Three Algorithms for Routing and Flow Control in Virtual Circuit Networks

Yeong-Sung Lin [1] James R. Yee[1, 2]
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-0781
TEL: (213)-743-3919

Abstract

In this paper, the joint routing and flow control problem in virtual circuit networks is considered. We consider the problem of choosing a path and adjusting the input rate for each origin-destination pair in the network. In the first model we minimize the average number of packets in the network plus a throughput limitation cost. In the second model, we maximize the allocation to the most poorly treated user(s) subject to link utilization constraints. The third model, which is a variation of the second model, has an additional constraint which limits the average number of packets in the network. These joint routing and flow control problems are formulated as mixed integer programming problems.

The emphasis of this work is to develop near-optimal algorithms to solve these three optimization problems. The basic approach in this work is Lagrangean Relaxation which has been a common and successful technique in solving many difficult combinatorial optimization problems. In computational experiments, our algorithms determine solutions that are within a few percent of an optimal solution in minutes of CPU time for networks with 26 to 61 nodes.

1 Introduction

The performance of a computer network is strongly dependent on the congestion control used in the network. A congestion control scheme consists of a routing algorithm and a flow control scheme. In early research in networking, the routing problem [2], [5], [6] and the flow control problem [11], [15], [17] have been considered separately.

In order to achieve better overall congestion control, Rudin and Mueller [20] recommended that routing and flow control be considered as a joint problem rather than isolated ones. Gallager and Golestaani [7] showed that the joint routing and flow control problem for datagram networks can be formulated as a convex programming problem. Thaker and Cain [21] used the Gallager-Golestaani model to investigate the interaction between routing and flow control in datagram networks.

Over the past decade, many networks have been implemented with virtual circuit service, e.g. SNA [13], TYMNET [22], TELENET [14], and TRANSPAC [3]. Furthermore, future networks such as ISDN will provide virtual circuit service [10]. As such, it is important to develop a good fundamental understanding of the joint routing and flow control problem for virtual circuit networks.

Gavish and Hantler [8] formulated the routing problem for virtual circuit networks as a combinatorial optimization problem. They applied the Lagrangean Relaxation method to obtain good heuristic solutions. Hayden [15] proposed a maxmin flow control scheme to find a fair capacity allocation in a virtual circuit network. In [15], a single fixed path was assumed to be given for each user pair.

[1]Supported by a contract from AT&T.

[2]Supported in part by Contract DAAL 03-88-K0059 from the Army Research Office.

This paper may be viewed as a natural continuation or integration of some of the aforementioned work. In the first model, we use the Gallager-Golestaani approach for virtual circuit networks. Then we show the flow control algorithm of Hayden can be generalized to consider the routing problem jointly.

The remainder of this paper is organized as follows. In Section 2, three joint routing and flow control problems are formulated as mixed integer programming problems. In Section 3, an algorithm is developed for each of the three models. In Section 4, we report the computational experience with each of the three solution procedures. A summary of the paper is given in Section 5.

2 Problem Formulation

A virtual circuit communications network is modeled as a graph where the processors are represented by nodes and the communication channels are represented by arcs. Let $V = \{1, 2, ..., N\}$ be the set of nodes in the graph and let L denote the set of communication links in the network. For each link $l \in L$, the capacity is C_l packets/sec. Let W be the set of origin-destination (O-D) pairs in the network. For each O-D pair $w \in W$, the arrival of new traffic is modeled as a Poisson process with rate γ_w (packets/sec). For O-D pair w, all of the traffic is transmitted over one path in the set P_w, a given set of simple directed paths from the origin to the destination of O-D pair w.

For each O-D pair $w \in W$, let x_p be 1 when path $p \in P_w$ is used to transmit the packets for O-D pair w and 0 otherwise. In a virtual circuit network, all of the packets of a user pair are transmitted over one path from the origin to the destination. Thus $\sum_{p \in P_w} x_p = 1$. For each path p and link $l \in L$, let δ_{pl} denote the indicator function which is one if link l is on path p and zero otherwise. Then, the aggregate flow of packets over link l is given by the left hand side of (1).

