

Ejercicio - Programación: Simulación de protocolos de control de acceso al medio ALOHA

1. Introducción

Los protocolos de control de acceso al medio permiten la compartición de un medio físico para la transmisión de múltiples fuentes. En estos medios compartidos la transmisión simultánea de dos o más fuentes provoca lo que se vienen a llamar "colisiones", durante las cuales la suma de las señales de las diferentes fuentes hace que un receptor no pueda recibir correctamente ninguna de ellas. Un protocolo de control de acceso al medio (MAC) para una tecnología de paquetes va a definir la forma de alternar el uso del medio compartido para lograr envíos con éxito de estos paquetes.

Según el protocolo de control de acceso al medio podremos obtener mejor o peor rendimiento del sistema, el cual podremos medir mediante la probabilidad de que un paquete colisione o con el tiempo hasta que un paquete logre atravesar el medio con éxito.

Algunos protocolos de control de acceso al medio permiten calcular una solución analítica con unas hipótesis más o menos razonables y herramientas matemáticas sencillas, mientras que otros se vuelven rápidamente problemas complejos o incluso intratables matemáticamente. Una alternativa para analizar el comportamiento del sistema es ponerlo en funcionamiento, lo cual claramente en las primeras etapas del desarrollo de un protocolo no es muy práctico. Otra alternativa es simular el comportamiento del sistema mediante un programa informático. Eso quiere decir simular los sucesos que se producirían en el sistema real (comienzo de un envío por una estación, finalización, comienzo de otro envío, comienzo de una colisión, etc). Mediante esta simulación podemos hacer una estimación de esos resultados de rendimiento (tiempo hasta que un paquete atraviese el sistema, probabilidad de que colisione, etc).

En este ejercicio se propone desarrollar pequeños simuladores para estimar o verificar resultados analíticos. Podremos por ejemplo cambiar condiciones del funcionamiento del sistema que lleven a un difícil cálculo analítico pero que se puedan simular con sencillez.

2. ALOHA puro

Cree un programa (en el lenguaje que prefiera) para simular el comportamiento de un enlace ALOHA (no ranurado). Los parámetros del programa serán:

- La tasa media de llegadas por unidad de tiempo del tráfico agregado al enlace
- El tiempo de transmisión (constante) de los paquetes
- El número de paquetes que se van a intentar enviar

Las llegadas representan los instantes en los que las estaciones tienen un paquete para enviar y van a comenzar la transmisión. Supondremos que hay muchas estaciones, con poco tráfico cada una, con lo que es seguro suponer que dos transmisiones cercanas en el tiempo provienen siempre de estaciones diferentes y pueden colisionar.

Vamos a ignorar el mecanismo de confirmaciones. Supondremos que o bien no existen o simplemente algunos de esos paquetes representados por las llegadas son las confirmaciones. Ignoraremos también las retransmisiones, es decir, el proceso de instantes de llegadas representa todos los paquetes que se intentan enviar (paquetes de datos, confirmaciones, retransmisiones).

Las llegadas siguen un proceso de Poisson de tasa el primer parámetro del programa. Eso quiere decir que los tiempos entre llegadas siguen una distribución exponencial con ese mismo parámetro y son independientes. En cualquier lenguaje de programación podrá encontrar una librería que le permita generar números aleatorios. Lo más habitual es que encuentre una función que le genere números aleatorios con una distribución uniforme, los cuales se pueden transformar de forma sencilla en otra distribución mediante la técnica de la transformada inversa [1]. Tenga la precaución de revisar el concepto de "semilla" en estas funciones de generación de números aleatorios.

Para cada paquete debe calcular si colisiona o no y llevar una cuenta de cuántos paquetes colisionarían, con lo que podrá estimar la probabilidad de colisión.

Como salida del programa muestre, para cada paquete de la secuencia de tiempos, si colisiona o no y la estimación hasta el momento (con los paquetes anteriores a él) de esa probabilidad de colisión. Con esa salida podrá ir viendo si el resultado se va estabilizando alrededor de un valor a medida que avanza la simulación y la probabilidad se estima con muchas más realizaciones del suceso de envío.

Puede correr la simulación para distintos valores de la tasa media de llegadas y del tiempo de transmisión de los paquetes y con ello generar los datos para hacer una gráfica de tráfico cursado frente a tráfico ofrecido. Puede automatizar esta tarea de correr múltiples simulaciones mediante otro programa o un script y comparar con resultados teóricos.

3. ALOHA: llegadas en fichero independiente

Para simplificar la depuración del programa puede hacer que los instantes de llegada aleatorios no se generen internamente, sino que sean valores que se van leyendo de un fichero. Modifique el programa para que el primer argumento sea el nombre del fichero que contiene esos valores, donde cada línea de texto es el tiempo entre una llegada y la siguiente, suponiendo que la primera llegada se produce en el instante 0. Eso quiere decir que hay una llegada más que el número de líneas del fichero. No habrá tercer parámetro del programa ya que el número de paquetes viene determinado por el número de líneas del fichero de entrada y el programa debe funcionar para cualquier tamaño del fichero de datos (eso quiere decir que debe funcionar para miles de millones de llegadas).

Para probar el programa tendrá que crear otro que genere el fichero con los tiempos entre llegadas. Ese programa tendrá como parámetros la tasa media y el número de llegadas y generará tiempos exponenciales independientes.

