

DQDB における帯域ロスのない動的な帯域配分のアルゴリズム*

4U-1

清松 和明 林 正薫 Bongkwan Cho 王 士杰 松下 温†

慶応義塾大学‡

1 はじめに

現在、DQDBは、MAN(Metropolitan Area Network)の通信プロトコルとしてIEEE802.6で標準として採択されている。また、DQDBでは、ノードの位置関係による不公平性を是正するため、BWB(Bandwidth Balancing)と呼ばれるメカニズムが採用されている。BWBでは、まず、負荷の低いノードに帯域を割り当て、次に、残りの帯域を負荷の高いノードに均等に割り当てる。また、この際、帯域の損失が生じる。

そこで我々は、帯域の損失をなくし、かつ各ノードの負荷に比例して帯域を割り当てるNPA(No slot waste and Proportional Assignment scheme in DQDB network)方式を提案する。

2 DQDB

DQDB網は、二重バスの構造になっており、図1に示すように伝送方向が相異なる二つ(BUS A、B)の単方向バスを持ち、また、すべてのノードは両バスに接続されている。図1からわかるように、バスAとバスBの動作は対称的なものである。そこで、説明を簡単にするため、すべてのノードはバスAを用いてデータの転送を行い、バスBを用いてリクエストを転送するものとする(バスAをデータバス、バスBをリクエストバスと呼ぶことにする)。

バスの先頭には空きスロットを周期的に生成するHOB(Head of Bus)があり、ここからバスに空きスロットを流す。ここで、HOBに近いノードを上流のノード、遠いノードを下流のノードと呼ぶことにする。データを転送するノードは、転送先のノードによって適当なバスを選択し、そのバスの空きスロットを使ってデータの転送を行う。従って、ノード間で全二重方式の通信が可能であり、データ転送に関しては、二つのバスの動作は独立である。

各ノードのチャネル使用を制御するため、各スロットのACF(Access Control Field)には、BUSYビットと

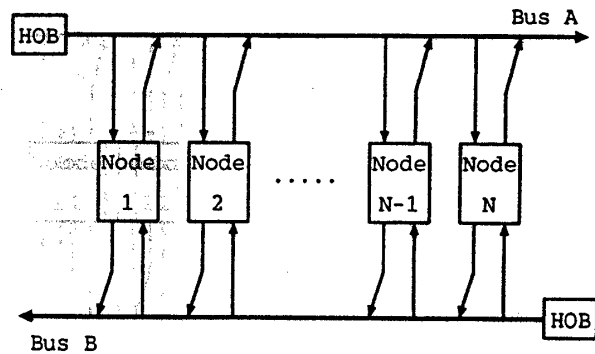


図1: DQDB網の構成

REQビットがある。また、各ノードには、両バスに関するRQカウンタとCDカウンタがある。この二つのビットと二つのカウンタを使って、データの転送は行われる。実際に転送方法は次のようになっている。

データの転送を行いたいノードは、まず、データを44バイト以下のセグメントに分割する。次に、リクエストバス上の1つのスロットのREQビットをセットし、上流のノードに空きスロットを要求する(この動作をリクエストという)。下流のノードからリクエストが来たノードは、RQカウンタの値を1増やし、データバス上に空きスロットが来るとRQカウンタの値を1減らす。データが発生したノードは、RQカウンタの値をCDカウンタにコピーし、RQカウンタの値を0にリセットする。このような状態のノードでは、データバスに空きスロットが来るとCDカウンタの値を1減らし、この値が0であるなら、実際に、セグメントの転送を行う。0でないなら、その空きスロットを使用することはできない。セグメントの転送の終了後、まだセグメントの転送を行いたい場合は、新たに上流にリクエストを転送しなければならない。つまり、1つのセグメントに対して1つのリクエストが転送されるわけである。

3 BWBメカニズム

DQDBでは、下流からのリクエストが上流のノードに届くまで時間がかかる。このため、上流のノードは、下流のノードと比べて、より早くより多くの空きスロット

*Proportional assignment scheme without Bandwidth loss in DQDB

†Kazuaki Kiyomatsu Jeonghun Lim Bongkwan Cho Shih-Chieh Wang Yutaka Matsushita

‡Keio University

トを使用できることになる。そしてこのことは、ネットワーク全体の負荷が大きくなると、またネットワークの規模が大きくなるとより顕著になる。そこで、このような不公平性を是正するため BWB が採用されている。

