클로즈 근사화를 이용한 등가 라우팅 알고리즘의 설계

이 봉 환†

요 약

본 논문에서는 컴퓨터 네트워크의 설계에 유용한 등가 라우팅 알고리즘(Equal-cost Bifurcated Routing Algorithm)을 제안하였다. 이 제안한 알고리즘의 성능은 기존의 몬테카를로 시뮬레이션 및 비정상 큐잉 근사화(Transient queueing approximation)를 이용하여 비교되었으며 그 결과 큐잉 근사화는 몬테카를로 시뮬레이션에 상당히 근접한 결과를 제공하였다. 또한, 큐잉 근사화는 몬테카를로 시뮬레이션에 비하여 매우 적은 수행시간을 요구하므로 제안한 등가 라우팅 알고리즘은 대부분의 경우에 우수한 결과를 제공하였다.

Design of Equal-Cost Bifurcated Routing Algorithm : A Case Study Using Closure Approximation

Bong Hwan Lee†

ABSTRACT

In this paper, we propose an equal-cost bifurcated routing algorithm which may be useful in practical computer network design problem. The performance of the routing algorithm is evaluated using the conventional Monte Carlo simulation and a transient queueing approximation. The relative errors between the closure approximation and the Monte Carlo simulation was fairly small. The closure approximation may be used to evaluate the performance of the load splitting algorithms, which results in considerable execution time reduction. The performance of the proposed algorithm is compared to that of the known algorithms based on average packet delay. For networks that have many non-disjoint equal-paths, the proposed algorithm performed better than other algorithms.

1. Introduction

Transient queueing approximations have been used in order to provide transient queue statistics comparable to the results from conventional Monte Carlo simulations. The closure approximation of the M/M/1 queueing system is extended to the general Jackson network to obtain transient queue statistics. Approximations are necessary since a closed form solution for the transient behavior of a network of queues remains an intractable problem. In this paper, the performance of the closure approximation has been evaluated. The routing protocols considered in this study are link state routing algorithms such as the ISO IS-IS(Intermediate System-Intermediate System) and OSPF(Open Shortest Path First) routing protocols [1, 2].

One of the new features of the ISO IS-IS and OSPF routing protocols is load splitting. If there exists multiple equal-cost routes for a given destination, the protocol discovers all the equal-cost routes and distributes incomming traffic over them. This traffic

† 정 회 원 : 한국통신 통신망연구소
논문접수 : 1994년 8월 22일, 심사완료 : 1994년 9월 24일

distribution feature is not available in IGPs (Interior Gateway Protocols) like RIP[3], where a single route is chosen for each destination. The algorithms by which traffic is distributed over the equal-cost routes is not specified in the protocols;it is dependent upon protocol implementation. In this paper, we consider three different load splitting algorithms including the proposed one. The performance of the algorithms is compared using the closure approximation to ensure the usefulness of the approximation in a practical network design problem.

In Section 2, we describe the closure approximation. The third section introduces the bifurcated equal-cost routing algorithms. In Section 4, we present the test cases and performance criteria. The fifth section presents simulation results. The final section summarizes the results presented.

2. Closure approximation

We consider a closure approximation based on a Jackson network model to estimate transient behavior of computer networks. The definition of Jackson network is described in [4]. Jackson's theorem says that the network acts as if each node is an independent M/M/1 queue. In the theorem, it is shown that the joint probability distribution of the number of customers in the system can be written as a product of marginal M/M/1 probability distribution only in steady state condition. Recently, a set of differential equations for transient queue statistcs in a Jackson network have been derived[5, 6]. The differential equations for the Jackson network transient mean at queue i, $n_i(t)$, is given by

$$\frac{d}{dt}E[n_i(t)] = \lambda_i(t)+\sum_{j\neq i}\theta_{ji(t)}(1-p_{j0}(t)) -(1-\theta_{ii}(t))\mu_i(t)(1-p_{i0}(t)), \quad (1)$$

