

Delay-Constrained Survivable Multicast Routing Problem on WDM Networks

Der-Rong Din and Jhong-Yan Jiang

Department of Computer Science and Information Engineering,
National Changhua University of Education, Changhua, Taiwan, R.O.C.

E-mail:deron@cc.ncue.edu.tw, m96612007@seed.net.tw

Tel: 886-4-7232105-7047 FAX: 886-4-7211284

Abstract—In WDM network, a link failure may cause service disruption and may lead to significant information loss. Especially, when the failed link is on a light-tree that carries traffic to multiple destinations, the traffic to all the downstream destinations along the failed link will be affected. Moreover, for a multicast request, the delay bound associated with it can be decided according to the emergence degree or priority of the data. Transmitted data with delay bound reflects the realistic demand in the future. For a given multicast request with delay bound, a delay-constrained survivable multicast protection provides the primary tree and a set of spare resources; when the single-link failure occurs, the pre-configured spare resources can be used to construct a new multicast tree under the delay constraint.

In this paper, given a multicast request with delay constraint, the goal is to allocate the delay-constrained primary multicast tree and protecting span p -cycles, when the single-link failure occurs, the multicast tree can be recovered by using protecting p -cycles such that the backup tree can also satisfy the delay constraint. This problem is defined as the *Delay-Constrained Survivable Multicast Routing Problem (DCSMRP)*. The span p -cycles protection scheme is studied in this paper and a heuristic method *Delay Constrained Span p -cycles Protection (DCSP)* is proposed to solve it.¹

Keywords WDM, survivability, delay-constrained, multicast routing, p -cycles

I. INTRODUCTION

There is a growing consensus that the next generation Internet will employ an IP-over-WDM (wavelength-division-multiplexing) architecture. In order to provide various applications on WDM networks, mechanisms must be developed to handle not only point-to-point communications (or *unicast*) but also *multicast*. Multicast is the transmission of information from one source to multiple destinations simultaneously. Thus, issues concerning supporting multicast on WDM networks need to be studied [1]. For a comprehensive survey, the reader may refer to [1].

In WDM networks, a fiber can provide tremendous bandwidth, so a network failure (such as a fiber cut or node failure) can lead to a lot of light-paths failed and may cause enormous data loss. Therefore, several schemes have been proposed to achieve *survivability* by *protection* or *restoration* in the optical layer.

To guarantee multicast traffic signals can be transmitted efficiently, transmission delay from the source to all destinations should be limited under a given delay bound. The delay bound of the multicast can be decided according to the emergence degree or priority of the data. For the WDM network with sparse light splitting and without wavelength conversion, several researchers [2] have studied the delay-constrained multicast routing problem for minimizing the number of fibers used or cost. The delay-constrained multicast routing problem in WDM networks is studied in [3], [4]. The goal is to minimize a linear combination of communication cost and wavelength consumption under delay constraint.

The *light-tree* [1] spanning all multicast members is used to establish point-to-multipoint connection for supporting multicast traffic. The fiber cut may cause service disruption and also may lead to heavy loss of significant information. For the failed link on a light-tree that carries traffic to multiple destinations, the traffic to all the downstream destinations along the failed link will be affected. The closer the link is to the source node, the more destinations will be affected [5].

Multicast protection schemes on WDM networks can be classified into five major protecting schemes: (1) tree-based protection [6]–[8]; (2) ring-based protection [9], [10]; (3) path-based protection [11]–[13]; (4) segment-based protection [13], [14]; (5) cycle-based protection [5]. The most common method to protect light-trees for optical multicasting is to deploy a pair of link-disjoint or arc-disjoint trees (LDTs/ADTs) [7]. For the ring-based multicast protection approaches, Leelarumee *et al* [10] proposed two methods: one ring for one multicast session (OFO) and one ring for all multicast sessions (OFA). For the optimal path-pair protection (OPP) approach, Singhal *et al* [13] reported an integer linear programming (ILP) model for protection of static multicast sessions. Moreover, Singhal *et al* also proposed OPP based-shared disjoint path (OPP-SDP) algorithm. In the OPP approach, a survivable multicast session is formed by all the link-disjoint primary and backup paths pairs from the source to every destination.

