M.I.T. Laboratory for Computer Science

Local Network Note No. 33
(Revised: March 22, 1983)

A COMPARISON BETWEEN TWO TOKEN-PASSING RING NETWORK DESIGNS

by Jerome H. Saltzer

Abstract

This paper examines two very similar token-passing ring networks, one designed by the IBM Zurich Research Laboratory and the other by the M.I.T. Laboratory for Computer Science, and identifies sixteen points on which the two designs differ. The three most significant differences are maximum link lengths (differing by a factor of ten), distributed as compared with designated ring supervision, and the extent to which a site-wide naming plan is embedded in the link level. Of these only the third is visible to users of a network, and thus of potentially far-reaching impact. The paper concludes that the differences are less significant than the similarities. Nevertheless, considerable insight into the design issues of local networks can be obtained by examining the reasons behind the differences.

Introduction

Recently the IBM Zurich Research Laboratory published several papers describing an experimental 4 Mbit/sec, distributed-control, token-passing, star-topology ring network that uses balanced transmission media, differential Manchester code, and synchronous detection with a phase-lock loop [1,2,3,4]. The M.I.T. Laboratory for Computer Science has designed a 10 Mbit/sec, distributed-control, token-passing, star-topology ring network that uses balanced transmission media, differential Manchester code, and synchronous detection with a phase-lock loop [5,6,7]. The conceptual approach taken by both designs is essentially the same, and the implementations are sufficiently similar that the paper "Why A Ring?" [8] applies without modification to both. At the same time, there are several differences in design details and some differences in design philosophy. This paper identifies the points of design difference and comments on the potential significance of those differences. It assumes that the reader is familiar with the design of at least one of the two rings. For brevity, this paper refers to the two designs respectively as the IBM ring and the M.I.T. ring.

* Author's address: Massachusetts Institute of Technology, Laboratory for Computer Science. Room NE43-505, 545 Technology Square, Cambridge, Mass. 02139 U.S.A.

The reader should recognize that the author's understanding of the two ring designs is not equally deep. His knowledge of the IBM ring comes from the published papers, while his knowledge of the M.I.T. ring comes from being one of its designers. Thus errors, omissions, and excessive use of imagination are more probable in the descriptions here of the IBM work, and especially in discussion of motivation for design decisions in that ring.

Both of the ring networks have undergone some evolution between their initial conception and their current implementation. The version of the IBM ring analyzed here is the one described in the cited papers. The version of the M.I.T. ring analyzed here is the one commercially available from Proteon Associates of Waltham, Mass., under the trademark PRONET.

Summary of similarities and differences

It is important, while reading this paper, to retain a sense of perspective. The paper emphasizes the differences between the two designs, with an eye to tracing origins and discussing implications. That emphasis is quite misleading, however, because in reality the two designs are much more similar to each other than either is to, say, the Cambridge ring [9], the Century data bus [10], the IBM Series 1 ring [11], or even the token-passing Primenet [12] and Apollo [13] rings. To reinforce this point, following is a recap of the similarities between the IBM ring and the M.I.T. ring:

- ring of digital repeaters
- distribution panel (star-shaped) topology
- token used for control of access
- originator removes its own message
- zero-delay protocol design
- each message originator drains any defective data from ring
- no special central monitor station
- phase-locked loop used for clock recovery
- balanced, shielded, twisted pair for wire transmission
- optical fibre links optional
- pulse transformers for station-to-station DC isolation
- differential Manchester code for phase insensitivity
- each station supplies its own power

For further information on these points of similarity, the already-cited references on the IBM ring and the M.I.T. ring provide details. In addition, the Communication Products Division of the IBM Corporation has made a series of presentations to standards committees and conferences in which it has offered comments on the relative advantages of token-passing rings, and suggestions on specific design alternatives [14,15,16]. Many of the comments and suggestions made in the presentations concern points in the above list.

The points of difference between the IBM ring and the M.I.T. ring fall into four general categories. In this list, the IBM ring parameter appears first, and M.I.T. ring parameter second:

1. link length goal (2 Km vs. 200 m)
   - 1.1 data rate (4 Mbit/sec vs. 10 Mbit/sec)
   - 1.2 predistortion vs. on-off signalling
   - 1.3 controlled vs. un-controlled line impedance
   - 1.4 cyclic redundancy check vs. link parity check
2. ring supervision philosophy (designated vs. distributed)
   - 2.1 supervisor selection by number passing vs. contention
   - 2.2 clock rate coordination (master clock vs. distributed clock)
   - 2.3 token failure detection (supervision bit vs. time-out)
3. protocol features
   - 3.1 address size (32-bit vs. 8-bit)
   - 3.2 token release (delayed vs. immediate)
   - 3.3 priority (yes vs. no)
   - 3.4 flow control (no vs. yes)
   - 3.5 protocol validation vs. protocol debugging
4. Miscellaneous physical level differences
   - 4.1 protocol signal method (code violation vs. bit stuffing)
   - 4.2 ground (central vs. per station)
   - 4.3 relay control (current vs. voltage)
   - 4.4 clock resynchronization (PLL freeze vs. stepped delay)

These differences are discussed in turn in the following sections.

