

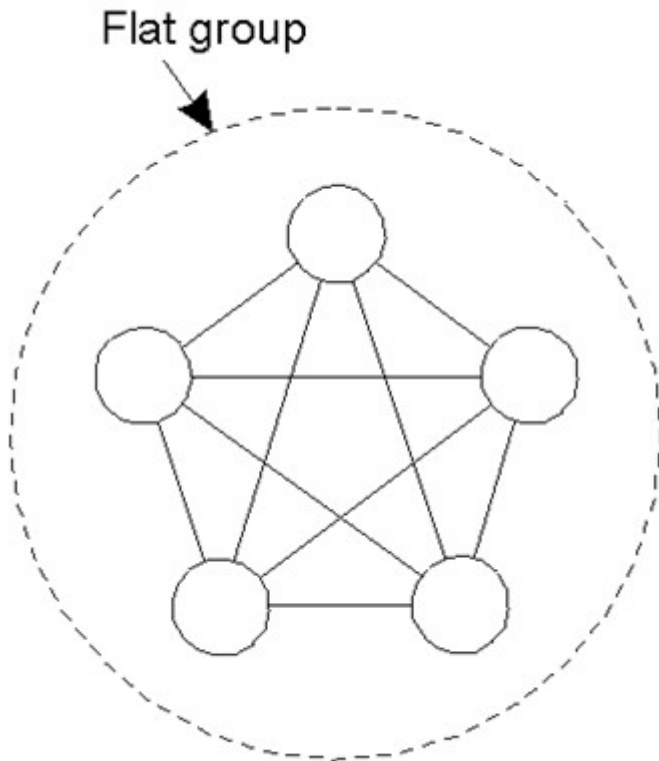
12. előadás

Hibatűrés

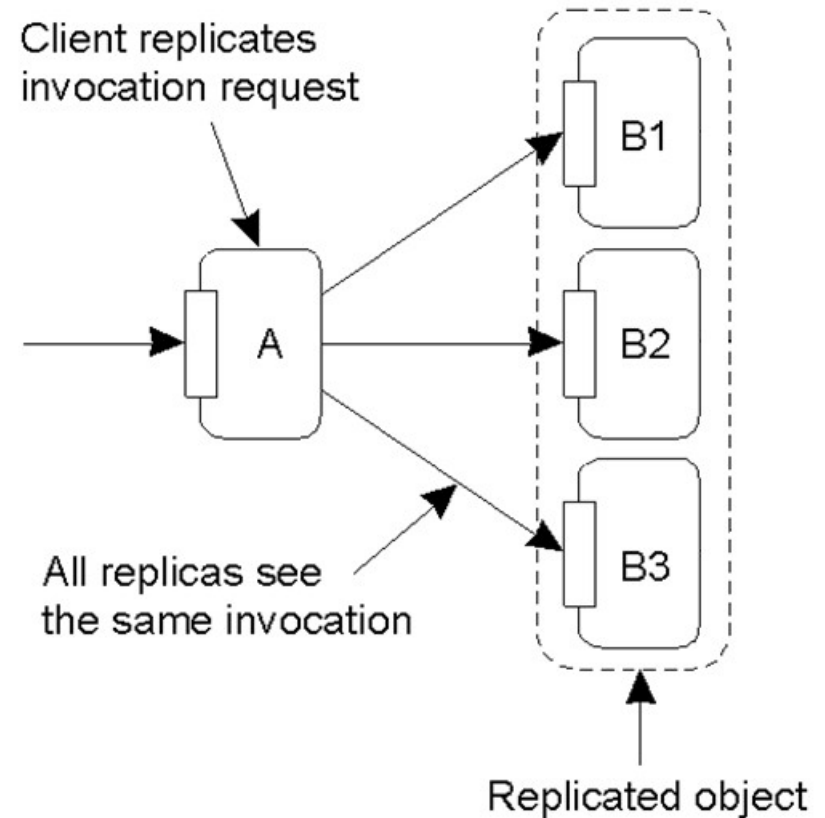
2. rész

Hibatűrés fokozása redundanciával

- Megbízhatatlan folyamatok



- Többszörözés megvalósítása



Megbízható csoportkommunikáció (1)

Megbízható többcímű átvitel:

a csoportnak küldött üzenet a csoport valamennyi tagjához eljut.

Átviteli réteg – megbízható pont-pont csatornákat kínál.

