



MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

PRESENTA: JUAN CARLOS
ZARATE TREJO

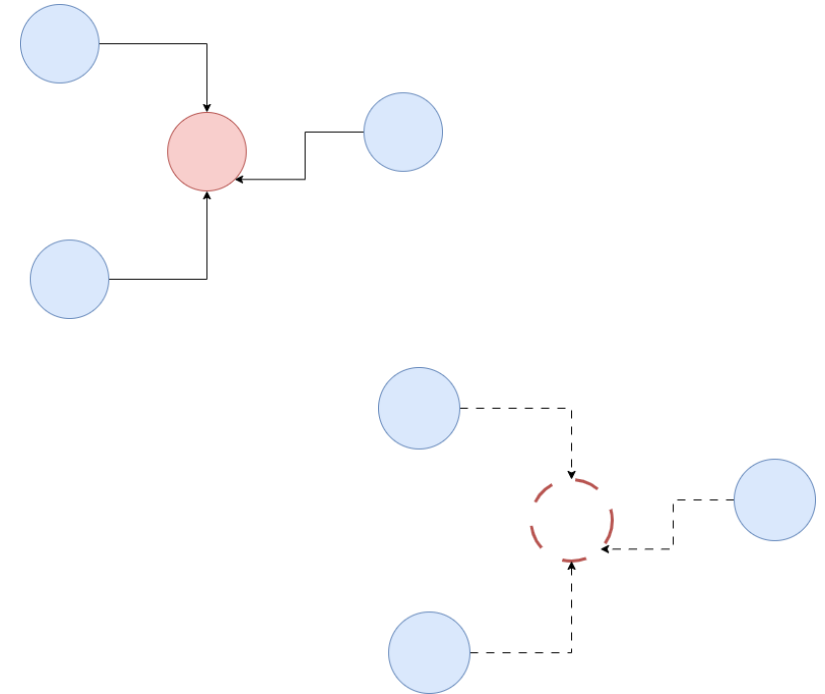
Sistemas Distribuidos – Dr. Félix Francisco Ramos
Cinvestav Unidad GDL

Introducción

Planteamiento del problema

En el contexto de los sistemas distribuidos con frecuencia se aborda el tópico de la tolerancia a fallos en el sistema. Esto fallos pueden ser de distintos tipos y bajo distintas condiciones. En este proyecto se abordó uno en particular: La caída del nodo principal.

Cómo otorgar características de una red centralizada a un sistema distribuido permitiendo flexibilidad en los roles de los nodos que la conforman.

