PAPER

Delay Constrained Routing and Link Capacity Assignment in Virtual Circuit Networks

Hong-Hsu YEN[†a)], ***Member and*** **FrankYeong-Sung LIN**[††b)], ***Nonmember***

SUMMARY An essential issue in designing, operating and managing a modern network is to assure end-to-end QoS from users perspective, and in the meantime to optimize a certain average performance objective from the systems perspective. So in the first part of this paper, we address the above issue by using the rerouting approach, where the objective is to minimize the average cross-network packet delay in virtual circuit networks with the consideration of an end-to-end delay constraint (DCR) for each O-D pair. The problem is formulated as a multicommodity network flow problem with integer routing decision variables, where additional end-to-end delay constraints are considered. As the traffic demands increases over time, the rerouting approach may not be applicable, which results in the necessity of capacity augmentation. Henceforth, the second part of this paper is to jointly consider the link capacity assignment and the routing problem (JCR) at the same time where the objective is to minimize the total link installation cost with considering the average and end-to-end delay constraints. Unlike previous research tackling this problem with a two-phase approach, we propose an integrated approach to considering the routing and capacity assignment at the same time. The difficulties of DCR and JCR result from the integrality nature and particularly the nonconvexity property associated with the end-to-end delay constraints. We propose novel Lagrangean relaxation based algorithms to solve the DCR and the JCR problems. Through computational experiments, we show that the proposed algorithms calculate near-optimal solutions for the DCR problem and outperform previous two-phase approach for the JCR problem under all tested cases.
key words: *link capacity assignment, end-to-end QoS, routing assignment, optimization, Lagrangean relaxation*

1. Introduction

1.1 Motivation

Delay constrained QoS routing has been an active research field since the beginning of the Internet. And as more multimedia applications (e.g. VoIP) emerging on Internet, the interest of delay constrained QoS routing has been rising for both academic researcher and business practitioner.

To ensure user-perceived end-to-end QoS requirement is one of the most important issues in providing modern network services, which typically requires sophisticated design of routing and capacity management policies. User-perceived end-to-end QoS measures include, for example, mean packet delay, packet delay jitter and packet lost probability. Besides users' perspective of QoS, from the service providers' perspective (which is a traditional view of network performance management), optimizing a certain system-level performance measure, e.g. overall network utilization or average cross-network delay among all users, is another major concern. Unfortunately, these two perspectives/objectives may not be entirely agreeable with each other. This then places a major challenge to network managers and therefore calls for an integrated methodology to consider these two perspectives in a joint fashion. In this paper, we first propose the optimization models based on routing assignment to address the DCR problem, where the objective is to minimize the average cross network design to ensure the end-to-end delay requirement for each O-D pair.

Most of existing networks face a common challenge, to satisfy more and more new traffic demand and QoS requirements. However, the existing link capacity may not be sufficient. As a result, the network operators have to make a link capacity augmentation plan in order to satisfy the traffic requirements. As a result, in the second part of this paper, the link capacity is also a decision variable. Hence, JCR jointly considers link capacity assignment and routing problem at the same time.

1.2 Related Work

Routing problem could be classified into two main domains in two different networks, namely datagram service network and virtual circuit network. In datagram service network, the traffic for O-D pair could be decomposed into different flows in different paths. However, in virtual circuit network, the traffic for O-D pair should follow the same path. From the mathematical structure, the routing problem in virtual circuit network is more complicated than in datagram service network due to the integer constraints associated with the routing decision variables.

There is a abundant research focusing on datagram service networks because of the Internet Protocol (IP). Previous research [3] focused on optimal routing based on complete status information, while latter research [12] focused on near optimal routing based on local information. On the other hand, routing problem in virtual circuit networks is more difficult because of its integrality nature, and it has attracted even more attention since the emergence of the ATM and MPLS technology.

Best-effort Internet routing protocol (OSPF) fails to address QoS (delay, bandwidth) requirements for the increasing demand of multimedia applications. This calls for the

Manuscript received January 9, 2004.
Manuscript revised July 13, 2004.
[†]The author is with the Department of Information Management, Shih Hsin University, Taiwan.
[††]The author is with the Department of Information Management, National Taiwan University, Taiwan.
a) E-mail: hhyen@cc.shu.edu.tw
b) E-mail: yslin@im.ntu.edu.tw
DOI: 10.1093/ietcom/e88–b.5.2004

QoS routing algorithms. QoS performance measures for a path can either be additive (e.g. delay) and non-additive (e.g. bandwidth). Additive QoS performance measure causes more difficulties [10]. QoS routing is in general a Multiconstrained Path problem and is proven to be a NP-Complete problem [7]. A number of heuristic algorithms are proposed to address this problem.

QoS routing could be classified as follows,

(A) Hop Constrained routing: In recent IETF QoS routing protocols, hop-constrained routing has been a emerging problem due to hop-minimization often turn out to consume less system resource (e.g. bandwidth) to achieve better performance metrics (e.g. delay) [2], [5]. Hop-minimization routing could easily be tackled with Bellman-Ford algorithm.

(B) Minimized Average Delay routing: For system operator, average delay minimization often leads to better resource utilization. Most existing researches tackle this problem by load balancing. In [8], optimization-based approach is proposed to tackle this problem. However, the approaches taken in (A) and (B) are to achieve global network optimization which might fail to address the user-specified delay.

(C) ε approximation routing: The basic idea of approximation routing algorithm [6], [11], [13] is to find a path whose cost is at most $(1 + \varepsilon)$ times the cost of optimal path where $\varepsilon > 0$. The smaller the ε, the better the performance of the approximation algorithms. However, it results in increasing computational complexity. Also, delay on a link is a constant such that it does not take queueing delay into account.

(D) User-optimization QoS routing: This approach directly takes user-perceived delay QoS into account. In [4], Cheng and Lin took a user-optimization approach and considered a fairness issue by minimizing the maximum individual end-to-end packet delay in virtual circuit networks by using the Lagrangean relaxation scheme. However, they only consider the user-perceived end-to-end delay, without taking the system delay optimization into consideration.

Based on these researches, we attempt to jointly consider both system and user perspectives typically in virtual circuit network. More precisely, we consider the virtual circuit routing problem of minimizing the average packet delay subject to end-to-end packet delay constraints for users. This problem is a difficult NP-complete problem as indicated in [14]. We propose an efficient and effective optimization based algorithms to obtain near-optimal solutions.

