thông điệp đó ở 1 tiến trình khác, thì  $a \rightarrow b$
- Mỗi qh bắc cầu:  $a \rightarrow b$  và  $b \rightarrow c$ , thì  $a \rightarrow c$
- Với các sự kiện tương tranh  $a$  và  $b$  thì không có  $a \rightarrow b$  và cũng không có  $b \rightarrow a$

# Cơ chế đbh đồng hồ logic của Lamport (2)

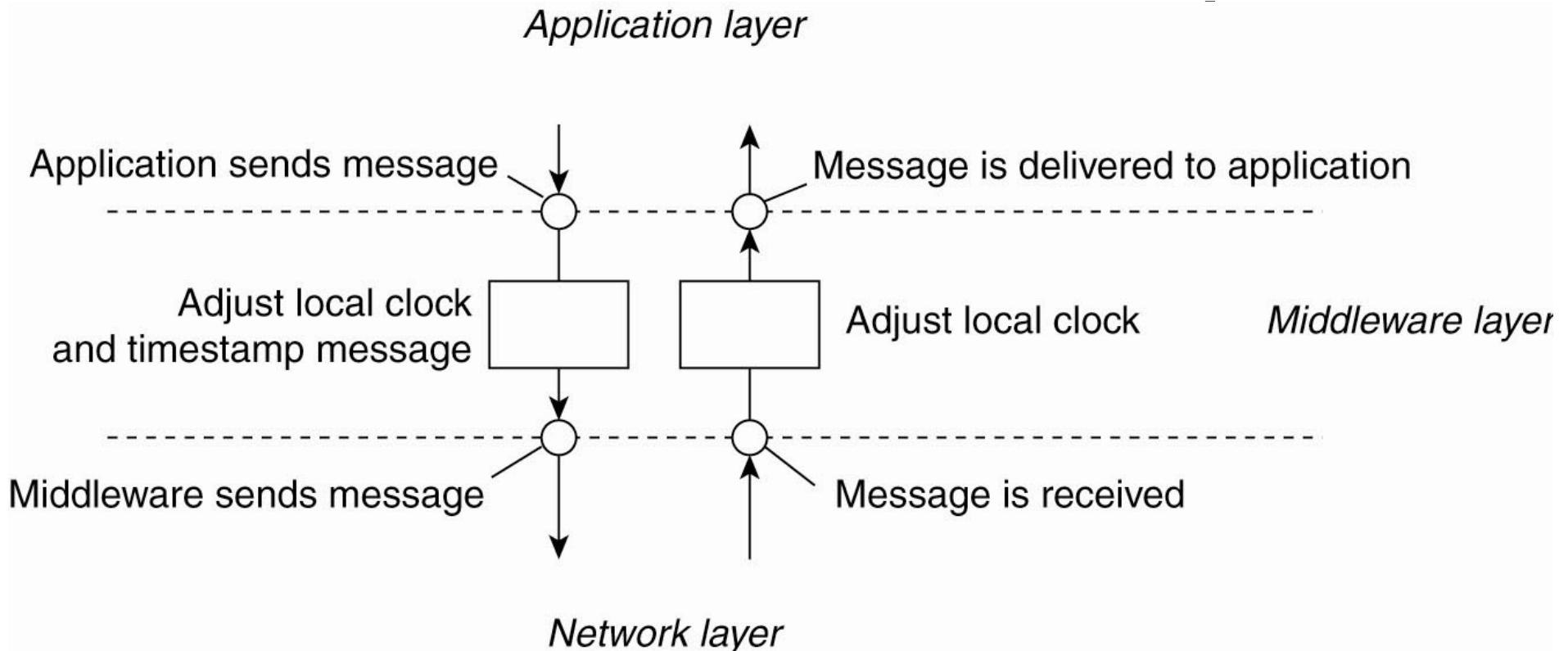


(a)

# Cơ chế đbh đồng hồ logic của Lamport (3)

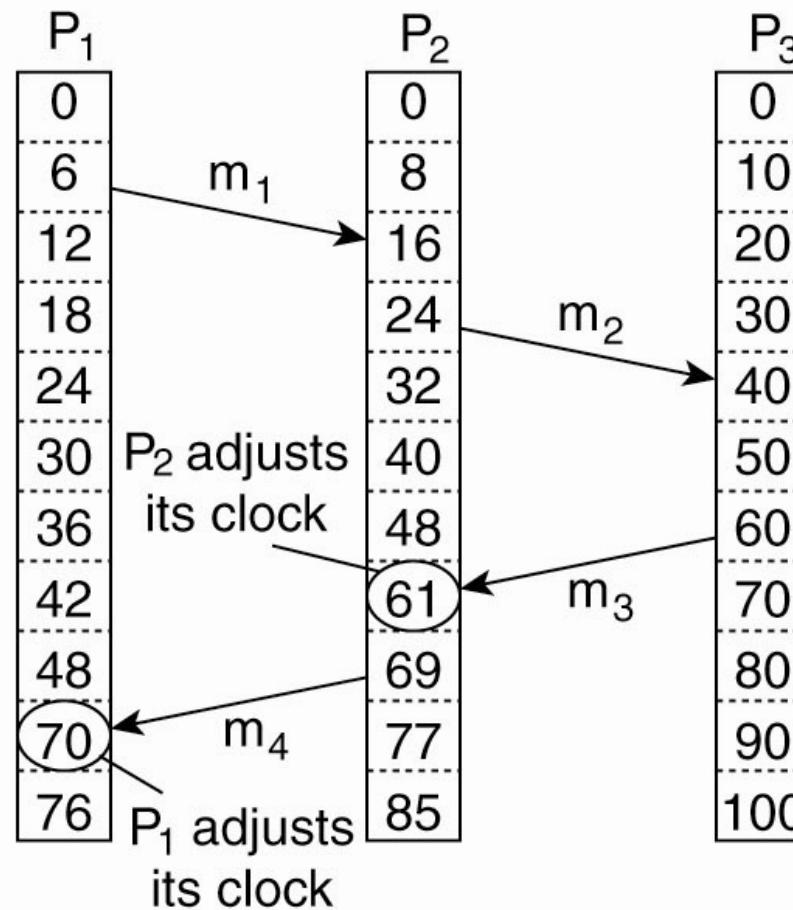
- Cập nhật bộ đếm  $C_i$  cho tiến trình  $P_i$ 
  1. Trước mỗi sự kiện,  $P_i$  thực thi:  
 $C_i \leftarrow C_i + 1.$
  2. Khi tiến trình  $P_i$  gửi thông điệp  $m$  tới  $P_j$ , nó sẽ đặt timestamp của  $m$  là  $ts(m)$  bằng với giá trị  $C_i$  (sau khi thực hiện bước 1).
  3. Khi nhận được thông điệp  $m$ , tiến trình  $P_j$  cập nhật lại giá trị bộ đếm cục bộ:  
 $C_j \leftarrow \max\{C_j, ts(m)\}$ , sau đó sẽ chuyển thông điệp lên tầng ứng dụng.

