Reducing Global Address Recognition Delays in Local Area Networks with Spatial Bandwidth Reuse

Rahul Simha

Department of Computer Science
College of William and Mary
Williamsburg, VA 23185
simha@cs.wm.edu

Yoram Ofek

IBM Research Division
TJ Watson Research Center
Yorktown Heights, NY 10598
ofek@watson.ibm.com

Abstract

A special class of local area networks have recently received much attention in which the destination nodes remove packets from the network, allowing spatial bandwidth reuse or concurrent transmission of packets without collision. Destination removal requires that each node will delay packets long enough so it can decode the destination address. If the address is a short label (say, 8-bit) the decoding time is short, but if we want to use global or MAC addresses of 48 or 64 bits, the sum of decoding times encountered by packets may result in unacceptably long delays, even at light traffic loads.

In this paper, we propose protocols for reducing global address recognition delays which are designed to supplement existing protocols for local area networks with spatial bandwidth reuse. They are also designed to coexist with other addressing modes used for individual LANs, which are based on the use of very short, locally administered labels. Of particular interest is a completely decentralized protocol which achieves the low global address recognition delays of multi-access protocols and yet seamlessly reverts to efficient spatial bandwidth reuse at high loads. We focus on the MetaRing and MetaNet architectures and apply our methods to these local area networks. A simple probabilistic model is presented which demonstrates significant improvement in delay when our protocols are used.

keywords: Local area networks, metropolitan area networks, computer communications, network protocols, routing protocols.

1 Introduction

Local area networks (LANs) with *spatial bandwidth reuse* allow packets to be transmitted concurrently and without collisions. One example of such a LAN is the buffer insertion ring, MetaRing [8], in which several concurrent transmissions may take place in distinct segments of the ring. In order to facilitate this simultaneous transmission without collisions, a packet must be removed from the ring by the destination so that segments of the ring beyond the destination are free to transmit. Destination removal of packets is in turn made possible when every node in the system is able to delay each packet long enough to decode or *recognize* the destination address. Thus, there is an *address recognition delay* at each node associated with the forwarding of each packet.

Reducing the recognition delay of destination addresses is important when address fields are long (as in the case of globally unique addresses of 48 to 64 bits defined by some international standards) and when there are several hundred nodes in the network. In these cases, the delay incurred by each packet, even in low loads, may be unacceptably large. This problem is alleviated by designing the system with very short address labels [8]. But using short labels, which are administered locally, may not be always possible in a complex inter-networking environment. In other words, short labels are conveniently used in a stand-alone LAN, but when packets arrive from external networks bearing globally unique destination addresses there is a need to be able to use global addresses directly for routing within the LAN. Global addresses are also more compatible with emerging standards.

In this paper, we propose a number of schemes to alleviate the problem of global address recognition delays. Since we focus on global addresses, we will use the terms address and global address interchangeably. Our protocols are designed to supplement existing protocols for LANs with spatial bandwidth reuse [2, 8, 9, 15], with special attention paid to the MetaRing [8] and MetaNet [9] architectures. Our main contribution is an examination of the problem of global address recognition delays. We present schemes that bridge the gap between the low address recognition delays of multi-access protocols and the increased performance of buffer insertion rings at high loads. One of the goals of this paper has been to ensure that our schemes for reducing address recognition delays integrate uniformly with existing access protocols for buffer insertion rings. In particular, we supplement the existing addressing modes on the MetaRing [8, 12] and MetaNet [9, 10, 11] access protocols in a manner that does not affect their efficiency and fairness at high-loads, but provides low global address recognition delays at light loads. We examine both centralized and decentralized protocols for the MetaRing. For the decentralized protocol, which is necessarily more complex in description, we provide a proof of correctness. Finally, through the use of a simple probabilistic model, we evaluate the performance benefits obtained by reducing average delay in address recognition. It is seen that at low to moderate loads, address recognition delays are significantly reduced.

