Design A Hueristic Algorithm Module for DElite Network Design and Simulation Tool

Annop Monsakul
Faculty of Information Technology
Thai-Nichi Institute of Technology
Bangkok, Thailand
annop@tni.ac.th

***Abstract*— Backbone IP network design, to support both unicast and multicast traffic under delay constraints, is a difficult problem. Real network have considered cost, performance and reliability. Therefore, the simulator helps the network designer to testing the functionality of the network before implementation. This paper proposes a heuristic design algorithm called D-MENTOR that develops in programming based on Mesh Network Topological Optimization and Routing Version 2 (MENTOR-II) for made a new module of DElite tool. The simulation results show that, in almost all test cases, the proposed algorithm yields lower installation cost.**

Keywords*— *IP Network Design; MENTOR Algorithm; Unicast/Multicast Traffic; Traffic Engineering; DElite Tool

I. INTRODUCTION

Internet Protocol (IP) network design which concerns both unicast and multicast routing [1] [2] remains a difficult problem. The problem is even more challenging if choose to manage the traffic by the appropriate setting of link weights in the Open Shortest Path First (OSPF) protocol instead of using the overlay network technique. This kind of problem can be classified as Mixed Integer Linear Programming (MIP) [3].

To reduce the complexity of the network design process, Kershenbaum *et al.* [4] developed a heuristic algorithm, called MENTOR (Mesh Network Topological Optimization and Routing). The networks designed by this algorithm may be able to give near-optimal routing performance [5]. MENTOR can also be used to design virtual circuit switching and packet switching networks such as Asynchronous Transfer Mode (ATM) and frame relay. However, it cannot be directly used to design routers or Multiprotocol Label Switching (MPLS) networks [6] that employ OSPF or Intermediate-System-to-Intermediate-System (ISIS) routing protocol [7]. This is because MENTOR does not perform an appropriate link weight setting. Cahn [8] improved the MENTOR algorithm such that appropriate OSPF link weights [9] can be set during the design process using Incremental Shortest Path (ISP). This improved version is known as MENTOR-II. However, it should be noted that almost all the above design algorithms were developed for networks with only unicast traffic.

Presently, several important emerging multicast applications [10] such as distributed database systems, radio, television, video conferencing systems, distance learning systems, are becoming more and more popular. As a result, IP multicast traffic is increasing rapidly for almost all organizations. Therefore, IP network design process should effectively route the multicast traffic in addition to the traditional unicast traffic. Monsakul *et al*. [11] proposed a modified version of MENTOR-II (M-MENTOR) that aims at supporting mixed unicast and multicast traffic in IP networks with the following features: (1) all network members are within the same Autonomous System (AS); (2) the employed routing protocol should support multiple link weights such as Multi-Topology Extension to OSPF (M-OSPF); and (3) the multicast traffic from different sources share the same multicast tree.

In order to efficiently design communication networks with delay constraints, especially the networks that can be represented by M/M/1 model [12], this paper develops an algorithm based on MENTOR-II. Instead of fixing all design parameters as in the MENTOR-II, this algorithm determines the maximum utilization of a link based on its delay and capacity. This allows us to directly control the network delay. Here, the performances of networks designed by the proposed algorithm are evaluated in terms of installation cost and compared with those of networks designed by the MENTOR-II for various traffic demands and different numbers of nodes.

DEsign tool LITE (DElite) [15] is an educational and practical wide area network (WAN) design tool, which can produce network designs of limited size using a set of the embedded network design algorithms. DElite can produce graphical displays representing network nodes and links as well as some additional analysis data (delay analysis, reliability analysis, average delay analysis of the link, average number of hops, link utilisation for every separate link between nodes, utilisation of each node, overall network model utilisation, etc.). There are five files that are handled by the DElite tool for each network design. However, the most important for the users are .gen (original node information, coordinates and available link types) and .net (additionally has table of links between nodes i.e. actual design) files. Links between nodes can be generated using a few design algorithms. Thus, various designs may have different costs, delays, reliability, average number of hops etc., .cst file describes the costs associated with link types (e.g. T1, T3, D96 etc.), .req file describes capacity of each link type, .inp file contains the names of files related to a particular model. The users can edit all of the files mentioned above as they are in ASCII text format. The relationship of file for link direction of the arrow between file types for information, .net file is only .inp file and .inp file is using .gen, .cst and .reg as shown in Figure 1.

