Minimax End-to-end Delay Routing and Capacity Assignment for Virtual Circuit Networks

Kwang-Ting Cheng
Dept. of Electrical & Computer Engineering
University of California
Santa Barbara, CA 93106

Frank Yeong-Sung Lin
Dept. of Electronic Engineering
National Taiwan Institute of Technology
Taipei, Taiwan, R.O.C.

Abstract — In this paper, we consider the routing and capacity assignment problem for virtual circuit networks where the objective is to minimize the maximum end-to-end packet delay. Compared with traditional aggregate type of performance measures, such as the average packet delay, this performance measure is consistent with those of many new services and achieves better fairness among users.

Two cases are considered. In the first case, the capacity assignment is assumed to be given and the routing strategy is to be determined. The problem is formulated as a nonlinear nonconvex multicommodity network flow problem with integer routing decision variables. A dual approach is proposed to calculate primal feasible solutions. The proposed algorithm is computationally shown to uniformly outperform a greedy heuristic and a linear programming relaxation approach in conjunction with a rounding scheme. The algorithm also calculates legitimate lower bounds on the optimal objective function value, which are not easily attainable using the linear programming relaxation approach due to the nonconvex nature of the problem. When the network is heavily loaded, these lower bounds are significantly higher than those obtained by solving the routing problem of minimizing the average end-to-end delay with continuous routing decision variables. We also propose another approach to calculating lower bounds by using a lower-bounding function to characterize the delay on each delay element. These bounds are shown to be tighter than those provided by the dual approach when the network is lightly loaded.

In the second case, we consider the joint routing and capacity assignment problem. The problem is formulated as a nonlinear nonconvex mixed integer programming problem. We develop a two-phase algorithm where the routing and the capacity assignment decision variables are optimized, respectively. A minimum-hop heuristic is used to calculate the routing assignment, and then a convex programming procedure is devised to solve the capacity assignment problem. Experimental results of this heuristic and comparison with the results of the first case will be presented.

1. Introduction

To ensure reliable and high-quality network services, routing and capacity assignment policies should be carefully designed. Traditional quasi-static routing algorithms attempt to optimize a certain aggregate measure, e.g. to minimize the average end-to-end packet delay [1, 2]. However, this kind of performance measures may not be consistent with the service objectives and may result in fairness problems.

Since end-to-end performance is users' direct perception about the service quality, service objectives are typically specified on an end-to-end basis for many new services, e.g. Switched Multi-megabit Data Service (SMDS), Frame Relay Service (FRS), Asynchronous Transfer Mode (ATM) and Advanced Intelligent Network (AIN). Typical service performance measures for advanced services include end-to-end packet/cell delay and end-to-end packet/cell loss probability. As such, from service providers' perspective, it is more appropriate to design a routing and capacity assignment policy such that the end-to-end quality of service for each user is satisfied than a policy to optimize an aggregate performance measure, which in many cases may result in good average performance but unacceptable performance for some users (fairness issues).

In this paper, we focus on routing and capacity assignment algorithms for virtual circuit networks. In virtual circuit networks, traffic of each session is transmitted over exactly one path. This discrete nature makes the problem more difficult than the datagram routing and capacity assignment problem where the routing decision variables are continuous. Previous research on virtual circuit routing mostly considers the objective function of minimizing the average end-to-end packet delay [3, 1, 4]. In this paper, for the first time, we consider the problem of minimizing the maximum end-to-end packet delay for virtual circuit networks. We consider two cases. In the first case, the capacity assignment is considered to be fixed. The remaining routing problem is formulated as a nonlinear nonconvex multicommodity network flow problem with integer routing decision variables. The nonconvex property together with the discrete nature of the problem makes it challenging in developing efficient and effective algorithms.

