

Algoritmo para la detección de fallos no fiable

- Cada proceso p envía “ p está aquí” al resto de los procesos y lo hace cada T unidades de tiempo.
- El detector de fallos utiliza una estimación del tiempo máximo de transmisión de un mensaje de D segundos.
- Si el detector de fallos local en otro proceso q no recibe un mensaje “ p está aquí” dentro de los $T+D$ segundos desde el último, entonces:
 - informa a q que p es *Sospechoso*.

Algoritmo del servidor central

- Se emplea un servidor que da los permisos para entrar a la S.C mediante un testigo (*token*) que se envía por mensajes, el cual significa permiso.
- Si otro proceso tiene al testigo, el proceso solicitante se envía a una cola de peticiones.
- La atención en la cola se escoge mediante antigüedad.

Evaluación del rendimiento del algoritmo del servidor central

- La entrada a la sección crítica conlleva 2 mensajes de entrada, (una petición y una concesión) y 1 mensaje de salida: una liberación.
- Retraso al proceso solicitante por la duración de este mensaje de ida y vuelta.
- El servidor puede convertirse en un cuello de botella para el rendimiento del sistema total.

Algoritmo basado en anillo

- Cada proceso tiene un canal de comunicación hacia el siguiente proceso en el anillo.
- *Testigo* circulando en una dirección.
- Si un proceso requiere entrar a la SC, espera el testigo y lo retiene y cuando sale de SC enviará el testigo hacia el vecino.
- Si no requiere entrar a SC, inmediatamente lo hace avanzar hacia su vecino.

Evaluación del rendimiento del algoritmo basado en anillo

- Consumo continuo de ancho de banda salvo si el proceso está en la SC.
- Los procesos siempre están enviando el testigo, aun cuando ningún proceso requiera entrar a SC.