

Efficient Communication Operations On Passive Optical Star Networks

F. Desprez ^{*†}, A. Ferreira[‡]

CNRS - Laboratoire LIP, URA 1398
ENS Lyon
46, Allée d'Italie
69364 LYON Cedex 07, France

B. Tourancheau ^{§¶}

Computer Science Department
The University of Tennessee
107 Ayres Hall
Knoxville, TN 37996-1301, USA.

Abstract

In this paper, we show how to use the Wavelength Division Multiple Access capabilities of Passive Optical Star Networks for efficiently implementing communication operations that are widely used in parallel applications. We propose algorithms for multiple broadcasting, scattering, gossiping and multi-scattering, which are very close to the lower bound for these problems.

1 Introduction

For massively parallel architectures, the hardware complexity of the interconnection network is much higher than that of the processing units, employing most of the hardware involved [17]. In many optical communication networks, Passive Optical Star technology using wavelength division multiplexing access (WDMA) offers optical multiple access channels that allow significant reduction in the complexity of the specialized routers connected to the processors, along

with a substantial reduction in the intermediary latencies for one-to-many communications [1, 20].

The WDMA technology allows us to share the enormous bandwidth of optical fibers among several messages, through the creation of multiple channels by the use of different wavelengths. This can be implemented by tunable feedback laser diodes for the transmitters, and wavelength tunable filters for the receivers [24]. In this paper, we use the processors capability of freely tuning the wavelength they want to access at a given moment, in order to design efficient global communication schemas for Passive Optical Star Networks. Such schemas are commonly used in all areas of parallel computing, ranging from neural networks to linear algebra, and are as follows.

- One-to-all multiple broadcasting: broadcasting multiple messages from one node to all other nodes.
- One-to-all personalized broadcasting (scattering): one node sends a distinct message to all other nodes.
- All-to-all communication: broadcasting from each node to every other nodes.
- All-to-all personalized communication: each node sends a distinct message to every other node.

We show how to implement the first operation in such a way that there is a machine-dependable tradeoff between the system parameters for multiple communication capabilities (k) and the tuning time (D). For all the others, our algorithms come very close to the lower bound for communication costs, while keeping the tuning time as low as possible.

Our paper is organized as follows. In the next section a brief overview of the used optical model is given.

^{*}Part of this work was done during a postdoc period at the C.S. dept. of the University of Tennessee, Knoxville, TN 37996-1301, USA.

[†]Partially supported by MRE grant No. 974, the CNRS-NSF grant No. 950.22/07 and the research program C3 of the French CNRS.

[‡]Partially supported by ANM and C3, and by the French-Israeli project on Optical Models for Parallel Computing.

[§]On leave from LIP, CNRS URA 1398, ENS Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France.

[¶]This work was supported in part by the National Science Foundation under grant ASC-8715728, the National Science Foundation Science and Technology Center Cooperative Agreement CCR-8809615, the DARPA and ARO under contract DAAL03-91-C-0047, PRC C³, CNRS-NSF grant 950.223/07, Archipel SA and MRE under grant 974, and DRET.

The general algorithm for the tuning communication pattern is introduced in section 2, and all the new algorithms are presented and discussed in section 3. We close the paper with a cross-analysis of all the algorithms and give ways for further research.

We assume that we have P processors, labeled from 0 to $P-1$, connected by a passive optical star-coupled structure.

The model consists of the P processors transmitting and receiving atomic messages over a passive medium. The cost of transmission of an atomic message is assumed to be τ as in classical communication models [10, 21, 22]. The processors can communicate through any wavelength from the set of wavelengths available for tuning, with the only constraint that for a given wavelength, at most one transmitter can use it at a given step. The tuning time for each new wavelength is denoted D . The life-span of an optical message is the number of receivers that can access the message before the message fades away because of the loss of energy caused by the receivers access. We suppose that the system allows each processor to communicate through k different wavelengths at a time step, where k is then the life-span of each optical message in the system.

Such a model is closely related to the reconfiguration network model described in [6, 11, 12] or the complete k -ports network model described in [7, 8, 9] for the classical switching networks (such as crossbar, multistage or hardware routing devices) of multi-computers.

Hence, with such a network, a processor can send and receive k messages of size L from different processors for a cost $D + L\tau$. In the following, we assume the transmission cost to be a unit of time, i.e. $L\tau = 1$.