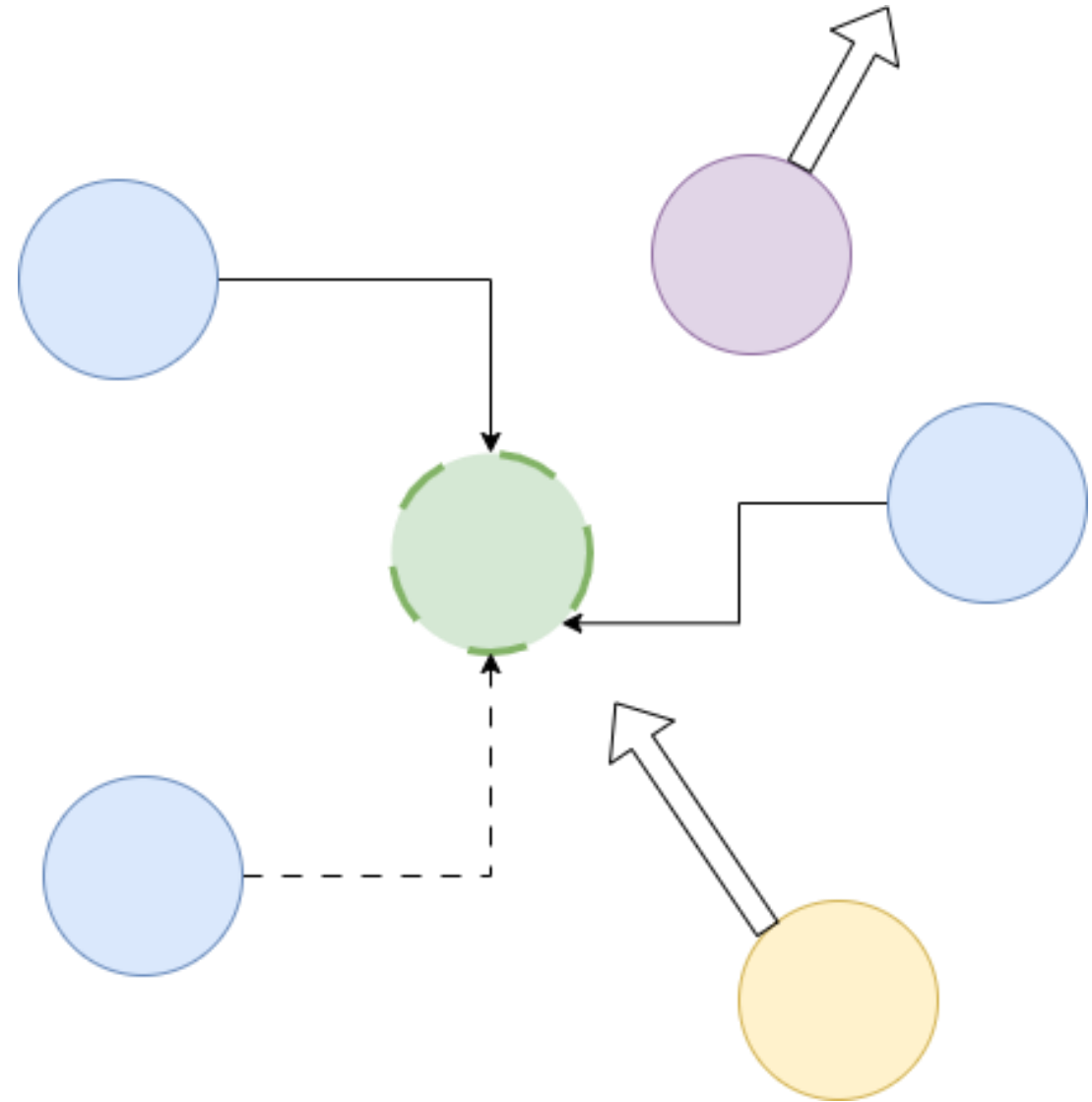


Propuesta

Se propone un modelo de comunicación orientado a conexión persistente para verificación de disponibilidad del nodo central y búsqueda de un candidato a tomar su rol.

El modelo se caracterizará principalmente por soportar la entrada y salida continua de nodos en la red, tolerancia* a conexiones inestables y la coordinación para determinar al nuevo nodo central.

