On Routing Performance of MENTOR Algorithm

KAIRAT JAROENRAT, PISIT CHARNKEITKONG
Faculty of Information Technology
Rangsit University
52/347 Muang-Ake, Phaholyothin Rd., Lak-Hok, Pathumtani,
THAILAND
kairat@kairat.com, pisitc@rangsit.rsu.ac.th

Abstract: - Mesh Network Topological Optimization and Routing (MENTOR) algorithm is a low complexity and efficient partial mesh networks design algorithm. This study explores the relation between design parameters and performance of traffic assignment of MENTOR algorithm. We analyze 432 networks designed by MENTOR for 4 sets of 50 nodes each with equivalently distributed demand and randomly generated locations. For each of these networks, the performances at normal load and at congestion threshold of MENTOR flow assignment are calculated and compared with the optimum solution obtained by solving the linear programming. It is found that the routing performances depend on the initial tree used in the MENTOR algorithm, as well as the allowable minimum and maximum link utilization. MENTOR networks start with star topology give much better performance than that start with minimum spanning tree. In term of utilization, routing performances keep very close to that of the optimal when the gap between maximum and minimum utilization is small and get as worse the gap increase. The impacts of node degree on routing performances are also investigated. We observed that the performances decrease as node degree increase, and get worse when maximum utilization increase.

Key-Words: - MENTOR Algorithm, network design, mesh networks, traffic routing, flow assignment

1 Introduction

Network design process composes of 2 major tasks the topology design and traffic routing. Topology design is to choosing links to be installed as well as to determining the link capacity such that the overall network cost is minimized. Traffic routing, often called traffic engineering, is to distribute load for given traffic demands over the installed link such that performances are optimized. Most network design algorithms are fairly complex. For example simple branch exchange algorithm [8] requires complexity of $O(N^5)$, where N is number of nodes, which is prohibitive for moderate to large size networks. On the other hand, as Internet become life-line of business and commercial application, to design of large data network, i.e. Internet Service Providers (ISPs) backbone, ones have to aware of several new issues. One of the most important is that IP network is a datagram network, in which the routing protocols route traffic over path with shortest distance, i.e. sum of link weight. However, link weight setting for an optimum routing pattern is also a complex problem or even unfeasible [3] [4] [9]. To solve the problem, most ISP backbone employ overlay approach which route traffic over Permanent Virtual Connections (PVC) of ATM and recently over Label Switch Paths (LSP) Multi Protocol Label Switching (MPLS). Benefits of implementing IP traffic engineering with MPLS are discussed in [5]. Another important issue is that the ISPs must be aware is the rapid growth traffic demand, and hence enough capacity must be reserved for the future. Reserved capacity of a network can be control by a number of network design parameters. Some of the most obvious and easy understand ones are the allowable maximum link utilization and the minimum link utilization. For example, setting low allowable maximum link utilization and minimum link utilization often leads to a network with higher cost but more reserved capacity, and vice versa.

Kershenbaum et. al. [1] have proposed a low complexity $O(N^2)$ heuristic network design algorithm called MENTOR (Mesh Network Topological Optimization and Routing). Network obtained by other known high complexity algorithm are only several percent better than that of MENTOR networks. MENTOR is flexible enough to use as a design algorithm for virtual circuit network such as ATM as well as MPLS that are used in overlay approach for ISP backbone network. However, traffic routing of MENTOR is not always optimal, and strongly depends on design parameters, i.e. maximum and minimum link utilizations.

This paper investigates the relation between design parameters and performance of flow assignment of MENTOR algorithm. We analyze 432 networks designed by MENTOR for 4 sets of 50 nodes each with equivalently distributed demand and randomly generated locations. For each of these networks, the performances at normal load and at threshold of congestion of MENTOR flow assignment are calculated and compared with the optimum solution obtained by solving the linear programming.

2 Problem Formulation

2.1 MENTOR Algorithm

MENTOR algorithm is a low complexity heuristic network design algorithm. This low complexity is achieved by doing implicit routing over a link at the same time it is considered to be installed. For a given set of nodes N, demand matrix D and link cost matrix X, let $d_{s,t}$ and $x_{s,t}$ are the amount of traffic flow and link installation cost from s and t, respectively. The characteristics of network obtained by MENTOR algorithm are (1) traffic demands are routed on relatively direct paths (2) links have reasonable utilization and (3) relatively high capacity links are used.

MENTOR starts with clustering process. In this stage, nodes are classified in to end nodes and backbone nodes using a clustering algorithm. Examples of possible clustering algorithms are threshold clustering and K-mean clustering. Here in this paper, we consider only the case where traffic demands are distributed equivalently among all nodes. Therefore, all nodes can be considered as backbone node.

Next, a good tree is formed to interconnect all (backbone) nodes. Kershenbaum et. al. [1] suggests to a use a heuristic, which can be thought of as a modification of Prim and Dijkstra algorithm to build the tree. The algorithm works almost the same manner as Dijkstra algorithm but with a tunable parameter α, 0≤α≤1. The tree is to be expanded one node at a time by connecting a tree node i to an out of tree node j such that $\alpha \cdot L_i + x_{ic}$ minimized, where L_i is the cost of path from root node along the tree to node i. Note that α = 0 and 1 is corresponding to Minimum Spanning Tree (MST) and Shortest Path Tree (SPT), respectively.