In the second part of this paper, besides routing assignment, we include another dimension, which is the link capacity assignment. Hence, we consider the virtual circuit network design problem of minimizing the total link installation cost subject to average and end-to-end delay constraints. As could be anticipated, this model is more general but more difficult than the previous one. Previous research tackles this problem with a two-phase approach [4]. The first phase is to determine the routing, which is traditional Minimum-Hop routing in order to consume the least system resource. The second phase is to determine the capacity assignment in order to fulfill the delay requirement. The two-phase heuristic is simple as after the routing is determined, the capacity assignment problem is a pure convex programming problem. However, it might sacrifice the optimality.

In order to jointly consider QoS routing and capacity assignment at the same time, an optimization based approach is then devised to attack these problems, where the problems are formulated as mathematical programming problems, followed by proposing algorithms based on Lagrangean relaxation. It is shown in the computational experiments that the proposed algorithm is both efficient and effective.

The remainder of this paper is organized as follows. In Sect. 2, a mathematical formulation of the DCR problem is proposed. In Sect. 3, a solution approach to the DCR problem based on Lagrangean relaxation is presented. In Sect. 4, heuristics are developed to calculate good primal feasible solutions for DCR problem. In Sect. 5, computational results for DCR problem are reported. In Sect. 6, mathematical formulation of JCR problem is proposed. In Sect. 7, solution approach to the JCR problem based on Lagrangean relaxation is presented. In Sect. 8, heuristics are developed to calculate good primal feasible solutions for JCR. In Sect. 9, computational results for the JCR problem are reported. Finally, Sect. 10 concludes this paper.

2. Problem Formulation to DCR

The virtual circuit network is modeled as graph where the processors are depicted as nodes and the communication channels are depicted as arcs. We show the definition of the following notation.

V	={1,2,... N}, the set of nodes in the graph
L	the set of communication links in the communication network
W	the set of Origin-Destination (O-D) pairs in the network
γ_w	(packets/sec): the arrival rate of new traffic for each O-D pair $w \in W$, which is modeled as a Poisson process for illustration purpose
C_l	(packets/sec), the link capacity of each link $l \in L$
P_w	a given set of simple directed paths from the origin to the destination of O-D pair w
x_p	a routing decision variable which is 1 when path $p \in P_w$ is used to transmit the packets for O-D pair w and 0 otherwise
δ_{pl}	the indicator function which is 1 if link l is on path p and 0 otherwise
D_w	the maximum allowable end-to-end delay for O-D pair w
g_l	aggregate flow over link l, which is equaled to $\sum_{p \in P_w} \sum_{w \in W} x_p \gamma_w \delta_{pl}$
f_l	the estimated aggregate flow over link l, which is the auxiliary variable
y_{wl}	Auxiliary variable, which is equaled to $\sum_{p \in P_w} x_p \gamma_w$

Under the assumption of Kleinrock's independence assumption [9], the arrival of packets to each buffer is a Poisson process where the rate is the aggregate flow over the outbound link. Assuming that the transmission time for each packet is exponentially distributed with mean C_l^{-1}. Each buffer is modeled as an $M/M/1$ queue, as in previous research [8].

Moreover, the formulation could be extended to any non $M/M/1$ model with monotonically increasing and convexity performance metrics. For the illustration purpose, the formulation will be based on the $M/M/1$ model. To determine a path for each O-D pair to minimize the average packet delay with maximum allowable end-to-end delay is formulated as a nonlinear combinatorial optimization problem, as shown below.

$$Z_{IP1} = \min \frac{1}{\sum_{w \in W} \gamma_w} \sum_{l \in L} \frac{f_l}{C_l - f_l} \tag{IP1}$$

subject to:

$$\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} \le D_w \qquad \forall w \in W \tag{1}$$
$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \tag{2}$$
$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \tag{3}$$
$$\sum_{p \in P_w} x_p \delta_{pl} \le y_{wl} \qquad \forall w \in W, l \in L \tag{4}$$
$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L \tag{5}$$
$$g_l \le f_l \qquad \forall l \in L \tag{6}$$
$$0 \le f_l \le C_l \qquad \forall l \in L. \tag{7}$$

Objective function is to minimize the average packet delay. Constraint (1) requires that the end-to-end packet delay should be no larger than D_w for each O-D pair. Constraints (2) and (3) require that the all the traffic for each O-D pair should be transmitted over exactly one path. Constraints (4) and (6) should be equalities, by changing equalities into smaller than or equals to is a relaxation, and we could prove that the equality should hold at the optimal point. Lemma 1 and Lemma 2 prove this argument. Constraint (5) is the integer constraint. Constraint (6) and (7) require that the aggregate flow on each link should not exceed the link capacity.

The above formulation is a nonlinear multicommodity flow problem, since each O-D pair transmits different type of traffic over the network. It is easy to show that (IP1) is a nonconvex programming problem by verifying the Hessian of $\frac{y_{wl}}{C_l - f_l}$.

Lemma 1. *In Constraint (4), the equality should hold at the optimal point of (IP1).*

Proof. Proof by contradiction. Assume the equality for Constraint (4) does not hold at the optimal point of (IP1), that is, some y_{wl} are one and the corresponding $\sum_{p \in P_w} x_p \delta_{pl}$ is zero. Under the circumstances, examine Constraint (1). If these associated Constraints (1) are binding, that is, equality for Constraint (1) holds, we could always decrease y_{wl} to zero to make Constraint (1) unbinding, that is, less than holds at Constraint (1). When these associated Constraints (1) are becoming unbinding, the optimal value of (IP1) could be further reduced due to the decreasing number of binding constraints of Constraints (1), resulting in the violation of the assumption of optimality. This proves that the equality for Constraint (4) should hold at the optimal point of (IP1). □

Lemma 2. *In Constraint (6), the equality should hold at the optimal point of (IP1).*

Proof. Proof by contradiction. Assume the equality for Constraint (6) does not hold at the optimal point of (IP1), that is, some f_l are greater than its corresponding aggregate flow. By decreasing f_l to the corresponding aggregate flow, the feasibility of Constraint (1) still holds. In addition, since the objective function of (IP1) is a monotonically increasing function with respect to f_l. We could always decrease f_l to the corresponding aggregate flow, which will lead to smaller objective value of (IP1), resulting in the contradiction of the assumption of optimality. That proves that the equality for Constraint (6) should hold at the optimal point of (IP1). □

3. Lagrangean Relaxation to DCR

The algorithm development is based on Lagrangean relaxation. We dualize Constraints (1), (4) and (6) to obtain the following Lagrangean relaxation problem (LR1), where t, u, v are the Lagrangean multiplier vectors and t_w, v_{wl}, u_l are the associated Lagrangean multipliers.

$$\begin{aligned} Z_{D1}(t,u,v) = \min\ & \frac{1}{\sum_{w \in W} \gamma_w} \sum_{l \in L} \frac{f_l}{C_l - f_l} \\ &+ \sum_{w \in W} t_w \left(\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} - D_w \right) \\ &+ \sum_{w \in W} \sum_{l \in L} v_{wl} \left(\sum_{p \in P_w} x_p \delta_{pl} - y_{wl} \right) \\ &+ \sum_{l \in L} u_l (g_l - f_l) \qquad \text{(LR1)} \end{aligned} \tag{8}$$

subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \tag{9}$$
$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \tag{10}$$
$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L \tag{11}$$
$$0 \le f_l \le C_l \qquad \forall l \in L. \tag{12}$$

We can decompose (LR1) into two independent subproblems.

