

La norma IEEE 802.6 (DQDB) operando en tiempo real duro

Rodrigo Santos[†], Marcelo Zambón[†], Javier D. Orozco[‡], Jorge Santos[‡]

[†]Dep. Ingeniería Eléctrica ; [‡]Dep. Ingeniería Eléctrica , Instituto de Ciencias e Ingeniería de Computación, CONICET.

Universidad Nacional del Sur, Av. Alem 1253, 8000 Bahía Blanca

E-mail {iesantos, ieorozco}@criba.edu.ar

RESUMEN: En trabajos previos, se ha demostrado que la norma IEEE 802.6 (DQDB) no puede implementar las disciplinas de Rueda Cíclica Justa o de Periodos Monotónicos Crecientes, salvo que se introduzcan modificaciones tan sustanciales que resulten no implementables con los mecanismos normalizados. El objeto del presente trabajo es analizar las condiciones de diagramabilidad de la red sin alterar sus protocolos de acceso al medio. Se encuentra una expresión para el período mínimo de los mensajes de cada nodo en función de la longitud del mensaje, de su distancia al extremo generador de ranuras, del número de nodos aguas arriba y de la función trabajo de los nodos aguas abajo.

1. INTRODUCCIÓN

Se dice que un sistema es de tiempo real cuando los resultados no sólo deben ser correctos desde un punto de vista aritmético-lógico, sino que además deben ser producidos antes de un determinado tiempo, llamado *tiempo de vencimiento* o simplemente *vencimiento* (*deadline*) [1]. Cuando el sistema no puede perder ningún vencimiento se dice que es de tiempo real duro. Si tolera la pérdida de algún vencimiento se llama de tiempo real blando. En este trabajo se analiza la red metropolitana especificada en la norma IEEE 802.6 (DQDB, Dual Queue Dual Bus) [2] operando en tiempo real duro. La importancia del tema es evidente a poco que se considere la creciente utilización de este tipo de red para la transmisión de audio, video o información multimedial, todas ellas aplicaciones en tiempo real duro. Como tales, en los nodos se generan mensajes periódicos, cada uno de los cuales debe ser transmitido antes de que se genere el mensaje siguiente. Por este motivo, el sistema opera en tiempo real duro con vencimientos iguales a los respectivos períodos. Cuando el sistema cumple con todas las constricciones de tiempo se dice que es *diagramable*.

El protocolo 802.6 nació como respuesta a las necesidades de redes metropolitanas (más conocidas como MANs, Metropolitan Area Networks) que permitieran intercomunicar confiablemente redes locales en un área mediana, con longitudes de red en el orden de centenares de km. La topología es la de una doble barra, en cada una de las cuales la información fluye en sentido opuesto.

DQDB pertenece a un tipo genérico de sistemas, aparentemente distintos pero conceptualmente isomorfos, caracterizados por tener múltiples usuarios que comparten el uso de un único recurso (sistemas multiusuario/monorecurso) [3]. Cuando se presentan conflictos entre dos o más usuarios por el uso del recurso, el mecanismo de prioridades del sistema debe arbitrar en el conflicto. Para ello utiliza una disciplina de prioridades, la cual genera una pila, real o virtual, en la que los usuarios están ordenados de acuerdo a la prioridad asignada a cada uno. Los que posean prioridades mayores se encontrarán al tope de la pila y los de prioridad más baja en el fondo. Las disciplinas más empleadas son la *Rueda Ciclica Justa* (RJC, *Fair Round Robin*) y la de *Periodos Monotónicos Crecientes* (PMC, *Rate Monotonic*) [4].

En la primera, la pila de prioridades se va modificando después de cada asignación del recurso. El usuario que termina de utilizarlo pasa a tener la prioridad más baja y el que le seguía en la pila, la prioridad más alta. De esta forma todos los usuarios poseen en algún momento la máxima prioridad.

En la segunda disciplina, la pila de prioridades se arma ordenando los usuarios por periodos monotónicos crecientes. De esta forma los usuarios con menor período obtienen prioridades más altas. La pila se arma sólo una vez, en la inicialización del sistema. Esta disciplina ha sido adoptada por el Departamento de Defensa de los EE. UU. y debido a esto se ha convertido en una norma *de facto*.

