Solving the WDM Network Operation Problem Using Dynamic Synchronous Parallel Simulated Annealing

Asheq Khan
Dept. of Computer Science & Engineering
State University of New York at Buffalo
akhan6@cse.buffalo.edu

Dale R. Thompson
Dept. of Computer Science & Engineering
University of Arkansas at Fayetteville
d.r.thompson@ieee.org

Abstract

Several variations of synchronous parallel simulated annealing (PSA) were applied to solve the static lightpath establishment wavelength selective cross-connect network operation problem for a WDM network. The goal was to find high-quality solutions to determine the efficiency of the dynamic routing and wavelength assignment algorithms. Multiple parallel processes ran the simulated annealing algorithm and exchanged solutions among them. A proposed dynamic PSA is presented that dynamically varied the simulated annealing parameters based on a success rate. We demonstrate that the proposed dynamic PSA algorithm obtains less-cost solutions than the other PSA algorithms and does not incur any communication over heads.

1. Introduction

Wavelength division multiplexing (WDM) is a technology that allows multiple digital signals to be carried over a single optical fiber by carrying each signal on a different wavelength [1]. If a WDM network routes the data from the source to the destination using wavelengths, then it is known as a wavelength-routed network. Every node is connected to cross-connects. One type of cross-connect is known as the wavelength selective cross-connect (WSXC) that does not perform wavelength conversion. Therefore, to transfer a signal from a source node to a destination node, the same wavelength must be present on all the links of the path between the two nodes. If a path can be established, then it is called a *lightpath* [2]. The problem of establishing a lightpath or optical transmission channel between every source and destination node is known as the static lightpath establishment (SLE) WSXC network operation problem.

The SLE WSXC network operation problem is a non-deterministic polynomial (NP) combinatorial problem with several local minima and there is no known algorithm that can find the solution in polynomial time [2]. Therefore, approximate heuristic methods have been used to solve the SLE problem by formulating it as an approximate integer programming (IP) problem [3]. With approximate IP, the problem is solved by either maximizing or minimizing an objective function. Some of the heuristic methods that have been applied to solve the SLE problem are the longer paths first policy [3], fixed shortest path routing [4], simulated annealing [5], and genetic algorithm [6].

Simulated annealing (SA) is a problem independent heuristic approach for solving combinatorial minimization or maximization problems [7]. It has been used to solve combinatorial problems [8-12]. Simulated annealing algorithm works by randomly searching a large solution space until it finds the best solution. The quality of a solution is determined by an objective function that assigns a value to each solution. Depending on the type of problem, a higher or lower value of the objective function determines the quality of a solution. In this paper, a minimization problem is assumed. The probabilistic hill climbing property of SA enables it to climb out of a local minimum and converge to a global solution. The major drawback of simulated annealing is the large amount of computational time it requires. To decrease the computational time, parallel simulated annealing (PSA) can be used.

The goal of the paper is to solve the SLE WSXC network operation problem of a WDM network by applying PSA. Provided with the complete physical topology and the traffic demand for the network, the objective is to apply PSA to assign wavelengths to each source-destination pair such that the overall wavelength cost is minimized. Different PSA algorithms are compared. A proposed dynamic PSA algorithm is introduced, which decreases the computational requirements and finds lower-cost solutions. It measures on-line statistics to determine when to lower the temperature and when to terminate. Unlike previous approaches, the mutation rate and the stopping condition of the algorithm are varied dynamically while searching for a solution. A significantly better final solution is obtained with less overhead.

This paper is organized as follows. In section 2, the simulated annealing algorithm and its application in solving the SLE WSXC network problem are discussed. The problem formulation and algorithm are presented in section 3. In section 4, the results of the different PSA algorithms and the proposed dynamic PSA are provided. Finally, a conclusion is provided in section 5.

0-7803-8865-8/05/$20.00 ©2005 IEEE.

