광 베니언-유형 교환 망에서의 누화를 회피하기 위한 교환소자를 달리하는 멀티캐스트 스케줄링(제2부):스케줄링 길이 및 넌블럭킹 특성

(Switching Element-Disjoint Multicast Scheduling for Avoiding Crosstalk in Photonic Banyan-Type Switching Networks (Part II):Scheduling Lengths and Nonblocking Property)

차 영 환 †
(Yeonghwan Tscha)

요 약 선행 논문(제1부)[13]에서는 광 베니언-형 교환 망에 있어 누화를 야기하는 멀티캐스트 접속들간의 관계를 그래프로 표현하고, 해당 그래프의 차수의 상한을 제시하였다.
본 논문(제2부)에서는 교환소자를 달리하는 멀티캐스팅의 스케줄링에서의 라우팅 횟수 즉, 스케줄링 길이에 대해 다룬다. 최적의 스케줄링 길이를 구하는 문제는 NP-complete이므로 최적 길이의 상한의 두 배 이내의 길이를 제공하는 근사 알고리즘을 제시한다. 아울러, 링크를 달리하는(즉, 넌블럭킹) 멀티캐스팅에 관한 스케줄링 길이를 고찰한다. 얻어진 스케줄링 길이 하에서 다양한 넌블럭킹 베니언-형 멀티캐스팅 망들을 규명한다.

Abstract In our predecessor paper(Part I)[13], we introduced a graph that represents the crosstalk relationship among multicast connections in the photonic Banyan-type switching network, and found the upper bound on the degree of it.
In this paper(Part II), we consider the number of routing rounds(i.e., scheduling length) required for SE(switching element)-disjoint multicasting in photonic Banyan-type switching networks. Unfortunately, the problem to find an optimal scheduling length is NP-complete thus, we propose an approximation algorithm that gives its scheduling length is always within double of the upper bound on the optimal length. We also study the scheduling length on the link-disjoint(i.e., nonblocking) multicasting. Various nonblocking Banyan-type multicasting networks are found under the scheduling lengths.

1. Introduction

Optical wide-band technologies has fueled the advent of photonic switching networks. Using 2×2 $LiNbO_3$ directional couplers, Banyan, Shuffle, and Baseline networks can be constructed for high-speed photonic switching systems[7]-[9]. Even though the photonic networks can afford very abundant bandwidth, they could not come without obstacles such as signal loss and crosstalk. The crosstalk problem is more outstanding in directional-coupler-based photonic networks[8],[11],[12].

A typical system-level approach to the zero-crosstalk is to ensure that at most one input of each SE will be used at any given time, i.e., SE-disjoint routing[7],[8]. Most of previous works

· 본 논문은 2000년도 상지대학교 교내 연구비 지원에 의한 것임.
† 정 회 원 : 상지대학교 컴퓨터정보공학부 교수
yhtscha@mail.sangji.ac.kr
논문접수 : 2001년 3월 7일
심사완료 : 2001년 8월 2일

were concerned with the crosstalk-free scheduling on the Banyan-type one-to-one connection networks [7],[8],[11],[12] and asymptotic bounds on the number of wavelengths needed for the multi-Benes multicasting networks[9]. Results on the SE-disjoint multicasting in Banyan-type switching networks have not been reported yet.

In our predecessor paper(Part I)[13], we introduced a graph, denoted by G_{SE}, that represents the crosstalk relationship among multicast connections in the photonic Banyan-type switching network, and found the upper bound on the degree of it. In this paper, Part II, we consider the number of routing rounds(i.e., scheduling length) required for the SE (switching element)-disjoint multicasting in photonic Banyan-type switching networks. In general, the problem to find an optimal scheduling length is NP-complete hence, we propose an approximation algorithm that guarantees its scheduling length is less than double of the upper bound on the optimal.

The rest of this paper is organized as follows. In Section 2, we discuss the optimal upper bound on the scheduling length and propose an algorithm that gives an approximation solution. Correctness and evaluation of the algorithm are explored. Section 3 deals with the result of this paper and comparison with related works. Various conditions for the Banyan-type network to be nonblocking and/or crosstalk-free are presented. Section 4 concludes this paper.

