Optimizing Interconnections in Multihop Lightwave Networks: An Approach Using Genetic Algorithms

George Develekos and Christos Douligeris
Department of Informatics
University of Piraeus, Piraeus, Greece 18534.
email: christos,gdevelek@unipi.gr

ABSTRACT

This paper addresses the Virtual Topology Problem for a Dedicated Channel Wavelength Division Multiplexing Lightwave Network. Given the external traffic matrix, the number of network nodes and the number of receivers per node we develop a heuristic procedure based on the principles of genetic evolution that assigns wavelengths to the nodes in a way that, by tuning the electro-optic tranceivers to these wavelengths, connectivity can be achieved. Subsequently, the procedure routes the externally offered traffic over the links of this virtual topology. The performance metric that we seek to optimize through this iterative procedure is the maximum flow on any of the links of the network. We chose this objective function because it maximizes the traffic scale-up that the network can handle. Results on specific case-studies indicate that the proposed procedure can match or even ourperform conventional mathematical procedures.

1 Introduction

In this paper we present a way to harness the unique potential inherent in lightwave technology, as it applies to the field of multiuser, distributed packet networks. By virtue of its excellent transmission characteristics (low loss, low error rates, immunity to electromagnetic interference), the single mode fiber has emerged as the reigning transmission medium, replacing twisted pair copper wire and coaxial cable and supporting network architectures like Distributed Queue Dual Bus (DQDB) and Fiber Distributed Data Interface (FDDI). What remains to be exploited, however, is the enormous bandwidth (approximately 30THz), that is available in the low loss region 1.1μ to 1.5μ.

The reason is that this bandwidth is four to five orders of magnitude greater that the speed at which light can be modulated/demodulated by a continuous electronic signal. As a dramatic technological breakthrough in the field of electronics is not foreseen, we cannot expect to see all-optical lightwave networks in the near future. Hybrid networks, employing both electronics (in the user-network interface) and photonics (throughout the network), will need to implement some form of concurrency in order to tap the huge optical bandwidth of the medium, so that a number of distinguishable messages can be carried simultaneously, each one at the maximum rate that electronics can support, which is in the order of a few Gbits per second.

In an all-optical network environment, concurrency can be provided through a multitude of unique frequency channels (WDMA, Wavelength Division Multiple Access), non-overlapping time slots (TDMA, Time Division Multiple Access), or spread spectrum techniques (CDMA, Code Division Multiple Access). TDMA and CDMA seem less and less attractive, as they require all nodes to be synchronized within one time slot (for TDMA), or one chip time (for CDMA). WDMA gets most of the attention, as it involves existing and well-tested technologies and only requires the end user to operate at the bit rate of one channel, which can be chosen to be the maximum attainable electronic speed. In a WDMA network, each node is typically equipped with a small number of transmitters and receivers. Each receiver is assigned a unique

In **Singlehop** networks the information reaches the destination in a single hop, without passing through intermediate nodes. Therefore source and destination must be in precise dynamic coordination. This requires lasers (transmitters) and/or filters (receivers) that are rapidly tunable on a time scale consistent with packet switching (a transmitter-receiver pair staying in tune for the duration of a packet transmission - several microseconds - and then tuning to another wavelength in a matter of nanoseconds) over a large portion of the optical band. This is not nearly possible with the current state-of-the- art laser technology, so sophisticated protocols need to be developed to coordinate the data transmission. One proposed approach assigns a fixed wavelength to each transmitter. This puts the burden on the receiver to select from a pool of WDM signals, on a packet by packet basis, the correct wavelength to tune to at each point in time. Additionally, prior to the actual transmission, there has to be a signalling period during which the receiver is informed of the appropriate channel to tune to.

In **Multihop** networks wavelengths are assigned to each transmitter and receiver of each node, thus creating several independent channels that are multiplexed into the single optical medium; this assignment is rarely changed. As the number of transmitters and receivers per node is limited, there cannot be a unique wavelength assigned to each source-destination pair, so a message may have to hop through more than one intermediate node, on a different wavelength in each hop, in order to reach its destination. Clearly each node has to provide a store-and-forward type self-routing function so that connectivity can be obtained. This study focuses on multihop networks.

