Link Set Sizing for Networks Supporting SMDS

Frank Y. S. Lin

***Abstract*— To size networks that support Switched Multi-megabit Data Service (SMDS), we must determine how much additional capacity is needed and where it is needed so as to minimize the total capacity augmentation cost. We consider two combinatorial optimization problem formulations. These two formulations are compared for their relative applicability and complexity.**

A solution procedure based upon Lagrangean relaxation is proposed for one of the formulations. In computational experiments, the proposed algorithm determines solutions that are within a few percent of an optimal solution in minutes of CPU time for networks with 10–26 nodes. In addition, the proposed algorithm is compared with a Most Congested First (MCF) heuristic. For the test networks, the proposed algorithm achieves up to 152% improvement in the total cost over the MCF heuristic.

I. Problem Description

SWITCHED Multi-megabit Data Service (SMDS) is a high-speed, connectionless, public, packet switching service that will extend Local Area Network (LAN)-like performance beyond the subscriber's premises, across a metropolitan or wide area [1], [2]. To ensure the performance objectives, a backbone network supporting the SMDS service (referred to as an SMDS network) must be carefully managed. The INPLANS™[1] system is developed by Bell Communications Research (Bellcore) to provide a *single* environment to support Bellcore Client Company (BCC) network planning and traffic engineering across different networking technologies instead of building individual systems for each type of networks [3], [4]. The INPLANS integrated network monitoring capability supports studies that monitor the ability of in-place networks to meet performance objectives. When performance exceptions are identified, corrective actions are needed to reduce the degree of overload [3]. One possible action is to adjust the routing assignments. In [5] a responsive routing algorithm is proposed to balance the network load. Usually, routing is a cost-effective solution to network overload caused by short-term traffic fluctuation. However, when the network load exceeds the network capacity and routing adjustment can no longer relieve the network overload, additional capacity is needed. The process of determining the minimum amount of additional capacity needed for an exhausted network and where to add the capacity is referred to as sizing. The sizing approach usually involves ordering/installing equipments and therefore is not intended to be adopted on a (near) real-time basis.

Manuscript received December 23, 1992; revised October 1993; approved by IEEE/ACM Transactions on Networking Editor V. Sahin.

The author is with Bell Communications Research, Piscataway, NJ 08854 (email: lin1@cc.bellcore.com).

IEEE Log Number 9215511.

[1] INPLANS is a trademark of Bellcore.

In this paper, a sizing approach to reducing network overload that persists for a long time is described. The proposed link set sizing algorithm can be used as one of the initial functionalities in the INPLANS integrated network servicing capability to support SMDS networks, which will take corrective actions when performance exceptions are identified by the integrated monitoring capability.

To size the link sets, network planners/administrators must know (i) the end-to-end traffic requirements and (ii) the routing strategy. The end-to-end traffic requirements can possibly be obtained from billing records. Another alternative is to estimate the end-to-end traffic requirements from aggregate link flows by using the Moore–Penrose pseudo-inverse[6]. The pseudo-inverse scheme has been shown to provide an optimal estimate in terms of the variance of estimation errors. However, it is possible that some of the aggregate link flows are underestimated when the above optimal estimate of end-to-end traffic requirements are used and the routing assignments are changed. One conservative estimation scheme is proposed below. The basic idea is to calculate the maximum value of each end-to-end traffic requirement subject to the aggregate link flow information (and perhaps other information to increase the estimation accuracy, e.g., the total external traffic requirement to each switch). Then the aggregate link flows, given any routing assignment, will never be underestimated. For each origin–destination pair, the problem is formulated as a linear programming problem. Unfortunately, no special structure of the linear programming problem has been identified so that more efficient algorithms than the simplex method can be applied. Nevertheless, since the basic idea is to obtain worst case estimates, one may apply the Lagrangean relaxation technique (introduced in Section III) to efficiently calculate tight upper bounds (exact in many cases) on the optimal objective function value.

The routing algorithm for SMDS networks is specified in [7]. A brief review of the default Inter-Switching System Interface (ISSI) routing algorithm is given below. The routing algorithm used for SMDS networks is referred to as ISSI Routing Management Protocol (RMP). The RMP is derived from the Open Shortest Path First (OSPF) specification Version 2 [8]. The main features of the RMP are as follows:

- All routers have identical routing databases where a router is defined to be a Routing Management Entity (RME);
- Each router's database describes the complete topology of the router's domain;
- Each router uses its database and the Shortest Path First (SPF) algorithm to derive the set of shortest paths to all destinations from which it builds its routing table.

Each link set is assigned a positive number in the RMP called the link set metric. The default link set metric of each link set is inversely proportional to the aggregate link set capacity. One can apply standard shortest path algorithms, e.g., Dijkstra's algorithm [9] to calculate a shortest path spanning tree for every origin. Ties are broken by choosing the switch with the lowest router ID number.

Two types of traffic are supported by SMDS—individually addressed message and multicast (group addressed) message. The individually addressed message is transmitted from the origin to the destination over the unique path in the shortest path spanning tree. The multicast message is destined for more than one destination (may not be for all destinations, which is referred to as broadcasting). However, one copy of the multicast message will be transmitted over every link in the shortest path spanning tree. A multicast message will be discarded by a leaf (termination) switch in the shortest path spanning tree if the message is not for any user connected to the switch.

The sizing problem for SMDS networks is difficult when the aforementioned routing algorithm and link set metrics are adopted. From the mathematical formulations shown in the next section, the difficulty is attributed to (i) the nonlinear arc weights with respect to the link set capacities (the default link set metric is inversely proportional to the aggregate link set capacity) and (ii) usually a discrete set of available link set capacities (e.g., in units of DS3 lines). If the probably most commonly used greedy heuristic, i.e., to place additional capacity on the most overloaded link set in each iteration, is applied, we predict and show by an example in Fig. 1 that it is possible that a link set will be even more overloaded after its capacity is augmented (more traffic than the added capacity is rerouted over this link set). Define overflow for a link set to be the aggregate flow deducted by the effective capacity (the link set capacity times the given utilization threshold) of the link set. For illustration and comparison purposes, we develop a Most Congested First (MCF) heuristic:

The MCF Heuristic:

1) Find the link set with the most overflow where ties are broken arbitrarily.
2) Add one link on the link set identified in Step 1.
3) Calculate the new link set metrics and reroute the traffic.
4) If no overflow is found, stop; otherwise go to Step 1.

Below, we go through the example where the MCF heuristic is applied.