- Minden folyamat helyesen üzemel
- Hibás folyamatok is vannak

Megbízható csoportkommunikáció (2)

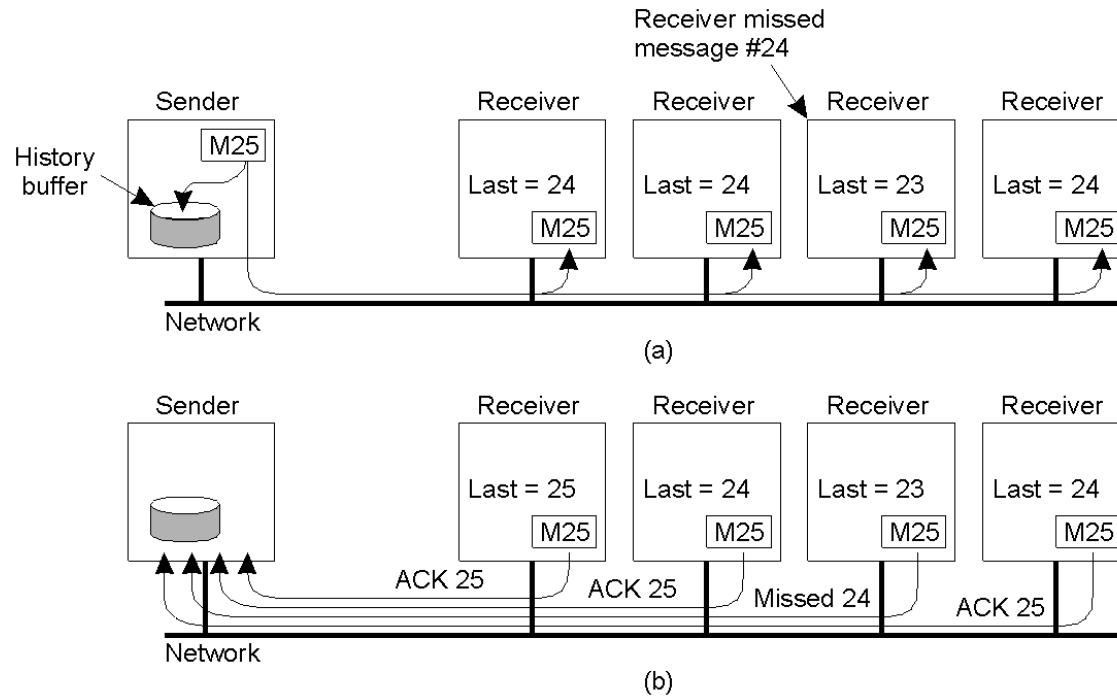
Tfh. Minden folyamat helyesen üzemel + egyezség van a csoport tagságokban

=> az üzenetet mindenkinek el kell küldeni

- A feladó minden többcímű üzenethez sorszámot rendel
- A fogadó
 - Sikeres fogadás esetén nyugta és/vagy
 - felismeri, ha valami hiányzik negatív nyugtát küld

Alapvető megbízható csoportcímű sémák

(1)



A megbízható csoportcímezés egyszerű megvalósítása, ha minden címzett ismert és egyikről sem feltételezhető a meghibásodás

- a) Üzenet átvitele
- b) Visszajelzés küldése

Alapvető megbízható csoportcímű sémák (1)

Feladónak tárolnia kell az üzenetet.

Hátrányok:

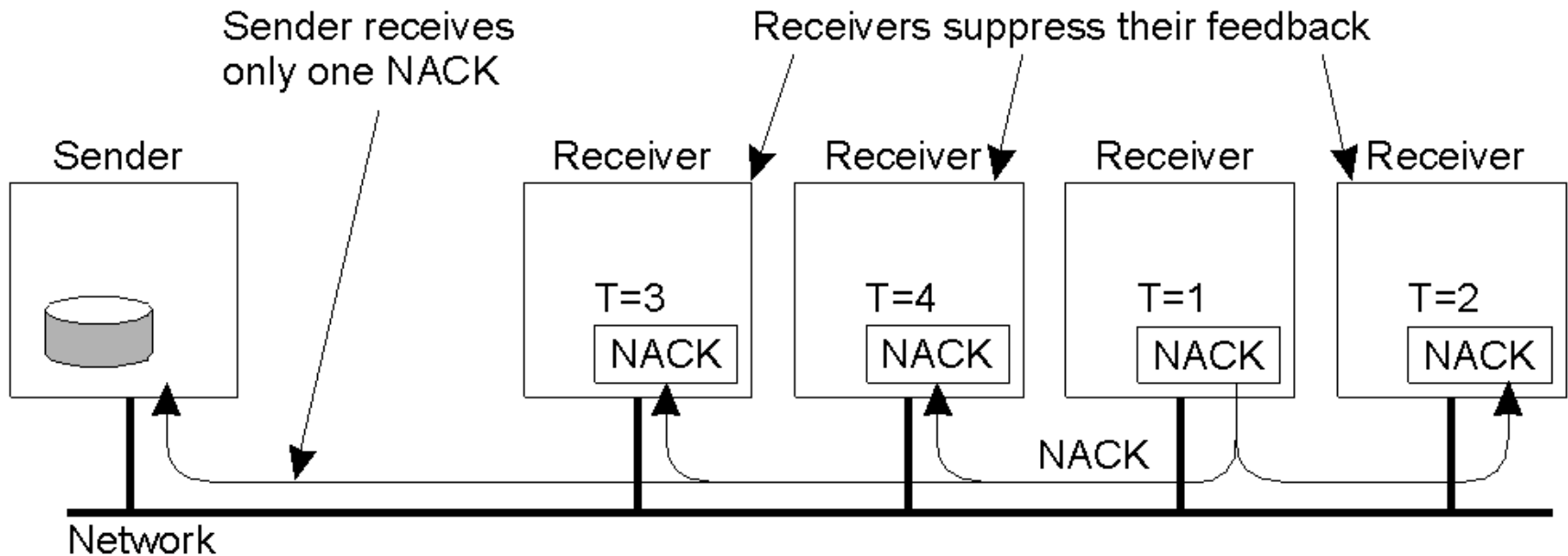
- Sikeres fogadás esetén nyugta – visszajelzés áradat
- Csak negatív nyugta van – feladónak örökre tárolnia kell, vagy nem megbízható az átvitel