The fact that all signals are multiplexed into a single optical medium results in a relative independence between the logical interconnection pattern among nodes (*Virtual Topology*) and the physical layout of the fiber, the couplers, the amplifiers and the location of the nodes themselves (*Physical Topology*). The *virtual topology* is dictated by the optimization of the performance metric of the designer's choice, e.g. average packet delay, maximum throughput, link utilization etc. The *physical topology* is determined by cost constraints and cable-plant considerations. Alternatives include the star, bus and tree topologies.

Another key property of the multihop approach comes from deploying slowly tunable transmitters and receivers: the designer has the ability to adaptively optimize the logical node interconnection to prevailing non-uniform traffic patterns or to a network element failure. Such an update in the connectivity diagram needs to be done rather rarely (as opposed to on a packet-by-packet or a circuit-by-circuit basis), it therefore can be handled be a slowly tunable transceiver that is available with current technology.

Multihop virtual topologies can be *regular* or *irregular*. Regular topologies have simpler routing schemes because of their structured connectivity pattern, though their regularity constrains the number of nodes that can be accommodated to a limited discrete set of integers. Notable examples of regular structures are the perfect shuffle (*ShuffleNet* , the *de Bruijn graph*, the toroid *Manhattan Street Network*, the *Hypercube* and the *Linear Bus*). On the other hand, irregular topologies go further with regard to optimizing the objective function under consideration and are therefore amenable to non-uniform traffic patterns.

Another issue that needs to be resolved by the network designer is whether to employ *dedicated* or *shared* channels. In the first case each virtual link has its own dedicated wavelength channel. No multiple access schemes are required, but the utilization of the link might be low, especially if bursty traffic is expected. In the second case a channel can be accessed be two or more virtual links, thus necessitating some multiple access protocol and achieving higher link utilization.

This paper focuses on the solution to the Virtual Topology Problem for the case of the Dedicated Channel WDM Lightwave Network. Given N nodes, each of which has been allocated T transmitters and T receivers, and a traffic matrix representing the external traffic between all source-destination pairs, we want to develop a method that will assign wavelengths to the nodes in a way that, by tuning the electro-optic tranceivers to these wavelengths, connectivity will be achieved. Subsequently, the procedure will route the externally offered traffic over the links of this virtual topology. The performance metric, whose optimization drives this whole procedure, is the maximum flow on any of the $2NT$ links of the network. We chose this objective function because it maximizes the traffic scale-up that the network can handle.

The external traffic matrix is:

$$\Lambda = \{\lambda_{sd}\}$$

where λ_{sd} is the amount of external traffic in units of flow per second originating at node s and terminating at node d. Let

$$f_{ij}^{sd} = \text{traffic flowing on the link between nodes } i \text{ and } j, \text{due to the source-destination pair } (s,t)$$

These flow variables must satisfy the multicommodity flow constraint: for each source-destination pair and each node, the traffic flowing into the node balances the one flowing out of the node.

$$\sum_j f_{ij}^{sd} - \sum_j f_{ji}^{sd} = \begin{cases} \lambda_{sd}, & \text{if } s = i; \\ -\lambda_{sd}, & \text{if } d = i; \\ 0, & \text{otherwise} \end{cases} \qquad \forall i, s, d \tag{1}$$

With regard to the connectivity problem, we introduce the following variables that capture any connectivity pattern:

$$Z_{ij} = \begin{cases} 1, & \text{if there is a link from node } i \text{ to node } j \\ 0, & \text{otherwise.} \end{cases}$$