Given a tree, the objective of MENTOR is to consider adding a direct link between each pair of nodes if the amount of traffic is reasonable. Let the maximum utilization be ρ, and the minimum utilization be defined in term of ρ and slack s as $(1-s)\,\rho$, where s, $0\le s \le 1$. Consider a pair of nodes A and B, let C_{AB} and l_{AB} be link capacity and accumulated load flow between A and B, respectively. If traffic between A and B is too small, i.e $l_{AB} < \rho\, C_{AB}\,(1-s)$, no link is added and all traffic l_{AB} is overflowed to the next most direct path. A link is added if traffic is in between maximum and minimum utilization, i.e. $\rho\, C_{AB}\,(1\text{-}s) \le l_{AB} \le \rho C_{AB}$. However, if $l_{AB} > \rho C_{AB}$, a direct link is added only when traffic bifurcation among multiple routes is possible. If bifurcation is possible, a new link of C_{AB} is added to serve a portion of traffic ρC_{AB}, and the left portion $l_{AB} - \rho C_{AB}$ is overflowed to the next most direct path. Otherwise, if no bifurcation is possible, no link is added and all traffic l_{AB} is overflowed to the next most direct path.

Node pairs are sequenced such that a link between a pair is considered only when all traffic flows that could overflow to it are already considered. Typically, routing pattern in which traffic bifurcation is possible tends to give better performance. So, in this study, we consider only the cases where the traffic bifurcation is possible.

MENTOR gives fairly good results and widely used to many type of networks, e.g. Frame Relay, ATM as well as MPLS. However, the impact of design parameters, e.g. ρ, s and α, on efficiency of traffic routing are not yet studied before.

2.2 Objective Function

Consider a directed network graph G = (N, A) with a capacity c_a for each $a \in A$ and as define in previous section, d_{st} denote the amount of traffic flow between s and t. Let f_a^{st} indicate how much of the traffic flow from s to t over arc a, traffic load l_a over link $a \in A$ is the sum of all f_a^{st}. It is suggested in [4] to measure the performance of network by cost function

$$\Phi = \sum_{a \in A} \phi_a(l_a, c_a), \qquad (1)$$

where $\phi_a(l_a, c_a)$ is an M/M/1 queuing theory style link cost function given by

$$\phi_a(l_a, c_a) = l_a / (c_a - l_a) \qquad (2)$$

With this function, it is more expensive to send flow along arcs whose loads approach capacity, which is what we want. However, the function does not deal with overloaded links, i.e. $l_a \ge c_a$. To overcome this problem, $l_a/(c_a - l_a)$ is approximated by a piece-wise linear function $\phi_a(0) = 0$ and derivative

$$\phi'_a(l_a, c_a) = \begin{cases} 1 \text{ for } & 0 \le l_a/c_a < 1/3, \\ 3 \text{ for } & 1/3 \le l_a/c_a < 2/3, \\ 10 \text{ for } & 2/3 \le l_a/c_a < 9/10, \\ 70 \text{ for } & 9/10 \le l_a/c_a < 1, \\ 500 \text{ for } & 1 \le l_a/c_a < 11/10, \\ 5000 \text{ for } & 11/10 \le l_a/c_a < \infty. \end{cases} \qquad (3)$$

2.3 Optimum Solutions

With piece-wise linear cost function define by (3), the general routing problem can be formulated as the following linear programming [3] [4].

$$\text{Min } \Phi = \sum_{a \in A} \phi_a \quad (4)$$

Subject to:

$$\sum_{u:(u,v)\in A} f_{u,v}^{s,t} - \sum_{u:(u,v)\in A} f_{v,u}^{s,t} = \begin{cases} d_{st} \text{ if } v = t \\ -d_{st} \text{ if } v = 0 \\ 0 \text{ otherwise} \end{cases} \quad v,s,t \in N, \quad (5)$$

$$\phi_a \geq l_a \quad a \in A, \quad (6)$$

$$\phi_a \geq 3l_a - 2/3c_a \quad a \in A, \quad (7)$$

$$\phi_a \geq 10l_a - 16/3c_a \quad a \in A, \quad (8)$$

$$\phi_a \geq 70l_a - 178/3c_a \quad a \in A, \quad (9)$$

$$\phi_a \geq 500l_a - 1468/3c_a \quad a \in A, \quad (10)$$

$$\phi_a \geq 5000l_a - 19468/3c_a \quad a \in A, \quad (11)$$

$$l_a = \sum_{t \in N} f_a^{s,t} \quad a \in A, \quad (12)$$

$$f_a^{s,t} \geq 0 \quad a \in A;\ t \in N. \quad (13)$$

Constraint (5) are flow conservation constraints; constraints (6) – (11) describe the cost function; and constraint (12) define the load on each arc.

We observed that it is not fair to compare the general optimum solution with other traffic routing strategies having limit maximum link utilization, e.g. that of MENTOR algorithm. This is because the purpose of limiting maximum link utilization is to reserve capacity to handle more traffic load when network get congest. To take in to account the capacity reservation, the additional maximum utilization constraint

$$l_a/c_a \leq \rho. \quad (14)$$

The optimum solutions used to compare with MENTOR networks in section 3 are obtained by solving (4) subject to (5) – (14).

2.4 Normalizing Routing Cost

Fortz and Thorup [3] proposed a normalizing scaling factor for the routing cost that makes possible comparisons across different network sizes and topologies:

$$\Phi_{\text{UNCAP}} = \sum_{s,t \in NxN} d_{s,t} h_{s,t} \quad (15)$$

where h_{st} = minimum hop count between s and t.

For any routing cost Φ, the scaled routing cost or normalized routing cost is defined as

$$\Phi^* = \Phi / \Phi_{\text{UNCAP}} \quad (16)$$

The above program is a complete linear programming formulation of the general routing problem. We shall use Φ to denote the optimal general routing cost.

3 Experiments

In order to evaluate the efficiency of flow assignment calculated by MENTOR algorithm, we analyze the performances of a number synthesized network and observe the relation between design parameters and performances.

3.1 Experiment Set Up

DELITE [6] is used to synthesize 4 sets of 50 nodes each with different node distribution obtained by varying SEED parameter. We shall refer to these set of nodes as N1, N2, N3 and N4. The traffic demand matrix for each set of nodes is also generated by DELITE with default setting and total traffic in and traffic out of each node are 100 Mbps.

