Multicast Routing under Delay Constraint in WDM Network with Different Light Splitting

Ming-Tsung Chen [1], Shian-Shyong Tseng [2]

[1] Department of Computer and Information Science

National Chiao-Tung University

Hsinchu 300, Taiwan R.O.C.

mtchen.cis88g@nctu.edu.tw

[2] Department of Computer and Information Science

National Chiao-Tung University

Hsinchu 300, Taiwan R.O.C.

sstseng@cis.nctu.edu.tw

Abstract

Because the optical *WDM* network will become a real choice to build up backbone in the future, multicast communications on the *WDM* network should be supported in communication model for various network applications. In this paper, we define a new multicast problem that is routing a request with delay bound to all destinations in *WDM* network with *different light splitting* and propose a new formulation to solve the problem, where the *different light splitting* means that nodes in the network can transmit one copy or multiple copies to other nodes by using same wavelength. The new problem can be reduced to *Minimal Steiner Tree Problem* (*MSTP*) which belongs to NP-Complete problem, and can be solved by an efficient three-phase (*Pre-Processing Phase*, *Generating Phase*, and *Refining Phase*) solution model with *Backward Stepwise Sub-path Replacing* (*BSSR*) and *Most Cost-Difference First Progressive Replacing* (*MCDFPR*) heuristics to find a feasible light-forest in polynomial time. Finally, experimental results show that our solution model can obtain a near optimal solution.

1. Introduction

Optical networks [1] are high-capacity telecommunications networks, based on optical technologies and optical components, which provide routing, grooming, and restoration at the wavelength level as well as wavelength-services. The technology of *WDM* (*Wavelength Division Multiplexing*) network [8], based on optical wavelength-division multiplexing on an optical fiber to form multi-communication channels at different wavelengths, provides connectivity among optical components to make optical communication to meet the increasing demands for high channel bandwidth and low communication delay. To transmit data between source and destination in *WDM* network, a *light-path* which connects two nodes should be established.

To support multicast communication in *WDM* network, nodes in *WDM* network may have the *light splitting capacity* which is used to split an optical signal of input port to multiple signals of output ports without electrical conversions, and a *routing-tree* used to transmit a request could be constructed. The *light-tree* [2] is a special routing-tree by configuring nodes in the physical topology and occupies the same wavelength in tree links. Each branch node of the light-tree is an optical switch had *light splitting capacity*. The nodes with the light splitting capacity named as *MC* (multicast capable) node are usually more expensive to build than those without named as *MI* (multicast incapable) node due to its complexity architecture [1].

Furthermore, two important measurements (communication cost and wavelength usage) for evaluating the performance of routing-tree are usually considered on *WDM* network for *QOS* (*Quality Of Service*). Another measurement, transmission-delay, will come into view in the problem of multicast in *WDM* network. In order to satisfy the requirement, several protocols and algorithms have been proposed on traditional network or *WDM* network to solve different problems. Recently, the multicast routing problem in *WDM* network with sparse light splitting is proposed and solved by X. Zhang, *et al*. [4]. Besides, the other researches of multicast routing with wavelength conversion [5] or with delay bound [6] but not both have been proposed.

In this paper, two characteristics of *WDM* network, nodes with different light splitting and a request with delay bounded, are considered simultaneously. A new multicast problem finding a minimal cost *light-forest* on a *WDM* network with different light splitting such that the request can be routed under delay bound is proposed. The *light-forest* is a set of light-trees and whose number is equal to the number of wavelengths used to serve a multicast request.

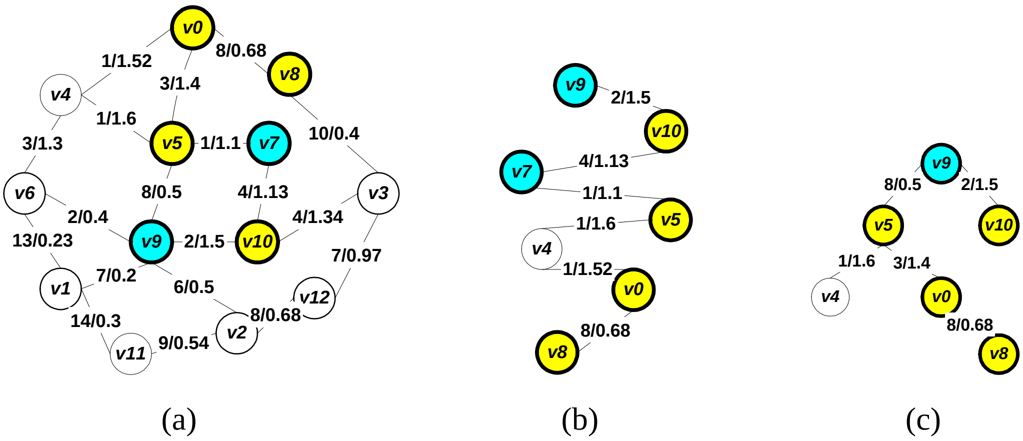


Fig. 1 *WDM* network and routing-trees for r (v_9, {v_0, v_5, v_8, v_{10}},3.3) .

For example, the graph in Fig. 1(a) represents a *WDM* network with 13 nodes, where nodes v_7 and v_9 are *MC* nodes. Each link in the graph is associated with a value-pair "a/b", where a and b are the communication cost and the transmission-delay of link, respectively. For a given request, r (v_9, {v_0, v_5, v_8, v_{10}}, 3.3), on the *WDM* network, the trees shown in Figs. 1(b) and (c) are two possible routing-trees for r, where v_9 is the source, v_0, v_5, v_8, and v_{10} are the destinations, and delay bound is 3.3 time unit. The routing-tree shown in Fig. 1(b) needs 1 wavelength, 17 communication cost units, and 7.53 time unit, and it is not feasible because 7.53 time unit is greater than 3.3. Nevertheless, because v_5 is not an *MC* node and the out-degree of v_5 is 2, the routing-tree shown in Fig. 1(c) needs 2 wavelengths for routing the request from v_5 to v_4 and v_5 to v_0, 30 communication cost units, and 2.58 time unit, and is feasible because 2.58 time unit is smaller than 3.3.

