On Broadcast with a Common Message over Networks

Dinkar Vasudevan and Satish Babu Korada
School of Computer and Communication Sciences, EPFL, Lausanne.
Email: {dinkar.vasudevan,satish.korada}@epfl.ch

***Abstract*—We consider the following communication problem over a directed graph. A source node wants to transmit a common message to a set of receiver nodes along with a private message to each of the receivers. We show a class of networks for which the achievable rate region is given by the cut-set bounds.**

I. Introduction

Consider a directed graph represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the set of edges. Let s be the source node and $\{t_1, \ldots, t_K\}$ be K receiver nodes. The source node wants to communicate a common message W_0 to all the receiver nodes and a private message W_i to receiver node t_i for all $i = \{1, \ldots K\}$. The messages are assumed to be independent. Let R_i be the rate of message W_i for $i \in \{0, \ldots, K\}$ and $\mathcal{R}$ denote the set of achievable rates $(R_0, R_1, \ldots, R_K)$ under reliable communciation.

Let $\mathcal{I} \subseteq \{1, \ldots, K\}$ and t_I denote $\{t_i : i \in \mathcal{I}\}$. Denote the min-cut between s and $t_{\mathcal{I}}$ by $c_{\mathcal{I}}$. Let $\mathcal{R}^o$ be the set of all $(R_0, R_1, \ldots, R_K)$ tuples satisfying,

$$R_0 + \sum_{i \in \mathcal{I}} R_i \leq c_{\mathcal{I}} \;\; \forall \mathcal{I} \subseteq \{1, \ldots, K\}. \tag{1}$$

From the cut-set bounds, we get $\mathcal{R}^o$ to be an outer bound to the rate region $\mathcal{R}$ ($\mathcal{R} \subseteq \mathcal{R}^o$). For the case $K = 2$, it was shown in [1],[6],[5] that the cut-set bound is tight. It was observed in [4] that the cut-set bound is not necessarily tight when $K > 2$. The authors provide the example of the network in Figure 1. The network does not support the rate point $(1, 0, 1, 0)$ even though it lies in $\mathcal{R}^o$. The authors also demonstrate achievable rates by combining the technique of network coding and linear precoding at the source.

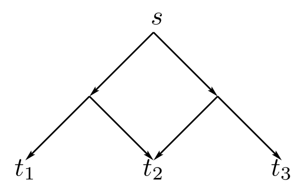

Fig. 1. Network for which the cut-set bound is not tight. All links have unit capacity.

In this work, we show a class of networks for which the cut-set bound is tight. This class consists of all *acyclic* networks for which the mincuts are characterized by

$$c_I = H_0 + \sum_{i \in I} H_i \qquad \forall I \neq \phi \tag{2}$$

and $I \subseteq \{1, \ldots, K\}$. The result is first proved for a subclass of networks – the Gray-Wyner networks introduced in Section II, which consist of independent point to point links joining the source node to the receivers. We extend the result to acyclic networks by converting the network into a Gray-Wyner network having the same mincuts. This conversion uses linear precoding at the source and network coding at the intermediate nodes. In Section IV, we examine the class of more general (acyclic and cyclic) networks with mincuts satisfying (2). For the $K = 3$ receiver case, the tightness of the cutset bound is shown using an algorithm that identifies an embedded Gray-Wyner network in the original network. We conclude with a discussion of the results in Section V.

II. Gray-Wyner Networks

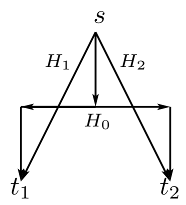

Fig. 2. The Gray-Wyner network.

The network consists of a common link of capacity H_0 connecting the source to all the receivers and K edge disjoint "private" links connecting the source to each of the K individual receivers. The capacity of the private link to receiver t_i is H_i. The network is illustrated in Figure 2 for the case $K = 2$. We refer to this class of networks as the Gray-Wyner network as it was studied in [7] in the context of a source coding problem for the case $K = 2$.

We make a few definitions. An extreme point of a convex region is one that cannot be expressed as a convex combination of two distinct points of the region. A point $x = (x_1, \ldots, x_n)$ in $\mathbb{R}^n$ is said to *majorize* point $y = (y_1, \ldots, y_n)$ in $\mathbb{R}^n$ if $x_i \geq y_i$ for all i.

