Washington University in St. Louis
Washington University Open Scholarship

All Computer Science and Engineering Research | Computer Science and Engineering

Report Number: WUCS-91-40

1992-05-01

A Protocol for Dynamic Assessment of Network Topology in DQDB MANs

Authors: Lakshmana N. Kumar and Andreas D. Bovopoulos

The 802.6 protocol of DQDB MANs aims to maintain a distributed queue for network access and yet inherits access unfairness. In this paper, the DANT protocol is proposed to provide head-end nodes as well as each active node in a DQDB network with real time information about the active node population, the intermodal distance, the position of each node along what bus and the length of a node's downstream bus segment. DANT's current implementation introduces an overhead of 5 bits per slot but alternative implementations which retain the current slot structure of 802.6 and implement the DANT protocol through the use of periodically issued control slots are possible. Results presented in {3,4,5} show that DANT holds much promise for its use in the context of new load balancing and access protection schemes for DQDB MANs.
... **Read complete abstract on page 2.**

Follow this and additional works at: http://openscholarship.wustl.edu/cse_research

Recommended Citation
Kumar, Lakshmana N. and Bovopoulos, Andreas D., "A Protocol for Dynamic Assessment of Network Topology in DQDB MANs" Report Number: WUCS-91-40 (1992). *All Computer Science and Engineering Research.*
http://openscholarship.wustl.edu/cse_research/658

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

A Protocol for Dynamic Assessment of Network Topology in DQDB MANs

Complete Abstract:

The 802.6 protocol of DQDB MANs aims to maintain a distributed queue for network access and yet inherits access unfairness. In this paper, the DANT protocol is proposed to provide head-end nodes as well as each active node in a DQDB network with real time information about the active node population, the intermodal distance, the position of each node along what bus and the length of a node's downstream bus segment. DANT's current implementation introduces an overhead of 5 bits per slot but alternative implementations which retain the current slot structure of 802.6 and implement the DANT protocol through the use of periodically issued control slots are possible. Results presented in {3,4,5} show that DANT holds much promise for its use in the context of new load balancing and access protection schemes for DQDB MANs.

This technical report is available at Washington University Open Scholarship: http://openscholarship.wustl.edu/cse_research/658

A Protocol for Dynamic Assessment of Network Topology in DQDB MANs

Lakshmana N. Kumar and Andreas D. Bovopoulos

WUCS-91-40

May 1992

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

This work was supported in part by an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT and SynOptics.

Proceedings of the First International Conference on Computer Communications and Networks, June 8-10, 1992, San Diego, California, USA

A PROTOCOL FOR DYNAMIC ASSESSMENT OF NETWORK TOPOLOGY IN DQDB MANs

Lakshmana N. Kumar[1] *and Andreas D. Bovopoulos*[2]
Department of Computer Science and
Computer and Communications Research Center
Washington University, St. Louis, MO 63130-4899

ABSTRACT

The 802.6 protocol of DQDB MANs aims to maintain a distributed queue for network access and yet inherits access unfairness. In this paper the DANT protocol is proposed to provide head-end nodes as well as each active node in a DQDB network with real time information about the active node population, the internodal distance, the position of each node along the bus and the length of a node's downstream bus segment. DANT's current implementation introduces an overhead of 5 bits per slot, but alternative implementations which retain the current slot structure of 802.6 and implement the DANT protocol through the use of periodically issued control slots are possible. Results presented in [3, 4, 5] show that DANT holds much promise for its use in the context of new load balancing and access protection schemes for DQDB MANs.

1 Introduction

The Distributed Queue Dual Bus (DQDB) architecture has two unidirectional buses. A node transmits its packets along a bus if the packets are bound for any other node in the downstream. With a symmetric load the traffic sent from a given node is assumed to be destined for any other node in the network with equal probability. With a symmetric load, a node's traffic is proportional to its downstream node population, the domain of potential destinations.

The 802.6 protocol restricts a node to just one outstanding request for a slot, and a next request can be issued only after its previous request has been honored. This restriction together with the latency in transporting a request contribute to unfairness. At symmetric heavy loads, the front-end nodes of a bus receive a heavy influx of requests. Honoring a request invites a fresh request from the downstream. As a result, for every access, the front-end nodes of a bus suffer longer delays. Depending on the exact traffic distribution, other heavy load unfairness patterns are possible as well [3].