A novel application of the Lagrangean relaxation technique is applied to calculate primal feasible solutions and lower bounds on the optimal objective function value. For comparison purposes, we also develop two other primal heuristics -- one based upon a greedy principle and the other based upon linear programming relaxation in conjunction with a rounding scheme. In computational experiments, the proposed algorithm is shown to outperform these two primal heuristics over a full range of loads in a test network. We also computationally show that the lower bounds calculated by the proposed dual approach are significantly higher than those obtained by solving the routing problem of minimizing the average end-to-end delay with continuous routing decision variables when the network is heavily loaded. We also propose another approach to calculating lower bounds by using a lower bounding function to characterize the delay on each network element. These bounds are shown to be tighter than those provided by the dual approach when the network is lightly loaded.

We next consider the joint minimax routing and capacity assignment problem for virtual circuit networks. The problem is formulated as a mixed integer programming problem where routing and capacity assignment are both decision variables. A two-phase algorithm for the joint problem is proposed. In the first phase, only the routing decision variables are considered. A minimum-hop routing heuristic is adopted for this part. Once the routing is determined, the remaining capacity assignment problem becomes a convex programming problem and can be solved optimally. In the computational experiments, it is demonstrated that for a fixed total capacity, to consider the joint routing and capacity assignment problem would result in significant performance improvement than to consider the routing problem with an even capacity assignment policy.

The remainder of this paper is organized as follows. In section 2, a mathematical formulation of the routing problem is presented. In section 3, a dual approach for the routing problem based upon Lagrangean relaxation is proposed. In section 4, two primal heuristics are proposed for comparison purposes. In section 5, the joint routing and capacity assignment problem is formulated and a two-phase algorithm is devised. Computational results are reported in section 6.

2. Routing 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. Let W be the set of origin-destination (O-D) pairs (commodities) 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, the overall 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 link $l \in L$, the capacity is C_l packets/sec.

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 in a session are transmitted over exactly 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 over link l, denoted as g_l, is $\sum_{p \in P_w} \sum_{w \in W} x_p \gamma_w \delta_{pl}$.

In the network, there is a buffer for each outbound link. Using Kleinrock's independence assumption [5], 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, as considered in [3, 1, 4].

The problem of determining a path for each O-D pair to minimize the maximum end-to-end delay in a virtual circuit network is formulated as the following nonconvex nonlinear combinatorial optimization problem.

$$Z_{IP'} = \min \max_{w \in W} \sum_{l \in L} \sum_{p \in P_w} \frac{x_p \delta_{pl}}{C_l - g_l} \qquad \text{(IP')}$$

subject to:

$$g_l = \sum_{p \in P_w} \sum_{w \in W} x_p \gamma_w \delta_{pl} \le C_l \qquad \forall\, l \in L \qquad (1.1)$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall\, w \in W \qquad (1.2)$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w,\ w \in W. \ (1.3)$$

Constraint (1.1) requires that the aggregate flow not exceed the capacity for each link. Constraints (1.2) and (1.3) require that all of the traffic for each O-D pair be transmitted over exactly one path. From the above formulation, it is clear that (IP') is a nonlinear multicommodity integral flow problem. In addition, by examining the Hessian of $\sum_{l \in L} \sum_{p \in P_w} \frac{x_p \delta_{pl}}{C_l - g_l}$ with respect to $\{x_p\}$, it can be shown that (IP') is a nonconvex programming problem.

An equivalent formulation of the above problem is given by (IP) below, which is more suitable for the application of the Lagrangean Relaxation method. We introduce two variables: y_{wl} for each O-D pair w and each link l and f_l for each link l. y_{wl} is defined as $\sum_{p \in P_w} x_p \delta_{pl}$ and f_l can be interpreted as *estimate* of the aggregate link flow. As will be shown in the next section, the introduction of these auxiliary variables facilitates the decomposition of the problem into independent and easily solvable subproblems in the Lagrangean relaxation. Extra constraints associated with these variables (2.4 - 2.7) are added.

