Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 28 (2005) 236–245
computer communications
www.elsevier.com/locate/comcom

Near-optimal tree-based access network design

Hong-Hsu Yen[a,*], Frank Yeong-Sung Lin[b]

[a]Department of Information Management, Shih Hsin University, no. 1, Lane 17, Mu-Cha Road, Sec. 1, Taipei 116, Taiwan, ROC
[b]Department of Information Management, National Taiwan University, no. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan, ROC

Received 2 March 2003; revised 2 August 2004; accepted 2 August 2004
Available online 25 August 2004

Abstract

Among various access network topologies, the tree topology is the most popular due to its simplicity and relatively low cost. A salient example is the CATV network. In this paper, we consider the tree-based access network design problem where the operational cost and the fixed installation cost are jointly minimized. The problem is formulated as a combinatorial optimization problem, where the difficulty of solving a Steiner tree problem typically encountered in a tree-based topological design problem is particularly circumvented. The basic approach to the algorithm development is Lagrangean relaxation and the subgradient method. In the computational experiments, the proposed algorithm calculates near-optimal solutions within 3.2% of an optimal solution in 1 min of CPU time for test networks of up to 26 nodes.

Keywords: Access network design; Network planning; Tree-based networks; Optimization; Lagrangean relaxation; Steiner tree problem

1. Introduction

A network design problem can be decomposed into two main parts, i.e. the access network design problem and the backbone network design problem. Typical network topology could be shown in Fig. 1.

In Fig. 1, we could see that the backbone network which is connected by the backbone router is usually a mesh topology for reliability concern. In designing an access network, a number of topologies may be selected, e.g. tree, mesh and star. However, the following two trends will strongly affect the access network topology eventually chosen. First, low cost, high bandwidth and reliable transmission technologies, such as fiber optics, have become popular. Second, traffic aggregation makes economical sense since multiplexing/demultiplexing is cheap and the cost of leasing or buying bandwidth often reflects economies-of-scale [2]. Both trends have made the tree topology more promising than others.

Research has been conducted to address the access network design problem. Gavish [4] models the network design problem by a rigorous mathematical formulation and solves the problem by Lagrangean relaxation techniques. However, in Ref. [4] each cluster of end users is connected to the network directly via a dedicated line in order to simplify the formulation.

Andrew [2] proposes an integer programming approach to solve the access network design problem, where the objective function is to minimize the access network deployment cost subject to flow constraints. Linear programming relaxation is then applied to solve the integer programming problem. Andrew provides performance guarantees independent of the network and the traffic volume under the weak assumption on the cost structure. Routen [7] proposes genetic and neural network approaches to solve the access network design problem. However, no simulation results are reported in Ref. [7] to evaluate the proposed approaches.

In this paper, the minimum-cost tree-based access network design problem is considered. The problem is formulated as a combinatorial optimization problem where the objective function is to minimize the total access network design cost subject to the multicast tree constraint. Then a sophisticated mathematical formulation is derived to facilitate the application of efficient and effective solution

* Corresponding author. Tel.: +886-2-2236-8225x3357; fax: +886-2-2236-0772.
E-mail address: hhyen@cc.shu.edu.tw (H.-H. Yen).

0140-3664/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.08.002

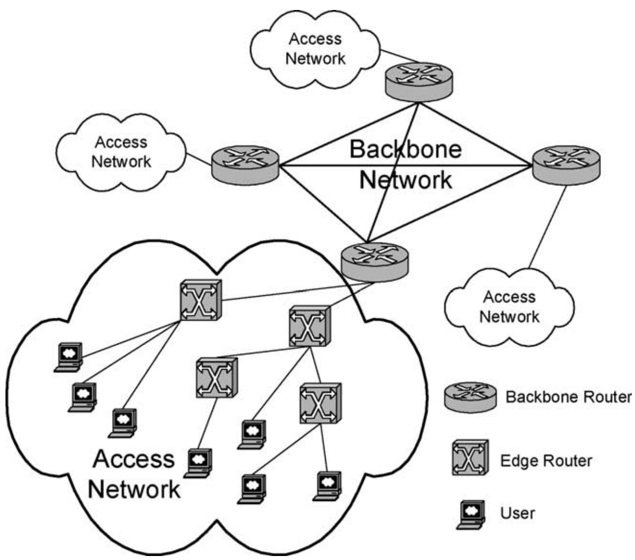

Fig. 1. Typical network topology (backbone+ access network).

approaches and to particularly circumvent the difficulty in solving a Steiner tree problem, which often significantly complicates this type of tree-based network design problems.

The cost considered in the access network design problem consists of the operational cost and the fixed installation cost. The fixed installation cost is the conduit construction cost, which is usually the dominant part of the total cost. We model the operational cost as the routing cost on the links. It is assumed that the upstream traffic is relatively small as compared to the downstream traffic. Therefore, only the downstream routing cost is considered in this paper. In order to provide a more generic problem formulation, multicasting traffic instead of broadcasting and unicasting traffic is considered in this paper. In addition, we investigate two different types of routing cost, namely linear and concave cost function. From the computational experiments, the solution quality for concave routing cost function is even better than linear routing cost function.

The remainder of this paper is organized as follows. In Section 2, a mathematical formulation of the access network design problem is proposed. In Section 3, a reformulation of the access network design problem, which avoids solving the Steiner tree problem, is proposed. In Section 4, a dual approach based on Lagrangean relaxation is presented. In Section 5, heuristics for calculating primal feasible solutions are discussed. In Section 6, computational results are reported. In Section 7, concave routing cost function is studied. Section 8 finally summarizes and concludes this paper.