Accurate knowledge of the population in the network and its position along the bus enable a node to claim at least its optimum share of the bandwidth. This paper proposes a protocol for Dynamic Assessment of Network Topology (DANT) for effective updates of such information — with an additional overhead of 5 bits in the DQDB slot format. The operation of the DANT protocol is presented in detail. The performance of 802.6 protocol can be tuned to different levels by changing a node's perception of the node population in the network. In [5, 3] the DANT protocol is used for the elimination of the access unfairness (in terms of the node throughput and access delay) at heavy symmetric loads.

2 The Proposed DANT Protocol

Need For A Dedicated Scheme : In a given interval, some nodes may go down or several new nodes may join the network. Hence with *static preset procedures,* a node may apply inappropriate limits, and the bandwidth allocation at high loads may not be fair to every node. Every node should adapt to such network parameter changes immediately and redefine access limits. Else the extent of improvement may be very much time-dependent.

Head-ends may employ *management control functions* to monitor the network population. However they may prove expensive for several reasons. Special control slots must be employed by the head-ends to gather and communicate population information. Every node must be informed individually (through dedicated slots) of its position along the bus and this leads to considerable overhead. The validity of the information communicated depends on how frequently these assessments are made. The more frequently, the greater the overhead. Gathering information such as the length of the bus or the segments of a bus in between nodes requires complex functions, and the overhead may be enormous. Thus a dedicated protocol is necessary.

The Dynamic Assessment of Network Topology (DANT) proposed here carries out dynamic updates frequently to provide to each node accurate information which can be used to control unfairness and access control.

Objectives of DANT : The word *topology* refers to the layout or configuration of the network, including the distances between successive nodes and between a node and the head-ends of a bus. This information can be measured in terms of slots. The goals of the proposed scheme are as follows : (1) to enable every node to know its position along a particular bus correctly, (2) to enable the individ-

[1]Currently with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33124, USA.

[2]This work was supported in part by an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT, SynOptics.

ual nodes to estimate their upstream and downstream bus lengths in terms of integer slots, so that they can collect history information[3] about incoming requests and use it to estimate future traffic, (3) to update this information dynamically and to make it available at every node, (4) to detect errors and restart a new assessment cycle, if some nodes go down in the middle of a cycle, (5) to ensure that new nodes joining the network do not interfere with the ongoing assessment cycle (new nodes should join the next assessment cycle), and (6) to enable every node to compute the distance (in terms of integer slots) between every *active adjacent pair* of nodes along a particular bus. Such information can be used by new access mechanisms to better the 802.6 performance.

Structure of the DANT Protocol : The structure of the proposed DANT scheme is summarized briefly. The following discussion relates to an individual assessment cycle.

- Each head-end runs an assessment cycle on its own bus. The nodes use the information collected by the cycle on a particular bus to access that bus.
- At the end of each cycle, the network-dependent parameters (node population, the position of a node and its downstream bus-length) are updated by each node. The updates are done regularly. The scheme is thus *dynamic* because it responds to changing network conditions at the end of every assessment cycle.
- In the proposed DANT scheme, the assessment cycles are continuous *i.e.* (immediately after the end of a cycle, the next one begins) as long as the network is functioning.
- The scheme operates on the contents of the newly defined *five additional bits*. The scheme progresses in *four different phases* defined as part of each assessment cycle. The initiation, execution and termination of each phase depend on the contents of these *five* bits.
- In case of inconsistency (or on the detection of an error) the current cycle can be aborted and a new assessment cycle can begin.
- During the *abort operation* every regular node will reset all the counters associated with the assessment cycle and estimates made so far. Previous estimates that are available will be used until a fresh update is effected.
- Any inconsistencies are short lived and last only through the length of the next assessment cycle.
- All four phases come in succession, and a final update is done at the end of the cycle. The information collected represents the topology of the network as it existed at the beginning of the cycle. New nodes which may have joined the network when the cycle was in progress shall not interfere with an ongoing cycle.

[3] [5] presents a 3-Tier protocol to control latency related unfairness at normal loads and retain 802.6 performance at low loads. At heavy loads, access protection scheme is employed with access protection limits defined to suit symmetric traffic. It makes use of the information that can be gathered dynamically through DANT protocol.