A few comments may be made about related literature, especially research on buffer insertion rings. The buffer insertion ring [2] is an important contender among several LAN architectures designed to handle high bit rates. Studies in LAN performance [3, 4, 5, 16] have consistently reported high performance in the case of buffer insertion rings. This property together with its distributed operation have recently resulted in increased attention towards buffer insertion rings [7, 8, 15, 6]. However, buffer insertion rings suffer from the disadvantage that the architecture allows for the phenomenon of starvation [15]. In [8, 15], protocols are presented for eliminating node starvation and providing fair access to a full-duplex buffer insertion ring. The focus of these research efforts was to maintain the advantages of buffer insertion rings over competing designs such as token-ring, while solving the starvation problem.

To maintain the advantages of a buffer insertion ring over other LAN designs, our protocols are designed, as mentioned earlier, to supplement protocols which solve the node starvation problem. The essential idea is to introduce a minor architectural modification, a copying feature, that allows a node to forward an incoming packet while copying it into a special buffer simultaneously. In this case the forwarded packet incurs no address recognition delay at the node. We note that an analogous idea for token rings has been proposed with reported success in [13, 14]. In [13, 14], a token ring is augmented with the facility to operate with a CSMA/CD-type [1] contention protocol at light loads. Our protocols are designed for buffer insertion rings; the results maintain several advantages that buffer insertion rings have over other multi-access protocols [15]. Finally, the problem of address recognition delays is also addressed in [6]. In [6], packets are shortened online by the destination after the address is recognized and the shortened packets travels until it is discarded by a transmitting node. We note that the protocol in [6] permits endlessly circulating packets, if all nodes are idle immediately following a transmission. Our decentralized protocol is similar to theirs, except that the problem of an endlessly circulating packet is explicitly eliminated. Our work also differs from theirs in that our protocols are intended to supplement the MetaRing and MetaNet architectures; we also present a centralized protocol and describe how both protocols may be integrated into the above LANs.

In the next section, we describe our protocols for reducing address recognition delays. We first focus on buffer insertion rings; in §2.1, basic operating principles of the MetaRing are described and in §2.2 a decentralized protocol compatible with the existing protocol for the MetaRing is presented. In §2.3 we describe a centralized protocol which is useful because of its simplicity and because it provides a reference point for the discussion of the decentralized protocol. Issues related to integrating our protocols with existing protocols for the MetaRing are discussed in §2.4. The MetaNet architecture and protocol are considered in §2.5. Finally, in §3, a simple probabilistic model is described through which reductions in address recognition delays are characterized.

2 Protocols for Reducing Address Recognition Delays

2.1 Basic Operating Principles

In this section we briefly describe the basic operating principles of a buffer insertion ring, with particular attention paid to the MetaRing. For a more complete description of buffer insertion rings the reader is referred to [1, 2]. In a buffer insertion ring, also commonly referred to as a register-insertion ring, each node is provided with an insertion buffer to store packets transmitted on the ring. A node can start to transmit from its output buffer only if its insertion buffer is empty. While transmitting the node will store incoming traffic in its insertion buffer. As a result, collisions are avoided; however, packets may be delayed in several intermediate buffers before reaching their destination. Since a node transmits a locally generated packet only when the insertion buffer is empty, i.e., when there is no immediate ring traffic, an insertion buffer that is continually filled with ring traffic from upstream will prevent a node from transmitting. In extreme circumstances this phenomenon is referred to as *starvation* [15]. In [15], it is demonstrated that with typical LAN traffic of file transfers, starvation can be a serious problem. Protocols are presented in [8, 15] to overcome this disadvantage of buffer insertion rings.

One such protocol for insertion ring architecture is the MetaRing designed and implemented in the IBM T. J. Watson Research Center [8]. In the MetaRing, which consists of two concentric rings carrying traffic in opposite directions. Special control characters are used to regulate the access to the ring in a manner similar to, yet quite distinct from, that of a token ring [1]. In a token ring, a special circulating packet (the token) controls access to the ring: only the node in possession of the token is permitted to transmit. By controlling the transfer of the token, both access as well as flow-control (preventing starvation) are achieved. In the MetaRing, these functions are decoupled.