$$Z_{IP} = \min S \qquad \text{(IP)}$$

subject to:

$$\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} \le S \qquad \forall\, l \in L \qquad (2.1)$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall\, w \in W \qquad (2.2)$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w,\ w \in W \qquad (2.3)$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le y_{wl} \qquad \forall\, w \in W,\ l \in L \qquad (2.4)$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall\, w \in W,\ l \in L \qquad (2.5)$$

$$g_l \le f_l \qquad \forall\, l \in L \qquad (2.6)$$

$$0 \le f_l \le C_l \qquad \forall\, l \in L. \qquad (2.7)$$

Constraints (2.4) and (2.6) should be equalities and the current form is a relaxation. However, it can be shown that at an optimal solution, equality should hold. As will be seen in the next section, this relaxation makes the formulation better suited for the application of Lagrangean relaxation.

3. Dual Algorithm for the Routing Problem

The basic approach to the algorithm development is Lagrangean relaxation. We dualize constraints (2.1), (2.4) and (2.6) to obtain the following relaxation. (Note that the constraints are dualized in such a way that the corresponding multipliers are nonnegative.)

$$Z_D(t, v, u) = \min \Big[S + \sum_{w \in W} t_w \Big(\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} - S \Big) + \sum_{w \in W} \sum_{l \in L} v_{wl} \Big(\sum_{p \in P_w} x_p \delta_{pl} - y_{wl} \Big) + \sum_{l \in L} u_l (g_l - f_l) \Big] \qquad \text{(LR)}$$

subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall\, w \in W \qquad (3.1)$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w,\ w \in W \qquad (3.2)$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall\, w \in W,\ l \in L \qquad (3.3)$$

$$0 \le f_l \le C_l \qquad \forall\, l \in L. \qquad (3.4)$$

We then add a redundant constraint:

$$\underline{S} \le S \le \overline{S} \qquad (3.5)$$

where $\overline{S}$ is the maximum allowable end-to-end delay and $\underline{S}$ is the minimum possible end-to-end delay. A possible value of $\underline{S}$ is $\frac{D}{\max_l C_l}$ where D is the diameter of the network when all O-D pairs have nonzero demand.

We can solve problem (LR) by solving three independent subproblems:

$$\min S(1 - \sum_{w \in W} t_w) \quad \text{(SUB1)}$$

subject to (3.5),

$$\min \sum_{w \in W} \sum_{p \in P_w} \sum_{l \in L} (v_{wl} + u_l \gamma_w) \, x_p \delta_{pl} \quad \text{(SUB2)}$$

subject to (3.1) and (3.2), and

$$\min \sum_{l \in L} [\frac{\sum_{w \in W} t_w y_{wl}}{C_l - f_l} - \sum_{w \in W} v_{wl} \, y_{wl} - u_l f_l] \quad \text{(SUB3)}$$

subject to (3.3) and (3.4).

The solution to (SUB1) is straightforward: if $\sum_{w \in W} t_w \geq 1$, $S = \bar{S}$; otherwise, $S = \underline{S}$. Problem (SUB2) can be further decomposed into $|W|$ independent shortest path problems with nonnegative arc weights and can be easily solved. Problem (SUB3), though looks complicated due to the coupling of $\{y_{wl}\}$ and $\{f_l\}$, can be solved analytically. We first further decompose (SUB3) into $|L|$ independent problems: For each link $l \in L$:

$$\min[\frac{\sum_{w \in W} t_w y_{wl}}{C_l - f_l} - \sum_{w \in W} v_{wl} \, y_{wl} - u_l f_l] \quad \text{(SUB3\_l)}$$

subject to: $0 \leq f_l \leq C_l$ and $y_{wl} = 0 \; or \; 1 \; \forall \; w \in W$.