2. Access network design problem formulation

An access network is modeled as a graph, where the processors are represented by nodes and the communication channels are represented by arcs. The notation adopted in the problem formulation is listed below.

T	The set of all spanning trees
T_g	The set of all trees for multicast group g
G	The set of all multicast groups
L	The set of all links in the graph
N	The set of all nodes in the graph
δ_{tl}	The indicator function which is 1 if link l is on tree t and 0 otherwise
σ_{gtl}	The indicator function which is 1 if link l is on tree t for multicast group g and 0 otherwise
r_g	Traffic requirement of multicast group g
a_l	Unit transmission cost associated with link l
b_l	Fixed installation cost associated with link l
M	An arbitrarily large number

The decision variables for the access network design problem are denoted as follows.

y_{gt}	1 if tree t is adopted by multicast group g, and 0 otherwise
z_t	1 if spanning tree t is selected to be shared by all the multicast groups and 0 otherwise

The access network design problem is then formulated as the following combinatorial optimization problem (IP1)

$$Z_{\mathrm{IP1}} = \min \sum_{l\in L}\sum_{g\in G}\sum_{t\in T_g} r_g a_l \sigma_{gtl} y_{gt} + \sum_{l\in L}\sum_{t\in T} b_l \delta_{tl} z_t \qquad \text{(IP1)}$$

subject to:

$$y_{gt} = 0 \text{ or } 1, \quad \forall g\in G, \quad t\in T_g \qquad (1)$$

$$\sum_{t\in T_g} y_{gt} = 1, \quad \forall g\in G \qquad (2)$$

$$z_t = 0 \text{ or } 1, \quad \forall t\in T \qquad (3)$$

$$\sum_{t\in T} z_t = 1 \qquad (4)$$

$$\sum_{g\in G}\sum_{t\in T_g} \sigma_{gtl} y_{gt} \le M \sum_{t\in T} \delta_{tl} z_t, \quad \forall l\in L. \qquad (5)$$

The first term in the objective function of Eq. (IP1) is the operational cost for the access network, which is equal to the total multicast routing cost. The second term in the objective function is the fixed link installation cost for the access network. Hence, the objective is to jointly minimize the total operational and fixed cost for the access network. Constraints (1) and (2) require that multicast group g adopts exactly one tree to carry its multicast traffic. Constraints (3) and (4) require that exactly one spanning tree be selected and shared by all the multicast groups. Constraint (5) requires that tree t selected by multicast group g be a subset of the shared spanning tree. The above formulation is an integer multicommodity flow problem.

After performing Lagrangean relaxation by relaxing constraint (5) in Eq. (IP1), we obtain the following Lagrangean relaxation problem (LR1). And α_l is the associated Lagrangean multiplier.

$$\min \sum_{l\in L}\sum_{g\in G}\sum_{t\in T_g} r_g\sigma_{gtl}y_{gt}a_l + \sum_{l\in L}\alpha_l\left(\sum_{g\in G}\sum_{t\in T_g}\sigma_{gtl}y_{gt} - M\sum_{t\in T}\delta_{tl}z_t\right) + \sum_{l\in L}\sum_{t\in T} b_l\delta_{tl}z_t \qquad \text{(LR1)}$$

subject to:

$$y_{gt} = 0 \text{ or } 1, \quad \forall g\in G,\ t\in T_g \qquad (6)$$

$$\sum_{t\in T_g} y_{gt} = 1, \quad \forall g\in G \qquad (7)$$

$$z_t = 0 \text{ or } 1, \quad \forall t\in T \qquad (8)$$

$$\sum_{t\in T} z_t = 1. \qquad (9)$$

Eq. (LR1) can be decomposed into the following two independent subproblems.

Subproblem 1: for z_t

$$\min \sum_{l\in L}\sum_{t\in T}(b_l - M\alpha_l)\delta_{tl}z_t \qquad \text{(SUB1)}$$

subject to Eqs. (8) and (9).

Subproblem 2: for y_{gt}

$$\min \sum_{l\in L}\sum_{g\in G}\sum_{t\in T_g}(\alpha_l + r_g a_l)\sigma_{gtl}y_{gt} \qquad \text{(SUB2)}$$

subject to Eqs. (6) and (7).

(SUB1) is a directed minimum cost spanning tree problem with arc weight ($b_l-\alpha_l M$), which can be solved by the minimum weight arborescence algorithm [3,5,8]. The computational complexity of the above algorithm is $O(|L||N|)$ [5]. However, Eq. (SUB2) is a difficult Steiner tree problem, and is known to be NP-hard problem [6]. We need to exhaustive enumerate all possible multicast trees to identify the minimum cost tree for each multicast group. As a result, we reformulate the access network design problem to circumvent the difficulty of solving the Steiner tree problem.

3. Revised access network design problem formulation

By examining Eq. (IP1), we observe that we need to solve the Steiner tree problem due to the need to identify the tree adopted by the multicast group (i.e. y_{gt}). However, if we could identify the links adopted by the multicast group and enforce the following two criteria, then we could identify the tree adopted by the multicast group implicitly by the selected links. First, there must be a path (i.e. x_{gdp} in the formulation below) from the source node of the multicast group to each group destination by using the selected links (i.e. y_{gl} in the formulation below). Second, the selected links must be a subset of the links selected by shared tree z_t. The above observation leads to the following formulation.

