

Cours 4

2.3.1.3 D'autres algorithmes

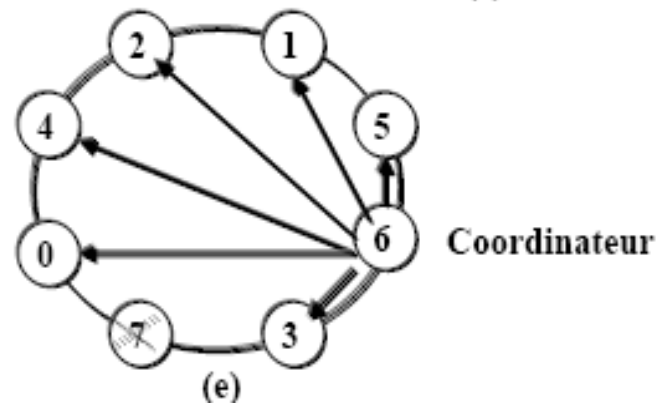
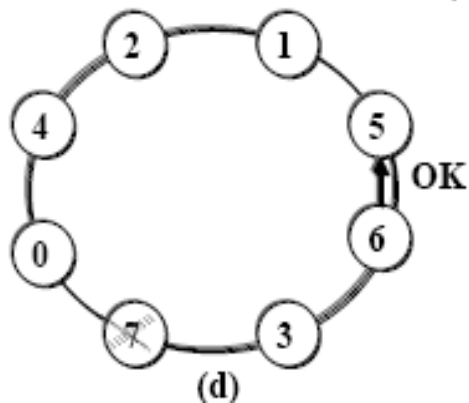
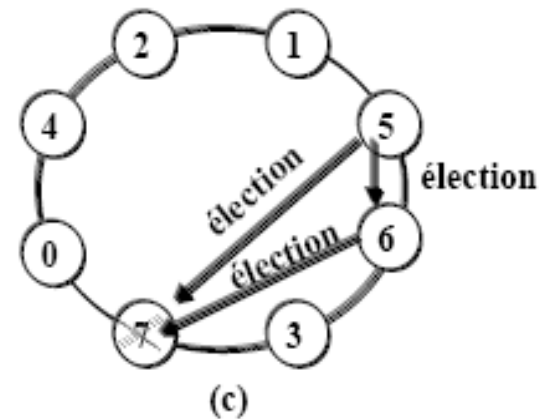
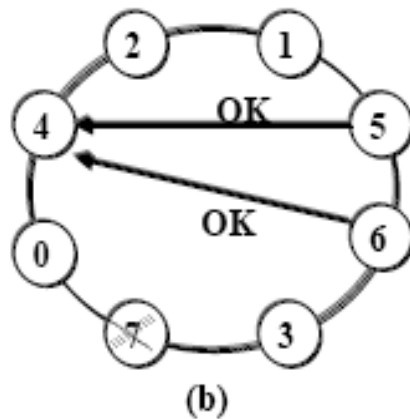
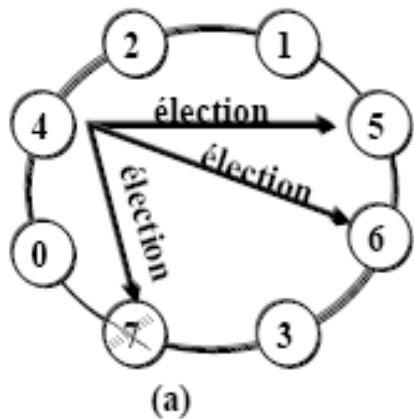
Algorithmes d'élection

1. Algorithme du plus fort

- **Hypothèses**
 - chaque processus a 1 numéro d'identification unique
 - il n'existe qu'un seul processus par machine
 - chaque processus connaît l'identité de tous les autres processus, mais ne sait pas qui est actif et qui ne l'est pas
- **But**
 - organiser une élection qui garantie que :
 - le processus élu soit actif
 - l'élection soit terminée avec l'accord de tous les processus sur l'identité du processus élu

Algorithme du plus fort

En général, les algorithmes d'élection tentent de trouver le processus ayant le numéro le plus élevé et le nomment coordinateur. Les algorithmes proposés diffèrent uniquement dans leur façon d'opérer.