For unicast traffic protection, the pre-configuration protecting cycle (*span p -cycles* or *p -cycles*) [15] is a promising approach for protecting working capacity in WDM networks. The *p -cycle* provides two protection paths to recover the straddling span and a protection path to recover the on-cycle span [15]. Moreover, it can achieve ring-like recovery speed

¹This work was partly supported by the National Science Council (NSC) of Taiwan, R. O. C. under Grant Number NSC-98-2221-E-018-009.

while maintaining the capacity efficiency of a mesh restorable network.

Recently, the goal of the studied multicast protection problems is to provide backup paths (or tree) to transmit the request. To the best of our knowledge, no related article takes the delay bound of multicast request into consideration. Given a delay-constrained multicast request, if a link on the multicast tree fails, the delay constraint of the backup paths (or tree) for the multicast request may not be satisfied. The problem is to find a multicast tree and a link-disjoint backup tree (paths or p-cycles) for a delay-constrained multicast request. When a link failure occurs, the pre-configured backup tree (paths or cycles) can be used to construct a new tree which satisfies the delay constraint.

In this paper, for a given network and a multicast request with delay bound, the goal is to find a primary multicast tree and the protecting p-cycles such that not only the primary multicast tree can satisfy the delay constraint, after recovering the multicast tree from single-link failure, the new tree can also satisfy the delay constraint. This problem is denoted as the *Delay-Constrained Survivable Multicast Routing Problem (DCSMRP)*. In this paper, the span p-cycles protection scheme used as the protecting scheme and the *Delay Constrained Span p-cycles Protection (DCSP)* algorithm is proposed to solve it.

II. PROBLEM DEFINITION

A. Notations

- $G(V, E)$: the physical topology of the network, where V is the set of nodes and E is the set of links of G , respectively.
- d_e : the propagation delay of link $e \in E$.
- $r = \{s, D, \Delta\}$: the multicast request, where $s \in V$ is the source node, $D = \{d_1, d_2, \dots, d_m\}$ is the set of destinations, m is the size of multicast request, $m = |D| \leq |V|$, and Δ is the delay constraint of the request.
- T_r : the primary multicast tree for the multicast request r .
- K : a fixed positive integer.
- δ : the set of destinations which are sorted according to the delay of the destinations on T_r in decreasing order.
- $P_{T_r}(s, d_i)$: the path from s to d_i on T_r , $d_i \in D$ for the multicast request r .
- EPC : the set of existing span p-cycles on network.
- $EPC(e)$: the set of the span p-cycles in EPC ($EPC(e) \subseteq EPC$), which can be used to protect the link $e \in E$ on either on-cycle or straddling way.
- $\overline{EPC}(e)$: $\overline{EPC}(e) = EPC \setminus EPC(e)$, $e \in E$.

B. Problem Definition

For the multicast request r , after finding the primary multicast tree T_r , the delay of destination d_i is the sum of delay of links on path P_{T_r} . That is, $delay(d_i) = \sum_{e \in P_{T_r}(s, d_i)} d_e, \forall d_i \in D$. The delay of the multicast tree, $delay(T_r)$, is the maximum delay of all destinations and should be less than or equal to Δ . That is, $delay(T_r) = \max_{\forall d_i \in D} \{delay(d_i)\} \leq \Delta$.

In DCSMRP, given a WDM network $G(V, E)$ and a multicast request $r(s, D, \Delta)$, the goal is to find the primary multicast tree and the set of protecting cycles such that after recovering the multicast from the single-link failure, the delay of multicast tree can satisfy the delay constraint.

C. Assumptions

For the DCSMRP the following assumptions are given :

- For each link in G , there is a fiber which connects the end-nodes such that data can be transmitted bidirectional.
- The number of wavelengths of links are equal.
- All nodes in network are with full light-splitting and wavelength-converting capabilities.
- Only the single-fiber failure scenario is considered.
- The physical topology is two-connected.
- The number of transmitters/receivers of each node is fixed and known.
- Multicast requests are serialized such that a request is processed at a time.

