Find the optimal path solution to achieve path-constrained path-optimization routing

Hasan H. Hamudy

Al-Mansour University College

Abstract :

Modern computer networks are increasingly being used to transmit multimedia traffic such as video and audio besides supporting traditional data communication applications require strong *Quality of Services QoS* constraints. However, network traffic is highly diverse and each traffic type has unique requirements in terms of bandwidth, delay, loss, and availability and with explosive growth of the Internet, most traffic has begun moving towards *QoS* routing techniques.

In this research, an overview of the QoS routing problems is described through presenting the strengths and weaknesses points of routing strategies used for this problem.

It is noted from the results of studying above points that there is an important problem appeared in QoS routing which is called the path-constrained path-optimization routing problem, so it is proposed a two iterative algorithms to solve this problem these are: DCLC-A (Delay Constraint Least Cost-First Iterative) and DCLC-B (Delay Constraint Least Cost-Second Iterative) that have a wide search to find the optimal path , From the experimental results of our two proposed algorithms, it is found that the DCLC-B gives the optimal path solution between two nodes.

1. Introduction:

The delay-constrained least-cost (DCLC) path problem is searching for a path that has the minimum cost and a delay not exceeding a given upper bound. The *single-mixed-weight* idea in this paper can be briefly described as follows : "Given a network with a delay and a cost associated with each link, it can first obtain a single mixed weight for each link by combining its delay and cost in terms of one parameter, and then use Dijkstra's algorithm to find the corresponding shortest path". It can be theoretically proved that as long as the parameter is appropriately chosen the obtained shortest path must be a feasible solution with a cost no greater than that of the Least Delay (LD) path. Based on this result, a heuristic algorithm is used that can produce good solutions by executing Dijkstra's shortest path algorithm at most three times. To further improve the quality of the solution, then two iterative algorithms are proposed that can generate a series of parameters gradually improving the corresponding solutions. A large number of numerical experiments are carried out and the results of them are compared[1].

2. The DCLC Path Problem:

Any network can be represented by a directed graph G(V,E), where V is the set of nodes, and E is the set of links. A weight w defines a nonnegative real number *w*(e) associated with each link e, i.e., w: $E \to \mathrm{R}_0^+$. In particular, weight d: $E \to \mathrm{R}_0^+$ is called delay, while c: $E \to \mathrm{R}_0^+$ is called cost, where R+ is a real positive numbers.

A path is a finite sequence of non-repeated nodes p=($v_0,v_1,\ldots,v_k$), such that for $0 \le \mathrm{i} < \mathrm{k}$, there exists a link from v_i to v_{i+1}, i.e.. (v_i,v_{i+1}) $\in \mathrm{E}$. A link e $\in$ p means that path p passes through link e. In particular, the delay and cost of a path p are given by two equations below:

$$d\,(\mathrm{p}) = \sum_{e \in r} \mathrm{d(e)} \qquad \text{.......(1)}$$

$$c(p) = \sum_{e \in p} \mathrm{c(e)} \qquad \text{.......(2)}$$

In a general sense, the delay of a link is the average transmission time on that link, while the cost of a link may be the fee charged to transmitting a message on that link. However, delay and cost may be redefined as other metrics such as jitter, loss etc, as long as they are additive.

Now a formal definition for the DCLC (Delay Constraint Least Cost) problem using the above notation is illustrated below [2].

Definition 1:

Given a network G(V, E), a source node s $\in$ V and a destination node d $\in$V, a delay and a cost for each link, and a delay constraint C_d, the delay-constrained least-cost (DCLC) path problem is to find a path p from s to d, such that.

(i) d (p) $\leq$ C_d,

(ii)c(p) $\leq$ c(q) for any path q from s to d that satisfies d(q) $\leq C_d$,

(iii) there doesn't exit a path q from s to d, for which c(p)=c(q),while d(p) > d(q).

it should be noted though that the third requirement is not included in a standard definition for the DCLC problem; it is introduced here to reflect the preference for a path resulting in better performance if there exist more than one solution for the standard problem. For convenience, a path that at least satisfies the first requirement in the above definition is called a *feasible solution* (or feasible path); a path that satisfies all three requirements is called an *optimal solution* (or optimal path).The following definitions and notations are needed to describe the algorithms to be proposed in this paper [3]..

Definition 2:

Given two additive weights w_1 and w_2, a mixed weight $w=w_1+a\, w_2$ means that for any link e,

$$w(e) = w_1(e)+ a\, w_2(e) \quad ....(3)$$

where $a \in R_0^-$. Apparently, a mixed weight of two additive weights is also additive.