Readers are assumed that they are familar with all definitions and terminologies given in the predecessor paper(Part I)[13]. For the sake of simplicity we go with appropriate citation only.

2. Scheduling SE-Disjoint Multicasting

In this section we consider the coloring problem on G_{SE} defined in [13]. The properly coloring[1)] of G_{SE} is equivalent to passing the SE-sharing subconnections in different routing rounds so that at most one subconnection holds each SE at any given time. The number of the colors used for G_{SE} becomes the number of routing rounds, i.e., the scheduling length.

2.1 Optimal Scheduling

It is well-known that the problem to find the chromatic number of a graph is NP-complete[3]. A self-evident result immediately follows.

Corollary 1: The problem to find the chromatic number of any G_{SE} is NP-complete.

The implication of Corollary 1 is that an optimal algorithm for coloring G_{SE} has a time complexity which is exponential to the number of subconnections to be set up. On the other hand, by definition of G_{SE}(Definition 3 in [13]), the subconnections intersecting at a common SE correspond to a *clique*[2)] in G_{SE}. For instance, in Fig. 1, the vertices <12,{12}>, <13,{13}>, <14,{14}> and <15,{15}> constitute the clique of size four. These subconnections will pass through some SE's at stage 2 and 3 all together in the 16×16 Banyan-type network.

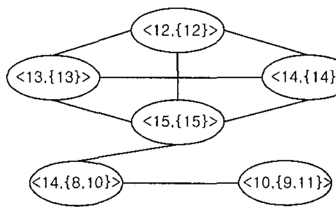

Fig. 1 G_{SE} example(subgraph of Fig. 1 in [13]).

Corollary 2: The size of the maximal clique of G_{SE} is at most δ_{SE} hence, the upper bound on the chromatic number for G_{SE} is at least δ_{SE} where, $\delta_{SE}(=2^{\lfloor (n+1)/2 \rfloor})$ is the intersection window size of the $N \times N$ Banyan-type network[13].

No matter how to design an efficient optimal coloring algorithm, it is evident that the upper

1) [2] For a graph G, we say that G is properly colored if no adjacent vertices are colored with the same color. The *chromatic number* of G is the smallest number of colors for which G can be properly colored.

2) [2] A *clique* of a graph G is a completely connected subgraph of G. The size of a clique is the number of vertices in it. The *maximal* clique is a clique whose size is not less than any others.

bound on the optimal scheduling is at least δ_{SE} because the clique of size k needs k colors for it to be properly colored.

2.2 Approximation Scheduling

We now give a scheduling algorithm that properly colors G_{SE} with $ld(G_{SE})+1$ colors. This does not guarantee the bound given by Corollary 2 nevertheless, the length based on Lemma 3 in [13] promises somewhat reasonable upper bound. The coloring of G_{SE} with $pd(G_{SE})+1$colors would be too high(see Lemma 2 in [13]).

Definition 1: Given G_{SE}, denote by *brother(j,w)* a set of vertices which are adjacent to a vertex w and correspond to the subconnections with an identical input j.

In Fig. 1, brother(14,<15,{15}>) = {<14,{14}>,<14,{8,10}>}, and |{brother(j,<15,{15}>)}| = |{{<14,{14}>, <14,{8,10}>}, {<13,{13}>}, {<12,{12}>}}| = 3 (equivalent to the logical degree of the vertex corresponding to <15,{15}>, i.e., ld(<15,{15}>)). Note that |brother(j,w)| $\leq N/\delta_{SE}$ for some j and w, and |{brother(j,w)}|=$ld(w)$.

In order for G_{SE} to be properly colored with $ld(G_{SE})+1$ colors, all vertices in each brother(j,w) must be colored with an identical color so that w and {brother(j,w)} are colored with |{brother(j,w)}|+1 colors, at most. For G_{SE} shown in Fig. 1, four colors will suffice for it to be properly colored, provided that <14,{14}> and <14,{8,10}> are colored with the same color.

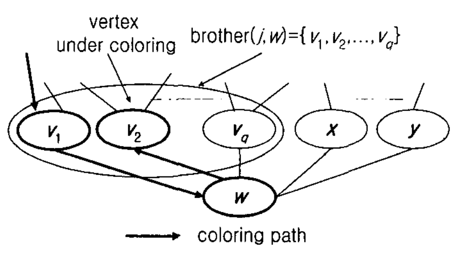

Fig. 2 Coloring the vertices in brother(j,w).