2.1 Definition of Additional Bits

In order to put the proposed DANT protocol to work, a slight modification in the slot format specified in the 802.6 standards is necessary. Five more bits[4] are needed to enable the operation of the assessment protocol. With the inclusion of these bits the slot format takes the following form :

Phase Header					
A	B	C	D	E	802.6 Format
1 bit	1 bit	1 bit	1 bit	1 bit	53 bytes

The five additional bits are named A, B, C, D and E.

Phase Header Bits: Bits A, B, and C are collectively termed as *phase headers*. These bits together indicate the ongoing phase of an assessment cycle. Regular nodes identify the phase by examining these bits. Head-ends issue these phase header bits, which are occasionally modified by the regular nodes in accordance with the operational scheme or algorithm.

Node Response Bit: The D-bit is used by the regular nodes in the network for writing their responses (either by setting or resetting it) as part of the operational schema.

Independent assessment cycles run on each bus. HOB_A[5] runs its own cycle on bus-A, and HOB_B runs another cycle on bus-B. HOB_A and HOB_B maintain their own cycles separately. A regular node uses the C-bit of the opposite bus slot to respond to an ongoing cycle on a particular bus. Any node that must write its response to the ongoing cycle on bus-A shall use the C-bit of slots arriving on bus-B and vice versa.

Head-end Response Bit: The E-bit is used by a head-end to write its response to the assessment cycle controlled by its peer. As before, the response goes in the opposite bus. Suppose HOB_B needs to respond to a cycle controlled by HOB_A on bus-A. Then HOB_B writes its response in the E-bit of its own slot and puts the slot on bus-B. This is done by HOB_B issuing a slot on bus-B with $E = 1$. Otherwise, all HOB_B slots have $E = 0$.

The *phase header* values are related to the different phases of the protocol and are presented in Table-1. There are eight different values for the phase header. Every value carries a specific meaning associated with the assessment cycle. The *phase header* information passing on a particular bus is related to the ongoing cycle of the same bus.

[4] The data unit for the Physical Layer Service is an octet. This may imply that an additional overhead of 1 byte (or one octet) is needed to support the proposed protocol. The extra bits may be used to support additional functions for future enhancements.

[5] HOB_A refers to the frame generating head-end of bus-A, and HOB_B refers to the frame generating head-end of bus-B.

Table 1: Meaning of Phase Header Bits

Phase Headers (A, B, C bits)	Meaning associated within an assessment cycle
1 1 1 xx	Beginning of a new assessment cycle.
0 0 0 xx	Abort current cycle. (Issued only by a head-end.)
0 0 1 xx	End-the-phase slot : End the current phase of an ongoing cycle and begin the next phase.
1 1 0 xx	Phase-I Slot
0 1 1 xx	Phase-II Slot
0 1 0 xx	Phase-III Slot
1 0 x xx	Phase-IV Slot

The 'x' value for a bit indicates that it can be either a '1' or '0'. This implies that the actual value does not relate to the context of discussion and hence is inconsequential. For example, a regular node might alter the response bit (D-bit) in a bus-A slot (as a response to the ongoing cycle in bus-B) and not affect the discussion of the cycle on bus-A.

2.2 Operation of the DANT Protocol

The following discussion is restricted to the assessment cycle run and controlled by HOB_A on bus-A. The following facts may be noted about the assessment cycle :

- The cycle is run on bus-A, and responses to this cycle are written or observed on bus-B slots. Thus only phase headers of bus-A slots and the D–E bits of bus-B slots are important for this cycle.
- The responses to this cycle by regular nodes are written on the D-bit of bus-B slots by a regular node.
- HOB_B responds to the ongoing cycle of bus-A by issuing a slot on bus-B with its E-bit set.
- An 'x' shall indicate that the corresponding bit can either be a '1' or a '0' and is irrelevant for that part of discussion.
- Responses to the cycle on bus-B are written by regular nodes on the D-bit of bus-A slots. This does not interfere with the operation of the cycle on bus-A.
- HOB_A may still issue a slot with its E-bit set, if necessary, as a response to the cycle maintained by HOB_B. This does not interfere with the functioning of the bus-A cycle.
- The head-end and regular nodes maintain a few counters as well. Specific information regarding the need and usage of the counters is presented during the discussion on individual phases.
- The operation can be extended to the assessment cycle maintained by HOB_B on bus-B on similar lines.