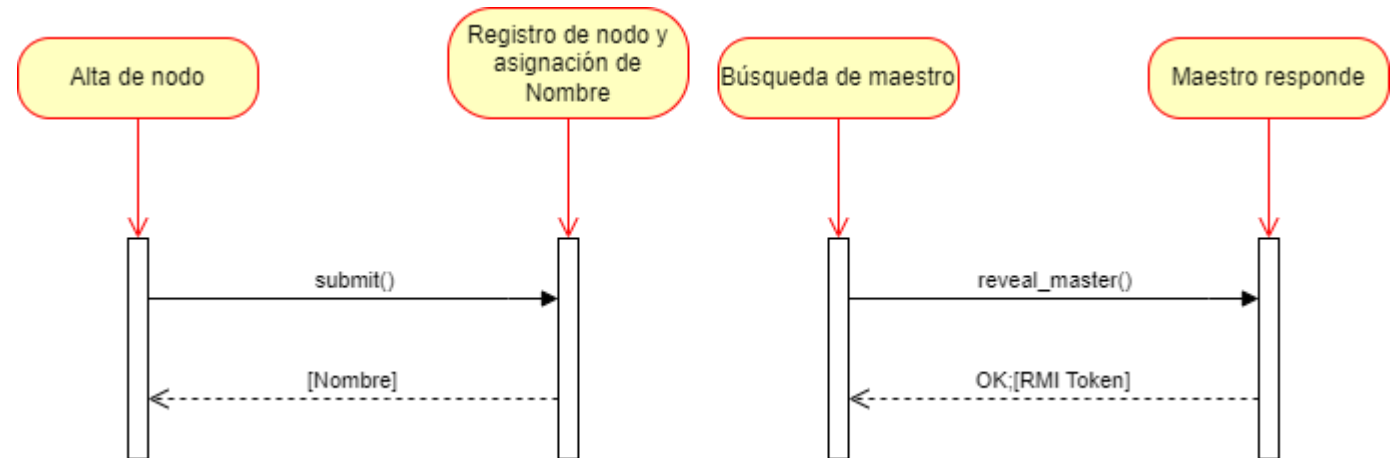
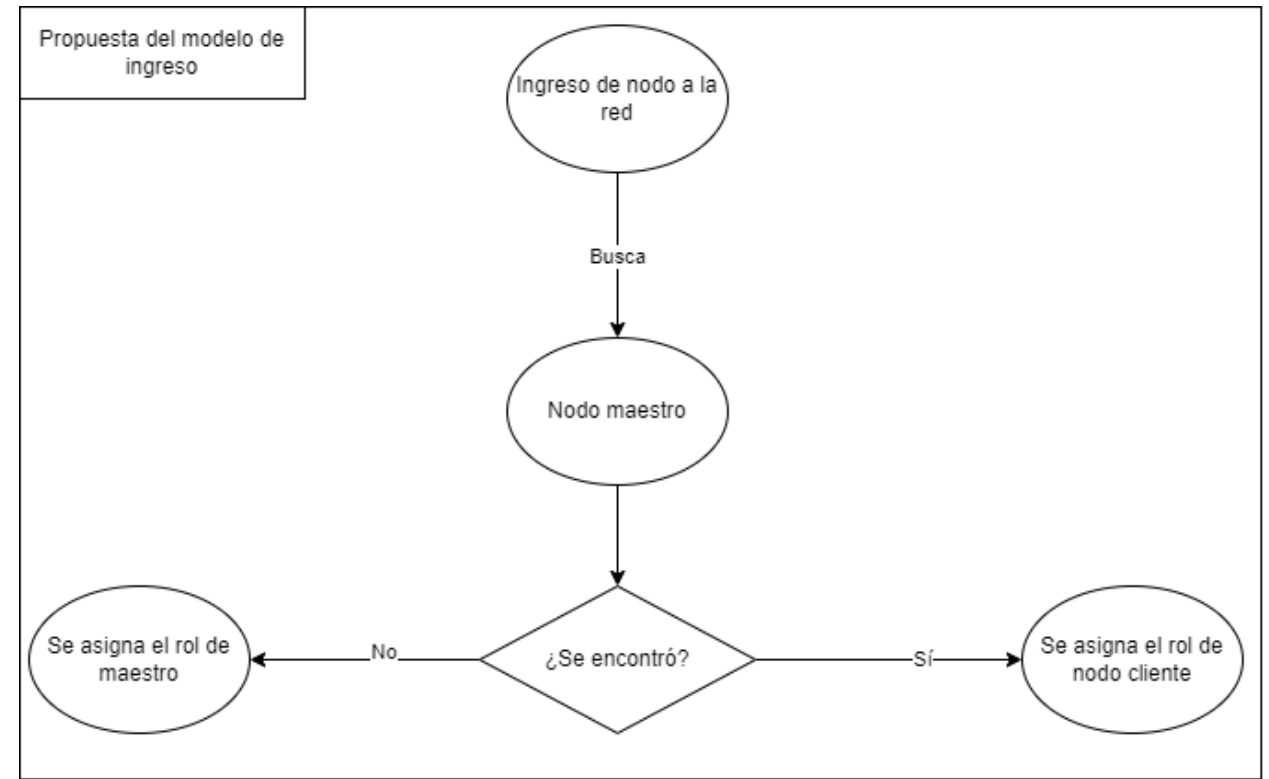
*Existe un margen de tiempo para ello