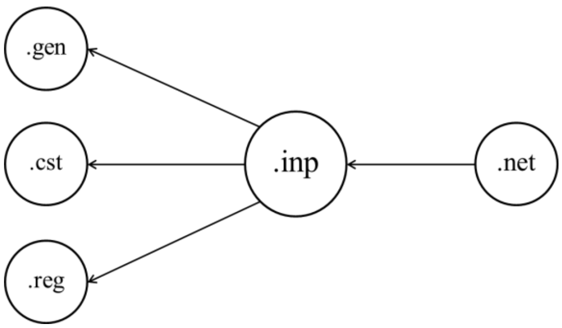

Fig. 1 Relationship of file used in DElite tool

The rest of the paper is organized as follows. In Section 2, the MENTOR-II algorithm and M/M/1 queuing delay are introduced. In Section 3, explain how the maximum link utilization and design D-MENTOR algorithm. In Section 4, an example of 6-node network design is given. In Section 5, given a maximum network delay of 5 ms and maximum link delays of 1.712 ms for 6-node networks, respectively, the cost of networks designed by the proposed algorithm is evaluated by the D-MENTOR algorithm.

II. BACKGROUND

A. MENTOR-II Algorithm

MENTOR-II is a low-complexity heuristic network design algorithm. The properties of IP networks designed by this algorithm are: 1) traffic is routed on relatively direct paths; 2) links have reasonable utilization; and 3) relatively high-capacity links are used.

Similar to the previous algorithm, MENTOR-II [4] starts with clustering network nodes and building a good spanning tree between backbone nodes. However, when considering adding a direct link to serve the traffic demand between a pair of nodes, MENTOR-II calculates an appropriate weight for this link by using Incremental Shortest Path (ISP) algorithm. The concept of MENTOR-II can be described as follows:

1) Set the weight for each link in the selected good spanning tree proportional to the installation cost of the link;

2) Let $d_{\text{spt}}(\text{A,B})$ be the shortest path distance between nodes A and B through the spanning tree, and consider adding a direct link between each pair of nodes in decreasing order of $d_{\text{spt}}(\cdot)$;

3) When considering whether to add a link L_{AB} between A and B, the weight w_{AB} of L_{AB} is initially set to a reasonably high value. ISP then tries to draw traffic flow through L_{AB} as much as possible by lowering w_{AB}. The constraint is that w_{AB} should be greater or equal to the installation cost;

4) L_{AB} is added if an eligible value of w_{AB} can be found and the amount of traffic flow though it falls in the reasonable zone defined by ρ, C_{AB}, and s.

When all possible direct links are considered, they are assigned with appropriate weights which ensure the shortest path routing.

B. M/M/1 Queuing Delay

M/M/1 delay is the time of traffic waits in a queue until router can be executed. Typically the arrival process is modeled as Poisson and exponential service times. In this paper, this focus on the problem of minimizing the installation cost with delay constraints such as the maximum link delay and maximum end-to-end delay, especially for networks that can be represented by M/M/1 model [14]. For this model, the average link delay is given by

$$T = T_p + T_q \tag{1}$$

where T_p is the propagation delay which depends on the link distance [16], and T_q is the average queuing delay:

$$T_q = \frac{P_s}{(1-U)C} \tag{2}$$

where P_s is the average packet size in bit, U is the link utilization, and C is the link capacity, i.e., for a network designed by MENTOR-II,

$$T_q \le \frac{P_s}{(1-\rho)C}. \tag{3}$$

III. DESIGN ALGORITHM

The MENTOR family allows us to efficiently construct good mesh networks. However, it does not give any idea of how to choose the design parameters, e.g., α, ρ, and s, to achieve the designed constraints. Hence, one may have to perform exhaustive search among all possible combination of such parameters to find the optimum solution.

