

Tarea 1

Información General

Diego Méndez Medina

1. Considera un sistema distribuido con $n \geq 2$ procesos p_1, p_2, \dots, p_n , en el que la gráfica de comunicación es la completa K_n . El sistema es síncrono pero la comunicación *no es confiable*; sea P el conjunto de todos los procesos que envían mensajes en el tiempo d ; entonces, hay dos posibilidades, todos los mensajes de P llegan a su destino en el tiempo $d+1$, o uno de ellos se pierde y nunca llega a su destino y los otros en P si llegan en el tiempo $d+1$.

Considera un algoritmo A en el que cada proceso p_i tiene como entrada un identificador ID_i , que es un número natural (diferente al de los demás), y cada proceso p_i simplemente envía su ID , a los otros $n-1$ procesos. Dibuja cuales son todos los estados *globales* posibles (mundos posibles) en el tiempo 1 (los procesos mandan sus mensajes en el tiempo 0). En cada estado global, especifica el estado local de cada proceso, es decir, la información que cada proceso tiene en estado global; y entre cada par de estados globales pinta una arista con los procesos que no pueden *distinguir* entre esos estados. ¿Es posible que cada proceso elija consistentemente uno de los ID s de entre los que recibió de forma tal que en cada estado global todos los procesos eligen el mismo ID ? Argumenta tu respuesta.

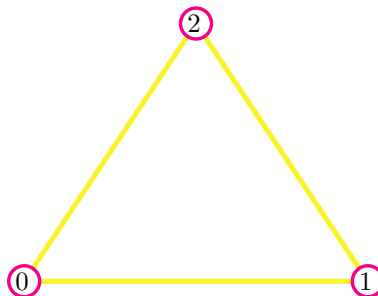
Solución:

Primero un poco de notación. Siguiendo la notación de la ayudantía, la siguiente imagen representa el proceso n en la gráfica y los números azules son su estado local, es decir, los procesos que se comunicaron con él.

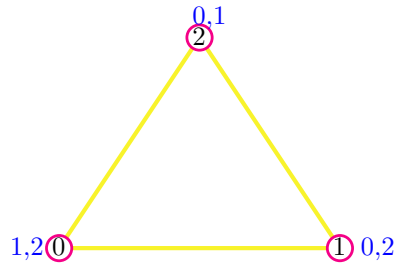


Tomemos $n = 3$. Enumeramos los tres procesos comenzando desde el cero, así tenemos el conjunto $P = \{p_0, p_1, p_2\}$, que de acuerdo al algoritmo descrito cada proceso p_i envía su ID al resto de procesos.

La siguiente gráfica representa el tiempo cero.

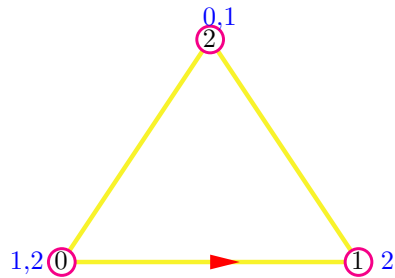


Lo siguiente es mostrar los casos globales posibles en el tiempo uno. El primer caso que mostraremos es el *perfecto*, cuando cada p_i en P recibe el ID del resto de los procesos.



Ahora, de acuerdo con la descripción del problema existen dos posibilidades, la primera antes mencionada y la segunda: “uno de los mensajes de P se pierde y nunca llega a sus destino y los otros en P si llegan en tiempo $d+1$ ”.