In the network, there is a buffer for each outbound link. Using Kleinrock's independence assumption [18], the arrival of packets to each buffer is a Poisson process where the rate is the aggregate flow over the outbound link. It is assumed that the transmission time for each packet is exponentially distributed with mean C_l^{-1}. Thus, each buffer is modeled as an M/M/1 queue.

We present three optimization models for the joint routing and flow control problem for virtual circuit networks. In these models, the routing assignment and the input traffic rate for each O-D pair are to be determined so as to optimize the average number of packets in the system or fairness.

2.1 Model 1

In this optimization model, we employ the same approach used by Gallager and Golestaani [7], [12] to develop a joint routing and flow control algorithm for datagram networks. For O-D pair w, let $\bar{\gamma}_w$, γ_w, and y_w be the offered, the accepted, and

the rejected traffic rates, respectively. Note that $\bar{\gamma}_w$ is given, while γ_w and y_w are variables. In the model, an artificial link is introduced for each O-D pair to carry the rejected traffic. Associated with each artificial link, a cost function to account for throughput limitation is given. For O-D pair w, the cost function of the artificial link, denoted by $E_w(y_w)$, is assumed to be (i) convex, (ii) monotonically increasing and (iii) differentiable. In addition, $E_w(y_w) \to \infty$ as $y_w \to \bar{\gamma}_w$. A joint routing and flow control problem is formulated as the following nonlinear mixed integer programming problem.

$$Z_{\overline{IP1}} = \min \sum_{l \in L} \frac{\sum_{w \in W} \sum_{p \in P_w} x_p \gamma_w \delta_{pl}}{C_l - \sum_{w \in W} \sum_{p \in P_w} x_p \gamma_w \delta_{pl}} + \sum_{w \in W} E_w(y_w) \quad (\overline{IP1})$$

subject to

$$\sum_{w \in W} \sum_{p \in P_w} x_p \gamma_w \delta_{pl} \leq C_l \quad \forall l \in L \quad (1)$$

$$\sum_{p \in P_w} x_p = 1 \quad \forall w \in W \quad (2)$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (3)$$

$$\gamma_w + y_w = \bar{\gamma}_w \quad \forall w \in W \quad (4)$$

$$\gamma_w, y_w \geq 0 \quad \forall w \in W. \quad (5)$$

The objective function represents the average number of packets in the actual network plus a throughput limitation cost (penalty incurred by rejecting traffic). Constraint (1) requires that the aggregate flow not exceed the capacity for each link. Constraints (2) and (3) require that all of the traffic of an O-D pair be transmitted over exactly one path. Constraint (4) requires that the offered traffic be either accepted by the network or routed through the artificial link (rejected). Constraint (5) requires that the input traffic rates (accepted and rejected) be nonnegative.

An equivalent formulation of $(\overline{\text{IP1}})$ is given by (IP1) below. We redefine x_p to be the rate at which packets for O-D pair $w \in W$ are transmitted over path $p \in P_w$. (IP1) is better suited for the application of the Lagrangean Relaxation method.

$$Z_{IP1} = \min \sum_{l \in L} \frac{f_l}{C_l - f_l} + \sum_{w \in W} E_w(y_w) \quad (IP1)$$

subject to

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \leq f_l \quad \forall l \in L \quad (6)$$

$$0 \leq f_l \leq C_l \quad \forall l \in L \quad (7)$$

$$\sum_{p \in P_w} x_p = \gamma_w \quad \forall w \in W \quad (8)$$

$$x_p = 0 \text{ or } \gamma_w \quad \forall p \in P_w, w \in W \quad (9)$$

$$\gamma_w + y_w = \bar{\gamma}_w \quad \forall w \in W \quad (10)$$

$$\gamma_w, y_w \geq 0 \quad \forall w \in W. \quad (11)$$

For each link l, a variable f_l is introduced. This technique has been applied in [8]. We interpret these variables to be "estimates" of the aggregate flows. Since the objective function is strictly increasing with f_l and (IP1) is a minimization problem, each f_l will equal the aggregate flow in an optimal solution.