Nem hierarchikus visszajelzés-vezérlés

(1)

- Átméretezhető megoldás - visszajelzés-elfojtás
- SRM – átméretezhető megbízható többcímű üzenetküldés (Scalable Reliable Multicasting)
- Siker esetén nincs nyugta
- Negatív nyugta a csoport összes tagjának (véletlenszerű késleltetéssel)
- Ha negatív nyugtát akar küldeni, de közben kap mástól => elfojtás

Nem hierarchikus visszajelzés-vezérlés (2)



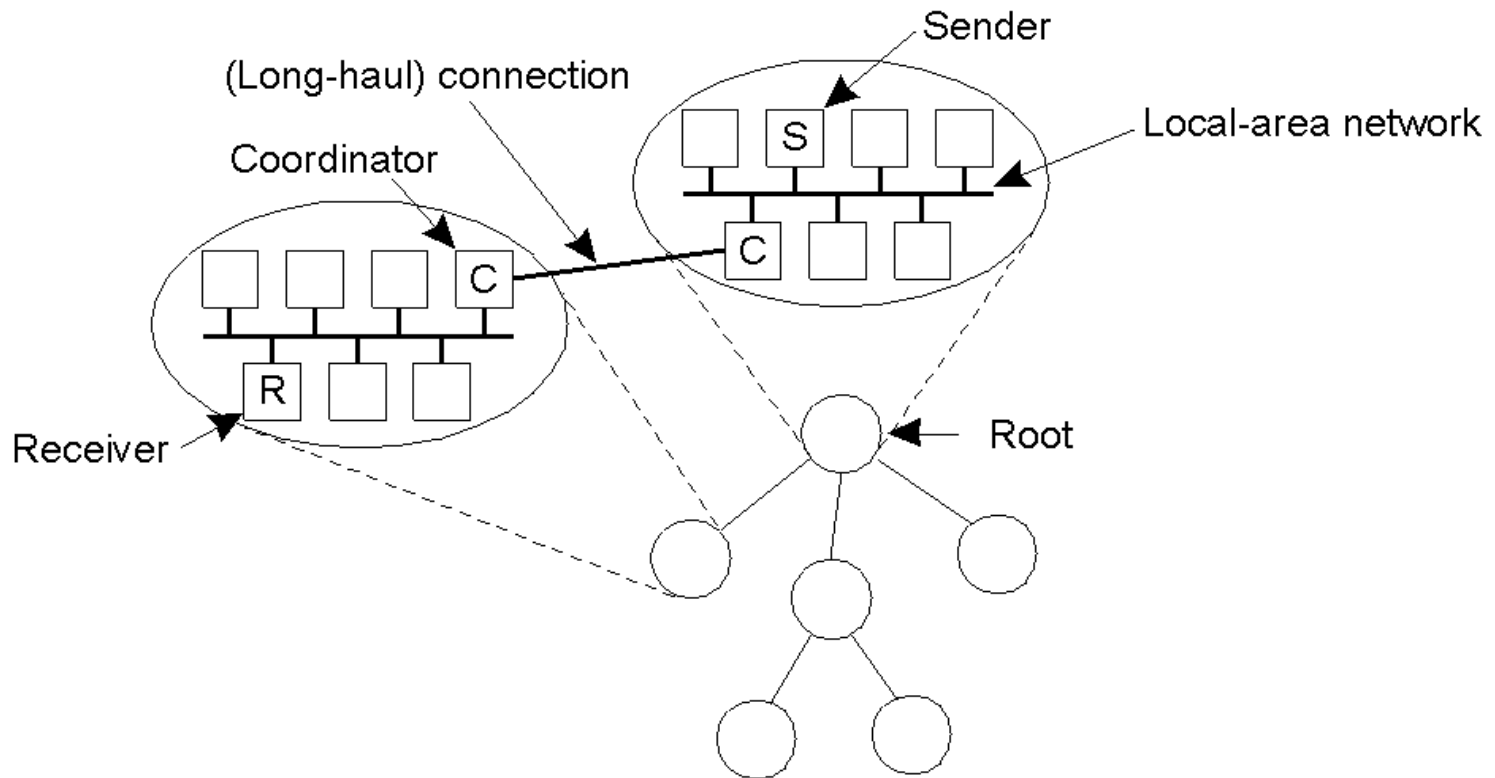
Több címzett is beütemezte az újraküldési kérést, de az első újraküldési kérés a többi elfojtásához vezet

Nem hierarchikus visszajelzés-vezérlés

(3)

- Jól átméretezhető
- Hátrányok:
 - folyamatok összehangolásának szükségessége
 - haszontalan üzenetek fogadása és feldolgozása

Hierarchikus visszajelzés-vezérlés



A megbízható csoportcímezés hierarchikus megvalósítása.

