Media Access Protocols For A Scalable Optical Interconnection Network

by Thomas S. Jones and Ahmed Louri
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721
email: louri@ece.arizona.edu

Abstract

Hierarchical optical interconnection networks have the potential of solving the communication bottleneck that has evolved in parallel processing systems due big increases in processor speeds. Hierarchical Optical Ring Interconnection Network (HORN) [12] is one network architecture that was proposed to provide scalability to a larger number of processing nodes (PNs) with low latency and high bandwidth. In HORN PNs are arranged in rings and those rings are grouped together hierarchically in higher level rings. While collisions of data from multiple sources in hierarchical networks like HORN are reduced by separating nodes spatially and by wavelength, they can't be prevented completely and a media access (MAC) protocols must be used to that end. An in depth analysis of five collision-free, single hop protocols is performed in terms of average delay and system throughput. The protocols analyzed are: time division multiple access (TDMA) [13, 14], TDMA with arbitration [12, 13, 14], FatMAC [6], DMON [15], and token hierarchical optical ring network (THORN). The first four protocols are documented in the literature but the fifth, THORN, was developed expressly for HORN. While all the protocols support the scalability objectives of HORN, THORN is shown to have the lowest delay and a throughput comparable to the other four protocols.

1. Introduction

The application of optical fibers to interconnection networks (INs) in parallel processing systems offers the potential to transmit on many wavelengths simultaneously thereby multiplying the number of available data paths. While current optical fiber technology permits simultaneous insertion onto a single fiber of about 20 wavelengths [1, 4], arranging networks hierarchically further increases the number of channels that can operate simultaneously by reusing these same 20 wavelengths over and over in portions of a network that are separated spatially [3, 16]. This approach is also supported by conclusions by Bell [2], Dandamudi [5] and Goodman [7] that PNs engage in data transfers more frequently with nearby neighbors than with more distant ones.

HORN is a hierarchical optical IN we presented previously [12], that uses a ring of rings topology. Processing nodes (PNs) are connected by optical fibers in rings of up to 20 PNs and these first level rings are interconnected at routing nodes to form rings of local rings. This process is repeated recursively to form higher levels in the hierarchy as many times as necessary to meet the wavelength limitation for the desired number of PNs. A simple example of HORN is shown in Figure 1 in which 234 PNs are connected in a three layer hierarchy. The main objective is to obtain a network that is scalable with low delay and high throughput for data transmissions.

Wavelength division multiple access (WDMA) is also used in HORN to separate nodes or subrings on the same ring by wavelength. At the first level each node is assigned a unique wavelength for reception; a source selects a destination by transmitting on the wavelength assigned to the destination. At the higher levels a wavelength is uniquely assigned to each *ring* of PNs, or ring of rings of PNs, depending on the level at which communication occurs. At these higher levels special routing nodes are used to route messages optically.

Although WDMA provides multiple data paths, thereby reducing communication contentions, they can't be prevented completely in HORN without some method to regulate access to each channel. A MAC protocol is necessary to prevent two sources from attempting to transmit to the same destination simultaneously.

Two classes of MAC protocols were deemed unacceptable for HORN. Most MAC protocols in use today are multi-hop in nature since some processing of each packet must be performed at each intermediate node to determine where the packet should be routed next [9]. HORN, by its very nature, is a single hop architecture; messages are

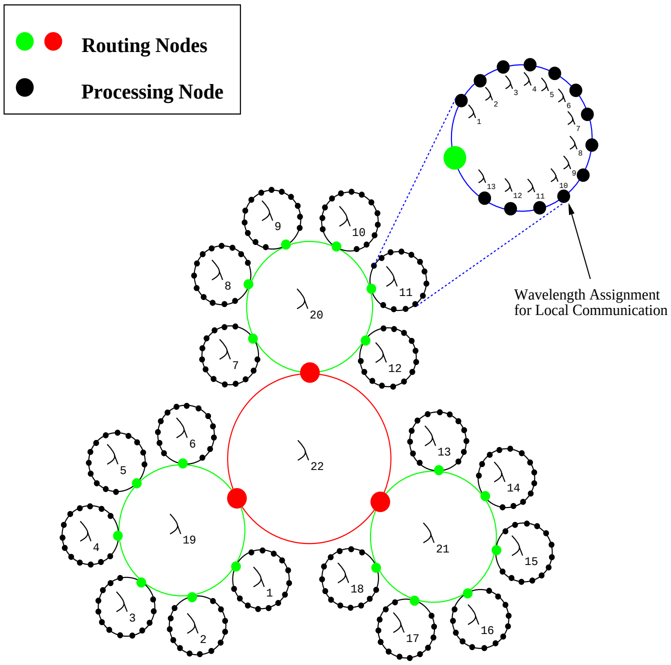

Figure 1. Sample HORN interconnection network showing wavelength assignment.