2.2 Model 2

For each O-D pair w, γ_w is a control variable which represents the capacity allocated to user pair w. Let $\Pi = \{(\mathbf{x}, \gamma) \geq 0 | \sum_{p \in P_w} x_p = \gamma_w \ \forall w \in W \text{ and } x_p = 0 \text{ or } \gamma_w \ \forall p \in P_w, w \in W\}$. The problem of determining a path and accepted input traffic rate for each O-D pair to maximize the smallest capacity allocation to any user pair is formulated as the following linear mixed integer programming problem.

$$Z_{\overline{IP2}} = \max \min_{w \in W} \gamma_w \quad (\overline{IP2})$$

subject to $(\mathbf{x}, \gamma) \in \Pi$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \leq C_l \quad \forall l \in L. \quad (12)$$

The objective function represents the minimum allocation to any user pair. Hayden [15] considered the problem of controlling the input rate (a single routing path for each user pair was assumed to be given) for each O-D pair to maximize the smallest capacity allocation to any user. Thus, $(\overline{\text{IP2}})$ is a generalization of the problem considered in [15].

The above formulation has the disadvantage that in any optimal solution some link(s) must be saturated. To eliminate this problem, C_l is replaced by $\alpha_l C_l$ where $0 < \alpha_l < 1$, $\forall l \in L$. In addition, let $s = \min_{w \in W} \gamma_w$. Then we convert the resulting maximization problem to the problem of minimizing $-s$. Our motivation for doing this is that it is easier to follow the algorithm development in Section 3 if each of the three models are stated as minimization problems. An equivalent formulation of $(\overline{\text{IP2}})$ is

$$Z_{IP2} = \min -s \quad (IP2)$$

subject to $(\mathbf{x}, \gamma) \in \Pi$

$$s \leq \gamma_w \quad \forall w \in W \quad (13)$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \leq \alpha_l C_l \quad \forall l \in L. \quad (14)$$

2.3 Model 3

This model is a variation of Model 2. In Model 2, each link was prevented from being saturated by constraining the link utilization factor for link l to not exceed α_l. In Model 3, we prevent each link from being saturated by using a system congestion constraint. That is, the average number of packets in the system is constrained not to exceed a given upper bound J. The formulation is given below.

$$Z_{IP3} = \min -s \quad (IP3)$$

subject to $(\mathbf{x}, \gamma) \in \Pi$

$$s \leq \gamma_w \quad \forall w \in W \quad (15)$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \leq f_l \quad \forall l \in L \quad (16)$$

$$f_l \leq C_l \quad \forall l \in L \quad (17)$$

$$\sum_{l \in L} \frac{f_l}{C_l - f_l} \leq J. \quad (18)$$

For similar reasons as in Model 1, a set of auxiliary variables $\{f_l\}$ is introduced.

3 Solution Procedures

The basic approach to solving the above three mixed integer programming problems 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.

3.1 A Solution Procedure for Model 1

For the first model we dualize constraint (6) to obtain the following relaxation

$$Z_{D1}(u) = \min \sum_{l\in L} \frac{f_l}{C_l - f_l} + \sum_{w\in W} E_w(y_w) + \sum_{l\in L} u_l\Big(\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pl} - f_l\Big) \quad (LR1)$$

subject to

$$0 \le f_l \le C_l \quad \forall l \in L \quad (19)$$

$$\sum_{p\in P_w} x_p = \gamma_w \quad \forall w \in W \quad (20)$$

$$x_p = 0 \text{ or } \gamma_w \quad \forall p \in P_w, w \in W \quad (21)$$

$$\gamma_w + y_w = \bar{\gamma}_w \quad \forall w \in W \quad (22)$$

$$\gamma_w, y_w \ge 0 \quad \forall w \in W. \quad (23)$$

It should be easy to verify the solution to (LR1) is as follows. For each link $l \in L$, the solution f_l is

$$f_l = \begin{cases} C_l(1 - \sqrt{\frac{1}{u_l C_l}}) & \text{if } u_l > \frac{1}{C_l} \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

To determine $\{x_p\}$, we need to solve $|W|$ (one for each O-D pair) shortest path problems where u_l is the arc weight for link l. For each O-D pair $w \in W$, a shortest path $p_w^1 \in P_w$ is found. Then,

$$x_p = \begin{cases} \gamma_w & \text{if } p = p_w^1 \\ 0 & \text{if } p \ne p_w^1. \end{cases} \quad (25)$$

Let d_w be the total cost of p_w^1. There are two cases to consider. If $d_w \le E_w'(0)$, then $y_w = 0$; otherwise, $y_w = \hat{y}_w$ where $E_w'(\hat{y}_w) = d_w$. γ_w can be determined by (22).