where $\lambda_i(t)$ represents the mean Poisson external arrival rate and $\mu_i(t)$represents the exponential mean service rsate at queue i, respectively. In (1), $\theta_{ij}(t)$ is the transition probability from queue i to queue j. The differential equation for the second moment of the number of customer at queue i is given by

$$\begin{aligned}\frac{d}{dt}E[n_i^2(t)] = {} & \lambda_i(t)+\lambda_i(t)\cdot 2E[n_i(t)]+\\ & \sum_{j\neq i}\theta_{ji}(t)\mu_j(t)(1-p_{j0}(t))\\ & +2\sum_{j\neq i}\theta_{ji}(t)\mu_j(t)\cdot\{E[n_i(t)]\\ & -E[n_i(t)\,|n_j(t)=0]\cdot p_{j0}(t)\}\\ & +(1-\theta_{ii}(t))\mu_i(t)(1-p_{i0}(t))\\ & -2(1-\theta_{ii}(t))\mu_i(t)E[n_i(t)].\end{aligned} \quad (2)$$

Note that there exists only one unknown variable,$p_{io}(t)$, in each differential equation. In other words, the first and second moments can be obtained once the idle probability, $p_{io}(t)$, is determined. This quantity can be approximated using the closure approximation methods. A number of closure approximations have been proposed in the literature[7, 8, 9, 10]. In the closure approximation, an assumption, known as the closure assumption, is made to reduce the infinite set of equations to some tractable number. With the closure approximation, the variables in the system can be represented in terms of a smaller number of variables. For example, the idle probability, $p_{io}(t)$, is often approximated in terms of mean of the number of customers in the system or mean and variance both.

The closure approximationm, therefore,

provides estimates for the mean and variance in an efficient manner. In this paper, we employ Clark's closure approximation since it was shown that Clark's approximation performed better than other approximations on most cases[11, 12]. The Jackson network approximations presented here are derived from closure approximations for the M/M/1 queue. Clark's approximation uses the Polya-Eggenberger distribution as a surrogate distribution for the M/M/1 state probabilities.

To represent the number of customers in queue i, two conditional random variables, A_i and B_i, are defined as follows.

$$A_i = n_i(t), \quad n_i(t) \le 1$$
$$B_i = n_i(t), \quad n_i(t) > 1$$

Since the conditional random variables A_i and B_i represent $n_i(t)$, their moments can be determined using the following parameters.

$$Q_i(t) = \Pr\{\text{ a queue exists at time t}\} = \Pr\{n_i(t) > 1\} \quad (3)$$

$$\overline{Q_i}(t) = \Pr\{\text{ no queue exists at time t}\} = p_{i0}(t)+p_{i1}(t) \quad (4)$$

$$C_i(t) = \Pr\{\text{ the server is busy}\}=1-p_{i0}(t) \quad (5)$$

If we assume the random variables $n_i(t)$ and $n_j(t)$ are independent, (1) and (2) can be written as (6) and (7), respectively.

$$\frac{d}{dt}E[n_i(t)] = \lambda_i(t)+\sum_{j\neq i}\theta_{ji}(t)\mu_j(t)C_j(t) -(1-\theta_{ii}(t))\mu_i(t)C_i(t) \quad (6)$$

$$\frac{d}{dt}E[n_i^2(t)] = \lambda_i(t)+\lambda_i(t)\cdot 2E[n_i(t)] + \sum_{j\neq i}\theta_{ji}(t)\mu_j(t)C_j(t) +2\sum_{j\neq i}\theta_{ji}(t)\mu_j(t)E[n_i(t)]C_j(t) +(1-\theta_{ii}(t))\mu_i(t)C_i(t) -2(1-\theta_{ii}(t))\mu_i(t)E[n_i(t)] \quad (7)$$

Using the derivatives for $p_{i0}(t)$ and $p_{i1}(t)$ from the general Kolmogorov differential-difference equations, the resulting differential equations for $\overline{Q_i}(t)$ and $C_i(t)$ are given by [13]

$$\frac{d}{dt}\overline{Q_i}(t) = \mu_i(t)(1-\theta_{ii}(t))p_{i2}(t) -\lambda_i(t)p_{i1}(t) - \sum_{j\neq i}\mu_j(t)\theta_{ji}(t)p_{i1}(t)C_j(t) \quad (8)$$

