A HYBRID RING/BUS APPROACH FOR FAST CONTENTION-FREE ACCESS

SANJAY K. BOSE

Department of Electrical Engineering, I.I.T., Kanpur – 208016, INDIA

This paper proposes a new "Ring/Bus" protocol to provide fast, contention-free access in ring-type LANs. Proposed as an improvement to the standard Token Ring network, this substantially reduces the ring latency time of the network by replacing the traditional one (or more) bit delay at each node for Token processing by a switching arrangement. The proposed protocol will always provide better service than a comparable Token Ring system. The improvement observed is significantly high for systems supporting a large number of nodes. Modifications to the basic ring/bus protocol are suggested to provide service to different priority classes. A simplified analytical model of this system has also been examined to obtain approximate results.

I. INTRODUCTION

The Token Ring protocol provides an efficient access mechanism for supporting a large number of user nodes in a Local Area Network (LAN). Its major attraction is that the access is contention-free and that, subject to certain constraints, an upper bound for the access delay may be specified. Contention-based protocols like CSMA/CD (i.e. in the IEEE 802.3 standard [1]) do not exhibit this bounded-delay property and also suffer from problems of potential instability. The IEEE 802.5 standard [2] incorporates the Token Ring access mechanism for use in ring-type LANs.

The basic approach of the Token Ring protocol is to circulate a Token around a unidirectional ring to which the various user nodes are connected. A node which wants to transmit, waits for the Token to arrive, holds the Token, transmits its packet(s) and then forwards the Token to the next downstream node. This effectively behaves as a set of queues served in cyclic order by a single server where the server requires a non-zero walk-time to move from one queue to the next after providing service at the former [3].

Consider a simple Token Ring network as shown in Fig. 1 [4]. In this, the bit pattern 01111110 acts as the Token (or Idle Token, IT) and the bit pattern 01111111 is used as the Connector (or Busy Token, BT). The nodes are assumed to transmit in bit synchronous fashion and that zero-bit insertion is appropriately used to prevent the appearance of IT or BT within the packet. The nodes on the ring (other than the one currently transmitting its packet) typically repeat on their RxD lines whatever appears on their TxD lines with a delay of one bit; they would simultaneously read any packet addressed to themselves.

After it transmits its packet(s), the node holding the IT sends it to the next downstream node. If this node does not have anything to send, it passes the IT downstream through its 1-bit delay. Otherwise, it converts the IT to BT by inverting the last 0 bit of the IT to 1, sends its packet(s) and then forwards the IT to the next node. While it is transmitting, it discards the bits received on its RxD line. After forwarding the IT, it must send "idle fill" until it receives the end of its packet transmission; it then reverts to the 1-bit delay mode. Note that the nodes of this simple Token Ring must incorporate this 1-bit delay in order to change the received IT to BT when access is needed.

For this ring network, the "walk-time" between nodes i and i+1 would be the sum of one bit time and the propagation delay between these nodes. Its "Ring Latency Time" (RLT) is the total time for a signal to go around the ring and will be the sum of all the individual walk-times; This should be large enough to contain at least one IT (i.e. 8 bits). Note that apart from a fixed propagation delay component, the RLT of the network will also contain a component which is linearly dependent on the number network of nodes. Our objective in proposing the Ring/Bus protocol is to get a ring network whose RLT consists only of the propagation delay around the ring depending on the actual media length but independent of the number of nodes in the network.
II. THE RING/BUS PROTOCOL FOR NON-PRIORITY SERVICE

This scheme [6] is identical to the basic Token Ring except that the Network Interface Unit (NIU) has two switches S1 and S2 replacing the 1-bit delay of the Token Ring protocol. The RLT reduction in the non-priority case is achieved by replacing the RLT by a 1-bit delay through switches S1 and S2. The system uses a 9-bit Token/Connector pattern where Token (IT)=01111110, Connector1 (BT1)=0111111110 and Connector2 (BT2)=0111111111. Note that the pattern 011111101 is not used and will occur only if there are bit errors. Moreover, the zero-bit insertion strategy is followed in the same way as for the non-priority Token/Bus protocol of Sec. II to ensure that the Token/Connector patterns do not accidentally appear within a packet. The physical layout of the system would still be as shown in Fig. 2 but the Node Interface Unit (NIU) for priority service will be more complex than the two-switch arrangement proposed for the non-priority case. Fig. 3 shows the NIU that should be used at each node of a Ring/Bus network supporting two priority classes.