Lemma 1. *For the Gray-Wyner network, $\mathcal{R} = \mathcal{R}^0$.*

Proof: The region $\mathcal{R}^o$ is the intersection of hyperplanes defined in (1). It is enough to show that every extreme point of $\mathcal{R}^o$ is achievable. Furthermore, it is sufficient to show the achievability of the extreme points of $\mathcal{R}^o$ which are not majorized by any other point of $\mathcal{R}^o$. There are at most $K+2$ such extreme points given by:

1) $(R_0, R_1, \ldots, R_K) = (H_0, H_1, \ldots, H_K)$.
2) K points of the type: $\forall i \in \{1, \ldots K\}$

$$R_0 = 0, R_i = H_0 + H_i, \ \forall\, j \notin \{i, 0\},\ R_j = H_j.$$

3) Let $i = \min_{j=1}^{K} H_j$.

$$R_0 = H_0 + H_i, R_i = 0, \ \forall\, j \notin \{i, 0\},\ R_j = H_j - H_i.$$

It is straightforward to check that the extreme points listed above are achievable. ∎

Remark 1. *For the case $K = 3$, if we choose $(H_0, H_1, H_2, H_3) = (1, 0, 1, 0)$, the vector of mincuts for the Gray-Wyner network (i.e, the 7 min-cut values from s to every subset of receivers) is majorized by the vector of mincuts for the network of Figure 1. Still, the rate point $(1, 0, 1, 0)$ is not achievable for the latter network while it is achievable in the Gray-Wyner network.*

III. Linear Precoding for the generalized Gray-Wyner Network

The generalized Gray-Wyner network is *any* directed graph whose mincuts satisfy (2). We use the notation $\text{In}(\cdot)$ and $\text{Out}(\cdot)$, respectively, for the incoming and outgoing edges of a vertex. The notation $\langle\cdot\rangle$ denotes the linear span of a set of vectors. We use I, J to represent subsets of the index set $\{1, \ldots, K\}$.

We recall, with changes in notation, some results in linear network coding from [2]. In linear network coding, on every outgoing edge, nodes transmit a linear combination of the symbols received on the incoming edges. Let F be a finite field and ω a positive integer. To every edge e, we associate the *global encoding kernel* $f_e \in F^{\omega}$. The source generates the message $x \in F^{\omega}$. A node v in the network receives the symbols $x^T f_d$, $\forall d \in \text{In}(v)$. On each edge $e \in \text{Out}(v)$, it sends symbol $x^T f_e$. We recall the definition of a *linear dispersion code* from [2].

Definition 1. *Let vectors f_e denote the global encoding kernels in an ω-dimensional F-valued linear network code on an acyclic network. Let $P_i = \langle\{f_e : e \in In(t_i)\}\rangle$. Then, the linear network code is a* linear dispersion *code if for every subset t_I of sink nodes, dim$(\langle\cup_{i\in I} P_i\rangle) = \min\{\omega, mincut(I)\}$.*

The following theorem (Corollary 2.22 in [2]) proves the existence of a linear dispersion code.

Theorem 1. *[2] Given a positive integer ω and an acyclic network, there exists an ω-dimensional F-valued linear dispersion code for sufficiently large base field F.*

We use the following lemma which is proved in Appendix A.

Lemma 2. *For all $I \subseteq \{1, \ldots, K\}$, let the dimensions of the vector spaces P_i satisfy*

$$dim(\langle\cup_{i\in I} P_i\rangle) = H_0 + \sum_{i\in I} H_i \qquad (3)$$

Then, dim$(\cap_{i\in\{1,\ldots,K\}} P_i) = H_0$.

Let $\{e_1^0, \ldots, e_{H_0}^0\}$ be a basis for $\dim(\cap_{i=1}^K P_i)$. We can construct a basis for each P_i in the following manner – First, consider the basis elements of the subspace $\cap_{i=1}^K P_i$ given by $\{e_1^0, \ldots, e_{H_0}^0\}$. Extend this basis to a basis for P_i – $\{e_1^0, e_2^0, \ldots, e_{H_0}^0, e_1^i, e_2^i, \ldots, e_{H_i}^i\}$.