El objetivo del presente trabajo es determinar las condiciones de diagramabilidad de una red DQDB operando en un ambiente de tiempo real duro. Luego de esta introducción, el resto de este trabajo se organiza como sigue: En la sección 2 se hace una descripción breve del protocolo DQDB, incluido el mecanismo de prioridades y los problemas que el mismo presenta. La sección 3 reseña algunas posibles soluciones a esos problemas, propuestas en otros trabajos, todas las cuales implican modificaciones o agregados a la norma. En la sección 4 se determina la mejor condición de diagramabilidad que la norma ofrece en su redacción actual. En la sección 5 se ilustra lo anterior con un ejemplo y en la sección 6 se exponen las conclusiones.

2. EL PROTOCOLO DQDB

En DQDB, cada nodo está conectado a dos barras, en cada una de las cuales la información fluye en sentido contrario (Fig. 1). Teniendo en cuenta el sentido de flujo, es posible entonces hablar de nodos *aguas arriba* o *aguas abajo* de un nodo dado. Es evidente que los nodos aguas arriba en una barra son aguas abajo en la otra.

Los datos se transmiten en bloques de tamaño fijo llamados *ranuras*, por lo que se dice que tanto el tiempo como las barras son *ranurados*. Al comienzo de cada barra hay un nodo especial, el *generador de ranuras*, que se encarga de emitir continuamente ranuras libres. Existen dos tipos de ranuras: *prearbitradas* y asignadas por *reservas*. Las segundas proveen un mecanismo de acceso al medio que es no sólo controlado sino también adaptativo a la demanda. Por tener estas características, imprescindibles en un ambiente de tiempo real duro [4], son las analizadas en este trabajo.

La longitud de una ranura es de 53 bytes, de los cuales el primero se utiliza para control. Su estructura es la siguiente:

- 1 bit, BSY, que indica si la ranura está libre u ocupado por algún mensaje
- 1 bit, SLOT TYPE, que indica el tipo de ranura
- 2 bits, RSVD, reservados para uso futuro
- 1 bit, PSR, que indica si el segmento de información puede ser borrado o no
- 3 bits, REQ, que indican el orden de prioridad de los pedidos.

Dado el sentido unidireccional del flujo de información en cada una de las barras, es evidente que un nodo sólo puede transmitir hacia nodos situados aguas abajo. En consecuencia, para los nodos situados a su izquierda deberá utilizar una barra y para los situados a su derecha, la otra barra.

El método de acceso al medio es mediante reservas. Las mismas se transmiten por una barra mientras que los mensajes son transmitidos por la otra. Obviamente, los nodos que reciben la reserva en una barra, están aguas abajo en la misma pero aguas arriba en la otra, por la cual transmiten los mensajes. Por ejemplo, para transmitir mensajes a nodos situados a su izquierda, el nodo *i* coloca sus reservas en la barra A y sus mensajes en la B. Para nodos situados a su derecha, las reservas utilizan la barra B y los mensajes la A.

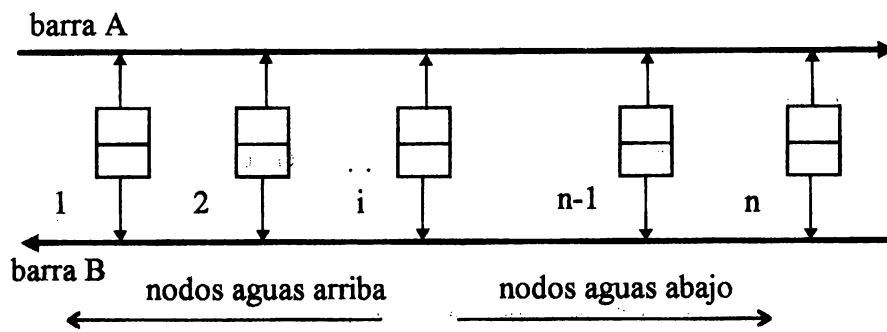


Fig. 1

Dada la dualidad del sistema, es suficiente considerar el tráfico de mensajes sobre una sólo de las barras. Las conclusiones obtenidas serán válidas para el caso dual en el cual esa barra funcionará como barra de reservas. En todo lo que sigue consideraremos el caso en que la barra A es utilizada para mensajes y la B para reservas. Agudas abajo y aguas arriba son expresiones que se refieren a la barra A.