Diseño

Ingreso de nodos

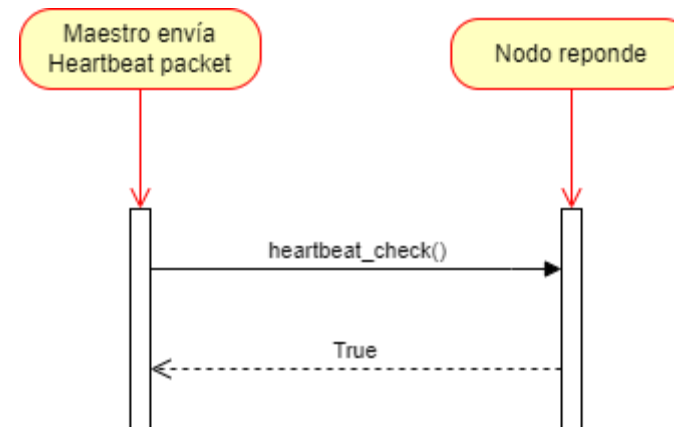
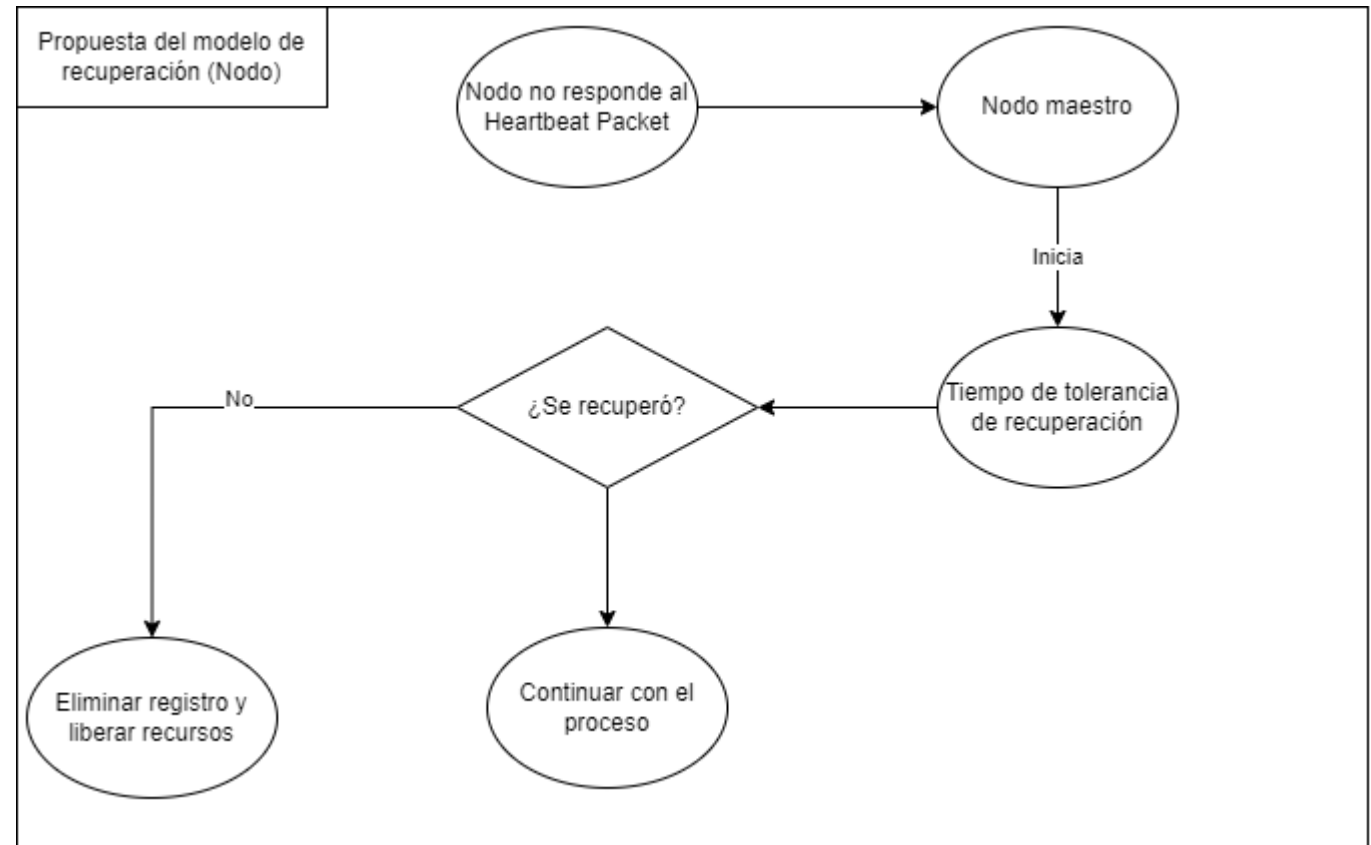
Los nodos se encargan de identificar un nodo maestro en la red. Si existe, se registran con éste para iniciar el protocolo de verificación de estado. De lo contrario, se configura para tomar el rol de maestro y estar a la espera de nuevos nodos.