Definition 3:

Given a source node s, a destination node d and an additive weight w. A function (or procedure) Dijk (*w*) which returns the shortest path w from s to d found using Dijkstra's algorithm can be defined. In particular, it is equivalent to let p_d=Dijk (d) be the Least Delay (LD) path, and p_c=Dijk(c) the Least Cost (LC) path between s and d. Note that the relations $d(p_d) \leq d\ (p_c)$ and $c(p_d) \geq c\ (p_c)$ always hold.

Another function to be used in these algorithms is ModiDijk (c,d). If there exist multiple LC paths with different delays from s to t, function ModiDijk (c,d) will return the one of the minimum delay. This can be done using a modified Dijkstra's algorithm [2].

3. The Single Mixed Weight idea:

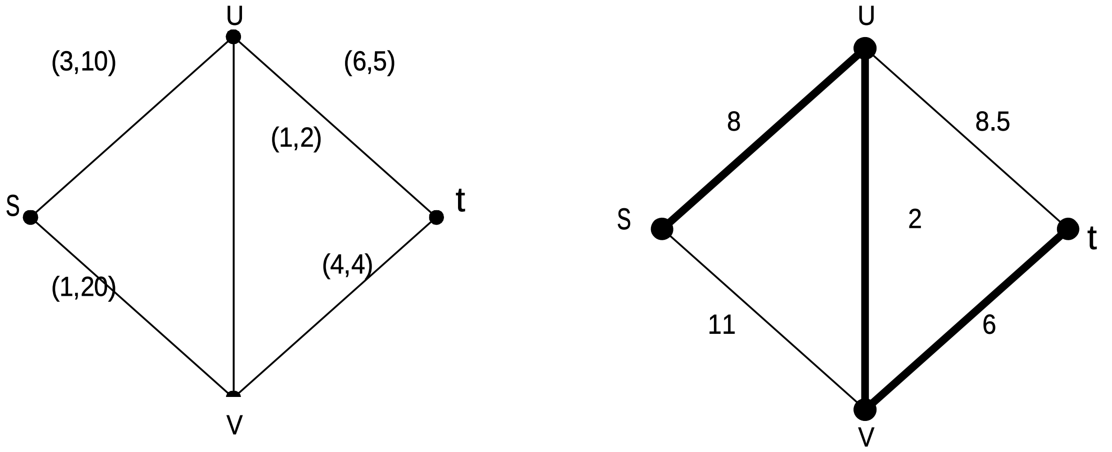
The basic idea of the algorithm in the next section is to solve the DCLC problem by first combining the delay and cost into a single mixed weight, and then using Dijkstra's algorithm to find a feasible path. In this section, the basic idea is illustrated through a simple network model.

Consider the problem defined in Figure 1(a), in which it is required to find a DCLC path from s to t with a delay bound of 8. Now, solving this problem manually, required to check all four paths between s and t, even though this network model is very simple. However, it is easy to find that the LC path is *s*®*u*®*t*, which has a delay of 9 and is infeasible. The LD path is *s*®*v*®*t*, which has a delay of 5 and a cost of 24, although the LD path is feasible,but it is not the optimal solution.

Now let us construct a mixed weight w=d + ac. If it can let a=0.5 . then the weight *w* associated with each link is shown in Figure 1(b). By using Dijkstra's algorithm, the shortest path from s to t, *s*®*u*®*v* ®*t*, can be easily found. This path has a delay of 8 and a cost of 16, and it turns out to be the optimal path.

This example indicates that selecting an appropriate parameter to construct a mixed weight, reducing the DCLC problem to the shortest path problem, this can be readily solved using Dijkstra's algorithm [4].

The key issue for this idea is how to choose the parameter for constructing the mixed weight. A randomly selected value for a **may result in a disappointing solution. For instance, in Figure 1 (c) let** a**= 0.2 hence the shortest path *w* becomes the LD path. While in Figure 1 (d), let** a**=2 hence the shortest path becomes the LC path [3,4].**