Según la norma, cada nodo tiene dos contadores: Uno llamado *request counter*, que se encarga de contar los pedidos de mensajes que vienen por la barra B, y otro, llamado *countdown counter*, que cuenta las ranuras libres que ve pasar por la barra A. El conjunto de *request counters* (RC) y *countdown counters* (CD) de todos los nodos conforman una cola distribuída que decide qué nodo tiene la posibilidad de transmitir.

Si un nodo no tiene ningún mensaje para transmitir está en un estado denominado *en espera*. Durante el mismo, el nodo se dedica a escuchar la red y actualizar su RC de la siguiente manera: Por cada ranura marcada con un pedido que pase por la barra B el contador es aumentado en 1; las ranuras libres que el nodo ve pasar por la barra A producen un decremento unitario en el mismo contador, hasta que su contenido llega a cero. Si el RC vale cero, el paso de una ranura libre no tiene efecto sobre el contador.

Cuando un nodo quiere transmitir un dato, se coloca en la cola distribuída ubicando su pedido en la barra B, haciendo conocer de esa manera su reserva a los nodos aguas arriba en la barra A. En ese momento se transfiere el valor de RC al CD (que es el que llevará la cuenta de la posición del nodo en la cola distribuída de transmisión) y coloca el RC en cero. A partir de aquí, el CD se decrementa con cada ranura libre que el nodo ve pasar por la barra A y RC se incrementa con cada pedido que ve pasar por la barra B. Cuando el CD vale cero, el nodo está en condiciones de transmitir. Un nodo no puede volver a poner en cola un mensaje nuevo hasta que no haya transmitido el anterior.

Hasta aquí se ha considerado que todos los nodos tienen la misma prioridad. Sin embargo, la norma prevee tres niveles, identificados por un 1 en alguno de los bits REQ y por 0s en los otros dos. En el caso de prioridades múltiples, cada nodo lleva un CD y un RC por cada nivel de prioridad. Si el nodo está ocioso, el RC de cada prioridad se incrementa por los pedidos de prioridad mayor o igual a la suya. Si tiene un mensaje en cola, el RC de cada prioridad se incrementa por pedidos de prioridad igual a la suya y el CD es incrementado por pedidos de prioridad mayor a la suya.

Este mecanismo de prioridades, sin embargo, tiene algunos inconvenientes. El principal motivo es que los nodos no poseen información completa del estado de la red en cada momento y, debido a ello, tienen que tomar continuamente decisiones distribuidas con información incompleta. Esto, unido a que no se puede pedir otro segmento hasta que haya sido transmitido el segmento anterior, provoca que bajo determinadas circunstancias, no se garantice el ancho de banda necesario a los nodos de mayor prioridad. Los problemas aumentan cuando un nodo de prioridad alta está ubicado aguas abajo de un nodo de prioridad menor, muy cargado. El nodo fuertemente cargado hace pedidos cada vez que puede y como el nodo de prioridad más alta está a mayor distancia del generador de ranuras, tiene menor posibilidad de conseguir una ranura libre que el nodo de menor prioridad. Esta posibilidad decrece más cuanto mayor es la distancia entre el nodo y el generador de ranuras.

3. POSIBLES SOLUCIONES AL PROBLEMA DE PRIORIDADES EN DQDB

El problema de asignación de anchos de banda a las distintas prioridades en DQDB en particular, y en las redes de doble barra en general, ha sido bastante tratado en la bibliografía [6, 7, 8].

Sha propone un *Protocolo Coherente* [5] en el cual el ancho de banda de un nodo es independiente de la posición del nodo en la red. También garantiza una tasa constante de transmisión luego de un retardo inicial, que está acotado. Esto es muy deseable en transmisión de datos que requieren un ancho de banda constante, como por ejemplo, multimedia. Sha define la coherencia como el conjunto de propiedades que permiten establecer un mapa entre una diagramación centralizada usando una cierta disciplina y la diagramación en DQDB utilizando la misma disciplina. Partiendo de una definición de los principales aspectos que debería presentar una red que soporte transmisión en tiempo real, a saber: *operación*

predecible, alto grado de diagramabilidad, ancho de banda independiente de la posición del nodo en la red y estabilidad bajo efectos transitorios de sobrecarga, se demuestran las condiciones necesarias para la coherencia de las colas DQDB con la disciplina PMC. Las mismas, sin embargo, no son obtenibles con el protocolo en su versión actual y, de hecho, requieren prácticamente un nuevo protocolo, que es el propuesto en el trabajo.