sent from source to destination without any intermediate electronic processing. Therefore multi-hop MAC protocols are unacceptable for HORN. Many protocols today also allow collisions to occur, requiring some collision detection mechanism and retransmission when collisions occur. Since we are attempting to maximize throughput, i. e. maximize the number of successful data transmissions, collision based MAC protocols are also unacceptable in HORN.

Four collision-free, single hop protocols presented in the literature [6, 12, 13, 14, 15] were selected for in depth analysis in terms of delay, throughput and node complexity. They are time division multiple access (TDMA), TDMA with arbitration, FatMAC and DMON. An additional protocol, called Token Hierarchical Optical Ring Network (THORN), is presented for the first time in this paper and is also analyzed against the same parameters.

2. Protocol Descriptions

2.1. TDMA

Under TDMA each cycle is divided into time slots and each node is assigned a slot, in turn, during which it has exclusive access to a channel for transmission. This process is repeated for every channel at all levels of the hierarchy in HORN. The length of a TDMA cycle is therefore determined by the length of each slot and by the number of slots needed, i.e. the number of nodes that need access to the channel, which varies under HORN according to the level in the hierarchy. If there are N_i nodes at level i in the hierarchy then cycle length is $N_i \times T_D$ where T_D is the data packet length in seconds of transmission time. The length of a time slot is a design issue and is primarily determined by the average message length. Figure 2 shows a typical channel and slot assignment for HORN using TDMA.

	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4
λ_1	P_1	P_4	P_3	P_2
λ_2	P_2	P_1	P_4	P_3
λ_3	P_3	P_2	P_1	P_4
λ_4	P_4	P_3	P_2	P_1

Time Scale

Figure 2. Typical TDMA slot assignment for a structure with 4 nodes.

2.2. TDMA with Arbitration

TDMA with arbitration is a variation on TDMA in which there are fewer time slots assigned to a level than there are nodes contending for access, requiring reservation or arbitration of the available slots. This has the potential advantage of greatly reducing the number of empty data slots when compared with pure TDMA.

2.3. FatMAC

FatMAC [6] is also a reservation or arbitration protocol but it is more sophisticated than TDMA/arbitration. Data slot reservation is broadcast optically in the first time slot, called the control slot, on each data channel. FatMAC uses a different approach than TDMA or TDMA/arbitration in assigning nodes and channels, channel assignment is *dynamic* with respect to destination nodes and sources reserve the first available slot on any channel in use at the level of transmission. This means that one slot is reserved on every channel before a second slot is reserved on any channel and that cycle lengths can vary by no more than one packet length between channels on the same ring.

2.4. DMON

DMON is a token based protocol presented by Pinkston [15] and is probably the most complicated protocol to be assessed here in terms of the hardware required at each node but it's a relatively simple protocol algorithmically. Any node may transmit on a data channel once it has reserved access on the control channel, but access to the control channel is controlled by a token on a dedicated channel. The same token also controls access to a dedicated broadcast channel, so a node may broadcast or transmit a reservation request each time it acquires the token, but not both. The procedure a node follows to transmit a data or broadcast packet is as follows: 1) node acquires token, 2) node transmits slot reservation on the control channel, 3) node releases token, and 4) node transmits data on the reserved channel (see Figure 3).

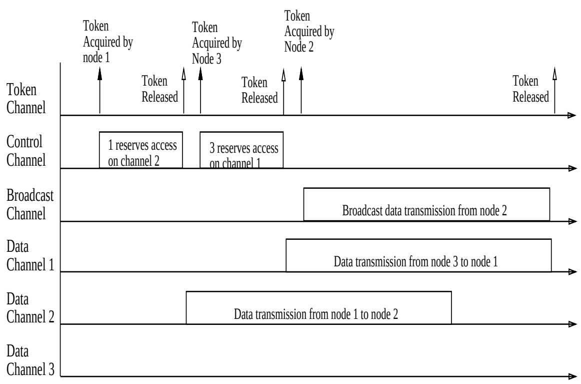

Figure 3. Timing diagram for DMON, only three data channels shown.