In Fig. 1(a), a network with three nodes (switches) and three link sets is shown. Assume that the default routing scheme (the OSPF protocol) and link set metrics (inversely proportional to the link set capacities) are applied and that the utilization threshold for each link set is 0.85 (the engineering thresholds can be calculated using the methods proposed in [10]). In Fig. 1(a), link set (A,B) is overloaded (with 0.88 utilization). Assume that the MCF heuristic is used to solve the sizing problem where one DS3 line (with 34 Mbps effective capacity for the SMDS service) is added to the most overloaded link set (in terms of the amount of overflow) in each iteration. The routing assignments are adjusted accordingly and the aggregate link set flows are recalculated. If any of the link sets is still overloaded, then this process is repeated until the utilization of each link set is no greater than 0.85. Fig. 1(b) shows the network status after 1 DS3 line is added to link set (A,B). This additional link changes the link set metric of (A,B) and therefore the routing assignments. After the traffic is rerouted according to the new set of link set metrics, it is observed that the utilization of link set (A,B) becomes 1.03 which is even larger than the value before a DS3 line is added, i.e., 0.88.

This seemingly counterintuitive phenomenon is attributed to the static nature of the OSPF routing with the default link set metrics, which does not react to the network load (and does not consider the capacity constraints). Also shown in Fig. 1(b) is that (C,A) becomes overloaded while (C,B) is completely unused. Fig. 1(c) to 1(e) depict the intermediate steps and the final result when the MCF heuristic is further applied. Consequently, four DS3 lines are added.

In contrast, Fig. 1(f) shows the optimal solution where the number of additional lines required to satisfy the capacity/utilization constraints is minimized. (For this example problem, the proposed algorithm to be introduced in a later section finds the optimal solution.) The optimality can be verified easily. In the optimal solution, two additional DS3 lines are needed. Compare the optimal solution with the solution obtained by the MCF heuristic, it is observed that the MCF heuristic is not effective in this case (the proposed algorithm achieves a 100% improvement in the total capacity augmentation cost over the MCF heuristic). Another observation from the optimal solution is that network planners/administrators may need to put additional capacity on a link set with normal load originally (e.g., link set (C,B) in this example). It is thus clearly demonstrated that once an overloaded area of the network is identified, one needs to study the whole network rather than an isolated area to determine how much additional capacity is needed and where it is needed.

In this paper, we present two integer programming formulations for the SMDS link set sizing problem. In the first formulation, the objective is to minimize the total routing cost (to enforce the OSPF routing with the default link set metrics) subject to a budget constraint. In the second formulation, we minimize the total capacity augmentation cost subject to a set of shortest-path-routing constraints. Below is a simple complexity analysis of the problem based upon the integer programming model. If the network has k link sets and the minimum number of additional links needed to resolve the overload problem is b (assuming equal cost for each additional link), it is shown below that the number of solution points needed to be evaluated is at least $\sum_{i=1}^{b-1}(k+i-1)!/[k!(i-1)!]$ for a breadth first search scheme. Since the minimum number of additional links needed is b, the problem is infeasible for a given number of additional links j from 1 to $b-1$. Let A_l be the number of links added to link set l. It is well known that the number of integer points satisfying

$$\sum_{l=1}^{k} A_l = j, \forall A_l \text{ being nonnegative integer}$$

is equal to $(k+j-1)!/[k!(j-1)!]$. This completes the proof. Even for a small network, this number can be so large that the exhaustive search scheme attempting to solve the problem optimally becomes impractical. Instead, in this paper we develop an efficient near-optimal solution procedure for the first formulation.

Three networks (eight test cases) with up to 26 nodes were tested in the computational experiments. The proposed algorithm determines solutions that are within a few percent of an optimal solution within minutes of CPU time. Compared with the MCF heuristic, the proposed algorithm achieved 7.1–152% improvement in the total capacity augmentation cost.

This work has the following significance. First, the problem is formulated as mathematical programs, which facilitates optimization-based solution approaches. Second, the proposed near-optimal sizing algorithm can help the BCC's expand SMDS network capacities in an economical way. Third, the formulations and algorithm developed can easily be generalized to consider the joint link set and node sizing problem for SMDS networks (by a different interpretation to the graph model). Last, by letting the existing node and link set capacities for potential locations be zero, this work can be used to solve the topological design and capacity assignment problem for SMDS networks.

The remainder of this paper is organized as follows. In Section II, two formulations of the SMDS link set sizing problem are given and compared. In Section III, a solution procedure based upon Lagrangean relaxation is proposed for the first formulation. In Section IV, generalization and additional constraints on the problem formulations are considered. In Section V, computational results are reported. Section VI summarizes this paper.

II. Problem Formulations

An SMDS network is modeled as a graph $G(V,L)$ where the switches are represented by nodes and the link sets are represented by links. Let $V=\{1,2,\cdots,N\}$ be the set of nodes and L be the set of links in the graph (network). As will be shown in Section 4.1, if the node (switch) sizing problem is considered jointly with the link set sizing problem, the links in the graph then represent either the switches or the link sets, while the nodes represent junctions between link sets and switches. Let W be the set of all origin–destination (O–D) pairs (single destination) in the network. According to the ISSI routing scheme, all traffic of an O–D pair is transmitted over exactly one (shortest) path. Furthermore, multicast traffic from an origin is transmitted over a shortest path spanning tree, which is the union of the shortest paths from the origin to each destination. As explained earlier, the multicast traffic from one origin to each of its associated multicast groups is *broadcast* to all the other Switching Systems (SS's) over the same shortest path spanning tree. For each O–D pair $(o,d)\in W$, the mean arrival rate of new individually addressed traffic is γ_{od} (packets/s), while the aggregate (sum over all associated multicast groups) mean arrival rate of multicast traffic originated at origin o is α_o (packets/sec). Let P_{od} be the set of all possible simple directed paths from the origin to the destination for an O–D pair (o,d). The overall traffic for O–D pair (o,d) is transmitted over one path in the set P_{od}. Let P be the set of all simple directed paths in the network. Let T_o be the set of all spanning trees rooted at o. The multicast traffic originated at o is transmitted over one spanning tree in the set T_o. Let T be the set of all spanning trees in the network. For each link $l\in L$, the existing capacity is C_l packets/s and the added capacity is A_l packets/s (a decision variable).