The remainder of this paper is organized as follows. In Section 2, we formally define the problem. In Section 3, a solution model composed of three phases to solve the problem is proposed and each phase of the solution model is described in detail. Section 4 presents the simulation for solution model, and Section 5 gives some conclusions.

2. Formulation

A weighted graph $G(V, E)$ denotes a *WDM* network, where the node set V represents the optical nodes (switches or routers), and the edge set E represents the optical links between nodes. The numbers of nodes and edges in the WDM network are defined as $|V|=n$ and $|E|=l$, respectively. Each link is composed of two oppositely directed fibers. For each link e, $c(e)$ and $d(e)$ are associated with edge e to represent the communication cost and transmission-delay, respectively. $\theta(v) \geq 1$ is used to represent the *splitting degree* of node $v \in V$, which is the number of copies that can be forwarded to other nodes. If $\theta(v)$ is equal to k, the node v can transmit k copies of request to other nodes concurrently by using same wavelength.

In this paper, a multicast request represented by $r(s, D, \Delta)$ goes from a certain source $s \in V$, passes several nodes, arrives at all destinations in set $D \subseteq V\text{-}\{s\}$ finally, where $|D| = m$, and the transmission-delays of light-paths between s and any destination d_i in D expect to be bounded by Δ.

Assume there are q paths, $P_i(u,v) = < e_{u,w_1^i}, e_{w_1^i,w_2^i}, \dots e_{w_{k^i}^i,v} >$ (i.e. $< u, w_1^i, w_2^i, \dots, w_{k^i}^i, v >$), between two nodes u and v, where $e_{w_j^i,w_{j+1}^i}$ is a link of nodes w_j^i and w_{j+1}^i, k^i is the number of internal nodes in $P_i(u,v)$, and $1 \leq i \leq q$. The $\mathbf{P}(u, v)=\{P_i(u,v) \mid 1\leq i\leq q\}$ is used to represent a set of all light-paths between two nodes u and v. The notations of $V(P_i(u,v))$ and $E(P_i(u,v))$ are used to represent nodes and links in path $P_i(u,v)$. The communication cost and transmission-delay of path $P_i(u, v)$ described by $c(P_i(u,v))$ and $d(P_i(u,v))$ are represented as

$$\textit{communication cost}: c(P_i(u,v)) = \sum_{e \in P_i(u,v)} c(e) = c(e_{u,w_1^i}) + c(e_{w_1^i,w_2^i}) + \dots + c(e_{w_{k^i}^i,v})$$

$$\textit{transmission-delay}: \quad d(P_i(u,v)) = \sum_{e \in P_i(u,v)} d(e) = d(e_{u,w_1^i}) + d(e_{w_1^i,w_2^i}) + \dots + d(e_{w_{k^i}^i,v})$$

Among light-paths in $\mathbf{P}(u,v)$, two critical paths, *critical-cost path* (*CCP*) whose communication cost is minimal and *critical-delay path* (*CDP*) whose transmission-delay is minimal, can be represented as $P^c(u, v)$ and $P^d(u, v)$:

$$P^c(u,v) = \min_{1 \le i \le q} c(P_i(u,v))$$

$$P^d(u,v) = \min_{1 \le i \le q} d(P_i(u,v))$$

Given k light-paths, $P(u_1, v_1)$, $P(u_2, v_2)$, …, and $P(u_k, v_k)$, they can be combined into a graph, $\bigcup_{i=1}^{k} P(u_i, v_i)$. Applying *Prim's MSpT* (*Minimal SPanning Tree*) *algorithm* for this graph, two routing-trees (spanning trees), $MSpT^c(\bigcup_{i=1}^{k} P(u_i, v_i))$ and $MSpT^d(\bigcup_{i=1}^{k} P(u_i, v_i))$, can be obtained ,where the $MSpT^c$ is used to find an *MSpT* whose multicast cost is minimal and $MSpT^d$ is used to find an *MSpT* whose transmission-delay is minimal.

Given a routing-tree T for $r(s, D, \Delta)$, root s of T has k sub-trees, T_1, T_2, …, T_k. Assume that the splitting degree of s is $\theta(s)$ which can route $\theta(s)$ copies to other nodes by using a wavelength. The lower bound of required wavelength of T is the maximum of required wavelengths of sub-trees. $\omega(T)$ and $\varpi(T)$ representing the number and the lower bound of required wavelengths of T. can be defined as :

$$\varpi(T) = \max_{1 \le i \le k} \omega(T_i)$$

$$\omega(T) = \max\left(\left\lceil \frac{\sum_{1 \le i \le k} \omega(T_i)}{\theta(s)} \right\rceil, \varpi(T)\right)$$

Because communication cost of an edge depends on the number of wavelengths passing through the edge, the total communication cost of the edge is directly proportional to the number of passing wavelengths. That is, because T_i needs $\omega(T_i)$ wavelengths, the total communication cost of e_{s,s_i} for routing a request from s to s_i should be $\omega(T_i) \cdot c(e_{s,s_i})$, where s_i is the root of T_i. The communication cost and transmission-delay of T is described as :

$$c(T) = \sum_{1 \le i \le k} (\omega(T_i) \cdot c(e_{s,s_i}) + c(T_i))$$

$$d(T) = \max_{1 \le i \le k}(d(T_i) + d(e_{s,s_i})) = \max_{1 \le i \le m} d(P^T(s, d_i))$$

Because communication cost and wavelength are critical resources usually in *WDM* network, α is defined as the ratio of the weight between two measures. If α effects how to choice a routing-tree that needs more wavelengths and lower communication cost or a routing-tree that needs fewer wavelengths and more communication cost. The *multicast cost function f* is defined as

$$f(T) = c(T) + \alpha\,\omega(T)$$

In our problem, the network does not provide wavelength conversion between different wavelengths and a light-tree can be routed by using a wavelength, so a routing-tree T need to be separated into $\omega(T)$ light-trees, T_L^1, T_L^2, …, $T_L^{\omega(T)}$, whose number of needed wavelengths is equal to 1; that is, $T = \bigcup_{i=1}^{\omega(T)} (T_L^i)$, $f(T) = \sum_{i=1}^{\omega(T)} f(T_L^i)$, and $\omega(T_L^i)=1$ for $1 \le i \le \omega(T)$. A *light-forest* $\Gamma = \{T_L^1, T_L^2, \ldots, T_L^{\omega(T)}\}$ is a set of light-trees T_L^i which route a request to the partial destinations in D, where D_i can be used to represent the set of partial destinations.

