On the Complexity of Variants of Cooperative Peer-to-peer Repair for Wireless Broadcasting

Gene Cheung, Danjue Li and Chen-Nee Chuah

***Abstract*— The well-known NAK implosion problem for wireless broadcast can be addressed by leveraging cooperative peer-to-peer connectivity to repair corrupted data. This paper studies the Cooperative Peer-to-peer Repair (CPR) framework for multimedia broadcast. We show that CPR can be formulated as an optimization problem that minimizes the number of iterations it takes to wirelessly disseminate a desired message from peers 'with' the content to peers 'without' it. Complicating the problem are transmission conflicts, where pre-specified sets of links cannot simultaneously transmit due to interference. In this paper, we formalize CPR as a discrete optimization problem, and prove that CPR and its many variants are NP-hard.**

I. Introduction

A new and promising distribution model for 3GPP networks is Multimedia Broadcast Multicast Service (MBMS) [1], where a piece of widely interested multimedia content (message) is broadcasted to large groups of 3G clients listening collectively in a pre-assigned broadcast channel. While it is clear that efficient usage of network resources is a benefit, in fear of the NAK implosion problem — scenario where server is overwhelmed by floods of individual retransmission requests from clients — broadcasting servers typically do not perform retransmission on request in the event of packet losses due to wireless transmission failures. Even with the use of Forward Error Correction (FEC) to correct predictable channel noise, temporary wireless link failures are often unavoidable, leaving groups of clients without the desired message at a given time.

Fortunately, many modern wireless devices are multi-homed and each contains multiple wireless interfaces, so that one can connect to a wireless wide area network (WWAN), like a 3G network, and to a wireless local area network (WLAN), like a wireless ad-hoc peer-to-peer network, simultaneously [2]. In such setting, a "have not" wireless peer can request retransmission of a message from a neighboring "have" peer listening to the same broadcast. Given a group of cooperative wireless peers willing to repair neighbors' dropped message, the problem is: how to schedule rounds of retransmissions within a group, so that the time required to complete repair to all peers is minimized? Care must be taken so that pre-specified sets of interfering transmission links are not activated simultaneously. We call this problem the *Cooperative Peer-to-peer Repair* problem (*CPR*).

In this paper, we present the following theoretical result: *CPR* and its many variants are NP-hard. We first formalize the unicast variant of *CPR*, *CPR-u*, in Section II. We then present the NP-hardness proof for *CPR-u* in Section III. We present the broadcast variant of *CPR*, *CPR-b*, in Section IV,

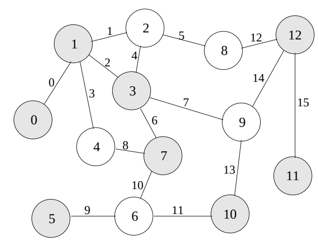


Fig. 1. Example of Connectivity Graph for Cooperative Peer-to-peer Repair

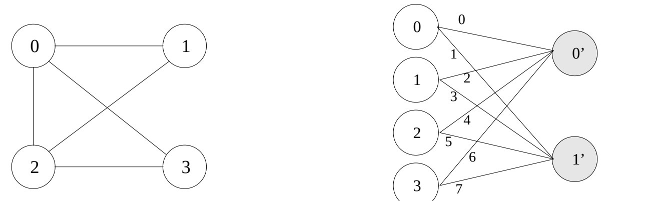
and its NP-hardness proof in Section V. Other variants are discussed in Section VI, and we conclude in Section VII.

II. Cooperative Peer-to-peer Repair: Unicast Mode (*CPR-u*)

We formulate the unicast variant of *CPR*, *CPR-u*, as follows. A connected graph Θ, modeling the connectivity of wireless peers in WLAN, has a set of nodes $\mathcal{N}$ and a set of *undirected* links $\mathcal{L}$. Links are labeled from $0, \ldots, |\mathcal{L}| - 1$, where link i connecting nodes m and n is represented by $i \leftrightarrow (m, n)$. At start time $t = 0$, each node $n \in \mathcal{N}$ has color $C_{0,n} \in \{0, 1\}$, where 0 (*blue*) means node n is in need of the desired message, and 1 (*white*) means the node has the message. As done in [3], a *conflict matrix* $\mathbf{I}$ of dimension $|\mathcal{L}| * |\mathcal{L}|$ dictates which links cannot be activated at the same time due to interference; in particular, $I_{i,j} = 1$ if link i and j cannot be activated simultaneously, and $I_{i,j} = 0$ otherwise. Matrix $\mathbf{I}$ is by definition symmetric. We assume $\mathbf{I}$ has the *unicast conflict property*: assignments of 1's and 0's so that two links stemming from the same node are in conflict. This is in compliance with standard 802.11 MAC behavior for unicast mode, where a node can be in communication with at most one other node at the same time.

