A Performance Comparison Between Graph and Hypergraph Topologies for Passive Star WDM Lightwave Networks*

H. Bourdin
LIP ENS Lyon
46, allée d'Italie
69364 Lyon Cédex 07
France

A. Ferreira†
CNRS - I3S - INRIA
Project SLOOP
BP 93
F-06902 Sophia-Antipolis
France

K. Marcus‡
Institut Eurecom
BP 193
F-06901 Sophia-Antipolis
France

Abstract

Wavelength division multiplexing (WDM) allows the huge bandwidth of optical fiber to be divided into several high-speed channels in optical passive star based networks. For such processor networks, most of the proposed architectures for interconnecting nodes are based on graph topologies. Recently, topologies based on the hypergraph theory have emerged, motivated by the observation that each multiplexed channel can actually be seen as a logical resource shared among many processors, and not only between two of them. In this paper, we show that these hypergraph passive star WDM lightwave networks present many advantages with respect to graph-based ones, in terms of simulated packet delivery time, average number of hops, link utilization, and throughput. Furthermore, they use only a constant number of transceivers per node, and a sub-linear number of multiplexed channels.

Keywords: Computer Network Topologies, Optical Passive Star, Lightwave Networks, Performance Models, Hypergraphs, Routing Simulations.

1 Introduction

Optical technologies such as tunable optical transmitters and receivers, and wavelength division multiplexing (WDM) allow the construction of very efficient local and metropolitan area networks (LAN and MAN, respectively). Using Optical Passive Star (OPS) Couplers, one can build single-hop systems, where every processor is able to directly communicate with one another with no intermediary nodes (see Figure 1). In order to implement such a system, the processors' transceivers have to dynamically tune to the channels through which the communication takes place; this tuning time varies from a few milliseconds to a few microseconds over a quite broad wavelength ([6]), which

*This work has been partially supported by the PRC PRS and ANM of the French CNRS, by the European Human Capital and Mobility project MAP, and by the Brazilian CAPES.

†Corresponding author. E-mail: ferreira@sophia.inria.fr. Part of this work was done when the author was with LIP ENS Lyon.

‡Part of this work was done when the author was with ARTEMIS - IMAG.

is considered to be very slow in comparison to a typical packet transmission time. Therefore, this could represent a severe drawback when building very large networks ([10]).

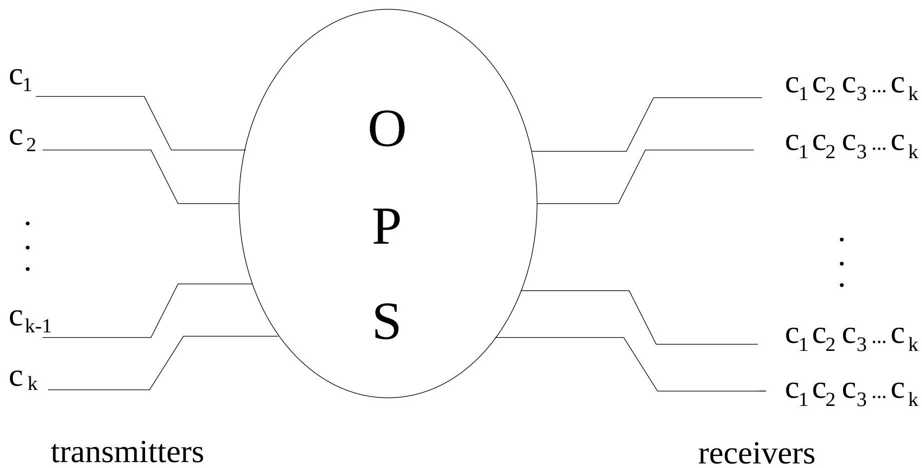

Figure 1: A single-hop passive star lightwave network. The $c_{i's}$ represent the channels which can be accessed by each processor's transceivers.

On the other hand, the same kind of OPS couplers can be used in the construction of multi-hop networks, where a node is assigned to a small and static set of predefined channels, that rarely change, usually to improve network performance (see Figure 2). Pairwise communication may then need to hop through intermediate nodes ([2, 11]). Thus, in multi-hop systems, communications take longer, but nodes are simpler, cheaper, and more reliable than in single-hop systems.

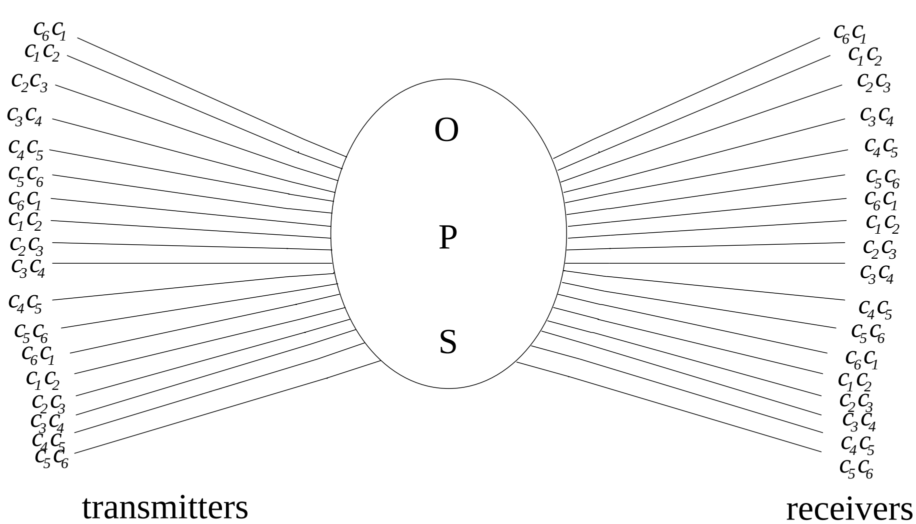

Figure 2: A multi-hop passive star lightwave network. The $c_{i's}$ represent the channels which can be accessed by each processor's transceivers.

