Optimal Delay-Constrained Minimum Cost Loop Algorithm for Local Computer Network

Yong-Jin Lee
Dept. of Computer Science
Woosong University
yjlee@wsu.ac.kr

M. Atiquzzaman
School of Computer Science
University of Oklahoma
atiq@ou.edu

Abstract

This study deals with the Delay-Constrained Minimum Cost Loop Problem (DC-MCLP) of finding several loops from a source node. The DC-MCLP consists of finding a set of minimum cost loops to link end-nodes to a source node satisfying the traffic requirements at end-nodes and the required mean delay of the network. In the DC-MCLP, the objective function is to minimize the total link cost. This paper proposes a dynamic programming based three phase algorithm that solves the DC-MCLP. In the first phase, the algorithm generates feasible solutions to satisfy the traffic capacity constraint. It finds the capacitated minimum loops in the second phase, and allocates the optimal link capacities to satisfy the mean delay constraint in the third phase. Performance evaluation shows that the proposed algorithm has good efficiency for network with less than thirty nodes and light traffic. Our proposed algorithm can be applied to any network regardless of its configuration, and used for the topological design of FDDI and SONET.

1. Introduction

Network topology affects important factors such as the communication cost and transmission speed of a network. Thus, topology discovery is one of important research areas in computer networks [2,4,8,10].

Modern computer networks consist of backbone networks that serve as the major highways to transfer large volumes of communication traffic, and local networks that feed traffic between the backbone node and end user nodes connected to the backbone.

In a local network, the total traffic volume of end-user nodes that can be served by a port of the backbone node is limited. So, the local network consists of a backbone node (source) and several trees or rings that cover all end user nodes to satisfy the constraints on traffic volume. Topology discovery is required to find all the trees or loops in a local network which satisfies the constraints.

Issues related to topology discovery for local network has been classified into two problems in the literature: capacitated minimum spanning tree problem (CMSTP) and minimum cost loop problem (MCLP) [5]. The CMSTP finds the best way in the least cost aspect to link end-user nodes to a source node. It determines a set of minimal spanning trees with a capacity constraint. In the MCLP, end-user nodes are linked together by a loop that is connected to a port in the backbone node (switch). The MCLP is known to be NP-hard [9], and consists of finding a set of minimum cost loops rooted at the source node satisfying the traffic constraint only. However, communication delay between end nodes deteriorates quality of service of users as the network size grows. To solve this issue, it is necessary to consider the traffic capacity and the mean network delay constraints together. However, no previous work has been reported on this problem. We, therefore, present the Delay-Constrained Minimum Cost Loop Problem (DC-MCLP).

A similar problem is the vehicle routing problem (VRP) in the transportation science. However, existing algorithms for the MCLP and VRP [1,6] do not consider network mean delay. Without modification, they cannot be applied to the DC-MCLP which has additional constraint, that is, the mean delay of network should be within the desired time. Although algorithms [3,7,11,12] for the topological design of backbone networks consider the network mean delay, they do not consider the traffic capacity constraint. Thus, they are not useful for the DC-MCLP.

The objective of this paper is to formulate the DC-MCLP and to develop an exact algorithm to solve the problem. We propose a dynamic programming-based exact algorithm for the solving the DC-MCLP. Our proposed algorithm solves the DC-MCLP in three phases: In the first phase, the algorithm uses dynamic programming to generate feasible solutions to satisfy the traffic capacity constraint. In the second phase, it finds capacitated minimum cost loops based on the matching procedure. In the final phase, the algorithm assigns link capacities to the set of capacitated loops optimally in order to satisfy the desired mean network delay constraint.

Our performance evaluation results demonstrate that our proposed exact algorithm can be effectively applied to problems where the number of nodes is less than thirty and traffic volume is light.

The main contributions of this paper are: (i) formulation of the DC-MCLP, (ii) proposing and evaluating its exact solution, and (iii) determining the threshold in choosing between heuristic methods and the exact algorithm depending on the size of the local network.

The suggested algorithm can be applied to the design of SONET (Synchronous Optical Network) and FDDI (Fiber Distributed Data Interface) network.