It should be noted that, for the MENTOR, the design parameters such as ρ and P_s are kept constant for all links. As a consequence, a link with small capacity always suffers more delay than the one with large capacity. To avoid the large delay of the former link, one should try to keep ρ as small as possible. This may lead to inefficient utilization of a large-capacity link, which is more expensive. Therefore, from (1) and (3), instead of using the same value of ρ for all links, let ρ be determined by

$$\rho < \Lambda := 1 - \frac{P_s}{T\,C} \tag{4}$$

where T is the maximum allowable link delay of the overall network. Based on (4), a link with large capacity is allowed to have more efficient utilization for given average packet size and maximum allowable link delay.

Another advantage of using the variable maximum link utilization of (4) is that the MENTOR-II search domain can be reduced. Let $C_x = x\,C_1$ where C_1 is the capacity of a single-channel link, e.g., 10 Mbps in Table 1. From (4), the upper limit Λ_x of ρ_x for a link of capacity C_x can be written in terms of Λ_1 as

$$\Lambda_x = 1 - \frac{1-\Lambda_1}{x}. \tag{5}$$

In other word, changing the value of Λ_1 changes all the values of other Λ_x. As a result, in the search process, only Λ_1 is subjected to be varied to find the optimum solution. In comparison with the MENTOR-II, the optimum search domain of the maximum link utilization is reduced from all possible ρ in $(0,1)$ to $(0,\Lambda_1)$.

I have designed a backbone network by following the steps of D-MENTOR.

A. *Essential steps of D-MENTOR*

/*(1) Compute Median Node:

Med=Median(β)

/*(2) Build a Good Backbone Tree:

BuildTree(Med, α)

/*(3) Install traffic:

route_traffic ()

/*(4) Calculate link capacity and delay:

Link_Cap_Delay()

/*(5) Record the best path on the Backbone Tree:

Record_Tree()

/*(6) Construct array of node pair $P[i]$ sequence in decreasing order of shortest path delay:

Construct_P()

/*(7) Construct array of node ISP node pair $P[i][j]$:

Construct_ISP()

/*(8) Delay Constrant MENTOR-II:

D_MENTOR()

B. *Compute Median Node*

- Select Median Node: Median() :

  Select node n which maximize merit

$$merit(n) = \beta T_1 + (1-\beta) T_2 \tag{6}$$

$$merit(n) \le 1, \quad 0 \le \beta \le 1 \tag{7}$$

$$T_1 = \frac{[d_{\max}(g) - d(n,g)]}{d_{\max}(g)} \tag{8}$$

where g is the gravity point, center is the center point, and H is the length of the diagonal of smallest rectangular that cover all nodes.

C. *Build a Good Backbone Tree*

```
BuildTree(med_node)
    {
    /*Build Backbone Tree with Prim-Bellman's Algorithm
    /*med_node: root node
    /*alpha: PB Tree coefficient
        PB_Tree(med_node, alpha)
    /*Created adjacent node database
        adj_node()
    }
```

- Prim-Bellman's Algorithm (no split) -1:

  Initialization Reset all $sp\_dist[][]$

  $Y = \{y_1, y_2, ...\}$ set of nodes to be considered each round.

- Calculate adjacent nodes:

  $adj[m] = \{adj[m][i]\}$ set of adjacent nodes of node m.

```
adj_node()
for (i=1; i<=|N| ; i++){
    k=1;
    for (j=1; j<= |N|; j++) {
        if (i != j)
          if w[i][ j] !=MAX
            adj[i][ k]= j              /* adj[i] ∪  j */
    k++;
    }
    n-adj[i]= k
}
```

D. Install traffic

- Install traffic: route_traffic () :
  Initialization Reset all $sp\_dist[][]$

  $Y = \{y_1, y_2, \ldots\}$ set of nodes to be considered each round.
- loading() :
  Find initial Unicast and Multicast load between pairs as shown in Figure 2.