2.2.1 Initiating an Assessment Cycle

HOB_A issues a 1 1 1 0x slot on bus-A. Only one such slot is issued. Every regular node then prepares for a new assessment cycle which proceeds in a phased manner. In short, the 1 1 1 0x slot serves as a declaration of an impending new cycle to every node in the network.

A regular node that sees a 1 1 1 1x slot understands that a new cycle is impending. It is possible that even though HOB_A issues a 1 1 1 0x slot, some node may still modify the D-bit of this particular slot in response to the cycle in bus-B.

Phase-I of the assessment cycle begins immediately after HOB_A issues its 1 1 1 0x slot. Once HOB_A starts issuing 1 1 0 0x slots, it continues to do so throughout this phase.

2.2.2 Operation of Phase-I

The goals of the first phase of the assessment cycle are :

- To enable an estimate by HOB_A of the bus-length in terms of integer slots.
- To enable an estimate by every regular node of the downstream bus-length[6] in terms of integer slots.

Role of Head-ends : Immediately after its 1 1 1 0x slot, HOB_A starts issuing 1 1 0 0x slots and continues to do so throughout this phase.

HOB_A also looks for incoming bus-B slots with the E-bit set to 1, indicating that HOB_B acknowledges the receipt of new cycle declaration on bus-A. Also HOB_A counts the phase-I slots it issues until it receives a response from HOB_B.

When the 1 1 1 xx slot is seen by HOB_B, it should respond by setting the E-bit of the next immediate slot on bus-B to 1.

When the acknowledgment from HOB_B is received by HOB_A, phase-I ends, and HOB_A immediately stops counting the phase-I slots issued thus far. Half the counter value is an estimate of the bus length in terms of integer slots.

$$\text{Length of bus-A} = \left\lceil \frac{\text{phase-I slot counter value}}{2} \right\rceil$$

After HOB_B's response is received in phase-I HOB_A issues a 0 0 1 0x slot that declares to every recipient

[6] For any node the portion of bus-A extending from the node itself to HOB_B corresponds to the downstream bus-length on bus-A. This length is expressed in terms of integer slots.

node the end of phase-I.

Role of Regular Nodes : The receipt of a 1 1 1 xx slot indicates to a regular node the beginning of Phase-I on bus-A. Immediately after receipt, the regular node starts counting the Phase-I slots sent on bus-A.

Each regular node also keeps a watch on bus-B. When it sees a slot on bus-B with the E-bit set to 1, it stops the counter. Half the value of the counter provides the node with the length of bus-A, downstream from the node.

$$\text{Bus-A downstream length} = \left\lceil \frac{\text{phase-I counter value}}{2} \right\rceil$$

Thus each regular node is able to estimate its downstream bus-length (in terms of integer slots) by itself.

When the *end-the-phase slot* 0 0 1 xx arrives on bus-A, every regular node updates its knowledge of the downstream bus-length by saving the estimate made in this phase.

Also it is possible for every regular node to estimate the length of bus-A, upstream from the node by counting the phase-I slots arriving on bus-A during the interval between the time a slot with E-bit = 1 is seen on bus-B to the time the end-the-phase slot on bus-A is seen. Half the counter value (upper ceiling applied) gives the length of bus-A upstream from the node.

2.2.3 Operation of Phase-II

During this phase the head-end estimates the *active population* in the network. The word *active* refers to the nodes that are participating in the network. Unlike the bus-length measurements of phase-I, the active population may change often as nodes go down or become active and join the network. During this phase the regular nodes do nothing for themselves except answering the call of attendance issued by HOB_A on bus-A under the rules of the DANT protocol as discussed below.

Commencement of Phase-II : Immediately after issuing an end-the-phase slot to conclude phase-I, HOB_A begins the next phase by issuing a 0 1 1 0x slot. It continues to issue 0 1 1 0x slots throughout phase-II.

Role of Regular Nodes : On receipt of the very first phase-II slot 0 1 1 xx, every regular node must write a response on the D-bit of the next available bus-B slot.

