Link Set Capacity Augmentation Algorithms for Networks Supporting SMDS

Frank Y.S. Lin

Bell Communications Research
444 Hoes Lane
Piscataway, New Jersey 08854
TEL:(908)699-4220 FAX:(908)336-2275
lin1@cc.bellcore.com

Abstract

To expand the link set capacities for networks supporting 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.

Solution procedures based upon Lagrangean relaxation are proposed for the formulations. In computational experiments, it is clearly demonstrated that there is a computation-time versus solution-quality trade-off between the algorithms for the two formulations. In addition, to demonstrate the effectiveness of the proposed algorithms, we compare the proposed algorithms with a Most Congested First (MCF) heuristic. For the test networks, the proposed algorithms achieve up to 250% (79% on the average) improvement in the total capacity augmentation cost over the MCF heuristic.

1. 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, backbone networks supporting the SMDS service (referred to as SMDS networks) must be carefully managed. The INPLANS™ 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.

In this paper, a capacity augmentation approach to reducing network overload that persists for a long time is described. The proposed link set capacity augmentation algorithms 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 expand the link set capacities, network planners/administrators must know the inter-switching system routing strategy. The routing algorithm for SMDS networks is specified in [6]. 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[7]. 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[8] 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 link set capacity augmentation 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).

In this paper, we present two integer programming formulations for the SMDS link set capacity augmentation 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.

Three networks (10 test cases) with 10 to 14 nodes were tested in the computational experiments. The proposed algorithms determine solutions within minutes of CPU time. It is also shown that the proposed algorithm for the second formulation consistently calculate as good or better solutions than the algorithm for the first formulation at the cost of more computation time. Compared with a Most Congested First (MCF) heuristic, the proposed algorithms achieved up to 250% 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 capacity augmentation algorithm can help the BCCs expand SMDS network capacities in an economical way. Third, the formulations and algorithms developed can easily be generalized to consider the joint link set and node capacity augmentation 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.

INPLANS is a trademark of Bellcore.

0-7803-1825-0/94 $4.00 © 1994 IEEE

The remainder of this paper is organized as follows. In Section 2, two formulations of the SMDS link set capacity augmentation problem are given and compared. Solution procedures based upon Lagrangean relaxation are proposed for the two formulations in Sections 3 and 4, respectively. In Section 5, computational results are reported. Section 6 summarizes this paper.

2. 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 $V = \{1,2,...,N\}$ be the set of nodes and L be the set of links in the graph (network). 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 (SSs) 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/sec), 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/sec and the added capacity is A_l packets/sec (a decision variable).

For each O-D pair $(o,d) \in W$, let x_p be 1 if path p is used to transmit the individually addressed packets for O-D pair (o,d) and 0 otherwise. 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 δ_{pl} be the indicator function which is 1 if link l is on path p and 0 otherwise.

For each origin o, let y_t be 1 if spanning tree $t \in T_o$ is used to transmit the multicast message for origin o and 0 otherwise. SMDS SSs 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 σ_{tl} be the indicator function which is 1 if link l is on tree t and 0 otherwise.

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 $\bar{\rho}_l$ 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[9]. The SMDS link set capacity augmentation 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. For each link $l \in L$, let l' denote the link set in the opposite direction.

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

subject to:

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

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{od},\ (o,d) \in W \quad (2)$$

$$\sum_{(o,d) \in W} \sum_{p \in P_{od}} x_p\, \delta_{pl}\, \gamma_{od} + \sum_{o \in V} \sum_{t \in T_o} y_t\, \alpha_o\, \sigma_{tl} \le (C_l + A_l)\, \bar{\rho}_l \qquad \forall l \in L \quad (3)$$

$$\sum_{d \in V-\{o\}} \sum_{p \in P_{od}} x_p\, \delta_{pl} \le (N-1) \sum_{t \in T_o} y_t\, \sigma_{tl} \qquad \forall l \in L,\ o \in V \quad (4)$$

$$\sum_{t \in T_o} y_t = 1 \qquad \forall o \in V \quad (5)$$

$$y_t = 0 \text{ or } 1 \qquad \forall t \in T_o,\ o \in V \quad (6)$$

$$A_l \in K_l \qquad \forall l \in L \quad (7)$$

$$A_l = A_{l'} \qquad \forall l \in L \quad (8)$$

$$\sum_{l \in L} \Phi_l(A_l) \le B. \qquad (9)$$

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 satisfy the end-to-end delay objectives). Constraints (5) and (6) require that all of the multicast traffic from one origin be transmitted over 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 link sets be installed in pairs with the same capacity in both directions. Constraint (9) 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 constraint (9), 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