1. <u>Link length</u>. Probably the biggest difference in design philosophy is in the design of the point-to-point link that connects one ring repeater with the next. The M.I.T. ring emphasizes simplicity of implementation, and to this end specifies a maximum station-to-station distance of 200 meters, anticipating application within a single office building. Long runs were

anticipated as an extra-cost option, for which a fibre-optic line extender is available. The IBM ring transmission system emphasizes the maximum possible station-to-station distance in a single design, and it achieves an allowable distance of 2 Km. between stations. Thus a single repeater is applicable to short runs within a building, to inter-building connections, and to long runs in manufacturing plants. Presumably the approach of the M.I.T. ring is intrinsically less expensive to implement, while the approach of the IBM ring simplifies installation and configuration management, since only one part number, without options, is involved. One might expect that VLSI implementation will equalize the costs, leaving the IBM ring with a long-term functional advantage.

This difference in design philosophy leads to several differences in design detail, discussed in the following sections: data rate, predistortion, impedance control, and cyclic redundancy check.

1.1 Data rate. The most obvious specification difference between the two ring networks is in their data rate: 4 Mbit/sec for the IBM ring and 10Mbit/sec for the M.I.T. ring. (For both systems, the maximum transition rate of the differential Manchester code is twice the nominal data rate.) This 4 versus 10 Mbit/sec difference stems directly from the difference in emphasis on transmission distance; all other things being equal, a lower data rate leads to less intersymbol interference for a given cable length, or else a longer cable run for the same tolerated intersymbol interference. The IBM ring data rate of 4 Mbit/sec is the maximum consistent with reliable transmission over a 2 Km. distance. The M.I.T. ring data rate of 10 Mbit/sec, on the other hand, is the maximum that discrete TTL technology and simple data decoding circuits allowed. (The circuitry and transmission system of the M.I.T. ring are actually designed to operate up to 16 Mbit/sec; the 10 Mbit/sec specification speed is intended to allow margin for production-line variations.) Another potentially significant consequence of the difference in speed lies in ease of VLSI implementation of the repeater. A student project undertaken to create a VLSI ring controller similar to the M.I.T. ring led to a design with a maximum capability between 3 and 5 Mbit/sec with easily accessible and modest cost HMOS VLSI technology.

From an application point of view, the difference between 4 Mbit/sec and 10 Mbit/sec is probably less significant than appears on the surface; peak data rates on local-area networks are a little like peak power output specifications of hi-fi amplifiers: they may have more advertising value than they have real effect on applications in the field. Because of software bottlenecks, few systems can achieve data rates of greater than 500 Kbit/sec when sending to or receiving from a network; even that rate may continue only momentarily until the computer must pause to figure out what to send next. Data rates above this level must be justified either on the aggregate behavior of many simultaneous network uses or of specialized sources and sinks of data. A 4 Mbit/sec system can handle 8 simultaneous connections of the suggested intensity, among 16 stations. If each station actively sends data 10% of the time, 160 stations could be handled. The reports by Shoch and Hupp of Ethernet measurements suggest that in real environments, 150 stations may only generate loadings in the range of 1 Mbit/sec during the busiest second of the day [17]. Thus one should probably not get very excited about the difference between 4 Mbit/sec and 10 Mbit/sec. Put another way, if in some application aggregate loading threatens to approach 4 Mbit/sec, one should probably be looking at 100 Mbit/sec technologies rather than 10 Mbit/sec stopgaps.

1.2 Predistortion. The 4 Mbit/sec IBM ring uses a coding technique called predistortion to reduce inter-symbol interference at the receiving end of long transmission lines. (With predistortion, much of the enery content of a bit is concentrated near the beginning of that bit, thereby helping charge a long line more rapidly, and effectively giving more time for the line to recover before the next bit is transmitted.) The M.I.T. ring uses simple on-off waveforms. This difference again directly stems from the difference in interest in long cable runs. Predistortion presumably requires a more complex transmitting circuit, to control the timing of the energy distribution in each transmitted bit.

1.3 Impedance control. The IBM ring specifies use of a low-loss, controlled impedance, balanced shielded cable, known in the trade as Twinax. The M.I.T. ring specifies a thinner, cheaper and easier-to-install shielded twisted pair, although it allows Twinax when long runs might exceed slightly the planned-for internode distance. This difference is again a consequence of the difference in philosophy on maximum inter-repeater distances. It probably leads to an important difference in cost and related operational issues. Twinax requires relatively expensive connectors that take some expertise to install correctly in the field, while the shielded twisted pair can be more-or-less casually soldered or crimped into, for example, a D-connector. Twinax takes up substantially more duct space than shielded, twisted pair; in some situations duct space is a very precious commodity. Pulling Twinax through ceilings, false floors, and ducts is hard work that requires care to avoid kinks. On the other hand, Twinax, once installed, is probably more durable and less likely to be mistaken for a telephone or control wire.