Algorithmes d'élection

2. Algorithme par anneau

- **Hypothèses**
 - Processus logiquement ou physiquement ordonnés
 - Détection de la panne du coordinateur, P construit un message de type ELECTION contenant son propre numéro et l'envoie à son successeur
 - Même travail sera réalisé par les processus actifs
 - Le message revient au processus P.
 - Le message est renvoyé au réseau avec le type COORDINATEUR (le nombre de la liste dont le numéro est le plus grand)

Homework 1

- Chercher d'autres applications actuelles de l'horloge logique de Lamport.
- Quelles applications distribuées actuelles nécessitent la synchronisation et quels algorithmes utilisent-elles?
- Modern Distributed Algorithms

2.4 ETAT GLOBAL D'UN SYSTEME REPARTI : Définition

1. ETAT LOCAL EL_i D'UN SITE S_i
 1. état initial et séquence d'événements locaux sur S_i
2. ETAT LOCAL EC_{ij} D'UN CANAL C_{ij}
 1. ensemble des messages en transit sur le canal C_{ij} émis par S_i et non encore reçus par S_j
3. EVENEMENTS (ATOMIQUES) FAISANT EVOLUER LE SYSTEME:
 1. $\{EL_i\}$ événement interne sur $S_i \rightarrow \{EL'_i\}$
 2. $\{EL_i, EC_{ij}\}$ émission de m par S_i sur $C_{ij} \rightarrow \{EL'_i, EC'_{ij} = EC_{ij} \cup \{m\}\}$
 3. $\{EL_i, EC_{ki}\}$ réception de m par S_i sur $C_{ki} \rightarrow \{EL'_i, EC'_{ki} = EC_{ki} - \{m\}\}$

ETAT GLOBAL GS:

$GS = \{ \text{pour tout } i, j, \cup EL_i, \cup EC_{ij} \}$

Intérêts de la capture d'état globale d'un système distribué

- Passage dans un état particulier
- Vérification d'assertions,
- Détection terminaison, interblocage, etc),
- Effectuer des mesures de performances
- Capturer des états pouvant servir de **points de reprise** en cas de défaillances ultérieures de l'exécution ou dans le cadre d'un retour au point, etc.
- ...

Difficulté de calcul d'état global dans un système distribué

- Difficile à réaliser dans un contexte de répartition: ensemble de processus s'exécutant en parallèle et communiquant seulement par envoi de messages sur des canaux les interconnectant, pas d'horloge commun , les temps de transfert des messages ne sont pas bornés)
- Dans ces conditions, il n'est pas possible d'effectuer une observation "simultanée" de l'état des différents processus et canaux

Difficulté de calcul d'état global dans un système distribué

- L'algorithme de détection d'un état global joue le rôle d'un groupe de photographes qui observent une scène dynamique, par exemple un vol d'oiseaux migrateurs.
- La scène est si vaste qu'elle ne peut être prise par un seul photographe.
- Différents photographes doivent donc se charger de prendre chacun une partie de la scène tout en sachant que les photographies ne peuvent être prises au même instant et en exigeant que les photographes ne perturbent pas le phénomène à observer (par exemple il n'est pas raisonnable de demander aux oiseaux d'arrêter leur vol pendant les prises de vue)

Difficulté de calcul d'état global dans un système distribué

- Il faut aussi que la scène reconstituée ait “un sens”;
 - Reste à définir ce qu'est “avoir un sens” et
 - Déterminer la manière de procéder pour faire ces prises de vue.