BWBでは、負荷の低いノードには帯域を要求通り割り当て、負荷の高いノードには残りの帯域を均等に配分する。この際、次の2通りの方法がある。1つ目の方法では、各ノードはある定数 M 個のスロットを使用したら、意図的に自分のノードの RQ カウンタの値を1増やす。もう一つの方法は、各ノードは GATE を持ち、 M 個のスロットを使用したノードは一時的にリクエストが送れなくなる(この状態を GATE が閉まっているという)。このような状態のノードは、RQ カウンタの値が0で、かつデータベースに空きスロットが来た時、初めてリクエストが送れるようになる(この状態を GATE が開いたという)。この際、この空きスロットを使用することはできない。結局、どちらの方法でも、各ノードは M 個のセグメントを転送したら1つの空きスロットを意図的に使用せずにやり過ごすことになる。

例えば、3つの負荷の高いノードがデータの転送を行い M の値が4の時、各ノードは $4/13$ の帯域を獲得し、帯域の損失は $1/13$ になる。また、 M の値が8の時は、各ノードは $8/25$ の帯域を獲得し、帯域の損失は $1/25$ になる。このことからわかるように、 M の値が大きいくほど帯域の損失は小さくなる。しかし、 M の値が大きいくほど各ノードのスループットが一定になる(定常状態になる)のにかかる時間は長くなる。

4 NPA 方式

この方式では、(1) 帯域の損失をなくす (2) 各ノードの負荷に比例して帯域を割り当てる という2つのことを目的としている(本方式は BWB の GATE 方式を利用している)。(1) は、意図的に通過させる空きスロットが下流のノードで使用されるかどうかかわかれれば良い。そのため、各スロットの ACF に新たに CONTROL ビットを加える。 M 個目のセグメントを転送したノードは、GATE を閉めると同時に、スロットの CONTROL ビットをセットする。CONTROL ビットがセットされているスロットを見た下流のノードは、そのノードの GATE が閉じているか、まだ転送したいセグメントが M 個以上ある場合、上流にリクエストを転送する(この際、GATE は開けられる)。結局、ノードは、RQ カウンタの値が0でデータベースに空きスロットが来た時か、CONTROL ビットがセットされたスロットが来た時に GATE を開けることができる。なお空きスロットが来て GATE を開けた場合、その空きスロットを使用

することができる。(2) は、各ノードの負荷に応じて各ノードの M の値を変えれば良い。そのため、各ノードに新たに SL カウンタと SG カウンタを加える。SL カウンタの値はデータベースをスロットが1つ通過するごとに1増え、SG カウンタの値はそのノードでセグメントが1つ発生するごとに1増える。各ノードは、スロットがネットワークを1周するごとに $M \cdot SG / SL$ の値を計算し、この値を新たに M の値とする(この際、SL カウンタと SG カウンタの値は0にリセットされる)。図2は、この方式について、ノード1が時刻0から16000の間 $0.4 \text{ segments/slot}$ の負荷で、ノード2が時刻2000から12000の間 $0.6 \text{ segments/slot}$ の負荷で、ノード3が時刻4000から8000の間 $0.8 \text{ segments/slot}$ の負荷で転送を行うものとして、シミュレーションを行ったものである。

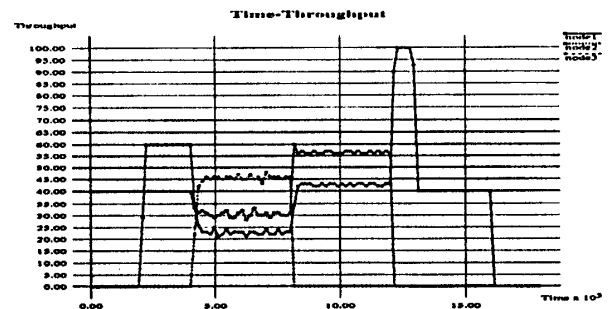


図2: NPA 方式のスループット特性

5 結論

本研究で提案した NPA 方式によって、BWB で生じる帯域の損失をなくし、かつノードの負荷に応じて帯域を割り当てることが可能となるといえる。

参考文献

- [1] Biswanath Mukherjee and Subrata Manerjee, "Alternative Strategies for Improving The Fairness in and An Analytical Model of DQDB Networks", IEEE INFOCOM, pp.879-888, Apr.9-11, 1991
- [2] Dionysios Karvelas and Michail Papamichail, "The No Slot Wasting Bandwidth Balancing Mechanism for Dual Bus Architectures", IEEE JOURNAL VOL.11, pp.1214-1228, October, 1993