For any $u \ge 0$, the optimal objective function value of (LR1), $Z_{D1}(u)$, is a lower bound on Z_{IP1} [9]. Naturally, one wants to determine the greatest lower bound by

$$Z_{D1} = \max_{u\ge 0} Z_{D1}(u). \quad (D1)$$

There are several methods for solving the dual problem (D1). The most popular method is the subgradient method [4, 16]. Let an $|L|$ vector b be a subgradient of $Z_{D1}(u)$. In iteration k of the subgradient optimization procedure, the multiplier for each link $l \in L$ is updated by $u_l^{k+1} = u_l^k + t^k b_l^k$. The step size t^k is determined by

$$t^k = \delta \frac{Z_{IP1}^h - Z_{D1}(u^k)}{\|b^k\|^2} \quad (26)$$

where Z_{IP1}^h is an objective function value for a heuristic solution (upper bound on Z_{IP1}) and δ is a constant, $0 < \delta \le 2$.

Alternatively, we can use the Multiplier Adjustment Procedure [19] to solve (D1). The multiplier updating rule is as follows.

$$u_l^{k+1} = C_l^{-1}\,[1 - \frac{(m_k - 1)f_l^k + g_l^k}{m_k C_l}]^{-2} \quad (27)$$

where g_l^k represents the aggregate flow on link l determined from $\{x_p, \gamma_w\}$. The multiplier adjustment procedure given above is better suited for distributed computation and provides better lower bounds.

3.2 A Solution Procedure for Model 2

For the second model, we dualize constraint (14).

$$Z_{D2}(u) = \min -s + \sum_{l\in L} u_l\Big(\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pl} - \alpha_l C_l\Big) \quad (LR2)$$

subject to $(\mathbf{x}, \gamma) \in \Pi$

$$s \le \gamma_w \quad \forall w \in W. \quad (28)$$

The following observations will help us to find a solution procedure for (LR2). First, for $\{x_p\}$, (25) is a solution. Second, $\gamma_w = s$, $\forall w \in W$. This is true because if there is any $\gamma_{w'} > s$, then $\gamma_{w'}$ can be reduced to s (s will not be changed while the total cost will be reduced). Note that in order to avoid the possibility of an unbounded solution we can add redundant constraints $\gamma_w \le \max_l \alpha_l C_l\ \forall w \in W$ to (LR2). Then, the objective function of (LR2) can be rewritten as $-s + s\sum_{w\in W} d_w - \sum_{l\in L} u_l\alpha_l C_l$. Therefore, if $\sum_{w\in W} d_w \ge 1$, then $\gamma_w = s = 0, \forall w \in W$; otherwise, $\gamma_w = s = \max_l \alpha_l C_l, \forall w \in W$.

Again, to find the greatest lower bound, we solve the following problem.

$$Z_{D2} = \max_{u\ge 0} Z_{D2}(u). \quad (D2)$$

To solve (D2), the subgradient method is used.

Two schemes are designed to find better lower bounds. The first one is to use the fact that when all of the multipliers are multiplied by a positive scalar κ, the shortest paths obtained in solving (LR2) do not change. However, every d_w should change by the same factor κ. Therefore, without changing the original solution s, $\{\gamma_w\}$, and $\{x_p\}$, we can easily calculate $Z_{D2}(u)$ at the point $\{\frac{u_l}{\sum_w d_w}\}$ and possibly find a higher objective value.

The second one is obtained by the fact that the set of possible objective function values of (IP2) is discrete. For the ease of illustrating the idea, assume that $\alpha_l C_l = C, \forall l \in L$. Then, $\frac{C}{-Z_{D2}}$ represents an lower bound on the number of user pairs sharing the bottleneck link. Since the number of user pairs should be an integer, the bound can be tightened by taking the ceiling function $\lceil \frac{C}{-Z_{D2}} \rceil$. Then, we can use this value to calculate a better lower bound to (IP2).