Suppose $v_1,v_2 \in$brother(j,w) for some j and w. Assume that v_1 is colored but v_2 is under coloring(see Fig. 2). Note here that w is not necessarily colored, thus v_2 may be visited via the other path on which w does not exist. The properly coloring of v_2 is done as follows.

If the color of v_1 has not been used for any vertex adjacent to v_2 then, it is used for v_2. Otherwise, choose some color used in G_{SE} but, neither for the vertices adjacent to v_1 nor to v_2,

```
SC(GSE) /* SSG GSE-Coloring */
/* C={1,2,3,...,p}: a set of available colors, integers.
   UsedColors: a set of colors used for GSE.
   Adj(V): a set of vertices adjacent to a vertex(or set of vertices) V.
   Colors(V): a set of colors used for V. */
Begin /* GSE has m connected components G^1SE, G^2SE,...,G^mSE */
  for i=1 to m
   pick up some vertex υ in G^iSE
   {UsedColors =∅; VC(υ); } /* call the routine coloring υ */
End

VC(υ) /* color uncolored vertex υ */
 Begin
   assume υ belongs to brother(j,w) for some j and w;
   if (Colors(brother(j,w))≠∅) then {
     if (Colors(Adj(υ))∩Colors(brother(j,w))=∅) then {
         θ=Colors(brother(j,w)); /* use the brother's color */
         color υ with θ; }
     else {
         θ=min{UsedColors-Colors(Adj(brother(j,w))};
         if (θ≠∅) then {/* use the used color in elsewhere */
            color υ and /* but not used for Adj(brother(j,w)) */
            re-color colored vertices in brother(j,w)
            with θ, respectively; }
         else {
            θ=min{C-UsedColors}; /* use a new color */
            color υ and
            re-color colored vertices in brother(j,w)
            with θ, respectively;
            UsedColors=UsedColors∪θ; }}}
   else { /* Colors(brother(j,w))=∅ or brother(j,w)={υ} */
     θ=min{UsedColors-Colors(Adj(υ))};
     if (θ≠∅) then { /* use the used color in elsewhere */
         color υ with θ; }/* but not used for Adj(υ) */
     else { /* use a new color */
          θ=min{C-Colors(Adj(υ))};
          color υ with θ;
          UsedColors=UsedColors∪θ;}}
   if there exists some uncolored vertex t in Adj(υ)
     VC(t); /* depth-first recursive call */
   else
      /* done for G^iSE; no. of used colors =|UsedColors| */
 End
```

Fig. 3 The algorithm SC(SSG-Coloring)

then, color v_2 and *re-color* v_1 with it, respectively. If this impossible, a new color is chosen, v_2 and v_1 are colored and re-colored respectively with it. In Fig. 2, assume vertices in brother(j,w) are colored in the order v_1, v_2, ..., v_q. In the worst case, each v_k, $1 \leq k \leq q$, will be colored one time and re-colored (q-k) times where, $q \leq N/\delta_{SE}$(Definition 2 in [13]).

A detailed description of our algorithm, SC(SSG-Coloring), is given in Fig. 3. Note that the path taken by the algorithm builds up the *depth-first spanning tree*[1] for each connected component of G_{SE}. That is, the vertex under coloring is always chosen by the depth-first search, with respect to the vertex colored most recently.

Let's consider Fig. 4, for example. Suppose the coloring is done in the order <3,{4,7}>, <2,{2}>, <1,{1}>, and <0,{0,3}>. Then, the component looks like Fig. 4 (a).