Bisdikian *et al* [6] proponen utilizar los dos bits reservados del campo de control de DQDB para crear un campo de reserva de prioridades, en cierto modo similar al campo homónimo en 802.5. Esto permite que en cada momento la prioridad más alta disponga de todo el ancho de banda por el tiempo que lo necesite. Cada nodo informa a los otros la prioridad más alta que tiene en cola para transmitir, y todos los nodos deben ceder el ancho de banda disponible al tráfico de mayor prioridad. Este mecanismo se llama *mantenimiento de prioridad* y consta de un bit por nivel de prioridad, excepto para la prioridad cero, que no tiene asignado ninguno. El campo de bits resultante se llama *campo de prioridades*.

De acuerdo con este protocolo, cada nodo hace sus pedidos de la misma forma que en DQDB, pero además sobrescribe el campo de prioridades con el mayor nivel de prioridad que tenga en cola para transmitir, haciendo la unión lógica de éste (OR) con el valor del campo de prioridades de la ranura que ve llegar por la barra B. Si el resultado de esta operación es 10 o 11, la prioridad más alta aguas abajo en la red es 2. Si es 01, la mayor prioridad para transmitir es 1, y si ese valor es 00, sólo hay mensajes de prioridad cero en cola para transmitir. Cada nodo sigue sobrescribiendo el campo de prioridades de las ranuras que ve pasar por la barra B mientras tenga por lo menos dos segmentos en cola para transmitir. Luego continúa escribiendo el valor de la mayor prioridad que tenga por lo menos dos segmentos en cola para transmitir. Si un nodo ve en el campo de prioridades de la ranura entrante un valor mayor a la prioridad que él tiene para transmitir, detiene los procesos de actualización de contadores de transmisión (CD), lo que produce el efecto de un bloqueo de las prioridades inferiores en beneficio de las superiores.

Si bien este método no tiene las ventajas del de Sha, permite aumentar significativamente la performance de DQDB sin alterar mayormente la norma. La única modificación sería la asignación de destino a los dos bits RSVD, por ahora reservados para usos futuros. No implementa, sin embargo, ni RCJ ni PMC.¹

Saha *et al* [7] proponen en su trabajo un método de asignación de ancho de banda basado en reservas cíclicas. A pesar de que dicen que el mecanismo se puede implementar con

pequeñas modificaciones a la norma, no explican cuales serían y por cierto que para generar ciclos de reserva y transmisión de modo de no desperdiciar ancho de banda, la longitud de los mismos debería ser adaptiva, y esto sin lugar a dudas no resulta una modificación sencilla.

4. ANÁLISIS DE DIAGRAMABILIDAD EN EL CONTEXTO DE LA NORMA ACTUAL

Como se demuestra en el trabajo de Sha, DQDB no puede implementar RCJ porque no forma una pila de prioridad sino una cola distribuida no-coherente, en la cual el ancho de banda disponible por un nodo depende de la distancia del mismo al generador de ranuras.

La coherencia con PMC, por otra parte, requiere tantas modificaciones a la versión actual de la norma, que de hecho se está hablando de una norma básicamente distinta.

En lo que sigue se tratará de determinar las condiciones de diagramabilidad de DQDB operando en un ambiente de tiempo real duro. A tal efecto, se aceptarán las siguientes hipótesis:

- a) La unidad de tiempo es la duración de una ranura. Esta unidad será indivisible. Es lo mismo decir “una ranura” o “el tiempo de duración de una ranura”.
- b) La distancia entre nodos se medirá en ranuras y en todos los casos la distancia será una cantidad entera de ranuras.
- c) Los procesos de transmitir un segmento, efectuar reservas y pedidos, tienen duración despreciable y ocurren al comienzo de la ranura.
- d) Los mensajes están caracterizados por tres parámetros: T (período), C (número de segmentos de una ranura de duración que conforman el mensaje) y D (*deadline*, o vencimiento).
- e) Los períodos de los mensajes se expresan como un número entero de ranuras.