Diseño

Fallos en nodos clientes

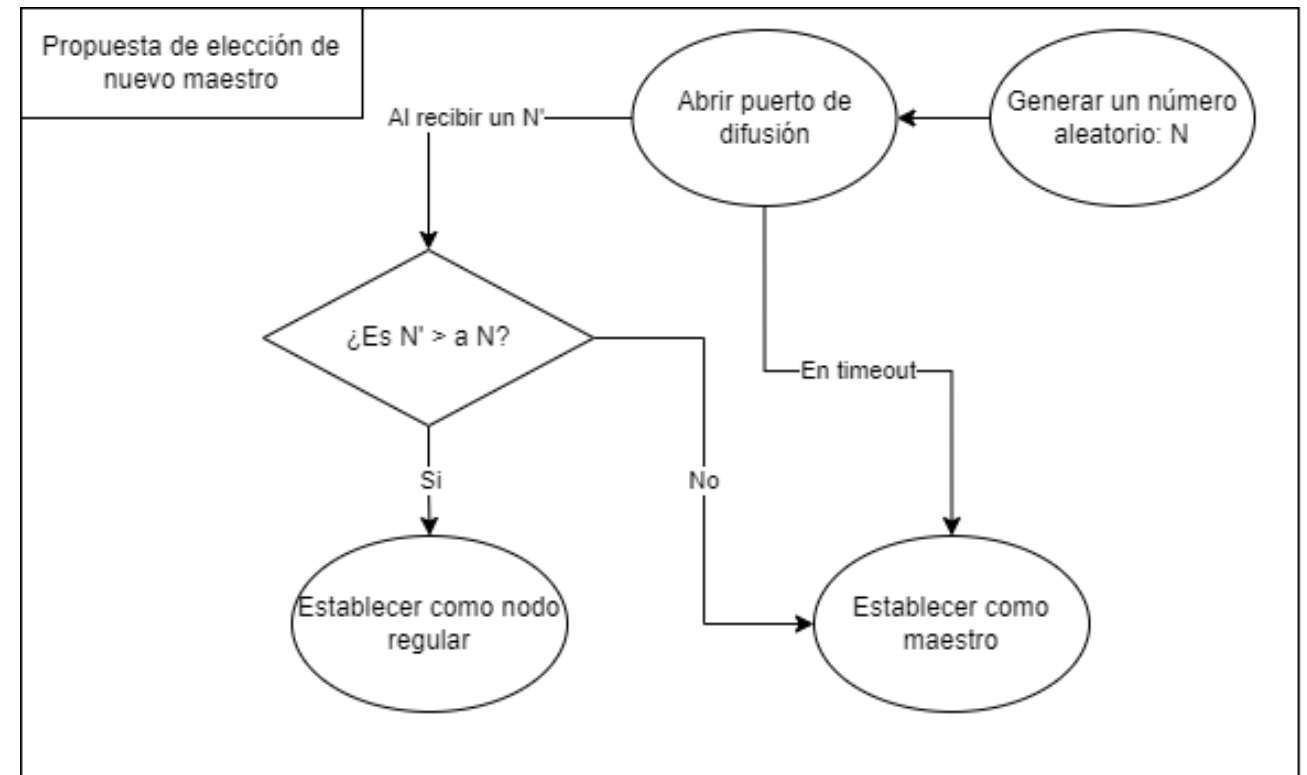
Cuando los nodos miembros presentan fallos en la conexión, el nodo maestro otorga un margen de tolerancia, dentro del cual, permite el reingreso del nodo. En caso contrario, descarta los recursos para la comunicación con dicho nodo y lo elimina del registro.



Diseño

Fallos en nodos maestro

Cuando los nodos miembros detectan un fallo en el nodo maestro (fallo por timeout), inician una subrutina para coordinar la elección de un nuevo maestro.



Pruebas

Comunicación entre múltiples
nodos y el maestro

El nodo central/maestro posee tres
canales de comunicación
esencialmente: Manejo de Seekers
(Nodos nuevos), Alta de nodos en
la red y Servicio de Heartbeat
(Verificación de estado)

```
[S] Timed out. Retrying...  
[Master] Listening seekers  
[Master] Announcing to ('192.168.56.1', 58604)  
[Master] Registered: Node_5368967957434806, Heartbeat: ('192.168.56.1', 16000)  
[Master] Trying heartbeat  
[Master] Ready!  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_5368967957434806  
[Master] Announcing to ('192.168.56.1', 51457)  
[Master] Registered: Node_6549400277156321, Heartbeat: ('192.168.56.1', 16002)  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] Announcing to ('192.168.56.1', 51458)  
[Master] Registered: Node_2863947303129494, Heartbeat: ('192.168.56.1', 16004)  
[Master] OK: Node_5368967957434806  
[Master] OK: Node_6549400277156321  
[Master] OK: Node_2863947303129494
```