1.4 Cyclic redundancy check. The IBM ring specifies that each transmitted packet be accompanied by a 16-bit cyclic redundancy check. The M.I.T. ring calls for no checksum whatsoever, although a single link parity check is provided to help warn of and isolate a trouble-causing link or repeater. This difference comes partly from the internode transmission length difference, and partly from a different perspective on the importance of detecting all data transmission errors at the lowest link level. There is a tradeoff between link length and raw error rate; if one wishes to maximize link length, it is usually done by allowing a non-negligible raw error rate, and providing redundancy to detect and possibly correct errors. This line of reasoning appears to have been applied to the IBM ring design, whereas the M.I.T. ring with its shorter wire runs chose the approach of designing for a negligible raw error rate.

The second consideration in the use of the cyclic redundancy check in the IBM ring and no check in the M.I.T. ring is more interesting and also more controversial: following an end-to-end argument [18], the M.I.T. ring is designed on the assumption that it is acceptable to allow undetected errors at the link level so long as they are infrequent. The reasoning is that similar errors can occur elsewhere, unprotected by the CRC, anyway. Any application that considers data errors important will check for such errors at a higher level. The IBM ring CRC is provided in accordance with an older school of design that insists that higher application levels should not be burdened with worries about possibly incorrect data. This difference in philosophy is probably attributable to more experience with protocols for advanced distributed applications in the M.I.T. ring design team, and more experience with terminal-to-computer links using telephone lines and HDLC in the IBM ring design team. It is not yet clear which of these approaches is preferable. Using a CRC adds circuitry and cost, but it reduces the frequency of higher-level recovery operations. (In a prototype 1 Mb/sec ring at M.I.T. there was a CRC; the only errors ever recorded by those circuits were traced to faults in the circuits themselves!) One reason why the approach of omitting redundancy checks may work out in practice is that an idle ring actually transmits a token continually;

if any link or repeater begins to exhibit a higher error rate, the token will probably be the first thing affected. Since repeated token loss is intolerable, such troublesome links will quickly be discovered, taken out of service, and repaired. Put another way, if the ring is in good enough shape to keep a token circulating, the chance that it will make data errors is already so small that error-detecting machinery doesn't provide enough improvement in error rate to be worth the effort.

2. Ring supervision. Both the IBM ring and the M.I.T. ring use a design philosophy in which all stations are identical; there is not a specially designed master station. At this point the design philosophies of the two rings part company. In the case of the IBM ring, some single station, dynamically selected, performs three supervision functions: it sets the clock rate for the entire ring, it launches a token, and it monitors the ring for lost or damaged tokens. Two of these functions, setting clock rate and monitoring the token, are continuing responsibilities, so we might describe the IBM ring as providing a dynamically selected master station. The dynamic selection distinguishes the IBM ring from the Cambridge ring [9] in which a specially designed and thus statically selected master station provides these (and other) functions.

Thus the IBM ring has one peer-oriented distributed algorithm, for selecting the supervisor. In contrast, the M.I.T. ring has three: ring initialization, clock rate determination, and damaged token detection. We discuss these three in turn.

2.1 Ring initialization. In the IBM ring, any station may discover lack of supervision, by noticing token loss. It then enters a supervisor mode that consists of repeatedly sending, using its internal clock, its own station number to its next neighbor. This transmission will cause the neighbor to enter the supervisor selection mode. Thus in the time it takes for a signal to propagate around the ring, every station will enter supervisor selection mode. Stations that are in supervisor selection mode operate with their receive and transmit sides disconnected: but each compares the incoming station number with the one it is repeatedly sending, and if the incoming one is larger it lapses back into repeater mode. Thus after a little while every station is repeating the station number of the highest-numbered station that is participating in the algorithm. That station, upon receiving its own number from its transmitting neighbor, realizes that it has been selected as the new supervisor and it proceeds to take over supervision of clock, launch a new token, and watch for token failure.

The M.I.T. ring approaches ring initialization in quite a different way. When any station wishes to originate a message, it watches for a token. If no token appears after one ring transit time, the station simply sends its message, followed as usual by a token. Thus sending of the first message has a side effect of initializing the ring. If two stations happen to try to originate messages on an uninitialized ring at about the same time, a collision will occur, much as in the Ethernet. Each station will drain the other's message, both will notice failure, and both will enter a backoff-retry algorithm. Note that this contention scheme is used in the M.I.T. ring only to recover from token loss, rather than, as in the Ethernet, for every message.