Contention for writing such a response is possible because of the concentration of nodes. Especially when the bus-length is smaller than the active population of nodes, (e.g., 100 nodes active along a bus of length 30 slots), special attention is required. In this example, at the end of 30 slots, HOB_B sends an $E = 1$ response to HOB_A authorizing it to end its phase-II. This response slot reaches HOB_A 30 slots later along bus-B. By this time only about 60 nodes have sent their responses. HOB_A can have no knowledge of impending responses.

To overcome this difficulty, a restriction is imposed. A regular node will not allow a slot on bus-B whose E-bit is set if it has not answered the call of attendance in phase-II. Instead it will reset[7] the E-bit to 0 and wait its turn to write its response. At the time of its own D-bit response, it will also set the E-bit in the same slot. In this way every regular node should be able write its response.

Role of Head-ends : After initiating phase-II HOB_A keeps a watch on the slots arriving on bus-B, looking for responses on either D-bit or E-bit. D-bit responses are the answers from the regular nodes to the call of attendance. E-bit response informs HOB_A to conclude phase-II.

During this phase HOB_A counts the D-bits responses received on bus-B and terminates the count when it sees a bus-B slot with E-bit set. If the bus-B slot with the E-bit set also has its D-bit set, then it too is counted as a response.

The counter value indicates the *active* population in the network. The E-bit response should be written by HOB_B in the next slot on bus-B, immediately following the receipt of the first slot of phase-II 0 1 1 xx on bus-A.

Immediately after evaluating the active population, HOB_A concludes phase-II by issuing an *end-the-phase* slot as 0 0 1 0x on bus-B.

Error and Abort : It is possible that a regular node may go down after resetting the E-bit but before writing it back. This can be detected by the protocol as follows. HOB_A has the estimate of bus-length from phase-I, say B_s. If the E-bit response is not seen by HOB_A on bus-B in $2(B_s+1)$ slots, then either there are more active nodes than B_s or a regular node may have gone down after resetting the E-bit to 0 but before setting back to 1. HOB_A should check every incoming bus-B slot beyond the $2(B_s+1)^{th}$ slot. HOB_A should observe a sequence of slots having the D-bit set and the E-bit 0, followed by a slot having both the D-bit and E-bit set to 1. If this rule is violated then HOB_A will abort the cycle by issuing an *abort* slot as 0 0 0 0x on bus-A.

Thus during phase-II, the HOB_A is able to assess the active network population. During phase-II, regular nodes make no assessments of their own.

2.2.4 Operation of Phase-III

The goals of phase-III are to enable the head-ends (in this discussion, HOB_A) to map the topology information of the network and allow the regular nodes to estimate the number of active nodes in the upstream of the bus on which this phase runs (in this discussion, the number of upstream nodes with respect to bus-A).

Topology refers to the inter-nodal distances between successive nodes along the bus in terms of integer slots, taking upper ceiling wherever necessary. Because of the use of the

[7] A node writes into a slot via a logical OR. Resetting of the E-bit poses a problem and needs specific attention. The simplest solution may require an extra bit which can be set by a node only when it has to reset the E-bit. This would thus indicate a resetting of the E-bit.

upper ceiling criteria, the sum of individual inter-nodal distances (in integer slots) need not sum up to the bus-length (in integer slots).

The knowledge of the active population in the network acquired in phase-II is made use of by the head-end during this phase. In order to store the inter-nodal distance information, the head-end maintains an array of $(N+1)$ elements for a network of N active nodes. It also employs a counter to count the phase-III slots issued.

Role of Head-ends : HOB_A makes use of the information acquired in previous two phases, namely the active node population N and the bus-length B_s.

Throughout phase-III HOB_A issues [0 1 0 0x] slots on bus-A and keeps a count of the number of such slots issued. It ends phase-III by issuing an end-the-phase slot [0 0 1 0x], after getting a bus-B slot with its E-bit set to 1 by HOB_B.

Every regular node that has answered the call of attendance will send a response through the D-bit of bus-B slots. When N such responses and the E-bit response from bus-B are received, HOB_A concludes the phase by issuing an *end-the-phase* [0 0 1 0x] slot on bus-A. The evaluation of the topology and other details are presented as a separate item below.

HOB_B sends its E-bit response only after receiving the first phase-III slot on bus-A as [0 1 0 xx], and the response is sent in the very next slot on bus-B.