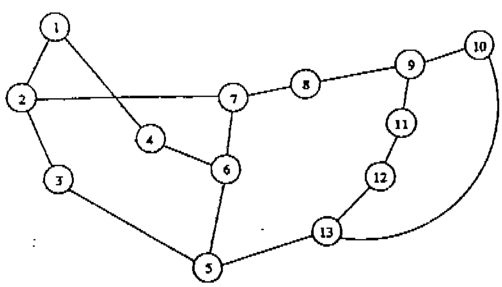
$$\frac{d}{dt}C_i(t) = \lambda_i(t)(1-C_i(t)) + \sum_{j\neq i}\mu_j(t)\theta_{ji}(t)(1-C_i(t))C_j(t) -\mu_i(t)(1-\theta_{ii}(t))p_{i1}(t) \quad (9)$$

The conditional state probabilities $p_{i1}(t)$ and $p_{i2}(t)$ can be obtained from the conditional state probabilities for the Polya-Eggenberger random variables A_i and B_i, respectively[13]. From the four differential equations, the mean and variance of customers at queue i, $E[n_i(t)]$ and $Var[n_i(t)]$, can be obtained by standard numerical integration methods.

3. Algorithms for Equal-cost Routes

We consider three different equal-cost routing algorithms. In the first algorithm, referred to as Algorithm 1, we select one of the equal-cost paths randomly. The routing table for the equal-cost paths remains unchanged during the entire simulation to ensure packets for a given destination to use the same path. In other words, there is no actual traffic distribution over equal-cost paths. This algorithm may not perform well when there are many equal-cost routes in the same area or routing domain.

In the second algorithm, referred to as Algorithm 2, we distribute traffic over the next hops equally for a given O-D(Origin-Destination) pair;thus, an equal number of packets will be sent on each equal cost-path. This algorithm can be implemented using a round-robin scheme, and provides a good routing scheme for disjoint equal-cost paths. This algorithm, however, may not perform well when there exists non-disjoint equal-cost paths. The disjoint paths represent paths that do not share common links from an origin node to the destination. Conversely, non-disjoint paths have at least one link in common. For example, in (Fig. 1), we assume that each link has the same link cost.



(Fig. 1) Sample Network

Then, there exists two equal-cost paths between origin node 1 and destination node 5. One equal-cost path is a path connecting nodes 1, 4, 6, and 5. The other one is a path connecting node 1, 2, 3, and 5. The two equal-cost paths are said to be disjoint since the two paths do not have any links in common. Similarly, there exists two equal-cost paths between origin node 1 and destination node 13 in the same network. One path is composed of nodes 1, 4, 6, 5, and 13, and the other one is a path connecting nodes 1, 2, 3, 5, and 13. In this case, however, the link between nodes 5 and 13 is a common link for the two equal-cost paths. Thus, these equal-cost paths are said to be non-disjoint from each other. In an effort to improve the performance of Algorithm 2, we propose another algorithm that uses the flow deviation scheme.

In the last algorithm, referred to as Algoithm 3, we employ an optimal routing which implements the Flow Deviation(FD) algorithm [14, 15]. Originally, the FD method was developed to assign optimum flow on each channel when the channel capacities are given. This approach is, in general, known as the flow assignment (FA) problem. In our case, when there are no equal-cost paths for a given O-D pair, a single route is determined by the SPF algorithm. The FD method will be used only for O-D pairs with multiple equal-cost paths. Thus, the FD routine needs to be invoked each time link costs change. The basic idea of the FD method is to route flow so as to minimize the average delay. The average delay is given by

$$\begin{aligned} T &= \sum_{i=1}^{N} \frac{\gamma_i}{\lambda} T_i \\ &= \sum_{i=1}^{N} \frac{\gamma_i}{\lambda} \left[\frac{1}{\mu C_i - \gamma_i} \right], \end{aligned} \quad (10)$$

where N is the number of nodes, γ_i is the total packet rate on the i^{th} channel, λ is the total packet rate the entering the network, and T_i is the average delay in the i^{th} channel. It is shown that the average delay T is a convex function of the flows, γ_i/μ[14]. Thus, finding a local minimum results in a global minimum. To find the shortest paths, the link length ℓ_i is definded as