A node with a packet to transmit monitors the Ring/Bus for the pattern 0111111 with S1 closed and S2 open. On seeing this, it opens S1, closes S2 and transmits a 1 regardless of the bit being currently received. If the bit received was a 0, then the node has indeed acquired the right to access the ring (since what it has effectively forwarded is a BT) -- it then operates just like node M in the earlier description. If the bit received was a 1, it fails to acquire access rights and returns to the monitor-only mode (with S1 closed, S2 open) looking once again for the pattern 0111111.

This basic Ring/Bus protocol has been analyzed in [6]. In general it was found that the RLT reduction provided by the Ring/Bus protocol (since it no longer forces 1-bit delays at every node) makes it always more efficient than the equivalent Token Ring system. The average packet delay in the Ring/Bus network is lower and it provides higher capacities than the Token Ring.

III. MULTI-PRIORITY RING/BUS

We present here a modification to the basic Ring/Bus protocol which supports traffic of two priority classes -- priority 2 for high priority (HP) and priority 1 for low priority (LP). (This approach can be easily extended to support more priority classes.) The system uses a 9-bit Token/Connector pattern where Token (IT)=01111110, Connector1 (BT1)=0111111110 and Connector2 (BT2)=0111111111. Note that the pattern 011111101 is not used and will occur only if there are bit errors. Moreover, the zero-bit insertion strategy is followed in the same way as for the non-priority Token/Bus protocol of Sec. II to ensure that the Token/Connector patterns do not accidentally appear within a packet. The physical layout of the system would still be as shown in Fig. 2 but the Node Interface Unit (NIU) for priority service will be more complex than the two-switch arrangement proposed for the non-priority case. Fig. 3 shows the NIU that should be used at each node of a Ring/Bus network supporting two priority classes.

The normal connections at a NIU when the node is not transmitting is \( A \rightarrow \{1A, 2A, 3A\} \) and \( \{1B \rightarrow B\} \). This is referred to as the Passive Receive (PR) connection and allows the node to passively monitor the RxD line for packets addressed to itself while the A-1A-1B-B connection passes data instantaneously from RxD to TxD. The A-2A connection is not required here but may be convenient for implementation.

The normal connections at a NIU when the node is not transmitting is \( A \rightarrow \{1A, 2A, 3A\} \) and \( \{1B \rightarrow B\} \). This is referred to as the Passive Receive (PR) connection and allows the node to passively monitor the RxD line for packets addressed to itself while the A-1A-1B-B connection passes data instantaneously from RxD to TxD. The A-2A connection is not required here but may be convenient for implementation.

Packets arriving for transmission at a node may be of either priority. The node always tries to transmit a waiting HP packet and attempts transmission of a LP packet only when it does not have any HP packet waiting. We still consider a Non-Exhaustive implementation with only one packet (either LP or HP) being sent per Token.

Consider a node which has a HP packet waiting for transmission in its priority 2 queue. With its NIU in the PR mode, it monitors the RxD line for the sequence 0111111x where x may be either 0 or 1. On seeing this, it changes the NIU's connections to
\{A->{2A,3A}, \{3B->{B}\} \} and sends a 1 on its TxD line. After sending this 1, it checks the bit received while this 1 was being sent for the following cases—

(a) If bit received is 0, then the node has successfully acquired the Token. It retains the NIU's present connection to send a HP packet followed by the IT. It then sends "idle fill" until it sees the end of its own packet on RxD. On seeing this it changes to the PR mode.

(b) If bit is 1, the node did not get the Token because an upstream node is sending a HP packet. It should then immediately revert to the PR mode and try acquiring the Token once again.