The two approaches differ in several ways. The IBM ring algorithm requires more elaborate hardware, for sending and comparing station addresses; the M.I.T. ring approach is implemented with only a timer. (Since contention is rare, the backoff-retry algorithm is implemented in host software.) The IBM ring algorithm operates correctly and in the same time no matter what the ring load, while the M.I.T. ring approach has a probability of collision that increases as the number of stations and the presented load increases. The IBM ring algorithm depends for its correctness only on accurate detection of the need for supervision; the M.I.T. ring algorithm depends also on individual stations each performing backoff-retry consistently.

Against the possibility that contention will prove inappropriate under heavy load, a virtual-token algorithm has been designed, but not yet implemented, for the M.I.T. ring [7]. In this approach, a station that detects ring failure jams the ring, causing each station to set a timer proportional to its own station number. The lowest-numbered station's timer will go off first, and that station launches a token to initialize the ring. One reason this virtual token algorithm has not been implemented is that in practice, the need for supervision (lost token) seems to be so infrequent that the exact algorithm used is not very important, so long as it work reliably.

2.2 Clock rate coordination. In the IBM ring, every station has two clocks; a receive clock, derived by a phase-locked loop (PLL) synchronized to the received signal, and a crystal oscillator running at the nominal ring data rate. During supervisor selection, each station transmits with its crystal clock and receives with its PLL clock. Whenever a station determines that it is not to be the supervisor, it switches over to use its PLL clock for transmission also.* Thus as part of the supervisor selection process, all non-supervisor stations become slaved to the supervisor, whose crystal oscillator thus becomes the master clock. The supervisor station continues to operate with both clocks (which are synchronized in frequency but differ in phase) and it provides a buffer to absorb the phase difference and any accumulated phase jitter.

In the M.I.T. ring, a genuinely distributed clock is implemented. Each station has a crystal oscillator that runs at the nominal ring data rate, but that can be adjusted slightly; the adjustment range is somewhat greater than the specified frequency tolerance of the crystal, so that there is guaranteed to be some common part of the frequency spectrum within the adjustment range of every station. In each station, a narrow-band PLL monitors the received signal and drags this station's crystal oscillator to match. Thus is created a ring of PLL-controlled crystal oscillators, all peers. The ring homes in on some frequency within the adjustment range that has the property that the ring transit time is an integer multiple of the bit time. If any station finds that it is operating at one end or the other of its adjustment range, it clamps its oscillator frequency to the center of the adjustment range, increases or reduces the amount of delay through that station by a small fraction of a bit time, and then unclamps the oscillator. This step adjustment is repeated every few milliseconds until all stations are satisfied with the mutually chosen operating frequency. (The stepped delay adjustment performs the same function as the buffer between clocks of the supervisor station of the IBM ring.)

The M.I.T. ring scheme has the advantage that all stations operate the same way at all times. In addition, analog initialization (frequency backup) is quite independent of digital initialization (token launching). It has the disadvantage that a ring of locked oscillators is a feedback

---

* This time of switchover is implied, but not explicitly described, by the published papers [1,2].

system, which is potentially unstable. Careful analysis and design are required to insure reasonably rapid initialization and stability in the face of small and large transients. The M.I.T. ring design parameters are chosen to be stable with more than 100 repeaters; so far, the design has been field-verified to be stable with 25.

2.3 Token failure detection. In the IBM ring, every message header contains a supervision bit. The message originator places a zero in this bit, and as the header passes the station that is currently providing supervision, the supervisor changes it to a one. If the ring supervisor notices a message header in which the supervision bit is already on, it resets the ring by draining it, and then reinitializes by launching a new token.

The M.I.T. ring does not specifically supervise individual messages; it relies instead on a token timeout to discover need for reinitialization. Every station has a token timer, and whenever any station that needs to originate a message detects that the token has been lost, it simply sends the message. In addition, as in the IBM ring, whenever any station originates a message, it drains all data ahead of its message from the ring; this rule tends to clean up any circulating garbage that might accidentally look like a valid message or extra token, leaving behind only a single, circulating token. If its own message never returns, it drains the ring long enough to insure that there is no token at all, so that the contention procedure can take over.

In practice, the M.I.T. ring strategy has proven to work acceptably, but with one minor flaw of interaction with the initial software. A circulating pattern may accidentally arise that consists of the beginning of a message with a valid address. The addressed station will start to accept this message fragment, but if the fragment is incomplete, it will immediately report the improper format to its host by an interrupt, reset its buffers, and begin listening again. Since the pattern is circulating, the addressed station will on the next circulation start to accept the message fragment again, and generate another interrupt. The resulting machine-gun barrage of interrupts continues until token timeout occurs, or the software handling the interrupts is clever enough to figure out that it should not simply reenable further interrupts. (The initial software was not so clever.)

3. Protocol features. In the previous two areas, link length and supervision, a single philosophical difference led to several differences in detail, each with its own implications. In the selection of link protocol features, however, the situation is more a matter of independent menu selection. Perhaps the one underlying philosophical difference, but one which seems to have affected only some of the menu selections, is the IBM ring application of protocol validation technology. Protocol validation certainly influenced choices in the area of ring supervision (already discussed) and token release timing.