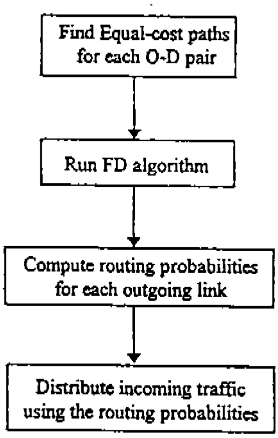
$$l_i = \frac{\partial T}{\partial(\gamma_i/\mu)} = \frac{C_i}{\lambda[C_i - (\gamma/\mu)]^2}. \quad (11)$$

The link length is computed iteratively using updated link flows on each channel. This procedure continues until an acceptable performance tolerance is reached. In order to find the optimum link flow $f_i^{(n)}$ of the ith channel after the nth iteration, the optimum link flow is given by

$$f_i^{(n)} = \frac{\gamma_i^{(n)}}{\mu},$$

where $\gamma_i^{(n)}/\mu$ is the total flow on the i^{th} channel after the n^{th} iteration. A detailed explanation of the FD algorithm is summarized in [15].

The resulting optimum flow on each link may be used to split traffic over equal-cost paths. Based on the optimum flow on each link of a node, we can compute routing probabilities for the equal-cost routes. For instance, assume that a node has two equal-cost routes for a given destination. Then, we run the FD algorithm and find optimal flow on each link of the node. We can determine the percentage of the total traffic on a link that is due to the optimum flow. This percent optimum flow may be used as an routing probability for the equal-cost paths. Once we have the routing probabilities for each O-D pair, we can distribute incoming traffic over equal-cost paths based on the routing probabilities. The optimum link flow computation can be executed whenever the network topology changes. A brief flow chart of the proposed algorithm is summarized in (Fig. 2).



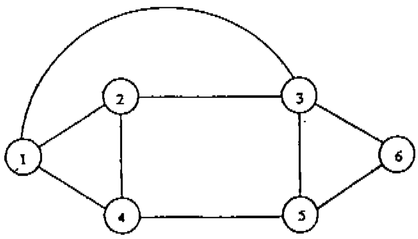
(Fig. 2) Flow chart of Algorithm 3

4. Test Cases and Performance Criteria

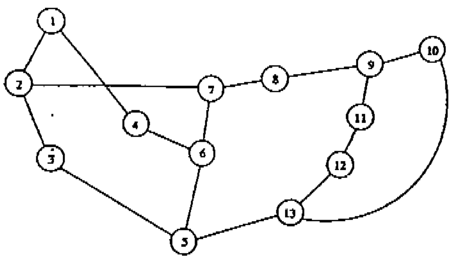
In order to test the performance of the equal-cost routing algorithms, we use five different network topologies consisting of 6, 13, 26, 32, and 61 nodes. The network topologies are shown in (Fig. 3)-(Fig. 7). The 13 node network is the T3 backbone network topology of the NSFNET. Each arc represents a bidirectional communication link. The number of links terminating at each node ranged from two to four. All links are assumed to have the same capacity.

The closure approximation results are compared to the Monte Carlo simulation of Jackson network in transient conditions. The differential equations for the first and second moments of the number of customers in each queue are solved using fourth-fifth order Runge-Kutta algorithm. In order to determine the Jackson network parameters, $\lambda_i(t)$, $\mu_i(t)$, and $\theta_{ij}(t)$, external arrival rates for a given O-D pair, routing table, channel capacity, and mean packet length are used. The Monte Carlo simulation, on the other hand, is required to handle every individual event that occurs in the network such as external arrivals, packet queueing, routing, packet transmission, link down/up, and collecting useful data. After repeating a number of simulations, ensemble average for queue statistics are computed. The number of simulation runs is set to 2500 in order to obtain 95% confidence interval widths of about 10% of the average values. To evaluate the performance of the routing algorithms, we use two relative error measures.