III. PROPOSED ALGORITHMS FOR MULTICAST ROUTING

In this section, a method DCSP consisting of three heuristic algorithms are proposed to solve the DCSMRP. The proposed heuristic algorithms are Delay Constrained Span p-cycles Protection Algorithm (DCSPPA), Extending Delay Constrained p-Cycle Algorithm (EDCPCA), and Shortest Delay Path Cycle Finding Algorithm (SDPCFA). The details of the proposed algorithms are described in the following subsections.

A. DCSPPA

In this subsection, the Delay Constrained Span p-cycles Protection Algorithm (DCSPPA) is described. We assume that a set (EPC) of span p-cycles is already constructed and used to protect the primary multicast tree of multicast request. For the arrived multicast request, the goal is to find a primary multicast tree (T_r) and a set of protecting cycles to protect the multicast. That is, for each edge of the primary multicast tree, there is a p-cycle in either on-cycle or straddling manner to protect the edge.

In general, a multicast traffic is unlikely to be symmetric in both directions between any two nodes. This implies that the number of working and protection wavelengths is not likely to be the same in both directions on a link. Therefore, without loss of generality, all trees and p-cycles considered in this paper are *unidirectional*, i.e., they are directed. The existing cycle can be used if there are enough capacities and can satisfy the delay constraint, otherwise a new protecting cycle is found to protect the edge. The Extending Delay Constrained p-Cycle Algorithm (EDCPCA) is proposed to find a cycle by extending existing cycles. If performing EDCPCA fails to find a cycle, then the Shortest Delay Path Cycle Finding Algorithm (SDPCFA) is used to find a whole new one.

Since the delay of the cycle and the path may affect the delay of the new multicast tree, to recover multicast tree such that the delay constraint can be satisfied, these delays should be talked into consideration. The delay of a p-cycle C is the

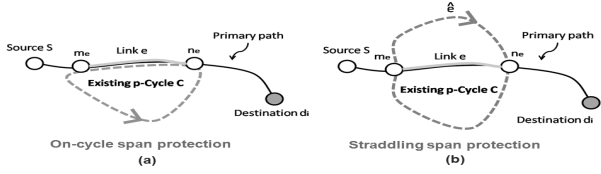


Fig. 1. Delay of backup path (a) on-cycle span, (b) straddling span.

sum of the delays of all edges and denoted as $delay(C) = \sum_{e \in C} d_e$. If the edge is an on-cycle span, as the case showed in Fig.1(a). The counter-clockwise cycle can be used to protect the edge under the delay constraint if the formula (1) can be satisfied.

$$delay(C) + delay(d_i) - 2 \times d_e \leq \Delta, e \in T_r \quad (1)$$

It is worth noting that, not all cycles can be used to protect the on-cycle span due to the cycle is unidirectional.

If the edge is a straddling span of the protecting cycle, as the illustration showed in Fig.1(b). The cycle can be used to protect the edge under the delay constraint if the formula (2) can be satisfied,

$$delay(\hat{e}) + delay(d_i) - d_e \leq \Delta, e \in T_r, \quad (2)$$

where $delay(\hat{e})$ is the delay of the default backup path on cycle.

First, let the initial value of the integral variable k be zero, the Dijkstra-based least-delay tree algorithm (LDTA) is used to find the primary multicast tree T_r^0 . If delay-constrained tree T_r^0 cannot be found, the request is blocked and return.

At the k^{th} iteration, for the current primary multicast tree T_r^k ($0 \leq k \leq K$), if the protecting cycles cannot be found for all edges in T_r^k , then another primary tree T_r^{k+1} whose delay is slightly larger than current one is used as the new primary multicast tree. Then another set of protecting cycles are found to protect all edges on new tree T_r^{k+1} . To find a new multicast tree, the k^{th} ($1 \leq k \leq K \leq n-1$) largest delay edge of the primary tree T_r^0 is removed from the topology G and form G^k , then LDTA is performed to find a multicast tree T_r^k under delay constraint on G^k . If T_r^k is found, then find a set cycles to protect all edges on tree T_r^k . If all edges can be protected by the set of cycles under delay constraint then return. Otherwise, if $k < K$, increase k by one and repeat the process to find a new tree T_r^k and protecting cycles. If trees T_r^k and cycles cannot be found in all the cases $0 \leq k \leq K$, the multicast request is blocked. The details of the DCSPPA is known as follows.