Theorem 2. *For an acyclic generalized Gray-Wyner network, $\mathcal{R} = \mathcal{R}^0$.*

Proof: From cutset bounds, it is clear that $\mathcal{R} \subseteq \mathcal{R}^0$. We show that $\mathcal{R} \supseteq \mathcal{R}^0$ by using precoding at s (in the same spirit as in [4],[5],[6]) and network coding at the intermediate nodes of $\mathcal{G}$, to overlay a Gray-Wyner network on top of the actual network.

Choose $\omega = \dim(\langle\cup_{i\in\{1,\ldots,K\}} P_i\rangle)$. From Theorem 1, there exists an ω-dimensional F-valued linear dispersion code such that kernels P_i satisfy (3). Let $\mathbf{A}$ be the matrix of the ω basis column vectors – $\{\cup_{i=0}^K \cup_{j=1}^{H_i} e_j^i\}$. We choose $(\mathbf{A}^{-1})^T$ as the precoding matrix. In particular, the data vector x is multiplied with the precoding matrix to give $(\mathbf{A}^{-1})^T x$. If f_e is the global encoding kernel for the edge $e \in \text{In}(t_i)$, let $\mathbf{K}_i$ be the matrix with the f_e's as its column vectors. Then, the received symbols at t_i is given by

$$x^T \mathbf{A}^{-1} \mathbf{K}_i = x_i^T \mathbf{B}.$$

Here, x_i is a subvector of x containing the data symbols that multiply the column vectors of $(\mathbf{A}^{-1})^T$ (in the product $(\mathbf{A}^{-1})^T x$) which are orthogonal with the column vectors $e_{(\cdot)}^j$ where $j \neq i, 0$. $\mathbf{B}$ is the $\dim(P_i) \times \dim(P_i)$ submatrix of $(\mathbf{A}^{-1})^T \mathbf{K}_i$ which consists of the non-zero rows of $(\mathbf{A}^{-1})^T \mathbf{K}_i$. One can check that the matrix $\mathbf{B}$ is invertible. The received symbols $x_i^T \mathbf{B}$ can be post multiplied with $\mathbf{B}^{-1}$ to recover x_i. This coding results in the graph $\mathcal{G}$ emulating the Gray-Wyner graph with the same mincuts. ∎

Remark 2. *We emphasize here that our precoding technique differs from that in [4]. In [4], the inner product over the field is assumed while constructing the precoding matrix. On the other hand, our precoding technique is more general as it works for any finite field.*

IV. Embedding the Gray-Wyner Network in the generalized network

For the case when $K = 3$, we show that a generalized Gray-Wyner network has embedded in it, a structure, very similar to the Gray-Wyner network defined in Section II. More precisely, we have the following theorem

Theorem 3. *Consider a generalized Gray-Wyner network with 3 receivers. For all i there exist H_i paths to t_i all of which (the $\sum_{i=1,2,3} H_i$ paths) are mutually edge disjoint. Furthermore, if one removes these private paths, the residual graph has a min-cut H_0 between s and each t_i.*

The proof is in the same spirit as the proof in [1], which considers the case $K = 2$.

Proof: Without loss of generality, using standard arguments based on time expansion, we assume that the links of

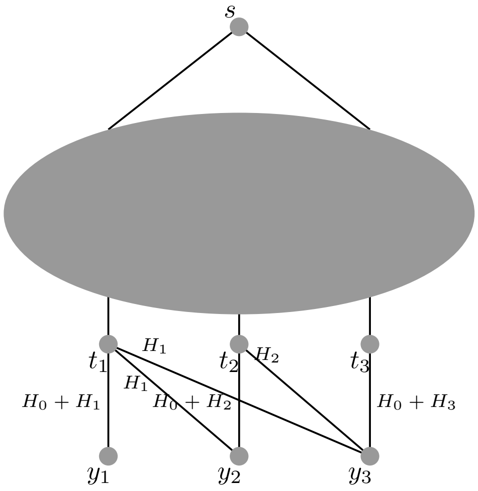

Fig. 3. The augmented graph $\mathcal{G}'$

$\mathcal{G}$ have integral capacity. Replace each link with an equivalent number of unit capacity edges. The proof uses a coloring scheme for the edges which will be explained later. Let us first introduce the notation involved. The colors used are denoted by C_i where $i \in \{1, 2, 3\}$. Let $I(\mathsf{e})$ denote the set of indices of the colors on edge e. We say an edge is coloured C_I if the edge is colored with $\{C_i : i \in I\}$. Let k_I be the number of edges connected to the source with color C_I. For a pair of colors we define the dominant color as follows. The dominant color in $\{C_1, C_2\}$ is C_2, in $\{C_2, C_3\}$ is C_3,in $\{C_3, C_1\}$ is C_1.