Podríamos generar ficheros de tiempos entre llegadas que no fueran exponenciales y el programa de simulación debería funcionar correctamente con ellos; con esto podremos ver si con otros procesos de llegadas se obtienen resultados similares de probabilidad de colisión o no.

4. ALOHA: tiempos de servicio variables

Continuando con el programa anterior modifíquelo para que el segundo parámetro sea también la ruta a un fichero, el cual contendrá los tiempos de transmisión de cada uno de los paquetes (en el mismo orden que los tiempos entre llegadas y con el mismo número de líneas que ese otro fichero).

Con esta modificación podría crear ficheros con tiempos de transmisión variables, diferentes entre los paquetes del mismo escenario, y ver si los resultados se parecen a lo que se obtenía con tiempos constantes. Por ejemplo, nos podemos preguntar si se obtiene la misma probabilidad de colisión promedio entre todos los paquetes que la que se obtiene en el modelo clásico de iguales tiempos de transmisión para una distribución de tiempo de transmisión que tenga como promedio el mismo valor de los tiempos constantes empleados antes.

Con igual tasa de llegadas e igual tiempo medio de servicio la carga ofrecida es la misma, por lo que intuitivamente podríamos esperar el mismo comportamiento de probabilidad de colisión. Sin embargo, en general paquetes más grandes (de mayor tiempo de transmisión) tienen una ventana de colisión más grande, así que intuitivamente se esperaría también que su probabilidad de colisión fuera mayor. En vez de calcular una probabilidad de colisión promedio independiente del tamaño (usando "tamaño" como equivalente a tiempo de transmisión ya que suponemos tasa de transmisión constante) podríamos calcular la probabilidad de colisión de los paquetes en función de su tamaño, o una aproximación para cuando su tamaño está entre un valor x y $x+d$. Estaríamos estimando una probabilidad condicionada al tamaño del paquete.

5. ALOHA: confirmaciones (Acknowledgments, ACKs)

Hemos supuesto que los ACKs eran parte de los intentos de envío que teníamos modelados en el proceso de llegadas. Ahora intentaremos introducir explícitamente las confirmaciones en el modelo de simulación.

El proceso de llegadas representará ahora solo los instantes en los que las estaciones intentan enviar un paquete nuevo. Si el paquete no colisiona supondremos que tras su tiempo de transmisión más un tiempo constante D se genera un nuevo paquete. Este nuevo paquete representa la confirmación, que no estaba en la secuencia de paquetes de llegadas original. Si el paquete de confirmación o el original colisionan no se producirá envío de un nuevo paquete de confirmación ni se intentará retransmitir el paquete (de momento dejamos esa parte fuera del modelo de simulación). El tiempo D intenta modelar el retardo desde la estación transmisora a la estación receptora.

El nuevo paquete de confirmación puede colisionar con cualquier otro paquete que se produzca dentro de su ventana de transmisión y es de tamaño constante. Por ejemplo, si supone el escenario original de paquetes de tamaño constante entonces el ACK tendrá el mismo tamaño que los de datos.

Fíjese en que ahora la carga sobre el sistema se ve incrementada con los paquetes de confirmación. Sin embargo, a la hora de calcular la probabilidad de colisión nos puede interesar la probabilidad de cualquier paquete o solo la de paquetes de datos (los paquetes originales del fichero

de entrada). De cara al usuario lo que importa es si llegan o no sus paquetes, así que de momento calcularemos solo esa probabilidad, pero tenga en cuenta que si incluyéramos el mecanismo de retransmisiones la colisión de una confirmación también provoca una retransmisión.

Por simplicidad puede suponer todos los paquetes (datos y confirmaciones) del mismo tamaño.

6. ALOHA: retransmisiones

Añada al modelo de simulación anterior que en caso de no recibirse la confirmación se produzca una retransmisión. Puede hacerlo de forma simplificada implementando que si una confirmación sufre una colisión se genera un nuevo paquete de usuario tras un tiempo T . Si el que sufre una colisión es el paquete de datos se generaría un nuevo paquete de usuario tras un tiempo $D+T$.

Con esto se modela la carga extra introducida por las retransmisiones. Sin embargo, este modelo simplificado no limita el número de intentos de retransmisión. Reflexione sobre cómo podría implementar que un mismo paquete no se intente retransmitir más de N veces y que tras N intentos con error (colisión del paquete o de la confirmación) se dé por perdido.

7. ALOHA: modelos matemáticos

No se sugiere leerlo, pero si quieren hacerse a la idea de la complejidad que puede tener analizar estos sistemas de forma matemática pueden oír algo como [2]. En la asignatura no vamos a hacer nada parecido y como mucho en alguna asignatura futura se verá en un tema un poco sobre la intuición por detrás de este tipo de soluciones. Tradicionalmente es una problemática tratada en nivel de máster y con conocimiento de más herramientas matemáticas generales (transformadas) y estadísticas en especial.

[1] https://es.wikipedia.org/wiki/M%C3%A9todo_de_la_transformada_inversa

[2] R. T. B. Ma, V. Misra and D. Rubenstein, "An Analysis of Generalized Slotted-Aloha Protocols," in *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 936-949, June 2009, doi: [10.1109/TNET.2008.925633](https://doi.org/10.1109/TNET.2008.925633).
<https://www.ee.columbia.edu/~tbma/StudyArea/Publications/ToNAloha.pdf>