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

subject to:

$$(1)-(9)$$

$$\sum_{(o,d) \in W} \sum_{p \in P_{od}} x_p\, \delta_{pl}\, \gamma_{od} \le f_l \qquad \forall l \in L \quad (10)$$

$$0 \le f_l \le (C_l + A_l)\, \bar{\rho}_l \qquad \forall l \in L. \quad (11)$$

For each link l, an auxiliary variable f_l is introduced. Since the objective function is strictly increasing with f_l and (IP1) is a minimization problem, equality of (10) 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 (11) gives the range of f_l.

2.2 Formulation 2

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

subject to:

$$\sum_{p\in P_{od}} x_p = 1 \qquad \forall (o,d)\in W \quad (12)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_{od}, (o,d)\in W \quad (13)$$

$$\sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p\, \delta_{pl}\, \gamma_{od} + \sum_{o\in V} \sum_{t\in T_o} y_t\, \alpha_o\, \sigma_{tl} \le (C_l + A_l)\,\bar{\rho}_l \qquad \forall l\in L \quad (14)$$

$$\sum_{d\in V-\{o\}} \sum_{p\in P_{od}} x_p\, \delta_{pl} \le (N-1) \sum_{t\in T_o} y_t\, \sigma_{tl} \qquad \forall l\in L,\ o\in V \quad (15)$$

$$\sum_{t\in T_o} y_t = 1 \qquad \forall o\in V \quad (16)$$

$$y_t = 0 \text{ or } 1 \qquad \forall t\in T_o,\ o\in V \quad (17)$$

$$A_l \in K_l \qquad \forall l\in L \quad (18)$$

$$A_l = A_{l'} \qquad \forall l\in L \quad (19)$$

$$\sum_{l\in L} \sum_{q\in P_{od}} \frac{x_q\, \delta_{ql}}{C_l + A_l} \le \sum_{l\in L} \frac{\delta_{pl}}{C_l + A_l} \qquad \forall p\in P_{od}, (o,d)\in W. \quad (20)$$

The objective function is to minimize the total cost of capacity augmentation. Constraints (12)-(19) are the same as (1)-(8). The left hand side of (20) (together with (12) and (13)) is the routing cost for O-D pair (o,d) (for one unit of flow on the selected path). The right hand side of (20) is the cost of path $p \in P_{od}$. Constraint (20) requires that for each O-D pair a shortest path be used to carry the individually addressed traffic.

An equivalent formulation of $\overline{\text{IP2}}$ is

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

subject to

$$(12)-(19)$$

$$\sum_{p\in P_{od}} x_p\, \delta_{pl} \le f_{odl} \qquad \forall l\in L,\ (o,d)\in W \quad (21)$$

$$f_{odl} = 0 \text{ or } 1 \qquad \forall l\in L,\ (o,d)\in W \quad (22)$$

$$\sum_{l\in L} \frac{f_{odl}}{C_l + A_l} \le \sum_{l\in L} \frac{\delta_{pl}}{C_l + A_l} \qquad \forall p\in P_{od}, (o,d)\in W. \quad (23)$$

For each link l and O-D pair (o,d), an auxiliary variable f_{odl} is introduced. (21) should be an equality and the inequality is a relaxation. However, it can be shown that equality of (21) will hold in an optimal solution. As the reader will see in Section 4, the introduction of f_{odl} decouples the problem into three independent subproblems in the Lagrangean Relaxation. Constraint (22) gives the range of f_{odl}.

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, 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 (23) 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) ((13) and (17) 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.