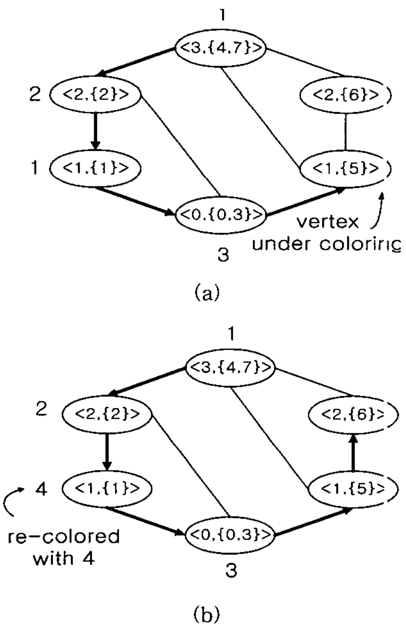

Fig. 4 Application example of the algorithm SC

We are now to color <1,{5}> adjacent to <0,{0,3}>. Noting brother(1,<0,{0,3}>) = {<1,{1}>,<1,{5}>} and Colors(Adj(<1,{5}>)) ∩ Colors(brother(1,<0,{0,3}>) = {1,3} ∩ {1} = {1} ≠ Ø, color "1" can not be used for <1,{5}>(violation of the properly coloring). Hence, by line 5 thru 6 and line 9 in Fig. 3, it follows that UsedColor - Colors(Adj(brother(1,<0,{0,3}>)) = {1,2,3}-{1,2,3} = Ø. Thus, new color "4" is chosen due to line 10, then <1,{1}> is re-colored and <1,{5}> is colored with it, respectively. Finally, since Colors(Adj(<2,{6}>)) ∩ Colors(brother(2,<3,{4,7}>)) = {1,4} ∩ {2} = Ø, color "2" is used for <2,{6}>(line 3 thru 4 in Fig. 3).

Therefore, all vertices are properly colored such that vertices in each brother(j,w) are colored with the same color(Fig. 4 (b)). The total number of colors used is 4(=$ld(G)$+1). That is, the SE-disjoint multicasting for these connections can be done in four routing rounds. Note that three rounds are optimal in this example.

Theorem 1: Using algorithm SC in Fig. 3, any G_{SE} is properly colorable with $ld(G_{SE})+1$ colors.

Proof: See Appendix. □

2.3 Evaluation of Algorithm

It is seen that the time to compute Colors(Adj (brother(j,w)))(line 6 of VC(v) in Fig. 3) for a vertex v dominates the time complexity of VC(v) and it requires $O(N_1+d(N_1+N_2))$ time at most for $G_{SE}=(V_{SE},E_{SE})$ represented by an adjacency list[1] where, $|V_{SE}|=N_1$, $|E_{SE}|=N_2$, $d=pd_{SE}(w)$. The computation will be repeated for N_1 vertices. Hence, the time complexity of SC(G_{SE}) in Fig. 3 is given as $O(N_1^2+dN_1(N_1+N_2))$. Since $N_1=O(N)$ and $d=O(n\,\delta_{SE})$(Lemma 2 in [13]), the complexity becomes $O(N^{2.5}\log_2 N)$ for $N_2=O(N)$ and $O(N^{3.5}\log_2 N)$ for $N_2=O(N^2)$, respectively.

Now, we consider the effectiveness of our approximation algorithm. Given an NP-complete problem Π, let OPT(Π) and APR(Π) be optimal and approximation solutions respectively found by their corresponding algorithms. It is said that the approximation is *good* provided that APR(Π)/OPT(Π)$\leq c$ for some constant c[3]. By Corollary 2, the upper bound on the optimal length by OPT(Π) is at least δ_{SE}, that is, OPT(Π)$\geq \delta_{SE}$. With Theorem 2(Section 3), it follows that APR(Π)$\leq 2\delta_{SE}-1$ for even n and APR(Π)$\leq 1.5\delta_{SE}-1$ for odd n, respectively. Our approximations appear to be good because APR(Π)/OPT(Π) $\le$ $[(1.5\delta_{SE}-1)/\delta_{SE}]$ < 2 for even n and APR(Π)/OPT(Π) $\le$ $[(2\delta_{SE}-1)/\delta_{SE}]$ < 1.5 for odd n[3].

3. Results and Comparison

3.1 Under the Crosstalk-Free Constraint

Given our algorithm and Theorem 1, it is a direct matter to show the upper bound on the scheduling length for the SE-disjoint multicasting in the Banyan-type switching networks.