3.3 A Solution Procedure for Model 3

To solve (IP3), we dualize constraints (16) and (18).

$$Z_{D3}(u,\beta) = \min -s + \sum_{l\in L} u_l\Big(\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pl} - f_l\Big) + \beta\Big(\sum_{l\in L} \frac{f_l}{C_l - f_l} - J\Big) \quad (LR3)$$

subject to $(\mathbf{x}, \gamma) \in \Pi$

$$s \le \gamma_w \quad \forall w \in W \quad (29)$$

$$f_l \le C_l \quad \forall l \in L. \quad (30)$$

We can find the optimal β and $\{f_l\}$ analytically by solving the following simultaneous equations.

$$\beta \frac{C_l}{(C_l - f_l)^2} - u_l = 0 \quad \forall l \in L \quad (31)$$

$$\sum_{l\in L} \frac{f_l}{C_l - f_l} - J = 0. \quad (32)$$

Equation (31) is obtained by equating to zero the derivative of the objective function of (LR3) with respect to f_l. Equation (32) holds since in an optimal solution to (IP3) constraint (18) is always binding.

After simple algebra, we can derive the following results.

$$\beta = \left(\frac{\sum_{l\in L}\sqrt{u_l C_l}}{|L|+J}\right)^2 \quad (33)$$

$$f_l = C_l\left(1 - \frac{\sum_{l\in L}\sqrt{u_l C_l}}{(|L|+J)\sqrt{u_l C_l}}\right) \quad \forall l \in L. \quad (34)$$

And the objective function of (LR3) becomes

$$-s+\sum_{l\in L} u_l \sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pl} - \sum_{l\in L} u_l C_l + \frac{1}{|L|+J}\left(\sum_{l\in L}\sqrt{u_l C_l}\right)^2. \quad (35)$$

(LR3) then becomes similar to (LR2) since the last term in (35) is a constant.

3.4 Getting the Primal Solutions

Recall that for each of the three models at each iteration when solving the dual problem, we find a shortest path for each O-D pair. Thus, a heuristic for finding the routing assignment for each of the three models is simply to use these shortest paths.

To determine the traffic rate for each O-D pair, in the first model the solution to (LR1) is used directly. For the second and the third models, the algorithm proposed by Hayden [15] is adopted to perform the max-min flow control. Due to the space limitation of the paper, the algorithm is not given here. However, the interested reader can find the statement of the algorithm in [1] or [15]. Note that for the third model, the capacity of link l is given by f_l specified by equation (34). This can guarantee that constraint (18) is always satisfied.

4 Computational Results

The three routing and flow control algorithms for virtual circuit networks described in Section 3 were coded in FORTRAN 77 and run on a SUN 4/60 workstation. The algorithms were tested on three networks. Their topologies are shown in Figures 1, 2, and 3. For each of the three networks, it was assumed that for each O-D pair there were at most three candidate routes. For each O-D pair, shortest paths were found with respect to 3 sets of randomly generated arc weights and the distinct paths were used as candidate routes.

For Model 1, the throughput limitation cost function used was $\frac{a_w}{\gamma_w - y_w}$. This was suggested in [12]. It was assumed that $a_w = a, \forall w \in W$. To solve (D1) the multiplier adjustment procedure given by equation (27) was used. In the multiplier adjustment procedure, we chose $m_k = (\log_2(k+3))^2$. The choice of the initial values of the multipliers was $\{\frac{1}{C_l}\}$. The paths and traffic rates found by solving (LR1) were used in the heuristic solution to (IP1).

Table 1 summarizes the results of our computational experiments with Algorithm 1. In the second column, the value of a_w for each O-D pair w is given. The third column specifies the capacity of each link in each network. The fourth column is the largest lower bound on the optimal objective function value in 300 iterations. Recall that this is the best objective function value of the dual problem. The fifth column gives the best objective function value for (IP1) in 300 iterations. The error bound [(upper-bound − lower-bound)×100/lower-bound] is an upper bound on how far the best feasible solution found is from an optimal solution. The seventh column provides the CPU times which include the time to input the problem parameters.

From an inspection of Table 1, it is clear that the joint routing and flow control algorithm is efficient and very effective in finding near-optimal solutions. For every test problem (networks with up to 61 nodes), the algorithm determines a solution that is within 1% of an optimal solution in less than 2 minutes of CPU time on a SUN 4/60 workstation.