For different values of f_l, the value of y_{wl} for minimum objective function, denoted as $y_{wl}^*(f_l)$, may be different. As an example, consider the case that $f_l = 0$. The objective function is minimized by assigning $y_{wl}^*(0)$ to 1 if $(\frac{t_w}{C_l} - v_{wl}) \leq 0$ and to 0 otherwise. We define a set of *break points* of f_l as those points where $(\frac{t_w}{C_l - f_l} - v_{wl}) = 0$ for each w. These break points are sorted and denoted as $f_l^1, f_l^2, \ldots, f_l^n$. Note that there are at most $|W|$ break points. *We observe that when* $f_l^i \leq f_l < f_l^{i+1}$, *the value of* $y_{wl}^*(f_l)$ *remains constant for all* $w \in W$. Within the above interval, $y_{wl}^*(f_l)$ is 1 if $(\frac{t_w}{C_l - f_l^i} - v_{wl}) \leq 0$ and is 0 otherwise. Therefore, within an interval, $[f_l^i, f_l^{i+1})$, the objective is only a function of f_l and the minimum point within the interval can be found analytically. By examining at most $|W|$+1 intervals, we can then find the global minimum point by comparing those local minimum points.

When examining an interval, we first determine $y_{wl}^*(f_l^i)$ within the interval for each w. We denote $\sum_{w \in W} t_w y_{wl}^*(f_l^i)$ as a_l and $\sum_{w \in W} v_{wl} y_{wl}^*(f_l^i)$ as b_l. Note that a_l and b_l are non-negative. Within the interval, the objective function can then be expressed as: $Z_{sub3\_l} = \frac{a_l}{C_l - f_l} - b_l - u_l \, f_l$. A typical curve of the objective function vs. f_l within the interval $f_l^i \leq f_l \leq f_l^{i+1}$ is shown in Fig. 1. The local minimum point is either at the boundary point, f_l^i or f_l^{i+1}, or at point $f_l^* = C_l \sqrt{\frac{a_l}{u_l}}$, $(u_l \neq 0)$.

For any $(t, v, u) \geq 0$, by using the weak Lagrangean duality theorem, the optimal objective function value of (LR), $Z_D(t, v, u)$, is a lower bound on Z_{IP}. We want to determine the greatest lower bound by:

$$Z_D = \max_{t,v,u \geq 0} Z_D(t, v, u) \quad \text{(D)}$$

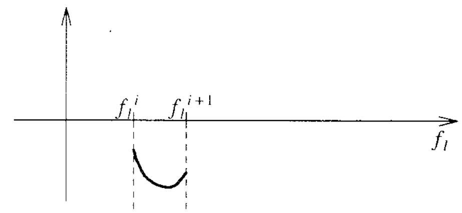

Fig. 1: A typical curve of the objective function of (SUB3_l) vs. f_l within the interval $f_l^i \leq f_l \leq f_l^{i+1}$

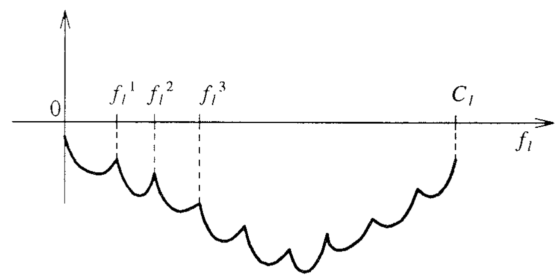

Fig. 2: A typical graph of the objective function of (SUB3_l) vs. f_l

We use a popular method, the subgradient method, for solving the dual problem (D). Let a $(|W| + |L| \cdot |W| + |L|)$ vector b be a subgradient of $Z_D(t,v,u)$. In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (t_w^k, v_{wl}^k, u_l^k)$ is updated by

$$m^{k+1} = m^k + \alpha^k b^k.$$

The step size α^k is determined by

$$\alpha^k = \delta \frac{Z_{IP}^h - Z_D(m^k)}{\|b^k\|^2}$$

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

It is observed in the computational experiments that the lower bounds calculated by the proposed dual approach is not tight when the network load is light. To calculate tighter lower bounds for lightly loaded networks, the following lower-bounding procedure is devised. One O-D pair $w \in W$ is considered at a time where the traffic from the other O-D pairs is ignored. Denote such a minimax routing problem for w by (LBw). The optimal objective function value for each of such problems is clearly a lower bound on the optimal objective function value of (IP), Z_{IP}. An optimal solution to (LBw) is to route γ_w over a shortest path where the arc weight for link l is $1/(C_l - \gamma_w)$. Solve each (LBw) and report the highest one among those lower bounds obtained.