2.5. THORN

In this paper we propose a new protocol called Token Hierarchical Optical Ring Network (THORN) that is a variation of the token ring and DMON protocols. There is one dedicated token channel for each level of the hierarchy which is shared by all nodes at that level and the tokens for each of the data channels circulate on this channel as a packet. In order to allow a node to hold a channel (token) for multiple cycles, the token packet has a busy field and a request field. A token is acquired when a node discovers that the appropriate busy bit is clear and sets it. Once a node acquires a token it can hold the token and the corresponding channel until another node requests them by setting the appropriate request bit. The procedure a node follows to transmit a data packet is as follows: 1) node acquires the necessary token on the token channel, if the token can't be acquired immediately it is requested by setting the coresponding request bit, 2) Node transmits on the channel for which the token was acquired, and 3) node releases the token. Figure 4 shows a sample timing diagram for THORN.

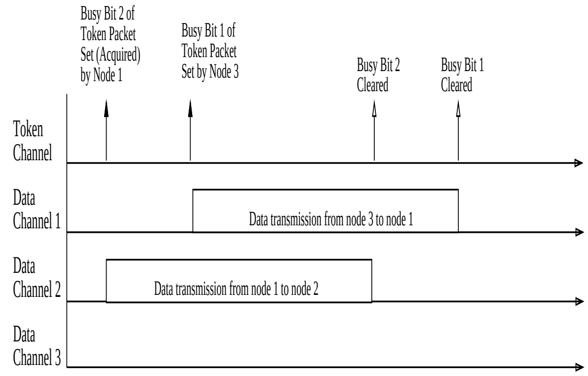

Figure 4. Sample timing diagram for THORN. Request bit usage is not shown.

3. Average Delay

3.1. TDMA

Average delay is defined as the expected delay in transmitting a message from any source node to any destination node, and includes data packet transmission time, time packet must wait in the output queue, and end-to-end transmission time of the network. For TDMA it will also include the time a node must wait for its dedicated slot to materialize on the channel. Spragins [17] gives an expression for average delay in a TDMA system:

$$D = \frac{X}{R} + \frac{NX}{2R} + \frac{NX}{2R}\frac{\rho}{1-\rho} + \tau \qquad (1)$$

Where X is the packet length in bits, R is the transmission rate in bits per second, τ is the end-to-end transmission time for the system, and ρ is the ratio of the average arrival rate, γ, to the average service rate, μ. The quantity $\frac{X}{R}$ is the length of time it takes to transmit a data packet of X bits at R bits per second. T_D is used for this quantity hereafter.

It can be shown for realistic numbers in HORN that τ is much less than T_D, so it's dropped from this Equation (1). In a network of 1000 contiguous nodes spaced 1 meter apart τ is on the order of $5\mu sec$ while even a small data packet of 100 kbits transmitted at 100 Mbits/sec has a transmission time (T_D) of 1 msec.

N is the number of slots in the cycle and for TDMA is the same as the number of nodes. In a hierarchical system the value to use for N will vary because communication is taking place simultaneously in many locations that require different numbers of slots, based on the level at which communication takes place, and we need to account for that in Equation (1). If we define communication locality, ℓ, as the number of data packets sent at level i versus the total number of data packets to be sent at higher levels in the hierarchy then the effective value for N for a system with r levels is:

$$\begin{aligned} N_{eff} &\equiv \text{Effective (average) number of TDMA slots} \\ &= \ell \sum_{i=1}^{r-1} n^i (1-\ell)^{i-1} + n^r (1-\ell)^{r-1} \end{aligned} \quad (2)$$

With this we can calculate the average expected delay for TDMA in HORN:

$$D^{<TDMA>} = T_D \left(1 + \frac{N_{eff}}{2} + \frac{\rho N_{eff}}{2(1-\rho)}\right) \quad (3)$$

The first term in Equation (3) represents the length of time it takes to transmit to data packet, the second term is the average delay due to waiting for the assigned time slot to materialize, and the last term is the queuing delay.