The rest of the paper is organized as follows. Section 2 describes the mathematical formulation of the DC-MCLP. Section 3 describes our proposed dynamic programming based modeling and exact algorithm for the DC-MCLP. Section 4 evaluates the computational complexity of our proposed algorithm using an analytical model. Finally, concluding remarks are given in Section 5.

2. Formulation of the DC-MCLP

We consider the graph G = (V,E). V and E represent set of nodes and edges respectively. In this paper, we use edge and link in the same meaning. Set of nodes, V is composed of (n+1) nodes. Indexes of nodes are numbered as {0, 1, 2,.., n}. Source node (node 0) can be viewed as an object such as the backbone router with several ports. Now, we want to construct loops connecting to end-user nodes (1,...,n) satisfying the delay constraint. End-user node can be switching hub or host and originates traffic.

While the MCLP deals with traffic capacity only, the DC-MCLP deals with both the traffic capacity and network mean delay constraints simultaneously. The objective of the DC-MCLP is to find a collection of least-cost loops rooted at the source node. Since the number of nodes a port can serve is limited in a real environment, several loops have to be found. In addition, network mean delay constraint, a Quality of Service (QoS) parameter of end-user nodes, is added to the problem. In this case, we have to consider the capacity allocation for links included in the loops to meet the required mean delay constraint. In the DC-MCLP (see Fig. 1), the computed mean network delay (T) should be less than equal to the predefined threshold (Delay). In the rest of this section, we formulate the DC-MCLP.

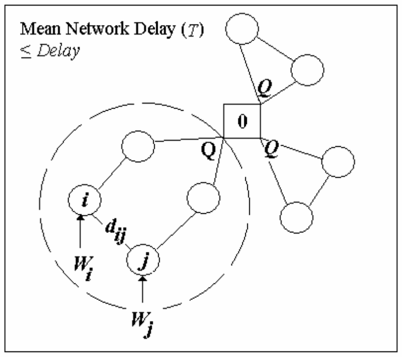

Fig. 1. DC-MCLP

2.1. Terminology

The following terminology will be used to formulate the DC-MCLP problem:

n: the number of nodes excluding the source node
0: the index of source node
m: the number of links included in any topology obtained in the progress of the DC-MCLP algorithm
N': set of node indexes excluding 0 = {1,2,..,n}
N_j: set of node indexes excluding j = N'-{j} = {1,2,..,j-1,j+1,..,n}
d_{ij}: distance (cost) from node i to node j
D: distance (cost) matrix
d: unit cost of link capacity
d_k: traversal distance cost of link k in topology obtained by the second phase of DC-MCLP algorithm (k=1,..,m)
S: subset of N_j composed of k nodes (cardinality of $S = k$)
Q: maximum traffic served by single loop
W_i: traffic amount generated at each node i (i=1,2,..,n)
$f_k(j,S)$: the minimum distance (cost) for a loop which covers set S composed of k nodes from the source node to node j to satisfy the constraint Q (k represents the number of nodes included in the set, S)
P_m: set, $\{j\} \cup S$ which is composed of the same elements ignoring the order
$f'_k(P_m)$: the minimum cost among $f_k(j,S)$ corresponding to the set, P_m
R_m: optimal node sets corresponding to $f'_k(P_m)$ considering the order
$f(R_m)$: cost corresponding to R_m
R: set of optimal loops
T: mean delay time of network (sec)
T_k: mean delay time of link k (sec)
v: total traffic between source-destination pairs

$1/\mu$: average packet length (bits/packet)
C_k: the capacity of link k obtained by the second phase of DC-MCLP algorithm (k=1,..,m) (bits/sec)
λ_k: traffic flow on link k (packets/sec) (k=1,..,m)
$link_k$: load of link k (=λ_k/C_k) (packets/bit) (k=1,..,m)
Delay: desired mean delay time of network (sec)
D_{cost}: total link cost in the third phase of DC-MCLP algorithm
F: optimal cost