Pruebas

Comunicación entre múltiples nodos y el maestro

Los nodos clientes realizan una búsqueda por la red y en cuanto encuentran al maestro, se dan de alta, de manera que el maestro los incluya dentro de la rutina de verificación de disponibilidad.

```
[0] Timed out. Retrying...
[1] Timed out. Retrying...
[2] Timed out. Retrying...
[3] Timed out. Retrying...
[4] Timed out. Retrying...
[5] Master founded
[None] Submitting...
[Node_5368967957434806] Submission succesfully
[Node_5368967957434806] Heartbeat service is running at ('192.168.56.1', 16000)
[Node_5368967957434806] Heartbeat received
192.168.56.1 - - [28/Apr/2023 14:40:10] "POST / HTTP/1.1" 200 -
```

```
[0] Master founded
[None] Submitting...
[Node_6549400277156321] Submission succesfully
[Node_6549400277156321] Heartbeat service is running at ('192.168.56.1', 16002)
[Node_6549400277156321] Heartbeat received
192.168.56.1 - - [28/Apr/2023 14:40:19] "POST / HTTP/1.1" 200 -
```

```
[0] Master founded
[None] Submitting...
[Node_2863947303129494] Submission succesfully
[Node_2863947303129494] Heartbeat service is running at ('192.168.56.1', 16004)
[Node_2863947303129494] Heartbeat received
192.168.56.1 - - [28/Apr/2023 14:40:36] "POST / HTTP/1.1" 200 -
```


Pruebas

Caída de nodos cliente - Recuperación

Cuando un nodo pierde comunicación con el maestro, se asume en primera instancia que se debió a inestabilidad en el canal o pérdida de paquete, por lo que el nodo puede reintegrarse a la red dentro del periodo de espera.

```
[Master] Exception: Cant't connect to node
[Master] Refused connections by: Node_2863947303129494 => 1
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
[Master] Announcing to ('192.168.56.1', 61799)
[Master] Exception: Cant't connect to node
[Master] Refused connections by: Node_2863947303129494 => 2
[Master] Recovered: Node_2863947303129494, Heartbeat: ('192.168.56.1', 16004)
```

```
[0] Master founded
[None] Submitting...
[Node_2863947303129494] Submission succesfully
[Node_2863947303129494] Heartbeat service is running at ('192.168.56.1', 16004)
[Node_2863947303129494] Heartbeat received
192.168.56.1 - - [28/Apr/2023 14:49:56] "POST / HTTP/1.1" 200 -
```

Pruebas

Caída de nodos cliente - Eliminación

Si el nodo cae y no es capaz de recuperarse en el tiempo de espera, el nodo central asumirá que se ha desconectado definitivamente y lo excluirá de futuras llamadas de verificación de estado.

```
[Master] OK: Node_2863947303129494
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
[Master] Exception: Cant't connect to node
[Master] Refused connections by: Node_2863947303129494 => 1
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
[Master] Exception: Cant't connect to node
[Master] Refused connections by: Node_2863947303129494 => 2
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
[Master] Exception: Cant't connect to node
[Master] Refused connections by: Node_2863947303129494 => 3
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
[Master] Exception: Cant't connect to node
[Master] Maximum refused connections reached. Removing Node_2863947303129494
[Master] OK: Node_5368967957434806
[Master] OK: Node_6549400277156321
```