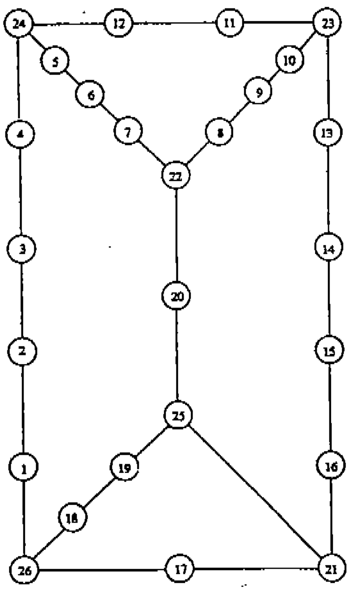
The first relative error measure represents the difference in percentage between the Monte Carlo simulation value and the closure



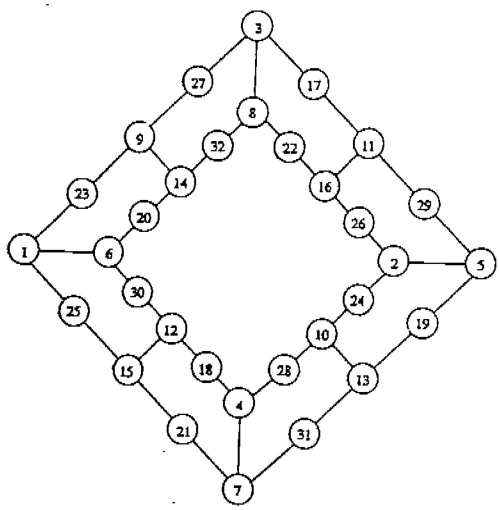
(Fig. 3) The 6 Node Network



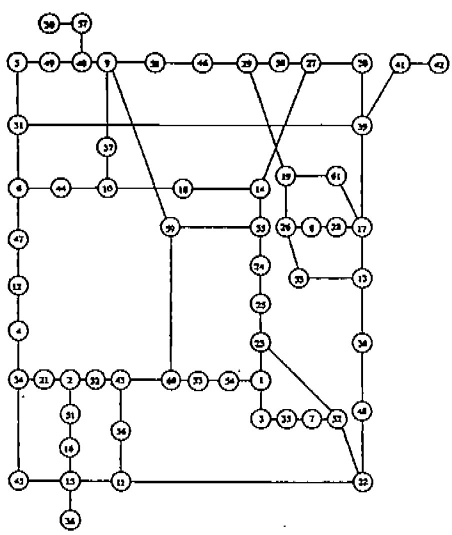
(Fig. 4) The 13 Node Network



(Fig. 5) The 26 Node Network



(Fig. 6) The 32 Node Network



(Fig. 7) The 61 Node Network

approximation value at each sample time. This relative error, denoted e_r, is defined as

$$e_r = \frac{|\text{Monte Carlo value} - \text{Closure value}|}{\text{Monte Carlo value}} \times 100(\%) \quad (12)$$

The second relative error measure represents half the width of the confidence interval with respect to the Monte Carlo value. The error, denoted e_s, is definded as

$$e_s = \frac{0.5 \times \text{confidence interval}}{\text{Monte Carlo value}} \times 100(\%) \quad (13)$$

The average percent error, e_{ave}, can be computed using the relative errors over all time steps for each queue. That is, e_{ave}, is defined as

$$e_{ave} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} e_i(j)}{n*m}, \quad (14)$$

where n is total number of queues in the test network, m is total number of samples, and $e_i(j)$represents either e_r or e_s at queue i at time j.

In a computer network, link utilization is defined as the ratio of arrival rate to service rate. If we ignore the processing time and propagation delay at link i, the service rate becomes $\mu' C_i$, where $1/\mu'$ is the average packet length in bits and C_i is the channel capacity in bits/secs. The link utilization ρ_i of link i is then given by

$$\rho_i = \frac{\gamma_i}{\mu' C_i},$$

where γ_i is arrival rate at link i(external arrivals plus arrivals from other nodes). Based on ρ_i, we define three different loading conditions as follows:

- Lightly loaded if $0 < \rho_i \leq 0.35$
- Moderately loaded if $0.35 < \rho_i \leq 0.70$
- Heavily loaded if $\rho_i > 0.70$

The performance of the routing algorithms is evaluated under the three different loading conditions.