3.1 Addressing. The M.I.T. ring has an 8-bit hardware-recognized address. This field size was chosen to emcompass the maximum number of stations that might be physically attached in a single ring. (That maximum number was in turn determined by considerations of jitter accumulation through long strings of repeaters, ease of configuration management, and reliability.) This design was developed in reaction to an elaborate 32-bit address with variable masks that the University of California at Irvine specified for its distributed computing system ring. That addressing hardware, designed for a particular distributed computing experiment, never proved to be of much use when M.I.T. applied it to the more general local network application. The choice of an 8-bit address field assumes that any higher-level internetwork interconnection architecture is handled above the physical link level; by choosing an 8-bit address field at the physical link level, the design forces all logical name management to higher protocol levels. In particular, the M.I.T. ring can be used in an obvious way with any of the United States Department of Defense TCP/IP naming plan [19,20], the Xerox Corporation NS naming plan [21], or the International Standards Organization X.25/X.75 naming plan [22,23].

The IBM ring station recognizes a 32-bit address field that is divided into a 16-bit ring number and a 16-bit station number. This approach introduces two distinct elements of philosophical difference with the M.I.T. design. First, by explicitly identifying distinct address fields for ring number and station number (and providing separately controllable hardware recognition of one, the other, or both) the physical link level of the IBM ring includes elements of an internet implementation. This approach allows construction of a bridge between rings that can be very efficient since no link-level repackaging of any kind is needed when a packet is forwarded.

(And such a bridge, called a block switch, accompanies the IBM ring [24].) The second element of philosophical difference lies in the choice of field sizes. A 16-bit node address can distinguish among many more stations than can be physically attached to a single ring, and a 16-bit ring number can distinguish among many more rings than are likely to be interconnected by a high-speed bridge. The larger-than-necessary address fields of the IBM ring seem to be provided so as to permit some logical address management.

For example, one could assign each station a station number that is uniform across all stations on all attached rings, and then moving a station from one ring to another would not require the mover to discover and assign an unused station number on the new ring. Similarly, one could combine two physical rings into one without assigning new station addresses. Curiously, another logical address management feature is also provided in the IBM ring; the station can set both parts of its own address: a protocol for address initialization that involves a name server seems to be required. Since this protocol would also allow stations to change physical attachment points with a minimum of fuss, this may be a "belt and suspenders" design. In addition, the choice of field sizes, while encouraging unique assignment, does not permit universal address assignments of the kind that the Xerox Corporation NS protocol plans. (That protocol uses 48-bit station identifiers and 32-bit network identifiers, which could be permanently assigned by equipment manufacturers and thus be distinct and unique even across customers, anticipating future inter-enterprise interconnection [25].) The IBM ring approach, by catering to a name management plan of limited size, requires that a further name management plan be developed for internetwork interconnections between physical sites.

Experience gathered in the Department of Defense Internet community suggests that the boundary between a larger internet plan and the site naming plan could prove to be a source of problems. Since within the site there is already a modest name management plan adequate for that site, there is a strong temptation to embed exactly that plan throughout the software of the site. When, later, interconnections between sites force addition of a more comprehensive name management strategy, the previous software will inhibit wholesale replacement of name management. More likely is a "compatible" patchwork addition, in which on-site and off-site communication take place differently.

3.2 Token release. In the M.I.T. ring, a message originator captures the token, sends the originating message, and follows that message immediately with a new token. A second station, further along the ring, thus has an opportunity to originate a second message that follows immediately on the heels of the one from the first station with no lost or idle time. In the IBM ring, the originating station waits until the header of the message being originated has traveled all the way around the ring and has been checked for a correct from-address before releasing a new token. If the message is long enough, the header will return around the ring before the trailer leaves the originating station, in which case the originator releases the token immediately following the trailer just as in the M.I.T. ring. But if the message is short enough that the header hasn't been checked by the time the trailer leaves the originating station, the originator transmits idle (zero) bits until the header returns and is checked; then it releases a new token. Thus following short messages on long rings, some ring transmission capacity is not usable. At 4 Mbit/sec, a bit occupies about 100 meters of cable. The IBM repeater design delays the signal by one bit in each repeater. Thus 100 stations each separated by 1000 meters of cable would create a ring of 1100 bits round-trip delay. Thus packets of length greater than about 140 bytes would have no idle delays. On rings of smaller physical size (say with an average of 100 meters of cable between stations) idle time would occur only on packets shorter than 25 bytes. Since the minimum packet size is in the vicinity of 15 bytes, the potential performance loss is negligible unless one anticipates heavily loading a physically large network primarily with packets containing only one or two useful data bytes. That mode of operation would require generation of 20,000 100-bit packets per second, or an average of 200 packets per second at each of 100 stations which is a rather implausible rate of generation of short packets. The conclusion of all this analysis is that there is no significant performance loss with the IBM ring strategy of delaying token release, if necessary, until the message header is verified.