Pruebas

Caída de Master

Cuando el nodo maestro envía los Heartbeat Packets, reinicia continuamente una rutina de alerta en los nodos que impide que asuman que el nodo maestro cayó. De manera que en cuanto cae, los nodos cliente proceden a determinar un nuevo maestro.

```
[Node_9141137987603043] MASTER MISSING
[Node_9141137987603043] Generated: 0.15997660871891306
[Node_9141137987603043] Waiting for other numbers...
[Node_9141137987603043] Sending my number...
[Node_9141137987603043] Smallest than Node_13299723083767723: (0.15997660871891306, 0.33721881775760354)
[Node_9141137987603043] Ill be Node
[1] Timed out. Retrying...
[2] Timed out. Retrying...
[3] Timed out. Retrying...
[4] Master founded
[None] Submitting...
[Node_9122814050553821] Submission succesfully
[Node_9122814050553821] Heartbeat service is running at ('192.168.56.1', 16002)
[Node_9122814050553821] Heartbeat received
192.168.56.1 - - [28/Apr/2023 15:08:32] "POST / HTTP/1.1" 200 -
```

```
[Node_13299723083767723] MASTER MISSING
[Node_13299723083767723] Generated: 0.33721881775760354
[Node_13299723083767723] Waiting for other numbers...
[Node_13299723083767723] Sending my number...
[Node_13299723083767723] Smallest than Node_7075112180145805: (0.33721881775760354, 0.4406531353248274)
[Node_13299723083767723] Ill be Node
[1] Timed out. Retrying...
[2] Timed out. Retrying...
[3] Timed out. Retrying...
[4] Master founded
[None] Submitting...
[Node_9578210123690737] Submission succesfully
[Node_9578210123690737] Heartbeat service is running at ('192.168.56.1', 16004)
[Node_9578210123690737] Heartbeat received
192.168.56.1 - - [28/Apr/2023 15:08:32] "POST / HTTP/1.1" 200 -
```

Pruebas

Caída de Master

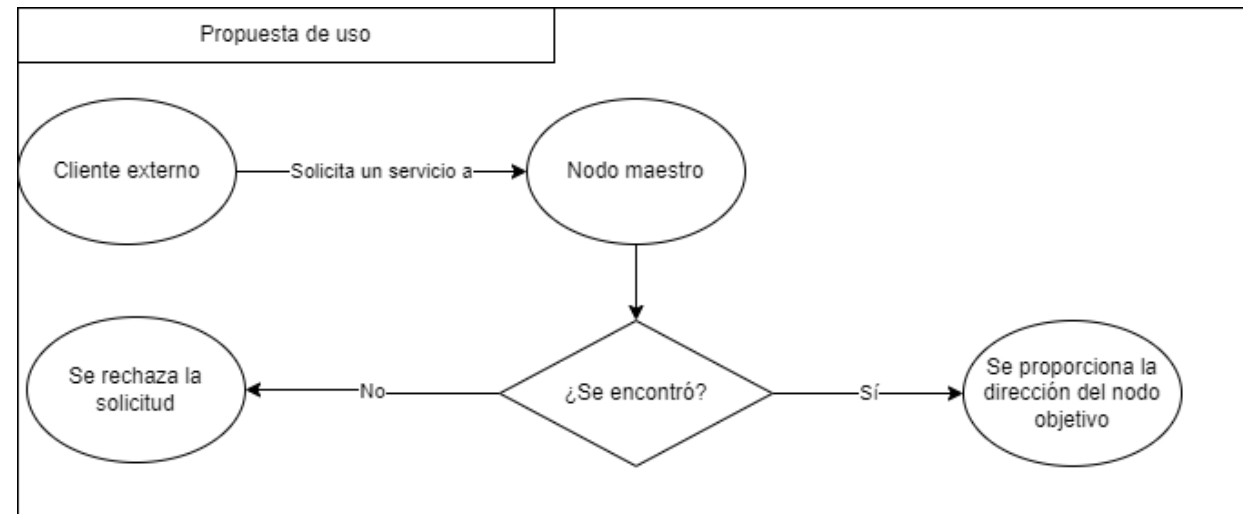
Se genera un intercambio de números aleatorios, el nodo que haya generado el número más grande se convertirá en el nuevo maestro y habilitará los servicios de administración para que el resto de nodos lo localicen y se den de alta.

```
[Node_7075112180145805] MASTER MISSING
[Node_7075112180145805] Generated: 0.4406531353248274
[Node_7075112180145805] Waiting for other numbers...
[Node_7075112180145805] Sending my number...
[Node_7075112180145805] Timeout
[Node_7075112180145805] Timeout
[Node_7075112180145805] Timeout
[Node_7075112180145805] Ill be master
[Master] Listening seekers
[Master] Announcing to ('192.168.56.1', 58506)
[Master] Announcing to ('192.168.56.1', 58505)
[Master] Registered: Node_9122814050553821, Heartbeat: ('192.168.56.1', 16002)
[Master] Registered: Node_9578210123690737, Heartbeat: ('192.168.56.1', 16004)
[Master] Trying heartbeat
[Master] Ready!
[Master] OK: Node_9122814050553821
[Master] OK: Node_9578210123690737
```

Aplicaciones

Entre los usos posibles y hacia el que se enfocó principalmente el modelo, es a la persistencia de disponibilidad de servicios.