Several topologies were proposed as point-to-point logical architectures for WDM passive star networks. In [12, 14], different aspects of the de Bruijn, torus, and supercube graphs are studied and simulated with respect to their possible use in network design. Even the well known Manhattan street network is based on a point-to-point torus architecture ([9]). Unfortunately, these topologies either use too many channels, have too many transceivers, or suffer of a large diameter.

Notice, further, that an intrinsic feature of optical communications is that the channels induced by WDM can span a large number of nodes in the network. Hence, point-to-point logical topologies do not efficiently use optical technology and new avenues have to be explored.

One-to-many topologies As discussed above, point-to-point topologies are based on graphs. On the other hand, *one-to-many* topologies are best represented by hypergraphs ([3]), that can be seen as a generalization of graphs in which edges are replaced by hyperedges[1] joining sets of nodes, instead of only two nodes. As for graphs, a hyperedge represents a communication means; a message sent on a hyperedge can be read by all nodes in that hyperedge. In the case of WDM passive star networks, a specific wavelength (or channel) can be seen as a logical bus and, as pointed out in [15], a logical bus can be modeled by a hyperedge (see Figure 3 for the hypergraph representation of the multi-hop network depicted in Figure 2). To make this idea clear, let stations A, B, and C form a hyperedge (i.e., share a channel) and the channel speed be v Mbit/s. Then, if A has v Mbits of data, they can be transferred to *both* B and C in one second.

Our work is focused on new such architectures for multi-hop systems, whose main performance characteristics are their regularity and modularity ([5, 8]). Instead of being based on models arising from graph theory, these networks are designed with ideas stemming from hypergraph theory, as described above. These topologies can be incrementally expanded, that is one can add a new node to the existing network without a major reconfiguration. Another positive point is that the number of links between two processors is fixed, thus allowing that each node has a set of transmitters and receivers tuned to some pre-determined frequencies. The fault tolerance is proved to be very high, assuring a reliable communication and network performance. Finally, only a small number of hops is required in any pairwise communication, and the network allows easy schemes for global routing operations ([5, 8]).

In this paper we present a comparative study of three hypergraph–based networks — namely the *stack-ring*, the *stack-torus*, and the *hypertorus* ([5]) —, and two very well known graph–based

[1] Although intuitive, it seems that the term hyperedge has been first used in an early version of [15].

topologies — namely the *hypercube* and the two–dimensional *torus* ([7]). It is clear that one-to-many topologies would be much more efficient than point-to-point topologies when implementing multicast (i.e., one-to-many) communications. However, since many applications require one-to-one communications, in the subsequent sections we study their routing–related stochastic behavior for one-to-one communications that were obtained through simulation. It will be shown that even in this case, hypergraph–based topologies can be more efficient than graph–based topologies.

In the next section we describe the hypergraph–based networks whose performance was simulated. The simulator itself is presented in section 3, where its model, assumptions, and experiments are discussed. The results are given and analysed in section 4. We close the paper with some concluding remarks and directions for further research.

2 Emerging networks

Many interconnection topologies have been proposed for the design of LANs and MANs using multihop lightwave techniques in passive star WDM networks. Among others, we can cite the Shuffle Exchange network ([1]), the Manhattan Street network ([9]), de Bruijn graphs based network ([14]) and the Supercube ([13]). These networks are based on graph models, and the possibility of a node to communicate with another one is represented by an edge joining the two nodes. Thus, in any communication step, only pairs of nodes are involved. In [5, 8], it was proposed to grow the number of nodes involved in each step, profiting mostly from the fact that multiplexed channels can actually be seen as logical resources shared among many processors and hence modeled by hyperedges. The network topologies are based on hypergraph models, and called hypertopologies.

2.1 Hypertopologies

In the hypergraph representation, a channel is a hyperedge joining many nodes, and the processors are the nodes of the hypergraph. In the remainder of the paper we shall use hypergraph or lightwave network interchangeably. In the following we formally define the hypertopologies to be simulated in sections 3 and 4. They use mainly the concept of *stack-graphs* ([4, 5]) and their Cartesian product. In a nutshell, a stack-graph of order m and size n is obtained by piling up m copies of the original graph of n nodes and subsequently replacing the edges by hyperedges (see Figure 3 for an illustration of a case where 3 copies of a ring of size 6 are piled up). Consequently, the stack-graph will have nm nodes.

2.1.1 Stack-ring

Let R be a ring with n vertices labeled from 0 to $n-1$, and edges of the form $(i, (i+1) \bmod n)$, for $0 \leq i < n$.

