

Radiado

Ernesto Jiménez

Prometeo

Índice

1. Mensajes.
2. Cliente-servidor
3. Calidades de servicio de multicast
4. Garantía de entrega en multicast
5. Garantía de orden en multicast
6. Servicio de membresía de grupos

Bibliografía

- *Distributed Systems: Concepts and Design.*
Dollimore, Colouris, Kindberg. 4th Edition. Addison
Wesley. Chpt. 12.

1. Mensajes

- Primitivas de envío y recepción de mensaje:
 - ✓ enviar (datos, destino)
 - ✓ recibir (datos, origen)
- Si un proceso quiere recibir de cualquiera:
 - ✓ recibir(mensaje)
- En el mensaje: los datos y el identificador del emisor.
- Las primitivas pueden tener diferentes semánticas:
 - ✓ Envío no bloqueante de mensajes,
 - ✓ Envío bloqueante de mensajes.

1. Mensajes

- Paso no bloqueante de mensajes
 - ✓ En paso no bloqueante de mensajes el emisor deja el mensaje en un buffer del emisor o del receptor y continúa sin bloquearse.

Fallos

- ✓ Si falla la red, el emisor puede recibir una excepción si el protocolo subyacente usa temporizadores.
- ✓ Con fallos de nodo tipo crash, los mensajes de los buffers desaparecen.
- ✓ En ambos casos, el proceso no sabe si el mensaje llegó a enviarse, ni si fue recibido, ni si fue procesado.

1. Mensajes

- Paso bloqueante de mensajes
 - ✓ El emisor se bloquea hasta que el receptor haga un recibir del mensaje.
 - ✓ Con este tipo de paso no es necesario buffer.

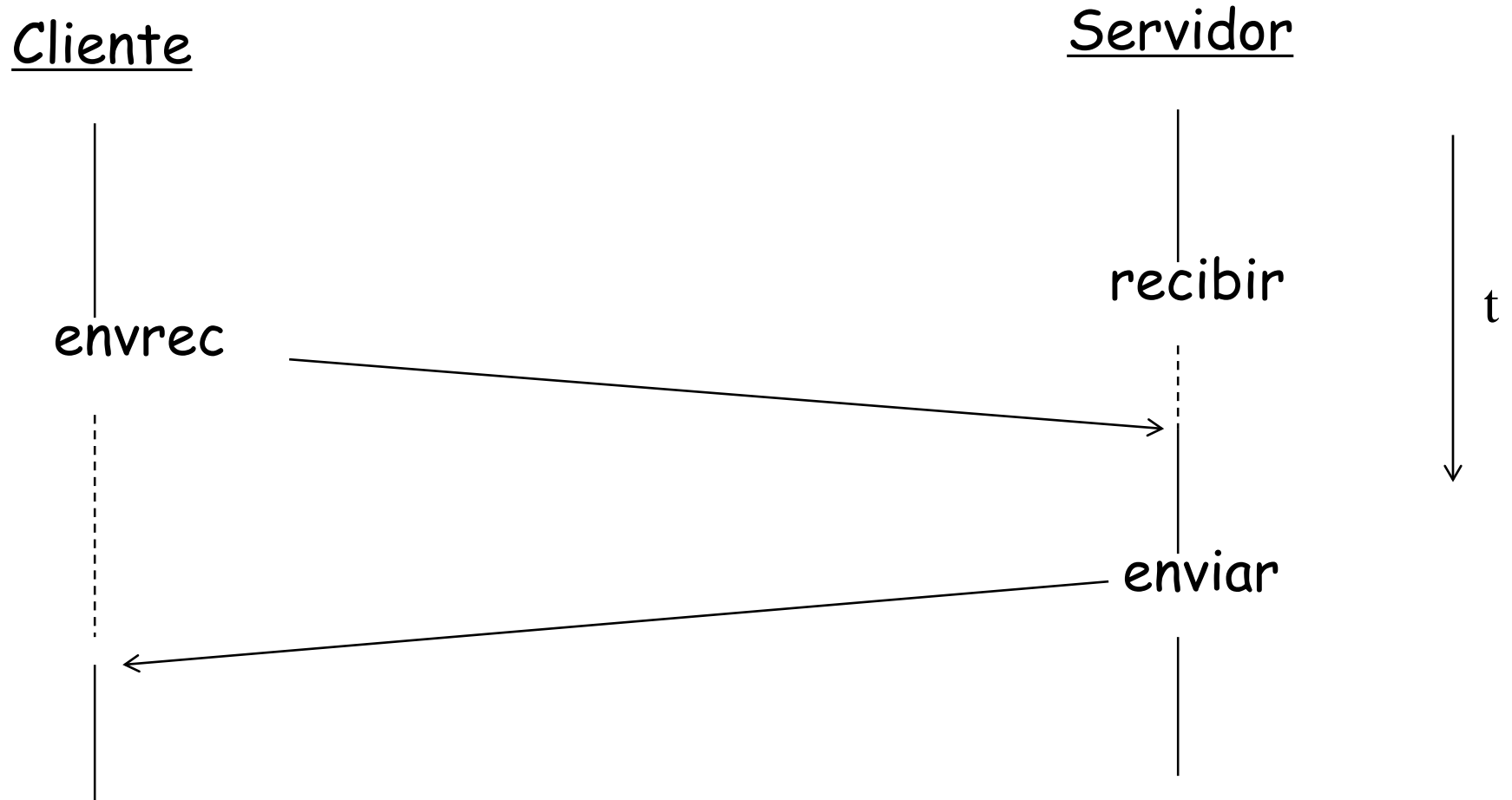
Fallos

- ✓ Cuando el emisor se desbloquea, está seguro de que el mensaje está en el nodo receptor a pesar de fallos en la red o del emisor.
- ✓ También sabe que el receptor está sincronizado en este punto con el emisor.