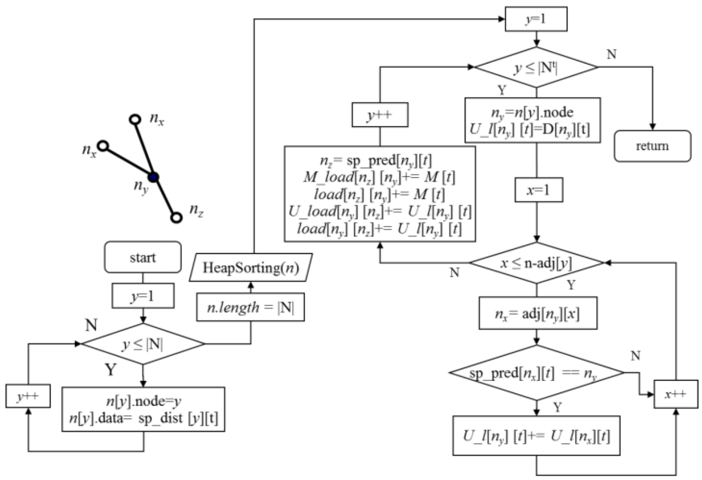

Fig. 2 Loading Mixed Traffic between pairs.

E. Calculate link capacity and delay

```
Link_Cap _Delay()
/*Calculate Link Speed and Link Delay on Network */
for ( x= 1; x <= |N| ; x++) {
for ( y= 1 ; y <= |N| ; y++) {
        If (x < y)&& w(x,y) != MAX {
                load = max ( load[x][y], load[y][x] )
                C[x][y] = Link_Cap(x,y,load)
                C[y][x] = C[x][y]
                U[x][y] = load[x][y] / C[x][y]
                        U[y][x] = load[y][x] / C[y][x]
                t_p = PROP_D*dist[x][y]
                t_r  = PKT_SIZE/C[x][y]
                del[x][y] = per_co(t_p+t_r/(1-U[x][y])+T_s)
                del[y][x] = per_co(t_p+t_r/(1-U[y][x])+T_s)
        }
}
}
```

F. Record the best path on the Backbone Tree

```
/*Record the best path on the Backbone Tree:
        Record_Tree()
                for(i=1; i <= |N|; i ++ ){
                        for(j=1; i <= |N|; j ++ ){
                                if i! = j
                                t_dst[i][j] = sp_dist[i][j]
                                t_nxt[i][j] = sp_prd[i][j]
                        }
                }
```

G. Construct array of node pair $P[i]$ sequence in decreasing order of shortest path delay

- Since this assume all unicast traffic are symmetric, consider only set node pairs :

$$NP = \{(x, y)\} \textbf{ s.t. } x < y \text{ for all } x, y \in N \quad (8)$$

- Compute :

  $sp\_del(x, y)$ for all node pair x, y of the initial tree.
- Sequence node pairs in decreasing order of shortest path delay $sp\_del(s,t)$:

$$P[1], P[2], \ldots, P[X], \ldots, P[|NP|] \quad (9)$$

where $P[i] = \{s, t\} \quad ; s, t \in NP$ and that the first X pairs are node pairs that unsatisfied their delay constraint, i.e.

$$sp\_del(s_i, t_i) \geq req-del[s_i][t_i] \quad (10)$$

where $i = 1, 2, \ldots, X$

- Array $P[i]$:

$$P_{[i]} = \{s, t\} \quad ; s, t \in N \quad (11)$$

where i^{th} node pair (s,t) sequenced in decreasing order of delay.