2. Background

A WDM network is faced with two problems [3]. The first problem is known as the *network design problem*. In this problem, given a general topology of the network, the problem is to find a configuration for the network in terms of the number of fibers needed, the placement of optical amplifiers and cross-connects, and the size of the cross-connects. The objective of the network design problem is to provide a solution that minimizes the cost of the network. The second problem is the *network operation problem*. Given a complete network topology and the traffic demand, the problem is to find the routing and the wavelength assignment for each route. The goal of the network operation problem is to minimize the blocking probability, maximize the carried traffic, or minimize the overall cost of the network. In this work, we focus on the network operation problem.

The static lightpath establishment (SLE) problem is a non-deterministic polynomial (NP) combinatorial problem with several local minima that can be solved with approximate heuristic methods based on integer programming [3]. Another such heuristic is the simulated annealing (SA) algorithm. SA consists of five components [13]. First, a complete description of the system's configuration is required. Second, a disturbance mechanism makes random changes to the configuration. In this paper, the disturbance mechanism is also referred to as *mutation*. Third, an objective function is defined that SA uses to minimize or maximize. A minimization problem is assumed in this paper. Fourth, the number of iterations or amount of time spent at each temperature is fixed or dynamically determined. Finally, SA requires an annealing schedule for lowering the initial temperature until the final temperature or the stopping point is reached. SA for minimization is shown in Fig. 1.

$T_{current} \leftarrow T_{initial}$
$Solution_{current} \leftarrow Solution_{initial}$
$Cost_{current} \leftarrow$ **Evaluate**($Solution_{current}$)
while $T_{current} > T_{final}$
  repeat N times
    $Solution_{new} \leftarrow$ **Random_disturbance** ($Solution_{current}$)
    $Cost_{new} \leftarrow$ **Evaluate**($Solution_{new}$)
    if ($Cost_{new} - Cost_{current} < 0$) OR (**Random**( ) < $\exp((Cost_{current} - Cost_{new}) / T_{current})$) *then*
      $Solution_{current} \leftarrow Solution_{new}$
      $Cost_{current} \leftarrow Cost_{new}$
    endif
  endrepeat
  $T_{current} \leftarrow$ **Lower_temp**($T_{current}$)
endwhile

Fig. 1 The serial simulated annealing algorithm.

An advantage of SA is that it converges to a near-optimal solution provided the initial temperature is decreased slowly and an adequate amount of time is spent at each temperature. However, SA requires a large amount of computational time. Therefore, parallel simulated annealing (PSA) algorithm is often used to run multiple SA processes in parallel.

In [5], a PSA algorithm is applied to assign wavelengths to each source-to-destination traffic demand in a WSXC WDM network such that the mean packet delay is minimized. With PSA, the computation of all the shortest paths for a configuration is distributed among the concurrent processes. Upon completion of the evaluation of shortest paths, processes communicate to determine the total cost of the configuration. This parallel implementation is known as single Markov chain (SMC) PSA because all processes follow a single search path or Markov chain [12]. SMC PSA is inefficient because it requires frequent communication and is confined to a single search space. Therefore, in this paper, multiple Markov chain (MMC) PSA has been implemented to allow each process to search different Markov chains in parallel and achieve better parallelism and efficiency.

3. Problem Formulation

The objective function could be to minimize the number of wavelengths required to satisfy all the traffic demands, to maximize the carried traffic of a network [14], to minimize the number of optical fibers [15], or to minimize the total wavelength cost. In this work, the wavelength cost is minimized. The physical topology was generated randomly and each link was assigned a cost based on distance. The cost of a single source-to-destination traffic demand is the sum of the link costs traversed by a wavelength. PSA assigns wavelengths for each source-to-destination traffic demand in a WDM network such that the sum of the link costs is minimized.

3.1. The Objective Function

The minimization of the total wavelength cost is a variation of the maximum carried traffic flow objective function in [14]. The *k-shortest path* algorithm is applied to find the shortest, the second shortest up to the *k*-shortest path for each pair of source-destination node. Then, the wavelengths along the route are assigned using the *first-fit heuristic* [2]. In the first-fit heuristic, the wavelengths on a link are indexed from zero to *n*-1, where *n* is the maximum number of wavelengths. A wavelength is assigned to a route by searching in ascending order through the index of wavelengths until it finds the wavelength that can be assigned to all the links of the route.