In the remainder of the paper, let $h = \log_{k+1}(P)$.

2 Tuning Communication Patterns

In this section, we describe the Tuning Communication Patterns (TCP), numbering schemas that define the algorithms.

2.1 TCP1 Broadcast

In TCP1, we define the broadcast of informations from a node, supposed hereinafter and without loss of generality to be node 0, to all the others. At each step, from nodes that possesses informations, k unidirectional communications are done to nodes which do not have informations in a tree-like fashion. At its completion, the TCP1 guarantees that for any node

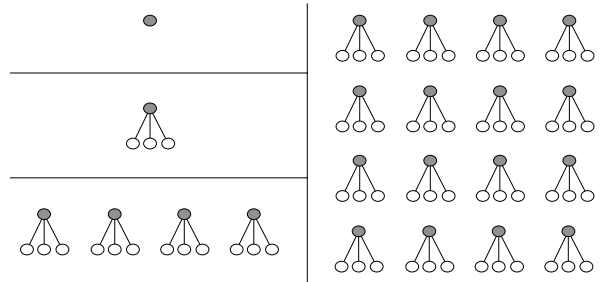


Figure 1: TCP1 description for 64 nodes and $k = 3$

in the system there was a path in time “connecting” it to node 0.

Hence, in a first step, node 0 communicate with k processors. In the next step, every already reached processor is able to have the same behavior as node 0 in the first step (which means that it will communicate with k other processors using the TCP1 procedure). This is iterated until all the processors involved in the communication operation are reached (see algorithm in Figure 2 and an example for $P = 64$ and $k = 3$ on Figure 1).

Clearly, $\log_{k+1}(P) = h$ is the number of steps needed to reach every node. Then, there are h tuning steps. Since the TCP1 can be seen as a forest of k -ary trees, the total number of modified wavelengths is $P - 1$.

Let $l > 0$ be the current step and $0 \leq i < (k+1)^{l-1}$ be the number of a node already reached. Therefore, we define the TCP1 in such a way that node i in step l will communicate with nodes:

$$(k+1)^l + ik, (k+1)^l + ik + 1, \dots, (k+1)^l + ik + k - 1$$

2.2 TCP2 Exchange

The second schema, TCP2, is a total exchange on a K_{k+1} clique, i.e. complete networks of k processors. In one communication step, each processor of the clique exchange with the k others the data it possesses, resulting in $k + 1$ times more information on each processor.

Figure 3 describes the procedure for several steps on a 64 nodes machine and the algorithm is depicted in Figure 4. A numbering schema is proposed in the following.

We define the TCP2 for one step in such a way that node $0 \leq i < (k+1)^{h-1}$ will communicate with nodes:

$$(k+1)^{h-1} + ik, (k+1)^{h-1} + ik + 1, \dots, (k+1)^{h-1} + ik + k - 1$$

```

i = my_node_id
for l = 1 to h do
  if i < (k + 1)l /* reached at step l - 1 */
    for j = 0 to k - 1 do
      i communicate with node (k + 1)l + ik + j
    endfor
  else
    if (k + 1)l ≤ i < (k + 1)l+1
      i receive information on its wavelength
    endif
  endif
endif
endfor

```

Figure 2: Algorithm implementing the TCP1

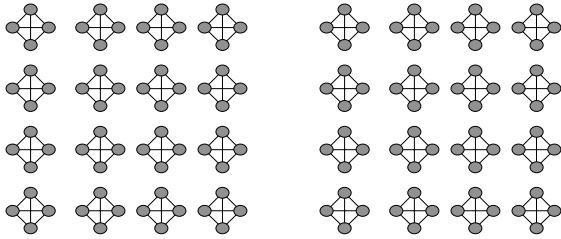


Figure 3: TCP2 description for 64 nodes and $k = 3$

For other steps, the numbering schema is of the same kind but will depend of the algorithm studied (all-to-all, broadcast, etc.). It has, for instance, to do the exact reverse of the broadcast-split phases in the broadcast algorithm.