- a) Valamennyi helyi koordinátor az általa felügyelt gépeknek továbbítja az üzeneteket, és
- b) kezeli a későbbi újraküldési kéréseket

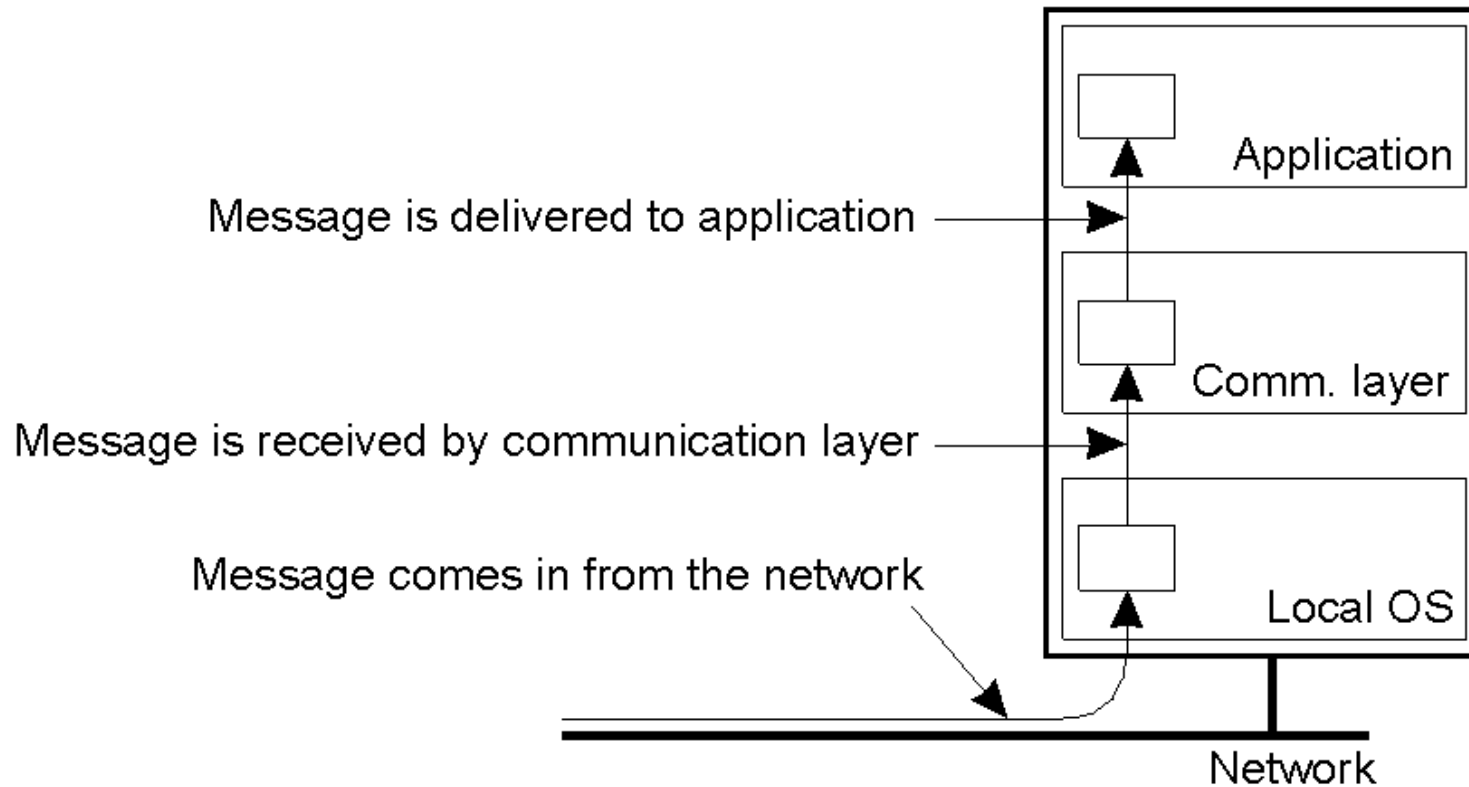
Elemi csoportcímezés

A folyamatok meghibásodhatnak

=> az üzeneteknek valamennyi működő folyamathoz el kell jutniuk (esetleg ugyanabban a sorrendben), vagy egyikhez sem (ha a feladó összeomlik a küldés során)

- Üzenet érkezése
- Üzenet fogadása

Látszólagos szinkronitás (1)