For each O–D pair $(o,d)\in W$, let

$$x_p=\begin{cases}1 & \text{if path } p\in P_{od} \text{ is used to transmit the individually addressed packets for O-D pair } (o,d)\\ 0 & \text{otherwise.}\end{cases}$$

In an SMDS network, all of the packets of an O–D pair are transmitted over one path from the origin to the destination. Thus $\sum_{p\in P_{od}}x_p=1$. For each path $p\in P$ and link $l\in L$, let

$$\delta_{pl}=\begin{cases}1 & \text{if link } l \text{ is on path } p\\ 0 & \text{otherwise.}\end{cases}$$

For each origin o, let

$$y_t=\begin{cases}1 & \text{if spanning tree } t\in T_o \text{ is used to transmit the multicast message for origin } o\\ 0 & \text{otherwise.}\end{cases}$$

SMDS switches have the capability of duplicating packets for multiple downstream branches of a spanning tree used to carry the multicast traffic. When a packet is multicast from the root to the destinations using tree t, exactly one copy of the packet is transmitted over each link in the tree. Similar to the single-destination case, $\sum_{t\in T_o}y_t=1$ for every origin o. For each tree $t\in T$ and link $l\in L$, let

$$\sigma_{tl}=\begin{cases}1 & \text{if link } l \text{ is on tree } t\\ 0 & \text{otherwise.}\end{cases}$$

Let $\Phi_l(A_l)$ be the cost to add capacity A_l to link l. This cost can include a fixed charge to change the capacity. Usually A_l is chosen from a discrete set K_l, e.g., in units of DS3 lines. A_l can be negative when existing capacities are allowed to be removed form the network. Let $\overline{\rho}_l$ be a prespecified threshold on the utilization factor of link l. The end-to-end delay objectives for SMDS networks will be satisfied if those utilization thresholds are not exceeded. These thresholds can be calculated using the schemes proposed in a recent work on allocating end-to-end delay objectives to individual network elements [10]. The SMDS link set sizing problem can be formulated as the following two combinatorial optimization problems.

2.1. Formulation 1

Let B be the total budget available for capacity augmentation.

$$Z_{\overline{IP1}}=\min\sum_{l\in L}\sum_{(o,d)\in W}\sum_{p\in P_{od}}\frac{x_p\delta_{pl}\gamma_{od}}{C_l+A_l}\qquad(\overline{IP1})$$

subject to Constraints (1)–(8) shown at the bottom of the next page.

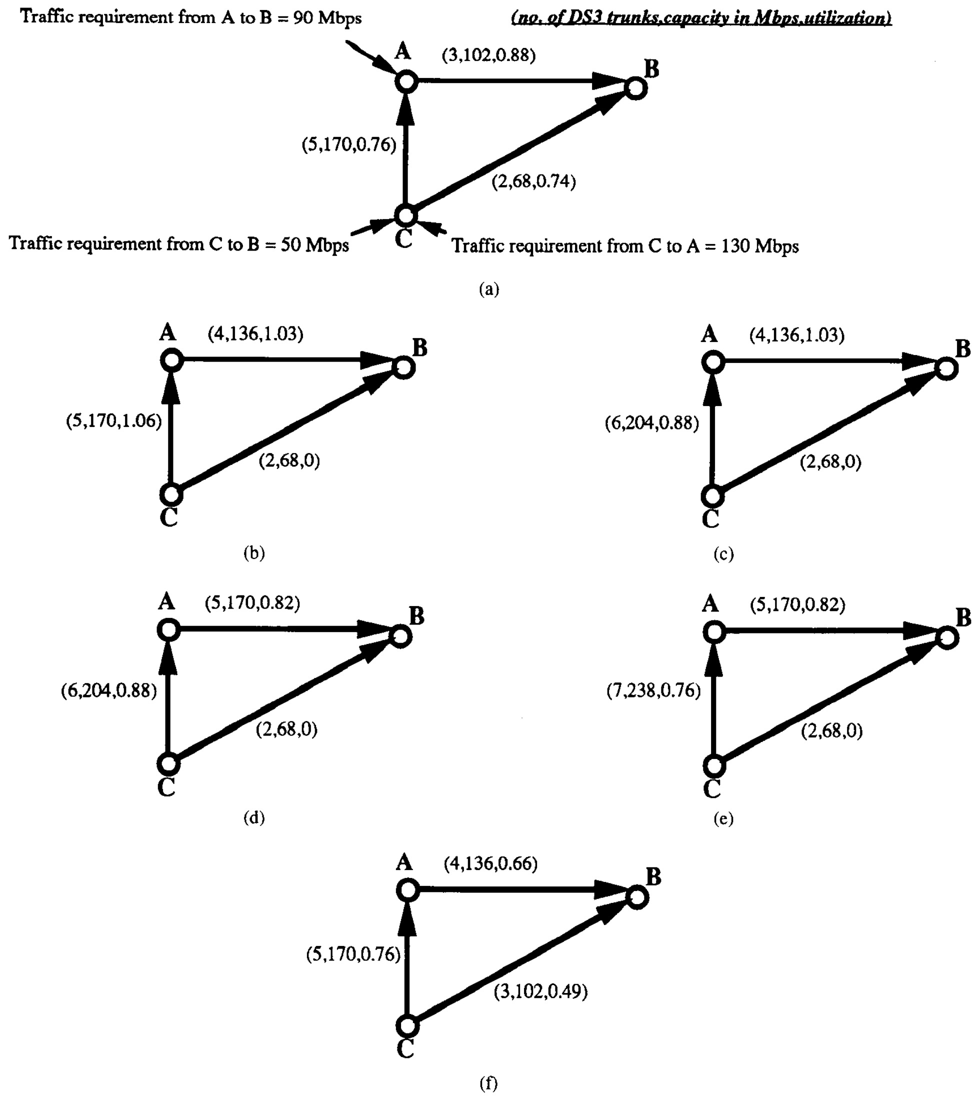

Fig. 1.

The objective function and Constraints (1) and (2) ensure that the individually addressed traffic for every O–D pair be routed over exactly one shortest path where each arc weight is inversely proportional to the corresponding link set capacity. The left hand side of Constraint (3) denotes the aggregate flow (including individually addressed and multicast traffic) over link l. Constraint (3) requires that the utilization factor of each link not exceed a prespecified value (to guarantee the end-to-end delay objectives). Constraints (5) and (6) require that all of the multicast traffic from one origin be transmitted over

$$\begin{array}{rlr}
\sum_{p\in P_{od}} x_p & = 1 & \forall (o,d)\in W \quad (1)\\
x_p & = 0 or 1 & \forall p\in P_{od}, (o,d)\in W \quad (2)\\
\sum_{(o,d)\in W}\sum_{p\in P_{od}} x_p\delta_{pl}\gamma_{od} + \sum_{o\in V}\sum_{t\in T_o} y_t\alpha_o\sigma_{tl} & \le (C_l + A_l)\overline{\rho}_l & \forall l\in L \quad (3)\\
\sum_{d\in V-\{o\}}\sum_{p\in P_{od}} x_p\delta_{pl} & \le (N-1)\sum_{t\in T_o} y_t\sigma_{tl} & \forall l\in L, o\in V \quad (4)\\
\sum_{t\in T_o} y_t & = 1 & \forall o\in V \quad (5)\\
y_t & = 0 or 1 & \forall t\in T_o, o\in V \quad (6)\\
A_l & \in K_l & \forall l\in L \quad (7)\\
\sum_{l\in L}\Phi_l(A_l) & \le B. & (8)
\end{array}$$

exactly one spanning tree. The left hand side of Constraint (4) (together with (1) and (2)) is the number of selected paths (for individually addressed traffic) rooted at origin o and passing through link l, while the right hand side of Constraint (4) (together with (5) and (6)) equals $N-1$ if link l is used in the spanning tree for root o to multicast messages and 0 otherwise. Recall that $N-1$ is the maximum number of selected paths originated at node o and passing through link l. Therefore, Constraint (4) requires that the union of selected paths from one origin to all the destinations for individually addressed traffic be the same spanning tree rooted at the origin to carry multicast traffic. (Note that this constraint implies that the selected paths from one origin to carry individually addressed traffic form a spanning tree.) Constraint (7) requires that the capacity added to each link be allowable. Constraint (8) requires that the total capacity expansion cost not exceed the given budget B.