If a node has a LP packet waiting for transmission in its priority 1 queue and no waiting HP packets, it will attempt to transmit a LP packet. For this, with the PR connection, it monitors RxD for the bit pattern 0111111. On seeing this, the NIU connections are changed to \{A->{2A,3A}, \{3B->{B}\}\} and a 1 is sent on the TxD line. After sending this, it checks the bit received while this 1 was being sent for the following—

(a) If bit received is 1, the node fails to get the Token. It should immediately revert to the PR mode to try acquiring the Token once again.

(b) If bit received is 0, the node has provisionally got the Token (for transmitting one LP packet). With connections retained as they are, it sends a 0 (ie. to form the last bit of the BT1 from this node) followed by data bits from the packet itself. (Note that this Token acquisition is provisional and will get taken away by a downstream node which gets a HP packet to send by the time it receives BT1 from this node.) The node continues to monitor RxD while it is transmitting bits from its LP packet. It does this to check whether the Connector coming back is BT1 or BT2. (Note that it had effectively sent BT1 preceding its packet.) Two possibilities may arise:

[i] If BT1 comes back, then the node has indeed got the Token. It then finishes transmission of its LP packet followed by the IT. It then sends "idle fill" until it receives the end of its own packet. It then reverts back to the PR mode.

[ii] If BT2 comes back, the LP packet transmission from this node is getting preempted by a HP packet from a downstream node. In this case, the node immediately terminates its LP packet transmission (to try again later) and changes the NIU's connections to \{A->{2A,3A}, \{2B->{B}\}\}. In this case, the 9-bit shift register (SR) will contain BT2 which will get forwarded downstream from the next bit interval.

From [ii] above, it follows that a node which starts a LP packet transmission but gets preempted adds a 9-bit delay to the ring. This delay has to be added so that the Connector BT2 preceding the preempting HP packet may be forwarded properly further downstream. The Passive Receive (PR) connection for such a node will remain as \{A->{2A,3A}, \{2B->{B}\}\} with the extra 9-bit delay until the next time the node tries to acquire the Token for either the previously preempted LP packet or a new HP packet. We will refer to this as the "Delayed-PR" connection of this node. Once this Delayed-PR connection is removed, as described next, the node reverts to the old PR connection which does not have the 9-bit nodal delay. Note that nodes in the Delayed-PR mode increase the RLT of the ring because of the extra 9-bit delay that each one of them introduces.

Consider once again a node whose LP packet transmission started but got preempted. It will have at least one LP packet in its priority 1 queue and will be operating in the Delayed-PR mode. If it does not get a HP packet in the meantime, it will wait until it finds an IT 011111100 in its 9-bit SR. On seeing this, it changes its NIU connections to \{A->{2A,3A}, \{3B->{B}\}\} and sends BT2 01111110 followed by its LP packet assuming that it has provisionally acquired the Token. It then follows the same procedure as in (b) above to check if it has actually got the Token or if its LP packet has once again been preempted. On preemption, it repeats the above procedure. However, if the LP packet succeeds, it sends IT following the packet and then sends "idle fill" until it sees the end of its own LP packet on RxD. It then goes back to the old PR connection, ie. \{A->{1A,2A,3A}, \{1B->{B}\}\}. This removes the extra 9-bit delay that had been introduced earlier. It is also possible that a node with preempted LP packet gets a HP packet in its priority 2 queue before it detects the IT in its 9-bit SR. In this case, it would wait until it finds either IT or BT1 in this SR (ie. check its contents for 011111100). On seeing this it changes the NIU connections to \{A->{1A,2A,3A}, \{1B->{B}\}\}, sends BT2 01111111 followed by its HP packet and the Token IT 011111100. It then sends "idle fill" until it sees the end of its own packet on RxD. On being this, it changes its NIU connections to the old PR mode thereby removing the extra 9-bit delay that had been introduced earlier. Once the extra 9-bit delay is removed, the NIU reverts to the operating mode that had been described earlier.