At each iteration t, we select links, each connecting a white node to a blue node, such that no two selected links are in conflict according to $\mathbf{I}$. By next iteration $t + 1$, blue nodes of the selected links have received the desired message and changed color to white. The optimization problem is: how to select a set of non-conflicting links in each iteration, so that all nodes are white in the minimum number of iterations? Figure 1 shows an example of connectivity graph $\Theta = \{\mathcal{N}, \mathcal{L}\}$ with initial coloring for *CPR-u*.

We write *CPR-u* mathematically as follows. Let $\mathbf{S}$ be a $T * |\mathcal{L}|$ 0-1 *link selection matrix*, where $S_{t,l} = 1$ if link l is selected at iteration t and $S_{t,l} = 0$ otherwise, and T is the total number of iterations. Let $\mathbf{C}$ be a $T * |\mathcal{N}|$ 0-1 *color matrix* where $C_{t,n} = 1$ if node n is white at iteration t and $C_{t,n} = 0$



a) *IndS* with indep. set $\{1, 3\}$ b) Mapping of *IndS* to *CPR-u*

Fig. 2. Example Construction for NP-Completeness Proof for *CPR-u*

otherwise. Given the first row of $\mathbf{C}$ is initialized to the starting colors of $\mathcal{N}$, the optimization is:

$$\begin{array}{rl} \min_{\mathbf{S},\mathbf{C}} \;\; row(\mathbf{S}) & \text{s.t.} \\ I_{j,k} = 0 & \forall j,k \mid S_{t,j} = S_{t,k} = 1 \\ C_{t,m} + C_{t,n} = 1 & \forall l \mid S_{t,l} = 1, \; l \leftrightarrow (m,n) \\ C_{t+1,m} + C_{t+1,n} = 2 & \forall l \mid S_{t,l} = 1, \; l \leftrightarrow (m,n) \\ C_{t+1,n} = C_{t,n} & \nexists l \mid S_{t,l} = 1, \; l \leftrightarrow (m,n) \\ \sum_n C_{row(\mathbf{S}),n} = |\mathcal{N}| & \end{array} \tag{1}$$

where $row(\mathbf{S})$ is the number of rows in matrix $\mathbf{S}$. 1^{st} constraint in (1) states that no two links selected in the same iteration t should be in conflict. 2^{nd} constraint states that only one node of each selected link at iteration t should be white. 3^{rd} constraint states that both nodes of a selected link at iteration t should be white at iteration $t+1$. 4^{th} constraint states that color of a node stays the same at iteration $t+1$ if no link connected to it was selected at iteration t. 5^{th} constraint states that all nodes must be white at iteration $row(\mathbf{S})$.

We now present the NP-hardness proof for *CPR-u*.

III. Proof of NP-Hardness: *CPR-u*

We first recast *CPR-u* as a decision problem: is there a schedule of non-conflicting links at each iteration, such that all nodes can be whitened in κ iterations? The *CPR-u* decision problem is obviously in NP; a solution $(\mathbf{S}^o, \mathbf{C}^o)$ can be checked against constraints in (1), and $row(\mathbf{S}^o)$ against κ, for feasibility in polynomial time.

We next show that the *CPR-u* decision problem is NP-complete via polynomial transformation from a well-known NP-complete problem *Independent Set* (*IndS*). The *IndS* decision problem can be stated[1] as follows (pg.361 of [4]):

> Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and an integer k, is there a set $\mathcal{I} \subset \mathcal{V}$ of k vertices such that no two vertices in $\mathcal{I}$ are connected by an edge?

Figure 2a shows an *IndS* example with independent set $\{1, 3\}$. We now describe a procedure to construct a *CPR-u* instance from an arbitrary *IndS* instance, so that the output of the *CPR-u* decision problem corresponds exactly to the decision in *IndS*, and hence proving that *CPR-u* is as hard as *IndS*.

A. Construction of CPR-u Instance from IndS

We first construct a *IndS conflict 0-1 matrix* $\mathbf{J}$ of size $|\mathcal{V}| * |\mathcal{V}|$, where $J_{i,j} = 1$ if $\exists e_{i,j} \in \mathcal{E}$ and $J_{i,j} = 0$ otherwise. In other words, $J_{i,j} = 1$ iff vertices i and j cannot be selected to the same independent set because they are connected by an edge. A *IndS* conflict matrix $\{J_{i,j}\}$, $0 \leq i, j \leq 3$, corresponding to the *IndS* instance in Figure 2a is:

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

We next construct a corresponding *CPR-u* instance with graph $\Theta = (\mathcal{N}, \mathcal{L})$ and conflict matrix $\mathbf{I}$. More specifically, we construct a bipartite graph with $|\mathcal{V}|$ nodes on the left ($\mathcal{N}_l$) and k nodes on the right ($\mathcal{N}_r$), so that $\mathcal{N}_l \cup \mathcal{N}_r = \mathcal{N}$. Nodes on the left, each labeled $n \in \{0, \ldots, |\mathcal{V}| - 1\}$, are white, and nodes on the right, each labeled $m' \in \{0', \ldots, (k-1)'\}$, are blue. We draw a link from left to right for every pair of left-node and right-node. We label a link $l \in \{0, 1, \ldots, k|\mathcal{V}| - 1\}$ connecting $n \in \mathcal{N}_l$ and $m' \in \mathcal{N}_r$ as follows:

$$l = n * k + m \leftrightarrow (n, m') \quad n \in \mathcal{N}_l, \; m' \in \mathcal{N}_r \tag{3}$$

The constructed *CPR-u* instance for our *IndS* example is shown in Figure 2b. To complete the *CPR-u* instance, we construct a *CPR-u* conflict matrix $\mathbf{I}$ of size $k|\mathcal{V}| * k|\mathcal{V}|$ from $\mathbf{J}$ as follows. For $0 \leq i, j \leq k|\mathcal{V}| - 1$:

$$I_{i,j} = \begin{cases} 1 & \text{if } J_{\lfloor \frac{i}{k} \rfloor, \lfloor \frac{j}{k} \rfloor} = 1 \\ 1 & \text{else if } (\lfloor \frac{i}{k} \rfloor \neq \lfloor \frac{j}{k} \rfloor) \;\&\; (i \bmod k = j \bmod k) \\ 1 & \text{else if } (\lfloor \frac{i}{k} \rfloor = \lfloor \frac{j}{k} \rfloor) \;\&\; (i \neq j) \\ 0 & \text{o.w.} \end{cases} \tag{4}$$

where $i \bmod k$ gives the integer remainder of i divided by k.

The *CPR-u* conflict matrix $\mathbf{I}$ corresponding to the *IndS* conflict matrix $\mathbf{J}$ in (2) is as follows.

$$\mathbf{I} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{5}$$

Given the constructed *CPR-u* instance, the corresponding decision is: is there a schedule of non-conflicting links, such that all blue nodes can be whitened in 1 iteration?

B. Remarks

First, we show that the constructed *CPR-u* instance is of polynomial size of the *IndS* instance. The node set $\mathcal{N}$ of bipartite graph for *CPR-u* is of size $|\mathcal{V}| + k \leq \mathbf{O}(|\mathcal{V}|)$. The number of links is bounded by $\mathbf{O}(|\mathcal{V}|^2)$, and the size of the *CPR-u* conflict matrix $\mathbf{I}$ is bounded by $k^2|\mathcal{V}|^2 \leq \mathbf{O}(|\mathcal{V}|^4)$. Hence we conclude that the size of the constructed *CPR-u* instance is $\mathbf{O}(|\mathcal{V}|^4)$, i.e. it is of polynomial size of the *IndS* instance.

Next, we show that the 1's and 0's assigned to $\mathbf{I}$ using (4) satisfies the unicast conflict property. From link labeling (3), we see that links stemming from the same left-node n but arriving at different right-nodes are in conflict with each other

[1]Notice we adopt the terminology of *nodes* and *links* when referring to *CPR-u*, and *vertices* and *edges* when referring to *IndS* to avoid confusion.

due to the 3^{rd} `if` statement of (4). Similarly, from (3), we see that links arriving at the same right-node m' but stemming from different left-nodes are also in conflict with each other due to the 2^{nd} `if` statement of (4). Since this covers all links, we conclude that the constructed $\mathbf{I}$ satisfies the unicast conflict property.

Finally, We discuss the intuition behind the construction of the *CPR-u* instance. Each node $\in \mathcal{N}_l$ of *CPR-u* corresponds one-to-one to a vertex $\in \mathcal{V}$ of *IndS*. Selecting one of k links stemming from a node $\in \mathcal{N}_l$ of *CPR-u* means selecting the corresponding vertex $\in \mathcal{V}$ of *IndS* into the independent set. The 1^{st} `if` statement in (4) prevents selection of links stemming from two nodes representing vertices $\in \mathcal{V}$ that are connected in $\mathcal{G}$.

Each node $\in \mathcal{N}_r$ corresponds to a unique, successful selection of an independent vertex $\in \mathcal{V}$. We track to see if all k nodes $\in \mathcal{N}_r$ can be whitened in 1 iteration. There is no under-counting, since the unicast conflict property prevents joint selection of links stemming from two nodes $\in \mathcal{N}_l$ going to the same node $\in \mathcal{N}_r$. There is also no over-counting, because the unicast conflict property also prevents joint selection of links stemming from the same node $\in \mathcal{N}_l$ going to different nodes $\in \mathcal{N}_r$.