# Cơ chế đbh đồng hồ logic của Lamport (4)



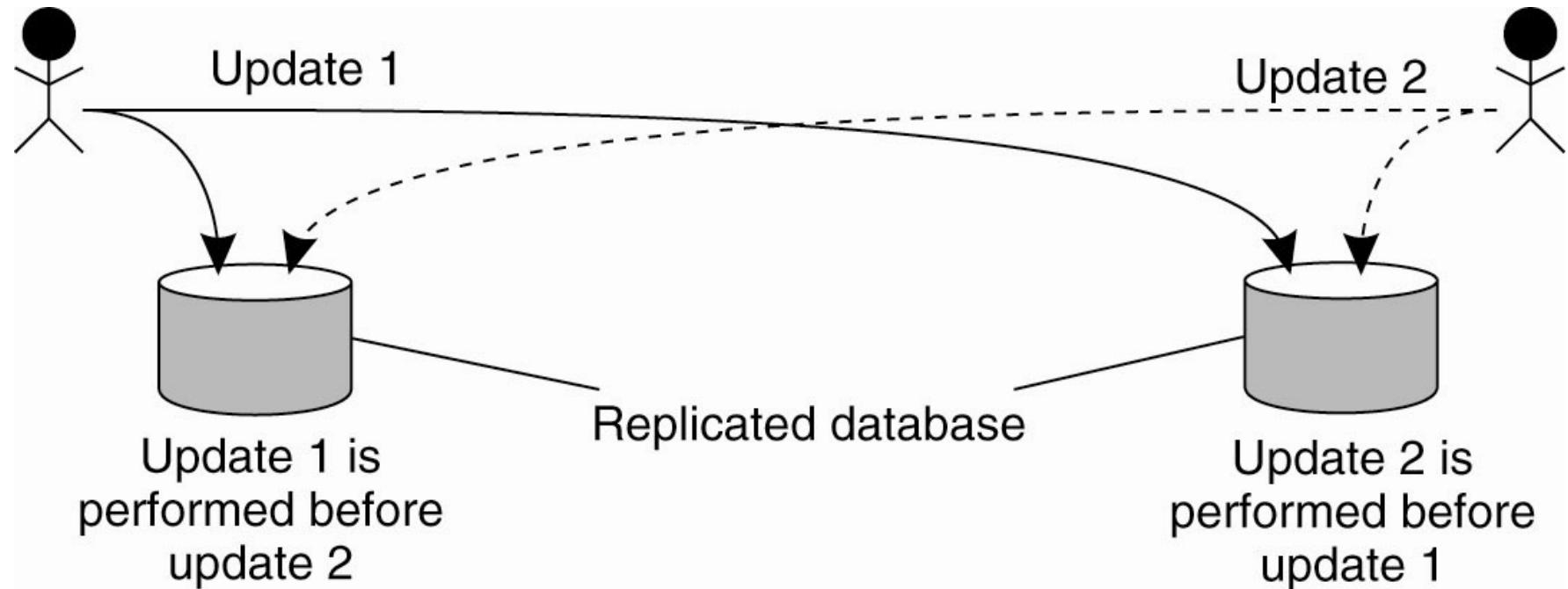
# Cơ chế đbh đồng hồ logic của Lamport (5)

(b) Giải thuật Lamport hiệu chỉnh lại các giá trị clock



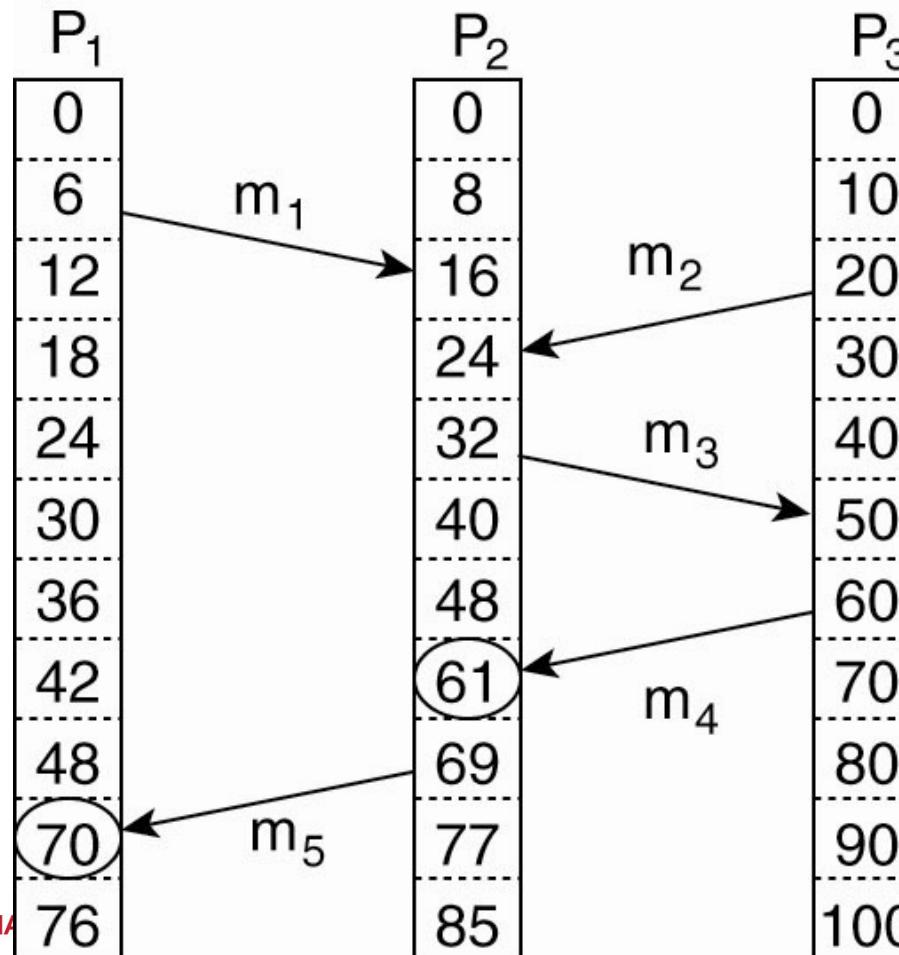
(b)

# Ứng dụng của đbh đh logic của Lamport: Đảm bảo thứ tự toàn cục của gửi thông điệp theo nhóm (Totally Ordered Multicasting)



## 2.2. Vector Clocks (1)

- Việc truyền thông điệp có tính tương tranh sử dụng đồng hồ logic



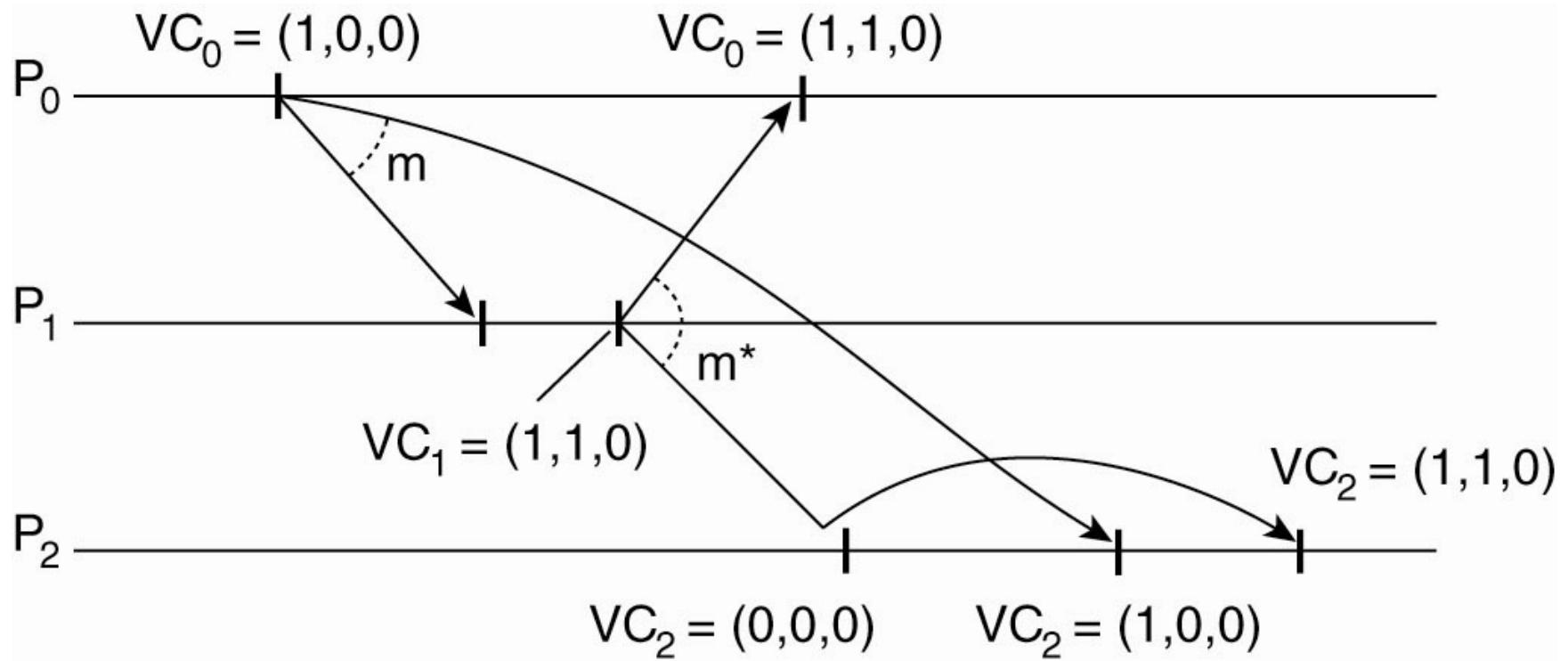
# Vector Clocks (2)