By varying design parameters, a total of 432 MENTOR networks are generated for N1-N4 using the full-duplex link of capacity 45 Mbps. For each node sets, two groups of networks corresponding to Minimum Spanning Tree (α=0) and Shortest Path Tree (α=1) are generated. For each type of spanning tree, 54 networks are generated by varying of ρ, $\rho \in$ (0.4, 0.5, 0.6, 0.7, 0.8, 1.0) and s, $s \in$ (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8).

3.2 Routing Performances

For each of 432 networks, optimum routing solution with maximum link utilization constraint proposed in section 2.3 is solved by GLPK[7]. On Intel Pentium IV Xeon 3.3 GHz machine, it takes maximum 2 hours to solve the optimal routing problem. Normalized cost Φ^* of MENTOR flow assignment and optimal solution are calculated for different scaling of projected demand matrix.

Given network demands, performance of MENTOR flow assignment at normal load is measured by % of cost different from optimality

$$\Delta C = \frac{\Phi_M^* - \Phi_O^*}{\Phi_O^*} \times 100, \quad (17)$$

where Φ^*_M and Φ^*_O are normalized cost of MENTOR flow assignment, and that of optimum solution, measured at demand used to design the network respectively.

As seen in section 2, the cost function increase rapidly toward 5000 after the ϕ_a = 10 ⅔.The performance of MENTOR flow assignment at the threshold of congestion is measured by % of demand different from optimality

$$\Delta D = \frac{D_M - D_O}{D_O} \times 100, \quad (18)$$

where D_M and D_O are the scaling traffic demand of MENTOR flow assignment, and that of optimum solution measured when the cost $\Phi^* = 10\%$, respectively

The results are presented in Fig.1-16.

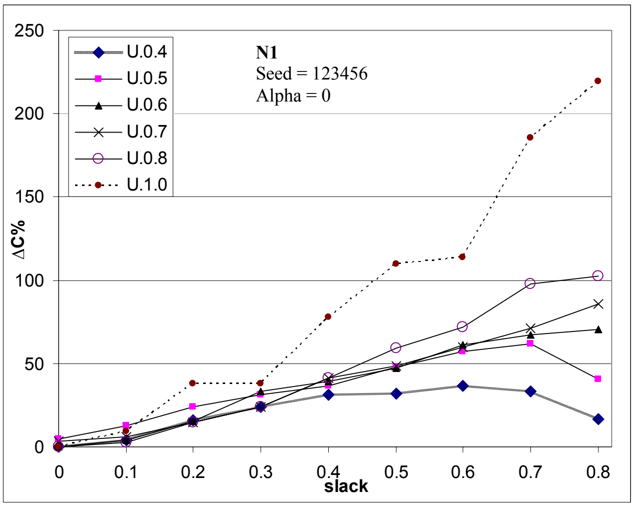

Fig.1. Δ*C* of networks with Alpha = 0 for N1.

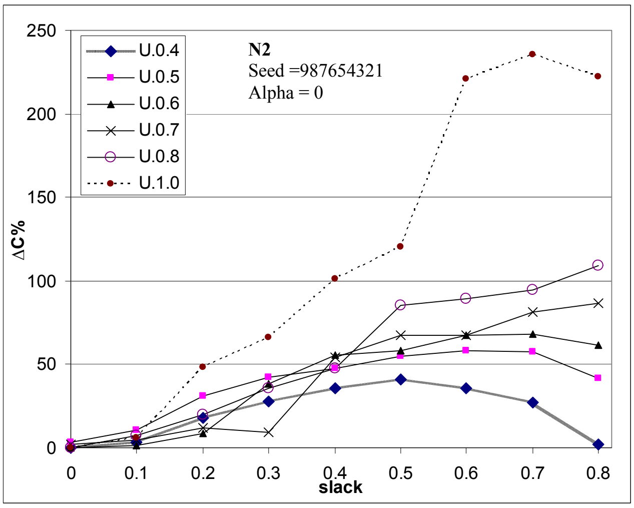

Fig.2. ΔC of networks with Alpha = 0 for N2.

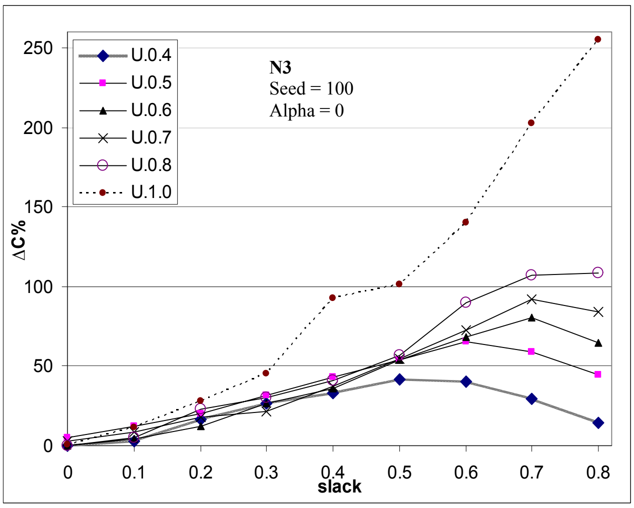

Fig.3. ΔC of networks with Alpha = 0 for N3.