3.2. TDMA With Arbitration

Average access delay for TDMA/arbitration will be the same as for TDMA except that the effective number of nodes will be much lower since there are fewer slots than nodes competing for them and there will be a term to account for the delay due to arbitration:

$$D^{<TDMA/arb>} = T_D + T_{queue} + \frac{1}{2} T_{channel} \quad (4)$$

The first two terms appear in the delay equation for pure TDMA. The last term, $\frac{1}{2}T_{channel}$, includes both the delay due to waiting for the assigned slot to materialize and the delay due to arbitration for the channel. $T_{channel}$ can then be expressed as:

$$T_{channel} = NT_D + Nt^* \quad (5)$$

As stated previously, the number of slots, N, will be much less than for pure TDMA. If there is one slot for every k_2 nodes then N becomes $\frac{N_{eff}}{k_2}$. The second term, Nt^*, is the arbitration time needed per node in a structure multiplied by the number of nodes to reflect the fact that as more nodes are added to a ring, more arbitration time will be needed; This will be a direct function of the number of nodes in the structure versus the number of slots available, but is also affected by other parameters such as processor speed. If we equate T_D to t^* by a constant, k_1, we obtain:

$$T_{channel} = NT_D(1 + k_1) \quad (6)$$

Combining Equations (4) and (5) and substituting $\frac{N_{eff}}{k_2}$ in for N as noted above yields:

$$\begin{aligned} D^{<TDMA/arb>} &= T_D \left(1 + \frac{N_{eff}\rho}{2k_2(1-\rho)}\right) \\ &\quad + T_D \left(\frac{N_{eff}(1-k_1)}{2k_2}\right) \end{aligned} \quad (7)$$

Note that this reverts to Equation (3), the average delay in a pure TDMA system, when there is exactly one slot per cycle for every node in the structure ($k_2 = 1$) and no arbitration ($k_1 = 0$).

3.3. FatMAC

Average delay for FatMAC is the sum of data packet transmission time, queue wait time and average time until the next control slot, since a data packet may arrive at the output buffer at any time in the cycle:

$$\begin{aligned} D^{<FatMAC>} &= T_D + \left(\frac{T_C + CT_D}{2}\right) \\ &\quad + \left(\frac{T_C + CT_D}{2} \frac{\rho}{1-\rho}\right) \end{aligned} \quad (8)$$

Where C is average cycle length in number of packets and is determined by the average arrival rate, the number of channels available and the number of nodes with messages to send. C can therefore be rather difficult to calculate since it changes dynamically under FatMAC. C is a ceiling function in that cycle length will increase by one for every Λ_i messages to be transmitted in a cycle, where Λ_i is the number of channels available to a structure at level i. If Λ_i is held constant at some value, Λ_0, for every layer and every structure at each layer then C is given by:

$$C = \left\lceil \frac{\gamma N_{eff}}{\Lambda_0} \right\rceil \quad (9)$$

The length of the control packet, T_C, is determined by the design and can be referenced to the data packet length:

$$T_D = LT_C \quad (10)$$

Substituting this into Equation (8) gives an expression for the average expected delay under FatMAC in terms of the data packet length:

$$D^{<FatMAC>} = T_D \left(1 + \frac{(1 + CL)}{2L(1-\rho)}\right) \quad (11)$$

3.4. DMON

An equation for average delay for a token based or polled protocol is given by Spragins [17]. The delay to acquire the token and transmit a control or broadcast packet is given by:

$$D_c = \frac{Y}{R} + \overline{\tau} + \frac{t'(1 - \frac{\rho}{M})}{2(1-\rho)} + \frac{\rho}{2(1-\rho)} \frac{Y}{R} \quad (12)$$

Where $\overline{\tau}$ is the average node to node transmission delay, M is the number of nodes and is equal to N_{eff} for HORN, Y is the length of the control packet in bits, and t' is the ring delay, that is the delay due to handling the token or polling all the nodes. The quantity $\frac{Y}{R}$ in this case is the length of the control packet in seconds, to which we previously assigned the variable T_C. We can again relate T_C and T_D by Equation (10). As assumed previously, $\overline{\tau}$ is usually much smaller than the other terms and can be ignored. The ring delay as given by Spragins is:

$$t' = \overline{\tau} N_{eff} + \frac{B}{R} N_{eff} \tag{13}$$

Where B is the bit delay per node and R is the transmission rate in bits per second. The first term here $\overline{\tau} N_{eff}$ here is dropped and $\frac{B}{R}$ can be related to T_D by a constant since the bit delay in a network should be relatively constant and much less than T_D to keep overhead low:

$$t' = N_{eff} k T_D \tag{14}$$

The constant k should be small, substantially less than one since it will be a design issue to keep the ring delay much lower than the data packet transmission time to minimize overhead. The effective number of system nodes, N_{eff}, is retained in this equation to reflect that fact that the ring delay will be a function of the number of system nodes and how those nodes are arranged.