It would be interesting to investigate the difficulty/complexity of the above problem. If A_l is a constant and only Constraints (1) and (2) are considered, the problem is a well known shortest path problem. However, with the consideration of the capacity constraint (3), the problem becomes NP-complete and no existing polynomial time algorithm is available to solve the problem optimally. Next, the arc weight $(C_l + A_l)^{-1}$ is a nonlinear function of the discrete decision variable A_l. Moreover, The knapsack type of constraint (8), the integrality constraint (6) and the routing constraint (4) for y_t add another degree of difficulty to the problem.

An equivalent formulation of $\overline{\text{IP1}}$ is

$$Z_{IP1} = \min \sum_{l \in L} \frac{f_l}{C_l + A_l} \quad (IP1)$$

subject to:

$$\begin{array}{rll} & (1)-(8) & \\ \sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p \delta_{pl} \gamma_{od} & \leq f_l & \forall l \in L \quad (9) \\ 0 \leq f_l & \leq (C_l + A_l)\overline{\rho}_l & \forall l \in L. \quad (10) \end{array}$$

For each link l, an auxiliary variable f_l is introduced. We interpret those variables to be aggregate flows attributed to individually addressed traffic. Since the objective function is strictly increasing with f_l and (IP1) is a minimization problem, equality of (9) will hold in an optimal solution. As the reader will see in the next section, the introduction of f_l decouples the problem into three independent subproblems in the Lagrangean Relaxation. Constraint (10) gives the range of f_l.

2.2. Formulation 2

$$Z_{IP2} = \min \sum_{l \in L} \Phi_l(A_l) \quad (IP2)$$

subject to Constraints (11)–(18) below.

The objective function is to minimize the total cost of capacity augmentation. Constraints (11)–(17) are the same as (1)–(7). The left hand side of (18) (together with (11) and (12)) is the routing cost for O-D pair (o, d) (for one unit of flow on the selected path). The right hand side of (18) is the cost of path $p \in P_{od}$. Constraint (18) requires that for each O–D pair a shortest path be used to carry the individually addressed traffic.

2.3. A Comparison between Formulations 1 and 2

It would be useful to make a comparison between Formulations 1 and 2 for their relative applicability and complexity. An apparent difference between (IP1) and (IP2) is the objective functions and the last constraints. However, there is a dual relation between these two formulations. The objective function of Formulation 1 (together with constraints (1) and (2)) enforces the shortest path routing strategy, while the last constraint of Formulation 2 explicitly serves this purpose. The objective function of Formulation 2 is to minimize the total capacity augmentation cost, while the last constraint of Formulation 1 imposes an upper limit on the total capacity augmentation cost.

One potential drawback of Formulation 1 is that the shortest path routing strategy is enforced by the objective function but not constraints. The constraint set of (IP1) allows an O–D pair to choose an alternative route when the true shortest path with respect to the default link set metrics is overloaded (under the capacity Constraint (3)). It is therefore possible that (IP1) is feasible with respect to the constraint set but is infeasible with respect to the shortest path routing strategy. One can increase the given budget when no desired solution is found. However, it is undesirable to assign too much budget, which will make link sets overengineered. Consequently, it may take several iterations to adjust the given budget when one wants to determine the minimum budget required. Whereas,

$$\begin{array}{rlrr} \sum_{p\in P_{od}} x_p & = 1 & \forall (o,d) \in W & (11) \\ x_p & = 0 \text{ or } 1 & \forall p \in P_{od}, (o,d) \in W & (12) \\ \sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p \delta_{pl} \gamma_{od} + \sum_{o\in V} \sum_{t\in T_o} y_t \alpha_o \sigma_{tl} & \leq (C_l + A_l)\overline{\rho}_l & \forall l \in L & (13) \\ \sum_{d\in V-\{o\}} \sum_{p\in P_{od}} x_p \delta_{pl} & \leq (N-1) \sum_{t\in T_o} y_t \sigma_{tl} & \forall l \in L, o \in V & (14) \\ \sum_{t\in T_o} y_t & = 1 & \forall o \in V & (15) \\ y_t & = 0 \text{ or } 1 & \forall t \in T_o, o \in V & (16) \\ A_l & \in K_l & \forall l \in L & (17) \end{array}$$

$$\sum_{l\in L} \sum_{q\in P_{od}} \frac{x_q \delta_{ql}}{C_l + A_l} \leq \sum_{l\in L} \frac{\delta_{pl}}{C_l + A_l} \quad \forall p \in P_{od}, (o,d) \in W. \quad (18)$$

the optimal objective function value of (IP2) is the minimum budget needed.

In view of the number of constraints, Formulation 1 is better than Formulation 2 since (18) is potentially comprised of a huge number of constraints (equals the number of simple paths in the network). It is difficult, if not intractable, to consider the numerous constraints in a solution procedure. Although (2) and (6) ((12) and (16) as well) have the same nature, in the proposed solution procedure, these two sets of constraints are considered implicitly in a shortest path problem and a minimal cost spanning tree problem, respectively. As a result, no aforementioned complexity problem will be incurred.

III. A Solution Procedure

Due to the difficulty of developing a solution procedure to Formulation 2 as mentioned in the previous section, we attempt to solve only Formulation 1 in this paper. A solution procedure to Formulation 2 will be developed and presented in a forthcoming paper so that the relative computational complexity and solution quality trade-off of Formulations 1 and 2 can be compared.

The basic approach to the development of a solution procedure to Formulation 1 is Lagrangean relaxation. Lagrangean relaxation is a method for obtaining lower bounds (for minimization problems) as well as good primal solutions in integer programming problems. A Lagrangean relaxation (LR) is obtained by identifying in the primal problem a set of complicating constraints whose removal will simplify the solution of the primal problem. Each of the complicating constraints is multiplied by a multiplier and added to the objective function. This mechanism is referred to as dualizing the complicating constraints.