En lo que sigue utilizaremos la siguiente notación:

- d_i indica la distancia medida en ranuras desde el nodo i hasta el generador de ranuras de la barra de mensajes A .
- R_i tiempo máximo que debe esperar un segmento de mensaje para ser transmitido.

Consideraremos a la red operando con una prioridad única para todos los nodos.

Lema

El tiempo máximo que un nodo debe esperar desde que reserva un segmento hasta que recibe una ranura libre en la cual puede transmitirlo es función de la ubicación en el medio y está dado por:

$$R_i = 2 \cdot d_i + (i - 1)$$

Para demostrar el lema procederemos por partes. En primer lugar consideraremos el primer término. Desde el momento que el nodo genera el mensaje y coloca su reserva en la barra B hasta que ésta alcanza el extremo de la barra, transcurren d_i unidades de tiempo. En ese instante se genera una ranura libre que teóricamente debería usar el nodo que realizó el pedido. Para que esta ranura lo alcance transcurrirán otras d_i unidades de tiempo, $2 d_i$ en total. Para el segundo término debemos recurrir a la máquina de estados de la norma 802.6 que implementa las colas distribuídas. Cuando un nodo genera un mensaje y la ranura que pasa por la barra B en ese instante ya tiene el campo de pedidos ocupado, el nodo ubica primero su mensaje y luego el del nodo aguas abajo. Si todos los nodos que se encuentran aguas arriba hacen esto, el nodo sufrirá una demora adicional de $(i-1)$ ranuras. \square

Teorema

Es una condición suficiente para garantizar la diagramabilidad de una red DQDB de n nodos operando con un único nivel de prioridad que:

$$\forall i = 1, 2, \dots, n \quad T_i \geq \min \{t \mid t = C_i \cdot \max(R_i, 1) + \sum_{j=i+1}^n C_j \left\lceil \frac{t}{T_j} \right\rceil\}$$

Demostración

El primer término en la desigualdad se demuestra con el lema, si en particular el nodo que consideramos es el primero, que puede coincidir con el generador de ranuras, entonces en ese caso $R_1=0$. Debido a esto se toma siempre el máximo entre R_i y 1 para calcular el período mínimo que puede tener la generación de mensajes en el nodo. Ahora si $T_1=C_1$ entonces la red estaría saturada con el primer nodo y ningún otro nodo podría transmitir mensajes sin ocasionar la pérdida de la diagramabilidad. La incorporación del segundo término, la función trabajo [4] de los nodos aguas abajo, garantiza que cada nodo vaya dejando suficientes ranuras libres para que aquéllos puedan también transmitir cumpliendo sus propias

constricciones de tiempo. El peso del segundo término va disminuyendo a medida que nos acercamos al final de la barra.

Corolario

El máximo Factor de Utilización (FU) admisible para el último nodo de una red DQDB operando con una única prioridad es:

$$FU_n = \frac{1}{R_n}$$

La demostración es trivial y surge a partir del teorema. El mínimo período que puede tener el nodo n es:

$$T_n = C_n \cdot R_n$$

$$FU = \frac{C_n}{T_n}$$

$$FU = \frac{1}{R_n} \quad \square$$

El teorema muestra que a pesar de haber sólo un nivel de prioridad en la red, la posición de los nodos en la misma determina el máximo factor de utilización disponible para cada uno de ellos de modo que se garanticen todos los vencimientos. Como se ve los nodos más cercanos al generador de ranuras pueden llegar a tener factores de utilización muy elevados mientras que los nodos más alejados cada vez se ven más relegados.

Excepto en el caso de que todos los nodos de la red estuvieran a menos de una ranura de distancia entre ellos, el primer término hace que los períodos de los nodos deban aumentar a medida que se alejan del generador de ranuras. Debido a esto, si se deseara distribuir equitativamente el ancho de banda disponible en la red, de modo tal que todos los nodos tuvieran el mismo factor de utilización, el factor de utilización total resultante sería muy bajo.

Los niveles de prioridad que la norma propone y el manejo que de los mismos se hace no ayudan a mejorar en nada la situación descripta.