A light-forest Γ is *feasible* if it satisfies three constraints, *destination constraint*, *delay constraint*, and *degree constraint* formulated as

(1) *destination constraint* : $D = \bigcup_{i=1}^{i=k} D_i$

(2) *delay constraint* : $d(T_L^i) \le \Delta$, where $T_L^i \in \Gamma$, $\forall i$, $1 \le i \le \omega(T)$

(3) *degree constraint* : $\omega(T_L^i)=1$, where $T_L^i \in \Gamma$, $\forall i$, $1 \le i \le \omega(T)$

A routing-tree is a *candidate* if it satisfies *delay* and *destination constraints*. An efficient candidate means that the candidate needs lower multicast cost. It should be noted that a candidate does not necessarily satisfy *degree constraints* and that a candidate could be separated into a feasible light-forest. Therefore, in our solution model, once an efficient candidate is found, a feasible light-forest can be obtained easily by separating this candidate. The multicast costs of light-forest and candidate are equivalent.

Two special cases, a network with no light splitting, which let all splitting degrees of nodes be equal to 1, and a network with sparse light splitting, which let all splitting degrees of nodes be equal to 1 or ∞, were proposed [6], [4]. In this paper, we first define the following generalized problem, given a *WDM* network $G(V, E)$ with different light splitting and a request $r(s, D, \Delta)$, find a feasible light-forest Γ such that $f(\Gamma)$ is minimal. We then propose a three-phase solution model (*Pre-Processing Phase*, *Generating Phase*, and *Refining Phase*) with several heuristics to find a light-forest to route the request. The detailed description will be depicted as follows.

3. Solution Model

This new problem is NP-Complete because this problem can be reduced to the *MSTP* which is NP-Complete [9]. It is difficult to find an optimal light-forest in polynomial time. Therefore, *Backward Stepwise Sub-path Replacing* (*BSSR*) and *Most Cost-Difference First Progressive Replacing* (*MCDFPR*) heuristics are proposed to find a near optimal light-forest in polynomial time. The *BSSR* heuristic is backward stepwise tracing reversely from d_i to s in order to find a node v such that $d(P^T(s, v)) + d(P^d(v, d_i)) \leq \Delta$, and replacing the sub-path $P^T(v, d_i)$ of $P^T(v, d_i)$ with the corresponding *CDP* $P^d(v, d_i)$). The *MCDFPR* heuristic is replacing the most *cost-difference path* $P^T(u,v)$ with maximizing of $c(P^T(u,v))$- $c(P^T(u,v))$ for any two nodes u and v in $V(T)$. The solution model consisting of the following three phases is shown in Fig. 2.

(1) ***Pre-processing Phase*** – Construct the information matrices, *Critical-Cost Path Matrix* (*CCPM*) which is the matrix of $P^c(v_i, v_j)$ for $v_i, v_j \in V$ and $1 \leq i, j \leq n$, and *Critical-Delay Path Matrix* (*CDPM*) which is the matrix of $P^d(v_i, v_j)$ for all $v_i, v_j \in V$ and $1 \leq i, j \leq n$, by using *Dijkstra's Shortest Path algorithm*.

(2) ***Generating Phase*** – Use *Backward Stepwise Sub-path Replacing* (*BSSR*) heuristic to find a candidate; otherwise, return a *null-tree* to indicate no candidate.

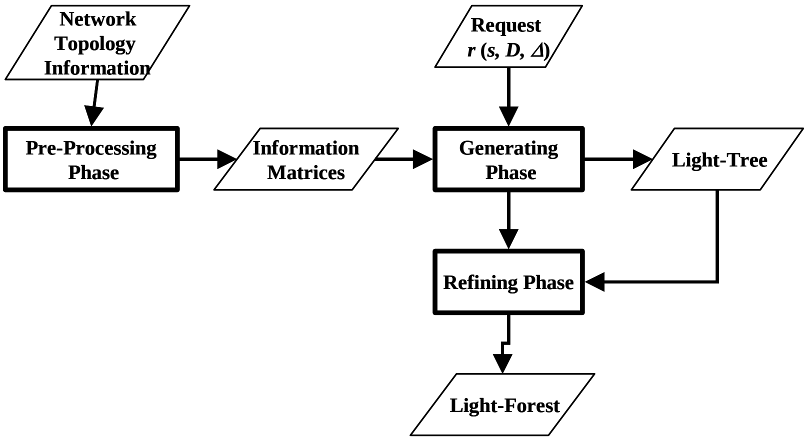


Fig. 2 Solution Model.

(3) ***Refining Phase*** – Use *Most Cost-Difference First Progressive Replacing* (*MCDFPR*) heuristic to refine the candidate to decrease multicast cost, and then separate the candidate into a feasible light-forest.

Because the *WDM* network topology is statically pre-constructed, the pre-processing for all critical-cost paths, all critical-delay paths, and corresponding communication costs and transmission-delays between any pair of nodes can reduce the computation time. By using the two matrices, *Generating Phase* can find a candidate for $r(s, D, \Delta)$. The third phase, the *Refining Phase*, can be used to refine the candidate to gain a near optimal candidate, and to separate the near optimal candidate into a feasible light-forest.

3.1. Pre-processing Phase

When the *WDM* network is constructed, the *Pre-processing Phase* is performed one time by applying the *Dijkstra's Shortest Path algorithm* which requires $O(n^2)$ time in linear array structure to construct and save two matrices (*CCPM* and *CDPM*) in disk or media for reusing. Because there are $\binom{n}{2}$ pairs of nodes for finding *CCP* and *CDP*, the time complexity of *Pre-processing phase* is $O(n^4)$.

3.2. Generating Phase

The *Generating Phase* consists of three steps :

(1) ***Checking Step*** **–** Check the multicast request whether a candidate exists or not to avoid non-necessary computation.

(2) ***Finding-MST Step*** **–** Apply the *Minimal Distance Network Heuristic* (*MDNH*) [3] to find a *Minimal Steiner Tree* (*MST*).