2. Cliente/servidor

- Si un proceso (cliente) quiere pedir un servicio a otro (servidor) y esperar bloqueado su respuesta puede usar una primitiva cliente/servidor:
 - ✓ `envrec(datos, destino, resultado)`
- El servidor recibirá la petición y después de realizarla enviará al cliente el resultado.

2. Cliente/servidor



2. Cliente/servidor

- Fallos en cliente/servidor
 - ✓ Cuando el cliente se desbloquea, sabe que se ha realizado el servicio a pesar de fallos.
 - ✓ Si durante la interacción existen fallos en la red o en el destino, el cliente se desbloquea recibiendo un resultado de error.
 - ✓ Dependiendo de los protocolos la semántica de la interacción ante fallos será de:
 - al menos una vez,
 - como mucho una vez y
 - exactamente una vez.

3. Calidades de servicio de multicast

- Hay aplicaciones distribuidas que pueden programarse mejor si se tiene un mecanismo de multienvío (multicast) a un grupo de procesos.
- Este es el caso de grupos cooperativos en los que los miembros del grupo reciben la misma petición y cada uno realiza una parte de él.
- También es el caso de los grupos replicados en los que todos los miembros hacen lo mismo ante las mismas peticiones.

3. Calidades de servicio de multicast

- Se habla de broadcast cuando el mensaje se manda a todos los procesos del sistema.
- Se habla de multicast cuando el mensaje se manda sólo a los procesos de un grupo.
- Se debe diferenciar entre recepción de un mensaje en un nodo y entrega del mensaje al proceso del grupo.

3. Calidades de servicio de multicast

- Tipos de grupos:
 - ✓ Grupos cerrados. Solo los procesos del grupo pueden radiar al grupo.
 - ✓ Grupos abiertos. También pueden radiar al grupo procesos externos al grupo.
 - ✓ Grupos estáticos. Se conoce desde el comienzo los miembros del grupo. La membresía del grupo es fija.
 - ✓ Grupos dinámicos. Los procesos entran y sale (fallan) del grupo dinámicamente. La membresía del grupo es variable.

3. Calidades de servicio de multicast

- El radiado se implementa: con comunicación punto-a-punto o con radiado hardware.
- En ambos casos los mensajes se pueden perder si hay fallos.
- También queremos ordenarlos.
- Los radiados pueden clasificarse por:
 - ✓ Garantía de entrega
 - ✓ Orden de entrega
- Ambas clasificaciones son complementarias.

3. Calidades de servicio de multicast

- Clasificación por garantía de entrega:
 - ✓ Best-effort: se trata de hacer llegar el mensaje a todos los destinatarios pero no lo garantiza. En algunas versiones se garantiza la entrega si el emisor vive.
 - ✓ Fiable: un mensaje enviado es entregado en todos los destinatarios vivos o en ninguno vivo.
 - ✓ Uniforme: un mensaje enviado es entregado en todos los destinatarios vivos si se entregó en algún destinatario.

3. Calidades de servicio de multicast

- Clasificación por orden de entrega:
 - ✓ Orden FIFO: los mensajes del mismo emisor se entregan en todos los destinatario en el orden de envío.
 - ✓ Orden Causal: los mensajes se entregan respetando la precedencia causal de los mismos.
 - ✓ Orden Total: los mensajes se entregan en el mismo orden en todos los destinatarios.
 - ✓ Orden Causal y Total: los mensajes se entregan respetando la causalidad y el orden total.

4. Garantías de entrega en multicast

- Garantía de entrega fiable:
 - ✓ El radiado fiable es entregado por todos los miembros vivos o por ningún miembro vivo del grupo.
 - ✓ Además, si el emisor sigue vivo, el mensaje será entregado por todos los vivos.
 - ✓ Decimos vivos porque puede darse el caso de que alguno muerto entregue el mensaje, pero ningún vivo lo hace.

4. Garantías de entrega en multicast

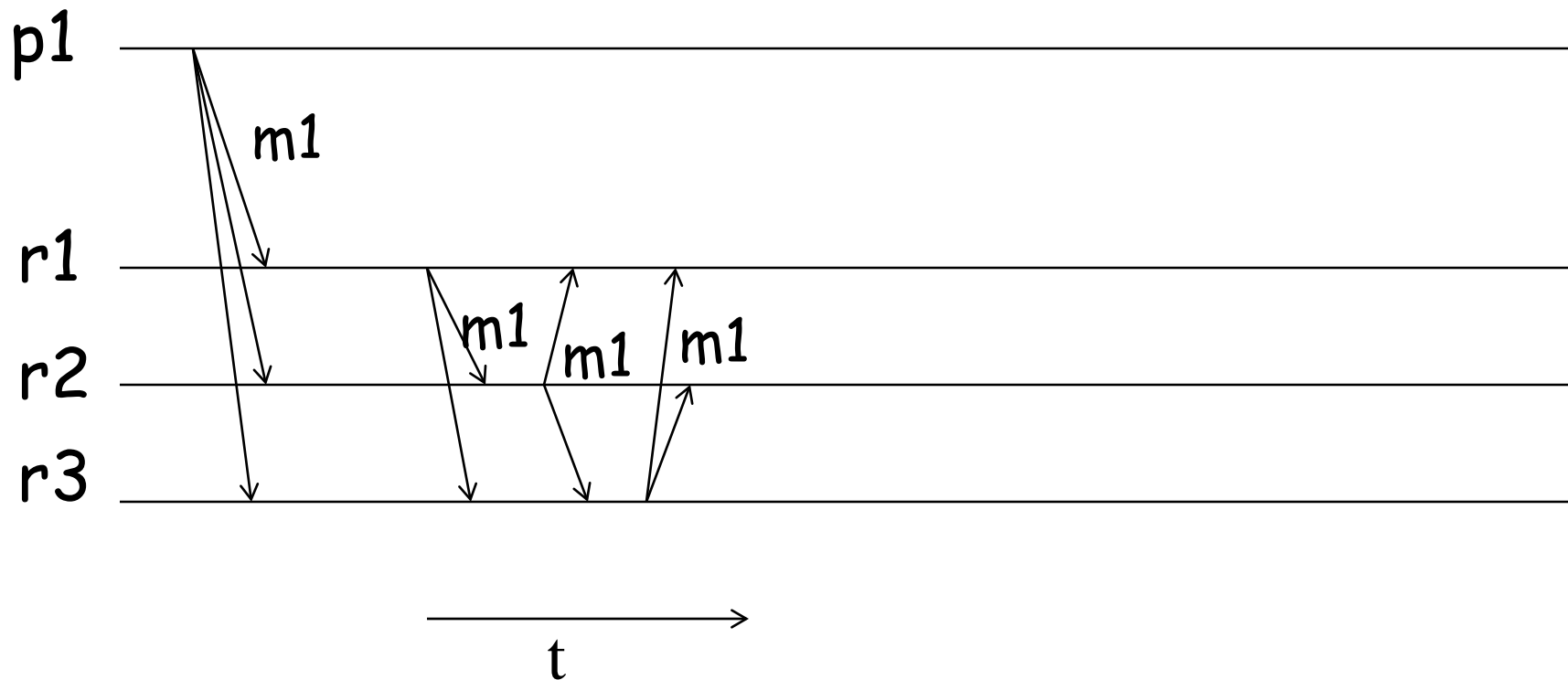
- Garantía de entrega fiable:
 - ✓ Implementar este tipo de radiado no es fácil debido a los fallos:
 - Con comunicación punto a punto, se puede morir el emisor a la mitad y los que faltan no se enteran.
 - Con radiado hardware, pueden perderse mensajes por sobrecarga en los nodos y el emisor puede morir antes de la retransmisión.