To calculate primal feasible solutions, we principally use the solution to each (LR). If the routing assignment calculated in solving an (LR) satisfies the capacity constraints, then a primal feasible solution is found. The best primal feasible solution is then reported.

4. Primal Heuristics for the Routing Problem

In this section, we propose two other primal heuristics for comparison purposes.

Primal Heuristic 1: The first heuristic is based upon linear-programming relaxation in conjunction with a rounding scheme. More precisely, the integrality constraints are first relaxed. Then the penalty function method and a Frank-Wolfe-like method are used to solved the relaxed problem. A simple routing scheme is next applied to find integer solutions.

Primal Heuristic 2: The second primal algorithm is a greedy heuristic. The O-D pairs are first sorted with respect to an attribute. Then at each iteration of the algorithm, minimize the maximum end-to-end delay of all the O-D pairs considered so far by including one more O-D pair from the ordered list. This process is repeated until all O-D pairs are considered.

5. The Joint Routing and Capacity Assignment Problem

In this section, the joint routing and capacity assignment problem for virtual circuit networks is considered. A solution procedure is then proposed.

Consider the following formulation

$$Z_{LB} = \min \max_{w \in W} \sum_{l \in L} \sum_{p \in P_w} \frac{x_p \delta_{pl}}{c_l - g_l} \qquad \text{(R\&CA)}$$

subject to:

$$g_l = \sum_{p \in P_w} \sum_{w \in W} x_p \gamma_w \delta_{pl} \le c_l \qquad \forall\ l \in L \qquad (4.1)$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall\ w \in W \qquad (4.2)$$

$$x_p \le 0 \ or\ 1 \qquad \forall\ p \in P_w,\ w \in W \qquad (4.3)$$

$$\sum_{l \in L} c_l = \sum_{l \in L} C_l \qquad (4.4)$$

where c_l is the capacity assignment variable for link l and $\sum_{l \in L} C_l$ is the total capacity of the network. (We use the same notation as in the previous sections.)

Compared with (IP), (R&CA) has more variables and similar discrete and nonconvex properties. It is also observed that (R&CA) with the routing decision variables fixed is a convex programming problem. We then propose a two-phase algorithm to solve (R&CA) where in the first phase a minimum-hop heuristic routing is used and in the second phase the remaining capacity assignment problem is optimally solved by applying convex programming techniques.

Algorithm R&CA

Step 1: For every node $n \in V$, calculate a shortest path spanning tree using Dijkstra's algorithm where the arc weight associated with each link is set to 1.

Step 2: Route the traffic demand for each O-D pair over the path calculated in Step 1.

Step 3: Calculate the aggregate flow for each link l, denoted by h_l.

Step 4: Solve the following convex programming problem.

$$Z_{LB} = \min S$$

subject to:

$$\sum_{l \in L} \sum_{p \in P_w} \frac{x_p \delta_{pl}}{c_l - h_l} \le S \qquad \forall\ w \in W \qquad (5.1)$$

$$h_l = \sum_{p \in P_w} \sum_{w \in W} x_p \gamma_w \delta_{pl} \le c_l \qquad \forall\ l \in L \qquad (5.2)$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall\ w \in W \qquad (5.3)$$

$$\sum_{l \in L} c_l = \sum_{l \in L} C_l \qquad (5.4)$$

where x_p's are fixed (determined in Step 1) and only c_l's are considered to be decision variables.