(3) ***Rerouting Step*** **–** Refine the *MST* to gain a candidate.

Checking Step

The *Checking Step* with $O(|D|) = O(m)$ is used to check $c(P^d(s, d_i)) \leq \Delta$ for all d_i in D. After the step, a Boolean value “TRUE” is returned, if it is positive; “FALSE” is returned, otherwise. The processing continues unless a FALSE-result is obtained.

In Fig.1 (a), assume a request r (v_9, {v_0, v_5, v_8, v_{10}}, 3.3) is given. Because $d(P^d(v_9, v_0))$=1.9, $d(P^d(v_9, v_5))$=0.5, $d(P^d(v_9, v_8))$=2.58, and $d(P^d(v_9, v_{10}))$=1.5 all are smaller than 3.3, the TRUE-value will be returned.

Finding-MST Step

The *Finding-MST Step* applying the *Minimal Distance Network Heuristic* (*MDNH*) [3] needs $O(mn^2)$ to find the *MST* (*Minimal Steiner Tree*) which is a routing-tree in *WDM*, where the *MDNH* algorithm only considers communication cost and *destination constraint*. Hence, the found *MST* may not be a candidate or a light-tree.

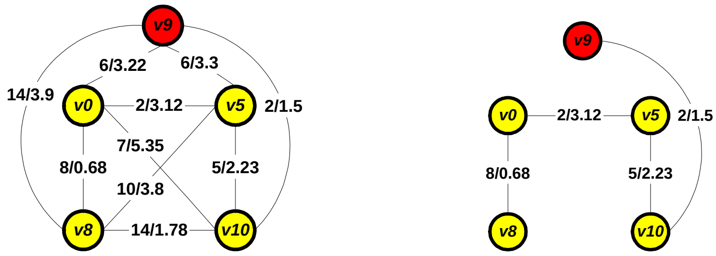
Finding_MST(r(s, D, Δ))
{

1. $G_1=(V', E')$, where $V'= s \cup D$, $E'=\{ e_{v_i,v_j} \mid P^c(v_i, v_j) \neq \emptyset, \forall\, v_i, v_j \in V' \}$,
   and $c(e_{v_i,v_j}) = c(P^c(v_i, v_j))$
2. $T_1 = MSpT^c(G_1)$
3. $T = MSpT^c\left(\bigcup_{e_{v_i,v_j} \in T_1} P^c(v_i, v_j)\right)$
4. delete all leaf nodes of T which do not belong to D

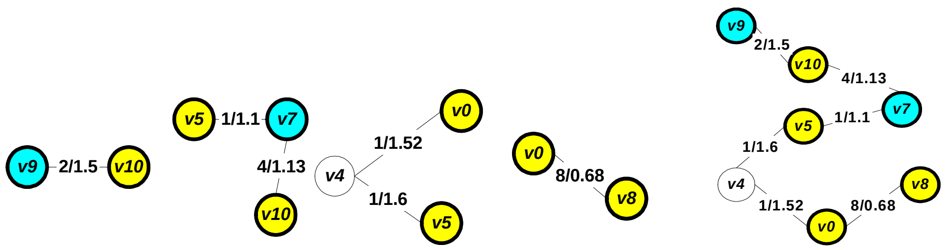
```
}
```

In Fig. 1, the sub-graph G_1 which covers source and destinations and whose *MSpT* with minimal communication cost are shown in Figs. 3(a) and (b), respectively. Each edge in Figs. 3(a) and (b) represents a path kept in *CCPM*. The corresponding paths of edges in Fig. 3(b) are shown in Fig. 3(c), and then these paths can be merged into a graph. After applying *Prim's MSpT algorithm* to this graph, a routing-tree T shown in Fig. 3(d) can be obtained.

In this step, a routing-tree T with $c(T)$=17 and $d(T)$ =7.53 for a request $r(v_9, \{v_0, v_5, v_8, v_{10}\}, 3.3)$ with near optimal communication cost can be found, but it does not satisfy the bounded delay 3.3. $V_{delay}(T)$ is used to represent the subset of destinations which violates the *delay constraint* in T. Because ($d(P^d(v_9, v_8))$=7.7, $d(P^d(v_9, v_0))$=7.02, and $d(P^d(v_9, v_5))$=3.73), $V_{delay}(T)$ is equal to $\{v_8, v_0, v_5\}$. T will be adjusted to become a candidate in the *Rerouting Step*.



(a) G' (b) T_1



(c) Corresponding path of edges in T_1 (d) Light-tree T

Fig.3 Processes of *Finding-MST Step*.

Rerouting Step

In *Rerouting Step*, the paths between s and d_i in $V_{delay}(T)$ must be rerouted and replaced with the shorter-delay path that is found by different heuristic algorithms. The path $P^T(v, d_{max})$ is rerouted first by using the *Backward Stepwise Sub-path Replacing* (*BSSR*) heuristic, where $d_{max} \in \mathrm{V}_{delay}(T)$ and $d(P^T(s, d_{\max})) = \max_{d_i \in \mathrm{V}_{delay}(T)} d(P^T(s, d_i))$. The skeleton of this step is described as follows.

```
Rerouting-Step (T, r(s, D, Δ)) /* T is a light-tree*/
{
    1. while(TRUE)
    2.    choose d_max ∈ V_delay(T), such that d(P^T(s, d_max)) = max_{d_i ∈ V_delay(T)} d(P^T(s, d_i))
    3.    if (d(P^T(s, d_max)) ≤ Δ)
              return TRUE    // satisfy the delay constraint
    4.    T = BSSR (T, r(s, D, Δ), d_max, V_delay(T))
    5.  end while-loop
}
```