3.2. The Algorithm

The solution to the SLE WSXC network operation problem is represented as a two-dimensional array. Each column of the array represents the routes for a particular traffic parcel. A *traffic parcel* is defined as the number of wavelengths required between each source and destination node in a WDM network [16]. The first row of the array represents the primary route and the second row represents the backup route. Each element of the array has a value of 0 to k-1 representing the shortest route up to the k-shortest route between a pair of nodes. The number of columns of the array represents the total number of traffic parcels in the network. For the experiments, k = 3. Fig. 2 represents an array with ten traffic parcels and three shortest routes, indexed from 0 to 2.

```
1 0 1 0 2 1 2 1 0 2
2 0 0 2 1 0 1 1 2 1
```

Fig. 2 An array representing the routing assignment for ten traffic demands.

The algorithm for solving the SLE WSXC network operation problem is presented in Fig. 3. The array of routes is initialized with random values between 0 and 2 because up to three shortest routes are considered. The temperature is set to its initial value. Then, the objective function is called to generate the cost of the initial solution. Multiple processes of serial simulated annealing are then applied to the initial solutions and solutions may or may not be exchanged periodically. At the end of the annealing schedule, $\text{Array}_{\text{solution}}$ contains the solution and $\text{Cost}_{\text{current}}$ has the cost of the solution.

1. $\text{Array}_{\text{solution}} \leftarrow$ **Random**(0 to k-1)
2. $T_{\text{current}} \leftarrow T_{\text{initial}}$
3. $\text{Cost}_{\text{current}} \leftarrow$ **Objective**($\text{Array}_{\text{solution}}$)
4. *while* $T_{\text{current}} > T_{\text{final}}$
5. Apply Serial Simulated Annealing
6. Exchange solutions
7. endwhile
8. **Report** the final configuration and its cost

Fig. 3 The algorithm for solving the SLE WSXC network operation problem.

3.3. Backup Route and Penalty

The algorithm maintains a backup wavelength for each wavelength [16]. But, the backup wavelength is assigned on an alternate route. The objective function penalizes a backup wavelength that is assigned on the same route as the primary wavelength by raising the route cost to the power 1.5. This is done to take into account the situation where a node does not have an alternate route to another node. The objective function also penalizes a link by raising the link cost to the power 1.5 when an assigned wavelength exceeds the maximum number of wavelengths on a link. With the two penalties, a better performance is achieved rather than simply rejecting these configurations.

3.4. Re-scaling the Traffic Parcels

The value for each traffic parcel was randomly generated with values between 1 and 14 such that the sum of the wavelengths, λ_T, exceeds the maximum number of wavelengths, λ_{max} that can be carried by the physical topology. The value of λ_{max} is estimated by multiplying the number of links in the topology by the maximum number of wavelengths that can be carried by a link. If λ_T exceeds λ_{max} then it is scaled down to 50% of λ_{max}. This is done by multiplying each traffic parcel by (.5) x (λ_{max} / λ_T) / (*Num_Routes*), where *Num_Routes* represents the number of alternative routes for each traffic parcel. At 50% of λ_{max}, only about 50% of the wavelengths can be assigned with the new scaled offered traffic. This provides PSA with the challenging task of assigning wavelengths for each route.

4. Results

The parallel simulated annealing (PSA) algorithms were implemented on a 32-node cluster computer. A WDM network comprised of 25 nodes with average node degree of 3.0 was generated with *gt-itm* using the Waxman model [17]. The offered traffic was 600 pairs (25 x (25-1)) of source-destination traffic parcels. A two-dimensional array comprising of 2 rows and 600 columns represented a solution. Message passing interface (MPI) was used to run multiple simulated annealing (SA) processes in parallel and the best solution among them was chosen.

The best cooling coefficient, the mutation rate, the number of iterations per temperature, and the initial and final temperatures were determined empirically. The best solution or minimum cost was achieved for a cooling coefficient of 0.995, mutation rate of 1% or 12 cells, 20 iterations per temperature, and 1.08 x 10^7 and 22 for the initial and final temperatures, respectively. Unless specified otherwise, the default values in the following algorithms were set to these values.