Notice that we do not precise here if the message size in these TCPs procedures is the same at each step. In fact, we will see that the TCP1 procedure can be used with with a splitting of the messages at each step (broadcast-split) and that the TCP2 can be used with a grouping of messages (exchange-group).

```

i = my_node_id
for l = 1 to h do
  i exchanges with the nodes in its  $K_{k+1}(l)$ 
endfor

```

Figure 4: Algorithm implementing the TCP2

	TCP1	TCP2
Tuning cost	kD	$(k + 1)k$
Communication cost	1	1

3 Global communication schemas

3.1 Scattering algorithm

The scattering operation, or personalized one-to-all, is used for instance in the distribution of data on a set of processors [15, 19]. We only need the TCP1 procedure to describe this global communication operation. We assume that node 0 owns P contiguous messages that have to be delivered one to each node in the system. After tuning, in the first step of the algorithm, node 0 divides the set of messages in $k + 1$ equal contiguous parts of size $\frac{P}{k+1}$. One of these subsets remains on the node (it contains the messages that will be used in the next steps), while the k others are sent to the k nodes “listening” to it at this step.

At step 2, when the communication has complete, tuning occurs and each processor having been reached in the previous step behaves as node 0 did at the first step. Thus, there are $k + 1$ nodes having $\frac{P}{(k+1)}$ messages. The $k + 1$ nodes then communicate each with k “sons” and send one $(k + 1)$ -th of their initial message to each one.

At step l , when the communication has complete, tuning occurs and each processor having been reached in a previous step behaves as node 0 did at the previous steps. Thus, there are $(k + 1)^{l-1}$ nodes having $\frac{P}{(k+1)^l}$ messages. All the reached nodes then communicate with k “sons” and send one $(k + 1)$ -th of the set of messages to each one.

The algorithm stops after h steps, when every node has its own message. It is not difficult to see that, at this point, all the nodes have been reached.

The tuning cost is given by the tuning cost of the TCP algorithm, giving:

$$T_{tune}^{scat} = T_{tune}^{TCP1} = (P - 1)D$$

And the communication cost equals the sum of the costs for each step:

$$T_{com}^{scat} = \sum_{i=1}^h \frac{P}{(k+1)^i} = \frac{P-1}{k}.$$

Notice that the scattering strategy (which is often used, for instance, to collect results for storage and

treatment on a host workstation) is exactly the inverse of the gathering strategy and that this global algorithm can be implemented with TCP operations using unidirectional communications in the other way.

3.2 Multiple message broadcasting

The multiple broadcast is one of the basic operations on a number of parallel algorithms. It is used for example, to load the same code or data, or to communicate results on a network of processors. This operation has already been studied under the postal communication model by [2, 3].

3.2.1 Naïve algorithm

If we suppose that in the TCP, node 0 has m messages to be broadcasted, then a straightforward broadcast algorithm is obtained, where at each step, every reached node send the m messages along its k wavelengths using the TCP1 procedure.

Since each step costs m in communication time, we have a total cost of:

$$T_{tune}^{bcast} = T_{tune}^{TCP1} + hm = (P - 1)D + hm$$

3.2.2 Two phase algorithm

Suppose now that we change the naïve algorithm above by splitting the original set of messages in $(k+1)$ parts in the beginning of the algorithm. Then, the naïve algorithm proceeds and once all the processors are reached, node 0 has the whole set and each processor has $1/(k+1)$ subset. Then, in order to finish the broadcasting operation, the set of messages must be rebuilt in the processors. For this, processors that possesses complementary informations use the TCP2 procedures. They tuned their wavelengths to build $(k+1)$ -cliques (K_{k+1}) and they exchange using all their wavelengths the missing parts of the set of message. Since each $(k+1)$ -clique has the whole initial set, this single all-to-all operation allows every processor to recover the original set.

Notice that we can recursively apply this idea. We get then a *first* phase, where the set of messages is continuously broadcast-split according to the TCP; followed by a *second* phase, where the no-split broadcast TCP1 procedures are employed and then a third phase where the set of messages is, step after step, completely recomposed, with TCP2 operations on $(k+1)$ -cliques of complementary processors.