Because each edge in the routing-tree T must be traveled one time to find d_{max} and compute $d(P^T(s, v))$ for all $v \in V(T)$, the computation of Sub-step 2 requires $O(l_T)$, where l_T is the number of edge in routing-tree T. The time of while-loop from sub-steps 1 to 5 can be limited by the number of destinations in $\mathrm{V}_{delay}(T)$. The sub-step 4, *BSSR* heuristic, is described as follows.

```
BSSR (T, r(s, D, Δ), d_max, V_delay(T)) /*T is a routing-tree and d_max is a heavy-delay
        destination*/
{
    1. u = Parent(d_max)              // parent of d_max in T
    2. while(u is not empty)          // if u is empty, u is a root of routing-tree T
    3.      if (d(P^T(s, u)) + d(P^d(u, d_max)) ≤ Δ)
    4.          return Reconstruct_tree(T, r(s, D, Δ), u, d_max, V_delay(T))
    5.      u = Parent(u)
    6.  end white-loop
    7. return Reconstruct_tree(T, r(s, D, Δ), s, d_max, V_delay(T))
}
```

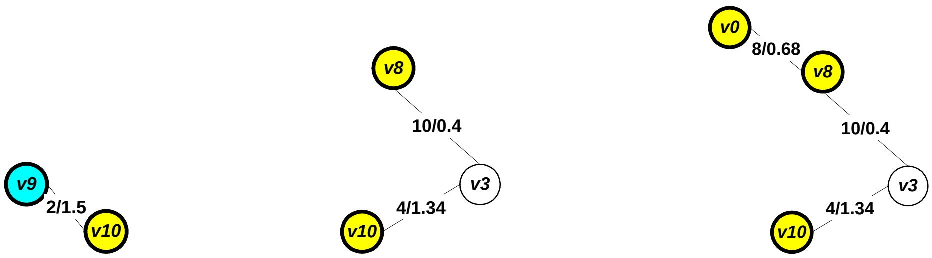
Reconstruct_tree$(T, r(s, D, \Delta), u, d_{max}, \vee_{delay}(T))$

{

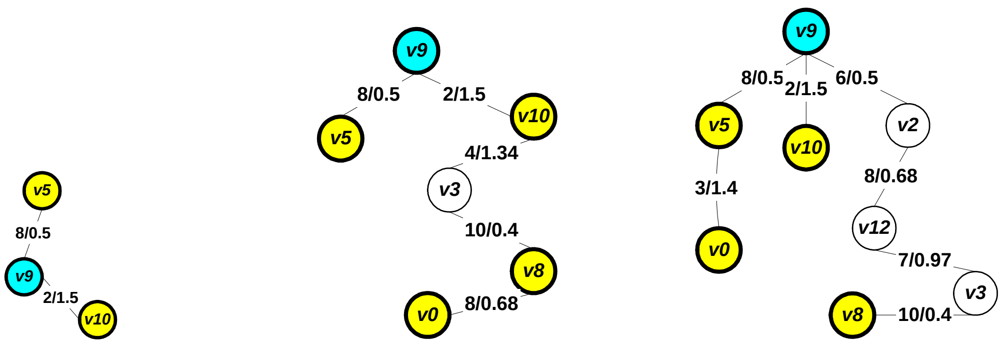
1. $D_N = \{x | x \in V(P^T(u, d_{max})) \cap D\}$ // $V(P^T(u, d_{max}))$ is a set of nodes in path $P^T(u, d_{max})$
2. $T = T - P^T(u, d_{max})$
3. $$T = MST^d(T \cup \bigcup_{v \in D_N} P^d(u, v))$$
4. return T

}

We know that the merging of T and $P^d(u,v)$ in *Reconstruct_tree* dominates the time complexity of *BSSR* heuristic and needs $O(n^2)$.



(a) The remnant (b) *CDP* between v_{10} and v_8 (c) *CDP* between v_{10} and v_0



(d) *CDP* between v_{10} and v_5 (e) After replacing of v_8 (f) After replacing of v_0

Fig. 4 Processes of *Rerouting Step*.

The routing-tree in Fig. 3 (d) is used in this phase as an input data and α =5, and d_{max} = v_8 are chosen first. Using *BSSR* heuristic, nodes in T would be checked reversely from v_8 to v_9 till the node v_{10} is found because $d(P^T(v_9, v_{10})) + d(P^d(v_{10}, v_8)) \leq 3.3$. So the path, $P^T(v_{10}, v_8)$ = $< e_{v_{10}v_7}, e_{v_7v_3}, e_{v_5v_4}, e_{v_4v_0}, e_{v_0v_8} >$, must be rerouted ; the remnant, the routing-tree in Fig. 3 (d) removed $P_T(v_{10}, v_8)$, which is shown in Fig. 4 (a). Nevertheless, v_8, v_0, and v_5 are destinations, so the paths of $P^d(v_{10}, v_8)$, $P^d(v_{10}, v_0)$, and $P^d(v_{10}, v_5)$ shown in Figs. 4 (b)~(d) need be merged into the remnant to obtain a new routing-tree shown in Fig. 4 (e). After the step, we can obtain a candidate shown in Fig. 4 (f), which satisfy *delay* and *destination constraints* and whose communication cost, transmission-delay, wavelengths, multicast cost are 44, 2.55, 1, 49, respectively.

The complexities of three steps (*Checking Step*, *Finding-MST Step*, and *Rerouting Step*) which are $O(m)$, $O(mn^2)$, and $O(n^2)$ can be summarized into $O(mn^2)$.

3.3. Refining Phase

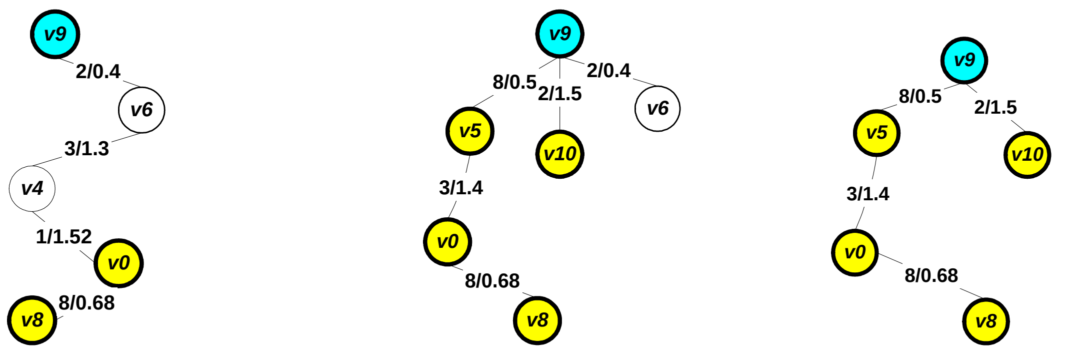
Because the candidate may be not efficient, the *Refining Phase* with the *Most Cost-Difference First Progressive Replacing* (*MCDFPR*) heuristic is used to obtain a near optimal candidate, and then the *Separating Step* is used to split the candidate into a light-forest. The detail of *MCDFPR* heuristic consisting of 9 *Steps* is described as follows.

```
MCDFPR (T)
{
    1.While computation is not exhausted
    2.  for each u, v ∈ V(T)
          Heapify(K, (u, v , c(P^T(u, v))- c(P^c(u, v))))
    3.  (u',v') = Pop(K)                          // K need re-heapify.
    4.  While((u',v')≠∅)
    5.        T'= MSpT^c ( (T- P^T(u',v'))  ∪P^c(u',v') )// u' is predecessor of v'
    6.        if (d(T')≤ Δ)
    7.              if (f(T') < f(T))            // // P^T(u',v') will be replaced with P^c(u',v')
                       T= T'
                       goto step 1
    8.  end-while loop
    9. end-while loop
}
```