- Véc-tơ clock được xây dựng bằng cách để mỗi tiến trình  $P_i$  quản lý một véc-tơ  $VC_i$  với 2 tính chất sau:
  1.  $VC_i [ i ]$  là số các sự kiện đã xảy ra ở  $P_i$ . nói cách khác,  $VC_i [ i ]$  là đồng hồ logic cục bộ của  $P_i$  .
  2. Nếu  $VC_i [ j ] = k$  thì  $P_i$  biết rằng đã có  $k$  sự kiện xảy ra ở  $P_j$ . Là tri thức của  $P_i$  về thời gian cục bộ ở  $P_j$  .

# Vector Clocks (3)

- Những bước cập nhật để đảm bảo tính chất 2:
  1. Trước mỗi sự kiện,  $P_i$  thực thi:  
$$\text{VC}_i[i] \leftarrow \text{VC}_i[i] + 1.$$
  2. Khi  $P_i$  gửi thông điệp  $m$  tới  $P_j$ , nó đặt véc-tơ timestamp  $ts(m)$  vào  $m$  bằng với  $\text{VC}_i$  sau khi thực hiện bước 1.
  3. Khi nhận được 1 thông điệp  $m$ , tiến trình  $P_j$  cập nhật lại véc-tơ của mình:  
$$\text{VC}_j[k] \leftarrow \max\{\text{VC}_j[k], ts(m)[k]\}$$
 với mọi  $k$ , sau đó mới gửi thông điệp lên tầng ứng dụng.

# Thực thi trao đổi thông tin có tính nhân quả



- 2 điều kiện:
- 1.  $ts(m)[i] = VC_j[i] + 1$
  - 2.  $ts(m)[k] \leq VC_j[k]$  for all  $k \neq i$



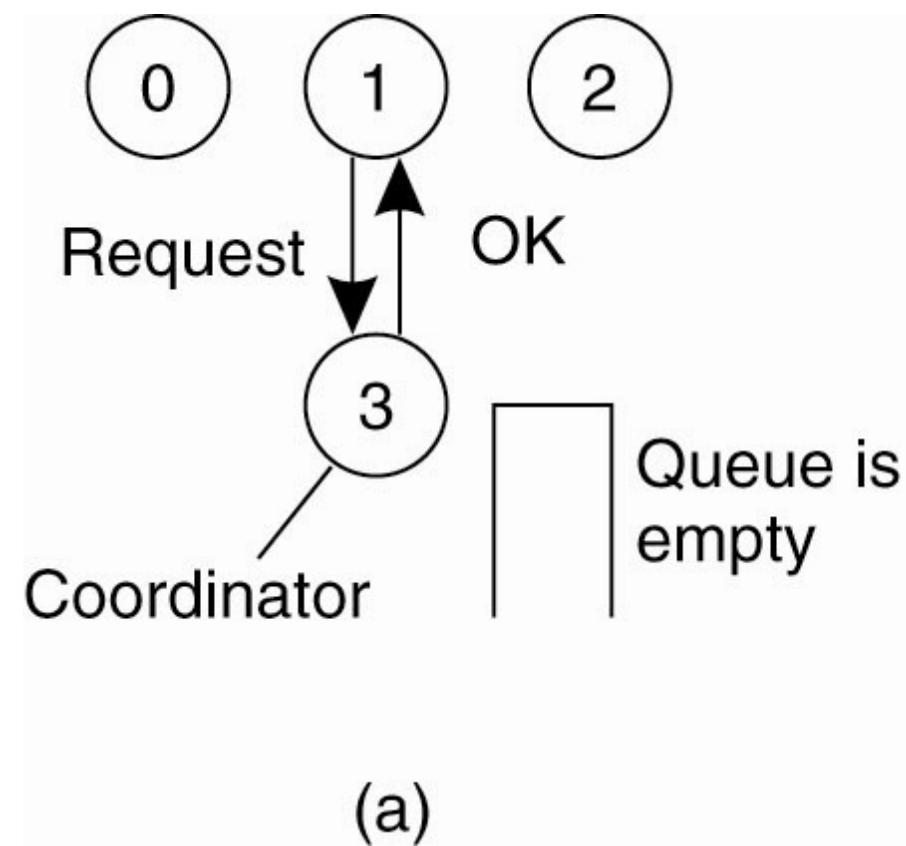
HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