The RLT of this protocol at any instant will be \(m(n+3)m\) where \(n\) is the number of nodes with the "Delayed-PR" connection at that instant. Even though this is higher than the \(m\beta SLT\) of the Ring/Bus scheme of [5], it will still be low since \(n\) will typically be small. It is expected that typically (especially for large \(m\)) the RLT of this is likely to be substantially lower than the \(m(\beta+\gamma)\) RLT of the Token Ring system leading to the improved performance of the Ring/Bus scheme. The priority service provided by the proposed protocol is also somewhat better than that of the IEEE 802.5 standards.

IV. ANALYSIS OF TWO-PRIORITY RING/BUS FOR NON-EXHAUSTIVE SERVICE

Consider a symmetric system with \(m\) nodes where the propagation delay between two neighbouring
nodes is assumed to be constant and is given by \(\beta\). The average RTL of the network will be 
\[
\text{m} + 9\text{rE}([n]) \text{ where E}([n]) \text{ is the average number of nodes with a Delayed PR connection at their NIUs (because of previous LP packet preemption). Let } \gamma \text{ be the average effective walk-time between neighbouring nodes given by}
\]
\[
\gamma = (m\beta + 9\text{rE}([n]) m) = \beta + 9\text{rE}([n])/m \tag{1}
\]
where E([n]) is estimated later and \(\tau\) is the bit duration. For our analysis, we assume that the walk time between any pair of neighbouring nodes is always \(\gamma\).

Let \(\lambda_1/m\) and \(\lambda_2/m\) be the average arrival rates of packets (from a Poisson process) at the priority 1 and 2 queues respectively at each node. Let \(X_1\) and \(X_2\) be the random variables representing the lengths of packets of priority 1 and 2 respectively (recall that priority 1 packets are the low priority LP packets.). Let \(\rho_1 = \lambda_1\text{E}([X_1])\) and \(\rho_2 = \lambda_2\text{E}([X_2])\) respectively be the total traffics of priority 1 and 2 offered to the system. The performance parameter of interest to us here is the average queueing delay \(W_1\) (\(W_2\)) for a packet of priority 1 (2), i.e. the average time interval between the instant when a packet of priority 1 (2) arrives at queue 1 of a node and the time when that node successfully (without preemption) starts transmission of that packet.

We consider first the queue of priority 2 (HP) packets at a particular node and the arrival of a new HP packet to this queue. The mean queueing delay for this packet has three components -

(a) The mean residual time \(R\) which is the average time taken to come to the end of the on-going LP or HP packet transmission (if such a transmission is currently going on) or the time taken to end the on-going walk time (if the Token is currently moving from one node to the next) when this packet arrives. From [4], \(R\) is given by
\[
R = [\lambda_1\text{E}([X_1]^2)] + \lambda_2\text{E}([X_2]^2) + \gamma (1-\rho_1-\rho_2)/2 \tag{2}
\]

(b) The average time taken to actually transmit all the priority 2 packets (from all nodes) that are transmitted between the time this packet arrives and the time when its transmission starts. This does not account for the priority 2 packet transmission, if any, which may be going on when this packet arrives; that is already accounted for in \(R\) above. Applying Little's Theorem, this will be \(\rho_2 W_2\).

(c) The average total Token walk-time \(Y_2\) between the arrival of this packet and the start of its actual transmission. From these three components of \(Y_2\) given in (3), \(Y_1\) will have a component \((\lambda_1 W_1\text{m})\text{m}\gamma\) contributed by priority 1 packets ahead of this packet in the priority 1 queue at node \(n_1\) where \(n_1\) is the average number of times a LP packet at node \(n_{i-1}\)starts transmission in order to finally succeed (ie after \(n_{i-1}\) preemptions). Another component \((\lambda_2 W_1\text{m})\gamma\) will be contributed by the priority 2 packets which arrive at node \(n_{i-1}\) while this packet is waiting in its queue. Finally, \(Y_1\) will have one more component \((n_{i-1})m\gamma\) to take into account the attempts to transmit this packet which were unsuccessful because of preemptions. From these, we get
\[
Y_1 = Y_2 + (n_1\lambda_1+\lambda_2)W_1\gamma + (n_{i-1})m\gamma \tag{5}
\]