4.1. PSA with Variable Cooling Coefficients

PSA was run with three different cooling coefficients to determine the affect on the quality of the final solution. The values for the cooling coefficient were set to 0.80, 0.90, and 0.995 to allow the temperatures to decrease at variable rates while producing a good solution. Each SA process ran for an arbitrary 4,500 iterations. A fixed number of iterations was chosen to allow each cooling coefficient to execute for the same amount of time. The final costs for the cooling coefficients equal to 0.80, 0.90, and 0.995, were 1.17903×10^6, 1.18882×10^6, and 1.18135×10^6, respectively.

4.2. PSA with Fixed Intervals of Communication

Next, one of the processes was selected as the master and the other processes were the slaves. After a fixed number of iterations, each slave process sent its present cost to the master process. The master process then compared the cost of all the processes including its own and broadcasted the process id of the least-cost solution to all processes. Then the process with the minimum cost broadcasted its present cost and solution to all the processes. At that point, each process resumed its computation but with the new cost and solution. In this algorithm, each process had to wait for the other processes to reach a communication interval before continuing. This algorithm is named PSA with fixed intervals of communication.

Each SA process ran for 2,615 iterations to reach the final temperature of 22. The values for the intervals of inter-process communication were set to 100, 300, and 800 iterations to allow 26, 8 and 3 inter-process communications, respectively. The final costs for the intervals of 100, 300, and 800 iterations, were 1.22147×10^6, 1.22349×10^6 and 1.23351×10^6, respectively. Although a communication interval of 100 produced a slightly lower cost, it also incurred the highest communication overhead.

4.3. PSA with Variable Intervals of Communication

PSA with fixed intervals was modified to have variable intervals of communication. In PSA with variable intervals of communication, the processes communicated only when two conditions were satisfied. First, each process had to run for at least a fixed number of iterations. Second, the *acceptance rate* of the process had to equal or fall below a threshold. The *acceptance rate* for a temperature was calculated as the ratio of the number of new solutions accepted to the total number of iterations performed per temperature.

When both conditions were satisfied, a process communicated with the other processes. After the communication, the fixed iteration counter was reset to 0. Unlike PSA with fixed intervals of communication, PSA with variable intervals of communication only communicated when its acceptance rate was less than or equal to a threshold, thus eliminating any unnecessary communication. To test this algorithm, we chose a fixed interval of 300 iterations to allow for enough inter-process communications and acceptance rates of 0.0, 0.30, and 0.60. Therefore, a process communicated when it had performed at least 300 iterations and its acceptance rate was less than or equal to the set acceptance rate. Each SA process ran for 2,615 iterations.

The final costs for the acceptance rate equal to 0.0, 0.30, and 0.60, were 1.20720×10^6, 1.21644×10^6, and 1.22018×10^6, respectively. An acceptance rate of 0.0 achieved the minimum cost, and required only 6 inter-process communications.

4.4. PSA with Different Initial Temperatures

Another PSA algorithm, called PSA with different initial temperatures, assigned different initial temperatures to each of the processes. The initial temperatures of each process was set by multiplying the original initial temperature, 1.08×10^7, by a constant called the initial temperature factor, a, which is calculated using the following formula

$$a = \left\{ \exp\left[\ln\left(\frac{t_{final}}{t_{initial}} \right) \Big/ (n-1) \right] \right\}^{rank}$$

where t_{final} is the final temperature, $t_{initial}$ is the initial temperature, n is the number of processes, and *rank* is the process rank, which is between 0 and $n - 1$.

After a fixed number of iterations, a process sent its present cost and solution to its neighboring process that started at a *lower* $t_{initial}$. Then each process compared its own cost with the one received from its neighboring process. If a process's own cost was higher than its neighbor's, the process accepted its neighbor's cost and solution. Otherwise, the process kept its own cost and solution.

PSA with different initial temperatures was tested with 4, 8 and 16 processes. The interval for inter-process communication was fixed at 300 iterations to allow adequate number of inter-process communications and each process ran for 2,615 iterations. The final costs for 4, 8, and 16 processes, were 1.20568×10^6, 1.20015×10^6, and 1.19128×10^6, respectively.