- Calculate shortest path delay between x and y :

```
sp_del (x,y) {
/*Calculate shortest path delay between x and y*/
If del[x][y] != MAX
   return(del[x][y])
d=del(x,sp_prd[x][y]) + sp_del(sp_pred[x][y], y)
return(d)
}
```

H. Construct array of node ISP node pair $P[i][j]$

- For each pair $P[i]$, computes a set of ISP node pairs :

$$\Pi[i] = \{\Pi[i][1], \Pi[i][2], \ldots\} \quad (12)$$

where $\Pi[i][j] = \{x_{i,j}, y_{i,j}\} \quad ; x_{i,j}, y_{i,j} \in NP$

- ISP node pairs in $\Pi[i]$ are sequenced such that :

$$url[i][j] \geq url[i][j+1] \quad (13)$$

where $url[i][j]$ is the upper bound distance of $\Pi[i][j]$.

- Since $url[i][j]$ is define as :

$$url[i][j] = t\_dst[x_{i,j}][y_{i,j}] - t\_dst[x_{i,j}][x_i] - t\_dst[y_j][y_{i,j}] \quad (14)$$

where traffic from $x_{i,j}$ to $y_{i,j}$ can be drawn to the new direct link between x_i, y_i for $P[i]$, if the metric $w[x_i][y_i]$ of the new link is less than $url[i][j]$.

$\Pi[i][j] = \{m,n\}$; $n \in N$ and that j^{th} ISP node pair (m,n) of $P[i]$ sequenced in decreasing order of the upper bound distance.

$url[i][j]$ is the upper bound distance of $\Pi[i][j]$.

I. Delay Constraint MENTOR-II

- Consider $P[i]$ in increasing order.
- x = Install Link $(P[i])$:

If $x = i$, install a link between $P[i]$ and visit next pair.

If $x \le 0$, impossible to satisfy delay requirement break.

If $x \le i$, fallback to $P[x]$.

- Consider Installing a link on $P[i]$:

Draw as most traffic as possible to link between $P[i]$, starting with set the link weight $(w[x_i][y_i])$ to the minimum $(j = \Pi[i].length)$ and increasing order as shown in Figure 3.

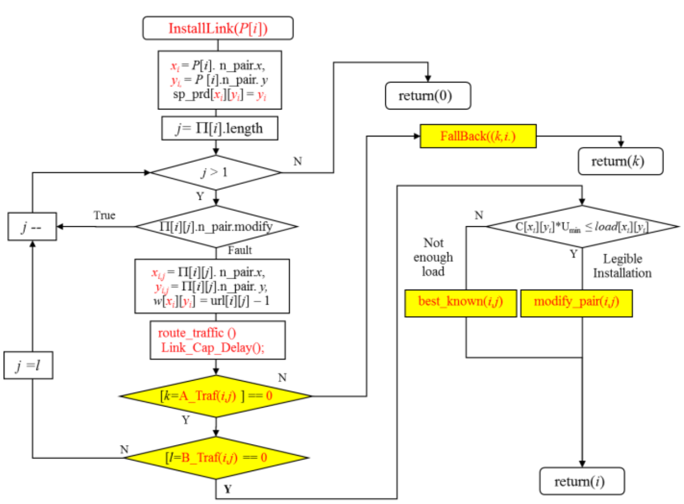

Fig. 3 Installing a link *on* $P[i]$

IV. Performance Evaluation

A. Setup

In order to evaluate the efficiency of the D-MENTOR, this analyze a number of design results for synthesized requirements in terms of installation cost with various delay constraints. To explore the effect of the number of nodes on network performance, 6-node design requirement sets as shown in Figure 4 and total multicast flows as shown in Figure 5.

Requirement set, which is synthesized by a design tool called DElite [8], includes a random node distribution and the associated traffic demand matrices. For all node distributions, the maximum node distance is limited to 100 km. The unicast traffic demands for each requirement set are also generated by DElite with the following assumptions :

1) All nodes have the same total amount of unicast traffic in and unicast traffic out, denoted by *Traff*;

2) The unicast traffic between any pair of nodes is inversely proportional to the distance between P_s them.