Note that we do not allow more than one directed link between two nodes. As mentioned before, a node may not have more than T transmitters and receivers allocated to it:

$$\sum_i Z_{ij} = T \qquad \forall j \tag{2}$$

$$\sum_j Z_{ij} = T \qquad \forall i \tag{3}$$

Traffic cannot flow from node i to node j unless there is a directed connection from i to j. This is the coupling constraint:

$$\sum_{s,t} f_{ij}^{sd} \leq M Z_{ij} \qquad \forall i, j \tag{4}$$

where M is a large constant, e.g., $M = \sum_s \sum_d \lambda_{sd}$. What we are looking to minimize is the maximum flow on any link, due to all the source-destination pairs, that is:

$$\text{minimize } \max_{\{i,j\}} \sum_{s,t} f_{ij}^{sd} \tag{5}$$

This performance measure does not take into account the delay, which is the sum of the propagation delay due to the distances between the nodes and the transmission delay which is due to the queueing delay at each node. It does, however, turn out that only the propagation delay remains unaccounted for, as previous studies have shown that, for highly loaded networks, this measure and the queuing delay measure have similar results. The reason is that, for highly loaded networks, the propagation delay is negligible compared to the queuing delay.

the dedicated channel structure. Shared channels can be considered, regardless of the access protocol used. Finally, no assumption has been made on the nature of the traffic: it can be packet switched or circuit switched.

The capacity C of the links of the network has not been taken into account so far. If, after we solve the problem, we see that the minimum value of objective function is larger than the capacity of the links of the network, we can only conclude that the external traffic cannot be accommodated by the current network.

The problem at hand, which is NP-hard, has been solved using a decomposition heuristic in [6], where the connectivity problem is solved first, looking for connectivity patterns that maximize the single-hop traffic. Then flows are assigned to the links that have been determined in the connectivity subproblem, so that the maximum flow on any link is minimized. Another approach involves the use of a popular heuristic called Simulated Annealing. We attempt to solve it using a **Genetic Algorithm**, a fast, robust method that has been gaining ground over the past years. It has so far been applied only to shared channel multihop networks, [4,5] with minimizing the maximum delay as an objective function.

3 The Genetic Algorithm Approach

Genetic Algorithms tackle complex problems by emulating the natural law of the "Survival of the Fittest". Potential solutions to the problem are mapped into a string representation called *Chromosomes* , which consists of a number of characters called *genes* , taking values from a set of *alleles*, that is usually the binary set {0,1}. A population of M initial solutions is generated randomly or by some heuristic, to create the first generation. Each subsequent generation is produced be applying a set of *Genetic Operators* to the members of the parent generation; these operators are *crossover, inversion and mutation.* A fixed percentage of chromosomes is generated by each operator.

Crossover, the most active operator, is a form of sexual reproduction. It constructs two new chromosomes by mating two parents that are selected according to their relative effectiveness in the current generation, their so called *fitness* . Mating is performed by breaking the two parent strings at the same randomly determined point and then producing the two offspring be swapping the substrings of the parents. This way above average schemata propagate to the new generation.

Inversion alters the positions of a number of bits in a string of a parent selected as explained above, by inverting the sequence of bits in a substring that has been randomly determined. The role of inversion is to provide a more fertile ground for recombination by crossover, by bringing together coadapted groups of genes. It is invoked rather rarely.

The last genetic operator, mutation, occurs by randomly changing a very small proportion of 1's to 0's and vice versa. Its role is to re-introduce features that may have been lost during the genetic evolution.

Genetic algorithms have been found to yield very good results faster than other heuristics like simulated annealing, by virtue of an intrinsic property called *implicit parallelism* , which identifies and preserves common components between strings having better than average performance. With a careful selection of parameters like the size of the population, the percentage of offspring that each operator generates and the fitness function, a synergistic balance can be achieved between *exploitation* of structures that have been found to be effective and *exploration* of new structures in the set of possible solutions, so that good results are obtained without the risk of premature convergence. Areas where genetic algorithms have seen success include the Travelling Salesman problem, the Iterated Prisoner's Dilemma, the Keyboard Configuration problem and the VLSI chip Layout problem.