4. Garantías de entrega en multicast

- Garantía de entrega fiable:
 - ✓ Un protocolo simple:
 - El emisor empieza enviando fiablemente a cada miembro del grupo.
 - Cada receptor entrega el mensaje a su proceso (miembro del grupo) si no lo ha hecho ya.
 - Cada receptor, reenvía fiablemente a todos los miembros del grupo si no lo ha hecho ya.
 - ✓ Tiene un coste en mensajes de $O(n^2)$.
 - ✓ Hay protocolos menos costosos.

4. Garantías de entrega en multicast

- Garantía de entrega fiable:
 - ✓ Un protocolo simple:

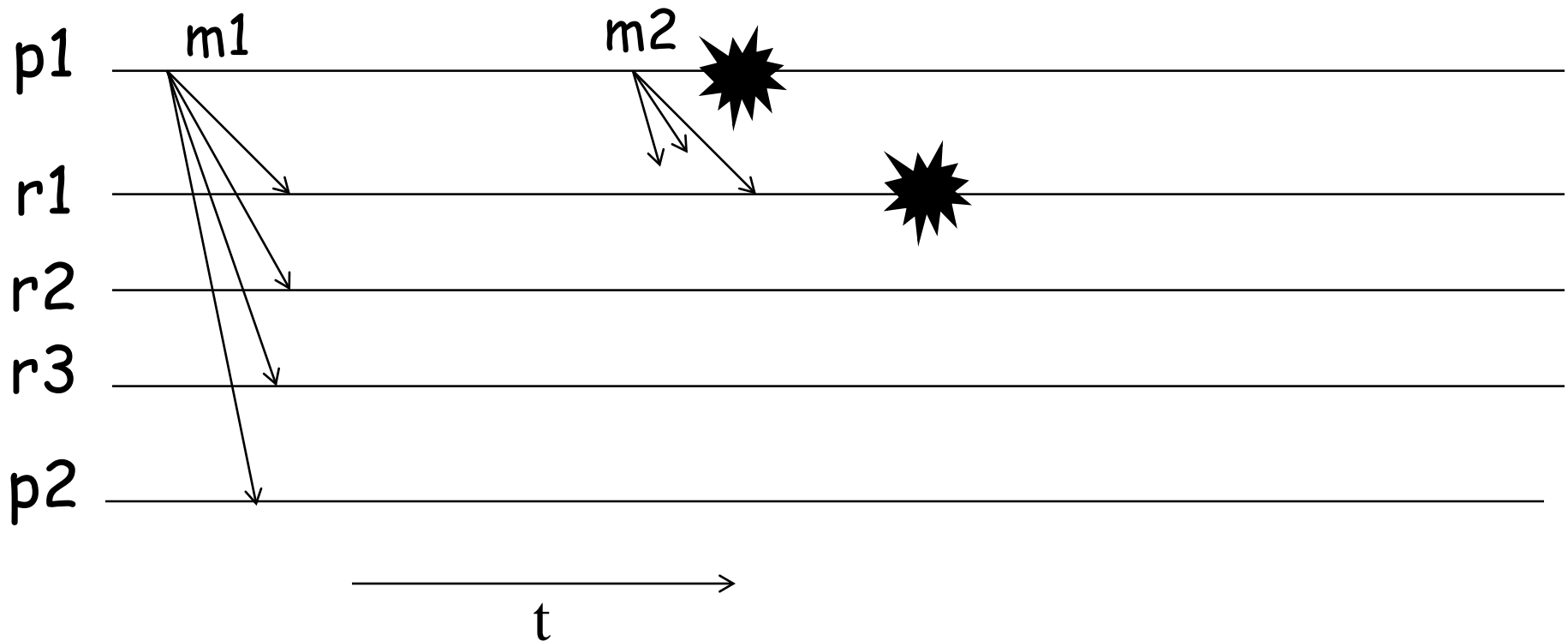


4. Garantías de entrega en multicast

- Garantía de entrega uniforme:
 - ✓ El radiado uniforme entrega un mensaje a un miembro del grupo sólo si está seguro de que será recibido por todos los vivos.
 - ✓ Sin uniformidad puede suceder:
 - Un proceso p_1 envía un mensaje m a todos.
 - Un nodo r_1 recibe el mensaje y lo entrega a su proceso r_1 .
 - Luego r_1 hace algo externo con el mensaje.
 - Luego r_1 y p_1 mueren sin que el resto del grupo reciba m .
 - Una situación como ésta puede darse en un protocolo fiable.