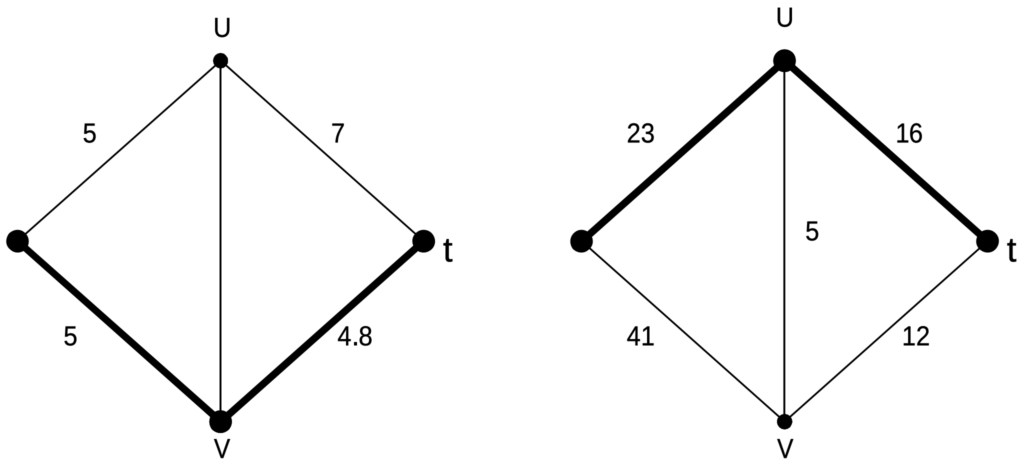


Find path from s to t with delay bound=8

(a) A DCLC problem

Shortest path *s*®*u*®*v*®*t*: delay=8, cost=16

(b) Mixed weights when a **= 0.5**



Shortest path *s*®*v*®*t*: delay=5, cost=24

(c) Mixed weights when a = 0.2

Shortest path *s*®*u*®*t*: delay=9, cost=15

(d) Mixed weights when a = 2

Figure 1: Illustration of the idea of mixed weight

4. Heuristic Algorithms for the DCLC Problem:[4,5]

In the DCLC problem, the attention will be focused on finding good feasible solutions. This section, will describe a basic algorithm which illustrates the other two proposed iterative algorithms based on it. As indicated in the previous section, the paramount importance in the single-mixed-weight idea is how to choose the parameter. Note that

If p=Dijk (d+ac), q=Dijk (d+bc), $a \geq b$, $a \in R_0^+$, $b \in R_0^+$, then : c(p) $\leq$ c(q), d(p) $\geq$ d(q). Previous part implies that the larger the parameter,

the smaller the cost of the resulting shortest path, while the larger the delay of the path. As long as the delay constraint is not violated, a larger parameter will absolutely result in a better solution. Therefore, the remaining work is to find the largest parameter such that the resulting path is feasible.This goal is achieved by (Lagrange relaxation technique) that calculates lower bounds and finds good solutions for DCLC problem. This technique produces the following equation to get the value of a that is given by:

$$a= \frac{C_d - d(p)}{c(p) - c(q)} \qquad ....(4)$$

The procedure of this algorithm (DCLC) is shown below [6].

Algorithm DCLC (G,s,t,c,d,C_d)

Step 0: Start.

Step 1: If there is more than one path with minimum cost, find the least cost path between s and d with minimum delay, and let this path nameḍq.

q ¬ ModiDijk (c,d)

Step 2: Check the delay of the selected path in (step 1) is less than or equal the delay constrained. If (d (q) $\leq$ C_d) then

Step 3: If the result of (step 2) is true then the suitable path is q.

return q

Step 4: If the result of (step 2) is false then let p be the shortest path in corresponding to delay (i.e. the path that has minimum delay).

p ¬ Dijk (d)

Step 5: Check if the delay of the path p if greater than delay constrained

if (d(p) > C_d) then

Step 6: If the result of (step 5) is true then there is no feasible path.

return NULL

Step 7: If the result of (step 5) is false then calculate the parameter of constructing the mixed weight ( ),from equation (4).

Step 8: Convert the two parameters in each link to one parameter (w), where

w = delay + . × cost for each link

then let P is the minimum path corresponding to (w) value.

p ¬ Dijk (d + ac)

Step 9: **P is the algorithm choice of most suitable path.**

return p

Step 10: **End.**

5. The two proposed iterative algorithms

Algorithm DCLC is very simple and fast, it only needs to execute Dijkstra's algorithm at most three times. In the following section, two proposed iterative algorithms will be described that can improve the quality of the solution.

5.1 The procedure of the first proposed iterative algorithm

After obtaining a feasible path p which is better than LD path, another parameter can be computed As long as the new parameter is larger than the previous one, a better solution, possibly can be found. The above description for first proposed iterative algorithm named as *(DCLC-A)* and the procedure of this algorithm is shown below:

Algorithm DCLC-A (G,s,t,c,d,C_d)

Step 0: **Start.**

Step 1: **if there is more than one path with minimum cost, find the least cost path between s and d with minimum delay, and let this path namedq.**

q ¬ ModiDijk (c,d)

Step 2: **Check the delay of the selected path in (step 1) is less than or equal the delay constrained. If (d (q) $\leq$ C_d) then**

Step 3: **If the result of (step 2) is true then the suitable path is q.**

return q

Step 4: **If the result of (step 2) is false then let p be the shortest path in corresponding to delay (i.e. the path that has minimum delay).**

p ¬ Dijk (d)

Step 5: Check if the delay of the path p if greater than delay constrained

if ($d(p) > C_d$) then

Step 6: If the result of (step 5) is true then there is no feasible path.

return NULL

Step 7: Create a flag variable named continue with Boolean type having initial value = True continue = TRUE

Step 8: Building a closed loop with a condition continue = true and exit the loop just when continue = false While continue do

Step 9: If the result of (step 5) is false then calculate the parameter of constructing the mixed weight ( ),from equation (4).