### 3. Các thuật toán loại trừ lẫn nhau

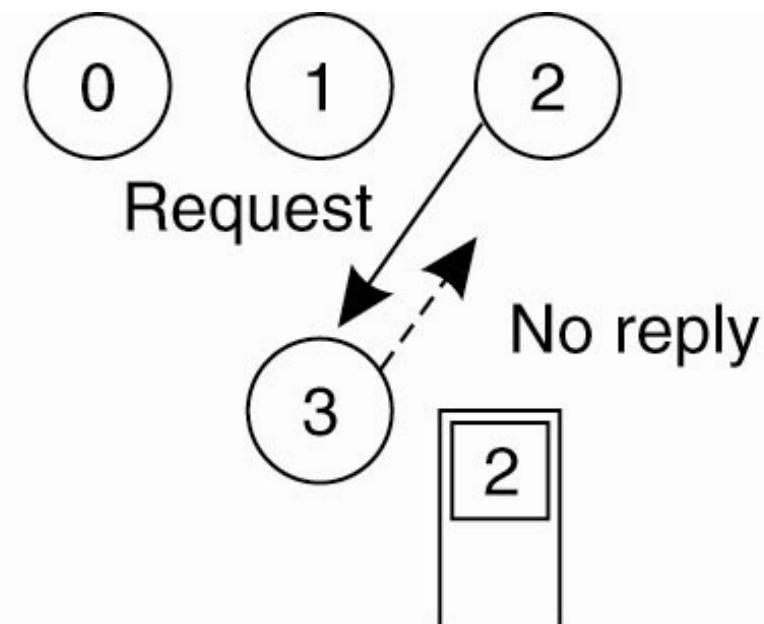
### 3. Các thuật toán loại trừ lẩn nhau

- Giải thuật tập trung
- Giải thuật phân tán
- Giải thuật Token Ring

### 3.1. Giải thuật tập trung (1)

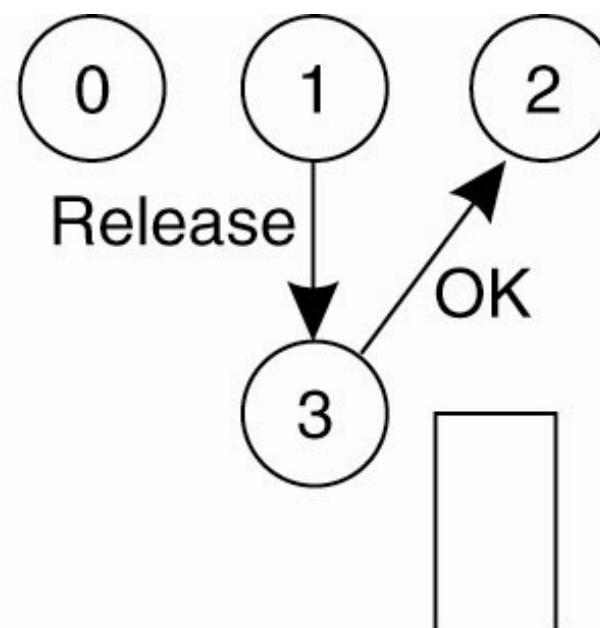


# Giải thuật tập trung (2)



(b)

# Giải thuật tập trung (3)

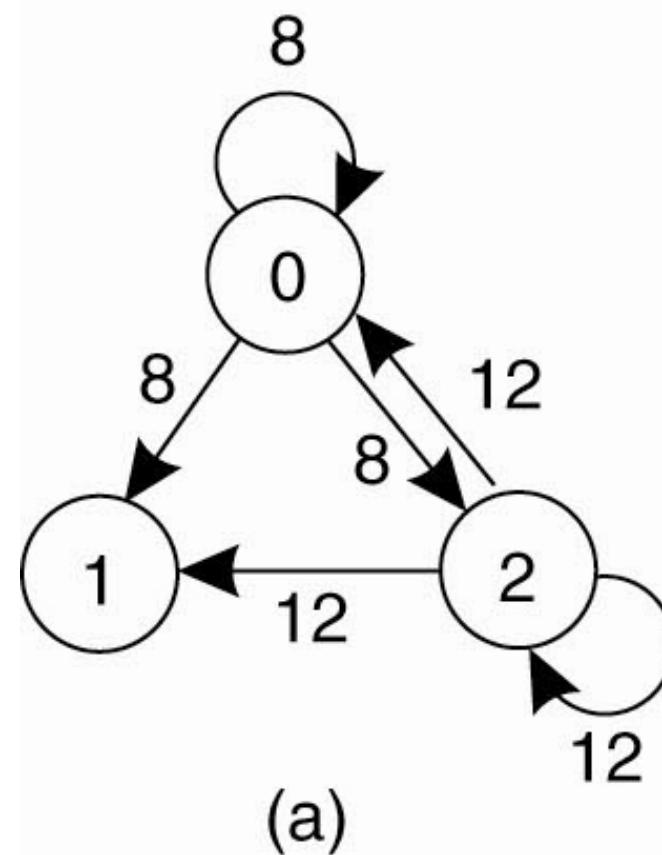


(c)

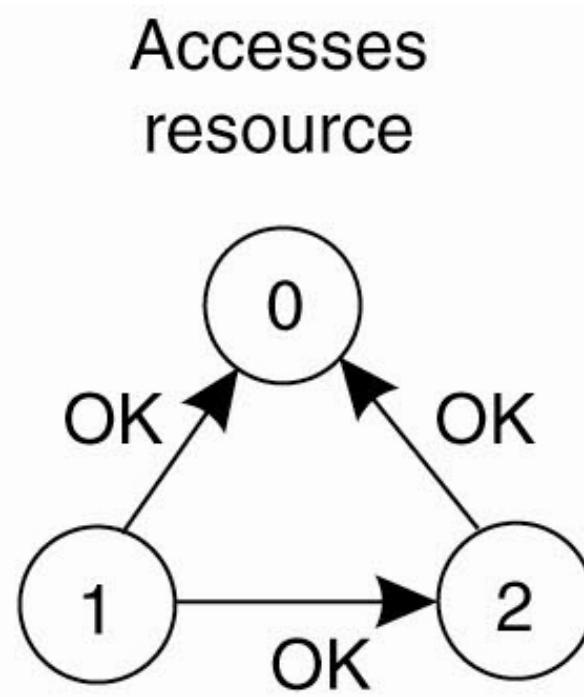
## 3.2. Giải thuật phân tán (1)

- Muốn vào dùng tài nguyên → quảng bá thông điệp request
- Với mỗi tiến trình nhận được request:
  1. Nếu đang không dùng và không muốn dùng TN: Trả lời OK
  2. Nếu đang dùng: không trả lời và lưu vào hàng đợi
  3. Nếu đang không dùng nhưng cũng muốn dùng: so sánh timestamp
- Chỉ vào dùng TN khi nhận đủ các OK của các tiến trình khác

# Giải thuật phân tán (2)

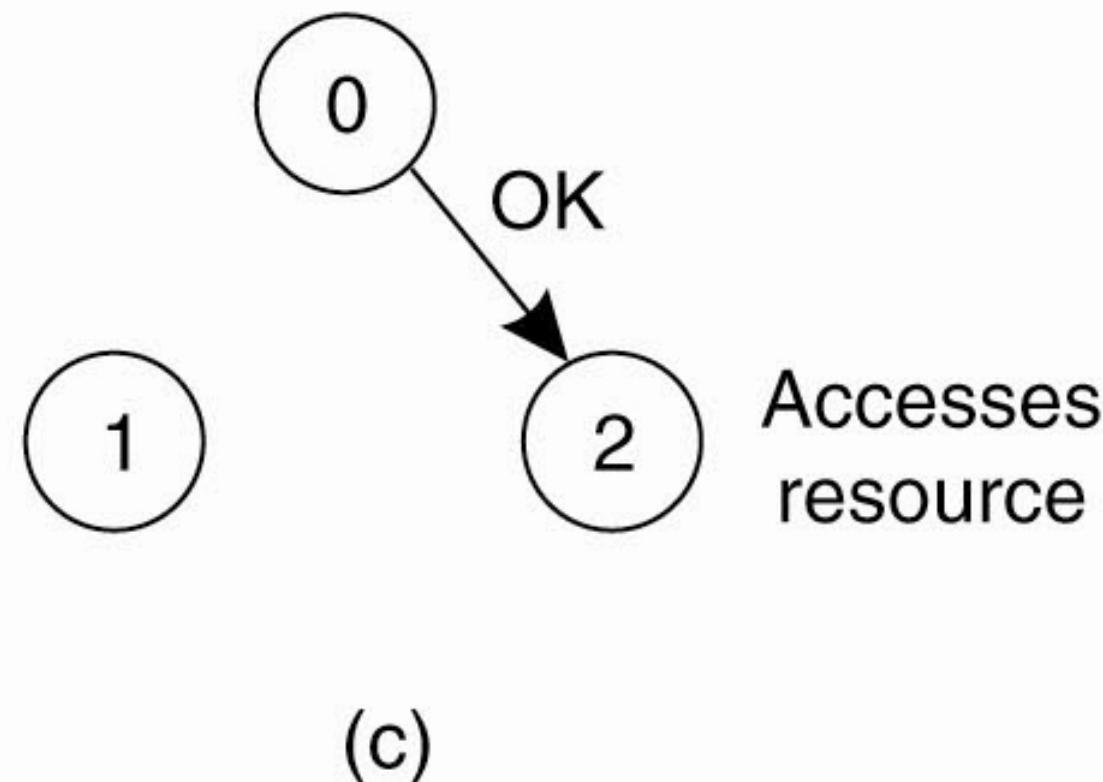


# Giải thuật phân tán (3)

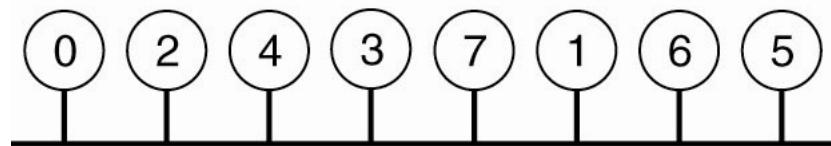


(b)