5. Simulation Results

5.1 Average Mean Results

For each load splitting algorithm, we first compare the performance of the closure approximation to that of the Monte Carlo simulation. If the relative errors between the closure approximation and the Monte Carlo simulation are fairly small, the closure approximation results may be used to evaluate the performance of the load splitting algorithms, which results in considerable execution time reduction. To investigate the performance of the routing algorithms, we use the error measures in (12) and (13). The resulting mean relative errors are summarized in ⟨Table 1⟩ for stationary input. ⟨Table 1⟩ compares the relative errors for the three algorithms when each test network was loaded lightly to heavily. In the table, "Simul. error" represents the Monte Carlo simulation error which was obtained from (13), while "Approximation error" represents the closure approximation from (12). The average approximation errors are smaller than the simulation errors, which means the closure approximation values always remain inside the 95% confidence intervals. The simulation errors for mean vary with the algorithm since they are dependent only on the simulation values. The individual queue statistics from the Monte Carlo simulation may vary depending upon the algorithms, but the average values over all queues should remain the same.

The approximation errors, however, are different slightly depending on algorithms. The approximation errors for Algorithm 2 are very close to those with the Algorithm 3. This is because both algorithms use random number generators to determine next hops when there exists multiple equal-cost paths. Algorithm 2 distributes equal-cost traffic with equal probability over the multiple equal-cost paths, while Algorithm 3 distributes the traffic based on the distribution probability computed by the optimum flow deviation algorithm.

〈Table 1〉 Comparison of Mean Average Errors(%)

Network Topology	Input Load	Simul. Error(e_s)	Approximation Error(e_r)		
			Algorithm 1	Algorithm 2	Algorithm 3
6 Node Net	Light	7.49	3.23	3.29	3.46
	Moderate	6.93	3.11	3.19	3.61
	Heavy	5.79	3.49	3.45	4.01
13 Node Net	Light	8.07	3.99	4.19	4.27
	Moderate	6.58	3.43	3.46	3.57
	Heavy	5.70	3.72	3.23	3.95
26 Node Net	Light	8.60	4.56	4.90	4.87
	Moderate	6.21	3.65	3.74	3.58
	Heavy	4.94	3.05	3.13	3.12
32 Node Net	Light	8.40	3.42	3.32	3.26
	Moderate	5.96	2.37	2.50	2.43
	Heavy	4.18	4.17	4.35	4.21
61 Node Net	Light	11.68	5.32	5.44	5.63
	Moderate	8.26	3.72	3.79	3.85
	Heavy	7.38	3.29	3.46	3.49

On the other hand, Algorithm 1 chooses one among equal-cost next paths randomly and fix the routing table for a given O-D pair. This fixed routing may provide better approximation of θ_{ij} and λ_i values in the closure approximation. As shown in 〈Table 1〉, Algorithm 1 produces slightly smaller approximation errors than Algorithm 2 and Algorithm 3. The relative errors in the table show that the routing algorithms were implemented correctly and the closure approximation can be used to estimate the queue statistics instead of Monte Carlo simulation.

5.2 Average Delay Results

The performance of the three algorithms is evaluated based on the average delay in the network. Average delay is one of the most popular performance measure for evaluating computer networks. It is desirable to compare the average delay for each load splitting algorithm to the optimal delay. The optimal delay is computed based on the optimal routing described in Section 3. In Algorithm 3, the optimal routing is determined only for the equal-cost paths, while the optimal delay is computed using the optimal routing tables in the network. The optimal routing tables are considerably different from the shortest path first(SPF) routing tables in the three algorithms. However, the optimal delay resulting from the optimal routing will be a useful measure in comparing the performance of the splitting algorithms.