The notation adopted to model the revised access network design problem is shown below.

T	The set of all spanning trees
G	The set of all multicast groups
L	The set of all links in the graph
N	The set of all nodes in the graph
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
σ_{tl}	The indicator function which is 1 if link l is on tree t and 0 otherwise
r_g	Traffic requirement of multicast group g
a_l	Transmission cost associated with link l
b_l	Fixed installation cost associated with link l
P_{gd}	The set of paths that destination d of multicast group g may use
h_g	The minimum number of hops to the farthest destination node in multicast group g
D_g	The set of destinations of multicast group g

As compared to the notations given in Section 2, h_g and D_g are added to increase the solution quality of the proposed algorithm. D_g could be calculated in advance. For example, if there are three destinations nodes for multicast group g. Then the $|D_g|$ is equal to 3. On the other hand, h_g could be calculated in advance. Consider an illustrative network topology given in Fig. 2, the multicast group source node is 1 and the destination nodes are 2 and 5. It is obvious that the $h_g=3$ since the farthest destination node is 5 and it is three hops away from the source node. We propose the Dijkstra's shortest path-based algorithm, denoted as **Caculate_h_g**, to calculate h_g for each multicast group g by setting each link arc weight to be one.

Caculate_h_g
begin
initialize all link arc weight to be 1.0;
for $g := 1$ **to** $|G|$ **do**

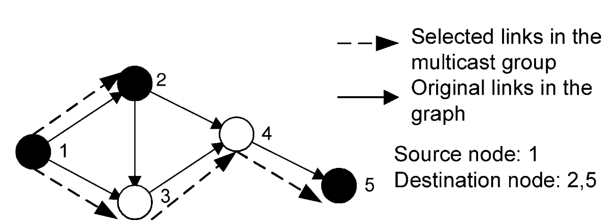

Fig. 2. Simple network topology example.

```
begin
   initialize h_g[g] := 0;
   for d := 1 to |N| do
   begin
      if d ∈ D_g then
      begin
         run Dijkstra's shortest path algorithm to
         determine the hop distance (hop[d]) from
         source node to this destination d;
         if hop[d] > h_g[g] then h_g[g] := hop[d];
      end;
   end;
 end;
end;
```

P_{gd} considers all possible paths that destination d of multicast group g may use, and we do not need to generate these kinds of paths in advance. In the algorithms proposed in Section 4, the arc weight u_{gdl} on each link l enable us to find the shortest path by using the Dijkstra's algorithm to identify the path used by destination d of multicast group g.

The decision variables for the revised access network design problem are denoted as follows.

y_{gl}	1 if link l is on the tree adopted by multicast group g and 0 otherwise
x_{gdp}	1 if path p is selected for group g destined for destination d and 0 otherwise
z_t	1 if spanning tree t is selected to be shared by all the multicast groups and 0 otherwise

The revised formulation for the access network design problem is given below.

$$Z_{\mathrm{IP2}} = \min \sum_{l\in L}\sum_{g\in G} r_g a_l y_{gl} + \sum_{l\in L}\sum_{t\in T} b_l \sigma_{tl} z_t \quad \text{(IP2)}$$

subject to:

$$y_{gl} = 0 \text{ or } 1, \quad \forall g\in G, \quad l\in L \quad (10)$$

$$\sum_{l\in L} y_{gl} \geq \max\{h_g, |D_g|\}, \quad \forall g\in G \quad (11)$$

$$z_t = 0 \text{ or } 1, \quad \forall t\in T \quad (12)$$

$$\sum_{t\in T} z_t = 1 \quad (13)$$

$$\sum_{p\in P_{gd}} x_{gdp}\delta_{pl} \leq y_{gl}, \quad \forall g\in G, \quad d\in D_g, \quad l\in L \quad (14)$$

$$y_{gl} \leq \sum_{t\in T} \sigma_{tl} z_t, \quad \forall g\in G, \quad l\in L \quad (15)$$

$$\sum_{p\in P_{gd}} x_{gdp} = 1, \quad \forall g\in G, \quad d\in D_g \quad (16)$$

$$x_{gdp} = 0 \text{ or } 1, \quad \forall g\in G, \quad d\in D_g, \quad p\in P_{gd}. \quad (17)$$

The objective function of Eq. (IP2) is to minimize the total operational and fixed cost for the access network. Constraints (10) and (11) require that the number of links on the multicast tree adopted by multicast group g be at least the maximum of h_g and the cardinality of D_g. Note that both h_g and the cardinality of D_g are legitimate lower bounds on the number of links on the multicast tree adopted by multicast group g. As an example shown in Fig. 2

$$\sum_{l\in L} y_{\mathrm{gl}} = 4$$

and $h_g=3$, $|D_g|=2$, $\max\{h_g, |D_g|\}=3$. Constraint (11) is a redundant constraint. From the computational experiments, the effectiveness of the proposed algorithm is enhanced when constraint (11) is considered.

Constraints (12) and (13) require that exactly one shared spanning tree be adopted by all multicast groups. Constraint (14) requires that if one path is selected for group g destined for destination d, the path must also be on the tree adopted by multicast group g. Constraint (15) requires that the tree adopted by any multicast group be a subset of the shared spanning tree. Constraints (16) and (17) require that exactly one path be selected for any group g destined for its destination d.