El modelo asume que todos los nodos son computacionalmente equivalentes (Pueden hacer lo mismo), pero que podrían contar con diferentes recursos (Cuentan con aditamentos que les permiten desempeñar mejor ciertas tareas). Entonces algún cliente externo puede solicitar algún servicio que algún nodo proporcione.

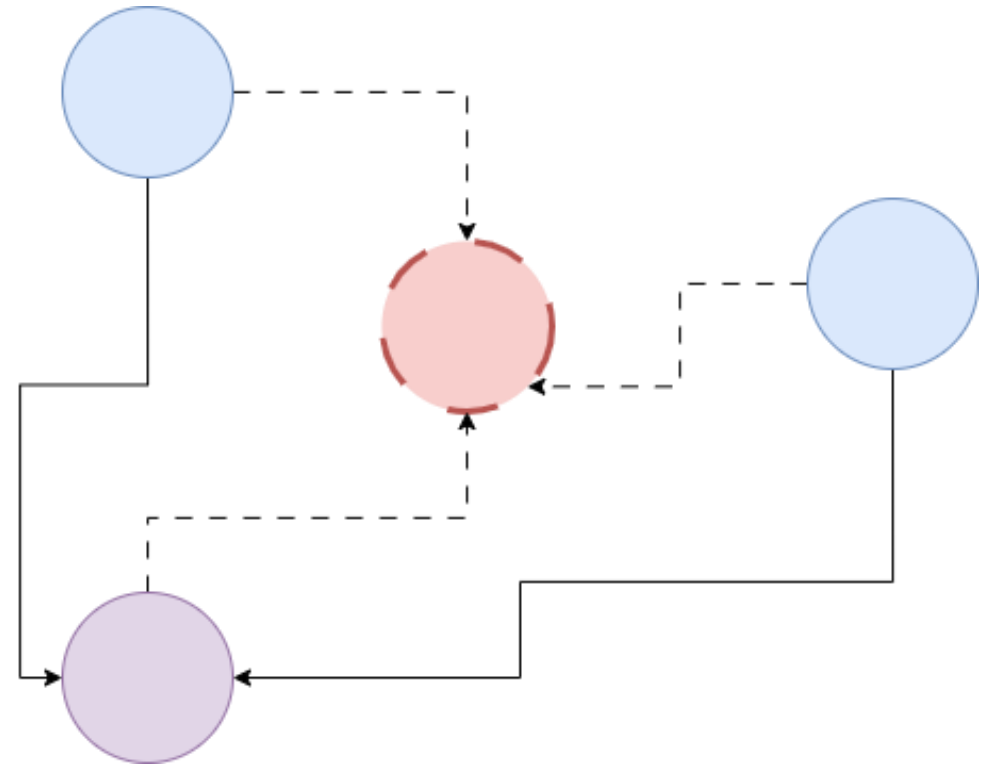


Aplicaciones

¿Por qué de este supuesto?

Porque se busca que el sistema se mantenga en línea en la mayor cantidad de casos posibles.

En realidad no está especificado, el modelo lo asume implícitamente, pero se podría definir explícitamente en la implementación que un nodo no es apto para asumir el rol de maestro, o que un maestro debe cumplir ciertas características.



Trabajo a futuro

- Integración de métodos para balaceo de cargas
 - Si un nodo que se encontraba ejecutando un servicio “x” cae, el maestro localizará o bien algún nodo libre, o bien, algún nodo apto para ejecutar el servicio “x”, hasta que el nodo original se reponga o ingrese algún otro nodo nuevo más adecuado. Podría existir más flexibilidad si se especifica qué servicios requieren respaldo y cuáles no.
- Persistencia de información
 - Los nodos deben poseer información suficiente para restaurar los estados en los que se encontraban antes de entrar en los eventos de caída, de manera que ahora sean aptos para sistemas más sensibles.
- Servicio de Heartbeat Ad Hoc al tipo de nodo
 - Determinar tiempos de verificación acordes al tipo de nodo, es decir, ¿Bajo qué cantidad de tiempo es probable que el nodo “i” haya caído?, para disminuir la congestión en la red en presencia de muchos nodos.
- Cifrado por claves RSA