Theorem 2: The maximum scheduling length given by SC in Fig.3 for SE-disjoint multicasting in the Banyan-type network is $2\delta_{SE}-1$ for even n and $1.5\delta_{SE}-1$ for odd n, respectively, where $\delta_{SE}=2^{\lfloor (n+1)/2 \rfloor}$.

Proof: Direct consequence from Lemma 3 in [13] and Theorem 1. □

Corollary 3: Let p_{SE} be the scheduling length required for SE-disjoint connection setup in the directional-coupler-based Banyan-type switching network. Assume p_{SE} is $2\delta_{SE}-1$ for even n and p_{SE} is $1.5\delta_{SE}-1$ for odd n, respectively. Then, the network is crosstalk-free and furthermore, 1) rearrangeable nonblocking for multicast traffic, 2) strictly nonblocking for one-to-one connection traffic, and 3) wide-sense nonblocking for multicast traffic such that $S \subseteq OW^{SE}_g (g \in \{0,1, ..., (N/\delta_{SE})-1\})$ for any $\langle w,S \rangle$.

Proof: Self-evident by definition of each nonblockingness(see footnote 2 in [13]) and Theorem 2 and Observation 2 in [13], respectively. □

Corollary 4: The photonic Banyan-type multicasting network is crosstalk-free and strictly nonblocking for multicast traffic if its SE-disjoint scheduling length is $0.5(n\delta_{SE})+1$ for even n and $0.25((n+1)\delta_{SE})+1$ for odd n, respectively.

It is interesting to note that Corollary 3 is not only new but also general in the sense that the bound in Theorem 2 coincides with that is required for strictly nonblocking one-to-one connection networks under crosstalk-free constraint[11],[12]. Compared with the strictly nonblocking multicasting network(Corollary 4).), the wide-sense nonblocking network given by a simple fanout restriction that keeps the output(s) of each multicast to the subset of an output intersection window greatly reduces the scheduling length. Routing is also much simpler than the rearrangeable nonblocking networks. For instance, the interval-splitting algorithm[4] is directly applicable. If we use a duplicated Banyan-type network(i.e., constructed by vertically stacking two copies of the network together) the optimal upper length can be achieved. It should be noticed that the length 2 is the lower bound of the SE-disjoint multicast scheduling because N (sub)connections are established in the $N \times N$ Banyan-type network in which there are $\log_2 N$ stages and $N/2$ SE's at each stage.

In reality, the subconnection-by-subconnection routing would be carefully considered whether it is admittable or not by the switching network under consideration because all destination outputs of each multicast connection may not be established simultaneously at a time. Also, recall that the bounds given by corollaries above correspond to the number of *network planes or wavelengths* required for making space-division or wavelength-division Banyan-type networks which are nonblocking and crosstalk-free under multicast traffic.

3.2 Under the Nonblocking Constraint Only

In the predecessor paper[13], we gave the upper bound on the logical degree of the graph G_L representing the link-sharing(i.e., blocking) relationship of the connections. By Corollary 3 in [13] and Theorem 1, we have the following result.

Corollary 5: The maximum scheduling length given by SC in Fig. 3 for link-disjoint multicasting in the Banyan-type network is $1.5\delta_L-1$ for even n and $2\delta_L-1$ for odd n, respectively, where $\delta_L=2^{\lfloor n/2 \rfloor}$.

It is easy to identify that the upper bound is also less than double of the optimal upper bound. The

3) Note that performance ratio is viable for G_{SE} only where, the size of the maximal clique(Corollary 2) and the maximal logical degree(Lemma 2 in [13]) are on hand. For any graph of N vertices, the best-known ratio on vertex-coloring is $O(N/\log_2 N)$(see pp.133 in [3]).

network satisfying Corollary 5 is of course *rearrangeable nonblocking for multicast traffic*(but allows crosstalk). With analogy to Corollary 3, it is seen that the network is also strictly nonblocking for one-to-one connection[5],[6],[11],[12] and wide-sense nonblocking for multicast traffic in which the output(s) of each multicast is restricted to the subset of an output intersection window defined by the window size δ_L[10]. The comparison with related results is depicted in Table 1.