In this procedure, $Heapify(K, (u, v , c(P^T(u, v))–c(P^c(u, v))))$ [10], is used to push the data structure $(u,v, c(P^T(u, v))–c(P^c(u, v)))$ into a heap K in decreasing order by the value of $c(P^T(u,$

v))−c(P^c(u, v)). If m_T is the number of leaves and l_T is the depth of T, $m_T \cdot \binom{l_T}{2}$ pairs of data structures will be pushed into K. Hence, the time complexity of *Step* 2 needs $O(m_T\, l_T^2 \log m_T\, l_T^2)$ to build a heap K. The *Step* 3 needs $O(\log m_T\, l_T^2)$ to pop root (u',v') from K and to tune K. Because *Step* 5 needs $O(n^2)$ at most to merge (T- $P^T(u',v')$) and $P^c(u',v')$. The time complexity of this phase needs $O(a(m_T\, l_T^2 \log m_T\, l_T^2 + b\cdot n^2))$, where b is the average loop time between *Steps* 4 to 8 to find a better candidate and the refining time, a, is the loop time between *Steps* 1 to 9 . Because b = $O(m_T l_T^2)$ in the worst case, the complexity needs $O(a(m_T\, l_T^2 \log m_T\, l_T^2 + m_T l_T^2\, n^2))$ and is reduced to $O(a\, m_T l_T^2\, n^2)$ due to $n > l_T$.

Because the refined candidate does not satisfy *degree constraint*, the *Separating Step* will be used to separate the candidate into a light-forest. The primary concept in *Separating Step* is to separate branches whose out-degree is greater than splitting degree ($\theta_T(v) > \theta_v$) into a set of light-trees in order to satisfy *degree constraint*.



(a) $P^c(v_8,v_9)$ (b) after replacing (c) final candidate

Fig. 5 Processes in *Refining Phase*.

The candidate in Fig. 4 (f) is an input data of this phase; the *MCDFPR* heuristic is used to separate the path $P^T(v_8,v_9) = \langle e_{v_9,v_2}, e_{v_2,v_{12}}, e_{v_{12},v_3}, e_{v_3,v_8} \rangle$, because $c(P^T(v_9, v_8)) - c(P^d(v_9, v_8)) \geq c(P^T(v_i, v_j)) - c(P^d(v_i, v_j))$ for each $v_i, v_j \in V(T)$-{v_8, v_9}. i.e., $P^T(v_8,v_9)$ is the most cost-difference path in T (Fig. 4 (f)). After the first iteration of *MCDFPR* heuristic, $P^T(v_8,v_9)$ is replaced with $P^c(v_8,v_9)$ shown in Fig. 5 (a) to gain a refined candidate in Fig. 5 (b). The near optimal better candidate shown in Fig. 5 (c) can be gained finally after *MCDFPR* heuristic is processed two times. In Fig. 5 (c), the communication cost, transmission-delay, wavelengths, and multicast are 21, 2.58, 1, 26, respectively. Hence, the multicast cost will be reduced from 44 to 26 successfully.

Finally, because the *Pre-Processing Phase* with complexity $O(n^4)$ is performed only one time for initializing *WDM* network, we can ignore the time complexity of the phase. Hence, the complexity of our solution model needs $O(m_T l_T^2 n^2)$ for finding a near optimal light-forest.

4. Simulation

Our work focuses on how to find a near optimal light-forest such that *destination*, *delay*, and *degree constraints* are satisfied. We simulate the solution model proposed in previous sections to evaluate the performance of our heuristics. The approach used in this simulation can be referred in Waxman [7]. In the approach, there are n nodes in the networks, these nodes are distributed randomly over a rectangular grid, and are placed on an integer coordinates. Each link between two nodes u and v is added with the probability function $P(u,v) = \lambda \exp(-p(u,v)/\gamma\delta)$, where $p(u,v)$ is the distance between u and v, δ is the maximum distance between any two nodes, and $0 < \lambda, \gamma \leq 1$. In the probability function, larger value of λ produce networks with high link densities, while small value of γ increases the densities of short links relative longer ones. In our simulations, we use λ=0.7, γ=0.9, size of rectangular grid = 50, and n = 100.

To reduce the complexity of problem, the cost function c of link (u, v) in the network is the distance between u and v on the rectangular coordinated grid and delay function d of link (u, v) is generated randomly. For each request $r(s, D, \Delta)$, source s, destinations D, and *delay constraint* Δ are generated randomly. Nevertheless, the *delay constraint* given by a request must be reasonable; otherwise, the light-forest cannot be found.

In our simulation, the network topology, communication costs of edges, transmission-delays of edges, and a request are generated randomly. Therefore, we discuss several comparisons consisted of execution time, multicast cost, transmission-delay, wavelengths on different number of destinations. Besides, the efficiency of *Refining Phase* will be discussed in our simulation because *Refining Phase* can improve utility of network.