3. A Solution Procedure to Formulation 1

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), (9) and (10) to obtain the following relaxation

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

subject to:

$$\sum_{p\in P_{od}} x_p = 1 \qquad \forall (o,d)\in W \quad (24)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_{od}, (o,d)\in W \quad (25)$$

$$\sum_{t\in T_o} y_t = 1 \qquad \forall o\in V \quad (26)$$

$$y_t = 0 \text{ or } 1 \qquad \forall t\in T_o,\ o\in V \quad (27)$$

$$A_l \in K_l \qquad \forall l\in L \quad (28)$$

$$A_l = A_{l'} \qquad \forall l\in L \quad (29)$$

$$0 \le f_l \le (C_l + A_l)\,\bar{\rho}_l \qquad \forall l\in L \quad (30)$$

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.

(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:

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

subject to:

$$A_l \in K_l \qquad \forall l\in L \quad (31)$$

$$A_l = A_{l'} \qquad \forall l\in L \quad (32)$$

$$0 \le f_l \le (C_l + A_l)\,\bar{\rho}_l \qquad \forall l\in L \quad (33)$$

Subproblem 2:

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

subject to:

$$\sum_{p\in P_{od}} x_p = 1 \qquad \forall (o,d)\in W \tag{34}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_{od},\ (o,d)\in W \tag{35}$$

and

Subproblem 3:

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

subject to:

$$\sum_{t\in T_o} y_t = 1 \qquad \forall o\in V \tag{36}$$

$$y_t = 0 \text{ or } 1 \qquad \forall t\in T_o,\ o\in V. \tag{37}$$

Subproblem 1 is composed of $|L/2|$ (one for each pair of links in opposite directions) problems. Since A_l is discrete and bounded, the problem can be solved by successively fixing A_l to all possible values that satisfy (31).

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.

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[10] to solve the problem.

For any $(v,s,\beta,u) \ge 0$, 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} [11]. The dual problem (D1) is

$$Z_{D1} = \max_{(v,s,\beta,u)\ge 0} Z_{D1}(v,s,\beta,u). \tag{D1}$$

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 are available 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

$$t^k = \delta\, \frac{Z_{IP1}^h - Z_{D1}(m^k)}{\|b^k\|^2} \tag{38}$$

where Z_{IP1}^h is an objective function value for a heuristic solution (upper bound on Z_{IP1}) and δ is a constant, $0 < \delta \le 2$. 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 following Most Congested First (MCF) heuristic to place additional capacity.

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.

The MCF Heuristic:

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

If the total cost is less than the given budget, then a primal feasible solution is found. Another alternative to the MCF heuristic is the following All Congested Simultaneously (ACS) heuristic to find primal feasible solutions.

The ACS Heuristic:

1. Set the counter limit K to be a prespecified value.
2. If $K = 0$, stop; otherwise, decrease K by 1.
3. Find the link sets where the capacity (utilization) constraints are violated.
4. For each pair of link sets in opposite directions which consist of at least one link set identified in Step 3, add the minimum number of links (the same number for both link sets) to satisfy the current flows for both link sets.
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 ((IP) is an integer programming problem). The following implementation attempts to achieve better efficiency.

The Overall Algorithm A1 for Formulation 1:

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.

4. A Solution Procedure to Formulation 2

The basic approach to the development of a solution procedure to Formulation 2 is also Lagrangean relaxation. For Formulation 2 (Problem (IP2)), we dualize constraints (14), (15), (21) and (23) to obtain the following relaxation

$$\begin{aligned} Z_{D2}(v,s,\mu,\theta) = \min & \sum_{l\in L} \Phi_l(A_l) + \sum_{l\in L} v_l \Big[\sum_{(o,d)\in W} \sum_{p\in P_{od}} x_p\, \delta_{pl}\, \gamma_{od} \\ & + \sum_{o\in V} \sum_{t\in T_o} y_t\, \alpha_o\, \sigma_{tl} - (C_l + A_l)\, \rho_l \Big] \\ & + \sum_{l\in L} \sum_{o\in V} s_{ol} \Big[\sum_{d\in V-\{o\}} \sum_{p\in P_{od}} x_p\, \delta_{pl} - (N-1) \sum_{t\in T_o} y_t\, \sigma_{tl} \Big] \\ & + \sum_{(o,d)\in W} \sum_{p\in P_{od}} \mu_p \Big(\sum_{l\in L} \frac{f_{odl}}{C_l + A_l} - \sum_{l\in L} \frac{\delta_{pl}}{C_l + A_l} \Big) \\ & + \sum_{l\in L} \sum_{(o,d)\in W} \theta_{odl} \Big(\sum_{p\in P_{od}} x_p\, \delta_{pl} - f_{odl} \Big) \end{aligned} \tag{LR2}$$

subject to:

Table 1. Summary of computational results

Net ID	Traf. Req.	Exis. Cap.	Z^{MCF}	(IP1)'s Solved	Z^{A1}	$\frac{Z^{MCF}-Z^{A1}}{Z^{A1}}$ (%)	Time (A1) (sec)	Z^{A2}	$\frac{Z^{MCF}-Z^{A2}}{Z^{A2}}$ (%)	Time (A2) (sec)	Lower Bound
PSS	0.60	5	172	2	166	3.6	59.5	130	32.3	237.9	88
PSS	0.50	5	116	2	108	7.4	61.4	94	23.4	234.0	54
PSS	0.40	5	48	2	42	14.3	59.1	42	14.3	233.8	22
PSS	0.30	5	10	2	6	66.7	50.4	6	66.7	243.5	0
GTE	0.60	4	46	3	26	76.9	64.0	24	91.7	205.0	6
GTE	0.45	3	30	2	20	100.0	34.3	20	100.0	200.4	4
GTE	0.30	2	26	2	14	85.7	41.8	14	85.7	201.6	2
SITA	1.00	5	24	3	16	50.0	32.4	10	140.0	164.5	0
SITA	1.50	5	126	2	54	133.3	27.5	46	173.9	163.5	4
SITA	0.20	1	14	2	10	40.0	23.0	4	250.0	141.0	0

$$\sum_{p \in P_{od}} x_p = 1 \qquad \forall (o,d) \in W \tag{39}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{od},\ (o,d) \in W \tag{40}$$

$$\sum_{t \in T_o} y_t = 1 \qquad \forall o \in V \tag{41}$$

$$y_t = 0 \text{ or } 1 \qquad \forall t \in T_o,\ o \in V \tag{42}$$

$$A_l \in K_l \qquad \forall l \in L \tag{43}$$

$$A_l = A_{l'} \qquad \forall l \in L \tag{44}$$

$$f_{odl} = 0 \text{ or } 1 \qquad \forall (o,d) \in W,\ l \in L \tag{45}$$

where v, s, μ and θ are the vectors of $\{v_l\}$, $\{s_{ol}\}$ $\{\mu_p\}$ and $\{\theta_{odl}\}$, respectively. Like (LR1), the constraints are dualized in such a way that the corresponding multiplers are nonnegative.

(LR2) can also be decomposed into three independent subproblems.

Subproblem 1:

$$Z^1_{D2}(v,\mu,\theta) = \min \sum_{l \in L} [\Phi_l(A_l) - v_l A_l \bar{\rho}_l + \sum_{(o,d) \in W} \sum_{p \in P_{od}} \mu_p \frac{(f_{odl} - \delta_{pl})}{C_l + A_l} - \sum_{(o,d) \in W} \theta_{odl} f_{odl}]$$

subject to:

$$A_l \in K_l \qquad \forall l \in L \tag{46}$$

$$A_l = A_{l'} \qquad \forall l \in L \tag{47}$$

$$0 \le f_l \le (C_l + A_l)\bar{\rho}_l \qquad \forall l \in L \tag{48}$$

Subproblem 2:

$$Z^2_{D2}(v,s,\theta) = \min \sum_{l \in L} \sum_{(o,d) \in W} \sum_{p \in P_{od}} [v_l \gamma_{od} + s_{ol} + \theta_{odl}]\, x_p\, \delta_{pl}$$

subject to:

$$\sum_{p \in P_{od}} x_p = 1 \qquad \forall (o,d) \in W \tag{49}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{od},\ (o,d) \in W \tag{50}$$