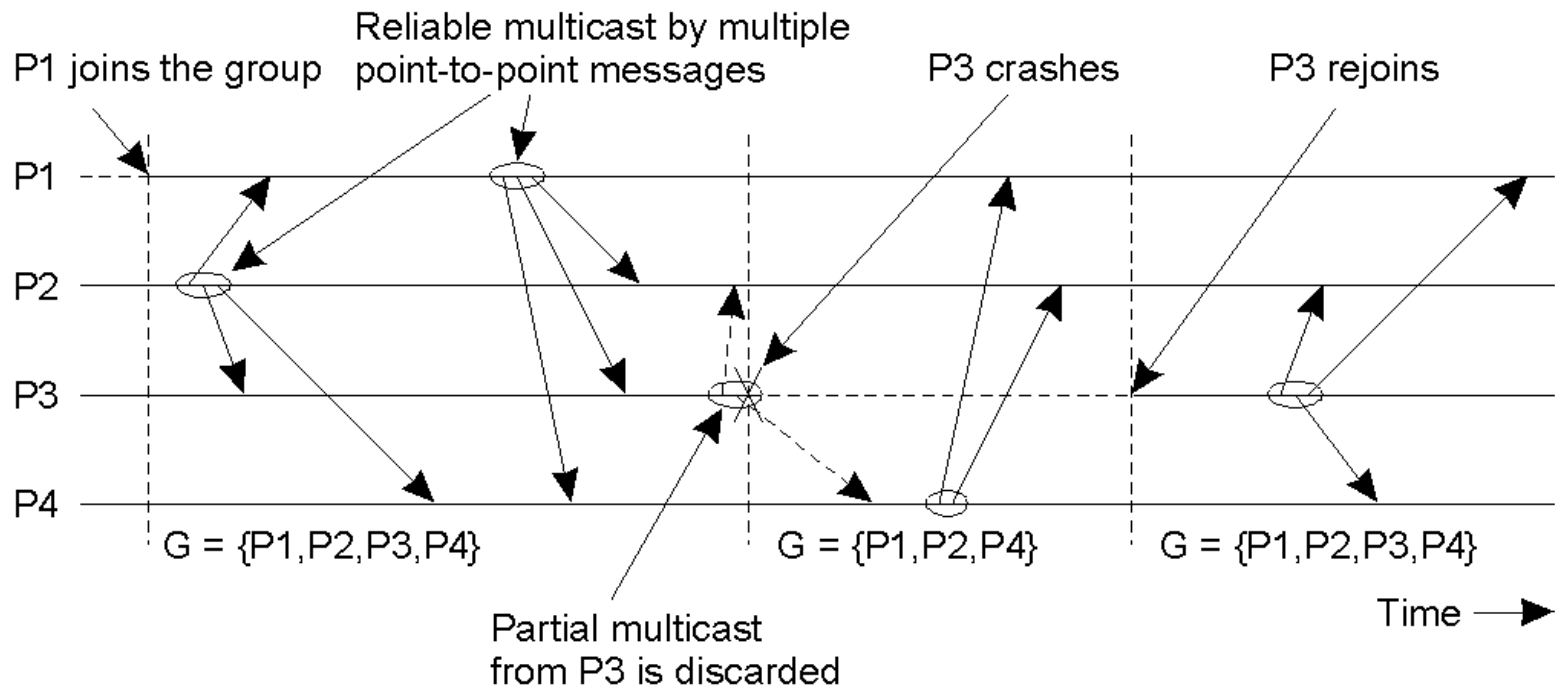


Az üzenet érkezését és kézbesítését megkülönböztető elosztott rendszer logikai felépítése.

Látszólagos szinkronitás (2)

- **Látszólagos szinkronitás:** ha a küldő összeomlik, akkor az üzenetet vagy minden fogadónak meg kell kapnia, vagy egyiknek sem
- **Csoportkép** – kézbesítése lista (minden üzenethez – küldő ismerete)
- **Látványváltozás** – üzenet, ha változik a csoportkép -- határ, amit egy üzenet sem léphet át

Látszólagos szinkronitás (3)



A látszólagosan szinkronizált többcímű üzenetküldés elve.

Üzenetek sorbarendezése

1. Rendezetlen üzenetküldés
2. FIFO-sorrendű üzenetküldés
3. Okozatilag sorbarendezett üzenetküldés
4. Pontosán sorbarendezett többcímű üzenetküldés

Megbízható, rendezetlen csoportcímezés

P1 folyamat

elküldi m1 üzenetet

elküldi m2 üzenetet

P2 folyamat

megkapja m1-et

megkapja m2-t

P3 folyamat

megkapja m2-t

megkapja m1-et

Egy csoport három, egymással kommunikáló folyamata. Az események folyamatonkénti sorrendjét a függőleges tengelyen ábrázoljuk.