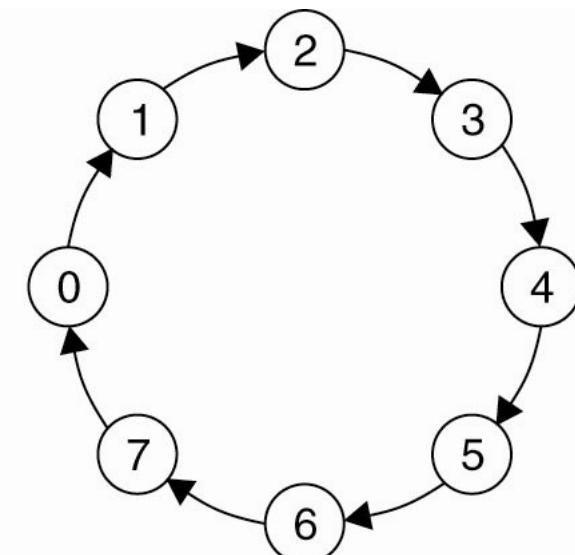
# Giải thuật phân tán (4)



### 3.3. Giải thuật Token Ring



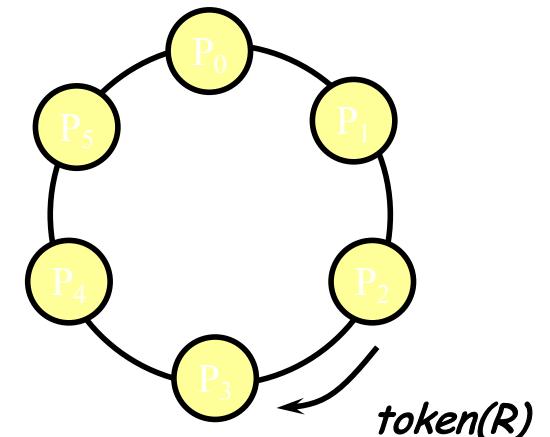
(a)



(b)

# Token Ring algorithm

- Khởi đầu
  - Tiến trình P<sub>0</sub> có token để vào sử dụng TN
- Token được truyền đi trong vòng topo
  - Từ P<sub>i</sub> truyền đến P<sub>(i+1)mod N</sub>
- Khi một tiến trình nhận được token:
  - Tự kiểm tra xem có muốn vào sử dụng TN không
  - Nếu không, truyền token cho nút kế tiếp
  - Nếu có, giữ lại token. Dùng xong thì truyền tiếp đi





HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# 4. Các giải thuật bầu chọn

# 4. Các giải thuật bầu chọn

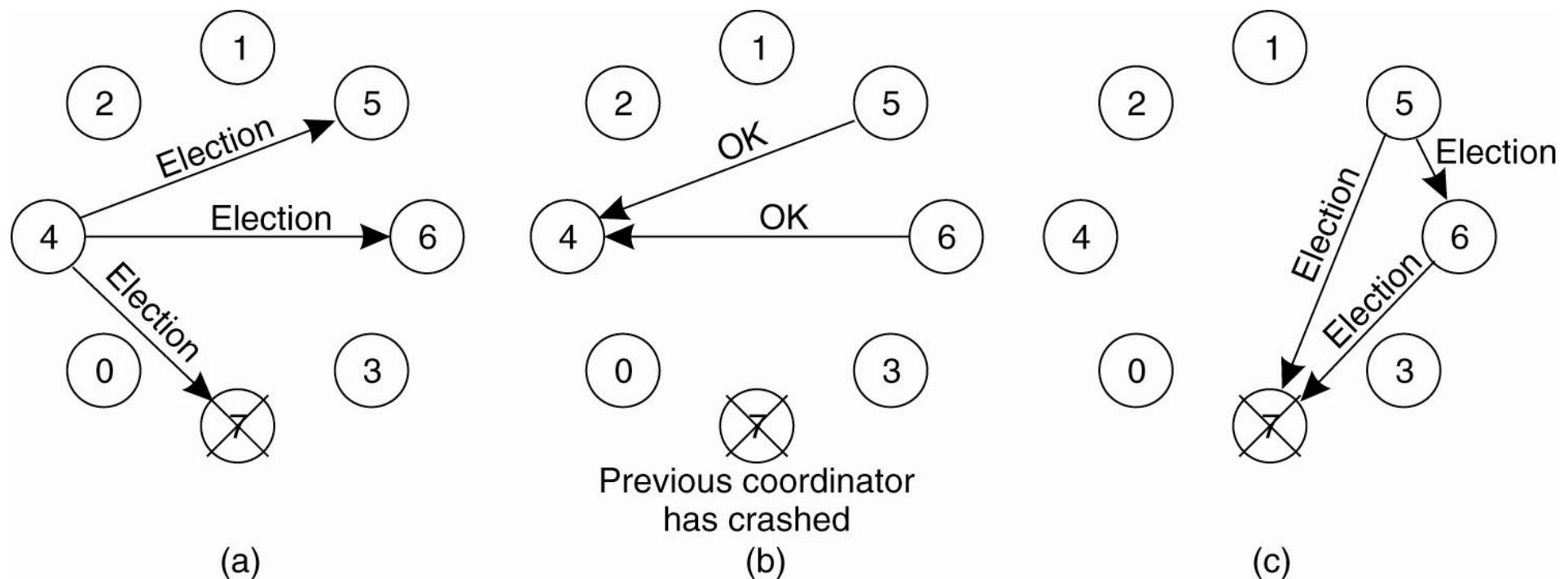
- Các giải thuật truyền thống
  - Giải thuật Bully
  - Giải thuật Ring
- Các giải thuật bầu chọn cho mạng không dây
- Các giải thuật bầu chọn cho mạng diện rộng