The reason for delayed token release in the IBM ring is to allow the originator, upon inspecting the header of the returned message, to discover that some other station has marked the header with a request for a priority token. If priority is a requirement (see the next section for a discussion of this point) then at the 4 Mbit/sec data rate, the IBM ring protocol is probably preferable. At a higher data rate--say 100 Mbit/sec--the potential performance loss could be appreciable, since in the earlier example any packets of length less than 3000 bytes would lead to insertion of idle time.

3.3 Priority features. In both the IBM ring and the M.I.T. ring, the token-passing protocol acts as a fair round robin scheduler, so the longest time a station could wait before being able to send a message might be the time required for every other station on the ring to send a message first. The IBM ring provides a token-marking protocol that allows for priority. A station that has a requirement for rapid access to the ring can mark the header of the next passing message; when that mark is noticed by the station originating the message, it will mark the token it releases with a "priority" label, which means that non-priority stations should pass this token along even if they have a message queued. The priority token continues to circulate until all priority traffic is cleared; then it is converted to an ordinary token.

Priority was excluded from the M.I.T. ring design primarily because a compelling reason could not be found to add it; on questions such as this the M.I.T. ring design is systematically biased toward the simpler answer. Priority would seem to be needed and useful only in the case that there is controlled quantity of real-time traffic, and a very heavy load of non-priority, deferrable traffic. When the net is loaded to less than, say, 50% of capacity, the average wait for access is perhaps two message times; one must hypothesize a 90% load to find an average wait of ten message times. The M.I.T. ring was designed under the assumption that busy-hour loads would be under 50% of capacity; thus priority seems unnecessary even as a measure to catch unusual, short-term variation in load. The load factor and delay situation is doubly sensitive to data rate. Not only does the average number of items queued for net access go up rapidly as the data rate is reduced, but the length of time required to send each queued message increases also. For example, with a 3 Mbit/sec offered load of 1 Kbyte messages, a 4 Mbit/sec ring will have an average queue of about

4 messages, and each message will occupy the ring for about 2.5 ms, so a 10 msec delay would be expected. The same offered load on a 10 Mbit ring would create an average queue of only 1.5 messages, and each message occupies the ring for only 1 ms, so the expected delay drops to about 1.5 ms.

Since the IBM ring has a lower data rate than the M.I.T. ring and the IBM ring was expressly designed to carry a substantial load of digital voice traffic, which can be delay sensitive, it is not surprising that the priority feature was included.

3.4 <u>Flow control feature</u>. The M.I.T. ring has a low-level protocol feature to aid in flow control: a message trailer bit that can be set by a receiving station to indicate that it recognized a packet but was unable to accept it. This bit appears in the M.I.T. ring design despite emphasis on simplicity, on the basis of experience and an end-to-end argument [18]. The experience is that it is quite easy for one host to generate traffic more rapidly than another host can accept it; the first manifestation of a rate mismatch is inability to accept packets. At the same time, our end-to-end arguments convinced us that there is no other routine requirement for low-level acknowledgements. If acknowledgements are omitted from the low protocol levels, rate mismatch would be discovered only by relatively high protocol levels, at relatively high cost. Thus an explicit negative acknowledgement at a low level for flow control appears to be a sound investment.

The IBM ring provides no low-level feature to aid in flow control. This omission may reflect a dominance of experience with SNA-style protocols, in which flow control (pacing) is negotiated and enforced entirely at higher protocol levels [26].

3.5 <u>Protocol validation vs. protocol debugging</u>. The IBM ring protocols for message handling, initialization, and token recovery have all been validated. Validation in this case means that someone has systematically mapped the protocols into a model of how they should function, and has thus verified that they perform the desired functions without error or endless repetition. All things being equal, one should certainly prefer a validated set of protocols to a set that has not been through any validation process except debugging with live traffic. One could also argue that the more complex the protocol set the more important to attempt validation; priority features are an example of a complexity that increases the interest in validation. At the same time, the intent to validate may lead to subtle pressures on the design, in hope of making validation easier or feasible. Finally, validation only verifies that the design matches the model; one must not expect that it assures that model is correct, useful, or performs efficiently, nor should one expect that it assures that the implementation matches the design. Thus validation of the IBM ring protocols probably has more value as a long-range contribution to research on system correctness. Its immediate impact on a to-be-implemented system is harder to discern. The reassurance that comes from seeing a protocol carry useful data in the field still seems to be important, even if it has been validated.

4. Miscellaneous physical level differences.

4.1 Protocol signals. In both the IBM ring and the M.I.T. ring, the single link from one repeater to the next carries information at several protocol levels. Some method is needed for distinguishing the lowest-level protocol signals from higher level data. (The token is an example of such a signal.) The M.I.T. ring uses a particular bit pattern (7 ones in a row) as a signal, and provides bit stuffing on all higher-level data to insure that the chosen pattern appears on the wire only when a signal is intended. The IBM ring uses a Manchester code violation as a signal.