Megbízható FIFO-sorrendű csoportcímezés

P1 folyamat

m1 elküldése

m3 fogadása

m2 elküldése

m4 fogadása

P2 folyamat

m1 fogadása

m3 fogadása

m2 fogadása

m4 fogadása

P3 folyamat

m3 fogadása

m1 fogadása

m2 fogadása

m4 fogadása

P4 folyamat

m3 elküldése

m1 fogadása

m2 fogadása

m4 elküldése

A csoport négy folyamata két feladó esetében, és a FIFO-sorrend által megengedett két lehetséges üzenetkézbesítési sorrend.

Üzenetek sorbarendezése

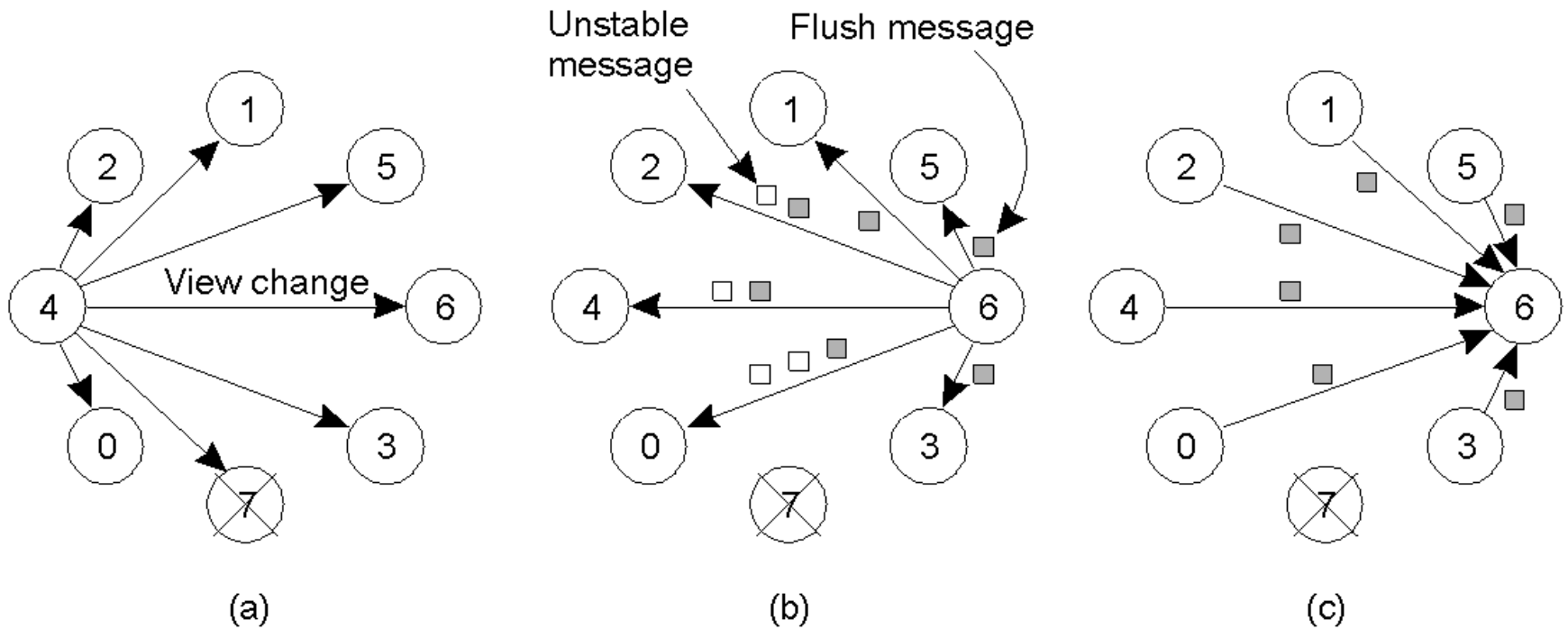
Csoportcímzés	Alapvető üzenetrendezés	Pontosan rendezett kézbesítés?
Megbízható csoportcímzés	Nincs	Nem
FIFO-csoportcímzés	FIFO-sorrendű kézbesítés	Nem
Okozati csoportcímzés	Okozati sorrendű kézbesítés	Nem
Elemi csoportcímzés	Nincs	Igen
FIFO elemi csoportcímzés	FIFO-sorrendű kézbesítés	Igen
Okozati elemi csoportcímzés	Okozati sorrendű kézbesítés	Igen

A látszólagosan szinkronizált megbízható csoportcímzés hat különböző változata.