# Các giải thuật bầu chọn

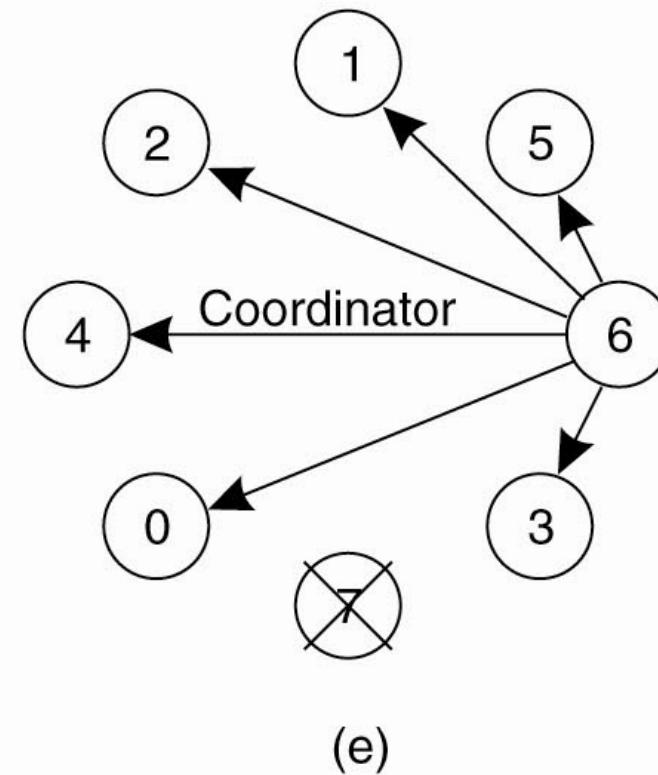
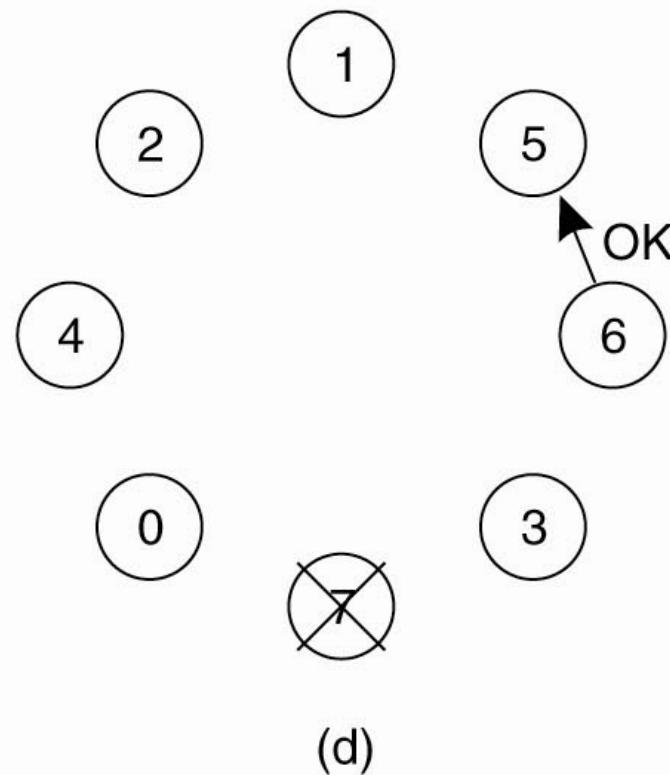
## □ Giải thuật Bully

1. *P* gửi thông điệp *ELECTION* cho tất cả các tiến trình có ID lớn hơn mình
  1. Nếu không có tiến trình nào trả lời (sau timeout), *P* biết mình được chọn
  2. Chỉ cần 1 tiến trình trả lời, *P* biết mình không được chọn và dừng lại.
2. Quá trình trên lặp lại với các tiến trình nhận được *ELECTION*

# Giải thuật BULLY (1)



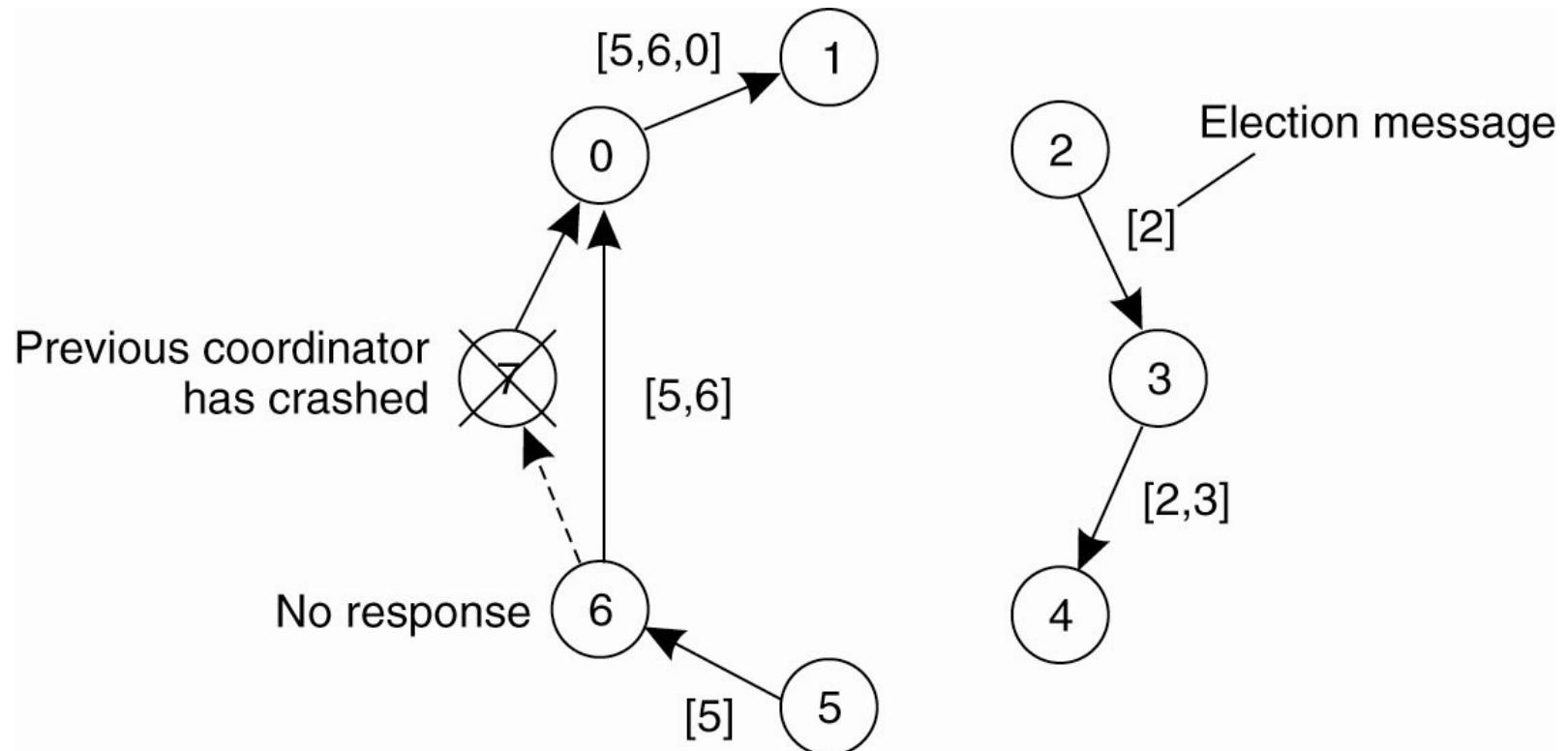
# Giải thuật BULLY (2)