Access is achieved through the buffer insertion principle and the external flow and fairness are controlled by the SAT control signal, such that, during each rotation of the SAT each node can transmit a predefine number of data units.

From the point of view of address recognition delays, we note that for efficient use of bandwidth, the above protocols are designed to allow concurrent collision-free transmissions, a procedure which relies on the destination removal of packets. Thus, each node must stop every packet long enough to be able to decode the address field and determine whether the packet is to be removed. We argue that, in high loads, the delays do not matter since, in other designs such as the token-ring, these address-delayed packets would have to wait at the source node instead. In low loads, however, address recognition delays can form a significant portion of the delay experienced by packets. We are thus interested in protocols which resemble the 'free-for-all' multi-access protocols in low loads (and therefore incur minimal address recognition delays) and which seamlessly revert to the carefully regulated buffer insertion protocols at higher loads. In addition, it is important to retain the collision-free properties of the insertion ring architecture.

We define the following states of a given node, i, used in our protocols:

- *Inert-state* (IS) - the node is simply connected onto the ring, but neither receives nor transmits packets.
- *copy-state* (CS) - In this state, incoming packets are copied into the receive buffer while being forwarded *concurrently*. The forwarded packet, then, encounters no delay in passing through node i. Note that in [8, 15] it is assumed that addresses are decoded all the time and hence there is no *copy-state* in these protocols.
- *address-read-state* (AS) - In this state, incoming packets are held for a sufficiently long time so that source and destination address fields in the packet can be decoded. Then, depending on the dictates of the protocol, the packet can be placed into the receive buffer, discarded or forwarded downstream. Thus, in this case, the forwarded packet encounters a small delay at node i - the time required to decode the address fields.
- *buffer-state* (BS) - In this state, the entire incoming packet is buffered in the insertion buffer. This corresponds to the usual operation of the buffer insertion ring. Note that the address field may be decoded while a packet is in the insertion buffer.

The introduction of the copy-state is motivated by observing that at low loads very few nodes transmit concurrently and hence, there is no need for a starvation protocol to be in effect. Since most of the nodes are 'quiet', address recognition by intermediate nodes is unnecessary. Intuitively, when most nodes are in the copy-state, the access behavior resembles that of multi-access protocols in low loads. However, in high loads, when a regulatory mechanism is desired, the nodes are frequently in the address-read-state or buffer-state; this allows a buffer insertion protocol to provide high efficiency and fair access.

2.2 A Fully Decentralized Protocol

2.2.1 Description

We now describe a fully decentralized protocol which almost entirely eliminates address recognition delays at low loads. As will be seen, the protocol supplements the MetaRing access protocol and therefore, at higher loads, there is a smooth transformation of the ring to regulate access. We describe our protocol for a single unidirectional ring with the assumption that additional concentric rings use the same protocol. The following notation is used:

- A packet m is represented by the quadruple $m =< s, d, r, c >$, where s is the source or originator of the packet, d is the destination, r is a counter used by the protocol and c is the contents of the packet.
- Let $N = \{1, 2, ..., n\}$, $0 \notin N$, denote the set of nodes on the ring. We assume that the ring has a predetermined *direction* of packet travel [2].
- Define $B(s, d) = \{n \in N | n \neq s \neq d, n$ is a node encountered by $m =< s, d, r, c >$ in travelling directly from s to $d\}$. Loosely speaking, $B(s, d)$ is the set of nodes 'between' s and d that do not include s and d. The notation $|B(s, d)|$ is used to represent the cardinality of the set $B(s, d)$.
- $S_i(t)$ denotes the state of node i at time t. Where there is no ambiguity, the dependence on time will be dropped. Additionally, R_i is a counter at each node initially set to zero.