Unlike phase-II wherein the regular node responses may arrive in succession with every bus-B slot, the rules specified for the regular nodes in phase-III force responses to be sent in a way that has a direct relationship to the inter-nodal distances in slots. As a result, the manner in which responses arrive at HOB_A helps map the topology of the network.

Topology Mapping by Head-end : Based on the way HOB_A receives the D-bit responses on bus-B, HOB_A can evaluate the inter-nodal distances (or map the topology) between successive nodes along bus-A as follows :

- As soon as phase-III starts, HOB_A starts a counter, say *S_CTR*, which is incremented with every incoming bus-B slot since the commencement of phase-III.
- HOB_A initializes a response counter, say *R_CTR*, which is incremented with every incoming bus-B slot with its D-bit response set.
- Since the active node population is estimated to be N, HOB_A must open an array of $(N+1)$ elements to store the inter-node distances. Including the two head-ends, there are $(N+2)$ nodes in the network and $(N+1)$ inter-nodal distances measured in terms of slot lengths.
- Let the array name be *lap-distance*. Then the individual elements have the following meaning.

lap-distance [1]	⇒	From HOB_A to node-1
lap-distance [2]	⇒	From node-1 to node-2
⋮	⋮	⋮
lap-distance [N]	⇒	From node-$N-1$ to node-N
lap-distance [N+1]	⇒	From node-N to HOB_B

- Another counter called *L_CTR* (lap counter) is reset and started. This counter helps keep track of slots in between successive D-bit responses received on bus-B. The counter is incremented with every incoming bus-B slot. When an incoming bus-B slot has its D-bit set, the counter is incremented, its contents are stored in the *lap-distance* array, and the counter is reset to zero.
- To summarize the evaluation done by the DANT protocol, HOB_A does the following.

1. At the beginning of phase-III, HOB_A initializes all the three counters, namely *S_CTR, R_CTR, L_CTR,* to zero.

2. When HOB_A receives a slot on bus-B with the D-bit not set, it does the following:

   S_CTR ++;
   if (*S_CTR* $\leq (2(B_s+1)+N)$) {
   L_CTR ++;
   } else {
   ABORT the current cycle.
   }

3. When HOB_A receives the first bus-B slot with its D-bit set, it does the following:

   S_CTR ++;
   R_CTR ++;
   L_CTR ++;
   lap-distance [*R_CTR*] $= \left\lceil \frac{L\_CTR}{2} \right\rceil$;
   L_CTR = 0;

4. For every subsequent bus-B slot received by HOB_A with the D-bit set and the E-bit not set HOB_A does the following:

   S_CTR ++;
   R_CTR ++;
   L_CTR ++;
   lap-distance [*R_CTR*] $= \lceil L\_CTR \rceil$;
   L_CTR = 0;

   It is possible that both the D-bit and the E-bit of a bus-B slot may be set. This can happen only when the distance between the last node along bus-A and HOB_B is less than one slot. This slot is handled according to the actions specified for a slot with the E-bit set.

5. Phase-III comes to an end when HOB_A receives a bus-B slot with its E-bit set. The following actions occur at HOB_A.

```
S_CTR ++;
R_CTR ++
L_CTR ++;
lap-distance [ R_CTR ] = [L_CTR];
if (D-bit == 1) {
        R_CTR ++
        lap-distance [ R_CTR ] = 1;
}
L_CTR = 0;
if (R_CTR == N) {
        issue end-the-phase slot
} else {
        ABORT the current cycle.
}
```

Role of Regular Nodes : Every regular node that has answered the call of attendance will send a response through the D-bit of bus-B slots.

After seeing the *end-the-phase* slot [0 0 1 0x] during phase-II, every regular node starts phase-III and waits for the first [0 1 0 xx] slot on bus-A.

After the commencement of phase-III, it is possible that a node may continue to receive [0 1 1 xx], for reasons to be explained below. Every regular node initializes a counter as soon as phase-III commences and counts the number of such [0 1 1 xx] slots received before the arrival of the first phase-III slot.

On seeing the first phase-III slot, every regular node sets the C-bit of that slot before allowing it to pass on bus-A. Each node also writes a response in the D-bit of the *immediate next* slot on bus-B. Second and subsequent phase-III slots require no action on the part of regular nodes.