2.2. Problem formulation

We formulate the DC-MCLP in order to satisfy the mean delay constraint (total network mean delay time is less than the desired mean delay) as well as satisfying the traffic capacity limitation imposed by the existing MCLP. Generally, the mean delay time (T) of the network is given by Eq. (1).

$$T = \frac{1}{\nu}\sum_{k=1}^{m}\lambda_k T_k \tag{1}$$

Since each link k can be regarded as an independent M/M/1 queue, mean delay time (T_k) at link k, from queuing theory, is given by Eq. (2).

$$T_k = \frac{1}{(\mu C_k - \lambda_k)} \tag{2}$$

Substituting Eq. (2) into Eq. (1) gives the network mean delay

$$T = \frac{1}{\nu}\sum_{k=1}^{m}\lambda_k[\frac{1}{(\mu C_k - \lambda_k)}] \tag{3}$$

The total link cost is assumed to be a linear function as given by Eq. (4).

$$D_{cost} = \sum_{k=1}^{m} dC_k d_k \tag{4}$$

Having defined the network mean delay (Eq. (3)) and the objective function (Eq. (4)), the DC-MCLP can be formulated by Eqs. (5)-(9).

The objective function of the DC-MCLP is to find a collection of loops with minimal link cost (Eq. (5)). Constraints of the DC-MCLP are: (i) Average traffic flow on the link should be smaller than capacity of the link (Eq. (6)), (ii) Mean delay time of network has to be dropped within allowable mean delay time threshold (*Delay*) (Eq. (7)) and, (iii) Maximum traffic flow on one loop must be below Q (Eq. (8)). If the number of nodes that a loop takes charge of should be smaller than a specific value, we change W_i to 1, $\forall i$ in Eq. (8). Eq. (9) represents that m links have to be included in the solution.

$$\textit{Minimize} \quad D_{cost} = \sum_{k=1}^{m} dC_k d_k \tag{5}$$

S.T.

$$\frac{\lambda_k}{\mu} \le C_k \tag{6}$$

$$T = \frac{1}{\nu}\sum_{k=1}^{m}\lambda_k[\frac{1}{(\mu C_k - \lambda_k)}] \le Delay \tag{7}$$

$$\sum_{i \in R_j} W_i x_i \le Q \tag{8}$$

$$\sum x_i = n \tag{9}$$

$$x_j = 0 \text{ or } 1$$

3. Solution of the DC-MCLP

Having formulated the DC-MCLP in the previous section, we now develop solution for the problem using dynamic programming. Our solution consists of three phases: path generation, matching, and link capacity allocation as described below.

3.1. Path generation phase

To solve the DC-MCLP using dynamic programming, we define stage variables and state variables.

- Stage variable, k (k=1,2…), is the number of nodes to form a path rooted at the source node to any arbitrary node j.
- State variables, j and S are the node index to be connected and the set of node indexes included in the path in order to connect node j, respectively. Then, using the principle of optimality, the recurrence relation can be obtained as shown in Eq. (10).

$$\begin{aligned} &\textit{If} \sum_{q\in S} W_q + W_j \le Q, \\ &\quad f_k(j,S) = \min_{q\in S, q\neq j} [f_{k-1}(q, S-\{q\}) + d_{qj}] \\ &\quad (k = 1,2\cdots, S \subseteq N_j) \\ &\textit{else} \\ &\quad f_k(j,S) = \infty \end{aligned} \tag{10}$$

$f_k(j,S)$ is set to infinity when the sum of all the traffic exceeds Q. Since BC (boundary condition) represents the cost to connect from the source node to node j directly without intermediate nodes, BC is defined by Eq. (11).

$$f_0(j,-) = d_{0j} \tag{11}$$

To obtain a feasible solution, we compute $f_1(j,S)$ for all (j,S) satisfying $\sum_{q\in S} W_q + W_j \le Q$ by using f_0. Then, $f_2(j,S)$ is computed using f_1. By repeating this procedure, we reach the phase where for all (j,S), $\sum_{q\in S} W_q + W_j > Q$.

Since $f_k(j,S)$ are infinity for all (j,S) at such a phase, we set L=k. This means that the loop cannot be extended any further. So, paths obtained at the previous stage k (0,1,2,..,L-1) are feasible solutions.