Látszólagos szinkronitás implementálása (1)

- Üzenetküldés mindenkinek megbízható pont-pont csatornán (pl. TCP)
- Ha a feladó összeomlott: a hiányzó üzenetet egymástól is beszerezhetik
- **Stabil üzenet:** minden folyamathoz megérkezett
 - csak ezeket lehet kézbesíteni
- **Üritési üzenet:** felkészült az új csoportkép installálására

Látszólagos szinkronitás implementálása (2)

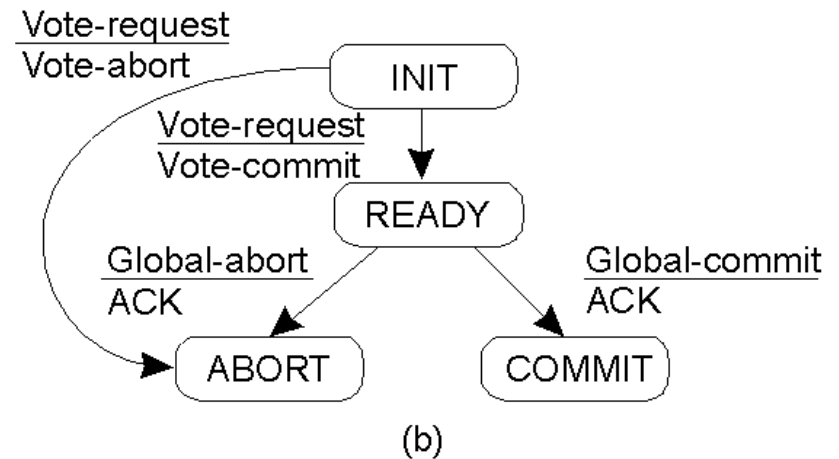
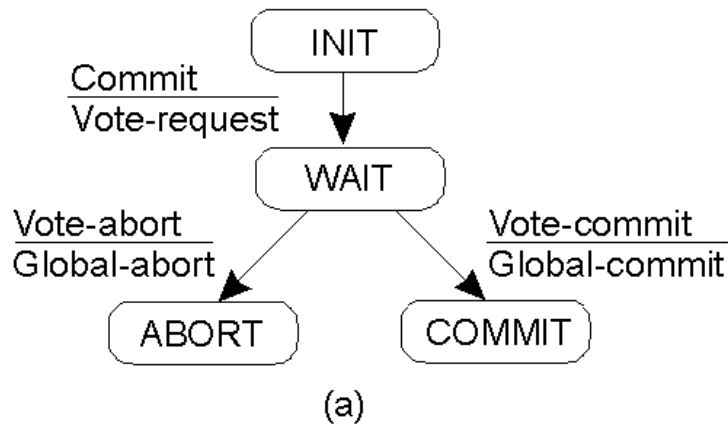


- a) A 4. folyamat észreveszi, hogy a 7-es összeomlott, ezért egy csoportképváltozási üzenetet küld.
- b) A 6. folyamat kiküldi valamennyi nem stabil üzenetét, majd egy ürítési üzenetet.
- c) A 6. folyamat akkor installálja az új csoportképet, amikor a többi folyamattól ürítési üzenetet kapott

Elosztott véglegesítés

- Az adott műveletet a csoport összes tagja hajtsa végre, vagy egyikük sem (az elemi csoportcímzési probléma az elosztott véglegesítés speciális esete – művelet: üzenet kézbesítése)
- **Egyfázisú véglegesítési protokoll:** koordinátor segítségével

Kétfázisú véglegesítés (1)



- a) A 2PC-koordinátort jelképező véges állapotgép.
- b) A résztvevőt jelképező véges állapotgép.

Kétfázisú véglegesítés (2)

- Első fázis: szavazás
- Második fázis: döntés
- Probléma meghibásodás esetén – blokkoló hívások:
 - résztvevő / INIT: megoldás: timeout
 - koordinátor / WAIT: megoldás: timeout
 - Résztvevő / READY:
 - => megvárja, míg újraindul a koordinátor
 - VAGY lekérdezi másik folyamattól

Kétfázisú véglegesítés (3)

Q állapota	P cselekedete
COMMIT	Átmegy COMMIT állapotba
ABORT	Átmegy ABORT állapotba
INIT	Átmegy ABORT állapotba
READY	Másik résztvevőt vizsgál meg

A *READY* állapotba beragadt *P* résztvevő által végrehajtható műveletek *Q* résztvevő állapotának megvizsgálása után.

Kétfázisú véglegesítés (4)

- A felépülés lehetőségéhez az állapotot menteni kell
- Koordinátor:
 - 2PC-protokoll megkezdése
 - döntés meghozása
- Résztvevők:
 - INIT, COMMIT, ABORT
 - READY

Kétfázisú véglegesítés (5)

A koordinátor lépései:

```
while START_2PC to local log;
multicast VOTE_REQUEST to all participants;
while not all votes have been collected {
    wait for any incoming vote;
    if timeout {
        while GLOBAL_ABORT to local log;
        multicast GLOBAL_ABORT to all participants;
        exit;
    }
    record vote;
}
if all participants sent VOTE_COMMIT and coordinator votes COMMIT{
    write GLOBAL_COMMIT to local log;
    multicast GLOBAL_COMMIT to all participants;
} else {
    write GLOBAL_ABORT to local log;
    multicast GLOBAL_ABORT to all participants;
}
```

A koordinátor által végrehajtott lépések vázlata a kétfázisú protokollnál.