Because the C-bit of phase-III slot is modified by successive nodes along bus-A, different slots are seen as the very first [0 1 0 xx] slot of phase-III by different nodes along bus-A. Hence regular nodes encounter no contention when writing their D-bit responses in next slot on bus-B.

The setting of the C-bit makes some phase-III slots on bus-A appear like phase-II slots (with phase headers transformed into [0 1 1 xx]). This is not problematic because the phases are ordered and the end of phase-II has already been declared as over (by the specific *end-the-phase* slot).

Only in phase-III may a regular node modify a phase header bit (C-bit), and this is permitted only once. Every regular node keeps a count of the number of [0 1 1 xx] slots received before the first phase-III slot. At the end of the phase *this counter value yields the number of nodes in the upstream of bus-A.*

After the commencement of phase-III, the upstream nodes modify and send [0 1 1 xx] slots. However once the first phase-III slot [0 1 0 xx] is seen, then a regular node will see only [0 1 0 xx] slots till the end of phase-III.

The participation of regular nodes in this phase is confined to writing their responses.

Error and Abort : It is possible that a regular node that answered the call of attendance might have gone down and hence no longer be active. This eventuality can be traced or detected as follows:

Even in the worst possible scenario of the distribution along the bus, all regular nodes should be able to write D-bit responses within $(2(B_s + 1) + N)$ slots on bus-B from the time phase-III started. Thus if HOB_A fails to read N (the active node population estimated in phase-II) D-bit responses within the said number of bus-B slots since the commencement of phase-III, then possibly a node may have gone down. In such a situation HOB_A *aborts* the cycle by issuing a [0 0 0 0x] slot on bus-A.

2.2.5 Operation of Phase-IV

During phase-III, HOB_A has collected information on network topology. Phase-IV is aimed at transmitting this topology information to the regular nodes. The regular nodes collect the information from bus-A.

Role of Head-ends : Throughout phase-IV, HOB_A sends two types of slots, [1 0 1 0x] and [1 0 0 0x] on the bus-A.

Every [1 0 0 0x] slot from HOB_A corresponds to one unit of inter-nodal distance, and every [1 0 1 0x] slot from HOB_A indicates the presence of an active node. Together these slots are sent in a fashion that represents the topology information of the network, collected by HOB_A during previous phases. For example, suppose *lap-distance* [1] = 3 and *lap-distance* [2] = 5. During phase-IV, HOB_A will send the following sequence of slots to carry the information.

First 3 slots as	[1 0 0 0x]	⇒	HOB_A to node-1 slot distance
Next slot as	[1 0 1 0x]	⇒	Node-1 present
Next 5 slots as	[1 0 0 0x]	⇒	Node-1 to node-2 slot distance
Next slot as	[1 0 1 0x]	⇒	Node-2 present
⋮	⋮	⋮	⋮
One slot as	[1 0 1 0x]	⇒	HOB_B present
Last slot as	[0 0 1 0x]	⇒	*end-the-phase* slot

Exactly $(N+1)$ slots are issued as [1 0 1 0x] from HOB_A, one slot for each active node population in the network and an additional slot for the other head-end.

Immediately after issuing the [1 0 1 0x] slot corresponding to the presence of HOB_B, the next slot is issued as the *end-the-phase* slot by HOB_A. The last [1 0 1 0x] implies the end of the phase-IV and the current assessment cycle.

Unlike previous phases, HOB_A does not collect any information or depend on any response bits along bus-B to run or terminate this final phase. HOB_A runs this phase on its own.

Role of Regular Nodes : Each regular node initializes a counter at the beginning of phase-IV and then counts the number of [1 0 1 xx] slots seen in phase-IV. Let this counter be denoted by *N_CTR*. At the end of phase-IV this *N_CTR* value decremented by one will represent the active node population in the network.

Similar to the evaluation of the topology explained in phase-III, every regular node can read the topology information from the incoming slots. Every regular node initializes a new counter, say *LA_CTR* at the beginning of phase-IV. Throughout phase-IV, *LA_CTR* counts the [1 0 0 xx] slots (arriving on bus-A) until a [1 0 1 xx] slot is seen. At this point, the *LA_CTR* value is moved to *lap_distance* [*N_CTR*]. Also at this point *LA_CTR* is again reset to zero and counting begins again. When *end-the-phase* slot is seen, every node can evaluate its position along bus-B from available information as follows. The number of active upstream nodes along bus-A is known from phase-III. Adding one to this value gives the actual position of a regular node along the bus. Also, the active node population in the network is known from phase-IV. The *N_CTR* used in phase-IV contains this value, as explained previously.