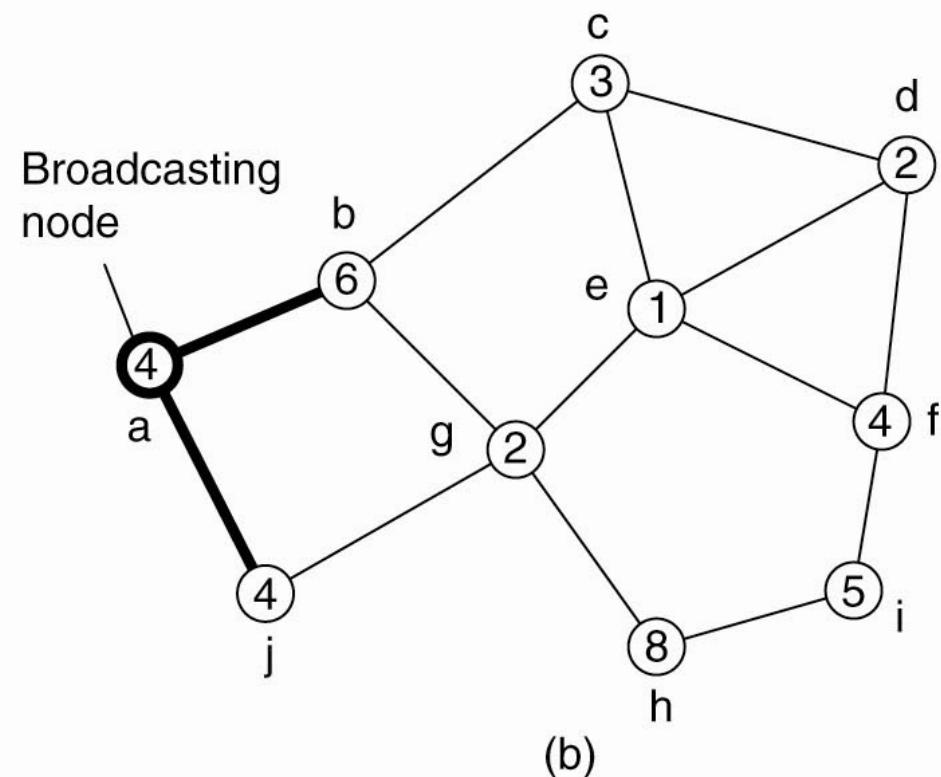
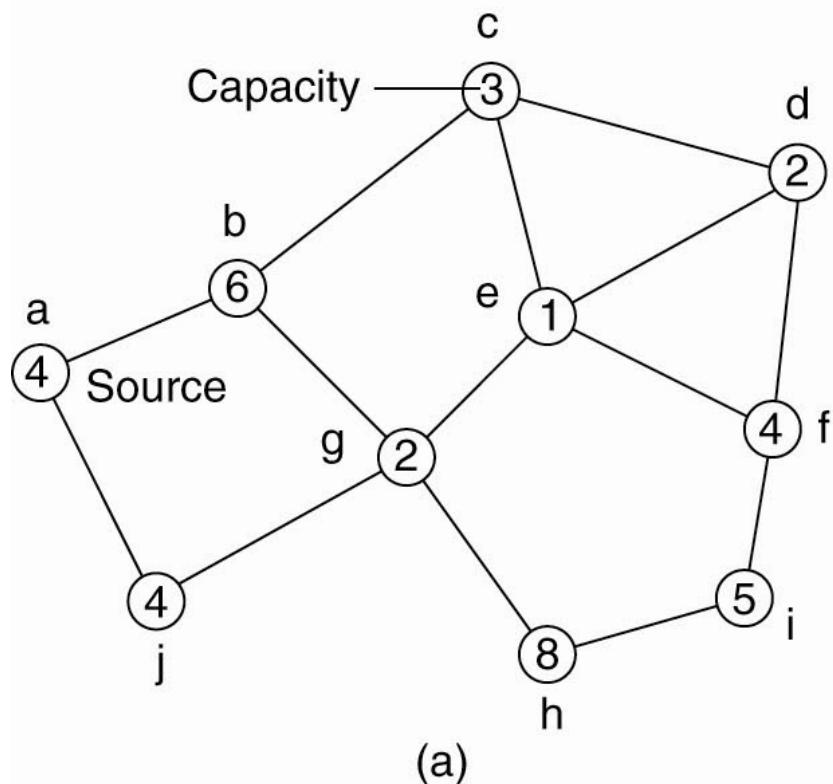
(d)

(e)

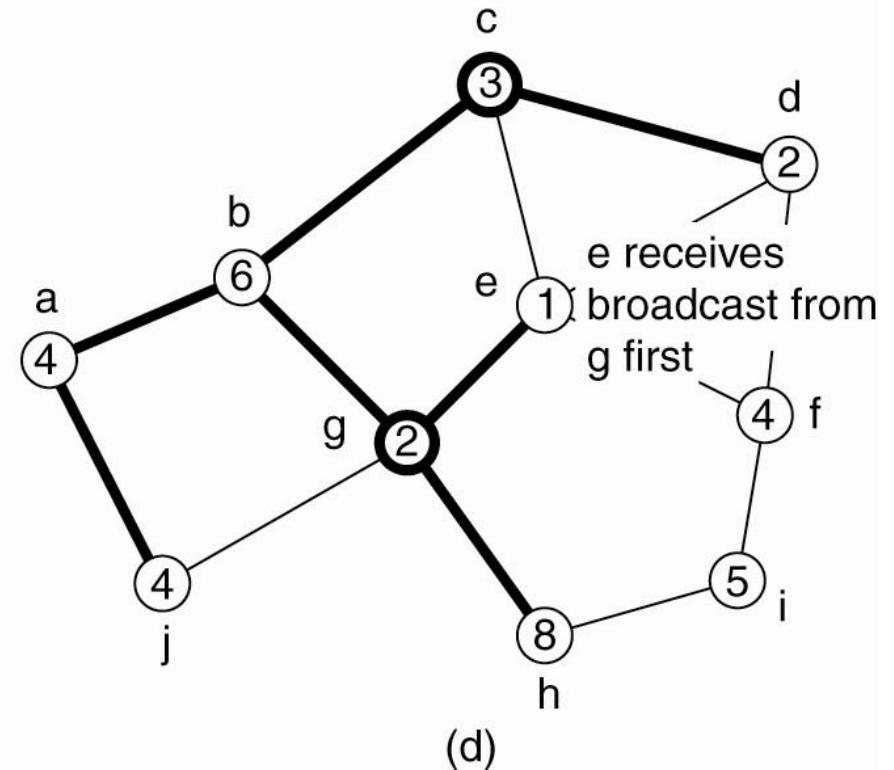
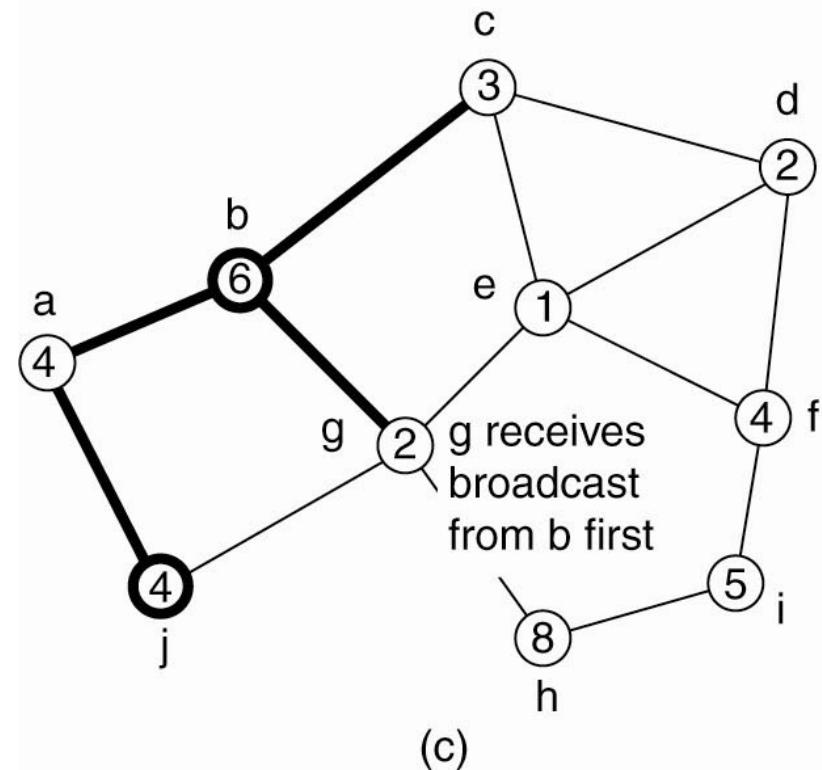
# Giải thuật RING