For Model 2, α_l was assumed to be 1 for all links. To solve (D2), the subgradient method was applied. In our implementation, Z^h_{IP2} was initially chosen as 0 and updated to the best upper bound found so far from iteration to iteration. In (26), δ was initially set to 2 and halved whenever the objective function value did not improve in 20 iterations. The choice of the initial values of the multipliers was 0. Both schemes mentioned in Subsection 3.2 to improve the lower bounds were implemented. A redundant constraint that limited each γ_w to be no greater than 50 times the maximum link capacity (we found that this constraint was more effective than $\max_l \alpha_l C_l$) was added. The results are reported in Table 2. The error bounds reported in Tables 2 and 3 were computed by [(lower-bound − upper-bound)×100/upper-bound].

In the experiments with Algorithm 2, we found that it was unnecessary to vary the level of link capacities for problem 2. This observation also applies to problem 3. This result can be proven to be generally true by modifying the proof of a similar result in [23].

Algorithm 3 was applied to the network OCT as J was varied. To solve (D3), we used the same implementation of the subgradient method used to solve (D2). To find a primal solution, Hayden's algorithm [15] was applied where the capacity for link l was determined by equation (34). The computational experience with Algorithm 3 is reported in Table 3.

5 Summary and Conclusions

In this paper the joint routing and flow control problem for virtual circuit networks is considered. The problem is formulated as three different mixed integer programming problems. The Lagrangean Relaxation technique is applied to solve the three optimization problems.

In computational experiments our algorithms are shown to be very effective and efficient in finding near-optimal solutions. For the worst case, our algorithm determines solutions that are within a few percent of an optimal solution in minutes of CPU time for networks with 26 to 61 nodes.

Since a distributed algorithm is more desirable than a centralized one due to higher reliability, a continuation of this work is to develop a distributed version of the routing and flow control algorithm for each of the three models presented in this paper. The formulations can be extended to include priorities and multiple sessions for each O-D pair.

References

[1] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Inc., 1987.

[2] D.G. Cantor and M. Gerla. Optimal routing in a packet switched computer network. *IEEE Transactions on Computers*, C-23:1062–1069, 1974.

[3] A. Danet, R. Despres, A.L. Rest, G. Pichon, and S. Ritzenthaler. The French public packet switching service: The TRANSPAC network. In *Proceeding Third International Computer Communication Conference*, pages 251–260, 1976.

[4] M.L. Fisher. The Lagrangian Relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, January 1981.

[5] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 3:97–133, 1973.

[6] R.G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, COM-25(1):73–85, January 1977.

[7] R.G. Gallager and S.J. Golestaani. Flow control and routing algorithms for data networks. In *Proc. 5th Intern. Conf. Commun.*, pages 779–784, 1980.

[8] B. Gavish and S.L. Hantler. An algorithm for optimal route selection in SNA networks. *IEEE Transactions on Communications*, COM-31(10):1154–1160, October 1983.

[9] A.M. Geoffrion. Lagrangean relaxation and its uses in integer programming. *Math. Programming Study*, 2:82–114, 1974.

[10] M. Gerla. Routing and flow control in ISDN's. In *Proc. 1986 ICCC*, pages 643–647, 1986.

[11] M. Gerla and L. Kleinrock. Flow control: A comparative survey. *IEEE Transactions on Communications*, COM-28:553–574, 1980.

[12] S.J. Golestaani. *A unified theory of flow control and routing on data communication networks.* PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, Cambridge, MA, 1980.

[13] J.P. Gray and T.B. McNeill. SNA multiple-system networking. *IBM System Journal*, 18:263–297, 1979.

[14] GTE Telenet Communications Corp., Vienna, VA. *Functional Description of GTE Telenet Packet Switching Networks*, May 1982.

[15] H. Hayden. Voice flow control in integrated packet networks. Technical report, MIT Laboratory for Information and Decision Systems, Cambridge, MA, 1981. Report LIDS-TH-1152.

[16] M. Held, P. Wolfe, and H.D. Crowder. Validation of subgradient optimization. *Math. Programming*, 6:62–88, 1974.

[17] J.M. Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, COM-29:954–962, 1981.