<u>**Message Transmission**</u>:

/* Node i transmits packet contents c to destination d */
construct $m =< s, d, 0, c >$

$S_i := AS$
$R_i := R_i + 1$
transmit packet m

Message Reception in Copy-state:

```
/* Node i receives packet m =< s, d, r, c > when in copy-state */
if d = i then
    receive packet m
else
    discard packet m
```

Message Reception in Address-read-state:

```
/* Node i receives packet m =< s, d, r, c > when in address-read-state */
if i ∈ B(s, d) and r ≤ |B(s, d)| then
    forward m =< s, d, r + 1, c >
else
    begin
        if d = i then
            retain packet m
        else
            discard packet m
        R_i := R_i − 1
        if R_i = 0 then
            S_i := CS
    end
```

Note that since addresses are read in the buffer-state, the same actions are taken in this state as in the address-read-state. Also, observe that different actions are taken by a node depending on the state and the fields of a packet. In the copy-state, a node cannot remove a packet from the ring, since it is simultaneously forwarded to other nodes; by the time the address fields are decoded a substantial part of the packet will have passed on to subsequent nodes. Thus, the only decision a node can make here is whether to discard or use the packet. The packet is discarded if it is intended for some other destination.

The actions taken when in the address-read-state need further elaboration. Intuitively, a node in the address-read-state keeps track of the packets it removes from the ring using the R_i counter. When the counter is low enough to facilitate transition into the copy-state, i.e., when the node is certain that it is 'safe' do so, the state is changed. The purpose of this counter and the set $B(s,d)$ will become clear in our proof of protocol correctness, a matter to which we next turn.

2.2.2 Protocol Correctness

With every decentralized protocol such as the one described in the previous section, there is concern about whether it will always operate correctly. Since the nodes behave independently and asynchronously it is important to determine the behavior of the protocol in all possible circumstances. Consider the following example in a 8-node unidirectional ring, in which nodes are numbered sequentially around ring. Node 1 sends $m_1 =< 1,4,0,c_1 >$, node 3 sends $m_3 =< 3,5,0,c_3 >$ and node 6 sends $m_6 =< 6,7,0,c_6 >$, all approximately at the same time. Since all are in the address-read-state, nodes 1 and 6 go into the copy-state after receiving m_6 and m_3, respectively. However, $3 \in B(1,4)$ and hence m_1 is not 'picked off' the ring by node 3 in the first pass. If, in the meantime, another node transmits and node 3 gets into the copy-state, then, it is not immediate from the protocol that m_1 does not circulate endlessly. We prove below that this cannot happen and the proof illustrates the use of the counter within the packet.

Proposition 1 *A packet $m =< s,d,0,c >$ does not circulate more than $|B(s,d)|$ times around the ring. In particular, since $|B(s,d)| < n$, a packet does not circulate more than n times.*

Proof: Assume that there is a packet $m =< s,d,r,c >$ that has circulated k times for $k > |B(s,d)|$. Let T_0 be the time of transmission of $m =< s,d,0,c >$, i.e., when the packet was first introduced into the ring and when $r = 0$. Similarly let $T_i, i \leq k$ be the time that packet passes by s for the i^{th} time.

First note that each transmission results in some counter R_i being incremented. Next, when a packet is removed from the ring, by some node in the address-read-state, the counter is decremented. Thus the number of packets in circulation at any given time is $\sum_{i=1}^{n} R_i$. In particular, for during each circulation of m, at least one counter must be strictly positive, and hence, the destination address of m (i.e., d) must have been read at least once in each of the k circulations $[T_{i-1}, T_i], 1 \leq i \leq k$. Furthermore, the address of m must have been read by nodes j such that $j \in B(s,d)$ or else m would have been removed from the ring (i.e., not forwarded). Then, since $j \in B(s,d)$, by the protocol, the counter r must have been incremented at least k times. But when $k > |B(s,d)|$, the packet is removed and hence we have a contradiction. ∎

Thus, we observe that the purpose of the counter and the definition of the sets $B(s,d)$ is to prevent extenuating circumstances from permitting an endlessly circulating packet. Intuitively, for the protocol to be truly distributed, the nodes $j \in B(s,d)$ need extra information in order to determine whether m has been 'seen' by the destination d. Since there are $|B(s,d)|$ nodes in the first traversal, each of these might have to mark the packet m and thus, a counter of size $|B(s,d)|$ appears to be required. We note that this operation of incrementing a counter can be an overhead but we argue that it will actually be used only infrequently. It is merely a safeguard for proper operation of the protocol in all circumstances. In low loads, the source is most likely to remain in the address-read-state and remove the packet itself. In high loads, nodes $j \notin B(s,d)$ will probably be in the address-read-state and will remove the packet sent by s. Note also that there is a cost associated with implementing the address recognition, usually accomplished through the use of some associative memory. However, it allows complete flexibility in changing network topology and global addresses.