Table 1 Result Comparison(RN: Rearrangeable Non-blocking, WN: Wide-sense Nonblocking, SN: Strictly Nonblocking, CF: Crosstalk-Free)

a) SE-Disjoint Scheduling

		This paper	Vaez & Lea [11],[12]	Pan *et al*[7], Qia & Zhou[8]
Connection		multicast	one-to-one	one-to-one
Scheduling Length Ensuring	RN & CF	$2\delta_{SE}-1$ (even n) $1.5\delta_{SE}-1$ (odd n)	-	Average Scheduling Length
	WN & CF	$2\delta_{SE}-1$ (even n) $1.5\delta_{SE}-1$ (odd n) ($S \subseteq OW^{SE}_f$ for $\langle w,S \rangle$)	$1.5\delta_{SE}-1$ (even n) $\delta_{SE}-1$ (odd n) (allow 1 crosstalk)	-
	SN & CF	$0.5n\delta_{SE}+1$ (even n) $0.25(n+1)\delta_{SE}+1$ (odd n)	$2\delta_{SE}-1$ (even n) $1.5\delta_{SE}-1$ (odd n)	-

b) Link-Disjoint Scheduling

		This paper	Tscha & Lee[10]	Lea[5],[6]
Connection		multicast	multicast	one-to-one
Length Ensuring	RN	$1.5\delta_L-1$ (even n) $2\delta_L-1$ (odd n)	-	δ_L
	WN	-	$1.5\delta_L-1$ (even n) $2\delta_L-1$ (odd n) ($S \subseteq OW^L_f$ for $\langle w,S \rangle$)	-
	SN	-	$0.25n\delta_L-1$ (even n) $0.5(n-1)\delta_L-1$ (odd n)	$1.5\delta_L-1$ (even n) $2\delta_L-1$ (odd n)

4. Conclusion

We have studied the problem of scheduling SE-disjoint multicasting in photonic Banyan-type networks. The problem to find an optimal scheduling length is NP-complete therein, an approximation algorithm whose upper bound is within double of the optimal upper bound has been proposed. The maximum scheduling length we have found is $2\delta_{SE}-1$ for even n and $1.5\delta_{SE}-1$ for odd n, respectively, where $\delta_{SE} = 2^{\lfloor (n+1)/2 \rfloor}$. This bound allows photonic Banyan-type multicasting networks to be rearrangeable nonblocking and crosstalk-free. It has been shown that the same bound is also sufficient for making the networks crosstalk-free and strictly(respetively, wide-sense) nonblocking under one-to-one(restricted multicast) connections. The upper bound on the scheduling length for the link-disjoint multicasting has been given for the rearrangeable nonblocking Banyan-type networks.

The theory developed in this paper gives us a unified foundation on designing nonblocking and/or crosstalk-free multicasting networks under the time/space/wavelength-division switching technologies. Extension of the theory to Banyan-type multicasting networks under the relaxed crosstalk constraints, as in [10],[11] is left for further research. A scheduling heuristic/algorithm with a better time complexity is also needed.

Appendix: Proof of Theorem 1

Proof. We first assume SSG $G_{SE}=(V_{SE},E_{SE})$ has one connected component. Let v be the vertex under coloring. Let $G'_{SE}=(V'_{SE},E'_{SE})$ be a subgraph of G_{SE} that consists of colored vertices and edges among them(not including v). Denote by $G''_{SE}=(V''_{SE},E''_{SE})$ the subgraph obtained by adding v to G'_{SE}, i.e., $G''_{SE}=G'_{SE} \cup \{v\}$. Recall that the vertex-coloring path taken by algorithm SC in Fig. 3 builds up the *depth-first spanning tree* of G_{SE}. The proof proceeds by mathematical induction on $|V_{SE}|$, the number of vertices in G'_{SE}. For $|V'_{SE}| \geq 1$, let w in G'_{SE} be a vertex adjacent to v, i.e., $w \in \text{Adj}(v)$. Letting $m=|V'_{SE}|$, the proof is as follows(the proofs for m=0 and 1 are trivial).