Step 10: Convert each path to one parameter (w), where w = delay + . × cost for each path , then let r is the minimum path corresponding to (w) value

r ¬ Dijk (d + a c)

Step 11: Check: If cost of path (r) equal the cost of path (q) or the cost of path P.

if (c (r) = c(q) or c (r) = c(p) then

Step 12: If the result of the condition in (step 11) is true then let continue = false to exit the loop (step 15) Continue = FALSE

Step 13: If the result of the condition in (step 11) is false then goto (step 14) else

Step 14: Let P = the minimum path (r) p ¬ r

Step 15: P is the algorithm choice of most suitable path.

return p

Step 16: End.

In the case that the LC path is infeasible and the LD path is feasible, the algorithm enters a loop. In each of the successive iterations, the parameter is computed in terms of the previous solution, and thus a new solution is obtained. Since the parameter increase at each step, the solution may possibly improve (at least remains the same as the previous one). When the mixed weight of the news solution is the same as that of the previous one, the algorithm cannot find a better path that has a smaller mixed weight. However, if a solution has the same mixed weight but a lower cost than the previous one, it can still be found because this algorithm stops only if the current solution has the same cost as the previous solution or the LC path.

5.2 The procedure of the second proposed iterative algorithm:

The first iterative algorithm can be further improved if c(q) can be "relaxed". This means that if a feasible path p will be found and an infeasible path q with $c(q) > c(p_c)$, then using

$$a=\frac{d(q)-d(p)}{c(p)-c(q)} \qquad ....(5)$$

instead of equation (4), a better feasible solution can be found. This is because the former parameter is larger, a further extension is to substitute C_d by d(q), resulting in an even larger parameter. In the case that the LD path p_d is feasible, while the LC path P_c is infeasible, this algorithm sets p_d to p and p_c to q as their initial values, respectively, and then enters an iterative procedure. In each of the successive iterations, either p is updated with a better feasible solution in the sense that its cost is lower, or q is updated with a better infeasible solution in the sense that its delay is lower. The algorithm terminates if the quality of the solution cannot be further improved. The above description for second proposed iterative algorithm named as (DCLC-B) and the procedure of this algorithm is shown below:

Algorithm DCLC-B (G,s,t,c,d,C_d)

<u>Step 0:</u> Start.

<u>Step 1:</u> If there is more than one path with minimum cost, find the least cost path between s and d with minimum delay if there is more than one path with minimum cost, and let this path namedq. q ¬ ModiDijk (c,d)

<u>Step 2:</u> Check the delay of the selected path in (step 1) is less than or equal the delay constrained. If (d (q) $\leq$ C_d) then

<u>Step 3:</u> If the result of (step 2)is true then the suitable path is q.

return q

<u>Step 4:</u> If the result of (step 2) is false then let p be the shortest path in corresponding to delay (i.e. the path that has minimum delay).

p ¬ Dijk (d)

<u>Step 5:</u> Check if the delay of the path p if greater than delay constrained

if (d(p) > C_d) then

<u>Step 6:</u> If the result of (step 5) is true then there is no feasible path.

return NULL

<u>Step 7:</u> Create a flag variable named continue with Boolean type having initial value = True continue = TRUE

<u>Step 8:</u> Building a closed loop with a condition continue = true and Exit the loop just when continue = false While continue do

<u>Step 9:</u> If the result of (step 5) is false then calculate the parameter of constructing the mixed weight ( ), from equation(6).

<u>Step 10:</u> Convert each path to one parameter (w), where w = delay + . × cost for each path, then let r is the minimum path corresponding to (w) value

r ¬ Dijk (d + a c)

<u>Step 11:</u> Check: If cost of path (r) equal the cost of path (q) or the cost of path P.

if (c (r) = c(q) or c (r) = c(p) then

<u>Step 12:</u> If the result of the condition in (step 11) is true then let continue = false to exit the loop (step 17) Continue = FALSE

Step 13: If the result of the condition in (step 11) is false then check if the delay of the path (r) is less than or equal to the delay constrained then

Else if (d(r) $\le$ Cd) then

Step 14: Let P = the path r, if the result of condition in (step 13) is true go to (Step 9)

p ¬ r

Step 15: If the result of the condition of (step 13) is false then go to (step 16) else

Step 16: Let q = the path r go to (Step 9) q ¬ r

Step 17: P is the algorithm choice of most suitable path.

return p

Step 18: End.