To observe the effect of the amount of traffic on the design performance, traffic demand matrices for *Traff* of 50 Mbps are generated for each node distribution. Each backbone node has multicast traffic of 128 kbps delivered to all other nodes. Let P_s be 12,288 bits. The goal of the network design is to find the network with minimum installation cost, given that the maximum end-to-end delay is 5 ms and maximum link delay is 1.715 ms, and Λ_1 = 0.3. Table 1 shows the installation cost of 6-Mbps channel links between all possible node pairs.

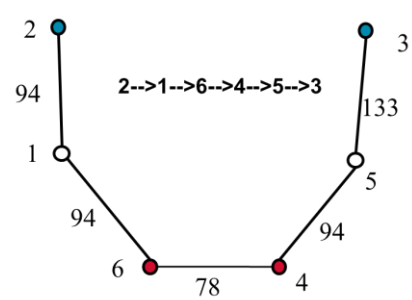

Fig. 4 The 6-nodes on network performance

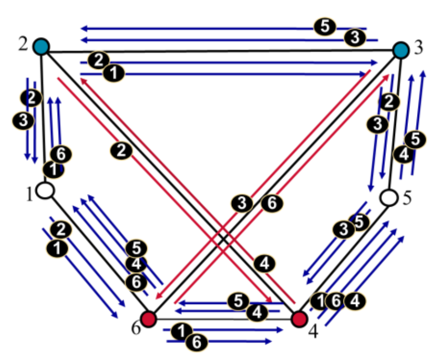

Fig. 5 Total Multicast Flows

TABLE I. UTILIZATION, LOAD, AND CAPACITY OF INSTALLATION COST

Node		*Uni_ Traff.*	*Multi_ Traff.*	*Link Cap.*	*U*	*Delay[x][y]*	*Cost*
N6	N4	5962	384	12,288	0.52	2.54	29,646.78
N4	N6	5962	384	12,288	0.52	2.54	30,368.48
N6	N1	3333	512	7,168	0.54	4.25	30,596.81
N1	N6	3333	256	7,168	0.5	3.98	28,786.61
N5	N4	6121	256	12,288	0.52	2.63	31,713.79
N4	N5	6121	512	12,288	0.54	2.72	30,253.76
N5	N3	2526	640	6,144	0.52	4.87	30,484.83
N3	N5	2526	128	5,120	0.52	5.73	31,782.43
N2	N1	2737	128	5,120	0.56	6	27,651.73
N1	N2	2737	640	6,144	0.55	4.99	30,986.59

B. Results

D-MENTOR module is developing by visual C# as shown in Figure 6. It request open file .req, .cst and .gen, that generated by DElite tool. Thus, D-MENTOR calculated all-parameter of a network design and create file .net for DElite build graph tree of a new network topology. The simulation results as shown Table I – Table III.