Subproblem 1: for x_p

$$\min \sum_{w \in W} \sum_{p \in P} \sum_{l \in L} (v_{wl} + u_l \gamma_w)\, x_p \delta_{pl} \tag{SUB1}$$

subject to (8) and (9).

Subproblem 2: for y_{wl} and f_l

$$\min \sum_{l \in L} \left(\frac{1}{\sum_{w \in W} \gamma_w} \frac{f_l}{C_l - f_l} + \frac{\sum_{w \in W} t_w y_{wl}}{C_l - f_l} - \sum_{w \in W} v_{wl} y_{wl} - u_l f_l \right) - D_w \sum_{w \in W} t_w \quad \text{(SUB2)}$$

subject to (10) and (11).

(SUB1) could be further decomposed into $|W|$ independent shortest path problem with nonnegative arc weights. It can be easily solved by the Dijkstras algorithm. The computational complexity of the above algorithm is $O\left(|V|^2\right)$ for each O-D pair. The $-D_w \sum_{w \in W} t_w$ term in the objective function of (LR1) can be dropped first and then added back to the objective value since it does not affect the optimal solution of (SUB2). Then (SUB2) can be decomposed into $|L|$ independent subproblems. For each link $l \in L$,

$$\min \frac{1}{\sum_{w \in W} \gamma_w} \times \frac{f_l}{C_l - f_l} + \frac{\sum_{w \in W} t_w y_{wl}}{C_l - f_l} - u_l f_l - \sum_{w \in W} v_{wl} y_{wl} \quad \text{(SUB2.1)}$$

subject to:

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W \qquad \text{and} \qquad 0 \le f_l \le C_l.$$

(SUB2.1) is a complicated problem due to the coupling of y_{wl} and f_l. On the other hand, the $\frac{1}{\sum_{w \in W} \gamma_w} \times \frac{f_l}{C_l - f_l}$ term in the objective function of (SUB2.1) is a nonnegative and monotonically increasing function with respect to f_l, and it will not affect the optimal value of the following terms in (SUB2.1). Therefore, the algorithm developed in [4] can be used to solve (SUB2.1). For the convenience of the reader, the algorithm is provided below.

Step 1. Solve $\left(\frac{t_w}{C_l - f_l} - v_{wl} = 0\right)$ for each O-D pair w, call them the break points of f_l.
Step 2. Sorting these break points and denoted as $f_l^1, f_l^2, \ldots f_l^n$.
Step 3. At each interval, $f_l^i \le f_l \le f_l^{i+1}$, $y_{wl}(f_l)$ is 1 if $\frac{t_w}{C_l - f_l} - v_{wl} \le 0$ and is 0 otherwise.
Step 4. With the interval, $f_l^i \le f_l \le f_l^{i+1}$, let a_l be $\sum_{w \in W} t_w y_{wl}(f_l)$ and b_l be $\frac{C_l}{\sum_{w \in W} \gamma_w}$, then the local minimal is either at a boundary point, f_l^i or f_l^{i+1}, or at point $f_l^* = C_l - \sqrt{\frac{a_l + b_l}{u_l}}$.
Step 5. The global minimum point can be found by comparing these local minimum points.

The computational complexity of the above algorithm is $O(|W| \log |W|)$ for each link.

Applying the algorithms proposed above, we can successfully solve the (LR1) problem optimally. According to the weak Lagrangean duality theorem, which states that $Z_{Dl}(t, u, v)$ is a lower bound on Z_{IP1} [1]. We construct the following dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method.

$$Z_D = \max Z_{D1}(t, u, v) \qquad \text{(D1)}$$

subject to : $t, u, v \ge 0$.

Let the vector S be a subgradient of $Z_{D1}(t, u, v)$ at (t, u, v). In iteration x of the subgradient optimization procedure, the multiplier vector $m^x = (t^x, u^x, v^x)$ is updated by $m^{x+1} = m^x + \alpha^x S^x$, and $S^x(t, u, v) = \left(\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} - D_w,\ g_l - f_l,\ \sum_{p \in P_w} x_p \delta_{pl} - y_{wl}\right)$. The step size α^x is determined by $\delta \frac{Z_{IP1}^h - Z_{D1}(m^x)}{\|S^x\|^2}$, where Z_{IP1}^h is the best primal objective function value found at iteration x (an upper bound on optimal primal objective function value), and δ is a constant ($0 \le \delta \le 2$).

4. Getting Primal Feasible Solutions to DCR

To obtain the primal feasible solutions to the DCR problem, solutions to the Lagrangean relaxation problems (LR1) are considered. For example, if a solution to (LR1) is also feasible to (IP1), i.e., satisfy the capacity constraints and end-to-end delay constraints, then it is considered as a primal feasible solution to (IP1); otherwise, it will be modified so that it would be feasible to (IP1).

Three getting primal heuristics are developed to improve the effectiveness of the algorithms.

(1) When a solution to (LR1) is found, the routing assignment for the maximum end-to-end delay path is reassigned to another path to reduce the value of maximum end-to-end delay.
(2) When the end-to-end delay constraints are violated, identify the paths that violate end-to-end delay constraints, increase the arc weights along these paths, then recalculate the routing assignments.
(3) When a solution to (LR1) is infeasible for capacity constraints, the arc weight for the overflow link is increased, and then the routing assignments are recalculated.

According to the computational experiments, the second heuristic can get a better solution in most of the cases. The computational complexity of the second heuristic is $O\left(|W||V|^2\right)$.

In the following, we show the complete algorithm (denoted as LGR-DCR) to solve Problem (DCR).

Algorithm LGR-DCR
begin
read user input file (network topology; IP link configurations; traffic and QoS requirements);
initialize the Lagrangean multiplier vectors (t, u, and v) to be all zero vector;
UB := very large number; *LB*:= 0;
quiescence_age :=0;
step_size_coefficient :=2;
for *iteration*:= 1 **to** *Max_Iteration_Number* **do**

begin
run sub-problem (SUB1);
run sub-problem (SUB2);
calculate Z_{D1};
if $Z_{D1} > LB$ **then** $LB := Z_{D1}$ and *quiescence_age* :=0;
else *quiescence_age* :=*quiescence_age*+1;
if *quiescence_age*=*Quiesceince_Threshold* **then** *quiescence_age*:=0 and *step_size_coefficient* := *step_size_coefficient*/2;
run *Primal_Heuristic_Algorithm*;
if $ub < UB$ then UB:=ub; /**ub* is the newly computed upper bound */
run update-step-size;
run update-multiplier;
end;
end;

And the computational complexity for LGR-DCR is $O\left(|W||V|^2 + |L||W|\log|W|\right)$ for each iteration.