Delay Constrained Span p-Cycles Protection Algorithm (DCSPPA)

Input: $G = (V, E)$, $r = (s, D, \Delta)$, K

Output: T_r, EPC ;

Let $k = 0$, $T_r^k = \emptyset$, for $k = 0, 1, \dots, K$.

Perform LDTA to find the multicast tree T_r^0 on G .

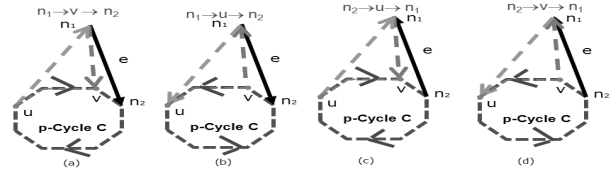


Fig. 2. EDCPC example (a) case 1, (b) case 2 (c) case 3, (d) case 4.

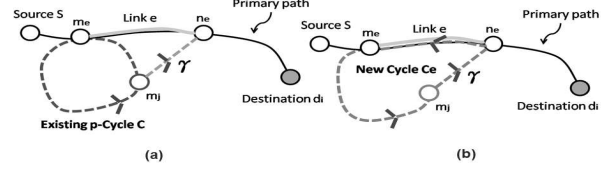


Fig. 3. EDCPC algorithm illustration (a) before, (b) after.

```

While (k ≤ K)
{
  if (k > 0) {
    Remove the link with the kth largest delay on Tr0 from G and form Gk.
    Perform LDTA to find the multicast tree Trk on Gk.
  }
  If (Trk can't be found) or (delay(Trk) > Δ)
  Block the multicast request and return.
  else
  {
    Temporarily allocate the working capacities of Trk.
    Sort destinations in δ in a decreasing order according to the delay of nodes.
    While (δ ≠ ∅)
    {
      Select a destination di, di ∈ δ.
      Let Ei be the set of edges on path PTrk(s, di) and not protected by cycles.
      found = FALSE.
      While (Ei ≠ ∅)
      {
        Select a link e = (me, ne) ∈ Ei.
        Find the set EPC(e) and EPC(e).
        While (EPC(e) ≠ ∅) && (found == FALSE)
        {
          Select a cycle c ∈ EPC(e).
          If cycle c satisfies equation (1) or (2).
          Add cycle c to PC, found = TRUE, and remove link e from Ei.
        }
        else
        Remove cycle c from EPC(e).
      }
      //end-of-while
      If (found == FALSE)
      {
        Perform EDCPCA to expand cycles in EPC(e) to protect e.
        If found (denoted as Ce)
        {
          Temporarily allocate the working capacity of Ce.
          Add Ce to PC.
          Remove link e from Ei.
          found = TRUE.
        }
        else
        {
          Perform SDPCFA to find a new cycle Cnew to protect e.
          If found
          {
            Temporarily allocate the working capacity of Cnew.
            Add Cnew to PC.
            Remove link e from Ei.
            found = TRUE.
          }
          else //not found
          {
            If (k < K)
            {
              Recover the kth largest delay link on Tr0 from Gk.
              k = k + 1.
            }
            else
            Release the resources and block the request.
          }
        }
      }
    }
    Check other edge e' (≠ e) ∈ Ei can be protected by the cycle PC
    and satisfy the delay constraint. If yes, remove e' from Ei.
  }
  Remove di from δ, union PC to EPC, and PC = ∅.
}
//end-of-while
Allocate the capacity of Trk and PC, return Trk and EPC.
}
//end-of-while

```

B. EDCPCA

Four cases are considered in performing the EDCPCA. In these cases (see Fig. 2), for a directed edge $e = (n_1, n_2)$, a cycle $c \in \overline{EPC}(e)$ which passes through end node n_1 (or n_2) is selected and to be extended to protect edge (n_1, n_2) . According to the relation between e and the selected cycle c , the LDTA is performed to find two link-disjoint paths from n_1 to u and v to n_1 ($u \neq v$), (nodes u and v are on the cycle c). Then, the path between u and v of the cycle is removed and paths u to n_1 , n_1 to v , and cycle c are joined to form a protecting cycle for e .

The details of the EDCPCA is showed as follows and the example is shown in Fig. 3.