Difficulté de calcul de l'état global dans un système distribué

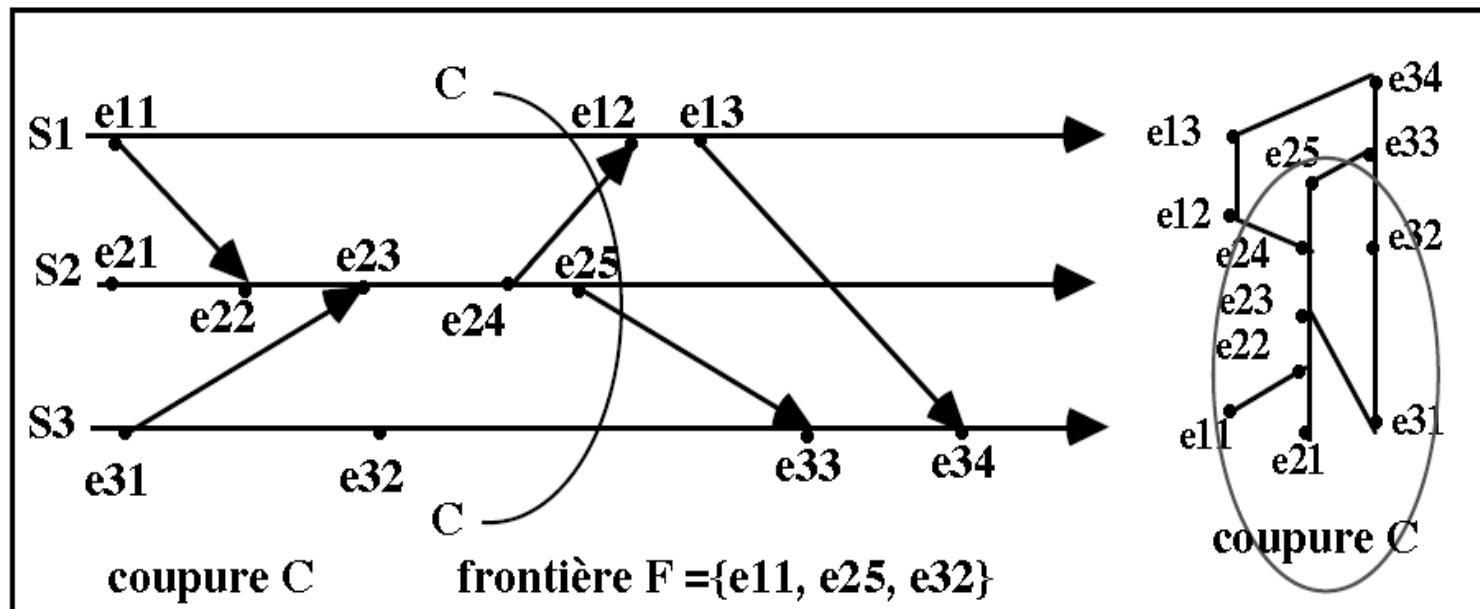


Définitions..

- DIFFICULTES
 - E_i n'est immédiatement observable que sur S_i
 - E_{Cij} n'est jamais directement observable, ni sur S_i , ni sur S_j
- OBJECTIF :
 - définir un état global déterminable localement par tout site
- REMARQUE :
 - toute définition d'état doit respecter la dépendance causale
- PASSÉ D'UN ÉVÉNEMENT
 - $\text{hist}(e) = e \cup$ ensemble des événements e' tels que $e' \rightarrow e$
- COUPURES
 - soit E un ensemble d'événements constituant une application répartie
 - Coupure = sous-ensemble fini C de E tel que pour $a, b \in E$, $a \in C$ et (b précède localement a) $\Rightarrow (b \in C)$