For Formulation 1 (Problem (IP1)), we dualize constraints (3), (4), (8) and (9) to obtain the following relaxation

$$\begin{aligned}Z_{D1}(v,s,\beta,u) = \min \sum_{l\in L} \frac{f_l}{C_l+A_l} &+ \sum_{l\in L} v_l \Bigg[\sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p \delta_{pl} \gamma_{od} \\ &+ \sum_{o\in V} \sum_{t\in T_o} y_t \alpha_o \sigma_{tl} - (C_l + A_l)\overline{\rho}_l \Bigg] \\ &+ \sum_{l\in L} \sum_{o\in V} s_{ol} \Bigg[\sum_{d\in V-\{o\}} \sum_{p\in P_{od}} x_p \delta_{pl} \\ &- (N-1) \sum_{t\in T_o} y_t \sigma_{tl} \Bigg] \\ &+ \beta \Bigg[\sum_{l\in L} \Phi_l(A_l) - B \Bigg] \\ &+ \sum_{l\in L} u_l \Bigg[\sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p \delta_{pl} \gamma_{od} - f_l \Bigg]\end{aligned} \quad \text{(LR1)}$$

subject to:

$$\begin{aligned}
\textstyle\sum_{p\in P_{od}} x_p &= 1 & \forall (o,d)\in W \quad (19)\\
x_p &= 0 or 1 & \forall p \in P_{od}, (o,d)\in W \quad (20)\\
\textstyle\sum_{t\in T_o} y_t &= 1 & \forall o \in V \quad (21)\\
y_t &= 0 or 1 & \forall t \in T_o, o\in V \quad (22)\\
A_l &\in K_l & \forall l \in L \quad (23)\\
0 \le f_l &\le (C_l + A_l)\overline{\rho}_l & \forall l \in L \quad (24)
\end{aligned}$$

where v, s, and u are the vectors of $\{v_l\}$, $\{s_{ol}\}$ and $\{u_l\}$, respectively. Note that the constraints are dualized in such a way that the corresponding multiplers are nonnegative.

Problem (LR1) can be decomposed into three independent subproblems. Note that the constant terms, e.g., βB, were omitted in the objective function in the subproblems.

Subproblem 1:

$$Z^1_{D1}(v,u,\beta) = \min \sum_{l\in L} \left\{ \frac{f_l}{C_l + A_l} - u_l f_l - v_l \overline{\rho}_l A_l + \beta\Phi(A_l) \right\}$$

subject to:

$$A_l \in K_l \qquad \forall l \in L \tag{25}$$

$$0 \le f_l \le (C_l + A_l)\overline{\rho}_l \qquad \forall l \in L \tag{26}$$

Subproblem 2:

$$Z^2_{D1}(v,s,u) = \min \sum_{l\in L} \sum_{(o,d)\in W} \sum_{p\in P_{od}} [(u_l + v_l)\gamma_{od} + s_{ol}] x_p \delta_{pl}$$

subject to:

$$\sum_{p\in P_{od}} x_p = 1 \qquad \forall (o,d) \in W \tag{27}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{od}, (o,d) \in W \tag{28}$$

and

Subproblem 3:

$$Z^3_{D1}(v,s) = \min \sum_{l\in L} \sum_{o\in V} \sum_{t\in T_o} (v_l \alpha_o - (N-1)s_{ol}) y_t \sigma_{tl}$$

subject to:

$$\sum_{t\in T_o} y_t = 1 \qquad \forall o \in V \tag{29}$$

$$y_t = 0 \text{ or } 1 \qquad \forall t \in T_o, o \in V. \tag{30}$$

Subproblem 1 is composed of $|L|$ (one for each link) problems. Since A_l is discrete and bounded, the problem can be solved by successively fixing A_l to all possible values that satisfy (25). The following observation may greatly improve the efficiency of the solution procedure. For a fixed A_l, the objective function becomes minimizing a linear function of f_l over a simple interval of f_l specified in (26). The minimum objective function value can be found at one boundary point of the simple interval. As a result, to solve the problem for each link, only $2|K_l|$ points need to be evaluated. If $\Phi(A_l)$ possesses a certain property, e.g. convexity or concavity, the computational load can be further reduced. For example, if

$\Phi(A_l)$ is linear, only the 4 extreme points of the convex hull of (25) and (26) need to be evaluated.

Subproblem 2 consists of $|W|$ (one for each O-D pair) shortest path problems where $(u_l + v_l)\gamma_{od} + s_{ol}$ is the arc weight for link l and O–D pair (o, d). Dijkstra's algorithm can be applied to solve the shortest path problems. It is worth mentioning that Constraint (4) can also be written as

$$\sum_{t \in T_o} y_t \sigma_{tl} \leq \sum_{d \in V-\{o\}} \sum_{p \in P_{od}} x_p \delta_{pl} \qquad \forall l \in L, o \in V. \quad (31)$$

However, this may cause the difficulty of negative cycles in the shortest path problems.

Subproblem 3 consists of $|V|$ (one for each root) minimum cost spanning tree problems where $v_l \alpha_o - (N-1)s_{ol}$ is the arc weight for link l and root o. One can apply Prim's or Kruskal's algorithm [11] to solve the problem.

For any $(v, s, \beta, u) \geq 0$, by the weak Lagrangean duality theorem, the optimal objective function value of (LR1), $Z_{D1}(v, s, \beta, u)$, is a lower bound on Z_{IP1} [12]. The dual problem (D1) is

$$Z_{D1} = \max_{(v,s,\beta,u) \geq 0} Z_{D1}(v, s, \beta, u). \quad \text{(D1)}$$

To find the greatest lower bound, we solve (D1). Another common approach to finding a lower bound on the optimal objective function value of a minimization integer programming problem is to use linear programming relaxation (the integrality constraints are relaxed). However, the objective function of (IP1) is in general nonconvex with respect to f_l and A_l (examining the Hessian of $f_l/(C_l + A_l)$). No standard procedure can be applied to solve the linear programming relaxation optimally to obtain a legitimate lower bound.

There are several methods for solving the dual problem (D1). One of the most popular methods is the subgradient method. Let a $(2 + |V|)|L| + 1$ vector b be a subgradient of $Z_{D1}(v, s, \beta, u)$. In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (v^k, s^k, \beta^k, u^k)$ is updated by

$$m^{k+1} = m^k + t^k b^k.$$

The step size t^k is determined by

$$t^k = \delta \frac{Z^h_{IP1} - Z_{D1}(m^k)}{||b^k||^2} \quad (32)$$

where Z^h_{IP1} is an objective function value for a heuristic solution (upper bound on Z_{IP1}) and δ is a constant, $0 < \delta \leq 2$. To solve (D1), the subgradient method is used.