4. Garantías de entrega en multicast

- Garantía de entrega uniforme:
 - ✓ Escenario sin uniformidad pero fiable

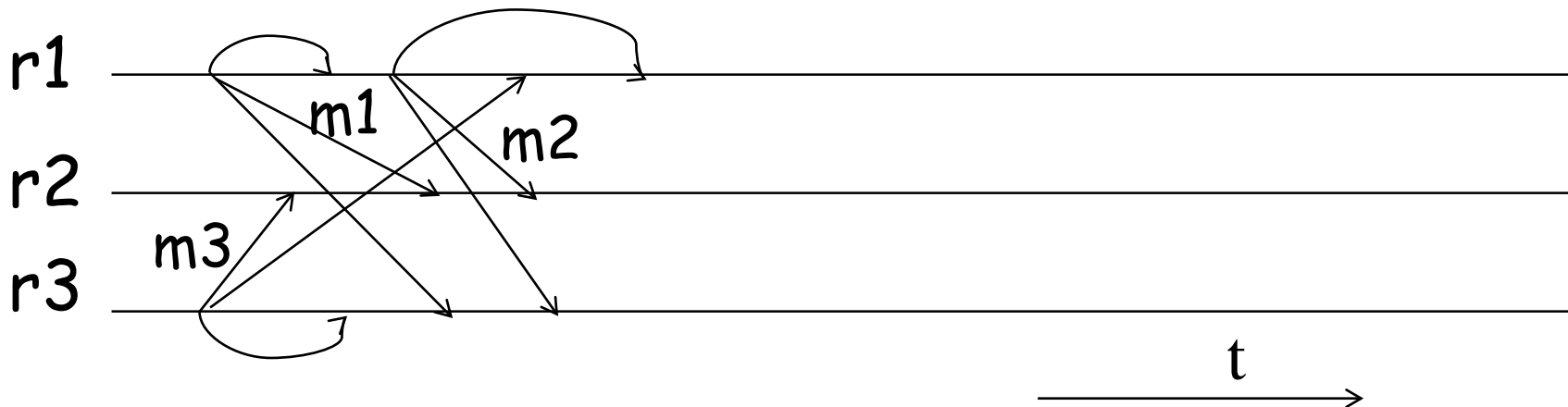


4. Garantías de entrega en multicast

- Garantía de entrega uniforme:
 - ✓ Un protocolo simple:
 - El emisor empieza enviando fiablemente a todos.
 - Cada receptor, reenvía fiablemente a todos los miembros del grupo si no lo ha hecho ya.
 - Cada receptor entrega el mensaje si:
 1. No lo ha hecho ya, y
 2. Ha recibido ACK de los otros miembros vivos.
 - ✓ Tiene un coste en mensajes de $O(n^2)$.
 - ✓ Hay protocolos menos costosos.

5. Garantías de orden en multicast

- Garantía de orden FIFO:
 - ✓ Los mensajes se entregan respetando el orden de envío del emisor.
 - ✓ Se implementa adosando al mensaje el identificador del emisor y un número de secuencia de emisión.
 - ✓ Cada proceso receptor del grupo entrega los mensaje respetando el orden de secuencia de cada proceso emisor.



5. Garantías de orden en multicast

- Garantía de orden Causal:

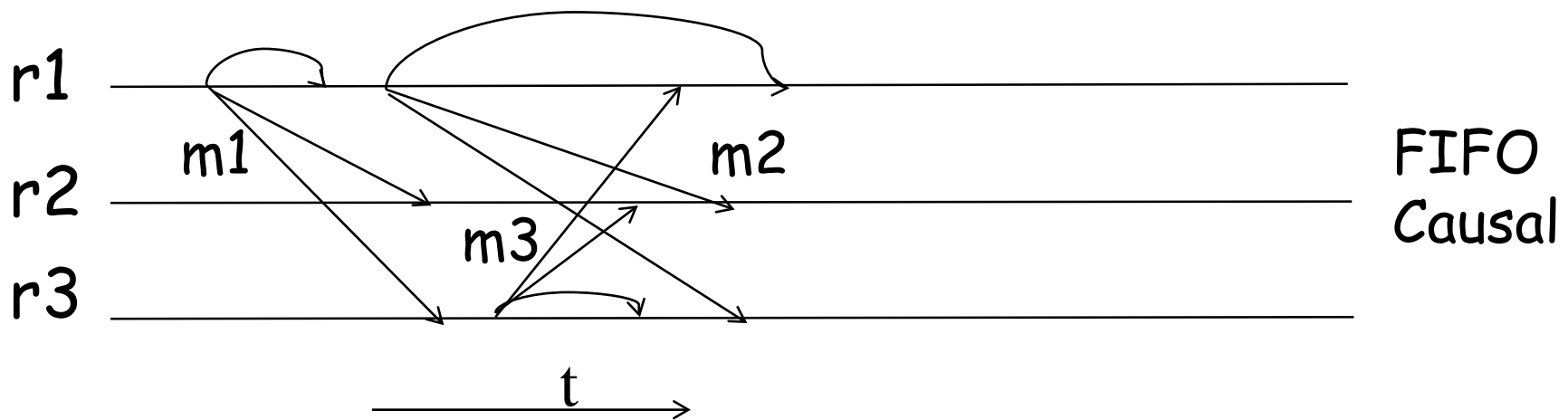
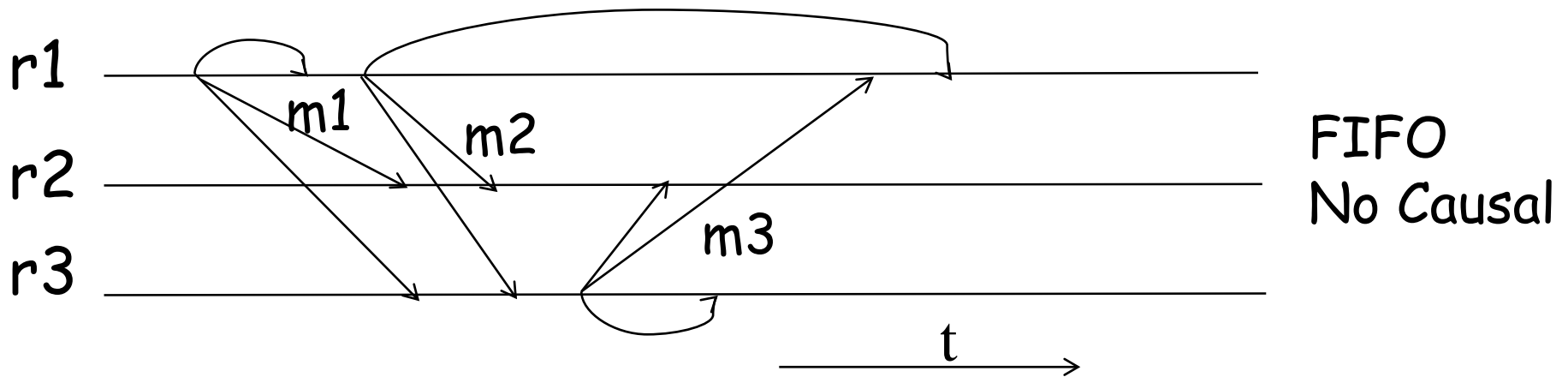
- ✓ Los mensajes se entregan a los miembros del grupo respetando la relación "ocurrió antes". Esto es, si:
 1. Si p_k invoca `multicast(m)` antes de invocar `multicast(m')`.
 2. Si p_k invoca `multicast(m)`, cualquier proceso p_s invoca `deliver(m)`, y posteriormente este mismo p_s invoca `multicast(m')`.