Using (a), (b) and (c) above, the mean queueing delay \(W_1\) for priority 1 packets is
\[
W_1 = \frac{R + Y_2 + \rho_2 W_2 + (n_{i-1})m\gamma}{1 - \rho_1 - \rho_2 - n_1\lambda_1\gamma - \lambda_2\gamma} \tag{6}
\]
In order to obtain an estimate of \( n_1 \) and \( E(n) \), consider a system with a large number of users (i.e., large \( m \)) such that the arrival rates \( \lambda_1/m \) and \( \lambda_i/m \) at each node is small. Applying Little’s Formula, the average number of packets of priority \( i \) at any given node will be \( \lambda_i(W_i+E(X_i))/m \). Let \( p_{ji} \) be the probability of finding \( j \) packets of type \( i \) at a node at any arbitrary instant of time. Since \( \lambda_i/m \) has been assumed to be small, we can estimate \( p_{ji} \) as

\[
p_{ji} = (\lambda_i(W_i+p_i))/m, \quad p_{0i} = 1-p_{ii},
\]

\[
p_{ji} = 0 \quad i=1,2, j=2,3,\ldots
\]

(7)

For calculating the average number of times \( n_1 \) that transmission is attempted for a particular LP packet at a given node (say node \( i \)), let \( P_S \) be the probability that a LP packet transmission from node \( i \) is not preempted, given that it is actually started. Consider a particular LP packet transmission from node \( i \). This LP packet will not be preempted if none of the other \( (m-1) \) nodes have a HP packet waiting at the time the LP packet’s transmission starts (probability\( = [p_{02}]^{m-1} \)) and if each of these other nodes do not get a HP packet in the time it takes for the LP packet’s Connector (BT1) to reach that node (probability \( = \exp[-0.5\lambda_2(m-1)\gamma] \)). This yields

\[
P_S = [p_{02}\exp(-0.5\lambda_2\gamma)]^{m-1}
\]

(8)

\[
n_1 = 1/P_S
\]

(9)

To find the average number of nodes who currently have a Delayed PR connection at their respective NIUs (because of earlier LP packet preemption), we should find the probability that a node has a Delayed PR connection at its NIU rather than the usual PR connection. Instead of computing this probability, we make a pessimistic estimate for this by assuming that any node which has one or more LP packets will introduce the Delayed PR connection at its NIU. With this approximation –

\[
E(n) = m(1-P_01)
\]

(10)

which may be substituted in (1) to find \( \gamma \). (We are currently trying to improve on these pessimistic estimates.) Subject to the same assumptions, the analysis given above may be extended to systems with more priority classes of service.

Given the system parameters \( m, \beta, \tau \) and \( \lambda_i \), \( E(X_i), E(X_i^2) \) for \( i=1,2 \), (1)–(10) may be used recursively to find the corresponding values of the average packet delays \( W_1 \) and \( W_2 \). For numerical computations, we have examined a system with constant length packets of unit length – i.e., \( X_i = 1 \), \( i=1,2 \), with \( \beta=0.0001 \) between neighbouring nodes, bit duration \( \tau = 0.001 \) and \( m=1000 \). Computational results have been shown in Fig. 4 for a system where the arrival rates of priority 1 and 2 traffic are equal, i.e., \( \lambda_1 = \lambda_2 = \lambda/2 \) with \( \lambda \) as the total packet arrival rate. Since the packet length is unity, \( \lambda \) is also the total traffic offered to the system.

For purposes of comparison, Fig. 4 also shows the average packet delay performance of an equivalent ideal M/G/1 queue (actually M/D/1) where there are two priority classes and the service is non-preemptive. This may be considered comparable to the Ring/Bus protocol since the latter allows preemption of LP packets only during the brief initial portion of the packet. As shown in Fig. 4, for the given system, the delay performance of the Ring/Bus protocol is only marginally poorer than that of the ideal M/G/1 queue except for priority 1 packets at fairly high values of the offered traffic. This may be explained by the preemption effect described earlier.