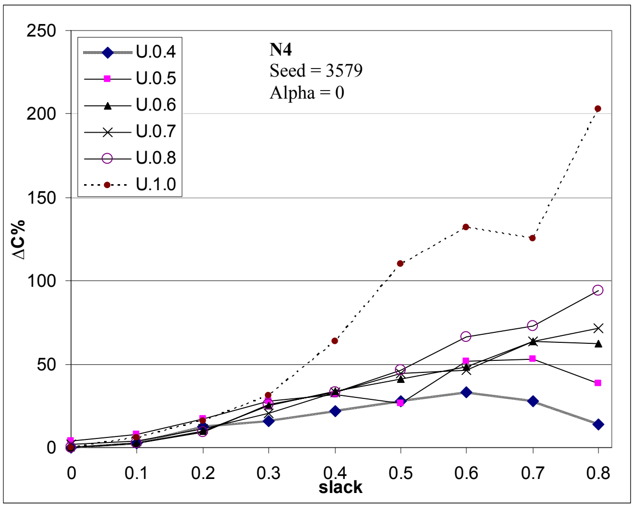

Fig.4. Δ*C* of networks with Alpha = 0 for N4.

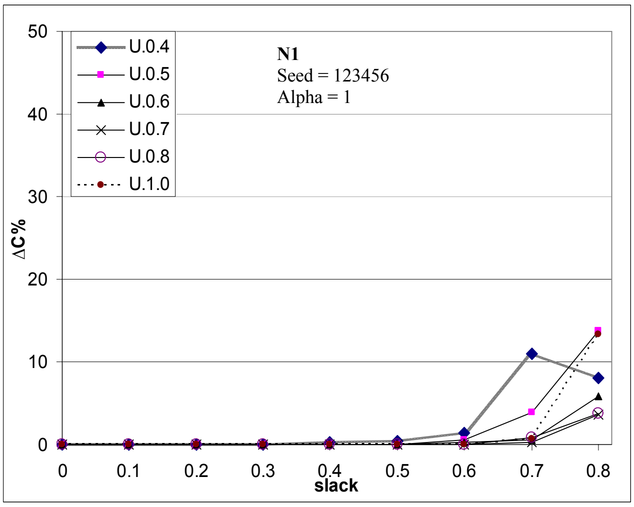

Fig.5. ΔC of networks with Alpha = 1 for N1.

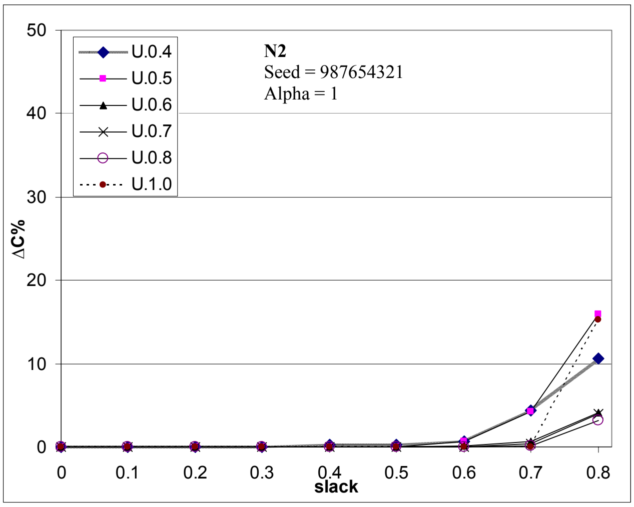

Fig.6. ΔC of networks with Alpha = 1 for N2.

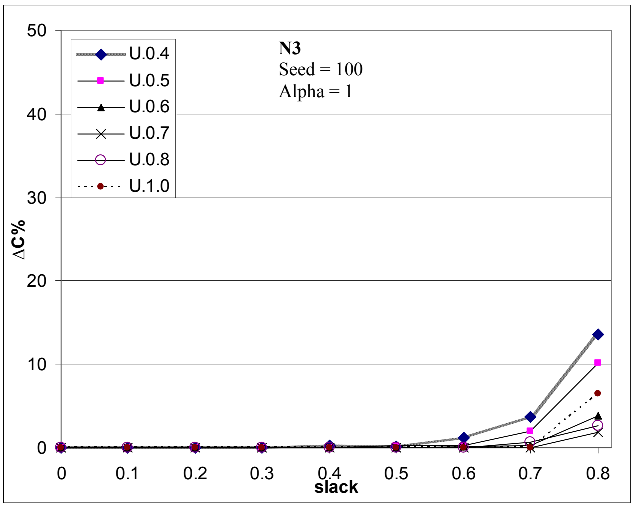

Fig.7. ΔC of networks with Alpha = 1 for N3.

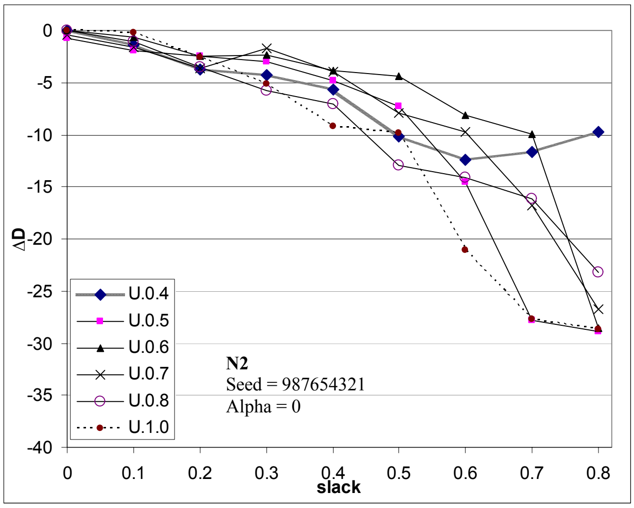

Fig.10. ΔD of networks with Alpha = 0 for N2.