Si 1. o 2. se cumple: todo proceso que entrega m' debe entregar antes m .

- ✓ Es una relación de orden parcial.
- ✓ Orden causal implica orden FIFO.

5. Garantías de orden en multicast

- Garantía de orden Causal:



5. Garantías de orden en multicast

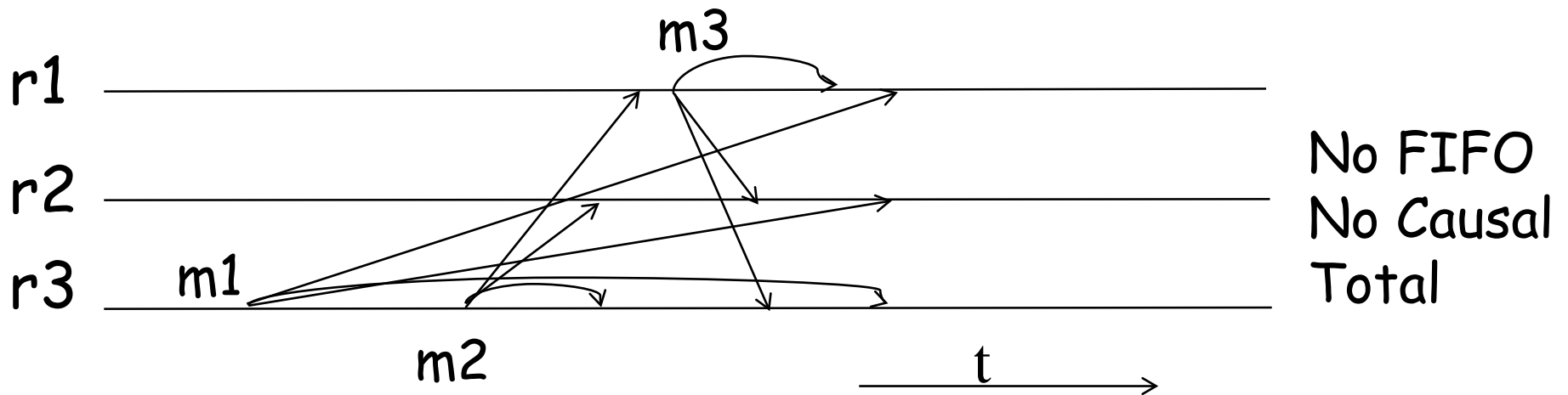
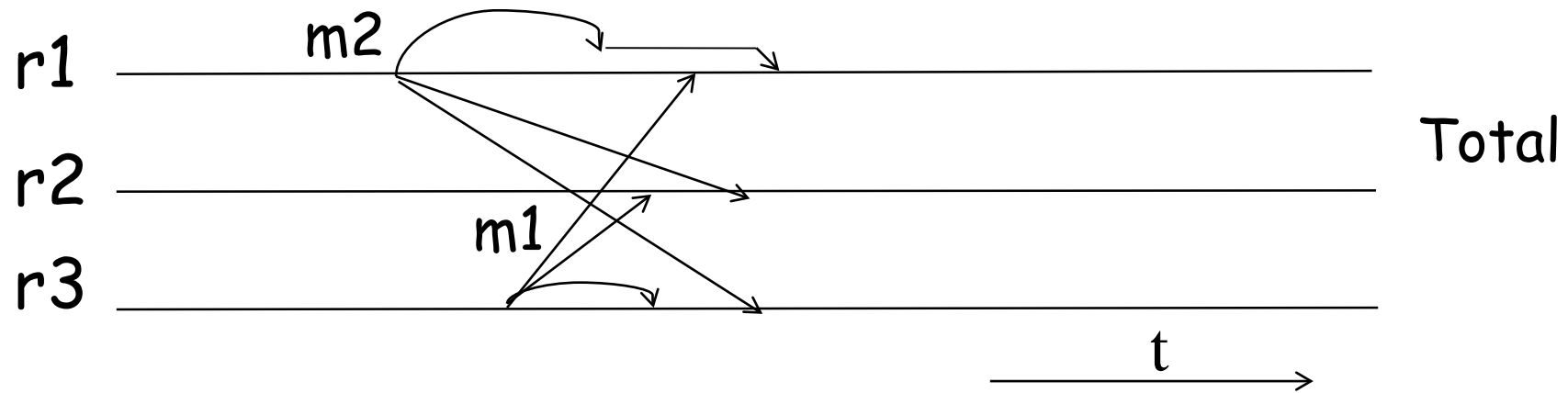
- Garantía de orden Causal:
 - ✓ Protocolo para entrega en orden causal:
 - Cada proceso p_i mantiene un reloj vectorial.
 - Cuando p_i hace un envío de un mensaje m :
 - Difunde el valor del vector V_i junto con el mensaje m , se entrega m así mismo, y luego $V_i[i] \leftarrow V_i[i]+1$.
 - Cuando p_i recibe un mensaje m del proceso p_s , se guarda en un cola y luego:
 - Espera hasta que $V_s[k] \leq V_i[k]$ (para todo k). Esto es, p_i conoce al menos tanto como p_s .
 - Entrega el mensaje m al proceso p_i .
 - $V_i[s] \leftarrow V_i[s] + 1$.