Using a code violation as a signal has two advantages. First, it avoids both the complexity and bandwidth loss of bit stuffing. Second, a signal occupies only one bit time, which makes loop-back testing easier. (In the M.I.T. ring, each station has a ten-bit delay register to allow a loop-back test with a circulating token.) The primary disadvantage of using code violations for signals is that it sacrifices some of the noise margin and robustness of the system. If one does not attempt to carry information in code violations every Manchester code violation is a result of link noise, and can be taken to be a potential indication of a failing link. In the IBM ring this seror detection feature is lost and instead some data transmission errors generate spurious low-level protocol signals. (Of course higher-level algorithms systematically suppress any such spurious signals.) This narrowing of margins is especially interesting, considering the IBM ring's emphasis on long links, discussed earlier.

4.2 Ground. An apparently minor difference in strategy appears in the approach used in grounding the shields of the twisted pairs used in links from a station to a distribution panel. In the M.I.T. ring, the rule is that the transmitting end of a link ground the shield for that link. In the IBM ring, the rule is that the distribution panel provide ground for all shields. (Both rules prevent ground loops and avoid connecting different station ground references through the network shields.) There appears to be no standard practice on this point, and it is not completely clear what, if any, difference in cross-talk, radiated interference, and susceptibility result. One might expect that a rule calling for an open shield circuit adjacent to a transmitter would carry a greater risk of radiation egress, while an open shield circuit at the receiver would carry greater risk of noise ingress; the IBM ring has open shield circuits at both places, thus taking on both risks simultaneously. In addition, the M.I.T. ring rule caters to the possibility of connecting two stations together in a small ring without going through a distribution panel. In the case of the IBM ring, such connections would require that some explicit step be taken to ground the shields.

4.3 Relay control. The mechanism used to actuate the distribution panel relay in the IBM ring is a current source generated by the station at the other end of the link. Although originally specified to use a current source, the M.I.T. ring implementation actually uses a voltage source. This is probably a mistake in the M.I.T. ring implementation. If a current source is used, the length (and thus the resistance) of the wire in the link is, within limits, not important; also it is simpler to connect two stations into a ring without benefit of a distribution panel, since two current sources can be placed in series. (The M.I.T. ring implementation has a strategically placed diode to allow for this case.)

4.4 PLL freeze. The IBM ring contains a feature that is missing in the M.I.T. ring. To insure rapid resynchronization of the ring when a station enters or leaves the ring, an energy detector in each station reports loss of incoming signal, causing the station to "freeze" its PLL at its current frequency for 2.5 ms. This freeze keeps downstream stations synchronized to this one, while allowing the newly arrived upstream station to get up to speed (or a departed upstream station to get switched completely out.) The hope is that by the end of the 2.5 ms. freeze, the upstream station will have stabilized at the same frequency as before (the frequency of the master clock) and that only a phase adjustment will be needed to close the ring. This is a clever technique, as it avoids propagating a relay disruption all the way around the ring, then repropagating the original frequency around the ring again.

The problem that the PLL freeze solves comes about because loss of upstream signal may lead a PLL to run to the end of its adjustment range searching for synchronization. In the M.I.T. ring this adjustment range is very much smaller (since crystal oscillators are used) so the feature does not appear to be needed. (Although, as mentioned earlier, the M.I.T. ring uses a PLL clamp as part of its phase adjustment procedure.)

Conclusions, implications, and observations.

We have discussed sixteen differences in design details between the IBM experimental token ring and the M.I.T. token ring. On the whole, these differences are at a low level of detail. Seven of the sixteen differences are consequences of two specific different design decisions: choice of maximum link length and choice of ring supervision method. And of the sixteen, only four (cyclic redundancy check, address size, priority feature, and flow control feature) are directly visible to even the lowest level user of the ring. That does not mean that the system designer can ignore the other twelve; they may have significant effects on such issues as installation planning, reliability, availability, and serviceability. But it suggests that most of the differences are indeed at a very detailed level. We should expect that both rings are quite workable in the field, and that fairly extensive field experience will be required to distinguish any of the design decisions as preferable. In contrast, the gross differences between either of the rings and other approaches to local networks (such as the CSMA/CD buses or broadband networks) should become evident with much less experience.

Acknowledgements

The pros and cons of the M.I.T. ring design came from debates with David Clark, Kenneth Pogran, David Reed, Howard Salwen, and Noel Chiappa. Discussions with Karl Kuemmerle, Ernst Rothauser, Hans Mueller, Harry Rudin, Marco Hurtado, Roy Dixon, and Philippe Janson provided insignt into token-ring design philosophy.