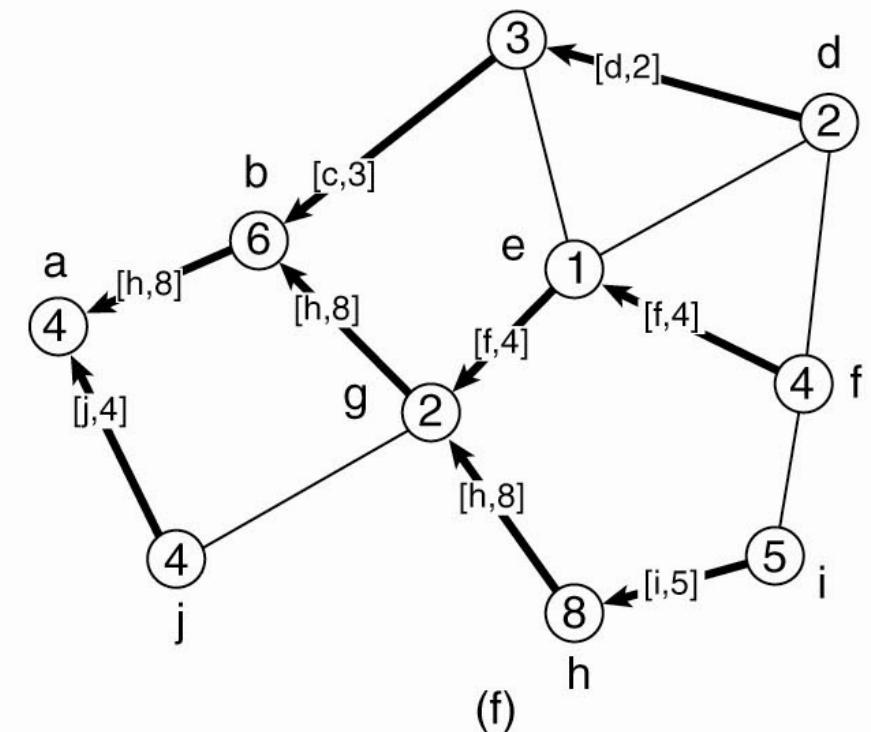
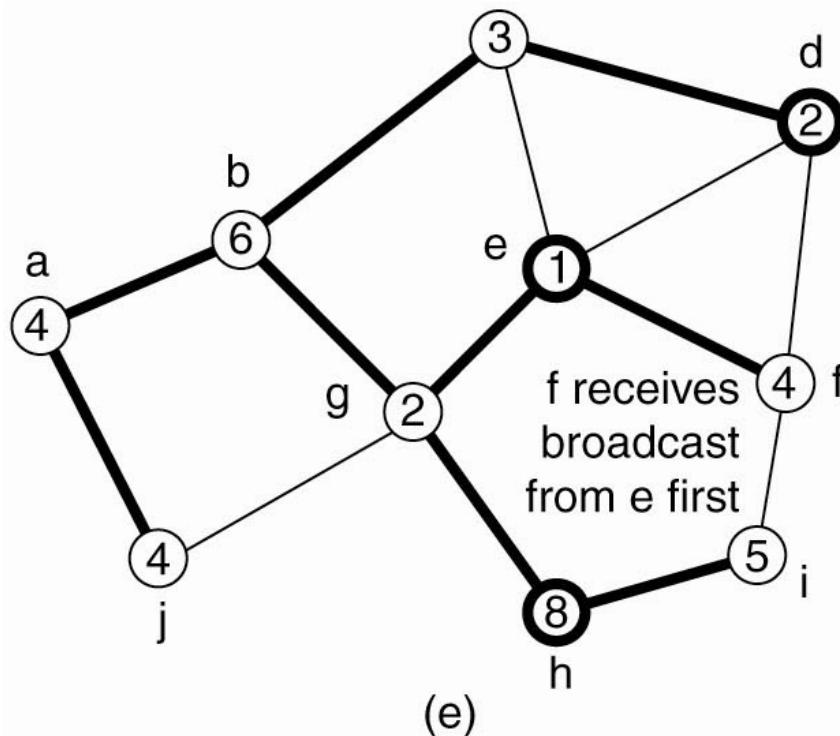
# Giải thuật bầu chọn cho mạng không dây (1)



## Giải thuật bầu chọn cho mạng không dây (2)



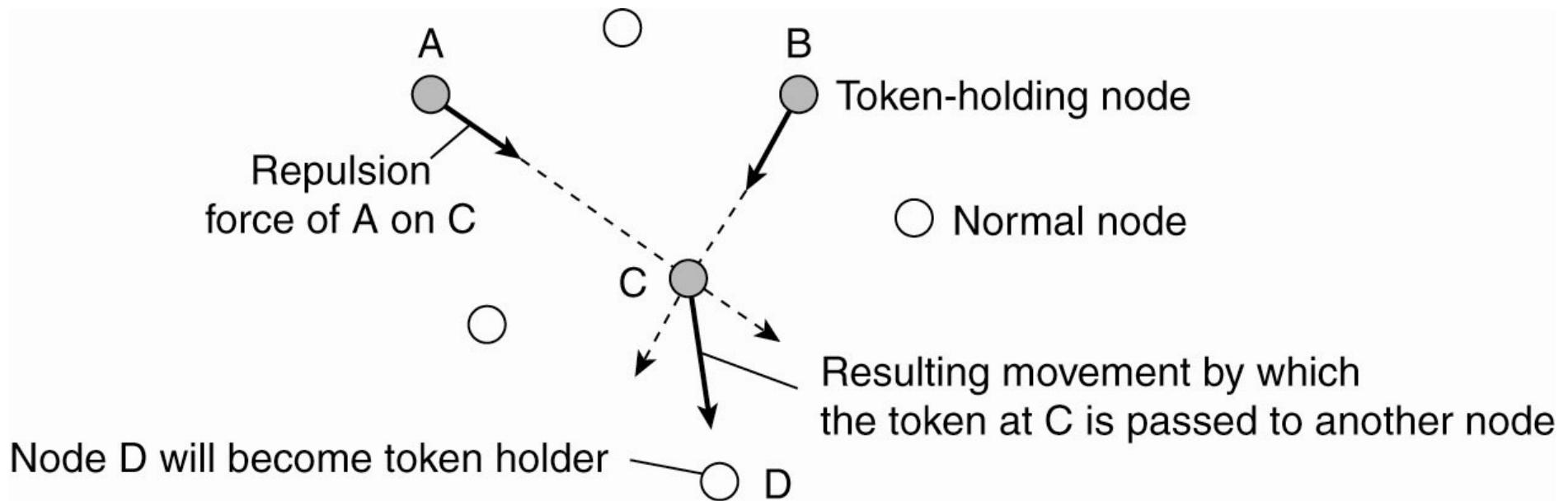
# Giải thuật bầu chọn cho mạng không dây (3)



# Bầu chọn cho mạng cỡ lớn (1)

- Yêu cầu cho các giải thuật chọn *superpeers*:
  1. Các nút thường khi truy cập đến các superpeers phải có độ trễ thấp
  2. Các superpeers phải được phân tán đều trong toàn bộ mạng overlay
  3. Phải định trước tỷ lệ các nút superpeers trong tổng số các nút trong mạng
  4. Mỗi nút superpeers không được phục vụ quá nhiều nút thường

# Bầu chọn cho mạng cỡ lớn (2)





DAI HOC  
BACH KHOA  
**25**  
YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Câu hỏi?