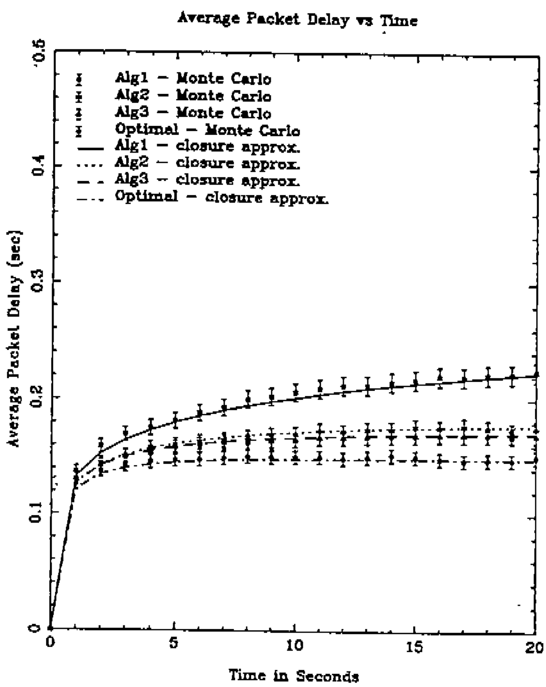
Before comparing the performance of the three splitting algorithms using the average delay, it is helpful to characterize the network topologies. 〈Table 2〉 compares the total number of equal-cost paths with respect to the total number of paths.

It also compares the total number of equal-

〈Table 2〉 Comparison of Equal-cost paths

Network Topology	Total Paths(A)	Equal Paths(B)	Non-disj. Paths(C)	B/A (%)	C/B (%)
6 Node Net	30	6	0	20	0
13 Node Net	156	17	5	10.9	29.4
26 Node Net	650	58	30	8.9	51.7
32 Node Net	992	272	192	27.4	70.6
61 Node Net	3660	411	254	11.2	61.8

cost paths to total number of non-disjoint paths for each topology. As can be seen in the table, the number of equal-cost paths with respect to the total paths ranges from 9 % to 28% . On the other hand, the number of non-disjoint paths with respect to all equal-cost paths ranges from 0% to 62%. The performance of the algorithms can be compared using either the closure approximation or the Monte Carlo simulation. (Fig. 8) compares the average packet delay when the 13 node network is heavily loaded. This figure compares the performance of the load splitting algorithms using both the Monte Carlo simulation and closure approximation. The optimal delay results from both the Monte Carlo simulation and closure approximation are also plotted for comparison purpose. In the figure, we see that the



(Fig. 8) The Average Delay Comparison for the 13 Node Network

closure approximation results for each case always remain inside 95% confidence intervals of the Monte Carlo simulation. We also observe that Algorithm 1 produces the largest average delay, while Algorithm 3 has the smallest delay. The performance difference between Algorithm 3 and Algorithm 2 is fairly small compared to that between Algorithm 3 and Algorithm 1. This is because both Algorithm 2 and Algorithm 3 split traffic for equal-cost paths with different routing probabilities, while Algorithm 1 chooses one randomly among equal-cost paths. Note that the 13 node network has 30 % non-disjoint equal-cost paths out of the total number of equal-cost paths. In addition, we observe that the optimal routing produces smaller delay than the shortest path first routing.

For the overall comparison, the average delay from the closure approximation is summarized in ⟨Table 3⟩.

The average delay results from the Monte Carlo simulation is very similar to the results in the table since the closure approximation results for total number of packets closely

⟨Table 3⟩ Comparison of Average Delay (in milliseconds)

Topology	Input Load	Algorithm 1	Algorithm 2	Algorithm 3
6 Node Net	Light	42	41	41
	Moderate	60	58	58
	Heavy	146	143	143
13 Node Net	Light	77	74	72
	Moderate	99	95	91
	Heavy	222	177	170
26 Node Net	Light	117	115	116
	Moderate	154	150	151
	Heavy	366	340	341
32 Node Net	Light	130	124	120
	Moderate	205	183	162
	Heavy	1322	929	892
61 Node Net	Light	183	180	180
	Moderate	570	551	550
	Heavy	1050	848	845

follow the Monte Carlo simulation results.