Then, the process size was fixed at 4, and the number of iterations between communications was varied. The values of the number of iterations between inter-process communications were set to 100, 300, and 800. The final costs for 100, 300, and 800 iterations, were 1.22297×10^6, 1.20568×10^6, and 1.21703×10^6, respectively.

4.5. Dynamic PSA Scheme

The three PSA algorithms presented in sections 4.2-4.4 have achieved similar minimum cost at the expense of inter-process communication overheads. Therefore, we wanted to explore a possible improvement in the solution. We had two objectives in mind. First, obtain a better final solution with less cost. Second, achieve that minimum cost with fewer or no inter-process communication. To this extent, we propose a new dynamic PSA scheme.

In the dynamic PSA algorithm, each SA process remained at the present temperature until a required number of successes occurred or the number of iterations per temperature exceeded a fixed value. A *success* was defined as accepting a new solution. If no success was achieved for a fixed number of consecutive temperature changes, the mutation rate was decreased and the cycle was repeated. We have experimented with initial mutation rates of 1% (12 cells), 2.5% (30 cells), and 5% (60 cells). The number of failures tolerated for consecutive temperature changes was set to 5 for the starting mutation rate of 12, 30, or 60 cells and increased by 5 for each change in the mutation rate. The mutation rates were changed such that, 50 failures were tolerated for the last mutation rate. As a result, the mutation rates of 12, 30, and 60 cells were decreased by 1, 3, and 6 cells respectively each time to ensure that 50 failures were tolerated for the last mutation rate of 3, 3, and 6 cells respectively.

The intuition behind this approach is that at high temperatures most new solutions are accepted whereas, at low temperatures almost none are accepted [10]. Therefore, at high temperatures, the probability of achieving no success for consecutive temperatures was lower. On the other hand, at lower temperatures, the probability of achieving no success for consecutive temperatures is higher. In terms of mutation, by mutating more cells at higher temperatures, each SA process can randomly search for a better solution. But, at lower temperatures, fewer cells need to be mutated as each SA process converges to a near optimum solution. A process terminates when the number of cells mutated decreased to zero.

The dynamic PSA algorithm was compared with the PSA scheme that used a fixed mutation rate and had no inter-process communication. For each of the starting mutation rate, the dynamic PSA scheme was first executed until it terminated and the number of iterations of temperature changes it completed before termination was noted. Then the fixed mutation PSA scheme was executed for the same number of iterations for a fair comparison. In both PSA schemes, each SA process started with the same initial temperature and remained at this temperature until either 5 successes was met or the number of iterations at the temperature had exceeded 20. The best solution among the four independent parallel SA processes was chosen and is shown in Table 1. As seen in Table 1, the dynamic PSA scheme outperforms the fixed mutation scheme with an improvement of 6%, 22% 41% for starting mutation rates of 12, 30 and 60 cells respectively.

Table 1. Minimum cost achieved for different initial mutation rate.

	12 cells	**30 cells**	**60 cells**
Dynamic PSA	1.108×10^6	1.114×10^6	1.135×10^6
PSA with fixed mutation rate and no communication	1.173×10^6	1.143×10^6	1.926×10^6

4.6. Comparison of the PSA Schemes

Finally, we compare the Dynamic PSA with the PSA with fixed intervals of communication, PSA with variable intervals of communication, PSA with different initial temperatures and the PSA with fixed mutation rate and no communication. Since the Dynamic PSA with starting mutation rate of 12 cells ran for 4,900 iterations, the other PSA schemes were run for 4,900 iterations for a fair comparison and the results are in Table 2. The proposed Dynamic PSA performed an average of 6% better than the other PSA algorithms and did not incur any communication overhead. Among the synchronous PSA algorithms, the PSA with variable intervals of communications performed the best in terms of providing both a lower final cost and the least communication overhead.

5. Conclusions

The static lightpath establishment wavelength selective cross-connects network operation problem for a WDM network was solved using several variants of the parallel simulated annealing algorithm. Five different synchronous PSA schemes were implemented to compare their performances. An unexpected result was that more communication among the processes sometimes increased the cost of the final solution. Finally, a dynamic PSA that varied the mutation rate and the number of iterations at a temperature based on on-line measurements was introduced.