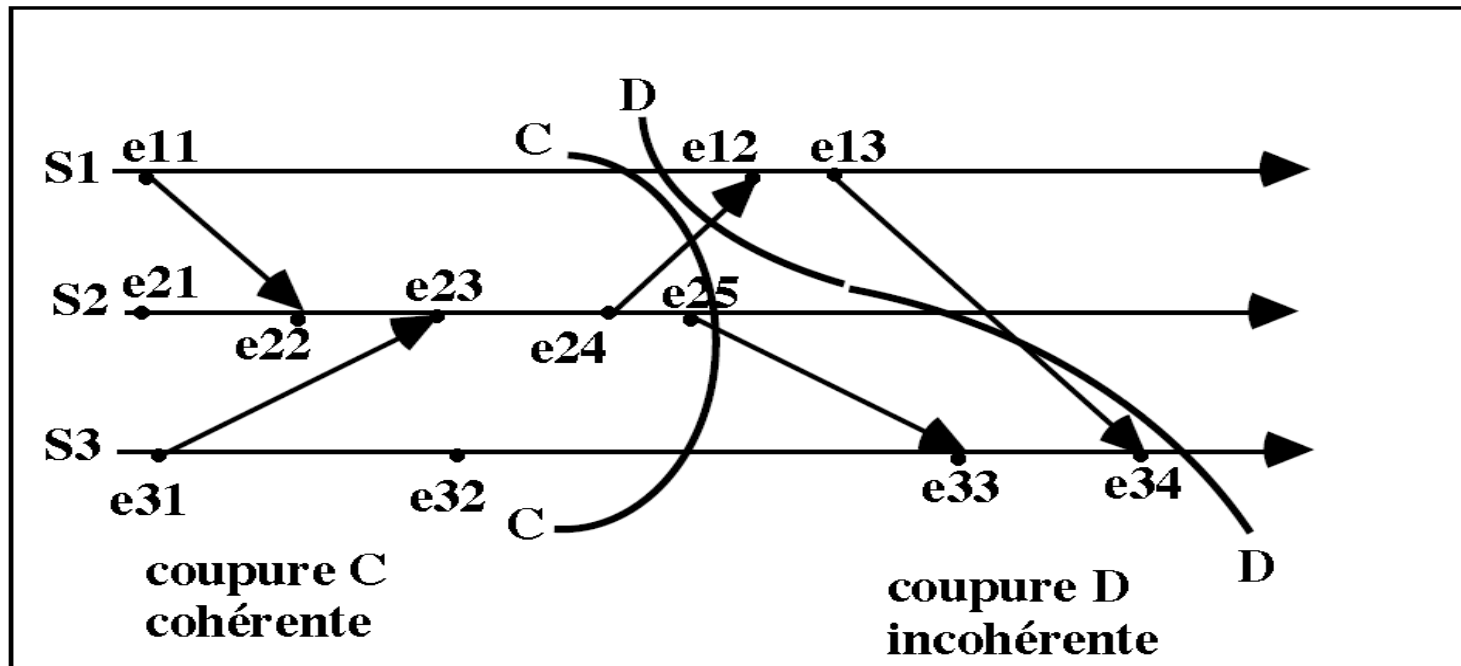
Définitions, cont..

- Frontière **F** d'une coupure **C** :
 - ensemble des événements les plus récents de la coupure, un événement par site, $a \in F \iff a \in C$ **et** il n'existe pas de $b \in F$ tel que $a \rightarrow b$



COUPURES COHERENTES

- Cohérence $\Leftrightarrow$ respect de la causalité dans la coupure
- $a \in C$ et $(b \rightarrow a) \Rightarrow (b \in C)$

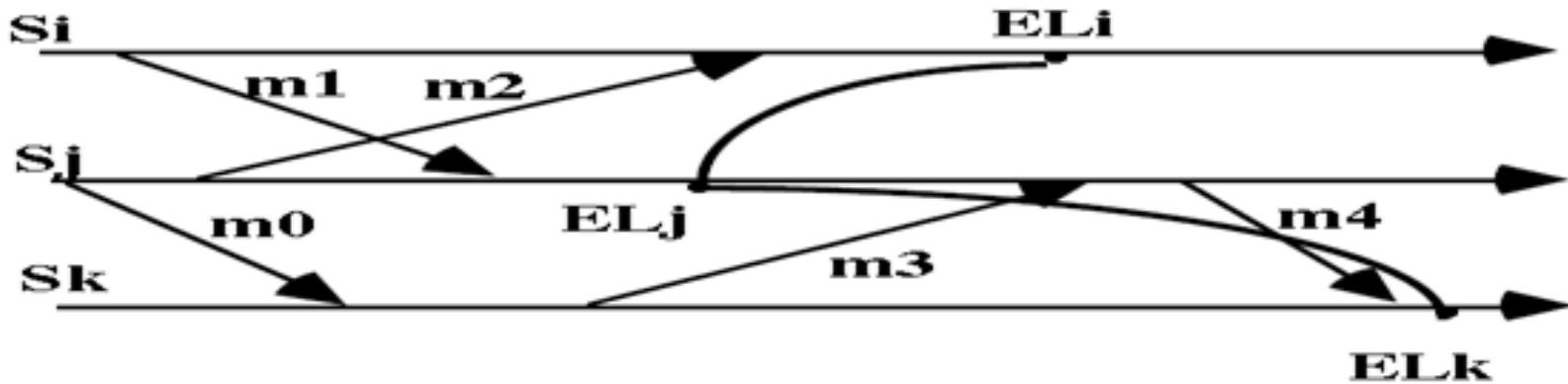


Etat global cohérent = état associé à une coupure cohérente

Cont..