The average delay is based on the average total number of packets in the network. As can be seen in the table, both Algorithm 2 and 3 perform better than Algorithm 1 for all test cases. On the other hand, the performance difference between Algorithm 2 and 3 is fairly small when the test network has a relatively small number of equal-cost paths with respect to the total number of paths. In some cases(i.e., 26 node network), Algorithm 2 performed slightly better than Algorithm 3, but the performance difference is very small. For networks that have a relatively large number of equal-cost paths and non-disjoint paths, Algorithm 3 performs better than Algorithm 2. The performance improvements are substantial as the network loading is heavier. The proposed algorithm, in general, performs better than the other two algorithms.

6. Conclusions

We proposed an equal-cost bifurcated routing algorithm which could be useful in practical network design problem. The performance of the proposed routing algorithm was compared to that of the existing equal-cost routing algorithms via both the conventional Monte Carlo simulation and the closure approximation. The closure approximation provided fairly close results to the Monte Carlo simulation results with considerably smaller execution times. We can, therefore, use the closure approximation to compare the performance of the three load splitting algorithms using the average delay.

In conclusion, the proposed algorithm, in general, performed better than the other two equal-cost routing algorithms.

References

[1] "Intermediate system to intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service(ISO 8473)," ISO/IEC 10589, 1992.

[2] R. Perlman, "A comparison between two routing protocols:OSPF and IS-IS," IEEE Network, pp. 18-24, Sept. 1991.

[3] C. Hedrick, "Routing Information Protocol," RFC1058, June 1988.

[4] D. Gross and C. Hariss, Fundamentals of queueing theory, Wiley, 1985.

[5] W. K. Tsai and P. E. Cantrell, "A simple derivation of transient queue statistics and applications," Performanc Evaluation, Vol. 10, pp. 103-114, 1989.

[6] W. A. Massey, "An operator-analytic approach to the Jackson networks," J. Applied Prob., Vol. 21, pp. 379-393, 1984.

[7] K. L. Rider, "A simple approximation to the average queue size in the time-dependent M/M/1 queue," J. ACM, Vol. 23, pp. 361-367, April 1976.

[8] M. H. Rothkopf and S. S. Oren, "A closure approximation for the non-stationary M/M/s queue," Manage. Sci., Vol. 25, pp. 522-534, June 1979.

[9] S. S. L. Chang, "Simulation of transient and time varying conditions in queueing networks," Proc. of the Seventh Pitts-burgh conference on modeling and simulation, pp. 1075-1078, 1977.

[10] G. M. Clark, "Use of Polya distribution in approximate solutions to non-stationary M/M/s queues," Commun. ACM, Vol. 24, pp. 206-217, April 1981.
[11] W. A. Baker, "Transient queueing approximation for computer networks," Master's thesis, Texas A&M University, College Station, TX, Dec. 1986.
[12] J. M. Andrewartha and P. E. Cantrell, "Transient approximation for Jackson networks," in Proc. of the IEEE GLOBECOM, Nov. 1989.
[13] B. H. Lee, "Transient queueing approximations for modeling computer networks," Ph.D. Dissertation, Texas A &M University, College Station, TX, Dec. 1993.
[14] L. Kleinrock, Queueing Systems, Volume II:Computer Applications, Wiley, 1975.
[15] M. Gerla, "The design of store-and-forward networks for computer communications," Ph.D. Dissertation, Univ. of California, Los Angeles,CA, 1973.

이 봉 환
1985년 서강대학교 전자공학과 졸업(학사)
1987년 연세대학교 대학원 전자과(공학석사)
1993년 미국 Texas A & M 대학 전기과(컴퓨터 네크워크 전공)공학박사
1987~1989년 한국통신 연수원 교수요원
1993~현재 한국통신 연구개발원 통신망연구소
관심분야 : 컴퓨터 통신망 모델링, 네트워크 시뮬레이션, ATM망에서의 멀티미디어 서비스(VOD)개발등