2.3 A Centralized Protocol

It is possible to reduce the complexity of the protocol at each node by using a single controller to read addresses and remove unwanted packets from the ring. We now describe how this may be done. For each ring a special controller node stops each ring packet going through it for long enough to decode the address field and take an appropriate action. Only two actions are possible: remove the packet from the ring or permit the packet to continue traversing the ring but with a modified counter. A packet is removed from the ring when the controller is certain that the destination has received the packet.

We will use the notation of the previous sections. As before, a packet is denoted by $m =< s,d,r,c >$. In this case, r is a 1-bit counter. Initially, a packet is transmitted by the source with $r = 0$. Only the controller is permitted to modify the counter. On receiving a packet $m =< s,d,0,c >$ the controller determines whether m has been seen by d, i.e., whether d lies between s and the controller. If that is the case then m is removed from the ring. Otherwise, the controller modifies the counter to $r = 1$ and the packet $m = < s, d, 1, c >$ continues around the ring. On seeing a packet with counter $r = 1$, the controller always removes the packet. Thus, the controller knows when a packet has been seen and can limit the number of traversals made by a packet:

Proposition 2 *Under the centralized protocol, a packet makes no more than two traversals of the ring.*

With the centralized protocol, a node does not need to enter the address-read state after transmitting a packet. Indeed, no special precautions are needed since responsibility for removing packets that have reached their destinations now lies with the controller. Therefore, the complexity of the protocol at each node is reduced, as might be expected with centralization.

A few comments may be made in comparing the two protocols. First, the centralized protocol is conceptually simpler and results in a simpler node interface. Second, since no more than two ring traversals are possible for any packet, it records better worst-case behavior than the decentralized protocol. Third, a smaller counter (1-bit only) is needed in the centralized protocol. In contrast, we point out that the decentralized protocol is less failure-prone. The centralized version causes serious problems if the counter does not get set properly or if it is corrupted (consider what happens when r is changed to zero). A packet is likely to travel longer distances before removal in the centralized protocol. In the decentralized protocol, most address checks past the destination will remove the packet. Finally, the worst case behavior for the decentralized protocol is much less likely to occur than for the centralized protocol. In the centralized version, a substantial portion of the packets will always traverse the ring twice. This will reduce the overall bandwidth, making the centralized protocol far more inefficient than the decentralized protocol.

2.4 Integration with the MetaRing

Both the centralized as well as the decentralized protocols in the previous sections supplement the MetaRing access protocol. Consider the case when the decentralized protocol is implemented along with the MetaRing access protocol. A node enters the address-read state after transmitting only to remove the possibility of an endlessly circulating packet. However, if it knows, through the protocol, that some other node has removed the packet, then it returns to the copy state. In light traffic, when no one is transmitting anyway, the packet returns to the transmitting node with no overhead on the system and is removed. As the traffic increases, i.e., as the load on the system is

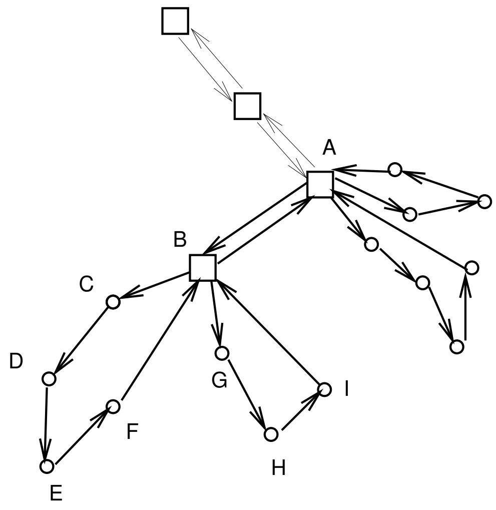

Figure 1: MetaNet: A Tree Embedded Ring

increased, the incoming packets will tend to get buffered or discarded after the address is decoded and thus the ring will tend to behave as if exclusively under the MetaRing access protocol. Finally, when the traffic is high enough to cause starvation, the SAT control signal regulates access to the ring.