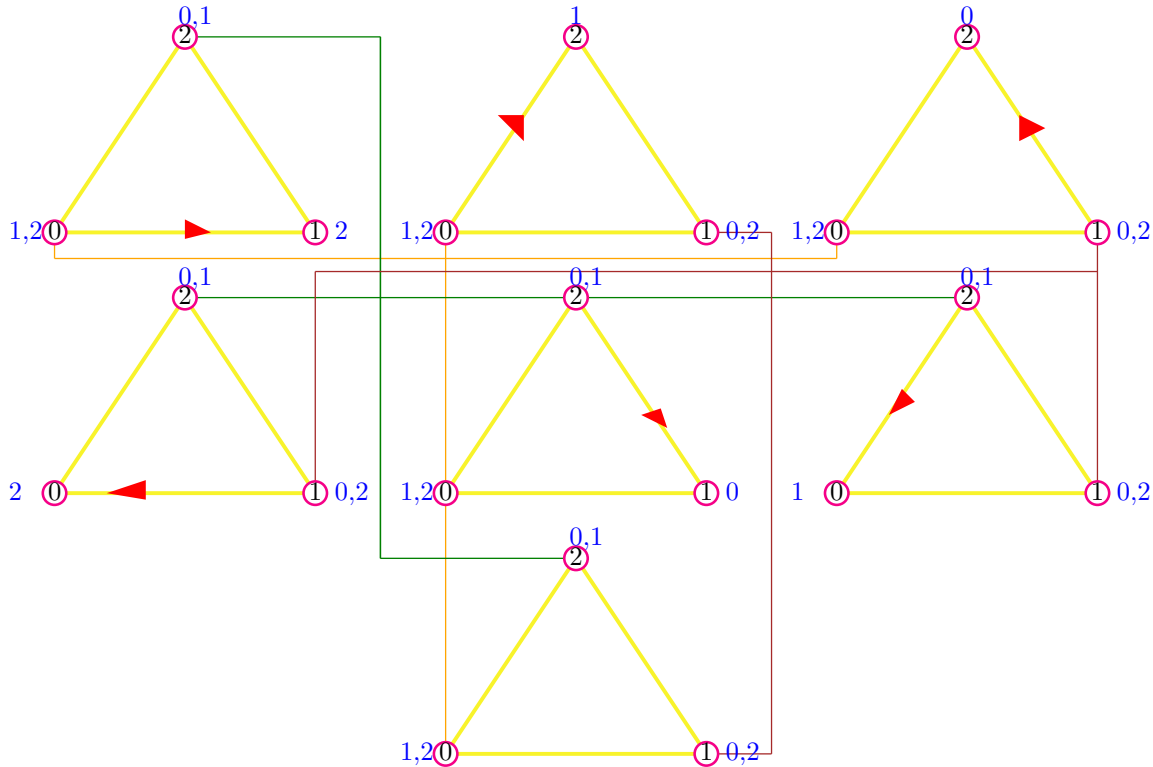
P es un conjunto de procesos no de mensajes, pero entendemos que lo que en realidad se pierde es un mensaje de algún proceso p_i , no todo el proceso. Por ejemplo, está el caso donde todos los mensajes de p_1 y p_2 llegan a sus destinos, pero se perdió alguno de p_0 , digamos el que tenía como destino p_1 , entonces el mensaje de p_0 a p_2 sí llegó. La siguiente gráfica muestra dicho caso.



La flecha roja no indica que la comunicación se haya roto, simplemente es para ver graficamente cual fue el mensaje no llegó.

Viendo el ejemplo anterior se observa que cada arista puede fallar dos veces, así el numero de casos a mostrar donde falla es $2 \cdot |E| = 2 \cdot 3 = 6$.

A continuación se muestran esos seis casos, junto al caso perfecto, añadiendo el requisito de la arista entre par de estados globales que no sean capaces de distinguir.



Se observa que para cada proceso cinco, donde para cada proceso uno de estos cinco es el estado global perfecto, de los siete estados globales es el mismo. Al solo conocer su estado local es imposible para el proceso saber que pasa con los demás procesos, ya sea que este en el caso perfecto o en el que un mensaje no llegó.

Por lo antes mencionado, dada la descripción del problema es imposible que los tres procesos escojan algún ID , si bien es cierto que existe el estado global donde no ocurren errores no sabemos si es el más frecuente o no; peor aún si todos los estados globales tienen la misma probabilidad de ocurrir y el proceso donde se ejecute el algoritmo, p_i , tiene como estado local la lista con los otros dos estados la probabilidad de que los demás no tengan la misma información que el proceso p_i es de .8. Por otro lado, podríamos esperar a tomar la decisión, el algoritmo mencionado no indica que se pueda hacer eso pero si nos lo planteamos salen otras cuestiones: si el estado p_i solo recibe un ID lo sensato sería que todos escojan ese, pero ahora surge la duda ¿De que manera les comunica que solo recibió uno?, ¿Qué tal si su mensaje indicando que solo recibió un ID no llega a todos?.

Resumiendo: En un sistema donde la comunicación no es confiable la decisión de escoger el mismo ID consistente con los otros dos procesos no es posible, por que no hay forma de asegurar que información tienen los demás.

2. Retomando el proceso de los dos enamorados con los mismos requerimientos vistos en clase, responda las siguientes preguntas:

- Suponga que las citas sólo se pueden realizar entre las 21:00 y las 22:00. ¿Tiene solución el problema en este caso?