However, Equation (12) only accounts for the delay to send the control packet or to broadcast a message. For point to point transmissions there will be additional term to account for the delay to send the data packet comprised of the queuing delay and the data packet transmission time. Note that the delay due to waiting for a time slot to materialize is absent since this is not time slotted protocol; once a node has captured the token it has exclusive access to the channel. The queuing delay in this case will be for a system with arbitrary or random arrival time and fixed service times [8, 10, 11]. This queuing delay and data transmission delay is given by:

$$\begin{aligned} D_d &= \frac{\rho^2}{2\gamma(1-\rho)} + T_D \\ &= \left[\frac{\rho}{2(1-\rho)} + 1\right] T_D \end{aligned} \tag{15}$$

So for HORN, the average delay for DMON is given by:

$$\begin{aligned} D^{<DMON>} &= D_c + D_d \\ &= \frac{T_D(2-\rho)}{2(1-\rho)}\left(1 + \frac{1}{L}\right) \\ &\quad + \frac{T_D k}{2(1-\rho)}\left(N_{eff} - \rho\right) \end{aligned} \tag{16}$$

3.5. THORN

The expression given by Spragins (Equation (12) above) also applies here but the token is referenced to the data channel rather than the control channel:

$$\begin{aligned} D^{<THORN>} = \frac{X}{R} + \overline{\tau} + \frac{t'(1-\frac{\rho}{M})}{2(1-\rho)} \\ + \frac{\rho}{2(1-\rho)}\frac{X}{R} \end{aligned} \tag{17}$$

The variable X has replaced the Y in the previous form of this equation because it now represents the length of the *data* packet in bits. Consequently $\frac{X}{R}$ represents the data packet transmission time, which we previously defined as T_D. As with DMON, $\overline{\tau}$ is dropped because it is much smaller than the other terms and the ring delay is given by:

$$t' = N_{eff} k T_D \tag{18}$$

Communication locality still determines the effective number of nodes participating at a given level so the effective number of nodes in the system with data to transmit is given by Equation (2). We can therefore substitute N_{eff} into Equation (17) for M. Making these substitutions and equating t' to T_D by Equation (14) yields:

$$\begin{aligned} D^{<THORN>} &= \frac{T_D(2-\rho)}{2(1-\rho)} \\ &\quad + \frac{T_D k}{2(1-\rho)}(N_{eff} - \rho) \end{aligned} \tag{19}$$

3.6. Summary of Average Delay Analysis

The average delay for the five protocols analyzed is graphed in Figure 5 as a function of the offered load and in Figure 6 as a function of the number of nodes for a system with data packet length of 1 millisecond. In Figure 5 the number of nodes is 1000; in Figure 6 the offered load is 0.5. Delays for THORN and DMON are nearly identical and are about ten times better than the nearest competitor, FatMAC. In contrast, the delay for TDMA is about 100 times longer than for THORN or DMON.

4. System Throughput

System throughput is generally accepted as a valuable metric for interconnection networks, yet few authors attempt to quantify it since it's affected by many factors that are difficult to fix except in very specific systems [3]. It is therefore with no small amount of trepidation that we attempt to assess these protocols in terms of system throughput. We are considering specific protocols on a specific type

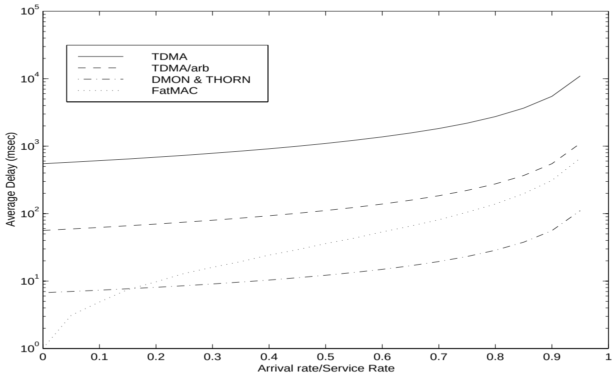

Figure 5. Delay as a function of the offered load for a hierarchy with three layers and a 1 msec data packet length (T_D).