In practice, a combination of the centralized and decentralized protocols is likely to be used — a *combined* protocol. The combination is simply the superposition of the two protocols, with both operating simultaneously and independently. The purpose of adding a special designated controller in spite of the self-sufficiency of the decentralized protocol is to take further precautions against corruption. If address fields get repeatedly corrupted then the controller node will detect two successive circulations of the corrupted packet and be able to remove it.

2.5 A Partially Decentralized Protocol and the MetaNet

In this section we consider the new MetaNet proposal [9] and examine the possibility of reducing address recognition delays in this architecture. In particular we focus on a Tree Embedded Ring (TER) [9]. Consider the unidirectional TER shown in figure 1. Branch nodes such as node A in the figure allow packets *not* destined for any of nodes C, D, E or F to *cut through* (in the terminology of [9]) node A directly into the segment GHI. For this reason, the branch nodes (in [9], nodes consisting of multiple virtual nodes) must read the address field of incoming packets. As it stands, all other nodes in a destination removal scheme must also read addresses. However, we may easily modify the protocol to strike a compromise between destination removal and the fully decentralized solution presented earlier in this paper. Since each branch node must read the address field of each packet, the branch nodes can act as controller nodes for segments of the tree. For example, nodes C, D, E and F need not perform address reads when idle; the branch node B, which reads addresses, can remove packets which arrive from F and which were destined for any of C, D, E or F. Thus, a packet will usually never travel beyond the branch node of the segment containing the destination. Since the packet removal function is assigned to the various branch nodes in the TER, this solution is partially decentralized. We assume that topology information is available at the branch nodes in order to make routing decisions. Note that a balanced tree design implies that a packet will typically travel through approximately $O(\log n)$ address reads in a network with n nodes.

As with the protocols in the previous sections, a counter is needed in each packet for special cases. The counter is once again a 1-bit counter taking values $r = 0$ and $r = 1$. Consider the following example. Node F sends a packet $m =< F, C, 0, c >$ to node C. Because of congestion [9] there may be no opportunity for m to cut through any branch node, and m may have to traverse the whole ring. In this case, branch node B must let the packet from F pass through on to node G. The counter is needed to distinguish between two successive trips through node B; the counter is set to $r = 1$ on the first trip. On the second trip, node B removes the packet. Node B can do this because sufficient topology information is available at B. In this manner packet removal can be achieved on the MetaNet and redundant packet travel is minimized. We state this as:

Proposition 3 *Under the partially decentralized protocol, a packet makes no more than two traversals of the ring.*

3 Performance Benefits

In this section, we estimate performance benefits accrued by incorporating the protocols presented in this paper. As a test case, we consider a unidirectional ring and focus on a randomly selected packet. We wish to determine the expected number of address reads that the packet will encounter in travelling from source to destination. We will assume that given a source node, a destination is chosen randomly from the other nodes in the ring with equal probabilities. For convenience, let $(n+2)$ denote the number of nodes in the system. Then, the probability of choosing any particular destination is $\frac{1}{n+1}$. Let A denote the random variable representing the number of address reads. Next, let p denote the probability that a node performs an address check. Thus, p is an indication of load in the system. Low values of p indicate that most nodes are inactive whereas, as $p \to 1$, the system gets saturated. We now characterize the random variable A by computing its generating function. Note that $A \in \{0, \ldots, n\}$. Let the random variable D denote the randomly chosen destination, $D \in \{1, \ldots, n+1\}$.

$$\begin{aligned}
g_A(z) &= E\left[z^A\right] \\
&= \sum_{i=1}^{n+1} E\left[z^A | D = i\right] P\left[D = i\right] \\
&= \sum_{i=0}^{n} (1 - p + pz)^i \frac{1}{n+1} \\
&= \frac{1 - (1 - p + pz)^{n+1}}{p(n+1)(1-z)}
\end{aligned}$$