The above procedure is for solving the dual problem and obtaining good lower bounds on the optimal primal objective function value. We next describe a procedure for finding good primal feasible solutions. In each iteration of solving the dual problem (where an (LR1) is solved), one can calculate the aggregate link set flows using the routing assignments form the solution to the (LR1). From these aggregate link set flows, the minimum link set capacities required to satisfy the capacity/utilization constraints can be calculated. We then use these minimum link set capacities to calculate a new set of link set metrics and to reroute the traffic accordingly. If any of the capacity/utilization constraints is violated, we may apply the MCF heuristic to place additional capacity. If the total cost is less than the given budget, then a primal feasible solution is found. Another alternative is to apply the following All Congested First (ACF) heuristic to find primal feasible solutions.

The ACF Heuristic:

1) Set the counter limit K to be a prespecified value.
2) If $K = 0$, return; otherwise, decrease K by 1.
3) Find the link sets where the capacity (utilization) constraints are violated.
4) Add the minimum number of links on each link set identified in Step 3 to satisfy the current flows.
5) Calculate the new link set metrics and reroute the traffic.
6) If no overflow is found, return; otherwise, keep the routing assignments, remove the links added in Step 4 and go to Step 2.

To find the tightest budget constraint (the lowest cost), one may apply the concept of bisecting search in adjusting the given budget B. However, this may require solving a significant number of (IP1)'s. In addition, for a given budget B, it may be difficult to determine whether the problem is feasible (an integer programming problem). The following implementation attempts to achieve better efficiency.

The Overall Algorithm:

1) Apply the MCF heuristic to calculate an initial value of the given budget.
2) Solve the current (IP1).
3) Record the lowest capacity augmentation cost for the feasible capacity augmentation plans in solving (IP1).
4) If the lowest cost from Step 3 is smaller than the current given budget, construct a new (IP1) by replacing the given budget with the lower value and go to Step 2; otherwise, stop.

IV. Generalization and Other Constraints

In this section, a generalization of Formulations 1 and 2 to jointly consider node (switch) and link (link set) sizing for SMDS networks is considered. A new graph model is presented and a number of associated modifications on Formulations 1 and 2 are described. In addition, two types of additional constraints, i.e., switch termination constraints and symmetric link set capacity constraints, are considered. The switch termination constraints require that the number of Subscriber Network Interface (SNI) and ISSI terminations to a switch not exceed a given number, depending upon the capacity requirement of each SNI and ISSI connection. The symmetric link set capacity constraints require that for two adjacent switches the link set capacities be the same in both directions.

4.1. Node Sizing

To jointly consider node and link set sizing, a new graph model is first introduced. An SMDS backbone network is modeled as a graph where each link in the graph corresponds

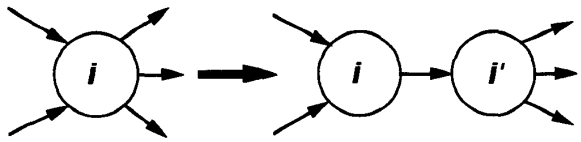

Fig. 2.

to a delay element (e.g., a link set or a switch). Let L_T be the set of links associated with link sets. Let L_S be the set of links associated with switches. Let L be the set of links in the graph $(L_T \cup L_S)$. The nodes in the graph represent junctions among the delay elements. Fig. 2 depicts a node splitting technique to model a switch, say switch i, as a link. The procedure is described as follows.

For each node i in the old graph model, do the following:

1) Split node i in the old graph model into two nodes, i and i', in the new graph model.
2) Connect all the inbound links to switch i in the old graph model to node i in the new graph model.
3) Connect all the outbound links from switch i in the old graph model to node i' in the new graph model.
4) Add a new link (i,i'), which now represents switch i.

After this node splitting procedure is applied, it is clear that all the incoming traffic to switch i must flow through link (i,i') and thus the aggregate nodal flow can be considered. In the event that the architecture of a switch involves more than one major delay elements, the node splitting procedure can be further applied in a similar fashion to capture the aggregate flows on those delay elements.

Since in the default OSPF routing algorithm only the link set metrics (but not the switch metrics) are considered, the arc weight associated with a link in L_S is set to 0. The current formulations can be modified as follows. For Formulation 1, the set L in the objective function is replaced by L_T. For Formulation 2, the set L in the last constraint (18) is replaced by L_T. Another modification that needs to be made is (i) to separate the set of nodes V in the new graph into 2 sets, V_I and V_O, where V_I is the set of originating nodes for links in L_S and V_O is the set of terminating nodes for links in L_S and (ii) to replace the node set V in both formulations by V_I. It is clear that in the generalized formulations, an origin o must be in V_I and a destination d must be in V_O. One last modification is to replace L in Constraints (9) and (10) by L_T since there is no need to introduce an auxiliary variable f for each link associated with a switch.

4.2. Switch Termination Constraints

A switch may have a given number of terminations (ports) where each termination can handle a fixed number of SNI and/or ISSI connections. Given this architecture, one may need to explicitly consider the switch termination constraints, which can possibly become the limiting factors than the switch capacity (utilization) constraints. Let R_i be the number of free terminations for switch i (represented by link i in the new graph model), taking into account the existing SNI and ISSI connections. For switch (link) i, let I_i and O_i be the sets of inbound and outbound links, respectively. Also let $\Omega_i(A_l)$ be the number of ports needed for switch i to connect an incident link set l of capacity A_l. Then the following switch termination constraint needs to be considered in the formulations.

$$\sum_{l \in I_i \cup O_i} \Omega_i(A_l) \le R_i \qquad \forall i \in L_S.$$

This constraint requires that for each switch the total number of terminations needed for installing additional links be no greater than the number of free terminations.

This constraint couples elements in $\{A_l\}$ and is dualized in the Lagrangean relaxation. Consequently, two more terms need to be considered in Subproblem 1 for each link. However, the solution procedure to Subproblem 1 remains the same.

4.3. Symmetric Link Set Capacity Constraints

For a switch, it may be required that the link sets be installed in pairs with the same capacity in both directions. In this case, a symmetric capacity constraint needs to be considered. To express this constraint, it is neater to denote a link by its originating node i and terminating node j as (i,j) than by an index number as l. Then the symmetric capacity constraint is given below.

$$\begin{cases} A_{(i,j)} = A_{(j,i)} \quad \forall (i,j),(j,i) \in L \text{ (for the old graph model)} \\ A_{(i',j)} = A_{(j',i)} \quad \forall (i,i'),(j,j') \in L_S, (i',j),(j',i) \in L_T \\ \qquad \text{(for the new graph model).} \end{cases}$$

This additional constraint has an impact on the decomposition of Subproblem 1 of the Lagrangean relaxation. Without the symmetric capacity constraint, Subproblem 1 can be further decomposed into $|L|$ independent problems. Whereas, with the constraint, those decomposed and independent problems need to be solved in pairs (problems corresponding to two link sets in opposite directions are solved jointly). Fortunately, this does not greatly complicate the solution procedure.