4. Lagrangean relaxation

In Eq. (IP2), constraints (14) and (15) are relaxed, which lead to the following Lagrangean relaxation problem (LR2). u and v Lagrangean multiplier vectors are introduced for performing Lagrangean relaxation, and u_{gdl} and v_{gl} represent corresponding individual Lagrangean multiplier.

$$\begin{aligned} Z_{D2}(u,v) = \min &\sum_{l\in L}\sum_{g\in G} r_g a_l y_{gl} \\ &+ \sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L} u_{gdl}\left(\sum_{p\in P_{gd}} x_{gdp}\delta_{pl} - y_{gl}\right) \\ &+ \sum_{g\in G}\sum_{l\in L} v_{gl}\left(y_{gl} - \sum_{t\in T}\sigma_{tl} z_t\right) \\ &+ \sum_{l\in L}\sum_{t\in T} b_l \sigma_{tl} z_t \end{aligned} \quad \text{(LR2)}$$

subject to:

$$y_{gl} = 0 \text{ or } 1, \quad \forall g\in G, \quad l\in L \quad (18)$$

$$\sum_{l\in L} y_{gl} \geq \max\{h_g, |D_g|\}, \quad \forall g\in G \quad (19)$$

$$z_t = 0 \text{ or } 1, \quad \forall t\in T \quad (20)$$

$$\sum_{t\in T} z_t = 1 \quad (21)$$

$$\sum_{p \in P_{gd}} x_{gdp} = 1, \quad \forall g \in G, \quad d \in D_g \tag{22}$$

$$x_{gdp} = 0 \text{ or } 1, \quad \forall g \in G, \quad d \in D_g, \quad p \in P_{gd}. \tag{23}$$

We can decompose Eq. (LR2) into the following three independent subproblems.

Subproblem 3: for z_t

$$\min \sum_{l \in L} \sum_{t \in T} \left(b_l - \sum_{g \in G} v_{gl} \right) \sigma_{tl} z_t \tag{SUB3}$$

subject to Eqs. (20) and (21).

Subproblem 4: for y_{gl}

$$\min \sum_{l \in L} \sum_{g \in G} r_g a_l y_{gl} - \sum_{g \in G} \sum_{d \in D_g} \sum_{l \in L} u_{gdl} y_{gl} + \sum_{g \in G} \sum_{l \in L} v_{gl} y_{gl} \tag{SUB4}$$

subject to Eqs. (18) and (19).

Subproblem 5: for x_{gdp}

$$\min \sum_{g \in G} \sum_{d \in D_g} \sum_{l \in L} \sum_{p \in P_{gd}} u_{gdl} x_{gdp} \delta_{pl} \tag{SUB5}$$

subject to Eqs. (22) and (23).

Again, Eq. (SUB3) can be solved by the minimum weight arborescence algorithm [3,5,8]. The computational complexity of the above algorithm is $O(|L||N|)$ [5]. Eq. (SUB4) can be further decomposed into $|G|$ independent subproblems. For each multicast group g

$$\min \sum_{l \in L} \left(r_g a_l + v_{gl} - \sum_{d \in D_g} u_{gdl} \right) y_{gl} \tag{SUB4-1}$$

subject to:

$$y_{gl} = 0 \text{ or } 1, \quad \forall l \in L \tag{24}$$

$$\sum_{l \in L} y_{gl} \geq \max\{h_g, |D_g|\}. \tag{25}$$

The proposed algorithm to solve Eq. (SUB4-1) is stated as follows:

Step 1. Compute the number of negative coefficients for all links where the coefficient for each link l is

$$r_g a_l + v_{gl} - \sum_{d \in D_g} u_{gdl}.$$

Step 2. If the number of negative coefficients is greater than $\max\{h_g, |D_g|\}$ for multicast group g, then let each y_{gl} whose corresponding coefficient is negative be 1 and 0 otherwise.

Step 3. If the number of negative coefficients, say c, is no greater than $\max\{h_g, |D_g|\}$ for multicast group g, then first let each y_{gl} whose corresponding coefficient is negative be 1. Second, assign the $(\max\{h_g, |D_g|\} - c)$ number of y_{gl} to be 1 whose corresponding coefficients are the smallest positive values. Third, let the remaining y_{gl} be 0.

The computational complexity of the above algorithm is $O(|L|(|D_g| + \log|L|))$ for each multicast group.

Eq. (SUB5) can be further decomposed into $\sum_{g \in G} |D_g|$ independent shortest path problems with non-negative arc weights. They can be effectively solved by the Dijkstra's algorithm. The computational complexity of the Dijkstra's algorithm is $O(|N|^2)$ for each destination of the multicast group.

It can be seen that the Steiner tree problem no longer exists in this Lagrangean relaxation problem. We can efficiently solve the Lagrangean relaxation problem optimally. By using the weak Lagrangean duality theorem (for any given set of non-negative multipliers, the optimal objective function value of the corresponding Lagrangean relaxation problem is a lower bound on the optimal objective function value of the primal problem [1]), $Z_{D2}(u,v)$ is a lower bound on Z_{IP2}. We construct the following dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method [1]

$$Z_D = \max Z_{D2}(u, v) \tag{D}$$

subject to: $u,v \geq 0$.