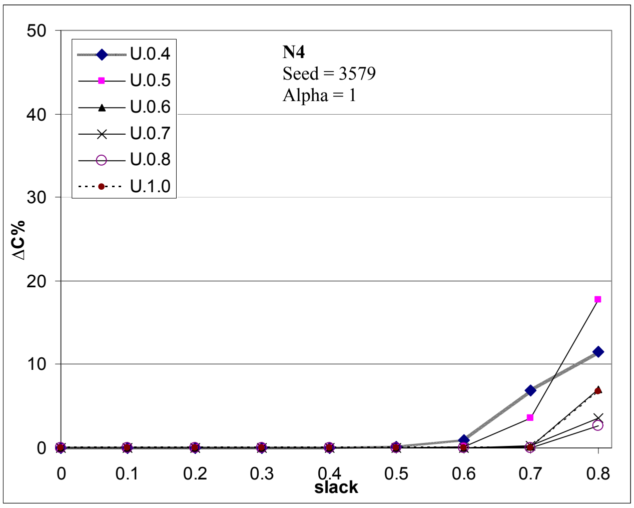

Fig.8. ΔC of networks with Alpha = 1 for N4.

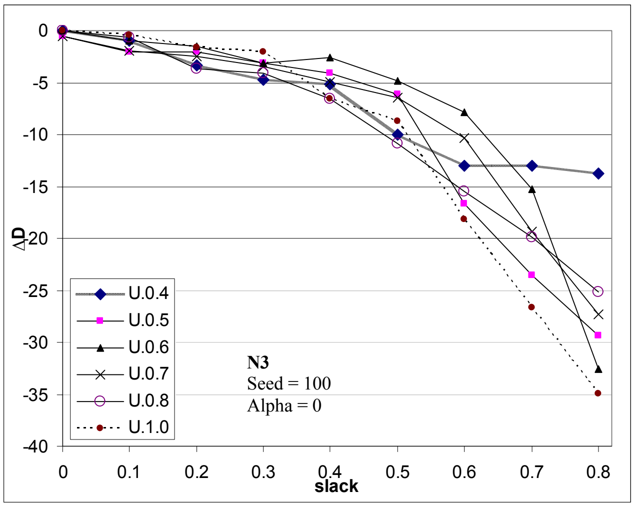

Fig.11. ΔD of networks with Alpha = 0 for N3.

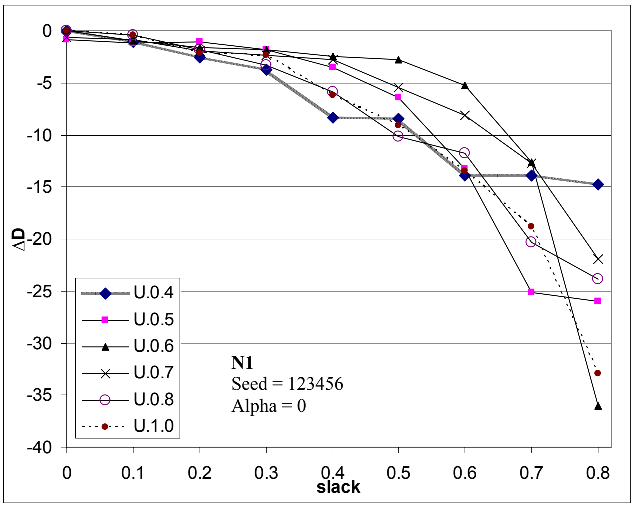

Fig.9. ΔD of networks with Alpha = 0 for N1.

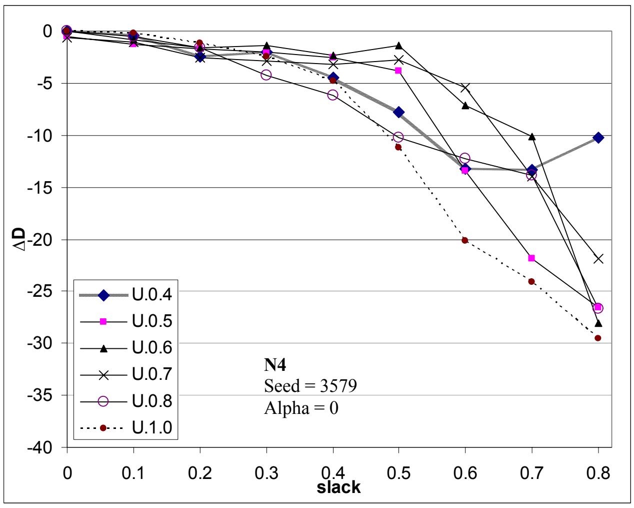

Fig.12. ΔD of networks with Alpha = 0 for N4.

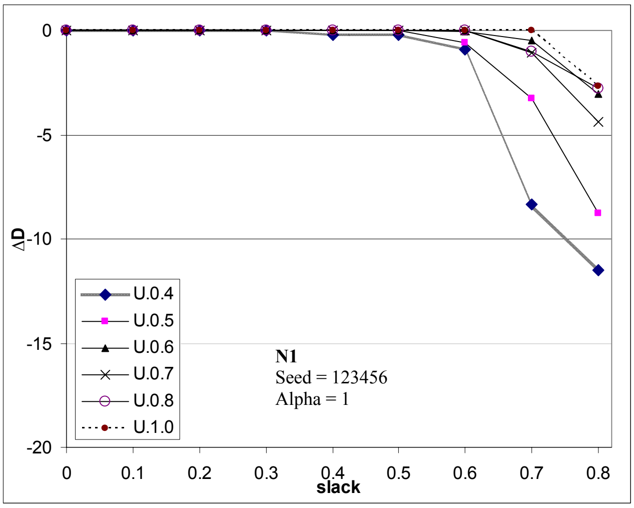

Fig.13. ΔD of networks with Alpha = 1 for N1.