From the expression for $g_A(z)$, we can compute the mean and variance of A:

$$E\left[A\right] = g_A'(1) = \frac{np}{2}$$

$$\mathrm{Var}[A] = g_A''(1) + g_A'(1) - \left(g_A'(1)\right)^2 = \frac{p^2 n^2}{12} + \left(\frac{p}{2} - \frac{p^2}{3}\right) n$$

In the case, when address reads are performed at all nodes, we will B to denote the number of address reads. In this case, it is easy to show $E\left[A\right] = \frac{n}{2}$ and $\mathrm{Var}[A] = \frac{n}{6}(\frac{n}{2} + 1)$. Figures 2 and 3 we plot the expected value and variance of A for different loads. The two figures are for different numbers of nodes, $n = 10$ and $n = 50$ respectively. As the figures show, considerable improvement in the statistics of address recognition delays may be obtained using the protocols in this paper.

4 Conclusions

We observed that spatial bandwidth reuse LANs must delay packets long enough to read their address fields and argued that this can be a serious problem with a large number of nodes, especially where global addresses are used. In this paper, we focused on reducing this address recognition delay and presented protocols to achieve lower delays. The protocols were designed to supplement existing protocols for spatial bandwidth reuse architectures such as the MetaRing and MetaNet.

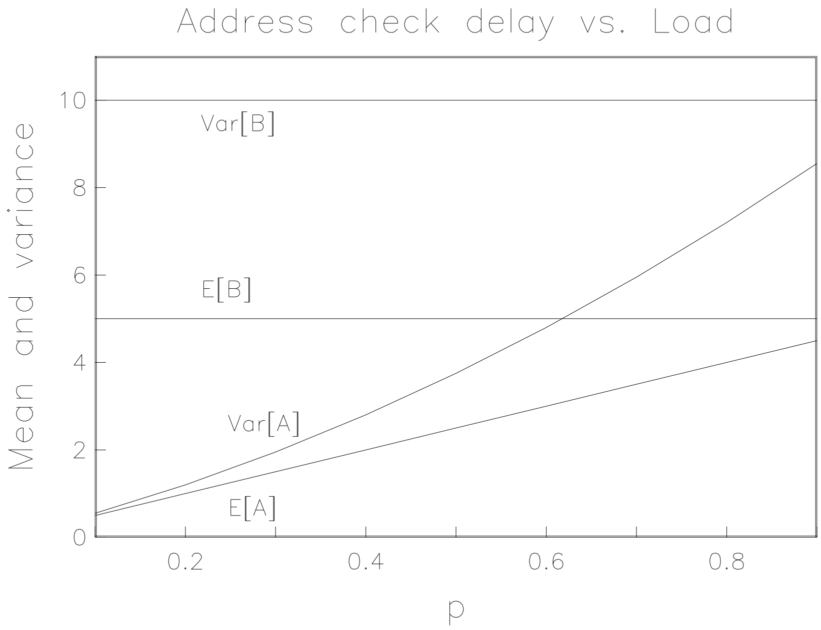

Figure 2: Address check delays with increasing load (10 nodes)