① m=2: Let x be another vertex in G'_{SE}, except w. Then, it follows that $x \in \text{Adj}(w)$ and $v \in \text{Adj}(w)$. Assume w and x are colored with color "1" and color "2", respectively(note that colors correspond to positive integers in the algorithm SC of Fig. 3). Supposing $v \in \text{brother}(j,w)$, whether $x \in \text{brother}(j,w)$ or not is checked by line 1 of Fig. 3. If so line 2 and otherwise line 13 are followed, respectively. In the former case, it is seen that $ld(G'_{SE})$= $ld(G''_{SE})$=1 and Colors(Adj(v))∩Colors(brother(j,w)) = Colors(w)∩Colors(x) = {1}∩{2} = ∅. The color of x, "2", is used for v and thus, the brother vertices x and v are colored with the same color "2"(line 3 thru 4). This does not violate the properly coloring. Hence, the total number of colors used is 2(=$ld(G''_{SE})$+1=1+1). The execution of line 13 implies $x\notin$ brother(j,w). If x is adjacent to v(that is, $x\in$ Adj(v)) it follows that $ld(G''_{SE})$(=2) > $ld(G'_{SE})$(=1) and UsedColors - Colors(Adj(v)) = {1,2}-{1,2} = ∅, thus line 18 thru 20 are considered. It is given that θ = min{C-Colors(Adj(v))} = min{{1,2,3,..,p}-{1,2}} = {3} (assume $p\geq3$). Color "3" is chosen for v hence, 3(=$ld(G'')$+1=2+1) colors are used up so far, without violating the properly coloring of G''_{SE}. If $x\notin$Adj(v), then it means that $ld(G'')$(=2) > $ld(G')$(=1) but UsedColors-Colors(Adj(v)) = {1,2}-{1} = {2} ≠ ∅(line 13 thru 14). Hence, v is colored with color "2"(line 15 thru 16). The number of used colors is 2(<ld(G''_{SE})+1) in this case. Therefore, it follows that ld(G'')≤2 and 3 colors suffices for all cases.

② $m=k(>2)$: By the induction hypothesis, let G'_{SE} be properly colored with $ld(G'_{SE})$+1 colors. Letting $ld(G'_{SE})=t(\geq1)$, it follows that |Colors(V'_{SE})|=t+1. For any $y\in V'_{SE}$ of G'_{SE}, the assumption implies that y and {brother(j,y)} must be properly colored with ld(y)+1 colors, at most. Since $ld(y)\leq ld(G'_{SE})$, y is adjacent to the vertices which are all properly colored with t different colors, at most(*please* note this means that $ld(y)\leq t$, not $pd(y)\leq t$). Denote by {1,2,3, ..., t+1} the number of colors used for G'_{SE}, that is, let UsedColors={1,2,3, ..., t+1}. We are now to prove the case for $m=k+1$.

③ $m=k+1$: Let $v\in$brother(j,w) such that brother(j,w)={v_1,v_2, ..., v_j,v, ..., v_q}. Without loss of generality, we assume the set of colored vertices in brother(j,w) is {v_1,v_2, ..., v_j}(note |Colors({v_1,v_2,...,v_j})|=1 if {v_1,v_2, ..., v_j}≠∅). Depending upon line 1, two cases are as follows, respectively.

1) line 2 is executed: This case means {v_1,v_2, ..., v_j}≠∅. Noting $v\in$Adj(w), we get $ld(G''_{SE})$ = $ld(G'_{SE})$(=t). If Colors(Adj(v))∩Colors(brother(j,w)) = ∅ then, line 3 thru 4 and otherwise line 5 are considered, respectively. In the former case, the color used for {v_1,v_2, ..., v_j} is also used for v, and G''_{SE} is still properly colored with t+1 colors. In the latter case, the color for v is given as θ =UsedColors-Colors(Adj(v)). If θ≠∅ then (line 7), v must be adjacent to the vertices that are colored with t different colors, at most. Since $ld(G''_{SE})$ = $ld(G'_{SE})$(=t), there exists at least one color not used for Adj(v)(note that |Colors(V'_{SE})|=t+1). Hence, with such θ, v is colored and v_1,v_2, ..., v_j are *re-colored*, respectively. G''_{SE} remains properly colored and the number of used colors is still t+1(=$ld(G''_{SE})$+1). Otherwise (i.e., θ=∅)(line 9), it is seen that ld(G''_{SE})> ld(G'_{SE}). This means $ld(G''_{SE})\geq t+1$ because G'_{SE} is a properly colored graph with t+1 colors(by the induction hypothesis) and v is adjacent to vertices which are colored with t+1 colors. Thus, a new color not used so far is required for v to be properly colored. By line 10 thru 12, a new color is given as θ = C-UsedColors = {1,2, ..., p}-{1,2, ..., t+1} = {t+2}(assume $p\geq t+2$) is chosen. And it is used for both v and its colored brothers {v_1,v_2, ..., v_j}, respectively. G''_{SE} is still properly colored. The maximal number of colors used is t+2(≤ $ld(G''_{SE})$+1).