5. Computational Experiments on DCR

The computational experiments for the algorithms developed in Sects. 3 and 4 are coded in C and performed on a PC with INTEL PII-233 CPU. We tested the algorithm for 3 networks - ARPA1, GTE, OCT with 21, 12, and 26 nodes. The network topologies are shown in Figs. 1–3. And the computational time for these network topologies are all within fifteen minutes.

Max_Iteration_Number and *Quiesceince_Threshold* are set to 1000 and 30 respectively. The *step_size_coefficient* is initialized to be 2 and will be halved when the objective function value of the dual problem does not improve for iterations up to *Quiesceince_Threshold*. It is assumed that the traffic demand of each O-D pair is one packet per second. Unlike [4], the candidate path set does not need to be prepared in advance and all possible candidate paths are considered for each O-D pair.

We perform two sets of computational experiments. In the first set of computational experiments, the choice of the D_w value is fixed so as to examine the solution quality of the proposed algorithms.

Table 1 summarizes the results. The first column is the type of the network topology. The second column is the link capacity. The third column is the maximum allowable end-to-end delay (D_w). The forth column reports the lower bound from the algorithms in Sect. 3. The fifth column reports the upper bound from the algorithms in Sect. 4. The sixth column reports the error gap between the lower bound and the upper bound. Here the error gap is defined as the ((Upper bound − Lower bound)/Lower bound) × 100%. The seventh column reports the maximum end-to-end delay among all O-D pairs. As can be seen in the sixth column, the gap between the lower bound and the upper bound are very tight for all different network topologies and link capacities when the value of D_w is loose as compared to the maximum end-to-end delay among all O-D pairs.

Since D_w has a significant impact on the solution of the DCR problem. In the second set of computational experiments, we try to examine the impact of the D_w value on the solution quality of minimum average delay. Figures 4–6

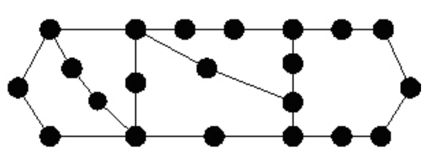

Fig. 1 ARPA1 network with 21 nodes.

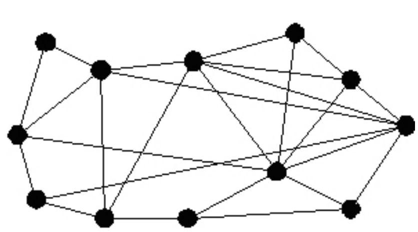

Fig. 2 GTE network with 12 nodes.

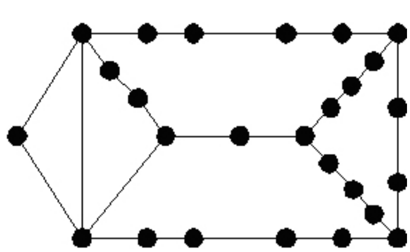

Fig. 3 OCT network with 26 nodes.

Table 1 Comparison of solution quality obtained by various networks.

Topo[a]	C_l[b]	D_w[c] (msec)	LB[d] (msec)	UB[e] (msec)	EG[f] (%)	M[g] (msec)
ARPA1	65	460	109.0	110.3	1.18	238.7
	70	300	93.7	94.3	0.65	200.7
	100	250	51.0	51.1	0.1	106.2
	150	130	29.1	29.1	0	60.4
	200	100	20.4	20.4	0	42.1
GTE	65	800	28.2	28.2	0	50.9
	70	500	26.0	26.0	0	46.9
	100	100	17.8	17.8	0	31.9
	150	100	11.7	11.7	0	20.8
	200	190	8.7	8.7	0	15.5
OCT	65	1400	351.3	357.8	1.88	805
	70	1000	237.8	240.6	1.2	525.6
	100	340	86.2	86.8	0.69	170.4
	150	150	42.4	42.4	0	81.7
	200	100	28.2	28.2	0	53.8

[a] Network topoloty: Topo
[b] Link Capacity: C_l
[c] Threshold of D_w : Th-D_w
[d] Lower bound : LB
[e] Upper bound : UB
[f] Error Gap : EG
[g] Maximum end-to-end delay : M
[h] Results from [4] : [4]
* The work in [4] does not perform the computational experiments in these network settings.

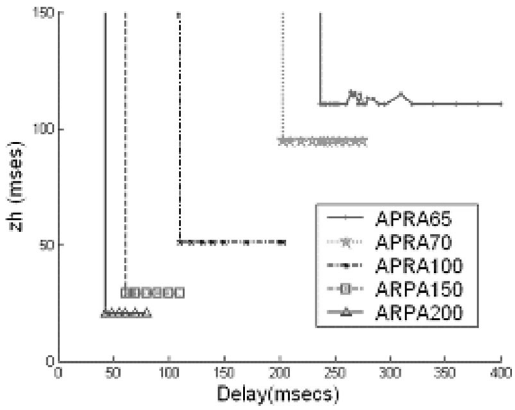

Fig. 4 Upper bound for different D_w values in the ARPA1 network.

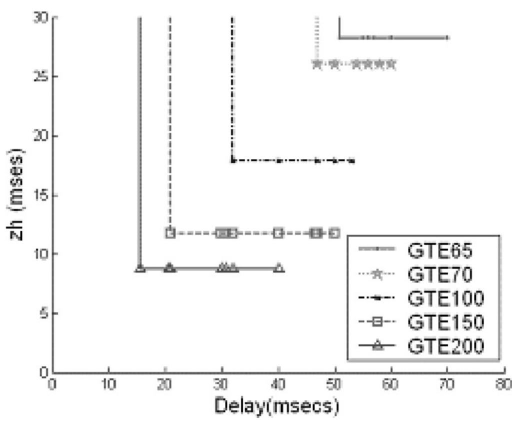

Fig. 5 Upper bound for different D_w values in the GTE network.