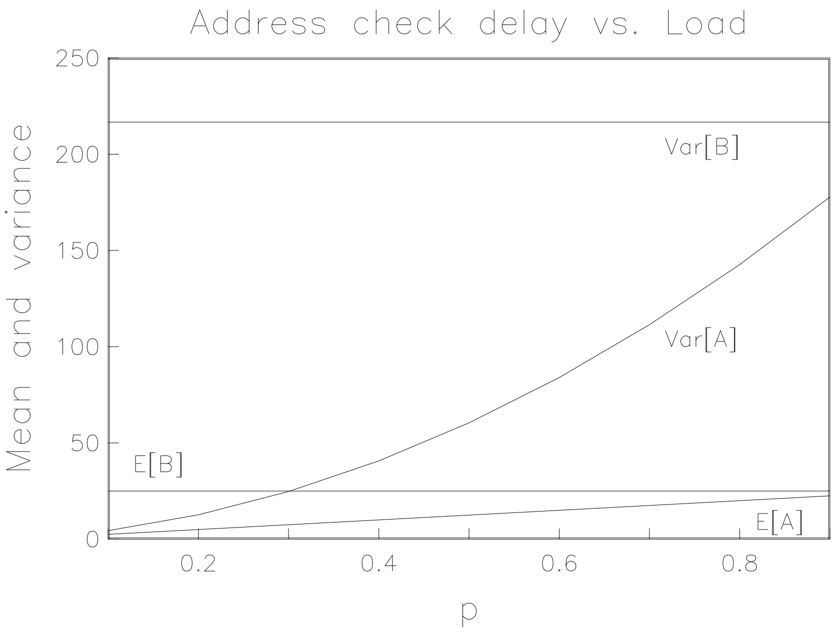

Figure 3: Address check delays with increasing load (50 nodes)

We presented both centralized and decentralized protocols together with proofs of correctness. A simple probabilistic analysis to assess the reduction in address recognition.

References

[1] D.C. Flint, "The Data Ring Main - An Introduction to Local Area Networks", *Wiley*, 1983.

[2] E.R.Hafner, Z.Nenadal and M.Tschanz, " A Digital Loop Communication System", *IEEE Trans. on Communications*, June 1974.

[3] W.Hilal and M.T.Liu, "Analysis and Simulation of the Register-Insertion Protocol", *Proc. Computer Networking Symposium*, 1982.

[4] M.T.Liu and D.M.Rouse, "A Study of Ring Networks", Ring Technology Local Area Networks", I.N.Dallas and E.B.Spratt (Editors), *Elsevier Science Publishers*, 1984.

[5] W.M.Loucks, V.C.Hamacher, B.Preiss and L.Wong, "Short-packet Transfer Performance in Local Area Rings", *Proc. of IEEE Infocom.*, San Francisco, 1984.

[6] M.Xu and J.H.Herzog, "Concurrent Transmission and Short Loop Latency Protocol", *Local Computer Networks Conference*, 1988.

[7] M.Misson, J-J.Mercier and A.El.Oussoul, "A Fair Management of Communication Establishment for a Ring LAN", *Local Computer Networks Conference*, 1988.

[8] I.Cidon and Y. Ofek, "MetaRing: A Full-Duplex Ring with Fairness and Spatial Reuse", *IEEE Trans. on Comm.*, Vol. COM-41, No. 1, Jan 1993, pp. 110-120.

[9] Y.Ofek and M.Yung, "Principles for High Speed Network Control: Loss-lessness and Deadlock-freeness, Self-routing and a Single Buffer Per Link", *Proc. Prin. Dist. Computing*, August 1990, pp. 161-175.

[10] Y.Ofek and M.Yung, "Asynchronous Lossless Broadcast-with-Feedback on the MetaNet Architecture", *Proc. INFOCOM*, April 1991, Bal Harbour, Florida, pp. 1050-1063.

[11] Y. Ofek and M. Yung, "Routing and Flow Control on the MetaNet: an Overview, *Computer Networks and ISDN Systems*, to appear.

[12] Y.Ofek. "Overview of the MetaRing Architecture", *Computer Networks and ISDN Systems*, to appear.

[13] H.Okada and S.Ohno, "A new resolvable-contention Access Control Scheme for Ring-typed LANs with High-speed Links: Rec-ring", *Proc. of ICC*, June 1987.

[14] H.Okada, S.Saekol and Y.Tezuka, "Performance Considerations on Resolvable-contention Ring Networks for Fiber Optic LANs", *Proc. INFOCOM*, April 1989.

[15] R.Simha and Y.Ofek, "A Starvation-free Access Protocol for a Full-duplex Buffer Insertion Ring Local Area Network", *Computer Networks and ISDN Systems*, Vol. 21, pp. 109-120, 1991.

[16] A.Thomasian and H.Kanakia, "Performance Study of Loop Networks Using Buffer Insertion", *Computer Networks*, Dec 1979.