4 The Proposed Solution

We apply the genetic approach in two characteristic cases: in the first the input flow matrix was very close to a ring-type traffic pattern, while in the second it was quasi-uniform.

$$Lambda_1 = \begin{pmatrix} 0 & 30 & 3 & 3 & 2 & 3 & 3 & 2 \\ 3 & 0 & 33 & 3 & 3 & 3 & 2 & 3 \\ 3 & 3 & 0 & 24 & 2 & 3 & 3 & 3 \\ 3 & 3 & 3 & 0 & 27 & 3 & 3 & 3 \\ 2 & 3 & 2 & 3 & 0 & 30 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 0 & 36 & 3 \\ 2 & 3 & 3 & 3 & 2 & 3 & 0 & 27 \\ 33 & 3 & 2 & 2 & 3 & 3 & 2 & 0 \end{pmatrix},$$

$$Lambda_2 = \begin{pmatrix} 0 & 11 & 10 & 9 & 9 & 10 & 8 & 7 \\ 8 & 0 & 7 & 10 & 11 & 10 & 9 & 9 \\ 9 & 10 & 0 & 11 & 12 & 8 & 9 & 8 \\ 11 & 11 & 10 & 0 & 7 & 8 & 9 & 9 \\ 9 & 10 & 11 & 7 & 0 & 8 & 8 & 10 \\ 10 & 8 & 8 & 9 & 10 & 0 & 10 & 10 \\ 11 & 10 & 9 & 9 & 8 & 7 & 0 & 9 \\ 8 & 8 & 10 & 10 & 11 & 9 & 9 & 0 \end{pmatrix}.$$

We represent the set of feasible connectivity patterns with an array Z_{ij} with dimension $N*N$, exactly as in the formulation of the problem, and the set of possible routing decisions with an array $FLOW_{ij}^{sd}$, which represents the flow on link {i,j} due to the source-destination pair {s,d}. The flow is represented in two ways: in binary form for the first problem, with N_b bits used for each flow $FLOW_{ij}^{sd}$, and in plain decimal form for the second problem.

An initial population of size 50 is generated randomly, for both the Z and the $FLOW$ arrays, taking into account that no more than T transmitters and receivers can be allocated to each node. In each subsequent generation, the algorithm goes through two phases: In the first phase the matrices Z and $FLOW$ that were generated during the previous generation are forced into compliance with the connectivity and multicommodity-flow constraints. This was deemed necessary because incorporating the constraints in the form of penalty factors into the fitness function did not provide good results, as the contraints are too complex to be reflected on a single number. Thus the first phase produces members that consitute possible solutions to the problem, with minimum deviation from the rules.

In the second phase a new generation is created with crossover or mutation; the inversion operator did not prove useful for this particular problem. The frequency at which the crossover was applied was initially set at 90%, but it was soon decreased to 80% to deter premature convergence.

The first phase proceeds in a number of steps:

- Transmitters and receivers are swapped between nodes, making sure that each node has exactly T of each.
- Flow showing in the $FLOW$ matrix on a link that does not exist in the Z matrix is eliminated.
- If there is a link in the Z matrix between the source and the destination, all the flow is loaded on that link, so as to maximize single-hop traffic.
- Flow that comes in and out of each node is reduced or increased appropriately, for all source-destination pairs, in an effort to minimize or eliminate the degree to which the multicommodity flow constraint is violated. The algorithm chooses to decrease flow on links that are heavily loaded due to previously examined source-destination pairs, so the full picture of the routing problem is brought into play, rather than the implications of the particular source-destination pair.
- Cycles of flow in the $FLOW$ matrix are identified and eliminated, and finally,

information flows along the path, according to the input matrix $\{lambda_{sd}\}$. Whenever flow needs to be pushed, this is done preferably to paths that are lightly loaded; the opposite applies to cases where flow needs to be pulled out. It should be noted that a path may consist of a number of branches.

At this point the fitness function is evaluated for each member. It is the sum of two factors: the maximum aggregate flow on any link of the network, and the amount by which the $FLOW$ matrix deviates from the connectivity and multicommodity flow constraints. The fittest member is then identified and introduced to the new generation as the first member, so that the monotonicity of the algorithm is maintained.