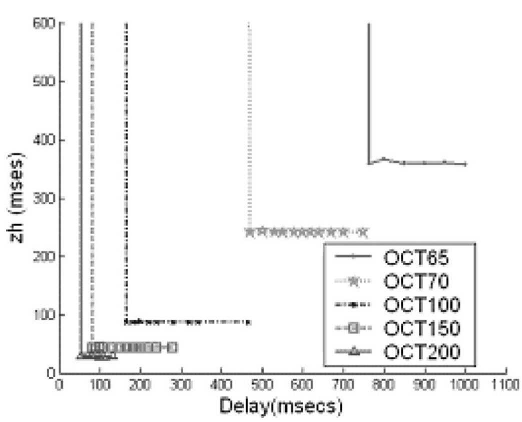

Fig. 6 Upper bound for different D_w values in the OCT network.

show the results for the ARPA1, GTE, OCT network. It is clear to see that the upper bound remains almost the same with different D_w values. When the D_w value is below a certain threshold (as indicated in third column of Table 2), the primal solution could not be found.

Table 2 summarizes this result. The first column is the type of the network topology. The second column is the link capacity. The third column is the threshold of the maximum allowable end-to-end delay (D_w). The forth column reports the lower bound from the algorithms in Sect. 3. The fifth column reports the upper bound from the algorithms in Sect. 4. The sixth column reports the error gap between the lower bound and the upper bound. The seventh column reports the maximum end-to-end delay among all O-D pairs. The eighth column reports the results from [4]. The maximum end-to-end delay value in the seventh column is the optimal value for the minimax end-to-end delay routing problem developed in [4]. It is clear to see the DCR performs better than the algorithms developed in [4] with performance improvement up to 19.1% (in OCT 65). There is one thing that needs to be addressed: all possible candidate paths are considered for each O-D pair in this work but only three candidate paths are pre-chosen for each O-D pair in [4]. Although we obtain a tighter upper bound than the minimax end-to-end delay routing problem developed in [4], this comparison is not on the same basis. On the other hand, the gap between the lower bound and the upper bound of the DCR problem is very tight, which indicates that the algorithms that we develop can achieve good system objective (average packet delay) even in stringent end-to-end delay requirements.

Table 2 Comparison of solution quality obtained by various networks at the of maximum allowable end-to-end delay requirements.

Topo[a]	C_l[b]	Th-D_w[c] (msec)	LB[d] (msec)	UB[e] (msec)	EG[f] (%)	M[g] (msec)	[4][h]
ARPA1	65	237.7	108.9	110.3	1.23	237.2	N/A*
	70	203	93.6	94.4	0.9	202.2	N/A
	100	110	50.7	51.2	1.02	108	N/A
	150	61	28.97	29.1	0.5	60.9	N/A
	200	43	20.35	20.37	0.09	42.3	N/A
GTE	65	50.88	28.17	28.23	0.2	50.876	N/A
	70	47	26.026	26.046	0.08	46.9	N/A
	100	32	17.82	17.83	0.08	31.9	N/A
	150	21.5	11.68	11.689	0.07	20.8	N/A
	200	15.5	8.6	8.7	0.8	15.49	N/A
OCT	65	727	351.3	364.2	3.67	722	860.3
	70	470	237.9	240.8	1.21	469.8	514.6
	100	166.8	86.24	86.9	0.8	166.7	168.4
	150	81.2	42.4	42.5	0.2	81.1	81.7
	200	54	28.04	28.16	0.4	53.9	54.1

An important observation could be found by examining the fifth and seventh column of Table 2. The maximum end-to-end delay is almost twice of average delay. Figures 4–6 show that the average delay remains the same if the D_w is greater than the threshold. These results indicate that any end-to-end constraint with D_w greater than twice of average delay could be ignored to speed-up the searching.

Parameter *Max_Iteration_Number* determine number of iterations that algorithm (LGR-DCR) should perform, and the criterion to set this parameter is to make sure that it will converge for both lower bound (LB) and upper bound (UB). For example in Fig. 7, we consider OCT network with link capacity 65. The LB steadily increase and UB steadily decrease from 200 to 600 iterations. After iteration 600, the LB and UB will all converge. In this case, *Max_Iteration_Number* should be set as 600. From our observation, setting this parameter as 1000 will ensure all these above experiments converge.

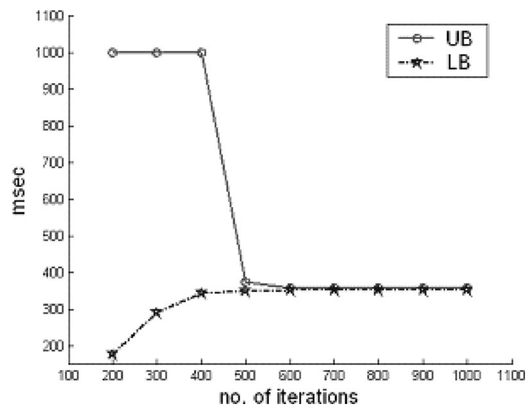

Fig. 7 UB and LB for different no. of iterations in OCT network.

6. Mathematical Formulations to JCR

We show the definition of the following new notation for JCR.

$\Psi_l(C_l)$	the link assignment cost for link $l \in L$, with respect to the link capacity C_l
k	the maximum allowable average cross network delay
A_l	the candidate set of link capacity assignment for link $l \in L$

$$Z_{IP2} = \min \sum_{l \in L} \Psi_l(C_l) \tag{IP2}$$

subject to:

$$\frac{1}{\sum_{w \in W} \gamma_w} \sum_{l \in L} \frac{f_l}{C_l - f_l} \le K \tag{13}$$

$$\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} \le D_w \qquad \forall w \in W \tag{14}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \tag{15}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \tag{16}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le y_{wl} \qquad \forall w \in W, l \in L \tag{17}$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L \tag{18}$$

$$g_l \le f_l \qquad \forall l \in L \tag{19}$$

$$0 \le f_l \le C_l \qquad \forall l \in L \tag{20}$$

$$C_l \in A_l \qquad \forall l \in L. \tag{21}$$

As shown in (IP2), the objective of JCR is to minimize the total link installation cost in order to satisfy the average and end-to-end delay requirements.

7. Lagragean Relaxation to JCR

The algorithm development is based on Lagrangean relaxation. We dualize Constraints (12), (13), (16) and (18) to obtain the following (LR2).

$$\begin{aligned} &Z_{D2}(a,b,c,d) \\ &= \min \sum_{l \in L} \Psi_l(C_l) + a\left(\frac{1}{\sum_{w \in W} \gamma_w} \sum_{l \in L} \frac{f_l}{C_l - f_l} - K\right) \\ &\quad + \sum_{w \in W} b_w \left(\sum_{l \in L} \frac{y_{wl}}{C_l - f_l} - D_w\right) \\ &\quad + \sum_{w \in W} \sum_{l \in L} c_{wl} \left(\sum_{p \in P_w} x_p \delta_{pl} - y_{wl}\right) \\ &\quad + \sum_{l \in L} d_l (g_l - f_l) \end{aligned} \tag{LR2}$$

subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \tag{22}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \tag{23}$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L \tag{24}$$

$$0 \le f_l \le C_l \qquad \forall l \in L \tag{25}$$

$$C_l \in A_l \qquad \forall l \in L. \tag{26}$$

We can decompose (LR2) into two independent subproblems.