Extending Delay Constrained p-Cycle Algorithm (ED-CPCA)

Input: $\overline{EPC}(e)$, $e = (m_e, n_e)$
Output: C_e
 $E' = \overline{EPC}(e)$, $C_e = \emptyset$.
while ($E' \neq \emptyset$)
{
 Select a cycle c in E' such that one end node of edge e (say m_e) is on-cycle.
 Find the on-cycle node m_j ($\neq m_e$) and the path $\gamma = P(m_j, n_e)$ with the minimal delay $\text{delay}(m_e m_j) + \text{delay}(P(m_j, n_e))$, where $m_e m_j$ is the minimal delay arc between m_e and m_j on cycle c .
 Form the protecting cycle C_e by combine arc $m_e m_j$, path γ , and arc (n_e, m_e) .
 If ($\text{delay}(C_e) + \text{delay}(d_i) - 2 \times d_e \leq \Delta$)
 then return C_e ;
 else remove c from E' .
}

C. SDPCFA

If performing EDCPCA fails to extend existing cycles to protect the edge e , the Shortest Delay Path Cycle Finding Algorithm (SDPCFA) is performed to find a new cycle under delay constraint to protect edge e in on-cycle manner. The details of the SDPCFA is shown as follows and the example of SDPCFA is shown in Fig. 4

Shortest Delay Path Cycle Finding Algorithm (SDPCFA)

Input: G_r , $e = (m_e, n_e)$
Output: C_{new}
Construct auxiliary graph G_r^e by removing e from G_r .
Use Shortest Delay Path algorithm to find backup path $B_{G_r^e}(m_e, n_e)$ for link e on graph G_r^e , where $\text{delay}(B_{G_r^e}(m_e, n_e)) + \text{delay}(d_i) - d_e \leq \Delta$.
If $B_{G_r^e}(m_e, n_e)$ is found, join arc (n_e, m_e) and path $B_{G_r^e}(m_e, n_e)$ to form a new cycle C_{new} .
return C_{new} .
else return false.

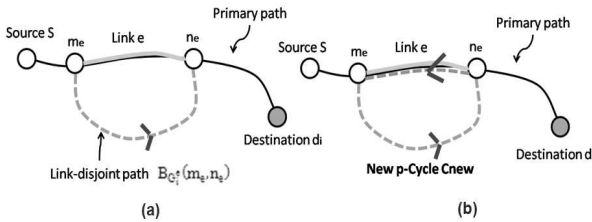


Fig. 4. SDPCFA illustration (a)before, (b) after.

IV. EXPERIMENTAL RESULTS

To know the efficiency of the proposed algorithms, three criteria are used and evaluated through experiments. They are blocking ratio (BR), resource utility ratio (RUR), and wavelength efficient ratio (WER). BR is the probability that the

primary multicast tree and protecting cycles cannot be found with delay constraint. RUR is the ratio of the spare capacity to that of the working capacity. WER is the ratio of the working capacity to total (working and spare) capacity.

The USNET topology (as shown in Fig.5(a)) with 24 nodes and 43 links was used for simulations. The delay of link in USNET is proportional to the length of the link. The proposed algorithms are conducted by C++ language together with the BGL (Boost Graph Library) [16]. All simulations were run on a PC with Core duo 1.66GHz CPU, 1.5GB RAM, and Windows XP operating system. In each simulation, multicast requests are generated randomly, the arrival rate of requests are fitted with the Poisson distribution with mean 3. The delay bound of multicast request is set between 5.0 and 9.0, and the value of K is in $\{0, 5, 8, 10\}$. For each multicast request, the size of D is randomly generated and within range 2–11 and the source node and the destinations are randomly selected.

A. Efficiency of DCSP

First, RUR and BR are evaluated for cases that with R ($= 30$ or 40) multicast requests, W ($=16$ or 32) wavelengths, and for different values of the Δ ; the results are shown in Fig.6. The result in Fig.6(a) shows that the value of RUR increases as the value of Δ decreases. This is may be the reason that the tighter of delay-constraint, the harder existing p-cycles can be used to protect the multicast requests. Thus, new p-cycles with smaller delay should be found and this may increase the RUR value. This appears on the cases for different numbers of multicast requests, the RUR value of $R = 40$ is better than that of $R = 30$. The RUR related to the different numbers of wavelengths provided by the WDM network is unapparent.