The execution time of *Processing Phase*, execution time of *Refining Phase*, improved ratio of multicast cost in *Refining Phase*, overhead ratio of transmission-delay in *Refining Phase*, and comparison of wavelengths for different number of destinations are shown in Table. 1. We observe that the execution time rises moderately and is directly proportional to the number of destinations *m*. However, the growth of execution time in *Refining Phase* is unstable because the loop in the *MCDFPR* heuristic can be executed several times to improve multicast cost until it can not be improved anymore. Hence, the computation time of m=26 needs 1900 milliseconds. Let $\Gamma_{before\_refining}$ and $\Gamma_{after\_refining}$ be light-forests of before *Refining Phase* and after *Refining Phase*, respectively. The average improvement ratio of multicast cost and the average overhead ratio of transmission-delay are 20.83% and 12.13% respectively, where the improvement ratio and overhead ratio are computed by $\frac{c(\Gamma_{before\_refining}) - c(\Gamma_{after\_refining})}{c(\Gamma_{before\_refining})}$ and $\frac{d(\Gamma_{after\_refining}) - d(\Gamma_{before\_refining})}{d(\Gamma_{before\_refining})}$.

The simulation results of α about comparisons of improvement of multicast cost, overhead of transmission-delay, and usage of wavelength are shown in Figs. 6 to 8, where *m* is number of destinations, variation of wavelength is equal to the wavelength improvement of *Refining Phase*, respectively. Although the value of α is increasing, the improvement ratio of multicast cost shown in Fig. 6 will hold on 12% at least. Furthermore, the overhead ratio of transmission-delay is almost constant shown in Fig. 7 and the numbers of wavelength is inverse proportional to α shown in Fig. 8.

For the improvement of multicast cost, overhead of transmission-delay, and variation of wavelength on *Refining Phase* under different network sizes (*n*=40, 50, …, 100) and different numbers of destinations (*m*=5, 10, 20, 30) are shown from Figs. 9 to 11, respectively. The average improvement ratios of *m*=5, 10, 20, and 30 under different network sizes shown in

Fig. 9 are 23.28%, 21.51%, 22.69%, and 19.50%, respectively. The Fig. 10 is described the comparison of overhead ration of transmission-delay and whose average overhead ratios of m=5, 10, 20, and 30 under different network sizes are 7.50%, 4.05%, 7.05%, and 5.76%. The average variation of wavelengths shown in Fig. 11 are 0.46, 0.81, 1.91, and 2.46 for m=5, 10, 20, and 30; the average improvement ratios of wavelength are 14.48%, 12.12%, 16.81%, and 14.76%.

Finally, we choose different requests with different number of destinations randomly and compare the results between using our solution model (3-Phase) and using *genetic algorithm* (GA) on network with 30 nodes. The GA in our simulation choices population size = 500, generation size = 200, crossover rate = 0.8, and mutation rate = 0.1. The comparisons of computation time, multicast cost, and transmission-delay are shown in the Table. 2. Because the average computation times of our model and GA model are 0.1024 seconds and 5818.688 seconds, we can find that the computation time of GA model is far greater than our model. Therefore, we can make sure that the three-phase solution model can obtain a near optimal light-forest in polynomial time.

5. Conclusion

In this paper, a new formulation and a new multicast routing problem under delay constraint in *WDM* network with different light splitting are studied. A three-phase solution model consists of *Pre-Processing Phase*, *Generating Phase*, and *Refining Phase*, where *Backward Stepwise Sub-path Replacing* (*BSSR*) and *Most Cost-Difference First Progressive Replacing* (*MCDFPR*) heuristics have been used in *Processing Phase* and *Refining Phase* to improve the efficiency, respectively. Using the solution model, a light-forest that needs fewer wavelengths and lower multicast cost to transmit a request to meet delay constraint can be found in polynomial time.

To evaluate the performance of our solution model, an experiment has been made. The experimental result shows the average improvement ratio of multicast cost, the average overhead ratio of transmission-delay, and average improved ratio of wavelength are 20.83%, 12.13%, and 8.1% in *Refining Phase*. For different values of α, different network sizes, and different numbers of destinations, the benefits of three-phase solution model will hold still. Finally, compared with GA, the three-phase solution mode is more efficient and better than GA.

For future research, it is important to discuss the multicast routing and wavelength assignment on-line or to accommodate multiple multicast requests currently in *WDM* network with different light splitting. We are now trying to refine our solution model to solve the problem of multi-hop system.

Acknowledgement

This work was partially supported by Ministry of Education and National Science Council of the Republic of China under Grand No. 89-E-FA04-1-4, High Confidence Information Systems and by Telecommunication Lab., Chunghwa Telecom Co., Ltd.

References

[1] P. E. Green, “Fiber-Optic Networks”, Cambridge, MA : *Prentice-Hall* 1992

[2] L.H. Sahasrabuddhe and B. Mukherjee, “Light-trees : Optical multicasting for imporoved performance in wavelength-routed networks”, *IEEE Commun. Mag.*, vol 37, pp 67-73, Feb, 1999.

[3] L. Kou, G. Markowsky, L. Berman, “A Fast Algorithm for Steiner Trees,” *Acta Informatica*, Vol. 15, pp. 141-145,1981

[4] X. Zhang, J, Wei, and C. Qiao “Constrained Multicast Routing in WDM Networks with Sparse Light Splitting”, *Proceedings of INFOCOM* 2000, May

[5] N. Sreenath, N. Krishna Mohan Reddy, G. Mohan, C. Siva Ram Murthy, “ Virtual Source Based Multicast Routing in DWM Networks with Sparse Light Splitting” (*IEEE* 2001)

[6] Xiao-Hua Jia, Ding-Zhu Du, Xiao-Dong Hu “Integrated algorithm for delay bounded

multicast routing and wavelength assignment in all optical networks", *Computer Communications* 24 (2001) p.1390-1399

[7] B. M. Waxman, "Routing of multipoint connections", *IEEE JSAC*, 6 (1988), pp 333-342

[8] E. Lowe, "Current European WDM Development Trends", *IEEE Communications Magazine* 36 (2) (1998) 46-50

[9] R. M. Karp, "Reducibility among Combinational Problems," In R.E. Miller, J. W. Thatcher(Eds.), *Complexity of Computer Computations, Plenum Press,* New York, pp. 85-103, 1972

[10] T. H. Cormen. *et al*. "Introduction to Algorithm", McGraw-Hill Book Company, pp. 142-152