3.2. Matching phase

At stage k of the path generation phase, $f_k(j,S)$ represents the cost of the path that is composed of the same elements as $j \cup S$, but the order of elements included in the set is different.

Among $f_k(j,S)$, the minimum cost, $f'_k(P_m)$ is computed. That is, the cost for P_m at stage k, $f'_k(P_m)$ is as follows.

$$\begin{aligned} f'_0(P_m) &= f_0(j,-) + d_{j0}, & k &= 0 \\ f'_k(P_m) &= \text{Min}[f_k(j,S) + d_{j0}], & k &= 1,2,..,\text{L-1} \\ &\forall\ (j,S) \text{ such that } P_m - \{\{j\} \cap S\} = \phi \end{aligned} \tag{12}$$

The value of $f'_k(P_m)$ from Eq. (12) represents the cost of the loop including the connection cost to the centre node, which is composed of the same node indexes of different order. Node set of the optimal policy, R_m corresponds to $f'_k(P_m)$. R_m is the set of node indexes included in the loop rooted from the source node and represents the node sequence of loop by adding the centre node(0) to the both end-side indexes. Of course, node 0 is not included in R_m.

Finally, since n nodes have to be included in any loop (R_m) rooted from the source node without duplicate inclusion in the optimal solution, the optimal solution can be obtained by substituting $f(R_m)$ for $f'_k(P_m)$ from Eq. (12). There can be several collections which have the element, R_m to satisfy the split condition of set N'. Thus, the global optimal value, F, is the least value among the cost, $f(R_m)$ corresponding to R_m and can be expressed by Eq. (13).

$$\begin{aligned} &F = \text{Min}\,[\textstyle\sum f(R_m)\,], \\ &\forall\ R_m \text{ such that } \cup R_m = N' \text{ and } R_i \cap R_j = \phi\ \ (i \ne j) \end{aligned} \tag{13}$$

3.3. Link capacity allocation phase

In this phase, we use fixed routing to first find the mean arrival rate (λ_k: k=1,2,..,m) of links on the capacitated loop. The reason to use fixed routing is that, in order to transfer the traffic to the backbone node, each node in the loops has to send it to the neighbor node on the loop obtained in the previous phase.

The link capacities are allocated temporarily as the maximum value to meet the average flow. The mean network delay (T) is computed from Eq. (1), and the total cost is computed by using Eq. (4). Using the network mean delay, link capacities are allocated optimally by

$$C_k = \frac{\lambda_k}{\mu}\left[1 + \frac{1}{vT}\sum_{j=1}^{m}\frac{\sqrt{d\lambda_j d_j}}{\sqrt{d\lambda_k d_k}}\right] \tag{14}$$

The load of link k is defined as

$$link_k = \lambda_k / C_k \tag{15}$$

If the computed mean delay (T) is greater than the allowable delay (*Delay*), T is decreased by increasing the capacity of the link with the maximum load to meet the delay constraint. If T is less than *Delay*, it is increased by decreasing the capacity of the link with the minimum load. In this case, the link capacity can not be decreased below the average link flow (λ_k/μ). If the capacity of the link to be decreased is less than the average link flow, the capacity of the link with the next minimum load is decreased. By repeating the above procedure, the network cost (D_{cost}) corresponding to the the delay (T) which is close to the allowable mean delay (*Delay*) can be obtained.