Error and Abort : In this phase the failure of a node can not be traced, since the head-end receives no information or response from the regular nodes. However this is the last phase, and the next assessment cycle should be able to remove any discrepancy.

2.3 Length of an Assessment Cycle and Slot Format

For each assessment cycle, five slots are needed to declare the commencement of a new cycle and declare the end of each of the four phases. The length of each phase can also be approximately estimated. Consider a network of N nodes with bus-length B_s slots. It is not very difficult to see that the length of a DANT cycle runs up to $(8B_s + 3N + 5)$ slots. As an example, consider a network operating at 44.7 Mbps with 100 nodes located on a bus of length 50 Kms. This corresponds to a slot interval of 8.59 μsecs and a bus-length of about 30 slots. Hence the DANT cycle can have a maximum length of 545 slots. This translates to about 4.68 msecs. Thus, for this example, a dynamic update is possible every 4.68 msecs. It may be possible to trade-off overhead for the cycle length by retaining the DANT algorithm but using control slots, instead of introducing the bits in the slot format of 802.6.

2.4 Safeguards for the DANT Protocol

A regular node that newly joins a network should not participate half way through the ongoing assessment cycle on a particular bus. Instead it should wait for the new cycle declaration [1 1 1 xx] slot to be seen on that bus.

When a node goes down in the middle of an assessment cycle, then the cycle may be aborted and a new cycle started. The failure of the node that caused the abort will also be detected in the next assessment cycle.

If a regular node finds the phases of an ongoing cycle out of sequence (this may happen if there is a bit error due to the noise in the channel), then the node can *ignore* the ongoing cycle and wait for the commencement of the next assessment cycle. Update in this case will be delayed until the end of the next cycle. Only head-ends can issue the abort slot. Regular nodes can not abort on their own.

3 Conclusion

In this paper the DANT protocol was proposed to provide head-end nodes as well as each active node in a DQDB network with real time information about the active node population, the internodal distance, the position of each node along the bus and the length of a node's downstream bus segment. DANT's current implementation introduces an overhead of 5 bits per slot, but alternative implementations which retain the current slot structure of 802.6 and implement the DANT protocol through the use of periodically issued control slots is possible.

Results presented in [3, 4, 5] show that DANT holds much promise for its use in the context of new load balancing and access protection schemes for DQDB MANs. For example, [5] proposes a 3-Tier protocol that retains the basic 802.6 performance at very low loads. At normal loads the 3-Tier protocol introduces a demand prediction scheme [4, 5] through the utilization of the length (in terms of slots) of the downstream bus segment. Finally under heavy loads the 3-Tier protocol, controls the access unfairness and demonstrates that uniform delay and throughput characteristics can be achieved through the utilization of the node population and the position of each node along the bus [3, 5].

References

[1] Editor, The IEEE 802.6 Working Group, "Project 802: Local an Metropolitan Area Networks – Draft copy of Proposed Standards," Institute of Electrical and Electronics Engineers, Inc, 1989.

[2] J. Filipiak, "Access Protection for Fairness in a Distributed Queue Dual Bus Metropolitan Area Network," Proceedings of the International Conference on Communications, Boston, June 11-14, 1989, pp. 635-639.

[3] L. N. Kumar and A. D. Bovopoulos, "An Access Protection Solution for Heavy Load Unfairness in DQDB" Proceedings of IEEE INFOCOM'92 Conference, Florence, Italy, May 1992.

[4] L. N. Kumar, and A. D. Bovopoulos, "A Proposed Solution For Latency Related Unfairness in DQDB," Technical Report WUCS-91-41, Department of Computer Science, Washington University, St.Louis, 1991.

[5] L. N. Kumar, "A 3-Tier Fairness Protocol: A Proposal for the Solution of the Unfairness Problem in DQDB MANs", M.S. Thesis, Washington University, St.Louis, MO, USA, Aug 1991.