Definition 1 (Stack-ring $\mathcal{R}_{n,m}(V, \mathcal{E})$) A stack-ring (see Figure 3) of size n and order m is a hypergraph $\mathcal{R}_{n,m}(V, \mathcal{E})$, with the set V of vertices and the set $\mathcal{E}$ of hyperedges defined as follows.

- ▹ $V = \{(0,0), (0,1), \ldots, (0, m-1), (1, 0), \ldots, (1, m), \ldots, (m-1, 0), \ldots, (m-1, n-1)\}$ and
- ▹ $e \in \mathcal{E} \iff e = \{(0, i), \ldots, (m-1, i), (0, (i+1) \bmod n), \ldots, (m-1, (i+1) \bmod n)\}$, for $0 \leq i < n$.

□

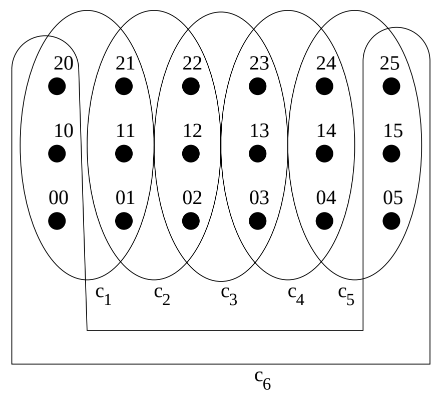

Figure 3: Stack-ring of size 6 and order 3 ($\mathcal{R}_{6,3}$). The $c_{i's}$ represent the channels which can be accessed by each processor's transceivers.

We recall that Figure 2 shows a passive star implementation of the stack-ring depicted above.

2.1.2 Stack-torus

Stack-rings can be seen as a generalization of rings from graphs to hypergraphs using the concept of *stack-graphs*. We can thus do the same thing to generalize tori into stack-tori (see Figure 4), by piling up m copies of a torus of n nodes in order to get a stack-torus with nm nodes.

2.1.3 Hypertorus

Stack-rings and stack-tori have been defined by a generalization of classical graphs to hypergraphs being in addition regular and uniform. In order to obtain another kind of hypertopology, the

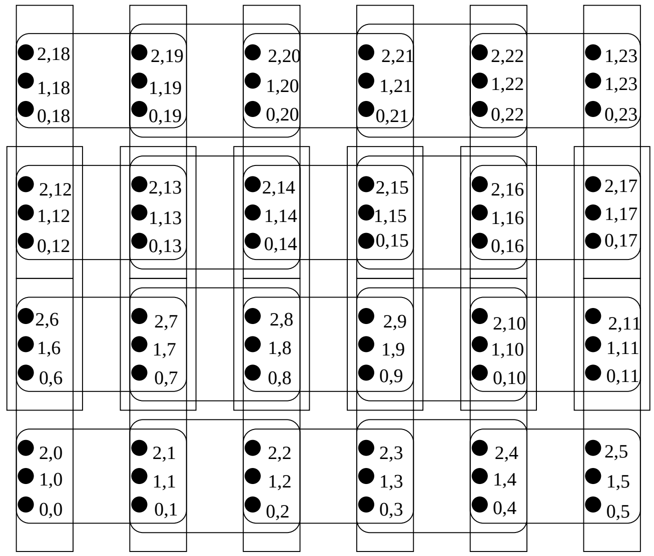

Figure 4: A stack-torus of size 4×6 and order 3 ($\varsigma(T, 3)$). For the sake of clarity, wrap-around channels are not represented.

hypertori, a Cartesian product of stack-rings is used, rather than the generalization of tori.

Among all the various existing definitions for the Cartesian product of hypergraphs, the one adopted in this paper is as follows.

Definition 2 (Cartesian product for hypergraphs) The Cartesian product of two hypergraphs $\mathcal{G}^1(V^1, E^1)$ and $\mathcal{G}^2(V^2, E^2)$ is a hypergraph $\mathcal{G}(\mathcal{V}, \mathcal{E}) = \mathcal{G}^1(V^1, E^1) \times \mathcal{G}^2(V^2, E^2)$ such that:

- $V = \{v_i\}$, $0 \leq i < n_1 n_2$, $v_i = (v_j^1, v_k^2)$ with $j \in \{0, \ldots, n_1 - 1\}$ and $k \in \{0, \ldots, n_2 - 1\}$, where $n_1 = |V^1|$ and $n_2 = |V^2|$.

- $\mathcal{E} = \{\{a, b\},\ \text{for all } b \in e_2\}, \forall a \in V^1, \forall e_2 \in E^2\} \bigcup$
  $\{\{c, d\},\ \text{for all } c \in e_1\}, \forall d \in V^2, \forall e_1 \in E^1\}.$

□

Below we show how this operation can be used to construct more complex hypergraphs from simple pieces as building blocks. Just for a start, notice that the stack-ring can be redefined as the Cartesian product of the original ring of size n by a single hyperedge of cardinality m. Roughly speaking, the idea is that each hypertopology is replicated as many times as there are nodes in the other, while maintaining the adjacency relations.

Definition 3 (Hypertorus $\mathcal{T}$) Using the given definition of the Cartesian product, a 2-dimensional hypertorus (see Figure 5) is a product of 2 stack-rings ($\mathcal{T} = \mathcal{R}^1_{n_1,m_1} \times \mathcal{R}^2_{n_2,m_2}$).