5. EJEMPLO

En esta sección mostraremos como a partir de la asignación del período del último nodo de la red, a medida que nos movemos hacia el generador de ranuras se incrementa el factor de utilización. Luego mostraremos que si se da a todos la misma utilización la red es subutilizada.

La red tiene una longitud de 17 ranuras y está conformada por cinco nodos. Supongamos que todos los nodos desean transmitir cinco segmentos y evaluemos la inecuación del teorema para calcular los períodos mínimos de cada uno. Los resultados se muestran en la siguiente tabla

nodo	C	T	d	R	FU
1	5	25	0	0	0.200
2	5	70	5	11	0.071
3	5	110	9	20	0.045
4	5	150	13	29	0.033
5	5	190	17	38	0.026

El factor de utilización total de la red en este caso es 0.375.

Si todos los nodos permanecieran con los parámetros que se muestran en la tabla, el primer nodo podría transmitir 50 segmentos cada 70 ranuras con lo que elevaría el FU de la red a 0.889, y aún con este FU la red sería diagramable.

Si se intentase una distribución equitativa del ancho de banda disponible asignando a todos los nodos de la red el mismo FU, éste debería ser igual al del último nodo con lo cual el FU global es bajo. En el ejemplo, 0.130.

Esto muestra que en DQDB aún cuando se utilice un sólo nivel de prioridad, la posición del nodo dentro de la red actúa como un elemento priorizante a diferencia de lo que ocurre en redes metropolitanas como FDDI que por tratarse de un anillo con ficha la posición dentro de la red no es relevante.

Es claro que el comportamiento de DQDB no es lo que se espera de una red metropolitana en la que la asignación de ancho de banda y del esquema de prioridades debiera ser independiente de la posición que en la misma ocupan los nodos.

6. CONCLUSIONES

El mecanismo que implementa DQDB para acceso al medio no es adecuado para la transmisión de mensajes en tiempo real, por lo que numerosas modificaciones han sido propuestas en la bibliografía.

En DQDB no se pueden aplicar ninguna de las dos disciplinas de prioridades más utilizadas, esto es ni RCJ, ni PMC. Además, aunque pudiera implementarse esta última, el escaso número de niveles de prioridad conduciría a granularidades muy gruesas.

El período mínimo de generación de mensajes de tiempo real en cada nodo es función de la distancia del nodo al extremo generador de ranuras. El hecho de que la topología de la norma sea de barra real hace prácticamente imposible una modificación de la distribución de los nodos por recableado para ajustar las prioridades. Aún así, sin utilizar el mecanismo de prioridades de la norma, en este trabajo se ha presentado una condición que, de verificarse, garantiza la transmisión de los mensajes antes de su vencimiento, cumpliendo, en consecuencia, todas las constricciones de tiempo real. Se debe notar que la utilización del mismo no mejora la dependencia frente a la ubicación en el medio de los nodos.

REFERENCIAS

- [1] J. A. Stankovic, A serious problem for next generation systems, *IEEE computer* 21 (10) (1988) 10-19.
- [2] IEEE Std 802.6 - 1990, *IEEE Standards for Local and Metropolitan Area Networks: Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)*, New York: IEEE, July 3, 1991.
- [3] J. Santos, J. Orozco, Rate monotonic scheduling in hard real-time systems, *Information Processing Letters*, 48 (1993) 39-45.
- [4] J. Santos, M. L. Gastaminza, J. Orozco, D. Picardi y O. Alimenti, Priorities and protocols in hard real-time LANs, *Computer Comm.* 14 (9) (1991) 507-514.
- [5] L. Sha, S. S. Sathaye, J. K. Strosnider, Analysis of Dual-Link Networks for Real-Time Applications, *Carnegie Mellon University, Internal Report*, (1994)
- [6] C. Bisdikian, A. N. Tantawy, A Mechanism for Implementing Preemptive Priorities in DQDB Subnetworks, *IEEE Transactions on Communications*, 42 (2/3/4) (1994) 834-839.

[7] D. Saha, M. Saksena, S. Mukherjee, S. K. Tripathi, On Guaranteed Delivery of Time - Critical Messages in DQDB, *Department of Computer Science, University of Maryland, College Park, Internal Report, (1994)*