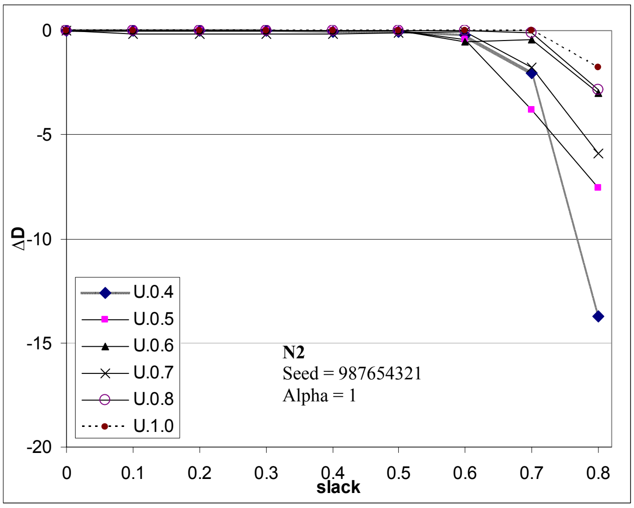

Fig.14. ΔD of networks with Alpha = 1 for N2.

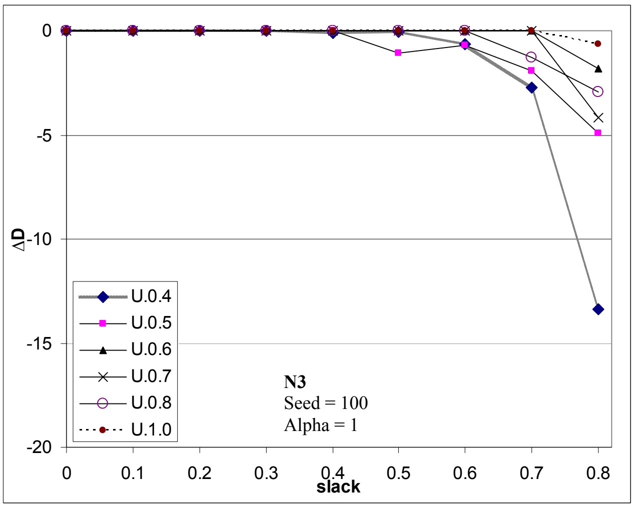

Fig.15. ΔD of networks with Alpha = 1 for N3.

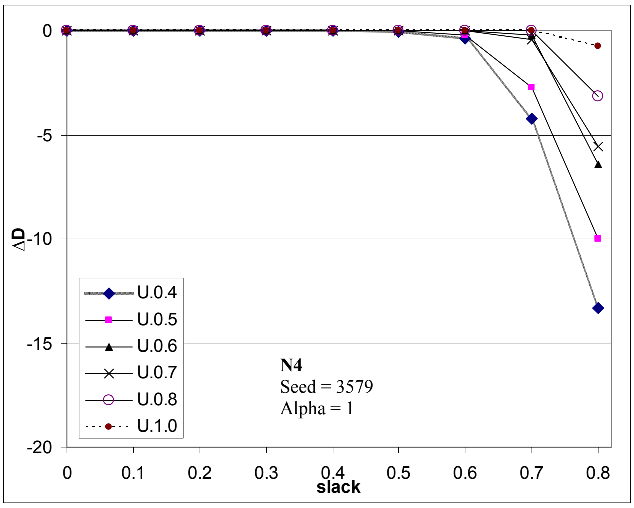

Fig.16. ΔD of networks with Alpha = 1 for N4.

3.3 Experiment Analysis

Fig.1-4 present ΔC of MENTOR networks with α=0 designed for N1-N4, respectively. It is clear that ΔC is more depend on s than ρ. For small s, s=0.1, ΔC is small with average 5.5244%, and hardly change with ρ. As s increase, ΔC get worse and more depend on ρ, ΔC achieve maximum of 250% at s=0.8 and ρ=1.

Fig.5-8 present ΔC of MENTOR networks with α=1 designed for N1- N4, respectively. For small s, s=0.1, ΔC is very small with average 0.0037%, and also hardly change with ρ. As s increase, ΔC get worse and more depend on ρ, ΔC achieve maximum of nearly 20% at s=0.8.

Fig.9-12 present ΔD of MENTOR networks with α=0 designed for N1-N4, respectively. For small s, s=0.1, ΔD is small with average 0.9629%, and also hardly change with ρ. As s increase, ΔD get worse and more depend on ρ, ΔD achieve maximum of ~35% at s=0.8 and ρ=1

Fig.13-16 present ΔD of MENTOR networks with α=1 designed for N1- N4, respectively. For small s, s=0.1, ΔD is very small with average 0.0077%, and hardly change with ρ. As s increase, ΔD get worse and more depend on ρ, ΔD achieve maximum of nearly 15% at s=0.8.

It is obvious from the results that performances of networks with α=1 are much better than that of networks with α=0.

To make the relation between routing performances and utilization more clear, ΔC and ΔD of the same α, ρ and s are averaged, and plotted versus the $\Delta U = s\rho$, the different between maximum utilization and minimum utilization, as shown in Fig. 17-20. Fig.17-18 showed that, for α =0, both

ΔC and ΔD get large as ΔU increase. In Fig.18, given ρ, the slope slowly increases as ΔU moves toward ρ. Fig.19-20 showed that, for α =1, the relations are knee curves. Given ρ, both ΔC and ΔD keep very close to the optimal and drastically get large as ΔU move close to ρ.

It has been observed in [1] and [2] that the node degree δ of the obtained MENTOR networks depends upon the selected α, ρ and s. Therefore, it is worth investigating the relation between δ and performances of traffic routing.

The average node degrees δ of networks generated in section 3.2 that have the same α, ρ and s are averaged and tabulated as shown in Table.1, Table.2. The relation between the ΔC, as well ΔD, and the average δ for networks of the same α, ρ and s are plotted in Fig.21-24

In Fig.21-22, for α =0, the results showed that first both ΔC and ΔD increase rapidly until certain value of δ, depending on ρ, then ΔC, as well as ΔD, start to keep constant or decrease slowly.

In Fig.23-24, for α = 1, the figures showed that both ΔC and ΔD get large as δ increase. The slopes of the graphs tend to increases as δ increase.