□

Notice that the hypertorus is different from the stack-torus proposed above (compare Figures 4 and 5), the former requiring more channels than the latter and consequently showing either a smaller diameter or fewer processors per channel. The hypertorus is not globally uniform and regular. However, both regularity and uniformity can be retrieved in each dimension separately.

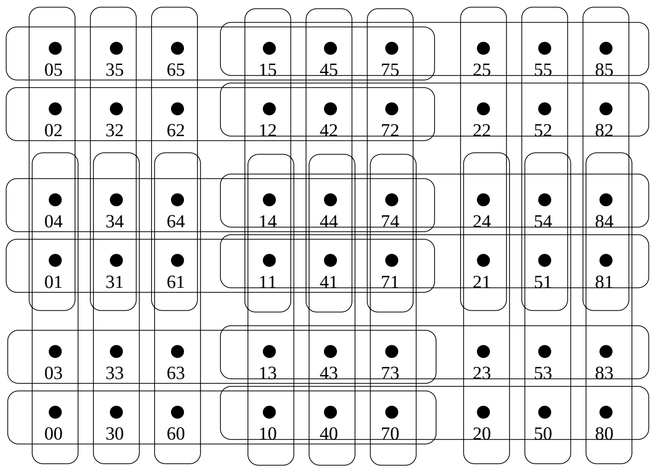

Figure 5: A hypertorus $\mathcal{R}_{3,3}\times\mathcal{R}_{3,2}$. For the sake of clarity, wrap-around channels are not represented.

3 Simulation

Sen and Maitra presented in [12] a simulator used to evaluate the dynamic quality of point-to-point lightwave networks. Using the same framework, we implemented a similar simulator for the hypergraph–based lightwave networks studied in this paper.

3.1 The simulator

The simulated networks communicate under the store-and-forward mode. Hence, when an intermediate node receives a message which is not at its final destination, the node forwards the message to the first node in its shortest path to the destination. To implement this, the hypertopology is described by an adjacency matrix and a routing table. For a node n and a hyperedge h, the value in the adjacency matrix of $adjacency[n][h]$ is 0 if the node n is not adjacent to the hyperedge h, i if the $i-th$ port of n is connected to h. For two nodes $n1$ and $n2$, the value in the routing table of $routing[n1][n2]$ is the second node of a shortest path going from the node $n1$ to the node $n2$ (we consider that $n1$ is the first node of this path). There are four main functions that are used to build the networks:

`createBus(size)` creates a logical channel interconnecting `size` processors.

`createRing(size)` creates a point-to-point ring with `size` processors.

`CartesianProduct(ht1, ht2)` creates the Cartesian product of two hypertopologies `ht1` and `ht2`.

`stackGraph(m, G)` creates the stack-graph $\varsigma(G, m)$.

By combining these functions, we can easily create functions for building tori, hypercubes, stack-rings, hypertori, and stack-tori. All these functions give an adjacency matrix and a routing table (`CartesianProduct()` and `stackGraph()` build the respective optimal routing tables with the help of the given hypertopologies).

3.2 Simulation parameters

There are three control parameters, namely **load**, **speed** and **number of nodes**. The *offered load* is the number of packets generated at each node per second. The inter-packet generation time is exponentially distributed. This load has been varied from 10 packets/s to 500 packets/s. The *channel speed* is the number of bits that can be transmitted per second. This speed has been varied from 40Mbit/s to 160Mbit/s. Finally, we also controlled the *number of nodes* in the considered network. Table 1 shows the corresponding configurations for the studied hypertopologies. We simulated configurations having between 36 and 320 nodes.

Three other parameters remained fixed during our simulations. They were the **packet size**, the **fault frequency**, and the **fault duration**. The *packet size* has been fixed to 8000 bits. The *fault frequency* is the number of transmission errors per second caused by a link failure. During the time a link is faulty, the packets flowing on this link will be lost and must be resent. Furthermore, when a link is faulty, it remains in that state for a duration of time given by an exponential distribution with mean equal to *fault duration*.

The five statistics of interest collected per simulation are:

Packet delivery time The mean time taken by packets from their creation up to their successful arrival at destination.

Waiting time The mean time taken by packets in the waiting for a communication link.

Link utilization The average link utilization is the mean taken over all links of the ratio of the link utilization time to the simulation time.

Throughput The number of successfully delivered packets per unit of time.

Number of hops The mean number of hops in the packet transmission path.

3.3 Simulation experiments

We performed three sets of tests. The first was done with a varying load, the second with a varying speed and the last with a varying number of nodes. When fixed, these parameters were set as follows: 100 packets per second as offered load, 100 Mbit/s as channel speed and 120 nodes.