5.3 Demonstration Examples:

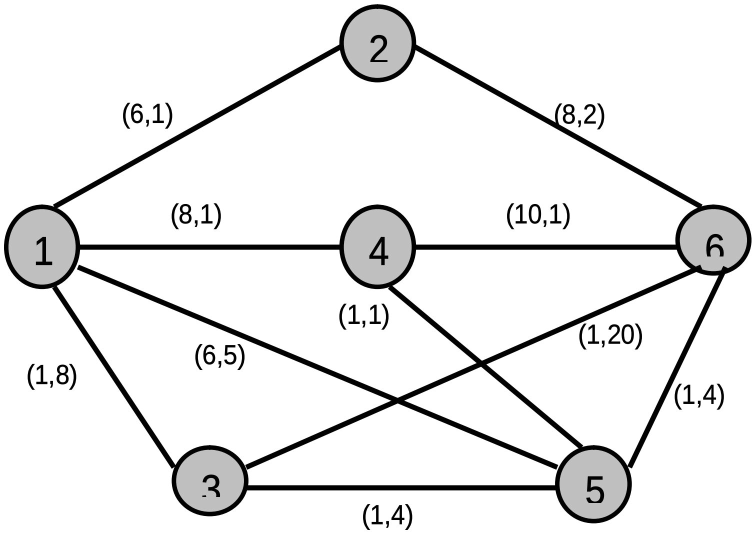
In the following sections, demonstration examples are used to describe the DCLC algorithm and the two proposed iterative algorithms(DCLC-A & DCLC-B) and how these algorithms proceed to find a solution.

5.3.1 Analysis and discussion the DCLC Algorithm:

Figure 2 shows a DCLC problem, in which it is required to find a DCLC path from node 1 to node 6 with a delay upper bound of 12. It is easy to find that the LD path is p_d= {1® 3 ® 6}, while the LC path is p_c= {1® 4 ® 6}. Thus it has d(p_d)= 2, c(p_d)= 28, d(p_c)=18, and c(p_c)=2. Using an exact algorithm, the optimal solution to this problem is

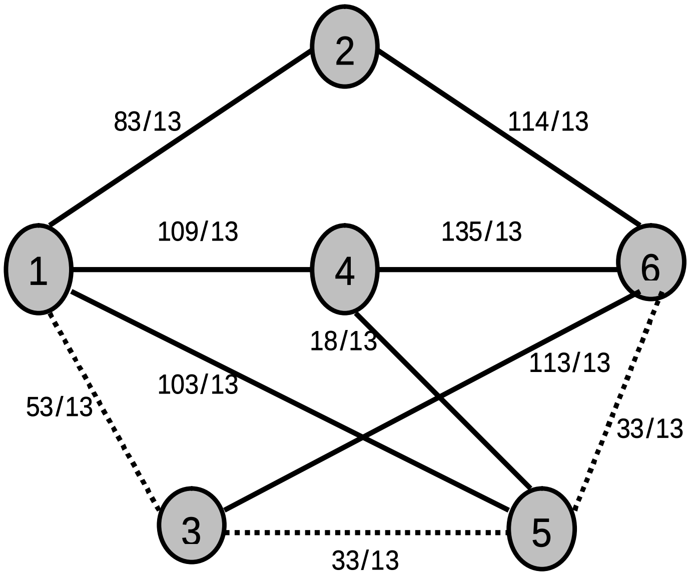
1®4®5®6, it has a delay of 10 and a cost of 6.When the basic algorithm DCLC is used to find a solution, it first finds the LC path and puts it in q. Since the LC path is infeasible, it continues to find the LD path and puts it in p. Since the LD path is feasible, it proceeds to compute the parameter a , hence a =5/13. By which a mixed weight is obtained for each link, as shown in Figure3. The shortest path from node 1 to Node 6, which is also the final solution of algorithm DCLC.

Hence the path is 1®3®5®6. This solution, which has a delay of 3 and a cost of 16, is not optimal, however it is better than the LD path.



Final a path from 1 to 6 with delay bound =12

Figure 2: A DCLC problem: Find the optimal path from node 1 to node 6 with a delay upper bound of 12