[18] L. Kleinrock. *Queueing Systems*, volume 1 and 2. New York: Wiley-Interscience, 1975 and 1976.

[19] Y.S. Lin and J.R. Yee. A distributed routing algorithm for virtual circuit data networks. In *Proc. 1989 INFOCOM*, pages 200–207, 1989.

[20] H. Rudin and H. Mueller. Dynamic routing and flow control. *IEEE Transactions on Communications*, COM-28(7):1030–1039, July 1980.

[21] G.H. Thaker and J.B. Cain. Interactions between routing and flow control algorithms. *IEEE Transactions on Communications*, COM-34(3):269–277, March 1986.

[22] L.R. Tymes. Routing and flow control in TYMNET. *IEEE Transactions on Communications*, COM-29:392–398, 1981.

[23] J.R. Yee and Y.S. Lin. A routing algorithm for virtual circuit data networks with multiple sessions. Technical report, Communication Science Insitute, USC, Los Angeles, CA, August 1989. CSI-89-09-06. Submitted to *Networks*.

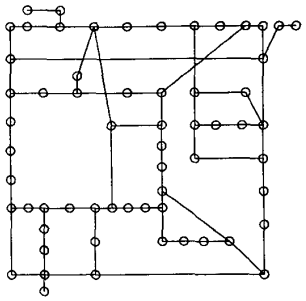

Fig. 1. 61-node 148-link ARPA net

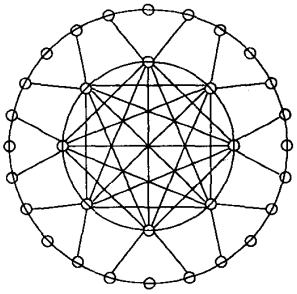

Fig. 2. 32-node 120-link RING net

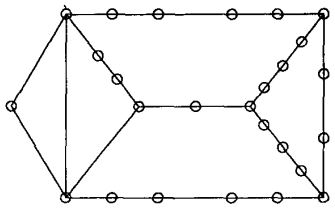

Fig. 3. 26-node 60-link OCT net

Table 1: Summary of computational results for Model 1

Net ID	a_w	Link Capacities	Lower Bounds	Upper Bounds	Error Bounds (%)	CPU Time (sec)	No. of Iter.
ARPA	0.033	100	247.572	247.579	0.003	77.6	300
"	0.100	300	1171.67	1171.76	0.007	73.2	300
"	0.167	500	2697.07	2697.73	0.024	74.5	300
"	0.333	1000	9184.43	9188.13	0.040	75.5	300
RING	0.167	60	251.811	251.870	0.023	29.3	300
"	0.333	30	635.553	637.338	0.284	36.3	300
"	0.500	20	1164.76	1173.50	0.751	36.9	300
"	0.667	15	1839.14	1853.60	0.786	38.4	300
OCT	0.100	100	114.580	114.635	0.049	20.4	300
"	0.200	50	288.618	288.763	0.050	20.7	300
"	0.500	20	1191.52	1192.72	0.101	21.6	300
"	1.000	10	3974.20	3982.81	0.217	20.4	300

Table 2: Summary of computational results for Model 2

Net ID	Lower Bounds	Upper Bounds	Error Bounds (%)	CPU Time (sec)	No. of Iter.
ARPA	-20.0803	-18.9394	6.024	334.1	1200
RING	-156.250	-151.515	3.125	133.4	2000
OCT	-84.7458	-83.3333	1.695	81.1	2000

Table 3: Summary of computational results for Model 3

J	Link Capacities	Lower Bounds	Upper Bounds	Error Bounds (%)	CPU Time (sec)	No. of Iter.
15.00	100	-0.42961	-0.42269	1.637	121.7	2000
25.71	100	-0.63614	-0.62204	2.266	121.7	2000
40.00	100	-0.83683	-0.80478	3.983	123.8	2000
60.00	100	-1.03039	-1.00771	2.251	120.5	2000
90.00	100	-1.21421	-1.17865	3.017	123.7	2000
140.0	100	-1.38420	-1.33143	3.962	123.9	2000
240.0	100	-1.53168	-1.47478	3.870	122.9	2000
540.0	100	-1.64358	-1.58369	3.782	124.1	2000
1140	100	-1.67864	-1.61426	3.988	123.1	2000