Let the vector S be a subgradient of $Z_{D2}(u,v)$ at (u,v). In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (u^k, v^k)$ is updated by $m^{k+1} = m^k + \alpha^k S^k$ where

$$S^k(u, v) = \left(\sum_{p \in P_{gd}} x_{gdp} \delta_{pl} - y_{gl}, y_{gl} - \sum_{t \in T} \sigma_{tl} z_t \right).$$

The step size α^k is determined by

$$\delta \frac{Z_{IP2}^k - Z_{D2}(m^k)}{||S^k||^2},$$

where Z_{IP2}^k is the best primal objective function value found by iteration k(an upper bound on the optimal primal objective function value), and δ is a constant $(0 \leq \delta \leq 2)$.

5. Getting primal feasible solutions

To calculate primal feasible solutions to the access network design problem, solutions to the Lagrangean relaxation problem (LR2) are considered. Three primal heuristics are hence developed. The first and the second primal heuristics are to begin with x_{gdp} and y_{gl}, respectively, by using an add-and-drop heuristic to construct a shared spanning tree for all the multicast groups. These two primal heuristics typically require a significant number of link adding and dropping operations since the links selected by these heuristics may not even be a tree.

The third heuristic is to directly apply z_t after solving Eq. (LR2), which seems to be the simplest among the three.

After the shared spanning tree is selected, x_{gdp} and y_{gl} can be determined. In the computational experiments, this heuristic is shown to be particularly effective and superior to the other two. The computational complexity for the third heuristic is $O\left(|L|\sum_{g\in G}|D_g|\right)$.

In the following, we show the complete algorithm (denoted as LGR) to solve problem (IP2).

Algorithm LGR

begin
read user input file (network topology; link configurations; multicast group configurations, traffic requirements);
calculate $|D_g|$ for each multicast group g;
run **Calculate_h_g**;
initialize the Lagrangean multiplier vector (u and v) to be all zero vector;
UB := very large number; LB := 0;
quiescence_age := 0;
step_size_coefficient := 2;
for *iteration* := 1 **to** *Max_Iteration_Number* **do**
begin
run sub-problem (SUB3);
run sub-problem (SUB4);
run sub-problem (SUB5);
calculate Z_D;
if $Z_D > LB$ **then** $LB := Z_D$ and *quiescence_age* := 0;
else *quiescence_age* := *quiescence_age* + 1;
if *quiescence_age* = *Quiesceince_Threshold* **then** *quiescence_age* := 0 and $\delta := \delta/2$;
run Primal_Heuristic_Algorithm;
if $ub < UB$ **then** $UB := ub$; /**ub* is the newly computed upper bound */
run update-step-size;
run update-multiplier;
end;
end;

And the computational complexity for LGR is $O(|L||N| + |L||G|\log|L| + (|N|^2 + |L|)\sum_{g\in G}|D_g|)$ for each iteration.

6. Computational experiments

The proposed algorithm for the access network design problem developed in Sections 4 and 5 is coded in C and run on a PC with INTEL™ PIII-500 CPU. We tested the algorithm for three networks—ARPA, GTE and OCT with 21, 12 and 26 nodes, respectively. The network topologies can be found in Ref. [9].

Max_Iteration_Number and *Quiesceince_Threshold* are set to 1500 and 30, respectively. The step size coefficient δ is initialized to be 2 and will be halved when the objective function value of the dual problem does not improve for iterations up to *Quiesceince_Threshold*.

Three sets of computational experiments are performed. In the first set of experiments, it is assumed that the traffic demand of each multicast group is one packet per second. In addition, the destinations of distinct multicast groups are mutually exclusive. In the second set of experiments, traffic demand of each multicast group is randomly selected from an interval ranging from 0 to 20. The destinations of distinct multicast groups are generated in the same way as in the first set of computational experiments. In the third set of experiments, traffic is randomly generated for each multicast group, and the destinations of each multicast group are also randomly generated. As a result, the destinations of distinct multicast groups may not be mutually exclusive.

Both the link cost and the unit routing cost are uniformly distributed between 1 and 2000 in these three sets of computational experiments. Ten different numbers of multicast groups, ranging from 11 to 20, are tested in these three sets of experiments.

Table 1 shows the results of the first set of computational experiment. As can be seen from Table 1, the error gaps ([(upper bound − lower bound)/lower bound] × 100%) between the lower bound and the upper bound are within

Table 1
Computational results for experiment 1

Network topology	No. of multicast groups	Lower bound (Lb)	Upper bound (Ub)	(Ub−Lb)/Lb×100%
ARPA	11	96,809	96,809	0.0
	12	95,972	95,972	0.0
	13	98,629	98,629	0.0
	14	102,476	102,476	0.0
	15	106,024	106,024	0.0
	16	103,964	103,964	0.0
	17	110,575	110,575	0.0
	18	107,980	107,980	0.0
	19	113,835	113,836	0.0[a]
	20	112,434	112,434	0.0
GTE	11	40,114	40,114	0.0
	12	43,497	43,497	0.0
	13	43,497	43,497	0.0
	14	43,497	43,497	0.0
	15	43,497	43,497	0.0
	16	43,497	43,497	0.0
	17	43,497	43,497	0.0
	18	43,497	43,497	0.0
	19	43,497	43,497	0.0
	20	43,497	43,497	0.0
OCT	11	144,360	144,362	0.0
	12	137,556	137,610	0.0
	13	135,222	135,370	0.1
	14	129,380	129,380	0.0
	15	133,767	133,767	0.0
	16	142,395	142,395	0.0
	17	151,322	151,322	0.0
	18	161,297	161,304	0.0
	19	164,900	164,911	0.0
	20	173,337	173,343	0.0

[a] The exact error gap is 0.001%. But for simplicity of expression, only one decimal place is reported.