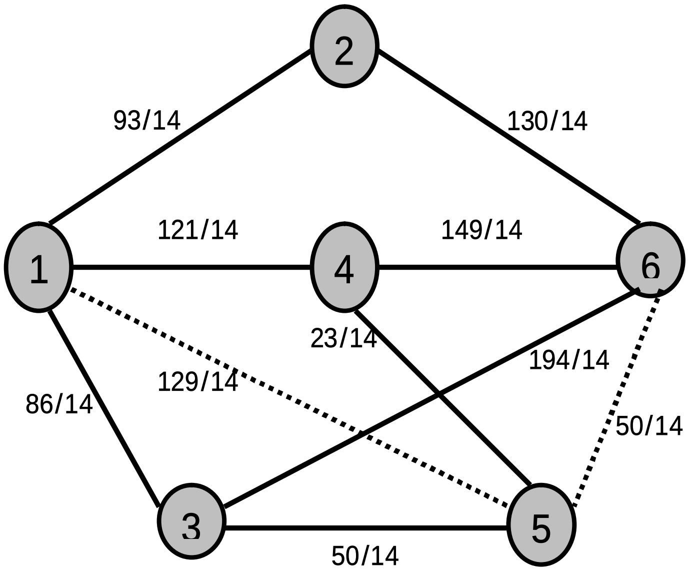


Shortest 1-3-5-6-: delay =3, cost=16

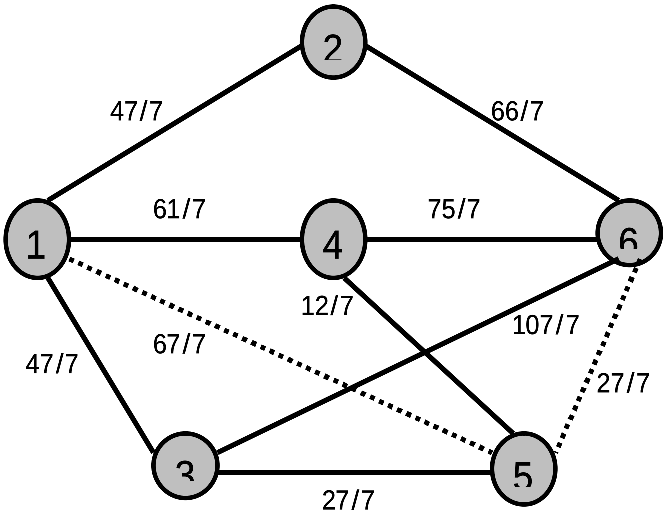
Figure 3: Solution found by algorithm DCLC (This is also the solution obtained by algorithm DCLC-A after the first iteration)

5.3.2 Analysis and discussion the DCLC-A Algorithm:

The first iterative algorithm DCLC-A is based on algorithm DCLC. After finding the LC path q and the LD path p, the algorithm enters the iterative procedure. In each iteration, a parameter a is first computed for constructing the mixed weight, and then the corresponding shortest path r is found. If r has a lower cost than p, then p is replaced by r. In the first iteration, the parameter is the same as the one used in DCLC, and correspondingly path r={1 ® 3 ® 5 ®6} is returned. Since r is better than p, p is replaced by r. In the second iteration, the parameter is reevaluated by a= (12-3)/ (16-2)= 9/14. The corresponding mixed weight is shown in Figure 4(a), and the shortest path r={1 ® 5 ®6} is returned. Again, p is replaced by r. In the third iteration the parameter becomes a=(12-7)/(9-2)=5/7, and the corresponding mixed weight is shown in Figure 4(b). Since the returned shortest path is the same as the one obtained in the previous iteration, this algorithm stops. The final solution is 1®5®6, which has a delay of 7 and a cost of 9. It is better than the solution of algorithm DCLC, but it is still not optimal.



Shortest 1-5-6: delay=7, cost=9
(a)Solution obtained after the second iteration



Shortest path 1-5-6: delay=7, cost=9
(b)Solution obtained after the third iteration

Figure 4 :Solving procedure of algorithm DCLC-A

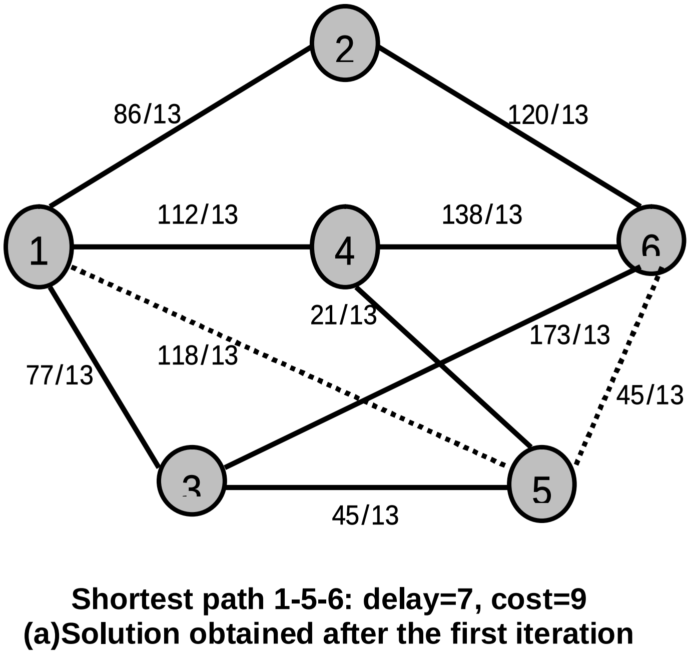
5.3.3 Analysis and discussion the DCLC-B Algorithm:

When the second iterative algorithm DCLC-B is used to solve this problem, will be found first the LC path q and LD path p. In each of subsequent iteration, the parameter a is computed, and either p or q is replaced by the corresponding shortest path r, if r is better. The parameter in the first iteration can be computed by a=(18-2)/(28-2)=8/13. As shown in Figure 5(a), the corresponding shortest path is r={1 ® 5 ®6}. Since r is feasible and better than p,p is replaced by r.