We augument $\mathcal{G}$ to graph $\mathcal{G}'$ as follows. For each sink t_i we introduce a new node y_i. Join t_i and y_i by $H_0 + H_i$ edges of unit capacity. Join t_i and y_j for $j > i$, with H_i edges of unit capacity (refer Figure 3).

One can show that the min-cut from s to y_i is $\sum_{j=0}^{i} H_j$. For $i = 1, 2, 3$, pull out $\sum_{j=0}^{i} H_j$ edge disjoint paths from s to y_i and color the edges in the paths with C_i. The existence of such paths follows from Menger's theorem[8]. Then the following equalities hold

$$k_1 + k_{1,2} + k_{3,1} + k_{1,2,3} = H_0 + H_1 \quad (4)$$
$$k_2 + k_{2,3} + k_{1,2} + k_{1,2,3} = H_0 + H_1 + H_2 \quad (5)$$
$$k_3 + k_{3,1} + k_{2,3} + k_{1,2,3} = H_0 + H_1 + H_2 + H_3 \quad (6)$$

Let O be the set of edges coming out of s. Label these edges as e_i, $i = 1, \ldots, |O|$. Apply the algorithm `Find private paths` (refer Figure 4) on this colored graph:

Lemma 3. *The algorithm* `Find private paths` *terminates.*

The proof is defered to the Appendix. The *No Swapping* condition is equivalent to the following four conditions being satisfied.

(i) Either there are no outgoing edges from s colored $C_{1,2}$, or, the C_2 path from an outgoing edge colored $C_{1,2}$ never intersects with a C_3 path,
(ii) Either there are no outgoing edges from s colored $C_{2,3}$, or, the C_3 path from an outgoing edge colored $C_{2,3}$ never intersects with a C_1 path,
(iii) Either there are no outgoing edges from s colored $C_{3,1}$, or, the C_1 path from an outgoing edge colored $C_{3,1}$ never intersects with a C_2 path.
(iv) The path along the outgoing edges of s with a single color never intersect with any other colors.

When the algorithm terminates the above conditions hold. Furthermore, the following lemma holds on the subgraph induced by the colored edges.

Data: $\mathcal{G}', I(\mathcal{E})$
begin
  for *i =1* **to** $|O|$ **do**
    switch $|I(\mathsf{e}_i)|$ **do**
      case 0
        Do nothing;
      end
      case 1
        Follow the path colored $C_{I(\mathsf{e}_i)}$ starting from e_i
        if *An edge* e' *in this path is colored* C_k *and* $k \neq I(e_i)$ **then**
          For each color in $C \in I(\mathsf{e}')\backslash I(\mathsf{e}_i)$, uncolor C along the path from s to e';
          Add the color(s) $I(\mathsf{e}')\backslash I(\mathsf{e}_i)$ on the edges in the path from s to e' through e_i
        end
      end
      case 2
        Follow the path of the dominant color C_j of $I(e_i)$ starting from e_i ;
        if *Any edge* e' *in this path is colored* C_k, $k \notin I(\mathsf{e}_i)$ **then**
          Uncolor C_k along the path from s to e';
          Add the color C_k on the edges in the path from s to e' through e_i ;
        end
      end
      case 3
        Do Nothing;
      end
    end
  end
  if *No Swapping* **then**
    EXIT
  end
  else
    Goto begin
  end
end

Fig. 4. Algorithm - `Find private paths` for identifying private paths to receivers

Lemma 4. *The number of edge disjoint paths to the receiver set R is given by $\sum_{I \cap R \neq \phi} k_I$.*

The proof is in the Appendix. The equations (4),(5),(6) hold at the end of the algorithm because the number of edge disjoint paths of a given color does not change at any step.