The analysis presented here can be easily extended to a Ring/Bus system with more levels of priority. The assumptions and approximations made in this analysis would probably hold for most real systems. Simulation studies have confirmed this for systems with 2–8 priorities and as few as ten nodes. The greatest drawback in the analysis given is the extremely pessimistic value used to estimate \( E(n) \). We have noticed that if for a given set of parameters, \( E(n) \) is obtained from simulations and this value is used instead of (10), the computed results match the simulations almost perfectly. This gives us more confidence in both our (pessimistic) computer and simulation results.

Through simulations, we have also compared the performance of the prioritized Ring/Bus with an equivalent Token Ring system having eight priority levels. When the number of nodes is large and the improvement in RLT is significantly higher, the Ring/Bus system offers lower packet delays than the Token Ring for all priority levels. However, when the number of nodes is small (and the RLT
improvement is not very high), the packet delays obtained from the Ring/Bus system is much lower for the higher priorities; the delay reduction is less for the lower priorities. As a matter of fact, for typical choices of system parameters, the packet delay spread for the different priorities is somewhat larger for the Ring/Bus, i.e. lower delays for the higher priorities but higher delays for the lower priorities as compared to the Token Ring. This is probably the effect of the better priority resolution offered by the Ring/Bus system rather than being caused by the reduction in the system's RLT. The overall performance of the Ring/Bus system is sufficiently better than that of the Token Ring and hence this approach should be seriously considered for low to medium speed (less than 10 Mbps) LANs.

V. CONCLUSIONS

In this paper, we have proposed a new Ring/Bus protocol for providing multi-priority service in a ring-type LAN. For this architecture, conventional Token Rings implement point-to-point links between neighbouring nodes with each node acting as a repeater with one or more bits of delay. In contrast, the proposed system operates as two bus sections connected by an active repeater with a one-bit delay. The access method is still based on a circulating Token which is modified appropriately to a Connector by the node getting the access right for transmission. In the Token Ring, this is done by first checking if a Token or Connector is received and then changing the Token to a Connector if access is required – one or more bits of delay are required by each node to effect this change. The Ring/Bus protocol replaces this bit delay by a switching arrangement which effectively does the required change "on the fly". The advantage is a reduction in nodal delays so that the RLT of the system is substantially lowered leading to significant improvements in performance. The proposed Ring/Bus system will always have this advantage over a comparable Token Ring system. However, the actual improvement in performance will be significant when the total nodal delay is a strong contributor to the total RLT of the Token Ring. The proposed Ring/Bus approach can be used both for non-priority or multi-priority service though the latter is more complex to implement.

In the analysis presented in [5] for non-priority service and in Sec. IV for the multipriority implementation, the single one-bit active repeater delay of the Ring/Bus has been ignored as its contribution will be negligibly small. Moreover, the Ring/Bus protocol has been described only for non-exhaustive service with only one packet transmission per token. This can be easily extended to the case where the service is exhaustive or is limited—in nature.

For non-priority service, the Ring/Bus [5] performs better than both the comparable Token Ring and the CSMA/CD (ETHERNET) system. It provides both lower average packet delays as a function of offered traffic as well as higher capacities. Simulation studies have confirmed our analytical results on the performance of the proposed Ring/Bus system. Simulations comparing the Ring/Bus and Token Ring for multi-priority service also indicate that the former shows better overall performance and should be considered for implementation in LANs operated at speeds of 10 Mbps or less. Efforts are being made to build a suitable analytical model for the multi-priority Token Ring so that the two systems may be compared analytically. Hardware has already been developed to implement a non-priority Ring/Bus network — efforts may also be made to augment this to allow a multi-priority implementation.

ACKNOWLEDGEMENTS

The work reported in this paper was supported in part by the ERNET Project, jointly sponsored by UNDP and the Dept. of Electronics, Govt. of India. Their support is gratefully acknowledged.

REFERENCES