- La frontière de la coupure associée définit un passé et un futur
- Il en résulte deux conditions nécessaires pour les messages m qui traversent une frontière de coupure.
- **condition C1 :**
 - si $\text{EMISSION}(m) \in \text{EL}_i$ alors
 - soit $\text{RECEPTION}_j(m) \in \text{EL}_j$,
 - soit $m \in \text{Ec}_{ij}$
- **Cela veut dire : tout message émis dans le passé est soit reçu dans le passé soit en transfert**
- **condition C2 :**
 - si $\text{EMISSION}_i(m) \notin \text{EL}_i$ alors $\text{RECEPTION}_j(m) \notin \text{EL}_j$
- **Cela veut dire : tout message émis dans le futur reste dans le futur**

Exemple (Exécution repartie)



état global $\{EL_i, EL_j, EL_k\}$

- 1• $EMISSION_k(m_3) \in EL_k$ mais
comme $RECEPTION_j(m_3) \notin EL_j$,
il faut que $m_3 \in ECK_j$
donc la condition C1 est vraie seulement si $m_3 \in ECK_j$
- 2• $EMISSION_j(m_4) \notin EL_j$ mais $RECEPTION_k(m_4) \in EL_k$
la condition C2 est fausse car
C2 : si $EMISSION_j(m_4) \notin EL_j$ alors $RECEPTION_k(m_4) \notin EL_k$

conclusion

$\{EL_i, EL_j, EL_k\}$ n'est pas un état global cohérent

DETERMINATION D'UN ETAT GLOBAL **COHERENT**

(Algorithme de Chandy - Lamport 1985)

- Hypothèses
 - le réseau de communication est connexe
 - les canaux respectent l'ordre d'émission des messages (canaux FIFO)
 - un site "élu" particulier lance la détermination d'état global
- Principe
 - association d'un message **marqueur** à chaque enregistrement d'état local d'un site pour que les autres sites puissent repérer les messages qui sont avant ou après cet enregistrement (les messages du passé et du futur)
 - chaque site détermine son état local et celui de ses canaux en réception, puis **envoie ces états au site "élu"**
- Propriétés
 - l'algorithme se termine
 - **l'état global enregistré correspond à une coupure cohérente**
 - les états enregistrés des canaux sont corrects pour cette coupure

SOLUTION DE CHANDY et LAMPORT (1985)

- HYPOTHÈSE : CANAUX FIFO
 - tout message mk envoyé sur un canal FIFO sépare les messages du canal en deux sous-ensembles : $<mk$ (**avant** mk) et $>mk$ (**après** mk).
- CONTRAINTES DE COHÉRENCE
 - **Condition C1 et C2**
- Mise en œuvre
 - Dès que S_i enregistre EL_i , il émet un message marqueur mk sur chaque canal C_{ij} .
 - S_j doit enregistrer EL_j au plus tard à la réception de mk et S_j doit capter tous les messages $<mk$, soit dans EL_j soit dans C_{ij} (condition C1).
 - Les messages $>mk$ ne doivent pas être captés car ils viennent du futur de EL_i sur S_i (condition C2).

SOLUTION DE CHANDY et LAMPORT (1985)
ALGORITHME SUR CHAQUE SITE S_i

- Début (S_i "élu") ou première réception d'un marqueur mk (émis par S_j) :
 - (1) enregistrer EL_i
 - (2) enregistrer EC_{ji} comme vide car S_i a déjà reçu tous les messages $<mk$ et ceux de $>mk$ ne font pas partie de l'état global.
 - Au démarrage sur "élu", aucun EC_{ji} n'est enregistré. (Juste une remarque)
 - (3) diffuser mk à tous ses voisins sur les canaux C_{ij}
 - (1-2-3) doit être atomique
- Réceptions suivantes du marqueur mk (émis par S_j)
- Enregistrer EC_{ji} comme constitué des messages reçus par S_i , entre l'enregistrement de EL_i et l'arrivée de mk envoyé par P_j .
- L'algorithme s'arrête lorsque chaque site reçoit mk sur tous ses canaux d'entrés.