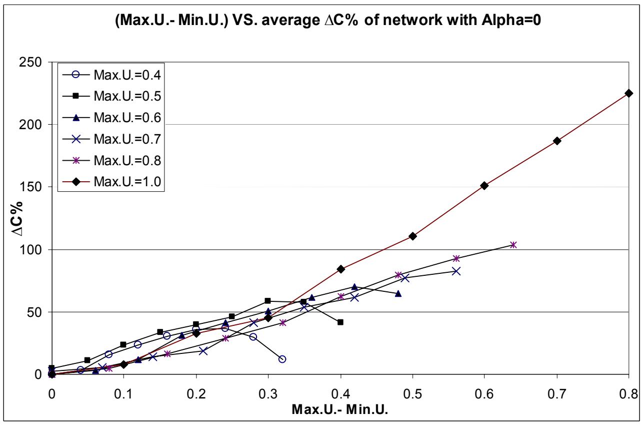

Fig.17 Average ΔC vs. ΔU for Alpha=0

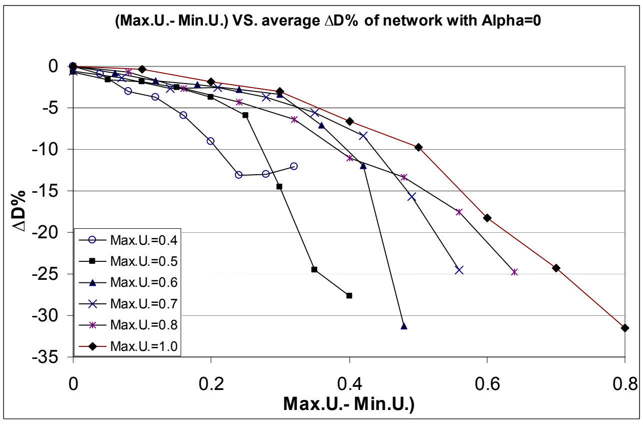

Fig.18 Average ΔD vs. ΔU for Alpha=0

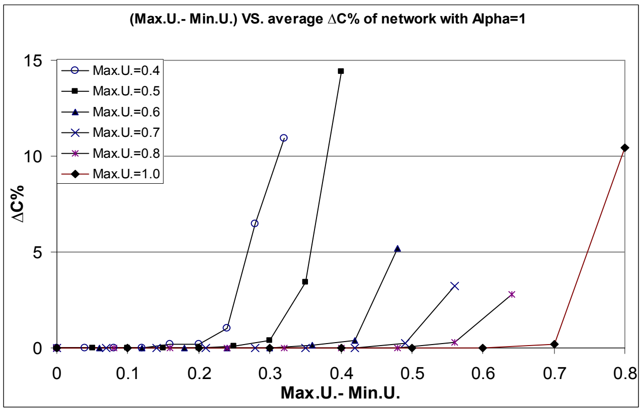

Fig.19 Average ΔC vs. ΔU for Alpha=1

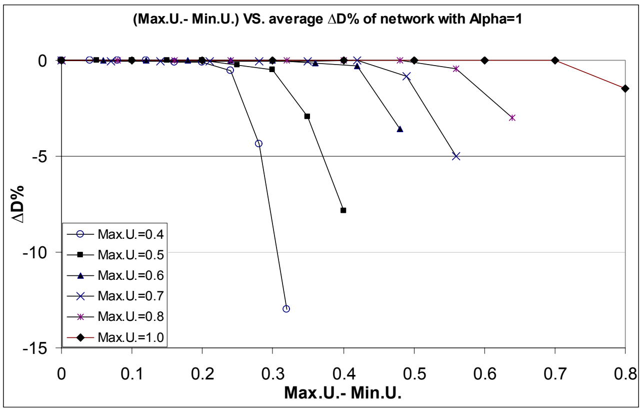

Fig.20 Average ΔD vs. ΔU for Alpha=1

4 Conclusion

In this study, the relations between design parameters and performance of traffic routing of MENTOR algorithm have been explored. Traffic routing of 432 MENTOR networks have been analyzed. Performances are evaluated in terms of percent deviation of routing cost ΔC from optimal at normal load, and percent deviation of traffic demand ΔD from optimal when routing cost equal to congestion threshold. We analyzed the relation between ΔC and ΔD and MENTOR design parameters which are coefficient α that control characteristics of MENTOR's initial tree, maximum link utilization ρ; and the different between maximum and minimum link utilization ΔU. The impact of ΔU is observed by varying slack s, $0 \le s \le 1$, where $\Delta U = s\rho$. It is found that the ΔC and ΔD of networks with α = 1, i.e. MST, is much better than that of network with α = 0, i.e. star networks. In term of utilization, routing performances keep very close to that of the optimal for small value of ΔU and get worse the ΔU increase. The impacts of node degree δ on routing performances are also investigated. Both ΔC and ΔD decrease as δ increase, and get worse when ρ increase.