Solución:

No, lo que hace es disminuir el margen de error. Ejemplo:

Antes el enamorado podía decirle que a las 10:00, ella le confirmaba, pero después le surge un imprevisto y le pide que mejor sea a las 14:00, ella no obtiene respuesta. El último mensaje forma parte del conjunto con el mínimo de mensajes para la solución al problema, es decir si el mensaje llega ambos se ven a las 14:00 hrs, si no él llegará a las diez y ella a las catorce. Para ambos no hay forma de actuar estando seguros de lo que hará o lo que conoce el otro. ¿Qué tal que la/el otra/otro envío un mensaje que no me llegó?

Con el planteamiento de este inciso puede que el le pida sea a las 21:05 ella confirma pero recuerda que no tiene permiso hasta las 21:45, supongamos que, de nuevo, no obtiene respuesta. Termina ocurriendo lo mismo.

El problema no radica en el rango de hora, si no en que el sistema de comunicación no es confiable, pues algún mensaje puede perderse en el camino.

- ¿Se puede resolver el problema de los dos amantes cuando se establece como requerimiento que, los amantes deben ser capaces de coordinar una hora para reunirse sólo cuando ningún mensaje se pierde, y en cualquier otro caso, ninguno de ellos debería presentarse?

Solución:

Si ellos son conscientes que hay mensajes perdidos puede que sí. Para empezar el requerimiento nos asegura que en caso de no haber mensajes perdidos ambos son capaces de coordinar una hora, es decir no hay fallas en la comunicación y si las llegase a haber, asumiendo que ellos lo saben, puede que los mensajes que no se enviaron no formen parte de la solución mínima, llamemosles mensajes basura. Así si ocurre una falla y todos los mensajes perdidos son mensajes basura ambos tenían la información necesaria para cenar juntos con lo que el fallo no repercutiría pero ellos al no saber mejor deciden no ir, por otro lado, si no todos los mensajes perdidos son basuras entonces al menos un mensaje es necesario para acordar la reunión; de nuevo asumimos que ellos saben que se perdió algún mensaje entonces no van, esta vez acertando pues el encuentro no se haría.

La respuesta asumiendo que ambos son conscientes de que existe pérdida de mensajes es *puede que sí* por que en el caso perfecto donde no hay pérdida de mensajes, ambos lo saben entonces se reúnen, **pero** al no ser un sistema confiable puede que este caso nunca se presente y no se lleve a cabo la cena, no logrando el cometido. Con lo que en realidad la respuesta es no.

Si ellos no son conscientes que hay mensajes perdidos la respuesta es definitivamente no pues nos encontramos en la misma situación que sin este requerimiento. Al no ser un sistema que garantice toda comunicación no pueden estar seguros si no se envió o recibió algún mensaje.

- Consideremos una variación: Los dos amantes se han dado cuenta de que no necesitan ponerse de acuerdo sobre una hora exacta para la reunión, está bien si sus horas de reunión son lo suficientemente cercanas. En otras palabras, cada uno debería eventualmente elegir un tiempo, de modo que los dos tiempos estén lo suficientemente cerca. ¿Se puede resolver su problema?

Solución:

La respuesta vuelve a ser no. Primero pense:

El problema nunca especifica cuando dos horas son suficientemente cercanas, entonces podría decir que dos horas son suficientemente cercanas en un rango de un día (24 hrs).

Pero luego me dí cuenta que eso no tiene sentido en la vida real, en el modo que alguien no va a esperar un día en un restaurante, y además que el sistema no sea confiable vuelve a ser un problema. Al pensar que dos horas suficientemente cercanas son de un día o hasta una semana solo hace lo contrario que el primer inciso de la pregunta dos, ampliar el margen de error.

Supongamos que un día es el tiempo suficientemente cerca para que se vean, ahora esta el escenario donde ella le pide se vean a las doce horas del lunes, él confirma pero después pide sea a las trece horas del mismo lunes, ella recibe el mensaje. Al estar en el rango de las veinticuatro horas ella sigue sin cambiar de hora, está dispuesta a esperar una hora. Pero ahora ella decide cambiar la hora a las nueve de la mañana del miércoles, solo que el mensaje se pierde, ya no entra en el rango de las veinticuatro horas, él debe cambiar su hora. Él cree que sigue en el rango así que no la cambia. Él llega el lunes a las trece, espera hasta las trece del martes y ella no llegó. Misión fallida.

3. Investigue y explica brevemente el protocolo TCP. ¿Es posible resolver el problema de los dos amantes si hay un canal TCP confiable entre ambos amantes?