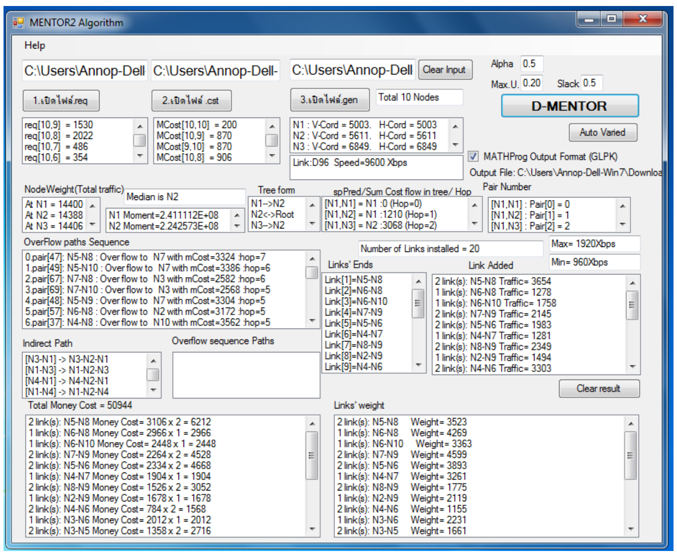

Fig. 6 D-MENTOR Module

V. CONCLUSION

In this paper, to cope with low installing cost and delay constraints for communication networks that can be represented by M/M/1 model, the upper limit of maximum link utilization has been introduced in D-MENTOR in terms of search space can be reduced by factor Λ_1, the upper limit of maximum link utilization for a single-channel link. To evaluate the performance of the proposed algorithm, various distributions of 6 networks nodes have been generated. The network designed by D-MENTOR is a better performance in terms of installation cost, especially when the maximum link delay is smaller than the maximum end-to-end delay. This performance improvement tends to decrease as the former delay approaches the latter delay. However, the majority of networks designed by D-MENTOR still achieve lower installation cost when the maximum link delay is close or equals to the maximum end-to-end delay.

TABLE II. LINK PATH, LOAD, AND DELAY

Srce	*Dest*	*Link Path*	*Origjnal Path*	*Uni_ Traff*	*Multi_ Traff*	*Delay*	*Cost*
N6	N5	6-->4-->5		2605	128	6.53	27,934.61
N6	N4	6-->4		566	128	3.17	29,242.68
N6	N3	6 - > 3	6-->4-->5-->3	513	128	6.51	27,508.46
N6	N2	6-->1-->2		497	128	16.36	27,725.92
N6	N1	6-->1		558	128	6.54	28,162.08
N5	N6	5-->4-->6		2605	128	6.45	29,069.79
N5	N4	5-->4		765	128	3.27	28,552.80
N5	N3	5-->3		617	128	12.63	27,269.08
N5	N2	5 -> 3 -> 2		382	128	12.63	28,097.17
N5	N1	5-->4-->6-->1		460	128	12.98	29,160.31
N4	N6	4-->6		566	128	3.17	29,446.08
N4	N5	4-->5		765	128	3.35	30,368.98
N4	N3	4-->5-->3		104	128	9.62	30,296.31
N4	N2	4 - 2	4-->6-->1-->2	72	128	10.63	27,686.68
N4	N1	4-->6-->1		72	128	9.71	27,913.79
N3	N6	3 -> 6	3-->5-->4-->6	513	128	6.51	27,853.72
N3	N5	3-->5		617	128	8.69	30,684.23
N3	N4	3-->5-->4		104	128	11.96	28,582.40
N3	N2	3-->2		1142	128	6.36	30,351.13
N3	N1	3 -> 2 -> 1		150	128	15.37	30,926.58
N2	N6	2-->1-->6		497	128	15.19	28,798.49
N2	N5	2 -> 3 -> 5		382	128	15.05	28,377.47
N2	N4	2 - 4	2-->1-->6-->4	72	128	10.63	27,508.46
N2	N3	2->3		1142	128	6.36	28,213.82
N2	N1	2-->1		644	128	9	28,162.08
N1	N6	1-->6		558	128	6.54	29,069.79
N1	N5	1-->6-->4-->5		460	128	12.71	28,552.80
N1	N4	1-->6-->4		72	128	9.36	27,269.08
N1	N3	1 -> 2 -> 3		150	128	16.18	28,912.41
N1	N2	1-->2		644	128	9.82	28,351.13

TABLE III. N-CHANEL, LOAD, AND CAPACITY OF DELAY

Srce	*Dest*	*Uni_ Traff*	*Multi_ Traff*	*UTmix*	*N-chan*	*Link Cap*	**Delay**	*Cost*
N6	N4	3775	256	4031	7	7168	3.08	29,242.68
N6	N1	1659	384	2043	4	4096	4.54	28,162.08
N5	N4	3934	384	4318	8	8192	2.66	28,552.80
N5	N3	1103	256	1359	3	3072	5.53	27,269.08
N4	N6	3775	256	4031	7	7168	3.08	29,446.08
N4	N5	3934	384	4318	8	8192	2.66	30,368.98
N3	N5	756	256	1012	2	2048	8.65	30,684.23
N2	N1	1363	256	1619	3	3072	6.19	28,162.08
N1	N6	1659	256	1915	4	4096	4.31	29,069.79
N1	N2	1363	256	1619	3	3072	6.19	28,351.13
N2	N3	1674	256	1930	4	4096	4.47	28,213.82
N3	N2	1674	256	1930	4	4096	4.47	30,351.13
N3	N6	513	128	641	2	2048	6.51	27,853.72
N6	N3	513	128	641	2	2048	6.51	27,508.46
N4	N2	72	128	200	1	1024	10.63	27,686.68
N2	N4	72	128	200	1	1024	10.63	27,508.46
Average 6-node								**28,651.95**