V. Computational Results

Two sets of computational experiments are performed. In the first set of experiments, we test the proposed sizing algorithm with respect to its (i) computational efficiency and (ii) effectiveness in determining good solutions. In the second set of experiments, we quantify how much the total capacity augmentation cost can be reduced by the sizing algorithm compared with the Most Congested First (MCF) heuristic described in Section I.

The link set sizing algorithm for SMDS networks described in Section III was coded in FORTRAN 77 and run on a SUN SPARC file server[2]. The algorithm was tested on three networks: OCT [13] (26 nodes), GTE [14] (12 nodes), and SITA[15] (10 nodes) whose topologies are shown in Figs. 3,

[2]Bellcore does not recommend or endorse products or vendors. Any mention of a product or vendor in this paper is to indicate the computing environment for the computational experiments discussed or to provide an example of technology for illustrative purposes; it is not intended to be a recommendation or endorsement of any product or vendor. Neither the inclusion of a product or a vendor in a computing environment or in this paper, nor the omission of a product or vendor, should be interpreted as indicating a position or opinion of that product or vendor on the part of the authors or Bellcore.

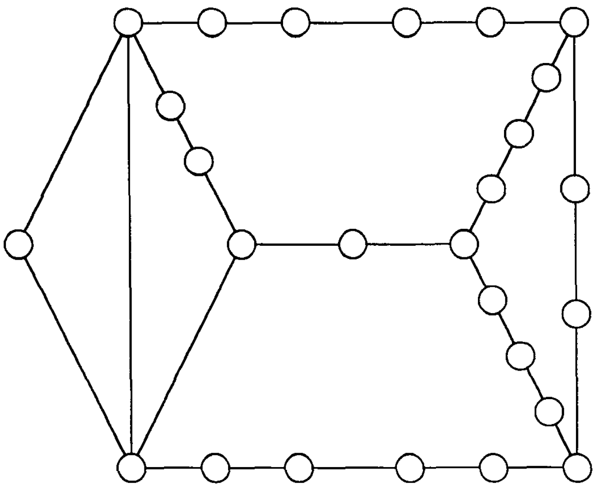

Fig. 3.

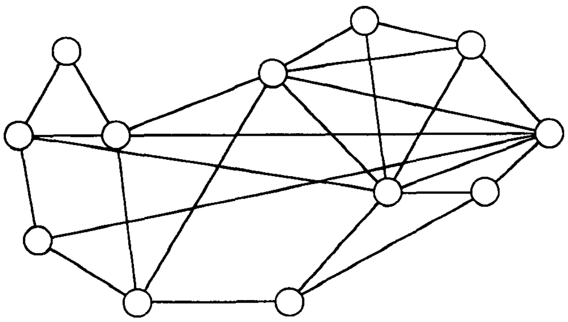

Fig. 4.

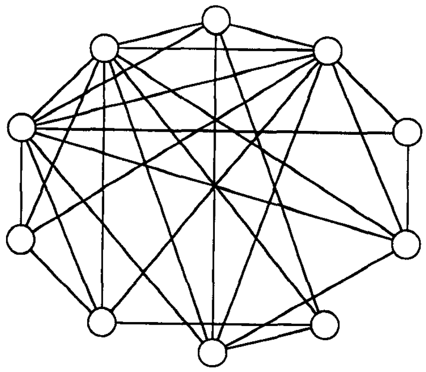

Fig. 5.

TABLE I
SUMMARY OF COMPUTATIONAL RESULTS OF THE SIZING ALGORITHM

Net ID	Traffic Req.	Existing Capacity	K	Ite.	B	Lower Bounds	Upper Bounds	Perc. Diff.	Time (s)
OCT	0.10	5	4	800	100	38.64	40.14	3.889	95.0
OCT	0.15	10	4	800	56	36.50	36.84	0.943	77.2
GTE	0.60	4	5	1000	28	28.00	28.33	1.161	20.2
GTE	0.45	3	5	1000	20	28.27	28.58	1.091	20.2
GTE	0.30	2	5	1000	16	27.20	27.91	2.610	19.8
SITA	0.20	1	5	1000	5	21.20	22.00	3.774	15.4
SITA	1.00	5	5	1000	16	22.64	22.91	1.190	14.4
SITA	1.50	5	5	1000	50	29.80	30.30	1.678	14.6

4, and 5, respectively. For each of the three networks, it was assumed that for each O–D pair the total individually addressed traffic rate at which packets are generated is the same (uniform traffic demand). Since the amount of group addressed traffic is expected to be small compared with the individually addressed traffic, the group addressed traffic is not considered in the computational experiments. It is also assumed that only one type of links (DS3 lines) are available and that the cost for each additional link is 1.

The given budget B is initially calculated by applying the MCF heuristic. The ACF heuristic is applied to find primal feasible solutions. As mentioned in Section III (the overall algorithm), the lowest feasible capacity augmentation cost found in solving the current (IP1) is recorded and used as the new (tighter) budget in a new (IP1). This process is repeated until no tighter budget is found.

To solve (D1), the subgradient method described in Section III was applied. In our implementation, Z^h_{IP1} was initially chosen as $\sum_{l\in L}\overline{\rho_l}$ (an upper bound on the total link set utilization factors if (IP1) is feasible) and updated to the best upper bound found so far in each iteration. In (32), δ was initially set to 2 and halved whenever the objective function value did not improve in 30 iterations. The initial values of u_l, v_l and β were chosen to be $1/C_l$, 0 and $\sum_{l\in L}\overline{\rho_l}/\sum_{l\in L}C_l$, respectively.

We first show that the proposed sizing algorithm performs well under tight budget constraints. The given budget is chosen so that in the course of solving (IP1) all the feasible capacity augmentation plans found use up the given budget. Table I summarizes the results of the computational experiments with the proposed sizing algorithm. The second column gives the traffic requirement for each O-D pair (normalized by the DS3 line capacity) . The third column specifies the existing capacity of each link in each network (also normalized by the DS3 line capacity). The fourth column specifies the value of counter limit K used in the ACF heuristic to find primal feasible solutions. The fifth column gives the number of iterations (the number of (LR1)'s solved) executed to solve (IP1)/(D1). The sixth column provides the given budget. The seventh column is the largest lower bound on the optimal objective function value found in the number of iterations specified in the fifth column. Recall that this is the best objective function value of the dual problem. The eighth column gives the best objective function value for (IP1) in the number of iterations specified in the fifth column. The ninth column reports the percentage difference ([upper-bound − lower-bound] × 100 / lower-bound) which is an upper bound on how far the best feasible solution found is from an optimal solution. The tenth column provides the CPU times which include the time to input the problem parameters.