Entering the second phase, one of the crossover or the mutation operators is selected, with a strong bias towards the former. In case of crossover, two members/parents are selected, based on their fitness, and mated to produce two offspring. Mating is achieved by splitting each member at a random point and them swapping parts between parents.

Mutation was implemented in two ways. During the first generations a chromosome was picked in each line of the Z and $FLOW$ matrices of a randomly selected member and mutated. Towards the end of the algorithm, the first and so far fittest member was selected and the flow on the heaviest loaded links was mutated, in the hope that rearranging performed during the subsequent generation would bring some improvement.

5 Results

The two matrices that follow depict the results obtained for the two cases considered. For the ringtype traffic the algorithm converged on the 156th generation, yielding the value 47 as the maximum flow on any link:

$$Solution_1 = \begin{pmatrix} 0 & 39 & 43 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 45 & 0 & 0 & 42 & 0 & 0 \\ 0 & 0 & 0 & 42 & 0 & 0 & 0 & 38 \\ 0 & 46 & 0 & 0 & 41 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 44 & 46 & 0 \\ 0 & 0 & 0 & 0 & 45 & 0 & 47 & 0 \\ 41 & 0 & 0 & 0 & 0 & 0 & 0 & 43 \\ 44 & 0 & 0 & 41 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

For the quasiuniform case the algorithm reached its optimum point, 72, after 185 generations.

$$Solution_2 = \begin{pmatrix} 0 & 72 & 0 & 64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 69 & 0 & 0 & 69 & 0 & 0 \\ 68 & 0 & 0 & 0 & 72 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 66 & 67 \\ 0 & 70 & 0 & 69 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 72 & 63 \\ 70 & 0 & 69 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 72 & 61 & 0 & 0 \end{pmatrix}.$$

Results were compared with solutions obtained in the literature for uniform, quasi-uniform, and various types of non-uniform traffic patterns, in terms of optimality and speed of convergence.

Upon comparison of these results with results obtained via other conventional or heuristic methods in the literature one can draw the following conclusions:

- The genetic approach produces solutions that are slightly better or slightly worse than other established methods.

that genetic algorithms are inherently very good candidates for running on architectures/compilers that support parallelism, due to their parallel nature. For instance, the first phase of the algorithm, the fixing phase, could execute concurrently on as many processors as the size of the population, by eliminating the very few portions of it that introduce some interdependency between optimizing modules.

- The genetic approach does not require that the programmer develop code that will deterministically guide the algorithm to the optimum solution. It only needs to be pushed to the right direction. It then finds its own way to the target. It has been confirmed in practice, as a matter of fact, that it can do so even when it has to tackle problems that have been concocted with the purpose to misguide it away from the global optimum. It does, however, require a good deal of fine tuning, that is dependent on the particular case, before it will perform to its ability. Hundreds of runs had to be executed, in our case, to produde good results as we went from the ring-type traffic to the quasi-uniform traffic. It is very possible that better results can be obtained by playing with the numerous parameters that control the direction the algorithm takes and their innumerous combinations.

6 REFERENCES

1. Lawrence Davis, "Genetic Algorithms and Simulated Annealing", Morgan Kaufmann Publishers, Inc., Los Altos, California.

2. Anthony S. Acampora, Mark J. Karol, "An Overview of Lightwave Packet Networks", IEEE Network, January 1989, pp. 29-40.

3. Biswanath Mukherjee, "WDM-Based Local Lightwave Networks, Part II: Multihop Systems", IEEE Network, July 1992, pp. 20-32.

4. J. Bannister, L. Fratta, M. Gerla, "Topological Design of the Wavelength-Division Optical Network", IEEE INFOCOM 1990, pp. 1005-1013.

5. J. Bannister "The Wavelength-Division Optical Network: Architectures, Topologies and Protocols", Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, California, March 1990.

6. Jean-Francois P. Labourdette and Anthony S. Acampora, "Wavelength Agility in Multihop Lightwave Networks", IEEE INFOCOM 1990, pp.1022-1029.