and

Subproblem 3:

$$Z^3_{D2}(v,s) = \min \sum_{l \in L} \sum_{o \in V} \sum_{t \in T_o} (v_l \alpha_o - (N-1)\, s_{ol})\, y_t\, \sigma_{tl}$$

subject to:

$$\sum_{t \in T_o} y_t = 1 \qquad \forall o \in V \tag{51}$$

$$y_t = 0 \text{ or } 1 \qquad \forall t \in T_o,\ o \in V. \tag{52}$$

Subproblem 1 is composed of $|L/2|$ (one for each pair of links in opposite directions) problems. Since A_l is discrete and bounded, the problem can be solved by successively fixing A_l to all possible values that satisfy (46). For a fixed A_l, f_{odl} is 1 if $\sum_{p \in P_{od}} \mu_p/(C_l + A_l) \le \theta_{odl}$ and 0 otherwise.

Subproblem 2 consists of $|W|$ (one for each O-D pair) shortest path problems where $v_l \gamma_{od} + s_{ol} + \theta_{odl}$ 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.

Subproblem 3 of (LR2) is the same as that of (LR1) and can be solved by the same way.

Like (D1), the following dual problem is constructed

$$Z_{D2} = \max_{(v,s,\mu,\theta) \ge 0} Z_{D2}(v,s,\mu,\theta). \tag{D2}$$

and the subgradient method is applied to solve (D2).

To find good primal feasible solutions, the same heuristic as used in solving (IP1) is applied. The overall algorithm is described below.

The Overall Algorithm A2 for Formulation 2:

1. Initialize parameters.
   - Apply the MCF heuristic to calculate an initial value of Z^h_{IP2}, an upper bound to be used in the subgradient method.
   - Initialize the multipliers.
   - Set the iteration counter K to be a prespecified value.
2. If $K = 0$, stop; otherwise, decrease K by 1.
3. Solve (LR2).
4. Calculate primal heuristic solutions.
5. Adjust the multipliers.
6. Go to Step 2.

5. Computational Results

In the computational experiments, we test the proposed algorithms with respect to their relative efficiency and effectiveness. We also quantify how much the total capacity augmentation cost can be reduced by the proposed algorithms compared with the MCF heuristic described earlier.

The link set capacity augmentation algorithms for SMDS networks described in Sections 3 and 4 are coded in FORTRAN 77 and run on a SUN SPARC file server[1]. The algorithms are tested on three networks: PSS (14 nodes), GTE (12 nodes) and SITA (10 nodes) whose topologies can be found in [12]. For each of the three networks, it is 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.

---

1. 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.

For Algorithm A1, the given budget B is initially calculated by applying the MCF heuristic. The ACS heuristic is applied to find primal feasible solutions. The iteration limit used in the ACS heuristic is 5. As mentioned in Section 3, 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 3 applied. In our implementation, Z_{IP1}^h is initially chosen as $\sum_{l \in L} \rho_l$ (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 Equation (36), δ is initially set to 2 and halved whenever the objective function value does not improve in 30 iterations. The iteration counter is initially set to 1000. The initial values of u_l, v_l and β are chosen to be $1/C_l$, 0 and $\sum_{l \in L} \rho_l / \sum_{l \in L} C_l$, respectively.

For Algorithm A2, the ACS heuristic is applied to to find primal feasible solutions. To solve (D2), the subgradient method is applied. The MCF heuristic is applied to calculate an initial value of Z_{IP2}^h, an upper bound on the optimal objective function value of (IP2), to be used in the subgradient method. The step size control parameter δ is initially set to 2 and halved whenever the objective function value does not improve in 30 iterations. The iteration counter is initially set to 500. The initial values of v_l, μ_p and θ_{odl} are chosen to be 0.