The dynamic PSA scheme outperforms all other PSA schemes and does not incur any communication overhead. The uniqueness of our scheme compared to the previous approach [5] is as follows. First, we attempted to minimize the total wavelength cost whereas previously the mean packet delay was minimized. Second, our parallel scheme was based on multiple Markov chain (MMC) unlike the single Markov chain (SMC) PSA scheme used previously. With MMC PSA, each process could search different Markov chains or search space in parallel hence, a better parallelism and efficiency was achieved. Finally, unlike SMC PSA that required frequent communication, the proposed dynamic PSA scheme did not require any inter-process communication. We also took the additional step to compare our scheme with four other MMC PSA schemes.

The results show that our scheme achieves the best solution and incurs zero communication overhead.

Table 2. Comparing the PSA schemes at 4,900 Iterations.

PSA Scheme	Minimum Cost Achieved	No. of Communication
Dynamic PSA	1.10751×10^6	0
PSA with variable intervals of communication	1.16337×10^6	12
PSA with different initial temperatures	1.16769×10^6	16
PSA with fixed mutation rate and no communication	1.17275×10^6	0
PSA with fixed intervals of communication	1.17889×10^6	16

6. References

[1] B. Ramamurthy, B. Mukherjee, "Wavelength Conversion in WDM Networking," *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 7, pp. 1061 - 1073, 1998.

[2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WAN's," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171-1182, 1992.

[3] E. Karasan and E. Ayanoglu, "Performance of WDM Transport Networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1081 – 1096, 1998.

[4] E. Karasan and E. Ayanoglu, "Effects of Wavelength Routing and Selection Algorithms on Wavelength Conversion Gain in WDM Optical Networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 186 – 196, 1998.

[5] J. A. Bannister, L. Fratta, and M. Gerla, "Designing Metropolitan Area Networks for High-performance Applications," *Computer Networks and ISDN Systems*, vol. 20, pp. 223-230, 1990.

[6] D. Saha, M. D. Purkayastha, and A. Mukherjee, "An Approach to Wide Area WDM Optical Network Design using Genetic Algorithm," *Computer Communications*, vol. 22, pp. 156-172, 1999.

[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[8] M. Malek, M. Guruswamy, and M. Pandya, "Serial and Parallel Simulated Annealing and Tabu Search Algorithm for the Traveling Salesman Problem," *Annals of Operation Research*, vol. 21, pp. 59-84, 1989.

[9] P. Banerjee, M. H. Jones, and J. Sargent, "Parallel Simulated Annealing Algorithm for Cell Placement in Hypercube Multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, pp. 91-106, 1990.

[10] E. E. Witte, R. D. Chamberlain, M. A. Franklin, "Parallel Simulated Annealing Using Speculative Computation," *IEEE Transactions on Parallel and Distributed Systems,* vol. 2, no. 4, pp. 483 – 494, 1991.

[11] R. Diekmann, R. Luling, B. Monien, and C. Spraner, "A Parallel Local Search Algorithm for the K-Partitioning Problem," in *Proceedings of the 28th Hawaii International Conference on System Sciences*, vol. 2, pp. 41-50, 1995.

[12] S. Y. Lee and K. G. Lee, "Synchronous and Asynchronous Parallel Simulated Annealing with Multiple Markov Chain," *IEEE Transactions on Parallel and Distributed Systems,* vol. 7, no. 10, pp. 993 – 1008, 1996.

[13] Parallel Simulated Annealing. Available at *http://www.jics.utk.edu/PCUE/MOD11_APP/sld001.html*

[14] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489-500, 1995.

[15] N. Nagatsu, S. Okamoto, and K. Sato, "Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 893–902, 1996.

[16] D. R. Thompson and M. T. Anwar, "Parallel recombinative simulated annealing for wavelength division multiplexing," in *Proceedings of the 2003 International Conference on Communications in Computing*, pp. 212-217, 2003.

[17] E. Zegura, K. Calvert, and M. Donahoo, "A quantitative comparison of graph-based models for internet topology," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770-783, 1997.