The equation (6) implies that $k_1 = k_2 = k_{1,2} = 0$. If $k_2 + k_{1,2}$ were non-zero, this would imply the existence of paths colored C_2 (corresponding to outgoing edges of s colored C_2 or $C_{1,2}$) that are edge disjoint with respect to the paths colored C_3. This would violate the overall mincut. Similarly, it follows that $k_1 = 0$.

The equation (5) implies $k_{3,1} = 0$. If not, there exists a path colored C_1 from outgoing edges of s colored $C_{3,1}$ which is edge disjoint with respect to the paths colored C_2. The number of edge disjoint paths from s to receivers (t_1, t_2) would exceed the mincut between s and (t_1, t_2).

Subtracting (5) from (6) gives $k_3 = H_3$. These k_3 paths pass through t_3 and hence can be considered as private paths to t_3.

Decolor the color C_3 from the H_1 paths which pass through receiver t_1 and from the H_2 paths which pass through receiver t_2. Apply the algorithm `Find private paths` again.

As before, (5) holds and by the previous reasoning, $k_1 = k_{3,1} = 0$. Equation (4) implies that $k_{2,3} = 0$. If not, there exist paths colored C_3 (corresponding to outgoing edges of s colored $C_{2,3}$) which are edge disjoint with respect to the paths colored C_1. This would imply the existence of edge disjoint paths to receivers (t_1, t_3) exceeding the mincut to (t_1, t_3).

Subtracting (4) from (5) gives $k_2 = H_2$.

Decolor the color C_2 on the H_1 paths that pass through receiver t_1 and again apply the algorithm. The following equalities hold –

$$k_1 + k_{1,2} + k_{3,1} + k_{1,2,3} = H_0 + H_1 \tag{7}$$
$$k_2 + k_{2,3} + k_{1,2} + k_{1,2,3} = H_0 + H_2 \tag{8}$$
$$k_3 + k_{3,1} + k_{2,3} + k_{1,2,3} = H_0 + H_3 \tag{9}$$

The three relations correspond to the number of edge disjoint paths colored C_1, C_2, C_3 respectively. Furthermore, $k_2 \geq H_2$ and $k_3 \geq H_3$ because once the single colored paths are created they are never decolored.

Counting the edge disjoint paths to all the receivers (t_1, t_2, t_3), we get

$$\begin{aligned} k_1 + k_2 + k_3 + k_{1,2} &+ k_{2,3} + k_{3,1} + k_{1,2,3} \\ &= H_0 + H_1 + H_2 + H_3 \end{aligned}$$

The equality is due to the following: The LHS is at least the RHS since it contains the terms in the LHS of (7), $k_2 \geq H_2$ and $k_3 \geq H_3$ and it is at most the RHS which is the mincut. Furthermore, the equality implies $k_2 = H_2$, $k_3 = H_3$ and $k_{2,3} = 0$. Substituting this in (8),(9) we get

$$k_{1,2} + k_{1,2,3} = H_0 \tag{10}$$
$$k_{3,1} + k_{1,2,3} = H_0 \tag{11}$$

Counting the edge disjoint paths to the receivers (t_2, t_3), we have

$$k_2 + k_3 + k_{1,2} + k_{2,3} + k_{3,1} + k_{1,2,3} = H_0 + H_2 + H_3. \tag{12}$$

The LHS is at least the RHS since $k_2 = H_2$, $k_3 = H_3$ and from (11) and is upperbounded by the RHS which is the mincut. The above equality implies that

$$k_{1,2} + k_{3,1} + k_{1,2,3} = H_0 \tag{13}$$

From (10),(11) and (13), it follows that $k_{1,2,3} = H_0$ and $k_{1,2} = k_{3,1} = 0$. Substituting in (7), we get $k_1 = H_1$.

It follows that if we remove the k_i private paths of color C_i, to the receiver t_i, the remaining graph still has a mincut of H_0 to each receiver. ■

Corollary 1. *The cut-set bound is tight for a generalized Gray-Wyner network $\mathcal{G}$.*

From cutset bounds, it is clear that $\mathcal{R} \subseteq \mathcal{R}^0$. From Theorem 3, for all i, one can find H_i private paths to receiver t_i such that when these paths are removed, the residual graph has mincut H_0 to all receivers. By network coding [3], this residual graph emulates a common link of capacity H_0 from the source node s to each of the receivers. From Lemma 1, it follows that the cut-set bound is tight for $\mathcal{G}$.