Jean-Michel Helary, Achour Mostefaoui, Michel Raynal. Déterminer un état global.
[Rapport de recherche] RR-2090, INRIA. 1993. <inria-00074582>

- Il est intéressant de remarquer que, dans un système disposant d'une horloge physique globale, le problème serait facilement résolu: il suffirait de décider d'une date d à laquelle tous les processus du système enregistreront leur état local. L'identification des messages en transit se ferait aisément en associant à chaque message la date de son émission. Ceux qui sont émis avant la date d et reçus après la date d sont les messages en transit et sont donc enregistrés dans l'état du canal c_{ij} par P_j . Dans un système réparti à canaux fifos, les messages m_k servent à simuler l'occurrence de la date ' d ', et délimitent sur chaque canal les messages émis avant et après cette date.

From Ozalp Babaoğlu Keith Marzullo. Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms.

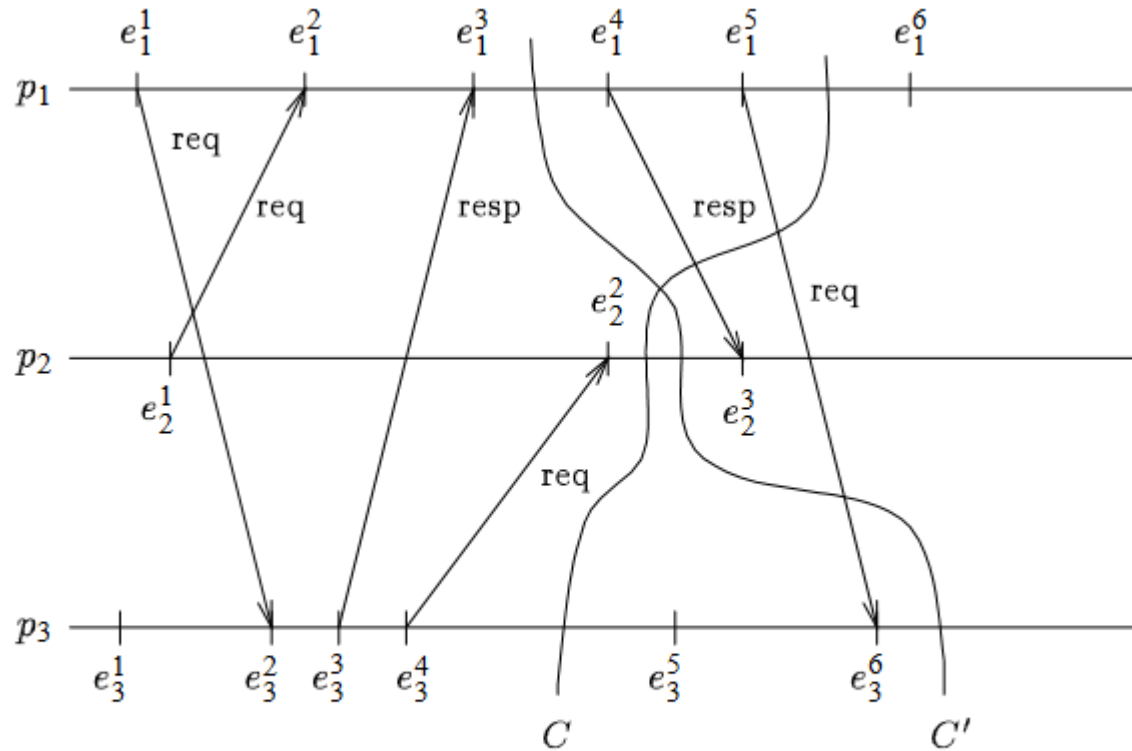


Figure 2. Cuts of a Distributed Computation

From "Ozalp Babaoğlu Keith Marzullo. Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms.

- 6 Consistency
- In other words, a consistent cut is left closed under the causal precedence relation. In its graphical representation, verifying the consistency of a cut becomes easy: if all arrows that intersect the cut have their bases to the left and heads to the right of it, then the cut is consistent; otherwise it is inconsistent. According to this definition, cut C of Figure 2 is consistent while cut C' is inconsistent
- A consistent global state is one corresponding to a consistent cut.

From "Ozalp Babaoğlu Keith Marzullo. Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms.

- 13 Distributed Snapshots : We will now develop a snapshot protocol that constructs only consistent global states. The protocol is due to Chandy and Lamport [5], and the development described here is due to Morgan [26].