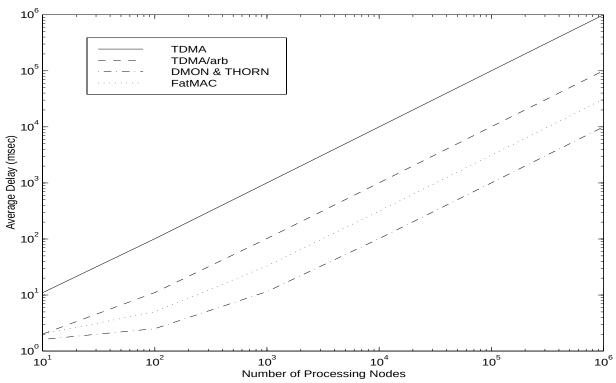

Figure 6. Delay as a function of the number of nodes for a hierarchy with three layers and a 1 msec data packet length (T_D).

of network and therefore can fix enough parameters to make the analysis valid and meaningful.

Using Equation (2) for the effective number of nodes in our system gives an expected throughput rate per channel:

$$S = \gamma N_{eff} \tag{20}$$

For each of the protocols considered the average service rate is the inverse of the cycle length as only one message can be transmitted by each node in a cycle. Therefore we can quantify the system throughput in a way that is meaningful and will allow comparisons between the protocols by fixing the cycle lengths.

4.1. TDMA

The cycle length for TDMA is a simple term, each channel has n_i slots of length T_D in each cycle. We can again apply Equation (2) to give the effective number of slots across the entire system. Throughput for each channel under TDMA can therefore be expressed as:

$$\begin{aligned} S^{<TDMA>} &= \gamma N_{eff} \\ &= \rho\mu N_{eff} \\ &= \frac{\rho N_{eff}}{N_{eff} T_D} \\ &= \frac{\rho}{T_D} \end{aligned} \tag{21}$$

This result says that the maximum channel throughput for TDMA is 1 packet for every period of time equal to the packet length in seconds, a result which should be apparent at least intuitively.

Maximum system throughput will be the maximum value possible for Equation (21) multiplied by the effective number of channels in the system. Channel throughput is maximum when ρ is its maximum value of one. Like N_{eff}, the effective number of channels in a system (Λ_{eff}) is determined by the communication locality of the system but is also determined by the number of channels available at each level. In HORN there is one channel assigned to each subring on a ring at any given level such that there are n channels at the highest level, n^2 channels at the next level down, n^3 at the third level down, and in general there are n^{r+1-i} channels at level i where $i = 1$ for the local level and $i = r$ for the highest level in a hierarchy of r levels. The channels at level i are used some percentage of the time based on ℓ, the communication locality. If communication were 100% local ($\ell = 1$) then Λ_{eff} would be equal to n^r because all communication would take place on the local rings. Conversely, if communication were 0% local, that is all communication occurring at the highest level of the hierarchy, then Λ_{eff} would be equal to n. Stated formally:

$$\Lambda_{eff} = \ell \sum_{i-2}^{r} n^i (1-\ell)^{r=i} + n(1-\ell)^{r-1} \tag{22}$$

System throughput for HORN under TDMA is therefore given by:

$$S_T^{<TDMA>} = \frac{\rho}{T_D} \Lambda_{eff} \tag{23}$$

4.2. TDMA with Arbitration

System throughput for TDMA/arbitration is also similar to pure TDMA with the added overhead due to arbitration but a reduced cycle length. Cycle length for TDMA/arbitration is $\frac{N_{eff}}{k_2}(1+k_1)T_D$ which comes directly from the average delay equation. The constant k_2 is the ratio of the number of slots in pure TDMA to the number of slots in TDMA/arbitration and the constant k_1 is the ratio of the arbitration time per node requesting access to the data packet length in seconds. The throughput per channel under TDMA/arbitration is therefore given by:

$$\begin{aligned} S^{<TDMA/arb>} &= \frac{N_{eff}}{k_2}\gamma \\ &= \frac{\frac{N_{eff}}{k_2}\rho}{\frac{N_{eff}}{k_2}T_D(1+k_1)} \\ &= \frac{\rho}{T_D(1+k_1)} \end{aligned} \tag{24}$$