V. Conclusion

Two coding techniques are presented for broadcasting a common and a private message to each receiver from a source. The techniques are shown to be optimal for a class of graphs satisfing the mincut property (2). The first technique combines linear precoding at the source and network coding at the intermediate nodes. However, we can only prove the Theorem 2 for *acyclic* networks. This is because we use linear dispersion network codes, whose existence/construction has been proved only for acyclic networks [2]. It is not clear how to extend the proof techniques for the cyclic case.

The second coding technique identifies an embedding of a Gray-Wyner network in a network with mincuts satisfying (2) via an algorithm. The technique is attractive since it proves the capacity region of the generalized Gray-Wyner network with 3 receivers for *both* the acyclic and the cyclic case. Furthermore, routing suffices to communicate the private message and network coding is required only to communicate the common message. The limitation of the Theorem 3 is we are not able to extend it for the $K > 3$ receivers case.

References

[1] A. Ramamoorthy, R.D. Wesel, "The Single Source Two Terminal Network with Network Coding ", in Canadian Workshop on Information Theory, 2005

[2] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network Coding Theory", now Publishers, 2005.

[3] R. Ahlswede, N. Cai, S.-Y.R. Li and R. W. Yeung, "Network Information Flow", *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[4] Q. Zou, A. Nosratinia, A.H.Sayed "Precoding for Broadcasting with Linear Network codes", in Proc. of the Allerton Conf. on Commun., Control, and Computing, Monticello, USA, Sep 27 - 29 2006.

[5] C. K. Ngai and R. W. Yeung, "Multisource network coding with two sinks", in *Proc. International Conference on Communications, Circuits and Systems (ICCCAS)*, vol. 1, pp. 3437, Jun. 2004.
[6] E. Erez and M. Feder, "Capacity region and network codes for two receivers multicast with private and common data", in *Proc. Workshop on Coding, Cryptography and Combinatorics*, China, 2003.
[7] R.M. Gray, A.D. Wyner "Source coding for a simple network", Bell Syst. Tech. J, 1974
[8] D B West "Introduction to Graph Theory" , *http://www.math.uiuc.edu/ west/igt/igtpref.ps*

APPENDIX

A. Proof of Lemma 2

The proof uses induction on K.
Induction basis: For $K=2$, we have

$$\begin{aligned}\dim(P_1\cap P_2)&=\dim(P_1)+\dim(P_2)-\dim(\langle P_1\cup P_2\rangle)\\&=(H_0+H_1)+(H_0+H_2)\\&\quad-(H_0+H_1+H_2)=H_0\end{aligned}$$

Induction hypothesis: Assume that

$$\dim(\cap_{i=1}^{K-1}P_i)=H_0 \tag{14}$$

Induction step:

$$\begin{aligned}\dim(\cap_{i=1}^{K}P_i)&=\dim\Big(P_K\cap\big(\cap_{i=1}^{K-1}P_i\big)\Big)\\&=\dim(P_K)+\dim(\cap_{i=1}^{K-1}P_i)-\dim\Big(\langle P_K\cup\big(\cap_{i=1}^{K-1}P_i\big)\rangle\Big)\\&\overset{(a)}{\geq}\dim(P_K)+\dim(\cap_{i=1}^{K-1}P_i)\\&\quad-\dim\Big(\langle P_K\cup P_1\rangle\cap\langle P_K\cup P_2\rangle\cap\cdots\cap\langle P_K\cup P_{K-1}\rangle\Big)\\&\overset{(b)}{=}\dim(P_K)+\dim(\cap_{i=1}^{K-1}P_i)-\dim(\cap_{i=1}^{K-1}\tilde{P}_i)\\&\overset{(c)}{=}\dim(P_K)+H_0-\tilde{H}_0\\&=(H_0+H_k)+H_0-(H_0+H_K)\\&=H_0\end{aligned}$$