Table 2
Computational results for experiment 2

Network topology	No. of multicast groups	Lower bound (Lb)	Upper bound (Ub)	(Ub−Lb)/Lb×100%
ARPA	11	897,525	897,766	0.0
	12	931,764	932,224	0.0
	13	908,229	908,395	0.0
	14	995,773	995,878	0.0
	15	946,534	946,651	0.0
	16	906,811	907,209	0.0
	17	824,612	833,030	1.0
	18	849,876	859,086	1.1
	19	826,705	835,378	1.1
	20	845,210	855,018	1.2
GTE	11	403,212	403,212	0.0
	12	460,723	460,723	0.0
	13	460,723	460,723	0.0
	14	460,723	460,723	0.0
	15	460,723	460,723	0.0
	16	460,723	460,723	0.0
	17	460,723	460,723	0.0
	18	460,723	460,723	0.0
	19	460,723	460,723	0.0
	20	460,723	460,723	0.0
OCT	11	1,346,297	1,346,319	0.0
	12	1,458,753	1,460,947	0.2
	13	1,460,953	1,464,893	0.3
	14	1,372,628	1,372,628	0.0
	15	1,421,285	1,421,285	0.0
	16	1,438,466	1,438,471	0.0
	17	1,493,311	1,493,312	0.0
	18	1,517,921	1,519,941	0.1
	19	1,496,299	1,496,357	0.0
	20	1,529,243	1,529,274	0.0

0.2% for all test cases. In other words, the proposed algorithm can almost optimally solve all the test problems.

Table 2 summarizes the computational results for the second set of experiments. As can be seen from Table 2, the error gaps are all within 1.2%. And Table 3 summarizes the computational results for the third set of experiments. From Table 3, the error gaps are all within 3.2% for all network topologies and traffic demand configurations. One result that is not shown in the tables is that every such near-optimal solution is calculated in one minute of CPU time.

In order to show the performance comparison between the developed solution approaches based on the formulation in Sections 2 and 3. We also design a computational experiment for comparison. The results are summarized in Table 4. Six randomly generated network topologies ranging from 3 to 8 nodes are tested for comparison. Note that ARPA, GTE, OCT network topologies are not tested for the solution approach based on Section 2 due to unreasonable long execution time. In these network topologies, the developed solution approaches based on Sections 2 and 3 all locate the optimal solutions. However, from the execution time, the developed solution approach based on Section 2 could not be scaled to large network. On the other hand, the solution approaches based on the formulation in Section 3 possess good scalability due to their polynomial time complexity.

7. Discussion on the transmission cost function

In Section 3, we assumed that the transmission cost associated with link (a_l) is constant, i.e. a linear cost function with aggregate flow assumption is made. In this section, we will investigate more general transmission cost function, especially concave cost function, for the access network design problem.

One more decision variable (f_l) is defined.

f_l	Aggregate flow associated with link l
$\psi_l(f_l)$	Transmission cost function on link l with respect to aggregate flow f_l

The mathematical formulation for the access network design problem considering concave transmission cost function is given below.

$$Z_{\mathrm{IP3}} = \min \sum_{l\in L} \psi_l(f_l) + \sum_{l\in L}\sum_{t\in T} b_l \sigma_{tl} z_t \qquad \text{(IP3)}$$

Table 3
Computational Results for Experiment 3

Network topology	No. of multicast groups	Lower bound (Lb)	Upper bound (Ub)	(Ub−Lb)/Lb×100%
ARPA	11	1,687,053	1,705,161	1.1
	12	2,011,822	2,031,961	1.0
	13	2,269,020	2,302,089	1.5
	14	2,051,741	2,072,944	1.0
	15	1,728,275	1,760,091	1.8
	16	2,110,499	2,125,279	0.7
	17	1,480,630	1,490,257	0.7
	18	2,341,788	2,365,094	1.0
	19	2,693,683	2,715,428	0.8
	20	2,386,016	2,420,162	1.4
GTE	11	1,018,769	1,018,821	0.0
	12	582,284	589,782	1.3
	13	888,953	888,953	0.0
	14	1,110,137	1,110,137	0.0
	15	1,171,455	1,176,496	0.4
	16	1,189,363	1,199,090	0.8
	17	1,260,059	1,264,811	0.4
	18	1,469,297	1,473,999	0.3
	19	1,262,947	1,272,030	0.7
	20	1,196,573	1,212,394	1.3
OCT	11	2,225,798	2,229,021	0.1
	12	2,977,833	2,994,648	0.6
	13	2,650,468	2,693,397	1.6
	14	3,201,981	3,218,372	0.5
	15	3,113,354	3,141,470	0.9
	16	3,158,896	3,195,048	1.1
	17	3,133,284	3,214,536	2.6
	18	2,811,159	2,883,546	2.6
	19	3,375,415	3,400,754	0.8
	20	3,060,634	3,157,325	3.2