REFERENCES

[1] Jonathan L. Wang, "Traffic routing and performance analysis of common channel signaling system 7 network", *Proc. GLOBECOM*, vol. 1, pp. 301-305, AZ, USA, 1991.

[2] Weihua Hu, Jia Lv, and Hongyuan Hou, "An improved strategy of PIM-SM protocol registered efficiency based on recombinant and slice", *Advances in Information Sciences and Service Sciences*, vol. 4, no. 23, pp. 396-402, 2012.

[3] Pisit Charnkeitkong, Computer Network Design - Requirement Analysis, Architecture and Topology Design, 1st Edition. Rangsit University Press, 2007 [in Thai].

[4] Aaron Kershenbaum, Parviz Kermani, and George A. Grover, "MENTOR: An algorithm for mesh network topological optimization and routing", *IEEE Transactions on Communications*, vol. 39, no. 4, pp. 503-513, 1991.

[5] Kairat Jaroenrat and Pisit Charnkeitkong, "On routing performance of MENTOR algorithm", *WSEAS Transaction on Communications*, issue 9, vol. 5, pp. 1769-1776, 2006.

[6] Ning Wang and George Pavlou, "Traffic engineered multicast content delivery without MPLS overlay", *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 619-628, 2007.

[7] Tony Przygienda, Naiming Shen, and Nischal Sheth, "M-ISIS: Multi-topology (MT) routing in IS-IS 2005 [Online]". Available: draft-ietf-isis-wg-multitopology-11.txt

[8] Robert S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, Morgan Kaufmann Publisher, San Francisco, CA, 1998.

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

[10] Nicholas F. Maxemchuk, "Video distribution on multicast networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 357-372, 1997.

[11] Annop Monsakul and Pisit Charnkeitkong, "M-MENTOR: A design algorithm for IP networks with mixed traffic", *WSEAS Transaction on Communications*, issue 10, vol. 8, pp. 1086-1095, 2009.

[12] Alireza Dehestani and Pedram Hajipour, "Comparative study of M/Er/1 and M/M/1 queuing delay models of the two IP-PBXs", *Journal of Convergence Information Technology*, vol. 5, no. 2, pp. 36-42, 2010.

[13] Kairat Jaroenrat and Pisit Charnkeitkong, "Enhanced MENTOR algorithm with ECMP routing enable", *International Journal of Digital Content Technology and Its Applications*, vol. 7, no. 7, pp. 10-20, 2013.

[14] Pedram Hajipour, Leila Mohammadi and Mohammad Reza Keshavarzi, "Measurements and comparative of M/M/1 and M/D/1 queuing models of resource management in satellite systems", *Journal of Convergence Information Technology*, vol. 6, no. 5, pp. 99-107, 2011.

[15] Muhammad Azizur Rahman, Algris Pakstas and Frank Zhigang Wang, "An Integerated Enviroment for Network Design and Simulation", *Advance in Computer, Information, and System Science, and Engineering(CISSE05)*,pp.315-322,2005.

[16] Pisit Charnkeitkong, Annop Monsakul, Adisorn Kheaksong and Kiattisak Maichalernnukul, "A Modified MENTOR II for Mesh IP Network Design with Link Delay Constraint," *Journal of Convergence Information Technology (JCIT).*, vol.9, no 2, pp. 167-180, 2014.