It is clear that the number of wavelengths used is larger during the third phase because it concerns

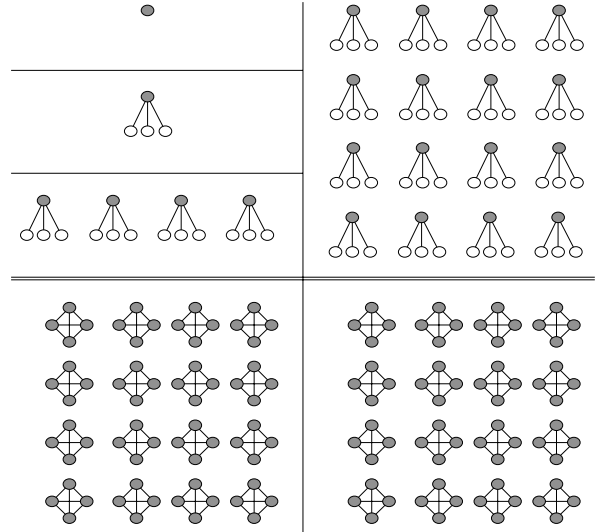


Figure 5: Broadcast algorithm using TCPs procedures with two steps (1.1 and 1.2) of TCP1 broadcast-split, for 64 nodes and $k = 3$

cliques and no longer trees subnetworks. Therefore, the tuning cost will be much higher. Actually, the longer the splitting phase, the more expensive the third phase because they must contain the same number of steps for the information to be completely rebuilt. Then, the savings in the communication time yielded by the splitting disappear under the weight of tuning. In order to reach the optimum, the idea is then to stop the broadcast-split of the first phase, at some step h' ($0 \leq h' \leq h$).

The first and second phases have h steps altogether in order to reach all nodes but in the TCP1-broadcast phase, the messages to be sent are smaller than the initial one because of the TCP1-broadcast-split steps.

With $P = 64$ and $k = 3$ is given in Figure 5, one can imagine that there are 1 steps in the first phase TCP1-broadcast-split, one in the second TCP1-broadcast and thus 2 TCP2-exchange steps in the second phase.

In the following, we give the complexity analysis of this algorithm:

$$T_{com}^{phase1} = \sum_{i=1}^{h'} \frac{m}{(k+1)^i}$$

$$T_{com}^{phase2} = \sum_{i=h'+1}^h \frac{m}{(k+1)^{h'}}$$

$$T_{com}^{phase3} = \sum_{i=h'}^1 \frac{m}{(k+1)^i}$$

giving a total communication cost:

$$T_{com}^{bcast} = \left(\frac{2}{k} \left((k+1)^{h'} - 1 \right) + h - h' \right) \frac{m}{(k+1)^{h'}}$$

The tuning cost is given by:

$$T_{tune}^{phase1+2} = T_{tune}^{TCP} = (P-1)D$$

$$T_{tune}^{phase3} = \sum_{i=h'}^1 (PkD) = h'PkD$$

giving a total tuning cost:

$$T_{tune}^{bcast} = [(P-1) + h'Pk]D.$$

If we assume that the broadcast-split phase lasts h steps, i.e., there is no second phase, the communication cost is given by:

$$T_{com}^{bcast} = 2 \left(\frac{P-1}{kP} m \right)$$

and the tuning cost is given by:

$$T_{tune}^{bcast} = [(P-1) + h'Pk]D,$$

The total cost of the TCP broadcasting T^{bcast} is given by the sum of the tuning cost T_{tune}^{bcast} and the communication cost T_{com}^{bcast} .

We now compute the optimal number of broadcast-split steps (h'_{opt}) which minimizes the total execution cost.

For our minimization, we have:

$$\frac{\partial(T_{com}^{bcast} + T_{tune}^{bcast})}{\partial h'} = 0.$$

Thus,

$$h'_{opt} = h - 1 + \frac{1}{\ln(k+1)} - \frac{\Omega \left(\frac{P^2 e k^2 D}{2m(k+1)} \right)}{\ln(k+1)}$$

Roughly, there is a tradeoff to find between the number of tunings and the communication cost. It will depend of the network parameters:

h'	Tuning	Communication
0	$(P-1)D$	hm
h	$[(P-1) + hPk]D$	$2 \left(\frac{P-1}{kP} \right) m$

Remark that the tuning patterns numbering can be pre-computed and stored in tables. Hence the Operating System or libraries will use it with no added cost during the execution.

3.3 Multiple all-to-all algorithm

The multiple all-to-all operation is usually very costly and is used in many parallel algorithms when all the processors have to exchange their data [18]. This operation is sometimes called gossiping [16] or complete broadcast [23].

This operation can be efficiently implemented on a Passive Optical Star network using several steps of the TCP2-exchange-group operations on K_{k+1} sub-networks.

The number of steps necessary to complete the algorithm is the same as for the one-to-all type operations (multiple broadcast and scattering).