The number of channels and the communication locality are unchanged for TDMA/arbitration so the system throughput is given by Equation (24) multiplied by Λ_{eff}:

$$S_T^{<TDMA/arb>} = \frac{\rho\Lambda_{eff}}{T_D(1+k_1)} \tag{25}$$

4.3. FatMAC

Channel throughput for FatMAC will also be a function of C as derived in Equation (9). Cycle length under FatMAC is one control packet plus C data packets in length. Average channel throughput under FatMAC will therefore be:

$$\begin{aligned} S^{<FatMAC>} &= \frac{\rho N_{eff}}{T_C + CT_D} \\ &= \frac{\rho N_{eff}}{(\frac{1}{L}+C)T_D} \end{aligned} \tag{26}$$

Although the effective number of channels under FatMAC is the same as under the other protocols considered so far, FatMAC doesn't use them the same way; channels at a given level aren't used independently of one another. The number of channels operating simultaneously in the system is reduced by Λ_0 over the other protocols as FatMAC fills the first slot in each channel before it lengthens the cycle. For this reason the throughput per channel is multiplied by Λ_{eff} and divided by Λ_0 to obtain the system throughput:

$$S_T^{<FatMAC>} = \frac{\Lambda_{eff}}{\Lambda_0}\frac{\rho N_{eff}}{(\frac{1}{L}+C)T_D} \tag{27}$$

4.4. DMON

Unlike the other protocols considered so far, DMON requires a control packet transmission for every data packet transmission which in turn requires acquiring the token for every data packet transmission. The cycle length therefore is $T_C + T_D + t'$, where t' is given by (14). When the average number of nodes transmitting is M then the channel throughput is given by:

$$\begin{aligned} S^{<DMON>} &= \frac{M\rho}{M(T_C+T_D+t')} \\ &= \frac{\rho}{(\frac{1}{L}+1+kN_{eff})T_D} \end{aligned} \tag{28}$$

This yields a system throughput given by:

$$S_T^{<DMON>} = \frac{\rho\Lambda_{eff}}{(\frac{1}{L}+1+k)T_D} \tag{29}$$

4.5. THORN

As with the other protocols the channel throughput for THORN is equal to the average arrival rate per node multiplied by the effective number of nodes. Cycle lengths and service rates vary under THORN. When a node must acquire the token before transmitting the cycle length is T_D+t'; but when streaming can occur, that is when no other node needs the channel, then the cycle length goes down to just T_D. Under heavier loads the former cycle length will predominate and the channel throughput when an average of M nodes are using the channel will be given by:

$$\begin{aligned} S^{<THORN>} &= \frac{M\rho}{M(T_D+t')} \\ &= \frac{\rho}{(1+kN_{eff})T_D} \end{aligned} \tag{30}$$

This yields a system throughput given by:

$$S_T^{<THORN>} = \frac{\rho\Lambda_{eff}}{(1+kN_{eff})T_D} \tag{31}$$

4.6. Summary of System Throughput

System throughput for the five protocols is graphed in Figure 7 as a function of the offered load and in Figure 8 as a function of the number of system nodes for a three level system with a data packet length of 1 millisecond. Locality (ℓ) for both figures is 0.5 for each level. Throughput for FatMAC is very nearly constant for all loads since the cycle length will vary with the offered load and there are very few empty slots. System throughput for DMON and THORN were expected to be much higher, but these results indicate that the ring delay has a significant impact on throughput and consequently must be held much lower than expected.

5. Conclusions

TDMA, TDMA/arbitration, FatMAC, DMON and THORN were analyzed in terms of delay and system

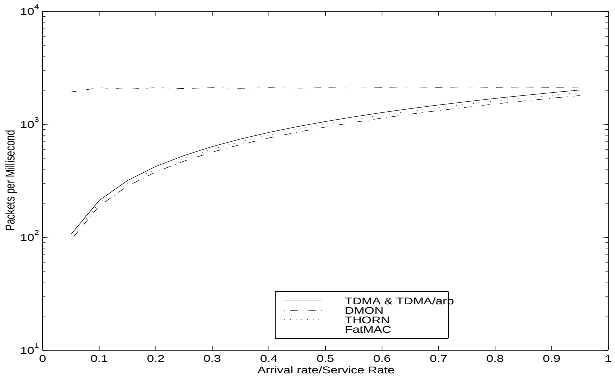

Figure 7. System throughput as a function of the offered load for a three level hierarchy.