The simulated hypertopologies are described in Table 1, and the characteristics of the simulated hypercubes and tori are given in Table 2.

diameter	number of nodes	stack-ring	hypertorus	stack torus
2	36	$\mathcal{R}_{4,9}$	$\mathcal{R}_{3,2} \times \mathcal{R}_{3,2}$	$\varsigma(T(3,3),4)$
	54	$\mathcal{R}_{4,13}$	$\mathcal{R}_{3,2} \times \mathcal{R}_{3,3}$	$\varsigma(T(3,3),6)$
	72	$\mathcal{R}_{4,18}$	$\mathcal{R}_{3,2} \times \mathcal{R}_{3,4}$	$\varsigma(T(3,3),8)$
3	90	$\mathcal{R}_{6,15}$	$\mathcal{R}_{5,2} \times \mathcal{R}_{3,3}$	$\varsigma(T(5,3),6)$
	120	$\mathcal{R}_{6,20}$	$\mathcal{R}_{5,2} \times \mathcal{R}_{3,4}$	$\varsigma(T(5,3),8)$
	150	$\mathcal{R}_{6,25}$	$\mathcal{R}_{5,2} \times \mathcal{R}_{3,5}$	$\varsigma(T(5,3),10)$
4	192	$\mathcal{R}_{8,24}$	$\mathcal{R}_{4,4} \times \mathcal{R}_{4,3}$	$\varsigma(T(4,4),12)$
	256	$\mathcal{R}_{8,32}$	$\mathcal{R}_{4,4} \times \mathcal{R}_{4,4}$	$\varsigma(T(4,4),16)$
	320	$\mathcal{R}_{8,40}$	$\mathcal{R}_{4,4} \times \mathcal{R}_{4,5}$	$\varsigma(T(4,4),20)$

Table 1: Number of nodes for the studied hypertopologies.

Remark: Notice that, because of topological constraints, the stack-ring $\mathcal{R}_{4,13}$ is composed of 52 nodes instead of 54 nodes. $\square$

Finally, when testing varying load and varying channel speed, we used fixed (hyper) networks. Thus, we decided to simulate, whenever possible, (hyper) networks with 120 nodes. (As a matter of

topology	configuration	diameter	number of nodes
torus	$T(6,6)$	6	36
	$T(6,9)$	7	54
	$T(8,9)$	8	72
	$T(10,9)$	9	90
	$T(10,12)$	11	120
	$T(10,15)$	12	150
	$T(16,12)$	14	192
	$T(16,16)$	16	256
	$T(16,20)$	18	320
hypercube	$H(5)$	5	32
	$H(6)$	6	64
	$H(7)$	7	128
	$H(8)$	8	256

Table 2: Number of nodes for the hypercube and the 2-dimensional torus.

fact, the throughput of the hypercube is better than all other networks simply because its number of nodes (128) is more important than in the other networks (120).) Furthermore, all hypernetworks have diameter 3. Table 3 shows the simulated (hyper) networks, giving the chosen configuration with number of channels (hyperedges), number of fixed transceivers per node (degree), and number of processors per channel (rank).

topology	configuration	nodes	hyperedges	degree	rank
Stack-ring	$\mathcal{R}_{6,20}$	120	6	2	40
Hypertorus	$\mathcal{R}_{5,2} \times \mathcal{R}_{3,4}$		90	4	8
Stack-torus	$\varsigma(T(5,3),8)$		30	4	16
Torus	$T(10,12)$		240	4	2
Hypercube	$H(7)$	128	448	7	2

Table 3: Hypertopologies with 120 nodes and a diameter of 3.

4 Results of the comparison

As expected, because of their good hypergraph-theoretic properties discussed in [8], the hypernetworks outperform graph–based networks in almost all aspects. If we further recall that these hypernetworks use only a constant number of transceivers per node, and a sub-linear number of multiplexed channels, then it seems that they represent a reasonable alternative to graph–based LANs and MANs. In the following we give figures plotting the collected data.

4.1 Varying number of nodes

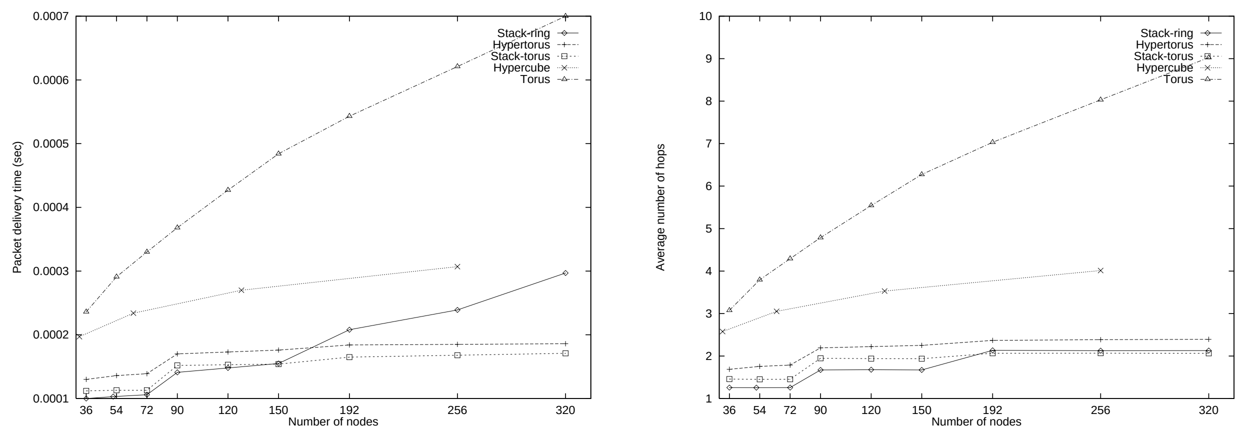

Figure 6: Packet delivery time and average number of hop versus number of nodes.