Subproblem 1: for x_p

$$\min \sum_{w \in W} \sum_{p \in P_w} \sum_{l \in L} (c_{wl} + d_l \gamma_w)\, x_p \delta_{pl} \tag{SUB3}$$

subject to (21) and (22).
Subproblem 2: for C_l, y_{wl} and f_l

$$\begin{aligned} \min \sum_{l \in L} &\left(\Psi_l(C_l) + \frac{a}{\sum_{w \in W} \gamma_w} \frac{f_l}{Cl - f_l}\right. \\ &\left. + \frac{\sum_{w \in W} b_w y_{wl}}{C_l - f_l} - \sum_{w \in W} c_{wl} y_{wl} - d_l f_l\right) \\ &- aK - D_w \sum_{w \in W} b_w \end{aligned} \tag{SUB4}$$

subject to (23), (24) and (25).

(SUB3) can be further decomposed into $|W|$ independent shortest path problem with nonnegative arc weights. It can be easily solved by the Dijkstras algorithm. The computational complexity of the above algorithm is $O\left(|V|^2\right)$ for each O-D pair. The $-aK - D_w \sum_{w \in W} b_w$ term in the objective function of (SUB4) could be dropped first and then added back to the objective value since it will not affect the optimal solution of (SUB4). Then (SUB4) can be decomposed into $|L|$ independent subproblems. For each link $l \in L$,

$$\begin{aligned} \min \Psi_l(C_l) &+ \frac{a}{\sum_{w \in W} \gamma_w} * \frac{f_l}{C_l - f_l} \\ &+ \frac{\sum_{w \in W} b_w y_{wl}}{C_l - f_l} - \sum_{w \in W} c_{wl} y_{wl} - d_l f_l \end{aligned} \tag{SUB4.1}$$

subject to:

$$y_{wl} = 0 \text{ or } 1 \quad \forall w \in W, \quad 0 \le f_l \le C_l \quad \text{and} \quad C_l \in A_l.$$

(SUB4.1) is a complicated problem due to the coupling of three decision variables, y_{wl}, f_l and C_l. Due to the possible limited candidate set of each link capacity configuration, we could exhaustively search all possible link capacity to identify the optimal solution with respect to the corresponding y_{wl} and f_l. Then the question lies in how we could find the optimal solution of y_{wl} and f_l to (SUB4.1) when the link capacity is constant, which is the same problem in (SUB2.1). Hence, for all link capacity configurations for link $l \in L$, we could apply the same algorithm in (SUB2.1) to solve (SUB4.1). Then the global minimum solution could be found by comparing the minimum solution with respect to each link capacity configuration. The computational complexity of the above algorithm is $O(|W||A_l|\log|W|)$ for each link l.

According to the algorithms proposed above, we could successfully solve the (LR2) problem optimally. By using the weak Lagrangean duality theorem, which states that $Z_{D2}(a, b, c, d)$ is a lower bound on Z_{IP2} [1]. And use the same subgradient procedure in (D1) to calculate the tightest lower bound and solve the dual problem.

8. Getting Primal Feasible Solutions to JCR

To obtain the primal solutions to the JCR problems, solutions to the (LR2) are considered. First consider the routing assignments from (SUB3). When the routing assignment is given, the aggregate flow on each link is also determined. In order to satisfy the average and end-to-end delay constraints, the **Add** and **Drop** heuristic are proposed to calculate good primal feasible solutions from the aggregate flow determined in (LR2).

The overall algorithm (denoted as JCR-Primal) is as follows.

Step 1. From the aggregate flow for each link determined from the routing assignments in (SUB3), under the link capacity constraints, the minimum link capacity could be determined. If no such link capacity in the candidate capacity configuration could satisfy the link capacity constraint, then no primal feasible solution exist, stop the JCR-Primal solution process.

Step 2. Verify the average cross network delay. If the average cross network delay is violated, enter the **Add** heuristic. After the **Add** heuristic, if the average cross network delay is still violated, then no primal feasible solution exist. Stop the JCR-Primal solution process.

Step 3. Verify the end-to-end delay for each O-D pair. If the end-to-end delay is violated for any O-D pair, reroute the traffic to another shortest path based on the current routing assignment where the arc weight is calculated as 1.0/(link capacity − aggregate flow). After the rerouting, if the end-to-end delay is still violated for this O-D pair, augment the link capacity along this new routing path in order to satisfy the end-to-end delay constraint. If all the links along this new routing path have been augment to the largest candidate capacity and the end-to-end delay is still violated, then no primal feasible solution exist. Stop the JCR-Primal solution process.

Step 4. Enter the **Drop** heuristic in order to obtain better primal feasible solutions.

The **Add** and **Drop** heuristic are depicted as follows.

Add heuristic:

a) If all the links have the largest capacity configuration, stop the whole add heuristic process. Else among all the links where their link capacity are not equal to the largest candidate link capacity, identify the most congested link, and augment the link capacity to the next higher link capacity configuration.

b) Verify the average cross network delay, if the average cross network delay is still violated, go to a) again.

Drop heuristic:

c) For a link, first decrease the link capacity to next lower capacity in the capacity configuration. Verify the capacity constraint, average cross network delay constraint and end-to-end delay constraints for all O-D pairs. If these constraints are all satisfied, then this link capacity could be decreased successfully, else augment the link capacity back to its original capacity configuration.

d) Repeat c) for all links in the network.

The computational complexity of the JCR-Primal is $O\left(|W|\left(|V|^2 + \sum_{l\in L}|A_l|\right)\right)$. As could be shown in the computational experiments, after introducing the **Add** and **Drop** heuristic, better primal feasible solutions could be obtained. On the other hand, since good initial primal feasible solution will lead to better solution quality and good convergence property, we propose the hop-minimization QoS algorithm (HopQoS) to calculate the initial good primal feasible solutions. The basic idea of HopQoS algorithm is to leverage on the hop minimization routing proposed by IETF [2], [5]. There are two steps in the HopQoS.

HopQoS heuristic:

Step 1. The routing assignment for each O-D pair is determined by hop-minimization algorithm. The hop-minimization could be easily implemented by Dijkstra algorithm or Bellman-Ford algorithm under the assumption that the arc-weight of each link is one and link capacity is infinite. After the routing assignment is fixed, the aggregate flow could be determined as well. Then, under the capacity constraint, the minimum capacity assignment could also be determined.