References

1. Bux, W., et al., "A Local-Area Communication Network Based on a Reliable Token Ring System," Proc. IFIP TC.6 International In-Depth Symposium on Local Computer Networks, Florence, Italy, April, 1982, pp. 69-82.

2. Muller, H.R., Keller, H., and Meyr, H., "Transmission in a Synchronous Token Ring," Proc. IFIP TC.6 International In-Depth Symposium on Local Computer Networks, Florence, Italy, April, 1982, pp. 125-147.

3. Bux, W., et al., "A Reliable Token-Ring System for Local-Area Communication," Proc. National Telecommunications Conference 1981, New Orleans, Louisiana, November, 1981, pp.

4. Rudin, H., "Validation of a Token-Ring Protocol," Proc. IFIP TC.6 International In-Depth Symposium on Local Computer Networks, Florence, Italy, April, 1982, pp. 373-387.

5. Saltzer, J.H., "Design of a Ten-megabit/sec Token Ring Network," M.I.T. Laboratory for Computer Science Technical Report in preparation.

6. Saltzer, J.H., and Pogran, K.T., "A Star-Shaped Ring Network with High Maintainability," Proc. MITRE-NBS Local Area Communications Network Symposium, Boston, Mass., May, 1979, pp.179-189. Reprinted in Computer Networks 4, (1980), pp. 239-244.

7. Saltzer, J.H., "Communication Ring Initialization Without Central Control," M.I.T. Laboratory for Computer Science Technical Memorandum, TM-202, December, 1981.

8. Saltzer, J.H., Clark, D.D., and Pogran, K.T., "Why A Ring?", Proc. IEEE Seventh Data Communications Symposium, Mexico City, Mexico, October, 1981, pp. 211-217.

9. Wilkes, M.V., and Wheeler, D.J., "The Cambridge Digital Communication Ring," Proc. MITRE-NBS Local Area Communications Network Symposium, Boston, Mass., May, 1979, pp. 47-61.

10. Okuda, N., Kunikyo, T., and Kaji, T., "Ring Century Bus--an Experimental High Speed Channel for Computer Communications," Proc. Fourth International Conference on Computer Communications, September, 1978, pp. 161-166.

11. "IBM Series/1 Local Communications Controller Feature Description," IBM Publication GA34-0142, 3rd edition, April, 1982.

12. Gordon, R.L., et al., "Ringnet: A packet switched local network with decentralized control," 4th Conf. on Local Networks, Minneapolis, MN, October, 1979, pp. 13-19.

13. Apollo
(no reference found yet)

14. Dixon, R.C., "Ring Network Topology for Local Data Communications," Proc. IEEE COMPCON82, Washington, D.C., September, 1982, pp. 591-605.

15. Markov, J.D., and Strole, N.D., "Token-Ring Local Area Networks: A Perspective," Proc. IEEE COMPCON82, Washington D.C., September, 1982, pp. 606-614.

16. Andrews, D.W., and Schultz, G.D., "A Token-Ring Architecture for Local-Area Networks," Proc. IEEE COMPCON82, Washington, D.C., September, 1982, pp. 615-624.

17. Shoch, J.F., and Hupp, J.A., "Performance of an Ethernet Local Network--A Preliminary Report," Proc. IEEE COMPCON80, February, 1980, pp. 318-322.

18. Saltzer, J.H., Reed, D.P., and Clark, D.D., "End-to-End Arguments in System Design," Proc. 2nd International Conference on Distributed Computing Systems, Paris, France, April, 1981, pp. 509-512.

19. "Internet Protocol," Information Sciences Institute, RFC-791, September, 1981, reprinted in Internet Protocol Transition Workbook, SRI International, March, 1982, pp. 49-100.

20. "Transmission Control Protocol," Information Sciences Institute, RFC-793, September, 1981, reprinted in Internet Protocol Transition Workbook, SRI International, March, 1982, pp. 131-221.

21. "Internet Transport Protocols," Xerox System Integration Standard, XSIS 028112, Xerox Corporation, December, 1981.

22. Recommendation X.25/Interface Between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks, CCITT Orange Book 7, International Telephone and Telegraph Consultative Committee, Geneva, May, 1978.

23. Proposal for Provisional Recommendation X.75 on International Interworking Between Packet Switched Data Networks, CCITT Study Group VII Contribution No. 207, International Telephone and Telegraph Consultative Committee, Geneva, May, 1978.

24. Rothauser, E.H., Janson, P.A., and Muller, H.R., "Meshed-Star Networks for Local Communication Systems," Proc. IFIP Working Group 6.4 International Workshop on Local Networks, Zurich, Switzerland, August, 1980, pp. 25-41.

25. Dalal, Y., and Printis, R.S., "48-bit Absolute Internet Host Numbers," Proc. IEEE Seventh Data Communications Symposium, Mexico City, Mexico, October, 1981, pp. 240-245.

26. Cypser, R.J., Communications Architecture for Distributed Systems, Addison Wesley, Reading, MA, 1978, pp. 299f.