Kétfázisú véglegesítés (6)

A résztvevők lépései:

```
write INIT to local log;
wait for VOTE_REQUEST from coordinator;
if timeout {
    write VOTE_ABORT to local log;
    exit;
}
if participant votes COMMIT {
    write VOTE_COMMIT to local log;
    send VOTE_COMMIT to coordinator;
    wait for DECISION from coordinator;
    if timeout {
        multicast DECISION_REQUEST to other participants;
        wait until DECISION is received; /* blokkolva marad */
        write DECISION to local log;
    }
    if DECISION == GLOBAL_COMMIT
        write GLOBAL_COMMIT to local log;
    else if DECISION == GLOBAL_ABORT
        write GLOBAL_ABORT to local log;
} else {
    write VOTE_ABORT to local log;
    send VOTE_ABORT to coordinator;
}
```

A résztvevők
lépés-
sorozatának
vázlata a
2PC-
eljárásban.

Kétfázisú véglegesítés (7)

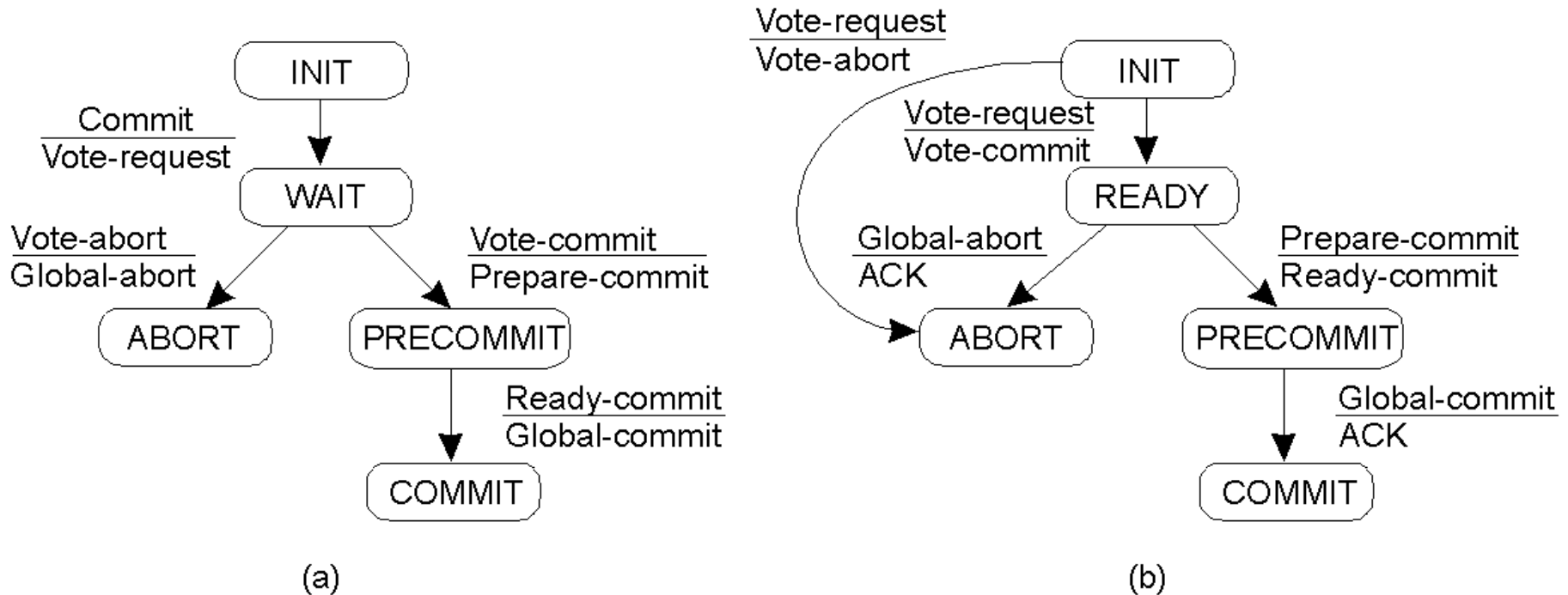
```
Akciók a döntési kérelmek kezeléséhez: /* végrehajtva elkülönített szálon */  
while true {  
    wait until any incoming DECISION_REQUEST is received; /* blokkolva marad */  
    read most recently recorded STATE from the local log;  
    if STATE == GLOBAL_COMMIT  
        send GLOBAL_COMMIT to requesting participant;  
    else if STATE == INIT or STATE == GLOBAL_ABORT  
        send GLOBAL_ABORT to requesting participant;  
    else  
        skip; /* résztvevő blokkolva marad */
```

A bejövő döntési kérések lépéssorozatának vázlata.

Háromfázisú véglegesítés (1)

- Megakadályozza a folyamatok blokkolódását megállási meghibásodások esetében
- Nincs egyetlen olyan állapot sem, amelyből közvetlen átmenet lenne akár a COMMIT, akár az ABORT állapotba
- Nincs olyan állapot, amelyben ne lenne lehetséges a végső döntés elérése, miközben közvetlen átmenet van a COMMIT állapotba

Háromfázisú véglegesítés (2)



- a) A 3PC koordinátorának véges állapotgépe.
- b) A résztvevő véges állapotgépe.