The result in Fig.6(b) shows the value of BR increases as the value of Δ decreases. The BR value of case $R = 40$ is worse than that of the case $R = 30$ for $\Delta < 7.0$. But for $\Delta > 7.0$, the BR values of two cases are very close. The BR value of $W = 16$ is worse than that of $W = 32$.

The relations of BR value, computational time, and the number of multicasts are shown in Fig.7. The value of BR decreases as the number of multicast requests increases (as shown in Fig.7(a)). This may the reason that the span p-cycles reserved by previous multicast requests can be shared and used to protect some new follow-up multicasts. Fig.7(a) also shows the results for different values of K , BR value can be reduced as K decreases. The difference of BR values of cases $K = 0$ and $K = 10$ is about 10%. Fig.7(b) shows that the computational time is proportional to the number of requests. Moreover, the computational time also increases as K increases.

B. Comparison

Two methods proposed in [17] were used for comparisons. They are DCLTP (Delay-constrained Link-disjoint Trees Protection) and DCDTP (Delay-constrained Dual-tree Protection). For the DCLTP, link-disjoint primary and backup trees with delay-constraint are found and used to protect the multicast

request. For the DCDTP, a primary multicast tree and a bi-directional path which connected source to all destinations on the leaves of primary multicast tree are used to protect the multicast request.

The results (in Fig.5(b)) show that the BR value of DCDTP rapidly increases from 25% to 69% as the number of requests increases, and is worse than that of DCSP (30%–40%). Thus BR value provided by the DCSP method is comparatively stable and better than that of the DCLTP. The computational complexity of the DCLTP, DCDTP, and DCSP is $O(K|V|^3)$, $O((K + |D|)|V|^3)$, and $O((K + |D| + 2)|V|^3)$, respectively. The result in Fig.8(a) shows that RUR value increases as the Δ decreases for $R = 40$. The RUR and WER of DCSP values are better than that of DCLTP and DCDTP in the cases of $\Delta < 7.0$ (as shown in Fig.8).

V. CONCLUSIONS

In this paper, we define and study the *Delay-Constrained Survivable Multicast Routing Problem (DCSMRP)*. The span p -cycles protection scheme is used and a heuristic algorithm *Delay Constrained Span p -cycles Protection (DCSP)* method is proposed to solve it. Simulations show that the value of BR of DCSP is worse than DCDTP but better than DCLTP when network with enough wavelength resources.

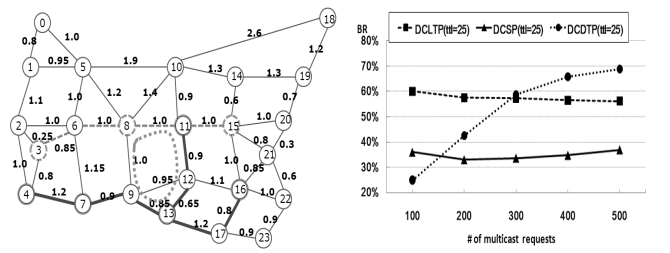


Fig. 5. (a) USNET, (b) comparison of BR for DCLTP, DCDTP and DCSP with TTL=25.

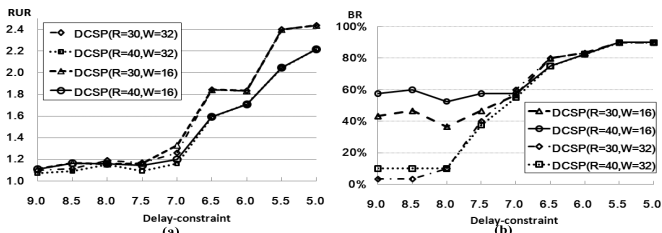


Fig. 6. Simulation results (a) RUR, (b) BR.

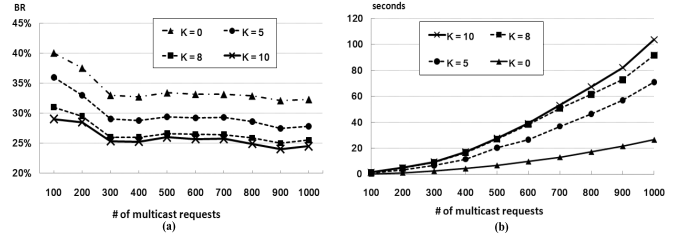


Fig. 7. Simulation results (a) K , (b) CPU time.