Table. 1.

m	Execution Time of Gent. Phase(ms	Execution Time of Ref. Phase(ms	Improv-ment Ratio(%)	Overhead Ratio(%)	Wavelengths Before Refining	Wavelengths After Refining	Variation of wavelenth
5	40	26.2	27.19	7.56	3	3	0
6	36	84.4	17.66	9.46	3.4	3.2	-0.2
7	38	46	18.55	13.59	4.6	4.6	0
8	46.4	142.2	20.58	12.6	4.2	4.2	0
9	42	160	21.82	12.63	4.6	4.6	0
10	42	142.2	24.81	13.83	5.8	4.8	-1
11	48.2	192.4	24.88	8.31	4.8	4.6	-0.2
12	50	312.6	20.53	7.86	7	7.2	0.2
13	54	304.2	20.35	16.33	5.2	4.6	-0.6
14	52	419	19.59	10.04	6.6	6.2	-0.4
15	60	354.4	19.23	16.03	5.6	4.8	-0.8
16	60	497	19.75	11.02	7.4	7	-0.4
17	62	559	21.94	14.1	8.2	9.2	1
18	76.2	594.6	19.37	13.98	7.6	5.8	-1.8
19	78.2	959.4	24.36	13.9	8.4	8	-0.4
20	72	647.2	18.93	16.06	10.8	11	0.2
21	78	905.4	20.59	12.48	7.6	6.6	-1
22	76	615.2	25.05	9.49	9.4	8.4	-1
23	82.4	654.4	20.9	8.7	9.6	10.4	0.8
24	92	741	19.52	10.93	13	12.6	-0.4
25	86	785.2	22.84	14.81	11	8.4	-2.6
26	92.4	1854.6	21.84	10.85	7.4	6	-1.4
27	92	795.2	16.06	13.6	7.2	5.8	-1.4
28	104.4	1384	18.3	9.36	10.6	7.4	-3.2
29	108.6	1173.4	18.65	13.64	10.6	9.2	-1.4
30	84	1206	18.36	14.18	9.4	8.4	-1

Table. 2 Comparisons of 3-Phase and GA.

m	Execution Time of 3-Phase(sec.)	Execution Time of GA(sec.)	Multicast Cost of 3-Phase	Multicast Cost of GA	Trans. Delay of 3-Phase	Trans. Delay of GA
5	0.001	3494	41.652	36.768	2	2.302
6	0.02	3322	47.571	42.75	2.268	2.676
7	0.03	3799	63.844	57.116	1.821	2.046
8	0.05	4385	76.403	58.889	2.622	3.098
9	0.04	4552	80.852	75.548	1.904	1.948
10	0.03	4786	77.801	73.184	1.867	2.22
11	0.08	6085	106.506	93.896	2.368	2.563
12	0.06	5365	97.203	97.029	2.451	2.355
13	0.211	6558	99.065	102.771	2.916	3.333
14	0.361	7711	119.658	112.977	3.387	3.194
15	0.071	5923	99.587	96.401	3.11	2.716
16	0.015	7235	96.539	116.361	2.38	2.737
17	0.18	7734	123.66	124.476	2.231	2.554
18	0.1	7724	159.509	120.805	2.333	2.529
19	0.11	7008	121.632	118.821	2.325	2.623
20	0.28	7418	119.62	126.739	3.115	3.133

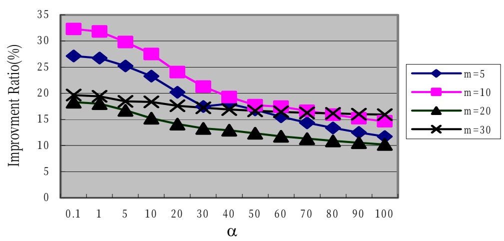


Fig. 6 Improvement ratio for different α.

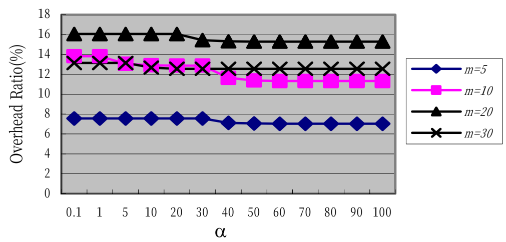


Fig. 7 Overhead ratio for different α.

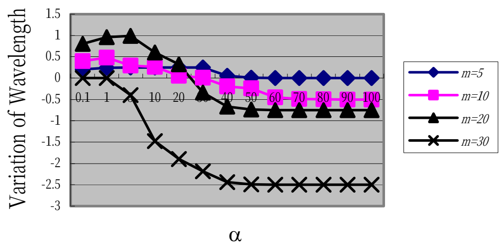


Fig. 8 Variations of wavelength for α.

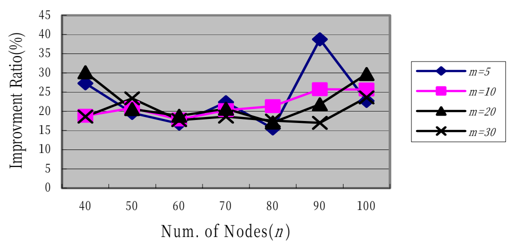


Fig. 9 Improvement Ratio of multicast cost.

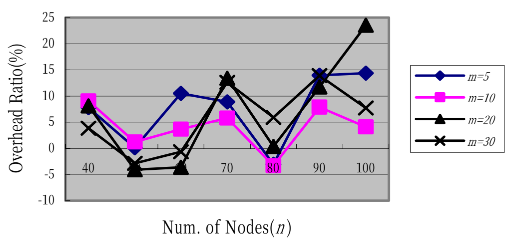


Fig. 10 Overhead Ration of Trans. delay.

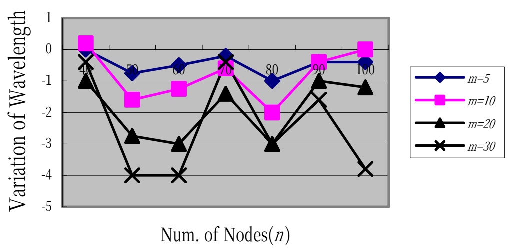


Fig. 11 Variations of wavelength.