The inequality (a) follows since $\langle P_K\cup(\cap_{i=1}^{K-1}P_i)\rangle$ is a subspace of $\langle P_K\cup P_i\rangle$ for all i. Hence $\langle P_K\cup(\cap_{i=1}^{K-1}P_i)\rangle$ is a subspace of $\cap_{i=1}^{K-1}\langle P_K\cup P_i\rangle$. In the equality (b), we define $\tilde{P}_i=P_K\cup P_i$. Note that the induction hypothesis applies also for $\cap_{i=1}^{K-1}\tilde{P}_i$, since $\tilde{P}_i$ satisfy

$$\dim(\langle\cup_{i\in I}\tilde{P}_i\rangle)=\tilde{H}_0+\sum_{i\in I}\tilde{H}_i,\ \forall I\subseteq\{1,\ldots,K-1\}$$

where $\tilde{H}_0=H_0+H_K$ and $\tilde{H}_i=H_i$. Therefore, $\cap_{i=1}^{K}\tilde{P}_i=\tilde{H}_0=H_0+H_K$ and the equality (c) follows. The proof now follows from

$$\dim(\cap_{i=1}^{K}P_i)\leq\dim(P_1\cap P_2)=H_0.$$

B. Proof of Lemma 3

The proof is by contradiction. Suppose the algorithm does not terminate. This implies that after each pass of the for loop of the algorithm, one of the *No Swapping* conditions is violated. We proceed to show that this cannot indefinitely happen.

1) The condition (i) in *No Swapping* does not occur infinitely often. Suppose it does, then for some choice of a C_1 path, a C_2 path and a C_3 path, the transition $(C_1,C_2)\rightarrow(C_1,C_2,C_3)$ (effected by case 2 in the for loop) occurs infinitely often. It is possible to check that successive transitions result in the colors C_2 and C_3 being conjoined over longer paths (paths with more number of edges). Since the graph is finite, this cannot happen infinitely often. Similarly it can be shown that the conditions (ii) and (iii) in *No Swapping* do not occur infinitely often.
2) The condition (iv) in *No Swapping* does not occur infinitely often – Each time (iv) occurs, then it results in a *complete decoloring* of at least one edge. (The case 1 in the for loop transfers all the new colors onto the single colored path thereby completely decoloring the edge which the path intersects with.) If an edge is completely decolored, it is never colored again in the algorithm. The decoloring of an edge therefore cannot occur infintely often. .

C. Proof of Lemma 4

The case where R is a singleton its trivial. Consider the case where R consists of two colors. Without loss of generality let $R=\{1,2\}$. The number of edge disjoint paths to the receiver set $\{1,2\}$ is given by the number of edge disjoint paths in the subgraph of $\mathcal{G}$ induced by the colors C_1 or C_2. We first count the paths having color C_2. Since all these paths are edge disjoint, the number of these paths equals the sum of terms k_A where $2\in A\subseteq\{1,2,3\}$. Now count the paths involving color C_1. After the termination of the algorithm, we know that the path colored C_1 outgoing from an edge of s colored $\{1,3\}$ does not interesect with a path colored C_2. Therefore the $k_{3,1}$ paths colored C_1 are edge disjoint with the paths colored C_2. Also the paths with the single color C_1 are edge disjoint with the paths colored C_2. Hence we get the total number of edge disjoint paths to be $k_1+k_2+k_{1,2}+k_{2,3}+k_{3,1}+k_{1,2,3}$ which is equal to the sum given in the lemma. Note that, we first considered all the edge disjoint paths colored C_2 and then considered edge disjoint (with respect to C_2) paths colored C_1. The proof would not have worked had we considered first all the edge disjoint paths colored C_1 and then considered edge disjoint (with respect to C_1) paths colored C_2. This is because, according to our algorithm the color C_1 tries to join C_2 but the opposite is not true.

Similar reasoning can be used to count the number of edge disjoint paths for the remaining subsets of receivers. The order in which we pull out the edge disjoint paths for the different receiver subsets are given by – For $R=\{2,3\}$, we follow the order $(3,2)$, for $R=\{1,3\}$ we follow the order $(1,3)$ and for $R=\{1,2,3\}$ we can follow any of $(1,3,2)$,$(3,2,1)$ or $(2,1,3)$.