5. Garantías de orden en multicast

- Garantía de orden Total:
 - ✓ Los mensajes se entregan a los miembros del grupo en el mismo orden:
 - Si un proceso p_i entrega m antes que m' , entonces ningún proceso puede entregar m' antes que m .
 - Todos los mensajes enviados por procesos correctos se deben entregar.
 - ✓ El orden de los mensajes es importante para implementar encima servicios tolerantes a fallos con replicación.
 - ✓ Los servicios replicados tienen la misma especificación, pasan por los mismos estados y devuelven los mismos resultados.
 - ✓ Para mantener este comportamiento, deben recibir las mismas peticiones (fiable) y en el mismo orden total.

5. Garantías de orden en multicast

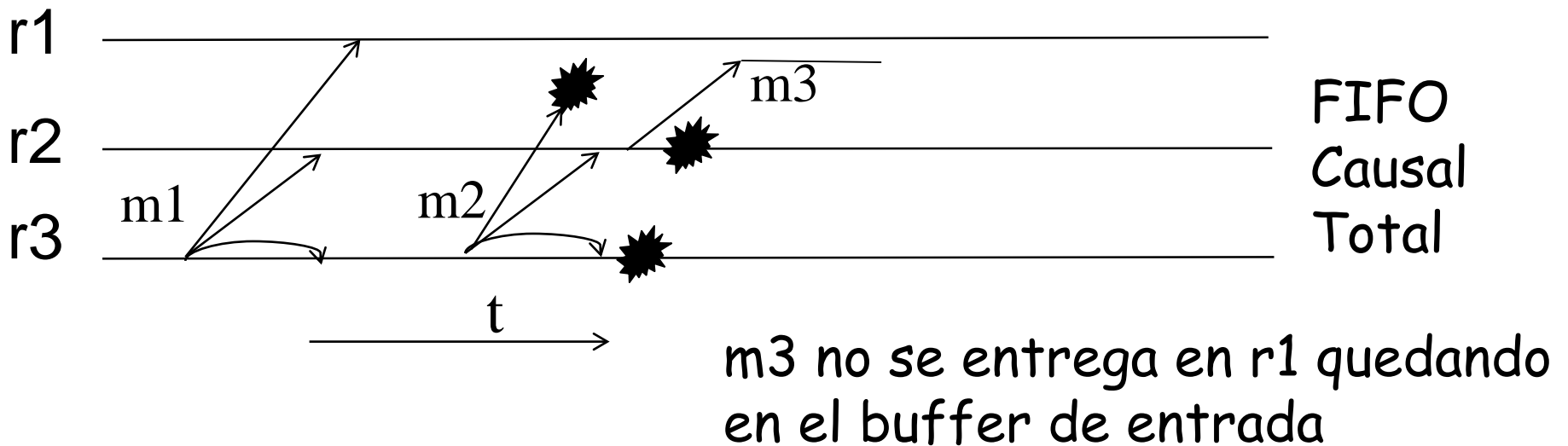
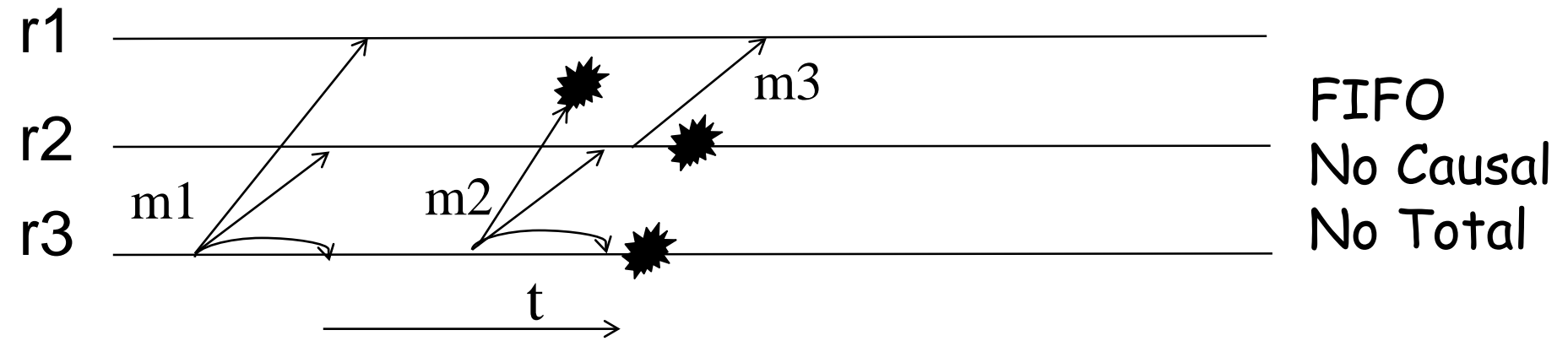
- Garantía de orden Total:



No FIFO
No Causal
Total

5. Garantías de orden en multicast

- Garantía de orden Total:

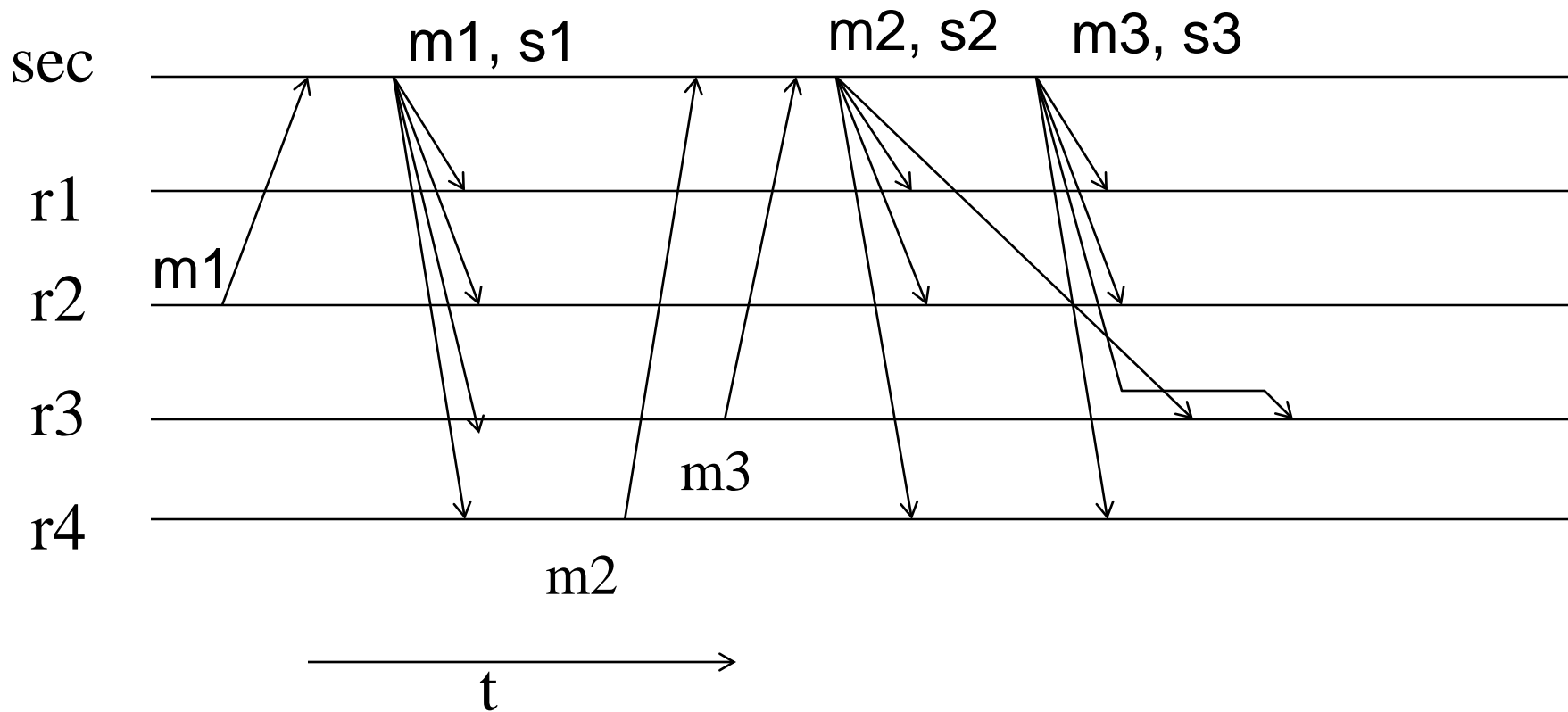


5. Garantías de orden en multicast

- Garantía de orden Total:
 - ✓ Un protocolo típico de ordenación total es usar un secuenciador (Chang & Machemchuk, Tanenbaum).
 - ✓ Un proceso secuenciador va asignando orden a los mensajes radiados.
 - ✓ Un proceso va entregando los mensajes en secuencia.
 - ✓ Problema si el secuenciador falla. Hay que realizar una fase de recuperación.

5. Garantías de orden en multicast

- Garantía de orden Total:



6. Servicio de membresía de grupos

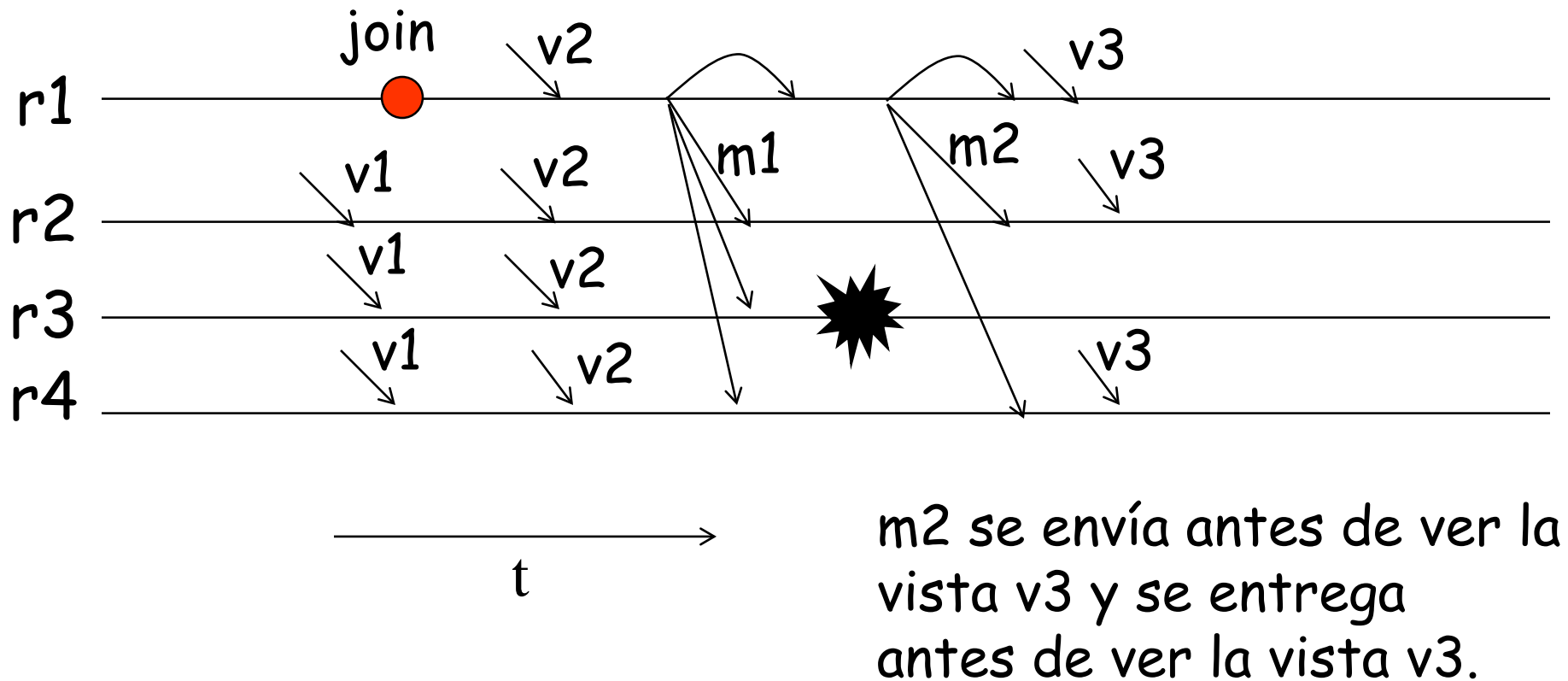
- Cuando los grupos son dinámicos se necesita un servicio de la membresía.
- Este servicio informa de los miembros actuales del grupo.
- Operación JOIN para entrar en el grupo.
- Operación LEAVE para dejar el grupo. Un fallo es un LEAVE implícito.
- Con cada operación JOIN/LEAVE, el servicio informa a los miembros del cambio de vista del grupo.

6. Servicio de membresía de grupos

- El mensaje de vista del grupo informa de los miembros actuales del grupo.
- Se debe preservar que un mensaje radiado se entregue en la misma vista en todos los miembros del grupo.
- Dos semánticas de entrega de mensajes de vistas del grupo:
 - ✓ Entrega "*sending view*". El mensaje radiado en una vista se entrega antes que el radiado de cambio de vista.
 - ✓ Entrega "*same view*". Un mensaje se entrega en todos los procesos en la misma vista.

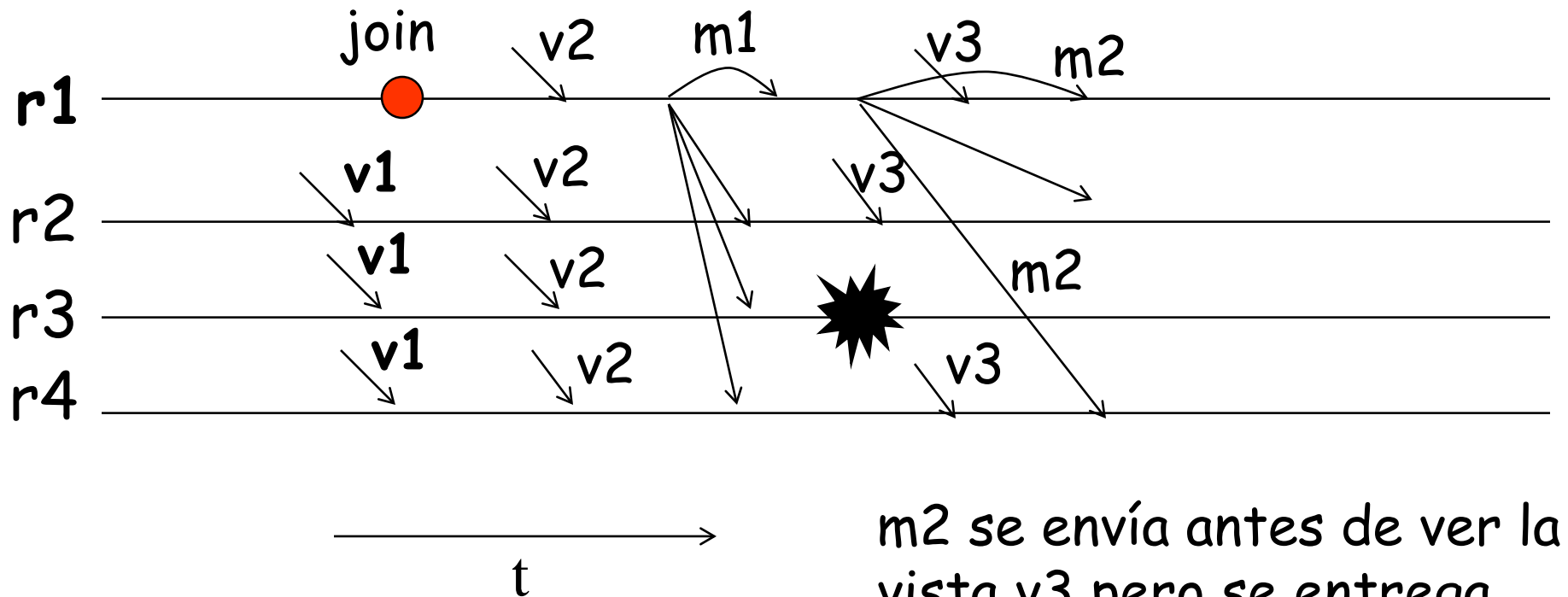
6. Servicio de membresía de grupos

- Entrega *sending view*:



6. Servicio de membresía de grupos

- Entrega *same view*:



m2 se envía antes de ver la vista v3 pero se entrega después de ver la vista v3.