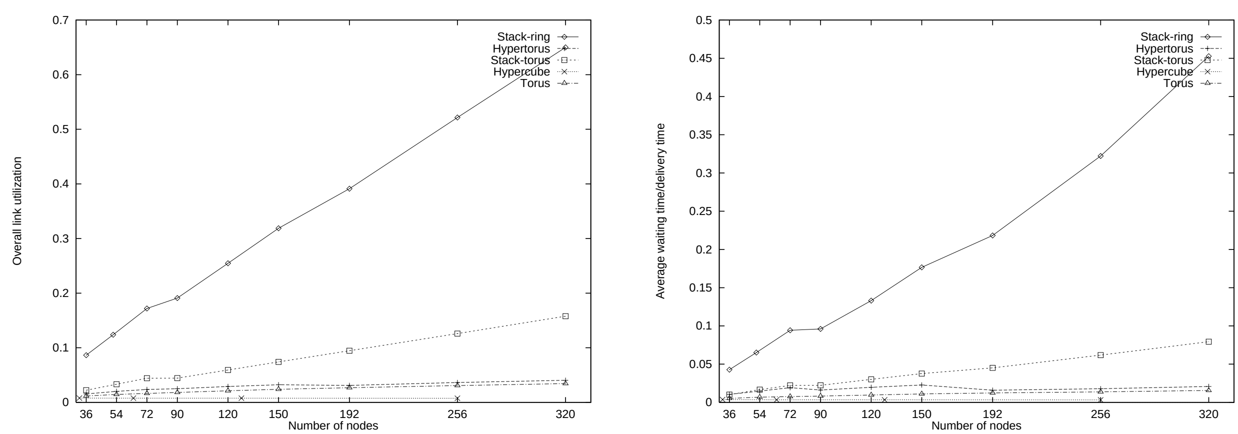

Figure 7: Link utilization and waiting ratio versus number of nodes.

In Figure 6, we see that the performance of the torus with respect to the packet delivery time and average number of hops is much worse than the other topologies. This is easily explained by the fact that the diameter of the torus augments more than the others when increasing the number of nodes. The hypergrid and the stack-torus have a similar performance, and the hypercube has a rather constant performance, but worse than the hypernetworks, also because of its greater diameter.

Figure 7 shows two parameters that strongly depend on the number of channels (edges or hyperedges). The greater this number, the smaller the overall channel (or link) utilization and the average ratio of waiting time to delivery time. This is confirmed by the rather bad performance of the stack-ring, the good one of the hypercube, and the almost similar behavior of the other three networks.

4.2 Varying channel speed

Recall that while varying channel and load, all the simulated networks have 120 nodes, except for the hypercube, which has 128 nodes because of topological constraints.

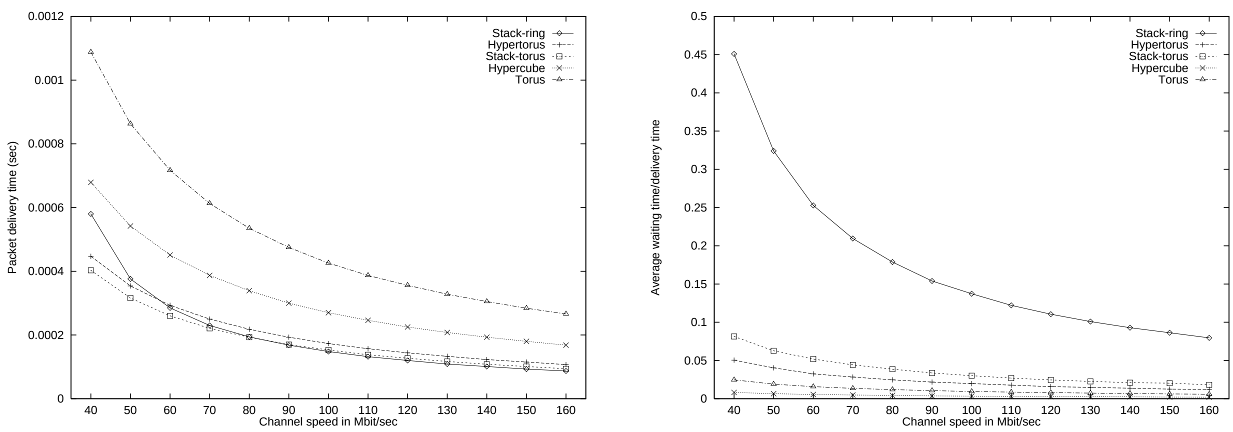

Figure 8: Packet delivery time and waiting ratio versus channel speed.

Figure 8 shows that, with respect to the packet delivery time, the hypertopologies have a better performance than the graph–based ones, although the asymptotic behavior is quite the same for all networks. Concerning the average waiting time / delivery time, and the overall link utilization shown in Figures 8 and 9, we recall that both parameters depend on the number of channels. Therefore, the stack-ring behaves poorly in comparison to the others, that have all a similar behavior.

Figure 9 depicts also the results on the throughput. The hypercube outperforms the others because it has 128 nodes instead of 120.