Table 4
Computational results for Sections 2 and 3

Network topology	$\|N\|=3$, $\|L\|=6$	$\|N\|=4$, $\|L\|=12$	$\|N\|=5$, $\|L\|=20$	$\|N\|=6$, $\|L\|=22$	$\|N\|=7$, $\|L\|=24$	$\|N\|=8$, $\|L\|=26$
Section 2 (s)	0.1	3.1	763	3055	12,220	48,670
Section 3 (s)	0.1	0.2	0.5	0.65	0.8	2.3
Section 2 (Ub)	23,927	68,785	100,758	134,586	89,295	167,844
Section 3 (Ub)	23,927	68,785	100,758	134,586	89,295	167,844

subject to:

$$y_{gl} = 0 \text{ or } 1, \quad \forall g \in G, \quad l \in L \tag{26}$$

$$\sum_{l \in L} y_{gl} \ge \max\{h_g, |D_g|\}, \quad \forall g \in G \tag{27}$$

$$z_t = 0 \text{ or } 1, \quad \forall t \in T \tag{28}$$

$$\sum_{t \in T} z_t = 1 \tag{29}$$

$$\sum_{p \in P_{gd}} x_{gdp}\delta_{pl} \le y_{gl}, \quad \forall g \in G, \quad d \in D_g, \quad l \in L \tag{30}$$

$$y_{gl} \le \sum_{t \in T} \sigma_{tl} z_t, \quad \forall g \in G, \quad l \in L \tag{31}$$

$$\sum_{p \in P_{gd}} x_{gdp} = 1, \quad \forall g \in G, \quad d \in D_g \tag{32}$$

$$x_{gdp} = 0 \text{ or } 1, \quad \forall g \in G, \quad d \in D_g, \quad p \in P_{gd} \tag{33}$$

$$\sum_{g \in G} r_g y_{gl} \le f_l, \quad \forall l \in L \tag{34}$$

$$f_l \ge 0, \quad \forall l \in L. \tag{35}$$

In Eq. (IP3), constraints (30), (31) and (34) are relaxed, which lead to the following Lagrangean relaxation problem (LR3). u, v and w Lagrangean multiplier vectors are introduced while performing Lagrangean relaxation

$$Z_{D3}(u,v,w) = \min \sum_{l \in L} \psi_l(f_l) + \sum_{g \in G}\sum_{d \in D_g}\sum_{l \in L} u_{gdl}\left(\sum_{p \in P_{gd}} x_{gdp}\delta_{pl} - y_{gl}\right) + \sum_{g \in G}\sum_{l \in L} v_{gl}\left(y_{gl} - \sum_{t \in T}\sigma_{tl}z_t\right) + \sum_{l \in L} w_l\left(\sum_{g \in G} r_g y_{gl} - f_l\right) + \sum_{l \in L}\sum_{t \in T} b_l \sigma_{tl} z_t \tag{LR3}$$

subject to:

$$y_{gl} = 0 \text{ or } 1, \quad \forall g \in G, \quad l \in L \tag{36}$$

$$\sum_{l \in L} y_{gl} \ge \max\{h_g, |D_g|\}, \quad \forall g \in G \tag{37}$$

$$z_t = 0 \text{ or } 1, \quad \forall t \in T \tag{38}$$

$$\sum_{t \in T} z_t = 1 \tag{39}$$

$$\sum_{p \in P_{gd}} x_{gdp} = 1, \quad \forall g \in G, \quad d \in D_g \tag{40}$$

$$x_{gdp} = 0 \text{ or } 1, \quad \forall g \in G, \quad d \in D_g, \quad p \in P_{gd} \tag{41}$$

$$0 \le f_l \le \sum_{g \in G} r_g, \quad \forall l \in L. \tag{42}$$

We can decompose Eq. (LR3) into the following four independent subproblems. The first three subproblems are almost identical to the three subproblems in Eqs. (SUB3)–(SUB5), except in Eq. (SUB4), a_l is replaced with w_l. The solution approaches to solve these three subproblems are the same as in Section 4. As for the subproblem for f_l,

Subproblem 6: for f_l

$$\min \sum_{l \in L} \psi_l(f_l) - w_l f_l \tag{SUB6}$$

subject to Eq. (42).

Eq. (SUB6) can be further decomposed into $|L|$ independent subproblems. For each link l

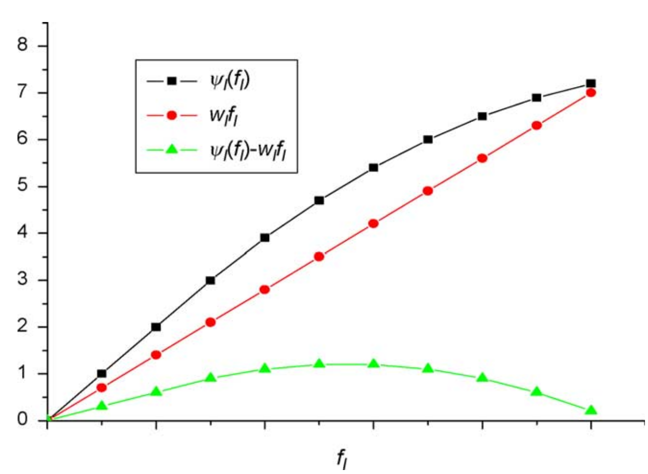

Fig. 3. Concave cost function example.