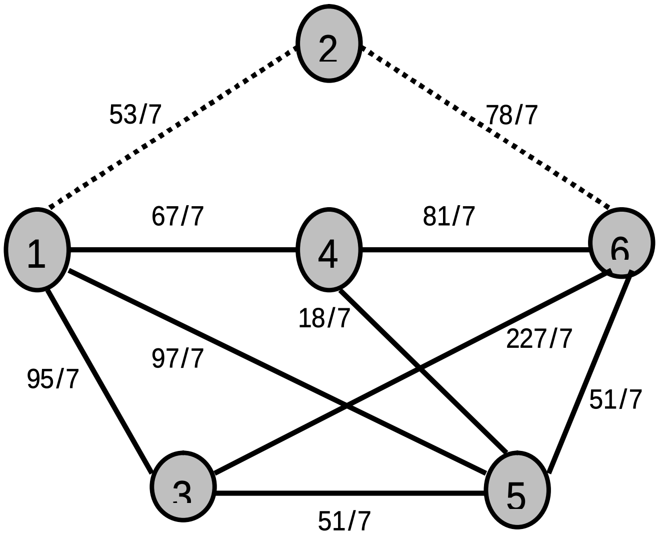
In the second iteration, the parameter becomes a=(18-7)/(9-2)=11/7. As shown in Figure 5(b), the corresponding shortest path is r={1®2®6}.

Although r is infeasible, it is better than q. Thus, q is replaced by r. In the third iteration, the parameter becomes a=(14-7)/(9-3)= 7/6. As shown in Figure 6(a), the corresponding shortest path is r={1®4®5®6). Again, since r is feasible and better than p , p is replaced by r .In fourth iteration , the parameter becomes a=(14-10)/(9-6)=4/3. As shown in Figure 6(b), the two paths {1®2®6} and{1®4®5®6}have the same minimum length. The returning of any of them will terminate the algorithm. As a result , the last feasible path

{1®4®5®6} will be returned as the final solution , and as mentioned earlier in this section , this solution will be optimal.

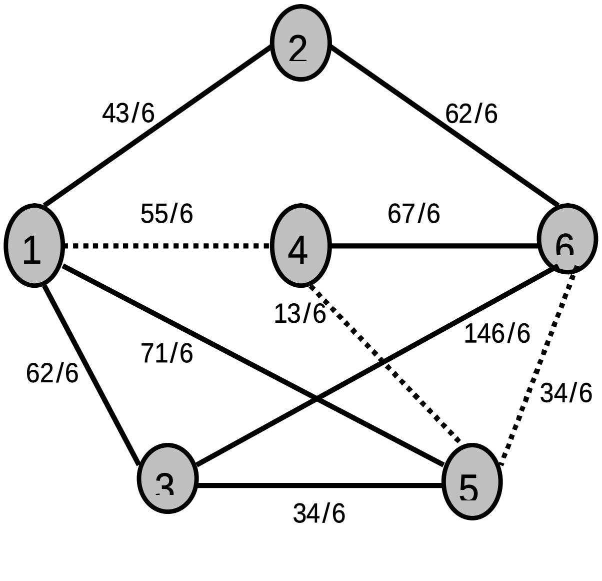


Shortest path 1-5-6: delay=7, cost=9
(a)Solution obtained after the first iteration

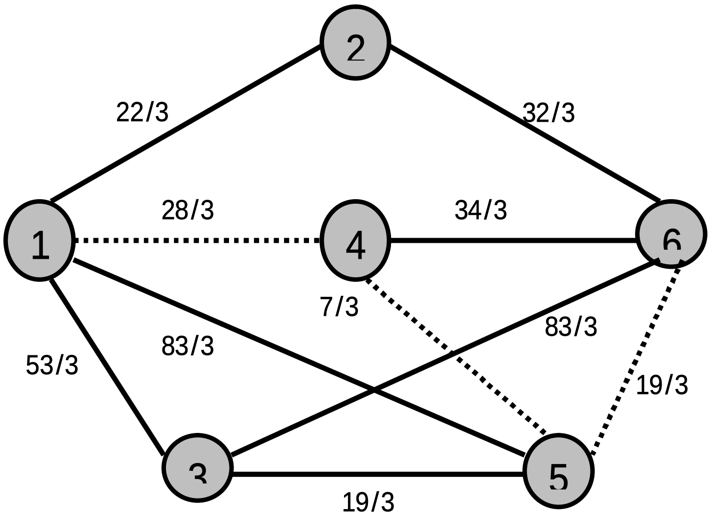


Shortest path 1-2-6: delay=14, cost=3
(b)Solution obtained after the second iteration