Table I shows that the sizing algorithm is efficient and effective in finding near-optimal solutions given tight budget

TABLE II
COMPARISON OF THE SIZING ALGORITHM WITH THE MCF HEURISTIC

Net ID	Traffi Req.	Existing Capacity	K	Ite.	# of (IP1)'s solved	B^{MCF}	B^h	Perc. Impru.	Time (s)
OCT	0.10	5	4	800	3	224	100	124.0	258.8
OCT	0.15	10	4	800	2	60	56	7.1	153.1
GTE	0.60	4	5	1000	2	45	28	60.7	40.1
GTE	0.45	3	5	1000	2	30	20	50.0	40.0
GTE	0.30	2	5	1000	2	27	16	68.8	40.3
SITA	0.20	1	5	1000	2	11	5	120.0	29.1
SITA	1.00	5	5	1000	2	24	16	50.0	27.1
SITA	1.50	5	5	1000	5	126	50	152.0	29.2

constraints. For every test problem (networks with 10–26 nodes), the algorithm determines a solution that is within 4% of an optimal solution in less than 2 min of CPU time on a SUN SPARC file server.

Another set of experiments was performed to compare the sizing algorithm with the MCF heuristic. For each test problem, a number of (IP1)'s are solved and the given budget is updated (reduced) until no more improvement is achieved (see the Overall Algorithm in Section III). A comparison of the performance of the sizing algorithm with the performance of the MCF is reported in Table II. The first five columns in Table II provide the same information as the first five columns in Table I. The sixth column gives the number of (IP1)'s solved before the algorithm terminates. The seventh column reports B^{MCF}, the cost obtained by applying the MCF heuristic. This value is used as the initial given budget. The eighth column reports B^h, the best primal objective function value found by the proposed sizing algorithm. The ninth column gives the percentage improvement of the sizing algorithm over the MCF heuristic $[100 \times (B^{MCF} - B^h)/B^h]$.

The results in Table II show that using the proposed sizing algorithm results in an 7–152% (79% on the average) improvement in the total capacity augmentation cost over the MCF heuristic. In addition, the number of (IP1)'s solved for each test case is at most 3.

VI. SUMMARY

Switched Multi-megabit Data Service (SMDS) is a high-speed, connectionless, public, packet switching service that will extend Local Area Network (LAN)-like performance beyond the subscriber's premises, across a metropolitan or wide area. The SMDS service is considered as the first step towards the BISDN-based services and is thus strategically important for the BCC's.

To satisfy the performance objectives and, on the other hand, to avoid excessive engineering, it is essential that the capacity of SMDS networks be carefully managed. When performance exceptions are identified by a monitoring process, one may either reroute the traffic or resize the network to reduce the degree of network overload. However, when the load exceeds the network capacity, routing alone cannot resolve the overload problem and additional capacity is needed.

In this paper, a sizing approach to reducing network overload is described. The objective is to determine the minimum amount of additional capacity needed for an exhausted network and where to add the capacity. As demonstrated by a simple example, a commonly used greedy heuristic failed to provide satisfactory solutions. An optimization-based approach is then taken to attack the problem. We consider two combinatorial optimization problem formulations. In the first formulation, the objective is to minimize the total routing cost (to enforce the default routing protocol in SMDS networks) subject to a budget constraint. In the second formulation, we minimize the total capacity augmentation cost subject to a set of shortest-path-routing constraints. These two formulations are compared for their relative applicability and complexity.

A solution procedure based upon Lagrangean relaxation is proposed for the first formulation. In computational experiments, the proposed algorithm determines solutions that are within a few percent of an optimal solution in minutes of CPU time of a SUN SPARC file server for networks with 10–26 nodes. In addition, the proposed algorithm is compared with a Most Congested First (MCF) heuristic. For the test networks, the proposed algorithm achieves 7–152% (79% on the average) improvement in the total capacity augmentation cost over the MCF heuristic.

This work has the following significance. First, the problem is formally formulated as mathematical programs, which clearly demonstrates the difficulty of the problem and facilitates optimization-based solution approaches. Second, the proposed sizing algorithm has been computationally shown to be efficient and effective. The algorithm can thus help the BCC's expand SMDS network capacities in an economical way. Third, the formulations and algorithm developed can easily be generalized to consider the joint link set and node sizing problem for SMDS networks (by a different interpretation to the graph model). Last, by letting the existing node and link set capacities for potential locations be zero, this work can be used to solve the topological design and capacity assignment problem for SMDS networks.

REFERENCES

[1] Bell Communications Research, "Generic system requirements in support of SMDS," TR-TSV-000772, issue 1, May 1991.

[2] T. Cox and F. Dix, "SMDS: The Beginning of WAN Superhighways," *Data Commun.*, Apr. 1991.

[3] J. L. Wang and E. A. White, "An integrated methodology for supporting packet network engineering," in *Proc. IEEE Globecom*, Dec. 1990.

[4] J. L. Wang, "An integrated methodology for supporting network planning and traffic engineering with considerations to SMDS service," in *Proc. IEEE Globecom*, Dec. 1991.

[5] F. Y. S. Lin and J. L. Wang, "Minimax open shortest path first routing algorithms in networks supporting the SMDS service," in *Proc. ICC'93*, 1993.

[6] A. Albert, *Regression and the Moore–Penrose Pseudo-Inverse*. New York: Academic Press, 1972.

[7] Bell Communications Research, "Generic requirements for SMDS networking," TA-TSV-001059, issue 2, Aug. 1992.

[8] The OSPF Specification, Ver. 2, Network Working Group Request for Comments (RFC 1131), OSPFIGP Working Group of the Internet Engineering Task Force, June 1990.

[9] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerical Mathematics*, vol. 1, pp. 269–271, Oct. 1959.

[10] F. Y. S. Lin, "Allocation of end-to-end delay objectives for networks supporting SMDS," presented at the Second IEEE Network Management and Control Workshop, Sept. 1993.

[11] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] A. M. Geoffrion, "Lagrangean Relaxation and its uses in integer programming," *Math. Programming Study*, vol. 2, pp. 82–114, 1974.
[13] B. Gavish and S. L. Hantler, "An algorithm for optimal route selection in SNA networks," *IEEE Trans. Commun.*, vol. COM-31, pp. 1154–1160, Oct. 1983.
[14] B. Gavish and I. Neuman, "Routing in networks with unreliable components," *IEEE Trans. Commun.*, vol. 40, pp. 1248–1258, July 1992.
[15] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis* Reading, MA: Addison-Wesley, 1987, p. 8.

Frank Y. S. Lin received the B. S. degree in electrical engineering from National Taiwan University in 1983 and the Ph. D. degree in electrical engineering from the University of Southern California in 1991.

Since 1991, he has been with Bell Communications Research, where he is involved in developing algorithms for network planning and engineering. His current research interests include computer network optimization, high-speed networks, distributed algorithms, and performance modeling.