3.4. Optimal DC-MCLP algorithm

From the above model, the optimal DC-MCLP algorithm is described as the following.

```
ALGORITHM Optimal DC-MCLP
PHASE 1: Path generation
STEP 1: initialization (For $k = 0$)
          compute boundary condition,
             $f_0(j,-) = d_{0j}$, for each $j$, $j \notin N'$
STEP 2: for $k$=1,2,3,…
          (1) check the feasibility and compute
              recurrence relation - Eq. (10)
```

```
        (2) end condition check
            if $f_k(j,S) = \infty$, $\forall$ $(j,S)$ let L = $k$,
                go to PHASE 2
            else repeat STEP 2
PHASE 2: Matching
STEP 1: for $k = 0$, For each $j, j \notin N'$,
        find $f'_0(P_m) = f_0(j,-)+d_{j0}$
STEP 2: for $k$=1,2,..,L-1,
        for $j$ such that $j < \underset{q\in S}{Min}\{q\}$, let $P_m = \{j\} \notin S$,
        find $f'_k(P_m)$ = Min $[f_k(j,S)+d_{j0}]$,
                $\forall$ $(j,S)$ such that $P_m - \{j \notin S\} = \phi$
STEP 3: find Min $[\sum f(R_m)]$, $\forall$ $R_m$
        such that $\cup R_m = N'$ and $R_i \cap R_j = \phi$ $(i \neq j)$
STEP 4: find loops corresponding to Min $[\sum f(R_m)]$
PHASE 3: Link Capacity Allocation
STEP 1: add $W_i$ for each $i$ included in $R_j$ up to $C_k$.
        compute $\lambda_k$ by using the fixed routing.
STEP 2: compute the mean network delay ($T$) by using
        Eq. (3).
        if $T = Delay$, go to STEP 3.
        if $T > Delay$, decrease the capacity of the link
          with the maximum load and repeat STEP 2.
        if $T < Delay$, decrease the capacity of the link
          with the minimum load and repeat STEP 2.
STEP 3: compute the network cost ($D_{cost}$) by using Eq.
        (4).
STEP 4: set $F = D_{cost}$. find the optimal set of loops ($R$)
        corresponding to $F$.
```

4. Performance evaluation

Having formulated the DC-MCLP and its solution in sections 2 and 3, in this section, we evaluate the performance of the algorithm in terms of its computational complexity.

4.1. Computational complexity analysis

We consider the amount of computation in the stage variable(k). It is maximum when the traffic requirement at each node is one (W_i =1, $\forall$ i) and the maximum traffic per single loop is Q. First, for any stage (k), $f_k(j,S)$ must be computed for $_kC_{n-1}$ different (j,S) pairs. Since such computation requires k additions and k-1 comparisons, where k = 1 to L, the number of additions and comparisons are represented by Eqs. (16) and (17), respectively.

$$\text{The number of additions} = n(n-1)2^L \quad (16)$$

$$\text{The number of comparisons} = n(n-2)2^L \quad (17)$$

4.2. Relation between maximum Q and n

Table 1 depicts that the number of nodes and maximum Q corresponding to the sum of the number of additions and comparisons for number of nodes (n) being 30 and Q = 29. The labels on the curve show the maximum value of Q.

In Table 2, we observe that the theoretical number of computations for (n=30, Q=29) and (n=100, Q=25) are almost the same, but the real execution time is different. This is because $_kC_{n-1}$ storage spaces are required in each stage k to store $f_k(j,S)$, i.e. as the number of nodes becomes large, memory access time increases sharply because the main memory cannot maintain all the results from the previous computation.

Table 1. Relationship between maximum Q and n

number of nodes (n)	30	50	70	100
maximum Q	29	27	26	25

4.3. Mean Execution Time

Table 2 shows the mean execution time of our proposed algorithm for three different cases $(Q = \frac{1}{2}\sum_{i=1}^{n} W_i$, $Q = \frac{1}{3}\sum_{i=1}^{n} W_i$ and $Q = \frac{1}{4}\sum_{i=1}^{n} W_i)$. For each case, ten problems were randomly generated. Computational experiments were carried out on an IBM-PC using the C language. As shown in Fig. 3, our proposed algorithm shows the best efficiency when the sum of the traffic is much smaller than Q $(Q = \frac{1}{4}\sum_{i=1}^{n} W_i)$. In Table 2, the mean execution time for $Q = \frac{1}{4}\sum_{i=1}^{n} W_i$ is about 60 seconds when the number of nodes is 30. On the other hand, the mean execution time for $Q = \frac{1}{3}\sum_{i=1}^{n} W_i$ and $Q = \frac{1}{2}\sum_{i=1}^{n} W_i$ are 134 seconds and 200 seconds respectively. The reason for the less execution time in the light traffics is that since L value in the algorithm becomes small in such a case, the amount of computations for our DC-MCLP algorithm also becomes small.