Step 2. Perform the **Add** and **Drop** heuristic to obtain good initial primal feasible solution and in the mean time to satisfy the average and end-to-end delay requirements.

In the following, we show the complete algorithm (denoted as LGR-JCR) to solve Problem (JCR).

Algorithm LGR-JCR
begin
read user input file (network topology; IP link capacity and cost configurations; traffic and QoS requirements);
initialize the Lagrangean multiplier vectors (a, b, c and d) to be all zero vector;
run HopQoS to get *UB*;
LB:= 0;
quiescence_age :=0;
step_size_coefficient :=2;
for *iteration*:= 1 **to** *Max_Iteration_Number* **do**
begin
run sub-problem (SUB3);
run sub-problem (SUB4);
calculate Z_{D2};
if $Z_{D2} > LB$ **then** $LB := Z_{D2}$ and *quiescence_age* :=0;
else *quiescence_age* :=*quiescence_age*+1;
if *quiescence_age* =*Quiesceince_Threshold* **then** *quiescence_age*:=0 and *step_size_coefficient* := *step_size_coefficient*/2;
run JCR-Primal;
if $ub < UB$ **then** UB:=ub; /**ub* is the newly computed upper bound */
run update-step-size;
run update-multiplier;
end;
end;

The computational complexity for LGR-DCR is $O\left(|W||V|^2 + |W|\,(\log|W|) \sum_{l\in L}|A_l|\right)$ for each iteration.

9. Computational Experiments on JCR

The computational experiments for the algorithms developed in Sects. 7 and 8 are coded in C++ and performed on a PC with INTEL PIII-800 CPU. In addition to the network topologies given in Figs. 1–3, we also tested the algorithm for 2 large networks—ARPA2 and CTNET with 61 and 38 nodes. The network topologies are shown in Figs. 8 and 9. The computational time for these network topologies are all within one hour.

Max_Iteration_Number and *Quiesceince_Threshold* are set to 1000 and 30 respectively. The *step_size_coefficient* is initialized to be 2 and will be halved when the objective function value of the dual problem does not improve for iterations up to *Quiesceince_Threshold*. It is assumed that the traffic demand of each O-D pair is one packet per second. All possible candidate paths are considered for each O-D pair. The candidate capacity configurations for each link ranges from 50 to 140, with step 10 for topologies in Figs. 1–3. The candidate capacity configurations for each link ranges from 50 to 190, with step 10 for topologies in Figs. 8 and 9. The cost for each capacity configuration is a concave function with respect to the number of candidate capacity configurations, ranging from 50 to 99.

In order to show the solution quality of our proposed algorithm, we implement the R&CA algorithm developed in [4] for comparison. The basic idea of R&CA algorithm is a two-phase algorithm, where the first phase is the hop minimization routing heuristic and the second phase is to determine the capacity assignment problem by convex programming technique.

Tables 3–7 summarize the results. The first column is the type of the network topology. The second column is the maximum allowable average cross network delay. The third column is the maximum allowable end-to-end delay (D_w).

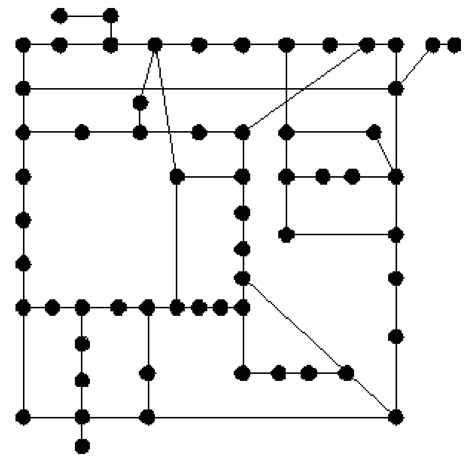

Fig. 8 ARPA2 network with 61 nodes.

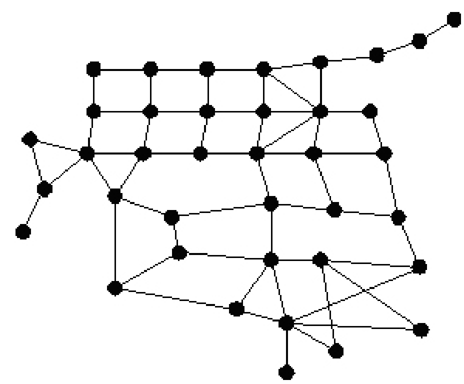

Fig. 9 CTNET network with 38 nodes.

Table 3 Solution quality for JCR at ARPA1 network topology.

Topo[i]	K (msec)	D_w (msec)	LB[j]	UB[k]	EG[l] (%)	R&CA [4]
ARPA1	40	50	2650	3997	51	4472
	50	100	2627	3814	45	4410
	60	120	2618	3571	36	4311
	70	140	2612	3431	31	4156
	80	160	2609	3295	26	3652
	90	180	2606	3177	21	3368
	100	200	2605	3085	18	3264
	110	220	2603	3010	15	3107

[i] Network topoloty: Topo
[j] Lower bound : LB
[k] Upper bound : UB
[l] Error Gap : EG
[@] No feasible solution could be found in R&CA [4].

Table 4 Solution quality for JCR at ARPA2 network topology.

Topo[i]	K (msec)	D_w (msec)	LB[j]	UB[k]	EG[l] (%)	R&CA [4]
ARPA2	210	420	9658	14100	46	$\infty^{@}$
	220	440	9646	13983	45	∞
	230	460	9645	13795	43	∞
	240	480	9630	13489	40	14644
	250	500	9604	13259	38	14601
	260	520	9540	13112	36	14304
	270	540	9456	12582	33	13951
	280	560	9406	12237	30	13021

Table 5 Solution quality for JCR at CTNET network topology.

Topo[i]	K (msec)	D_w (msec)	LB[j]	UB[k]	EG[l] (%)	R&CA [4]
CTNET	80	160	8129	11549	42	∞
	90	180	8104	11344	40	∞
	100	200	8098	11093	37	12013
	110	220	8095	10850	34	11978
	120	240	8069	10648	32	11767
	130	260	8013	10501	31	11589
	140	280	7986	10302	29	11421
	150	300	7915	10055	27	11310

Table 6 Solution quality for JCR at GTE network topology.

Topo[i]	K (msec)	D_w (msec)	LB[j]	UB[k]	EG[l] (%)	R&CA [4]
GTE	20	40	2501	3294	31	3574
	30	60	2500	2799	11	2917
	40	80	2500	2500	0	2749
	50	100	2500	2500	0	2610
	60	120	2500	2500	0	2500

Table 7 Solution quality for JCR at OCT network topology.