Solución:

Primero hay que entender que TCP, protocolo de control de transmisión, es un modelo que busca estandarizar la conexión de computadoras. Originalmente contaba de cuatro capas:

4. *Aplicación*
3. *Transporte*
2. *Internet*
1. *Enlace*

Los cuales como podemos ver están enumerados de abajo hacia arriba, aunque en realidad la dirección depende si estás recibiendo o enviando tráfico(datos).

El protocolo se actualizó, ahora cuenta con una capa extra y renombraron la capa internet. Aquí la nueva lista y quienes la conforman:

5. *Aplicación*
En esta capa se encuentran los protocolos de las aplicaciones como lo son HTTP, FTP y SMTP.
4. *Transporte* Aquí se encuentran los protocolos de transporte, los más comunes son TCP y UDP. También se encuentran los puertos.
3. *Red* Se encuentra el protocolo de internet (IP) y los enrutadores.
2. *Enlace* Ethernet y switches.
1. *Física* Cables, dispositivos con adaptadores de red.

Como sirven las capas al enviar información:

Cuando alguien envía información está comienza en la capa cinco y va bajando, cada capa agrega su propia información en bits, pasar por las cinco capas se llama *encapsulación*.

En cuanto alcanzamos la capa física, la información es transmitida a través del dispositivo del receptor. Breve descripción del encapsulamiento:

5. *Aplicación*

En esta capa se recibe la información.

El paquete se ve así:

[DATA] y simplemente se le llama data(información).

4. *Transporte*

La información de transporte se añade, en este caso con un encabezado TCP. Cada vez que un encabezado se agrega éste contiene información específica. Los encabezados TCP tienen la información de origen, destino, números de puerto, secuencias de número y algunos bits más de información.

Aquí la información se ve así:

[TCP][DATA] y se llama segmento

3. *Red*

Se agrega el encabezado IP, que contiene las direcciones IP de origen y destino y más bits de información.

La información se ve así:

[IP][TCP][DATA] Y se llama Paquete

2. *Enlace*

Aquí se agrega un encabezado con las direcciones MAC de origen y destino más y un *trailer* que es información adicional que se agrega al inicio y al final de un bloque de información que se va a emitir, sirve para marcar el final de bloque y para manejar su información, este en particular lleva información para que el receptor pueda hacer manejo de errores y asegurarse de recibir bien la información.

La información se ve así:

[ETHERNET][IP][TCP][DATA][ETHERNET] Y se llama Frame.

1. *Física*

Se transmite físicamente.

El dispositivo receptor comienza a *decapsular* la información, chequea la dirección MAC de destino del frame, si el frame va dirigido a nuestro computador o dispositivo entonces sigue subiendo. Ahora el computador chequea si la dirección IP del paquete está dirigida a él, si es así sigue subiendo. Se lee la información de transporte y la información de la aplicación se envía a la aplicación receptora(WhatsApp, Browser, etc).

De acuerdo a lo entendido y a una de mis fuentes¹ TCP es un medio confiable para la recepción de datos, en el sentido que puedes saber quien la envía, si es para ti e incluso maneja errores para verificar que la información no este dañada, perdida, duplicada o entregada sin haberla pedido.

Lamentablemente el protocolo no arregla nuestro problema, de hecho en algún sentido lo habíamos dado como un hecho. El protocolo es lo que permite que haya la comunicación entre enamorada y enamorado, nunca especificamos que se comunicaran por cartas o mediante una persona, al menos yo siempre creí que era a través de dispositivos conectados a internet, que para funcionar necesitan algún protocolo(TCP).

¹<https://www.ibm.com/docs/en/aix/7.1?topic=protocols-transmission-control-protocol>

Lo que TCP garantiza es que los mensajes que envíe la enamorada no lleguen a otra persona por accidente y saber cuáles de los que recibe son de su enamorado, también garantiza que el mensaje de su enamorado no sea interceptado por un tercero y modificado (Esto último también depende un poco de la aplicación que se utilice, pero TCP implementa un pequeño sistema de seguridad), por último TCP asegura que el mensaje recibido sea el mismo al enviado.²

TCP únicamente garantiza que intentará un número indeterminado de veces la entrega de información, y que los frames serán entregados, en caso de encontrar al receptor, en el mismo orden en el que fueron enviados. TCP no influye en la decisión ni en los inconvenientes que tengan los al menos dos miembros de la comunicación, como es que se caiga la conexión, pérdida de dispositivos, etc. El problema de los enamorados radica más bien en el problema de la aplicación, no de el protocolo.

²<https://cs.stanford.edu/people/eroberts/courses/soco/projects/1999-00/internet/tcp.html>