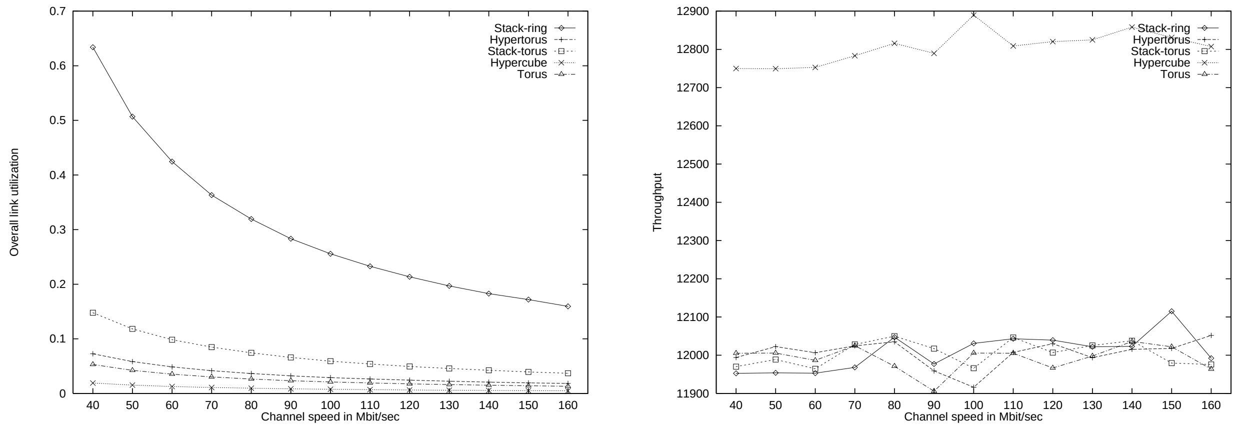

Figure 9: Link utilization and throughput versus channel speed.

4.3 Varying load

In Figure 10 one can see that the small number of channels of the stack-ring is a drawback for a good performance. With respect to the packet delivery time, the hypertopologies have the same behavior, better than both the torus and the hypercube. Concerning the average ratio waiting time to delivery time (figure 10) and the link utilization (figure 11), they present similar behaviors, showing that the number of channels in the network plays an important role, as it can be confirmed by Table 3. Hence, graph–based networks had better performance because of their larger number of edges.

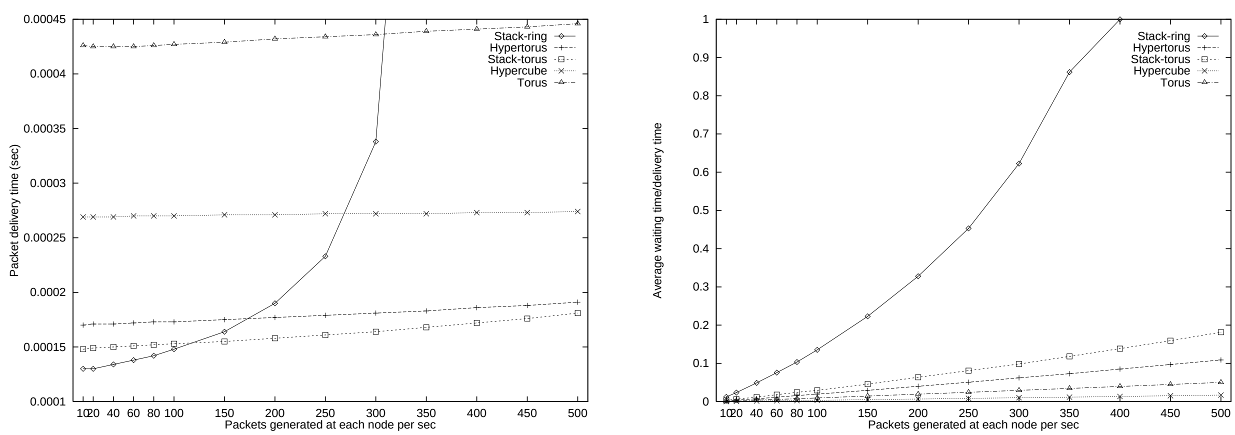

Figure 10: Packet delivery time and waiting / delivery time versus load.

Figure 11 (throughput) shows a homogeneous behavior of all networks, with a slight advantage for the hypercube, that has more nodes than the others. We remark that we do not collect results over 400 packets for the stack-ring due to an excessively long simulation running time.