From Table 1 and 2, it is seen that the proposed DC-MCLP algorithm is affected by the traffic volume and Q, and is effective in the case when the number of nodes is less than thirty and the traffic volume is light. Since the execution time increases exponentially when the number of nodes is more than thirty, it is desirable to use the heuristic method in that case.

Table 2. Mean Execution Time (sec)

maximum traffic (Q)	number of nodes (n)		
	10	20	30
$Q = 1/2\sum W_i$	0.33	174	200
$Q = 1/3\sum W_i$	0.32	24	134
$Q = 1/4\sum W_i$	0.27	25	60

5. Conclusions

In this paper, we presented the problem formulation and an exact algorithm for the DC-MCLP, which has not been reported in any previous work. The proposed exact algorithm minimizes the total cost to discover the optimal loop topology with additional mean network delay constraint. It consists of generating the feasible paths using dynamic programming, finding the capacitated minimum loops by using the matching procedure, and allocating the optimal link capacities to meet the network mean delay constraint. The algorithm can be applied to any network regardless of its configuration. Computational complexity analysis and performance evaluation show that the proposed algorithm is effective when the number of nodes is less than thirty and the total traffic volume is much smaller than the maximum traffic to be served by a port of the source node. Our proposed algorithm can be used by network designers for the topological design of FDDI and SONET, or centralized networks with a small number of nodes. In addition, it can be used to discover the broadcast loops for real-time multimedia traffic. Future work consists of developing a heuristic algorithm applicable to local networks with large number of nodes, in addition to an alternative exact algorithm for the small networks with further reduced computation time requirements.

References

[1] K. Altinkemer and B. Gavish, "Heuristics for Delivery Problems with Constant Error Guarantees", *Tran. Sci.*, Vol. 24, No. 4, 1990, pp. 294-297.

[2] I. Astic I and O. Festor, "A hierarchical topology discovery service for IPv6 networks", *IEEE/IFIP Network Operations and Management Symposium*, 2002, pp. 497-510.

[3] Y. Bejerano Y, M. Breitbart M and R. Rastogi, "Physical topology discovery for large multi subnet networks", *INFOCOM,* 2003, pp. 342-352.

[4] Y. Breitbart, M. Garofalakis, C. Martin, S. Seshadri and A. Silberschatz, "Topology discovery in heterogeneous IP networks", *INFOCOM*, 2000, pp. 265-274.

[5] B. Gavish, "Topological Design of Tele-communication Networks-Local Access Design Methods", *Annals of Operations Research*, Vol. 33, 1991, pp. 17-71.

[6] M. Haimovich and Rinnooy Kan, "Bounds and Heuristics for Capacitated Routing Problems", *Mathematics of Operations Research*, Vol. 10, No. 4, 1985, pp. 527-542.

[7] B. Huffaker, D. Plummer, D. Moore and K. Claffy, "Topology discovery by active probing", *Applications and the Internet (SAINT) Workshops*, 2002, pp. 90-96.

[8] Y. Lee, "Minimal Cost Heuristic Algorithm for Delay Constrained Loop Network", *International Journal of Computer Systems Science & Engineering*, Vol. 19, No. 4, CRL Publishing, 2004, pp. 209-219.

[9] J. Lenster and Rinnooy Kan, "Complexity of Vehicle Routing and Scheduling Problems", *Networks*, Vol. 11, 1981, pp. 221-227.

[10] H. Lin H, Y. Wang, C. Wang and C. Chen, "Web-based Distributed topology discovery of IP Networks", *15th International Conference on Information Networking*, 2001, pp. 857-862.

[11] D. Reeves and H. Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing", *IEEE/ACM Transaction on Networking* Vol. 8, No. 2, 2000, pp. 239-250.

[12] W. Zhengying, S. Bingxin and Z. Ling, "A Delay-Constrained Least-Cost Multicast Routing Heuristic for Dynamic Multicast Groups", *Electronic Commerce Research* Vol. 2, 2002, pp. 323-335.