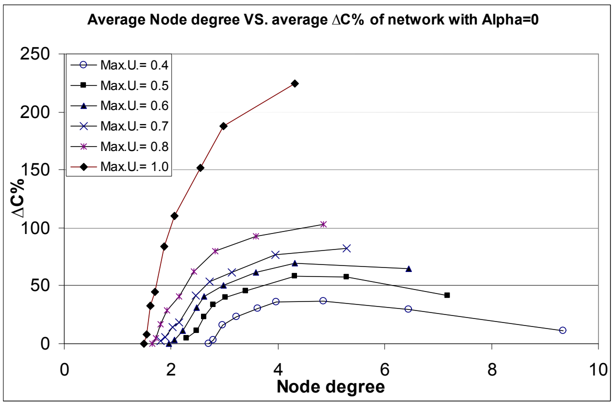

Fig.21 Average Δ*C* vs. δ for Alpha=0

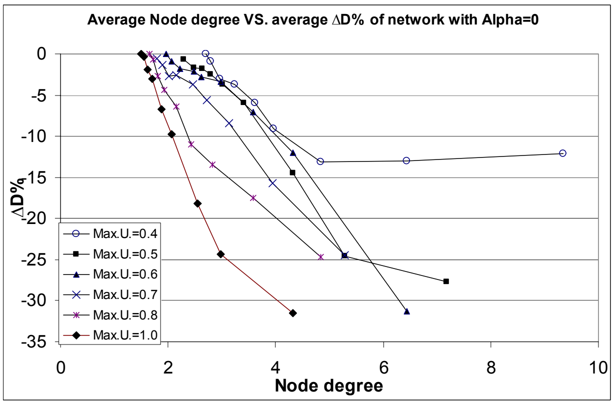

Fig.22 Average Δ*D* vs. δ for Alpha=0

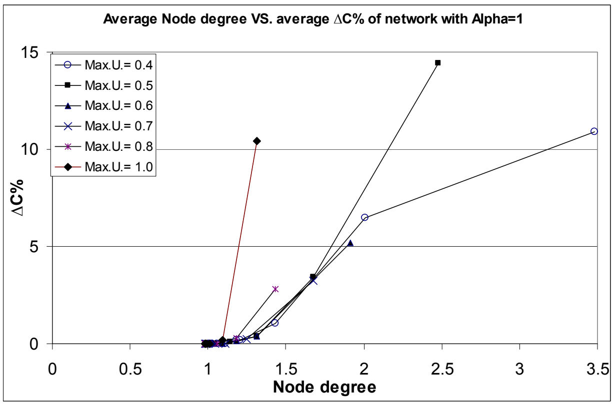

Fig.23 Average Δ*C* Vs. δ for Alpha=1

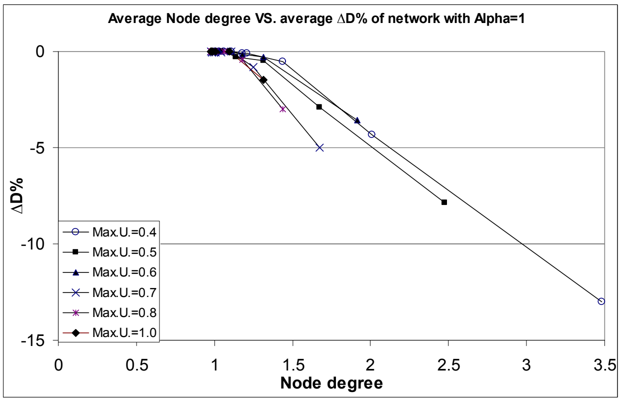

Fig.24 Average Δ*D* vs. δ for Alpha=1

Node Degree of Network with Alpha=0						
slack	max U.0.4	max U.0.5	max U.0.6	max U.0.7	max U.0.8	max U.1.0
0	2.705	2.295	1.965	1.795	1.655	1.49
0.1	2.795	2.485	2.06	1.895	1.715	1.545
0.2	2.96	2.625	2.22	2.02	1.8	1.615
0.3	3.23	2.785	2.48	2.145	1.925	1.7
0.4	3.615	3.01	2.62	2.46	2.145	1.875
0.5	3.965	3.4	2.98	2.715	2.435	2.07
0.6	4.845	4.315	3.585	3.14	2.825	2.54
0.7	6.44	5.29	4.315	3.95	3.585	2.98
0.8	9.345	7.18	6.445	5.29	4.845	4.315

Table.1. Node degree of networks with Alpha=0

Node Degree of Network with Alpha=1						
slack	max U.0.4	max U.0.5	max U.0.6	max U.0.7	max U.0.8	max U.1.0
0	1.005	0.99	0.985	0.98	0.98	0.98
0.1	1.03	1	0.985	0.985	0.98	0.98
0.2	1.06	1.01	0.995	0.985	0.985	0.98
0.3	1.11	1.045	1.005	0.995	0.985	0.98
0.4	1.18	1.095	1.03	1.005	0.995	0.985
0.5	1.2075	1.14	1.095	1.045	1.005	0.99
0.6	1.435	1.315	1.18	1.11	1.06	1.01
0.7	2.01	1.675	1.315	1.25	1.18	1.095
0.8	3.485	2.475	1.915	1.675	1.435	1.315

Table.2. Node degree of networks with Alpha=1

References:

[1] Aaron Kershenbaum, Parviz Kermani, and George A. Grover, MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing, *IEEE Transaction on Communications,* Vol.39, April 1991, pp.503-513.

[2] Robert Cahn, *Wide Area Network Design.* Morgan Kaufmann Publisher, San Francisco, CA, 1998.

[3] Bernard Fortz and Mikkel Thorup, Internet traffic engineering by optimizing OSPF weights, *in Proc. IEEE INFOCOM, vol. 2*, Mar. 2000, pp.519–528.

[4] Bernard Fortz, Jennifer Rexford, and Mikkel. Thorup, Traffic engineering with traditional IP routing protocols, *IEEE Commun. Mag., vol. 40*, Oct. 2002, pp. 118–124.

[5] Walid Ben-Ameur, Eric Gourdin, Bernard Liau, Nicholas Michel, "Designing Internet Networks"; *in Proceeding of DRCN2000*", pages 56-61, Munich, April 2000

[6] Robert Cahn, *The Design Tool: Delite (software)*, 1998, http://www.kt.agh.edu.pl/~mwa grow/projektowanie sieci telekomunikacyjnych/ DeLite/setupex.exe

[7] Andrew Makhorin. *GLPK (GNU Linear Programming Kit)*, 2000, http://www.gnu.org/ software/glpk/

[8] Aaron Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw-Hill, New York, NY, (1993).