The algorithm routes the traffic for each O-D pair over a minimum-hop path. In addition, for each root the union of selected paths to the other nodes in the networks form a spanning tree. Dijkstra's algorithm is used to calculate a shortest path spanning tree for each root. Given the routing calculated in Steps 1 & 2, an optimal capacity assignment policy to achieve the minimum longest end-to-end delay is then found in Step 4.

6. Computational Experiments

The minimax end-to-end delay routing algorithm described in section 3 and the joint algorithm described in section 5 were coded in C and run on a workstation. The network topology used in the experiments [1, 3, 4] is shown in Figure 3. The maximum number of iterations allowed for the proposed dual routing algorithm is 1000. It is assumed that the traffic demand of each O-D pair in the network is one packet per second. For each O-D pair, at most three candidate paths are considered. It is reported in [1] and [4] that the optimal objective function value will not be significantly improved when the number of candidate paths for each O-D pair is greater than 3. We use randomly generated arc weights and Dijkstra's shortest path algorithm to generate the candidate paths.

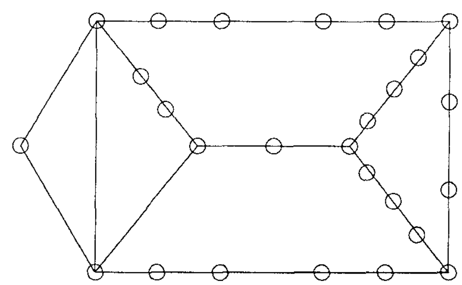

Fig. 3: 26-node 60-link OCT net

Table 1 - Comparison of upper bounds obtained by various algorithms				
Link capacity	Upper bounds (msec.) dual	greedy	F-W	Best known lower bound*
65	860.3	∞	∞	569.8
70	514.6	∞	∞	311.5
100	168.4	188.6	183.5	86.9
150	81.7	85.1	86.0	53.7
200	54.1	55.8	55.8	40.2

*: the best lower bounds obtained by the three approaches proposed in this paper

Three sets of computational experiments are performed. In the first set of experiments, we compare the proposed routing algorithm with the two primal heuristics described in section 4. The attribute used in the greedy heuristic to sort the O-D pairs is a uniformly distributed random variable with a range between 0 and 1. Table 1 summarizes the results. The first column is the link capacity. The second column reports the upper bounds generated by the proposed dual routing algorithm. The third column shows the upper bounds generated by the

Table 2 - Comparison of lower bounds obtained by various algorithms				
Link capacity	Lower bounds (msec.) dual	min_avg†	one-at-a-time§	Best known upper bound*
65	569.8	384.9	125.0	860.3
70	311.5	248.3	115.9	514.6
100	86.9	86.7	80.8	168.4
150	42.4	42.5	53.7	81.7
200	28.1	28.2	40.2	54.1

†: lower bounds generated by solving the problem of minimizing the average packet delay
§: lower bounds generated by considering one O-D pair at a time where the traffic from the other O-D pairs is ignored
*: the lowest values of columns 2, 3 and 4 in Table 1

greedy heuristic, where infinity indicates that the heuristic fails to find a feasible solution. The fourth column is the upper bounds generated by the other primal heuristic based upon linear programming relaxation, the penalty function method, a Frank-Wolfe-like approach and a simple rounding scheme. The last column presents the best known lower bounds, which are obtained from Table 2. From an inspection of Table 1, it is shown that the proposed dual algorithm consistently calculates the best results compared with the two primal heuristics. Another observation from Table 1 is that the two primal heuristics fail to calculate primal feasible solutions when the network is heavily loaded.