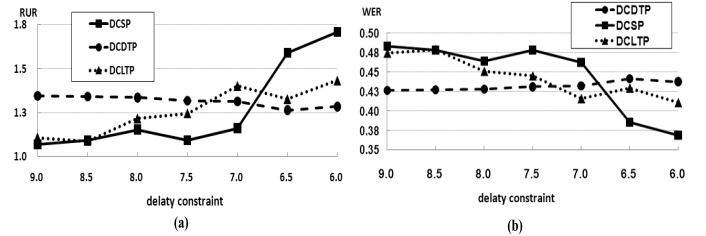


Fig. 8. Comparison DCLTP, DCDTP and DCSP (a)RUR, (b) WER

REFERENCES

- [1] Biswanath Mukherjee, Optical WDM Networks, Springer press, 2006.
- [2] A. Yan, M. Ali, J. Deogun, "Routing optimization of multicast sessions in sparse light-splitting optical networks," in Proc. of IEEE GLOBE-COM 2001, vol.4, pp.2134–2138.
- [3] DR Din, "Heuristic algorithms for finding light-forest of multicast routing on WDM network," *JISE*, vol. 25, no.1, pp.83–103, 2009.
- [4] DR Din, "Genetic algorithm for finding minimal cost light-forest of multicast routing on WDM network," *Artificial Intelligence Review*.
- [5] F. Zhang, WD Zhong, and Y. Jin, "Optimizations of p -cycle-based protection of optical multicast sessions," *J. Lightw. Technol.*, vol. 26, no. 19, pp.3298–3306, 2008.
- [6] N.K. Singhal, C. Ou, and B. Mukherjee, "Cross-sharing vs. self-sharing trees for protecting multicast sessions in mesh networks," *Comput. Netw.*, vol. 50, no. 2, pp.200–206, 2006.
- [7] N. K. Singhal and B. Mukherjee, "Protecting multicast sessions in WDM optical mesh networks," *J. Lightw. Tech.*, 21(4), pp.884–892, 2003.
- [8] M. Medard, S. G. Finn, R. A. Barry, and R. G. Gallager, "Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs," *IEEE/ACM Trans. Netw.*, 7(5), pp.641–652, 1999.
- [9] T. Rahman and G. Ellinas, "Protection of multicast sessions in WDM mesh optical networks," in Proc. of OFC'05, Mar. 2005, p. 3.
- [10] P. Leelarumsee, C. Boworntummarat, and L. Wuttisittikulkiij, "Design and analysis of five protection schemes for preplanned recovery in multicast WDM networks," in Proc. of IEEE SAWWC'04, Apr.2004, pp.167–170.
- [11] H. B. Luo, H. F. Yu, L. M. Li, and S. Wang, "On protecting dynamic multicast sessions in survivable mesh WDM networks," in Proc. of OFC'06, Anaheim, CA, Mar. 2006, p. 3.
- [12] N. K. Singhal and B. Mukherjee, "Dynamic provisioning of survivable multicast sessions in optical WDM mesh networks," in Proc. of Optical Fiber Communications Conf., Mar. 2003, pp.207–209.
- [13] N. K. Singhal, L. H. Sahasrabudde, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks," *J. Lightw. Technol.*, 21(11), pp.2587–2594, 2003.
- [14] C. Lu, H. Luo, S. Wang, and L. M. Li, "A novel shared segment protection algorithm for multicast sessions in mesh WDM networks," *ETRI J.*, 28, pp.329–336, 2006.
- [15] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: Ring-like speed with mesh-like capacity for self-planning network restoration," in Proc. of ICC'98, Nov. 1998, vol. 1, pp. 537–543.
- [16] http://www.boost.org/doc/libs/1_41_0/libs/graph/doc/index.html
- [17] JY Jiang, A study of Delay-constrained survivable multicast routing on WDM networks, Mstrer Thesis, 2009, National Changhua University of Education.