The tuning cost is given by:

$$T_{tune}^{ata} = \sum_{i=0}^{h-1} (PkD) = hPkD$$

and the communication cost is given by:

$$T_{com}^{ata} = \sum_{i=0}^{h-1} ((k+1)^i m) = \frac{(k+1)^h - 1}{k} m = \frac{(P-1)m}{k}$$

3.4 Personalized all-to-all algorithm

This operation is also called total exchange [15] or multi-scattering [16].

We use the TCP2-exchange operations as for the multiple all-to-all schema, and at each step of the algorithm, each node sends a message of constant size $\frac{P}{k+1}$:

$$T_{tune}^{pata} = T_{tune}^{ata}$$

$$T_{com}^{pata} = \sum_{i=0}^{h-1} \frac{P}{k+1} = h \left(\frac{P}{k+1} \right)$$

4 Results summary and conclusion

In tables 1 and 2, we give a summary of the complexity results obtained. The lower bounds in our model were obtained by a simple evaluations of the best cases where the hardware is supposed to be fully used at any moment. One can see that our algorithms match the lower bound for communication costs in two of the problems and is in a factor 2 and h in the two others. As far as broadcasting is concerned, there is a tradeoff between the communication and the tuning costs, depending on the length of the broadcast-split phase.

Algorithm	Communication cost	Lower bound
ota ($h' = 0$)	hm	$\frac{m}{k}$
ota ($h' = h$)	$\frac{2(P-1)m}{k}$	$\frac{m}{k}$
pota	$\frac{(P-1)}{k}$	$\frac{(P-1)}{k}$
ata	$\frac{(P-1)}{k}$	$\frac{(P-1)}{k}$
pata	$h \left(\frac{P}{k+1} \right)$	$\frac{(P-1)}{k}$

Table 1: Summary of the Communication costs and lower bounds

Algorithm	Tuning cost
ota ($h' = 0$)	$(P-1)D$
ota ($h' = h$)	$((P-1) + hPk)D$
pota	$(P-1)D$
ata	$hPkD$
pata	$hPkD$

Table 2: Summary of the Tuning costs

Remark that the numbering schemas introduced in the preceding works only for multi-computers with a number of nodes which is exactly a power of $(k+1)$. Finding a numbering schema for an arbitrary number of nodes is not difficult (one can just assume that dummy nodes exist until the next power of $(k+1)$) but it will result in a little lose of efficiency in the algorithms (at most one step of TCP procedures). Anyway, we think that, matching the lower bounds for any P is a very challenging theoretical problem but that it is not worth the gain in practise.

With these results in mind, the WDMA technology seems to be very promising for massively parallel computers and will guarantee a very good efficiency for most of the algorithms by reducing the communications costs.

Remark that if the message are not atomic, i.e. $L\tau \neq 1$, the preceding results can be easily adapted and compared directly with the results of the classical studies (usually with $m = 1$) on non-optical wormhole protocol networks [4, 5, 6, 14].

Our ongoing work is the implementation of these communication routines on an Intel Paragon machine to validate the TCP approach. We also want to have timings to compare the TCP algorithms with the classical communications routines implemented on a non-optical wormhole interconnection network because this step by step communication strategy can fit very well with our works on communication/computation

overlap [13].

References

- [1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient Routing and Scheduling Algorithms for Optical Networks. Technical report, IBM - Research Division - T.J. Watson Research Center, 1993.
- [2] A. Bar-Noy and S. Kipnis. Designing Broadcasting Algorithms in the Postal Model for Message-Passing Systems. In *SPAA*, 1992. To Appear in *Mathematical Systems Theory*.
- [3] A. Bar-Noy and S. Kipnis. Broadcasting Multiple Messages in Simultaneous Send/Receive Systems. In *SPDP*, 1993. To appear in *Discrete Applied Mathematics*.
- [4] M. Barnett, R. Littlefield, D.G. Payne, and R. Van De Geijn. Efficient Communication Primitives on Mesh Architectures with Hardware Routing. In R.F. Sincovec, D.E. Keyes, M.R. Leuze, L.R. Petzold, and D.A. Reed, editors, *Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pages 943-948. SIAM, 1993.
- [5] M. Barnett, D.G. Payne, and R. Van De Geijn. Optimal Broadcasting in Mesh Connected Architectures, December 1991.
- [6] C. Bonello, F. Desprez, and B. Tourancheau. Parallel Blas and Blacs for Numerical Algorithms on a Reconfigurable Network. In S. Atkins and A.S. Wagner, editors, *NATUG6 - Transputer Research and Applications 6*, pages 21-38. IOS Press, 1993.
- [7] J. Bruck, R. Cypher, and C.T. Ho. Efficient Algorithms for the Index Operation in Message-Passing Systems. Technical Report RJ 9300 (80030), IBM Research Division - Almaden Research Center - San Jose, April 1993.
- [8] J. Bruck and C.T. Ho. Concatenating Data Optimally in Message-Passing Systems. Technical Report RJ 9191 (81499), IBM Research Division - Almaden Research Center - San Jose, January 1993.
- [9] J. Bruck and C.T. Ho. Efficient Global Combine Operations in Multi-Port Message-Passing Systems. Technical Report RJ 9333 (82457), IBM Research Division - Almaden Research Center - San Jose, May 1993.