Topo[i]	K (msec)	D_w (msec)	LB[j]	UB[k]	EG[l] (%)	R&CA [4]
OCT	40	80	3419	5103	49	∞
	50	100	3383	4998	47	5794
	60	120	3335	4812	44	5565
	70	140	3333	4650	39	5212
	80	160	3308	4532	37	4950
	90	180	3298	4401	33	4808
	100	200	3307	4279	29	4412
	110	220	3274	4213	28	4263

The reason that we choose D_w to be $2K$ is the insight from the computational experiment of DCR, any $D_w > 2K$ end-to-end delay constraint would be unbinding. The fourth column reports the lower bound from the algorithms in Sect. 7. The fifth column reports the upper bound from the JCR-Primal in Sect. 8. The sixth column reports the error gap between the lower bound and the upper bound. The seventh column reports the primal feasible solutions from R&CA.

As can be seen in the sixth column, the gap between the lower bound and the upper bound are reasonably tight for all different network topologies. As shown in these tables, the solution quality is better in smaller network topologies. In addition, the solution quality is better in loose average and loose end-to-end delay requirements in these network topologies (e.g. have optimal solution at K=40,50,60 of Table 6).

Another interesting point is that the maximum end-to-end delay obtained by JCR should be better than DCR since DCR is a special case of JCR with even capacity assignment. The computational experiments did justify the argument. For example, in Table 7 D_w =80 msec, we could still obtain feasible solutions (the maximum end-to-end delay is 72.6 msec) with IP link capacity at most 140. However, in OCT network of Table 2, the maximum end-to-end delay is 81.1 msec with IP link capacity are all 150. Hence, the performance improvement of JCR over DCR is at least 12%.

As shown in Table 3–Table 7, all the solutions (upper bound) obtained from LGR-JCR always outperform R&CA. It is not surprising to see these results since R&CA implements two-phase algorithm to determine the routing and capacity assignment separately that will sacrifice the optimality. When the delay requirement is stringent, the improvement of LGR-JCR over R&CA is even more significant (e.g. in Table 4, K=210 msec, R&CA could not find feasible solutions).

10. Conclusions

As compared to the work in [4], which tried to achieve better fairness among users by minimizing the maximum end-to-end delay for virtual circuit networks without considering the system perspective (minimize the average packet delay). In this paper, for the first time, we considered the problem of minimizing the average packet delay with maximum allowable end-to-end delay requirements, which indicate that we try to obtain good system performance under user's end-to-end delay requirements.

We formulate this problem as a nonconvex and nonlinear multicommodity integral flow problem. These nonconvex and discrete (integer constraints) properties make the problem very difficult. We take an optimization-based approach by applying the Lagrangean relaxation technique to propose LGR-DCR algorithm. According to the computational experiments, the error gap (1–3%) between the upper bound and the lower bound is so tight that we can claim a near optimal solution is found. In addition, we could achieve better fairness than the work in [4] with performance improvement up to 19.1%. In addition, when the maximum end-to-end delay requirements are closer to the threshold, the upper bound (average packet delay) remains almost the same. This indicates that this LGR-DCR can obtain good average packet delay solution under stringent end-to-end QoS requirements.

Besides rerouting, capacity augmentation approach is proposed to address the increasing traffic demand and more stringent QoS requirements in network servicing. In the second part of this paper, we jointly consider the capacity assignment and routing assignment in virtual circuit network. This work is more complicated than the DCR problem since the link capacity is also a decision variable. We formulate this problem as a nonconvex and nonlinear multicommodity integral flow problem, where the objective is to minimize the total network capacity installation cost subject to average and end-to-end delay constraints. Besides the non-convexity associated with the end-to-end delay constraints, the concavity associated with the capacity cost in the objective function makes this problem more complicated than the DCR problem. Lagrangean relaxation techniques in conjunction with the **Add-Drop** heuristics are proposed to devise the LGR-JCR algorithm. As shown in the computational experiments, by including another decision variable - link capacity, the JCR outperform DCR in terms of maximum end-to-end delay at least 12%. In addition, the capacity assignment the LGR-JCR outperforms traditional two-phase approach (R&CA) under all tested cases and the performance improvement is more significant in more stringent delay QoS requirements.

Acknowledgement

This work is supported by the National Science Council, Taiwan, under grant number NSC 90-2219-E-128-001.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network Flows-Theory, Algorithms, and Applications, Prentice Hall, ISBN0-13-027765-7, 1993.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, T. Przygienda, and D. Williams, "QoS routing mechanisms and OSPF extensions," RFC no.2676, IETF, Aug. 1999.

[3] D. Bertsekas, Data Networks, Prentice-Hall, ISBN 0-13-201674-5, 1992.

[4] K.T. Cheng and F.Y.-S. Lin, "Minimax end-to-end delay routing and capacity assignment for virtual circuit networks," Proc. IEEE Globecom, pp.2134–2138, 1995.

[5] E. Crawley, R. Nau, B. Rajagopalan, and H. Sandick, "A framework for QoS based routing in the Internet," RFC no.2386, IETF, Aug. 1998.

[6] F. Ergun, R. Sinha, and L. Zhang, "QoS routing with performance-dependent costs," Proc. IEEE INFOCOM, pp.137–146, 2000.

[7] M.R. Garey and D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, 1979.

[8] B. Gavish and S.L. Hantler, "An algorithm for optimal route selection in SNA networks," IEEE Trans. Commun., vol.COM-31, no.10, pp.1154–1160, Oct. 1983.

[9] L. Kleinrock, Queueing Systems, vol.1 and 2, Wiley-Interscience, New York, 1975–1976.

[10] F.A. Kuipers, T. Korkmax, M. Krunz, and P. Van Mieghem, "Overview of constraint-based path selection algorithms for QoS routing," IEEE Commun. Mag., vol.40, no.12, pp.50–55, Dec. 2002.

[11] E.H. Lorenz, A. Orda, D. Raz, and Y. Shavitt, "Efficient QoS partition and routing of unicast and multicast," IEEE International Workshop on Quality of Services, pp.75–83, 2000.

[12] S. Nelakuditi, Z.L. Zhang, and R.P. Tsang, "Adaptive proportional routing: A localized QoS routing approach," Proc. IEEE INFOCOM, pp.1566–1575, 2000.

[13] A. Orda and A. Sprintson, "QoS routing: The precomputation perspective," Proc. IEEE INFOCOM, vol.1, pp.128–136, 2000.

[14] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," IEEE J. Sel. Areas Commun., vol.14, no.7, pp.1228–1234, 1996.

[15] H.-H. Yen and F.Y.-S. Lin, "Near-optimal delay constrained routing in virtual circuit networks," Proc. IEEE INFOCOM, vol.2, pp.750–756, 2001.

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

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