In the second set of experiments, we compare the lower bounds generated by various methods. Table 2 summarizes the results. The first column is the link capacity. The second column reports the lower bounds generated by the proposed dual algorithm. The third column shows the lower bounds generated by solving the virtual circuit routing problem where the objective function is to minimize the average packet delay. It is clear that the optimal objective function value of this minimizing-average-delay virtual circuit routing problem is a lower bound on the optimal objective function value of the minimizing-maximum-delay virtual circuit routing problem. The algorithm proposed in [4] is applied and the lower bounds calculated are reported. The fourth column is the lower bounds generated by considering one O-D pair at a time where the traffic from the other O-D pairs is ignored as described in section 3. The last column presents the best known upper bounds, which are obtained from Table 1. From an inspection of Table 2, it is shown that the proposed dual routing algorithm calculates tighter lower bounds than the minimizing-average-delay approach when the network is heavily loaded. Another observation from Table 2 is that the lower bounds obtained by the "one-at-a-time" lower-bounding procedure is tighter than the other two approaches when the network is lightly loaded.

In the third set of experiments, Algorithm (R&CA) described in section 5 is tested. The purpose is to show the efficiency of the algorithm and the effect of considering the joint problem rather than considering the routing problem with a fixed even capacity allocation policy. Table 3 summarizes the results. The first column shows the total link capacity. The second column reports the results obtained by applying Algorithm (R&CA). The third and fourth columns report the best known lower and upper bounds of the worst end-to-end delay assuming even link capacity assignment. Consider the first case in Table 3. Assuming even link capacity assignment (65 for each link), the optimal objective function value is between 569.8 and 860.3. The result corresponding to the joint routing and capacity assignment problem by applying Algorithm (R&CA) is 373.0. Thus the performance improvement is between 35% and 57%. Note that when the network is lightly loaded, it is intuitive that the performance improvement achieved by considering the joint problem should be limited.

Table 3 - Results of Algorithm (R&CA) for the joint problem			
Total link capacity	worst end-to-end delay (msec.) solving (R&CA)	assuming even capacity assignment lower bound	upper bound
3900	373.0	569.8	860.3
4200	296.8	311.5	514.6

7. Summary

In this paper, we, for the first time, consider the problem of minimizing the maximum end-to-end delay for virtual circuit networks. Compared with traditional aggregate types of performance measures, e.g. average packet delay, this performance measure is consistent with the service objectives of many new services and achieves better fairness among users.

We first consider the routing problem assuming a fixed capacity assignment. We formulate the problem as a nonconvex nonlinear multicommodity integral flow problem. The nonconvex and discrete property makes the problem very difficult. We take an optimization-based approach by applying the Lagrangean relaxation technique in the algorithm development. Compared with two other primal heuristics, the proposed algorithm achieves better performance over different loads in the test network.

We further consider the joint routing and capacity assignment problem where both the link capacities and routing assignments are considered to be decision variables. Although this generalized problem is also a nonconvex nonlinear combinatorial optimization problem, we develop an efficient and effective two-phase heuristic. In the first phase, the routing assignments are determined by applying the Dijkstra's algorithm. In the second phase, a convex programming problem is solved to determine an optimal link capacity allocation. It is observed from the computational experiments that when the traffic requirement distribution is uniform and the total capacity is given, then by considering the routing and the capacity assignment decision variables jointly rather than using the even capacity assignment strategy and considering the routing decision variables alone, up to 57% improvement on the maximum end-to-end delay is achieved.

REFERENCES

1. B. Gavish and S.L. Hantler, "An Algorithm for Optimal Route Selection in SNA Networks," *IEEE Trans. on Communications* **COM-31**, pp. 1154-1160 (October 1983).
2. L. Fratta and M. Gerla and L. Kleinrock, "The flow deviation method: An approach to tore-and-forward communication network design," *Networks* **3**, pp. 97-133 (1973).
3. P.J. Courtois and P. Semal, "An Algorithm for the Optimization of Nonbifurcated Flows in Computer Communication Networks," *Performance Evaluation* **1**, pp. 139-152 (1981).
4. F. Y.S. Lin and J.R. Yee , "A New Multiplier Adjustment Procedure for the Distributed Computation of Routing Assignments in Virtual Circuit Data Networks," *ORSA Journal on Computing, Summer Issue* (1992).
5. L. Kleinrock, "Queueing Systems," *Wiley-Interscience*, New York, **Volumes 1 and 2** (1975-76).