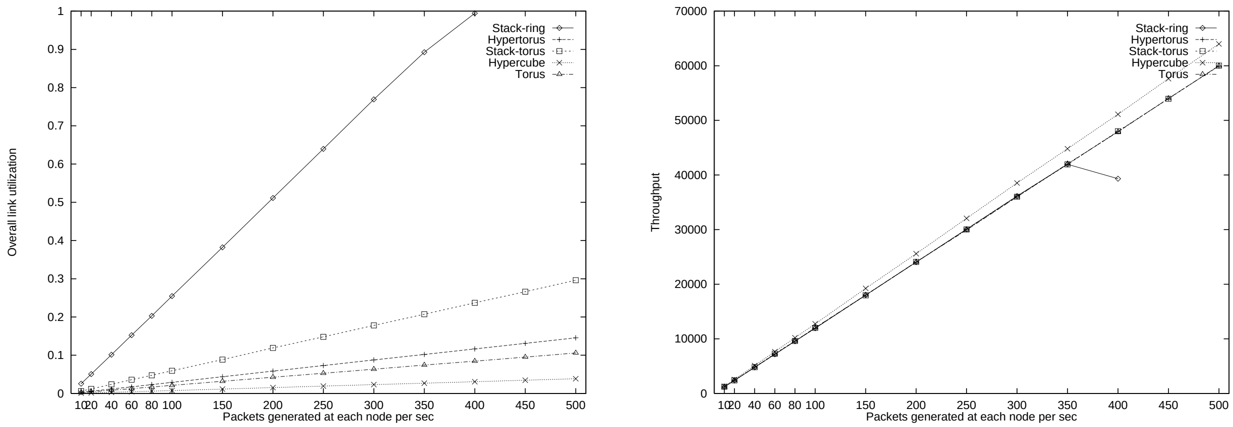

Figure 11: Link utilization and throughput versus load.

4.4 Remarks

Clearly, hypertopologies are best suited for one-to-many communications. Nevertheless, the results above show that, in general, the hypertopologies have a better performance even in the case of one-to-one communications. However, it is absolutely necessary to find a good tradeoff between their number of channels and diameter. For instance, the stack-torus used in the simulations had 30 channels and diameter 3, showing a good average performance, while the stack-ring that has diameter 3, but only 6 channels, behaves quite poorly with respect to experiments where the number of channels is involved. Therefore, an adequate balance among channels, degree, and diameter should be obtained for hypernetworks.

5 Conclusions and perspectives

We presented a comparative study of three hypergraph–based networks and two well known graph–based networks. Our goal was to test these emerging proposals for the logical interconnection of high performance LANs and MANs against usual ones. It had previously been shown that hypernetworks outperform graph–based ones in their graph–theoretic properties. In this paper, our simulations showed that the hypernetworks can be more efficient than graph–based ones also with respect to their routing–related stochastic behavior, even for one-to-one communications.

An important issue in this subject is to decide which of the several possible hypertopologies is best suited for implementation of high performance local and metropolitan area networks. Also, the study of routing-related stochastic characteristics and access protocols for multicast (one-to-many) communications in hypertopologies is a very interesting direction for further research.

Acknowledgments

We are grateful to the anonymous referees whose insightful remarks helped improve considerably the quality of the original manuscript.

References

[1] A.S. Acampora. A multichannel multihop local lightwave network. In *Proc. IEEE GLOBE-COM'87*, pages 459–467, nov 1987. Tokio, Japan.

[2] A.S. Acampora and M.J. Karol. An overview of lightwave packet networks. *IEEE Network*, 3(1):29–41, jan 1989.

[3] C. Berge. *Hypergraphs*. North Holland, 1989.

[4] P. Berthomé and A. Ferreira. Improved embeddings in POPS networks through stack-graph models. In *Proceedings of the 3rd IEEE International Conference on Massively Parallel Processing using Optical Interconnections*, Hawaii (USA), oct 1996. IEEE Press.

[5] H. Bourdin, A. Ferreira, and K. Marcus. A comparative study of one-to-many WDM lightwave interconnection networks for multiprocessors. In *Proceedings of the 2nd IEEE International Workshop on Massively Parallel Processing using Optical Interconnections*, San Antonio (USA), oct 1995. IEEE Press.

[6] C. A. Brackett. Dense Wavelength Division Multiplexing Networks: Principles and Applications. *IEEE J. Sel. Areas Commun.*, 8:948–964, aug 1990.

[7] A. Ferreira. *Handbook of Parallel and Distributed Computing*, chapter Hypercubes. McGraw-Hill, New York (USA), 1995.

[8] A. Ferreira and K. Marcus. Modular multihop WDM–based lightwave networks, and routing. In S. I. Najafi and H. Porte, editors, *Proceedings of The European Symposium on Advanced Networks and Services, Conference on Receivers, Transmitters, and WDMs for Fibre Optic Networks*, volume 2449 of *Proc. SPIE*, pages 78–86, Amsterdam, March 1995. SPIE – The International Society for Optical Engineering.

[9] N.F. Maxemchuk. Regular mesh topologies in local and metropolitan area networks. *AT&T Tech. J*, 64(7):1659–1685, 1985.

[10] B. Mukherjee. WDM-Based Local Lightwave Networks Part I: Single-Hop Systems. *IEEE Network*, 6(3):12–27, may 1992.

[11] B. Mukherjee. WDM-Based Local Lightwave Networks Part II: Multi-hop Systems. *IEEE Network*, 6(4):20–32, jul 1992.

[12] A. Sen and P. Maitra. A comparative study of Shuffle-Exhange, Manhattan Street and Supercube network for lightwave applications. *Computer Networks and ISDN Systems*, 26:1007–1022, 1994.

[13] A. Sen, A. Sengupta, and S. Bandyopadhyay. Generalized Supercube: An incrementally expandable interconnection network. *J Parallel Distrib. Comput.*, 13:338–344, 1991.

[14] K.N. Sivarajan and R. Ramaswami. Lightwave Networks Based on de Bruijn Graphs. *IEEE/ACM Transactions on Networking*, 2(1):70–79, apr 1994.

[15] T. Szymanski. Hypermeshes: Optical Interconnection Networks for Parallel Computing. *Journal of Parallel and Distributed Computing*, 26:1–23, 1995.