Figure 5 : Solving procedure of algorithm DCLC-B



Shortest path 1-4-5-6: delay=10, cost=6
(a)Solution obtained after the third iteration



Shortest path 1-2-6or 1-4-5-6
(b)Solution obtained after the fourth iteration

Figure 6 : Solving procedure of algorithm DCLC-B

Finally Table 1 gives the results obtained from above demonstration example.

Table 1: results of Demonstration example

Algorithm	S	d	C_d	Path	Cost	Delay
DCLC	1	6	12	1 – 3 -5 - 6	16	3
DCLC-A	1	6	12	1- 5 - 6	9	7
DCLC-B	1	6	12	1 – 4 - 5 - 6	6	10

6. Conclusion:

The need for optimal path among the multiple feasible paths, the optimaztion problem is appeared, so two iterative algorithms were proposed to achieve this goal, DCLC-A and DCLC-B then the second algorithm (DCLC-B) appears the best solutions for optimization problem (less delay and suitable delay).

1. With high speed networks and increasing requirements of multimedia applications such video, voice, and text, the need for solving the problem of QoS routing become a necessary insistent.

2. QoS Routing is not only selecting a path for transmitting data from source to destination, but doing this to satisfy constraints or to optimize the user requirements (delay as example).

3. QoS routing and resource reservation are two important, closely related network components. In order to provide the guaranteed services, the required resources (bandwidth) must be reserved when a QoS connection is established.

4. The need for optimal path among the multiple feasible paths, the optimaztion problem is appeared, so two iterative algorithms were proposed to achieve this goal, DCLC-A and DCLC-B then the second algorithm (DCLC-B) appears the best solutions for optimization problem (less delay and suitable delay).

7.References:

[1] Cisco, “Quality of Service Networking”,

http://www.cisco.com/univered/cc/td/doc/cisintwk/ito-doc/QoS.html

[2] A.S. Tanenbaum, “Computer Networks”, Third edition, Prentice-Hall, 2000.

[3] Behrouz Forouzan,“Introduction to Data Communications and Networking”, Prentice-Hall, 2004.

[4] J. Wang, W. Wang, J. Chen, and S. Chen, “A randomized QoS routing algorithm on networks with inaccurate link-state information,” Proceeding of WCC-ICCT 2004, vol. 2, pp 1617-1622, 2004.

[5] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," Proceedings of the IEEE INFOCOM 2004 Conference, pp. 834-843, April 2004.

[6] A. Shaikh, J. Rexford, and K.G. Shin, “Evaluating the impact of stale link state on quality-of service routing,” IEEE/ACM Transactions on Networking, vol. 9, no. 2, pp 162-176, April, 2002

ايجاد الطريق المثالي لانجاز محددات الطريق والموجهات الامثل له

م.م. حسن حسين حمودي

كلية المنصور الجامعة

المستخلص:

شبكات الحاسبات الحديثة اصبحت بأزدياد تستخدم لأرسال الوسائط المتعددة مثل الفيديو والصوت بجانب دعمها تطبيقات اتصالات البيانات الحالية ,هذه تطبيقات الوسائط المتعددة تتطلب بقوة تحديدات جودة الخدمة ومع ان ازدحام الشبكة يأخذ اشكال مختلفة حيث ان كل شكل له متطلباته الخاصة من عرض الحزمة ,زمن التأخير,مقدار الخسارة والتعددية. ومع النمو الهائل للانترنيت معظم ازدحامات الشبكة بدأت باتجاه استخدام تقنيات تحديد المسار من نوع QoS

في هذا البحث النظرة العامة عن مشاكل تحديد المسار بأستخدام ال QoSتم وصفها من خلال توضيح نقاط القوة والضعف في تقنيات تحديد المسار لهذه المشاكل.

من نتائج دراسة النقاط اعلاه تم ملاحظة مشكلة مهمة تتمثل بتحديد المسار المثالي لذلك تم اقتراح خوارزميتين لحل هذه المشكلة هما: خوارزمية اقل كلفة بتحديد زمن التأخير/الاختبار الاول DCLC-A و خوارزمية اقل كلفة بتحديد زمن التأخير/الاختبار الثاني DCLC-B من خلال النتائج العملية للخوارزميتين وجد انه الخوارزمية الثانية تعطي الحل الامثل لتحديد المسار بين عقدتين.