- [10] M. Cosnard, Y. Robert, and B. Tourancheau. Evaluating speedups on distributed memory architectures. *Parallel Computing*, 10:247–253, 1989.
- [11] F. Desprez and B. Tourancheau. Matrix Multiplication and Matrix Transpose: Fixed Topology vs. Switching Network (poster). In *EDMCC2 Munich*, 1991.
- [12] F. Desprez and B. Tourancheau. A Theoretical Study of Reconfigurability for Basic Communication Algorithms. In L. Bougé, M. Cosnard, Y. Robert, and D. Trystram, editors, *CONPAR 92 - VAPP V*, number 634 in Lecture Notes in Computer Science. Springer Verlag, 1992.
- [13] F. Desprez and B. Tourancheau. LOCCS: Low Overhead Communication and Computation Subroutines. In *High Performance Computing and Networking Conference - Amsterdam*. Elsevier, May 1993.
- [14] J.J. Dongarra, R. Van De Geijn, and R.C. Whaley. Two Dimensional Basic Linear Algebra Communication Subprograms. In J.J. Dongarra and B. Tourancheau, editors, *Environments and Tools For Parallel Scientific Computing*. Elsevier, September 1992.
- [15] A. Edelman. Optimal Matrix Transposition and Bit Reversal on Hypercubes: All to All Personalized Communication. *Journal of Parallel and Distributed Computing*, 11:328–331, 1991.
- [16] P. Fraigniaud and E. Lazard. Methods and Problems of Communication in Usual Networks. Technical Report 91-33, Laboratoire LIP, 1991.
- [17] H. Ito, N. Komagata, H. Yamada, and Humio Inaba. New structure of laser diode and light emitting diode based on coaxial transverse junction. Technical report, Research Institute of Electrical Communication, Tohoku University, Sendai 980, Japan.
- [18] S.L. Johnsson and C.T. Ho. Algorithms for Matrix Transposition on Boolean n-cube Configured Ensemble Architectures. Technical Report YALEU/DCS/TR-572, Department of Computer Science - Yale University, September 1987.
- [19] S.L. Johnsson and C.T. Ho. Optimum Broadcasting and Personalized Communication in Hypercubes. *IEEE Transaction on Computers*, 9(38):1249–1268, 1989.
- [20] P. Lalwaney, L. Zenou, A. Ganz, and I. Koren. Optical Interconnect for Multiprocessors Cost Performance Trade-Offs. In H.J. Siegel, editor, *The Fourth Symposium on the Frontiers of Massively Parallel Computation*, pages 278–285. IEEE Computer Society Press, 1992.
- [21] Y. Robert, B. Tourancheau, and G. Villard. Data allocation strategies for the gauss and jordan algorithms on a ring of processors. *Information Processing Letters*, 31:21–29, 1989.
- [22] Y. Saad. Gaussian Elimination on Hypercubes. In M. Cosnard, Y. Robert, P. Quinton, and M. Tchente, editors, *Parallel Algorithms and Architectures*. North-Holland, 1986.
- [23] S.R. Seidel. Circuit-Switched vs. Store and Forward Solutions to Symmetric Communication Problems. In *HCCA 4*, 1989.
- [24] S.R. Tong and D.H.C. Du. Design Principles for Multi-Hop Wavelength and Time Division Multiplexed Optical Passive Star Networks. Technical report, Computer Science Department - University of Minnesota, 1993.