Table 5
Computational results for concave cost function

Network topology	Threshold	Lower bound (Lb)	Upper bound (Ub)	(Ub–Lb)/ Lb×100%
OCT	0.1	169,073	171,523	1.4
	0.2	169,073	171,715	1.6
	0.3	169,073	171,899	1.7
	0.4	169,073	172,070	1.8
	0.5	169,073	172,221	1.9
	0.6	169,073	172,351	1.9
	0.7	169,073	172,451	2.0
	0.8	169,073	172,513	2.0
	0.9	169,073	172,529	2.0

$$\min \psi_l(f_l) - w_l f_l \tag{SUB6-1}$$

subject to

$$0 \le f_l \le \sum_{g\in G} r_g.$$

Since $\psi_l(f_l)$ is a concave cost function with respect to f_l, then the global minimum of Eq. (SUB6-1) will only occur in two boundary points of $f_l \to 0$ and $\sum_{g\in G} r_g$. In Fig. 3, we illustrate an example to show the objective value of Eq. (SUB6-1) when $\psi_l(f_l)$ is a concave cost function. We could solve Eq. (SUB6-1) by comparing these two boundary points.

The getting primal feasible heuristic algorithm is to start with z_t, which is the same as in Section 5. We conduct computational experiments to show the impact of the concave cost function on the solution quality. $\psi_l(f_l)$ is defined as follow to be a concave cost function with respect to f_l

(1) $\psi_l(f_l) = f_l, \quad \text{if } f_l \le \text{threshold} \times \sum_{g\in G} r_g$

(2) $$\psi_l(f_l) = \psi_l(f_l - 1) + 1 - \frac{f_l - \text{threshold} \times \sum_{g\in G} r_g}{\left(\sum_{g\in G} r_g\right) \times (1 - \text{threshold})},$$

otherwise.

In the above definition, parameter threshold is with the range $0 \le \text{threshold} \le 1$. Before f_l reaches threshold $\times \sum_{g\in G} r_g$, $\psi_l(f_l)$ is a linear function and after that it is a concave function. Traffic is randomly generated for each multicast group, and the destinations of each multicast group are also randomly generated. The number of multicast groups is 20.

In Table 5, we summarize the computational results for OCT network. In the second column of Table 5 is the parameter threshold. As could be seen from Table 5, when the concavity characteristic of $\psi_l(f_l)$ is strong, we could get even better solution quality (e.g. error gap = 1.4% when threshold = 0.1).

8. Summary and conclusion

The access network design problem is crucial when deploying a large-scale network. The cost-effect of the economies-of-scale and the rapid growth of more reliable and high bandwidth transmission technology have favored tree-based topologies for access networks. In this paper, we consider the access network design problem where a tree topology shall be selected. We formulate this problem as a combinatorial optimization problem where the installation and routing cost are jointly minimized. In order to circumvent the difficulty incurred by the inherent Steiner-tree property, a reformulation of the problem is proposed. We take an optimization-based approach by applying the Lagrangean relaxation technique in the algorithm development. From the computational experiments, the proposed algorithm is shown to be efficient and effective. More precisely, the proposed algorithm calculated near-optimal solutions, which are within 3.2% of an optimal solution for linear routing cost function and within 2% of an optimal solution for concave routing cost function in one minute of CPU time for networks of up to 26 nodes.

Acknowledgements

This work is supported by the National Science Council, Taiwen, under grant number NSC91-2213-E-128-0083.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows—Theory, Algorithms, and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[2] M. Andrews, L. Zhang, The access network design problem, Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 40–49.
[3] Y.J. Chu, T.H. Liu, On the shortest arborescence of a directed graph, Scientific Sinica 14 (1965) 1396–1400.
[4] B. Gavish, Topological design of computer communication networks—the overall design problem, European Journal of Operational Research 58 (1992) 149–172.
[5] P.A. Humblet, A distributed algorithm for minimum weight directed spanning trees, IEEE Transactions on Communication COM-31 (6) (1983) 756–762.
[6] R.M. Karp, Reducibility among combinatorial problems in: R.E. Miller, J.W. Thacher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 83–103.
[7] T. Routen, Genetic algorithm and neural network approaches to local access network design, Proceedings of the Second International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems 1994; 239–243.
[8] R.E. Tarjan, Finding optimum branchings, Networks 7 (1977) 25–35.
[9] H.-H. Yen, F.Y.-S. Lin, Near-optimal delay constrained routing in virtual circuit networks, Proceedings of the IEEE INFOCOM, vol. 2, 2001, pp. 750–756.

Hong-Hsu Yen received his BS degree in Industrial Engineering from National Tsing Hua University in 1990, and MS degree in Electrical Engineering from National Taiwan University in 1995 and PhD degree in Information Management from National Taiwan University in 2001. He joined the faculty of the Department of Information Management at Shih Hsin University in 2001. His research interests include network planning, network optimization, performance evaluation and QoS routing.

Frank Yeong-Sung Lin received his BS degree in electrical engineering from the Electrical Engineering Department, National Taiwan University in 1983, and his PhD degree in electrical engineering from the Electrical Engineering Department, University of Southern California in 1991. After graduating from the USC, he joined Telcordia Technologies (formerly Bell Communications Research, abbreviated as Bellcore) in New Jersey, USA, where he was responsible for developing network planning and capacity management algorithms. In 1994, Prof. Lin joined the faculty of the Electronic Engineering Department, National Taiwan University of Science of Technology. Since 1996, he has been with the faculty of the Information Management Department, National Taiwan University. His research interests include network optimization, network planning, performance evaluation, high-speed networks, wireless communications systems and distributed algorithms.