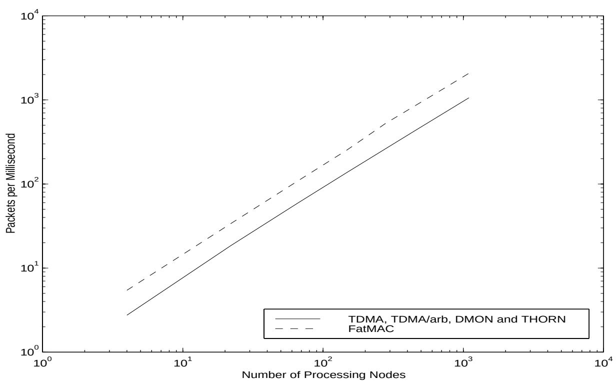

Figure 8. System throughput as a function of the number of nodes for a hierarchy with three layers.

throughput. Clearly THORN and DMON are the preferred protocols with respect to average delay, outperforming FatMAC by a factor of about ten at loads above about 0.5 and outperforming the TDMA based protocols by a factor of about 100 at all offered loads. In terms of throughput, most of the protocols performed about the same but FatMAC was shown to be the best at low offered loads due to the variable cycle length. For loads above about 0.5 this advantage was reduced to only about twice the system throughput of the other protocols.

HORN is scalable under any of the protocols analyzed as can be seen from Figures 6 and 8. Delay and throughput both increase linearly or very nearly linearly for all protocols as the number of PNs is increased. THORN was selected for use in HORN based on the above considerations.

References

[1] M. Adda. A Scalable Multibus Configuration for Connecting Transputer Links. *IEEE Transactions on Parallel and Distributed Systems*, 8:245 – 263, March 1997.

[2] G. Bell. Ultracomputers: A Teraflop Before Its Time. *Communication of the ACM*, 35:27 – 47, August 1992.

[3] C. Brackett. Dense Wavlength Division Multiplexing Networks: Principles and Applications. *IEEE Journal on Selected Areas in Communications*, 8:948 – 964, August 1990.

[4] L. A. Buckman. *Applications In Optical Communication: Optical Transmission of Millimeter Wave Signals; and, An All-Optical Wavelength Routed Switching Network*. PhD Dissertation, University of California, Berkeley, 1996.

[5] S. Dandamudi and D. Eager. Hierarchical Interconnection Networks for Multicomputer Systems. *IEEE Transactions on Computers*, 39:786 – 797, June 1990.

[6] P. Dowd, K. Bogineni, K. A. Aly, and J. A. Perreult. Hierarchical Scalable Photonic Architectures For High Performance Processor Interconnection. *IEEE Transactions on Computers*, 42:1105 – 1120, September 1993.

[7] J. W. Goodman, F. Leonberger, S.-Y. Kung, and R. A. Athale. Optical Interconnections for VLSI Systems. *Proceedings of the IEEE*, 72:850 – 866, July 1984.

[8] L. Gorney. *Queuing Theory: A Problem Solving Approach*. Petrocell Books, New York, 1982.

[9] K. Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programability*. McGraw-Hill Book Company, New York, 1993.

[10] L. Kleinrock. *Communication Nets: Stochastic Message Flow and Delay*. McGraw-Hill Book Company, New York, 1964.

[11] G. Louchard and G. Latouche. *Probability Theory and Computer Science*. Academic Press, New York, 1983.

[12] A. Louri and R. Gupta. Hierarchical Optical Interconnection Network (HORN): Scalable Interconnection Network for Multiprocessors and Multicomputers. *Applied Optics*, 36:430 – 442, January 1997.

[13] N. Maxenchuck. Twelve Random Access Strategies for Fiber Optic Networks. *IEEE Transactions on Computers*, 36:942 – 995, August 1988.

[14] B. Mukherjee. Architectures and Protocols for WDM-based Local Lightwave Networks, Part I: Single Hop Systems. *IEEE Network*, pages 12 – 27, May 1992.

[15] T. Pinkston and C. Kuznia. Smart Pixel-Based Network Interface Chip. *Applied Optics*, 36:4871 – 4880, 10 July 1997.

[16] K. Sivalingam and P. Dowd. A Multilevel WDM Access Protocol For An Optically Interconnected Multiprocessor System. *IEEE Journal of Lightwave Technology*, 13:2152 – 2167, November 1995.

[17] J. Spragins, J. Hammond, and D. Powlikowski. *Telecommunications: Protocols and Design*. Addison-Wesley Publishing Co., Reading, MA, 1991.