2) line 13 is executed: This case implies either brother(j,w)={v} or Colors({v_1,v_2,...,v_j})=∅. Hence, the same procedures applied to 1) above are followed such that brother(j,w) in line 6 thru 12 is replaced with the singleton v. And the

re-coloring for $\{v_1,v_2,...,v_j\}$ in brother(j,w) is void. Thus, G''_{SE} is properly colored with $ld(G''_{SE})+1$ colors, at most.

So far G''_{SE} has been properly colored with $ld(G''_{SE})+1$ colors, at most, for all cases. Letting $G''_{SE}=G_{SE}$, G_{SE} remains a graph properly colored with $ld(G_{SE})+1$ colors. Let G_{SE} have $r(>1)$ connected components G^1_{SE}, G^2_{SE}, .., G^r_{SE}. Evidently, there exists some G^k_{SE} such that $ld(G^k_{SE}) = ld(G_{SE})$, $k \in \{1,2,...,r\}$. Each G^i_{SE} can be colored independently and properly with $ld(G^k_{SE})+1$ colors, at most. This completes the proof of the theorem. □

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading: Addison-Wesley, 1974.

[2] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, New York: American Elsevier, 1976.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, San Francisco: Freeman and Company, 1979.

[4] T. T. Lee, "Non-blocking copy networks for multicast packet switching," *IEEE J. Selected Areas in Commun.*, vol. 6, no. 9, pp. 1445-1467, Dec. 1988

[5] C.-T. Lea, "Bipartite graph design principle for photonic switching systems," *IEEE Trans. Commun.*, vol. COM-38, no.4, pp.529-538, Apr. 1990.

[6] —, "Multi-$\log_2 N$ networks and their applications in high-speed electronic and photonic switching systems," *IEEE Trans. Commun.*, vol. COM-38, no. 10, pp. 1740-1749, Oct. 1990.

[7] Y. Pan, C. Qiao, and Y. Yang, "Optical multistage interconnection networks: new challenges and approaches," *IEEE Commun. Maga.*, vol. 37, no. 2, pp. 50-56, Feb. 1999.

[8] C. Qiao and L. Zhou, "Scheduling switching element(SE) disjoint connections in stage-controlled photonic banyans," *IEEE Trans. Commun.* vol. COM-47, no. 1, pp. 139-148, Jan. 1999

[9] R. K. Pankaj "Wavelength requirements for multicasting in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 414-424, Jun. 1999.

[10] Y. Tscha and K.-H Lee, "Yet another result on multi-$\log_2 N$ networks," *IEEE Trans. Commun.*, vol. COM-47, no. 9, pp. 1425-1431, 1999.

[11] M. Vaez and C.-T. Lea, "Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 316-323, Feb. 2000.

[12] —, "Wide-sense nonblocking Banyan-type switching systems based on directional couplers," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1327-1332, Sept. 1998.

[13] Y. Tscha, "SE-disjoint multicast scheduling for avoiding crosstalk in photonic Banyan-type switching networks(Part I): graph theoretic analysis of crosstalk relationship," Journal of KISS: Information Networking, Vol. 28, no. 3, pp. 447-453, Sept. 2001.

차 영 환

1983년 인하대 전자계산학과(학사). 1985년 KAIST 전산학과(석사). 1993년 인하대 전자계산학과(박사). 1985년 ~ 1990년 한국전자통신연구원 선임연구원. 1986년 ~ 1987년 미국 NIST Guest Scientist. 1994년 ~ 현재 상지대학교 컴퓨터정보공학부 부교수. 관심분야는 통신 망 구조 및 프로토콜, 스위칭 및 라우팅, 응용 그래프론임