Table 1 summarizes the results of the computational experiments with the proposed algorithms. 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 reports the cost (number of additional links) calculated by the MCF heuristic, denoted by Z^{MCF}. The fifth column gives the number of iterations (the number of (IP1)'s solved) executed when Algorithm A1 is applied. The sixth column shows the cost calculated by Algorithm A1, denoted by Z^{A1}. The seventh column provides the percentage difference between Z^{MCF} and Z^{A1}. The eighth column is the CPU time for Algorithm A1, which includes the time to input the problem parameters. The ninth column shows the cost calculated by Algorithm A2, denoted by Z^{A2}. The tenth column provides the percentage difference between Z^{MCF} and Z^{A2}. The eleventh column is the CPU time for Algorithm A2, which includes the time to input the problem parameters. The last column is the largest lower bound on the optimal objective function value found by Algorithm A2. This is calculated by rounding up the best objective function value of the dual problem (D2) to the closed even number (the optimal objective function value of (IP2) is an even number).

From Table 1, we have the following observations. First, the proposed algorithms determine solutions in minutes of CPU time on a SUN SPARC file server for networks with 10 to 14 nodes. Second, for Algorithm 1, the number of (IP1)'s solved for each test case is at most 3. Third, Algorithm A2 takes more CPU time than Algorithm A1, but calculates as good or better solutions. Fourth, compared with the MCF heuristic, the proposed algorithms result in up to 250% (78% on the average) improvement in the total capacity augmentation cost over the MCF heuristic. Last, the lower bounds are loose compared with Z^{A2}, the best upper bounds obtained. We believed this is due to the large duality gap between Z_{D2} (the optimal objective function value of (D2)) and Z_{IP2} (the optimal objective function value of (IP2)).

6. 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 BCCs.

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 expand the network capacity to reduce the degree of network overload. Rerouting is usually a cost effective solution. However, when the load exceeds the network capacity, rerouting alone cannot resolve the overload problem and additional capacity is needed.

In this paper, a capacity augmentation approach to reducing network overload is described. The objective is to determine (i) the minimum cost to place additional link set capacity for an exhausted network to alleviate the overload problem and (ii) where to add the capacity. 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.

Solution procedures based upon Lagrangean relaxation are proposed for the formulations. In computational experiments, the proposed algorithms determine solutions in minutes of CPU time of a SUN SPARC file server for networks with 10 to 14 nodes. It is shown that the algorithm for the second formulation consistently calculate as good or better solutions than the algorithm for the first algorithm at the cost of longer computation time. In addition, the proposed algorithm is compared with a Most Congested First (MCF) heuristic. For the test networks, the proposed algorithm achieves up to 250% (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 algorithms have been computationally shown to be efficient and effective. The algorithms can thus help the BCCs expand SMDS network capacities in an economical way. Third, the formulations and algorithms developed can easily be generalized to consider the joint link set and node capacity augmentation problem for SMDS networks. 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 Communications, April 1991.
3. J.L. Wang and E.A. White, *An Integrated Methodology for Supporting Packet Network Engineering*, Proc. IEEE Globecom, December 1990.
4. J.L. Wang, *An Integrated Methodology for Supporting Network Planning and Traffic Engineering with Considerations to SMDS Service*, Proc. IEEE Globecom, December 1991.
5. F. Y.S. Lin and J.L. Wang, *Minimax Open Shortest Path First Routing Algorithms in Networks Supporting the SMDS Service*, Proc. ICC'93.
6. Bell Communications Research, *Generic Requirements for SMDS Networking*, TA-TSV-001059, Issue 2, August 1992.
7. *The OSPF Specification, Version 2*, Network Working Group Request for Comments (RFC 1131), OSPFIGP Working Group of the Internet Engineering Task Force, June 1990.
8. E.W. Dijkstra, *A Note on Two Problems in Connection with Graphs*, Numerical Mathematics, Vol. 1, pp. 269-271, October 1959.
9. F. Y.S. Lin, *Allocation of End-to-end Delay Objectives for Networks Supporting SMDS*, Proc. IEEE Globecom, November 1993.
10. A.S. Tanenbaum, Computer Networks, Prentice-Hall, 1988.
11. A.M. Geoffrion, *Lagrangean Relaxation and Its Uses in Integer Programming*, Math. Programming Study, Vol. 2, pp. 82-114, 1974.
12. F. Y.S. Lin and J. Yee, *A Real-time Distributed Routing and Admission Control Algorithm for ATM Networks*, Proc. IEEE Infocom, April 1993.