We now state the proof formally as a theorem.

Theorem 1: The *CPR-u* decision problem is NP-complete.

Proof: We prove the theorem by showing that "yes" to the constructed *CPR-u* instance implies "yes" to the original *IndS* instance, and vice versa. Suppose the output of the constructed *CPR-u* decision problem is "yes". That means the solution composes of *exactly* k non-conflicting links, originating from k *distinct* left-nodes $\in \mathcal{N}_l$ and terminating at all k blue right-nodes $\in \mathcal{N}_r$. We know there are *exactly* k selected links, because all k blue right-nodes $\in \mathcal{N}_r$ are whitened in 1 iteration, and links terminating at the same right-node $\in \mathcal{N}_r$ are conflicting due to the unicast conflict property. We know these k links originated from k *distinct* left-nodes because links originated from the same left-node $\in \mathcal{N}_l$ are conflicting, again due to the unicast conflict property. Finally, these distinct k left-nodes $\in \mathcal{N}_l$ of the k selected links must correspond to k independent vertices $\in \mathcal{V}$ in *IndS* due to (3) and 1^{st} `if` statement of (4). Therefore, we conclude that "yes" to the *CPR-u* decision problem corresponds to "yes" in the original *IndS* decision problem.

Suppose there is an independent set $\{v_0^o < \ldots < v_{k-1}^o\} \subset \mathcal{V}$ of size k in the *IndS* instance. We can correspondingly select a set of k non-conflicting links $\in \mathcal{L}$, where for each v_i^o, we pick link $v_i^o * k + i$. First, we know each of these k links connect to a different right-node $\in \mathcal{N}_r$ due to (3). Second, we know each of these k links connects to a different left-node $\in \mathcal{N}_l$, again due to (3). Hence, given each of these k links has a different left-node and a different right-node, they do not violate unicast conflicts (2^{nd} and 3^{rd} `if` statements in (4)). They also do not conflict due to 1^{st} `if` statement of (4), since vertices v_i^o's, where $v_i^o = \left\lfloor \frac{v_i^o * k + i}{k} \right\rfloor$, do not conflict in $\mathcal{G}$ by assumption. Therefore, this set of k non-conflicting links $\in \mathcal{L}$ converts all k blue nodes $\in \mathcal{N}_r$ in 1 iteration. We can now conclude that since both directions of the implication have been proven, Theorem 1 is also proven. ∎

Corollary 1: The *CPR-u* optimization problem is NP-hard.

Proof: Given the *CPR-u* decision is NP-complete as stated in Theorem 1, it follows by definition of NP-hardness [4] that the corresponding optimization is NP-hard. ∎

IV. Cooperative Peer-to-peer Repair: Broadcast Mode (*CPR-b*)

A different variant of the *CPR* problem can be formulated where *nodes* are selected instead of *links* at each iteration t. Using broadcast mode of 802.11 MAC, each selected node *locally broadcasts* to all directly connected nodes so that they all receive the message in one iteration. Instead of using an *undirected* graph to model peer-to-peer connectivity as done for *CPR-u* in Section II, we use here a *directed* connectivity graph. The reason is the following: while a node i can broadcast to nodes j and k at the same time, node i cannot listen to both nodes j and k simultaneously. Hence there should be no conflict for links[2] connecting $[i,j]$ and $[i,k]$, while there is a conflict between $[j,i]$ and $[k,i]$. Only a directed graph can model the asymmetric relationship.

Even with the minor caveat of using directed links instead, the optimization remains similar. The new variant we call *CPR-b*. The optimization variable is now $\mathbf{S}^*$, a $T * |\mathcal{N}|$ 0-1 node selection matrix, where $S^*_{t,n} = 1$ if node n is selected for local broadcasting at iteration t and $S^*_{t,n} = 0$ otherwise. Color matrix $\mathbf{C}$ is as defined earlier. The conflict matrix $\mathbf{I}$ will now have the *broadcast conflict property*: links starting from the same node cannot be in conflict, so that conflict-free broadcasting by any node is possible. The optimization is:

$$\begin{array}{rl}
\min_{\mathbf{S}^*,\mathbf{C}} \;\; row(\mathbf{S}^*) & \text{s.t.} \\
I_{j,k} = 0 & \forall j,k \mid S^*_{t,m} = S^*_{t,n} = 1, \\
 & \quad j \leftrightarrow [m,p], k \leftrightarrow [n,q] \;\; \forall p,q \in \mathcal{N} \\
C_{t,n} = 1 & \forall n \mid S^*_{t,n} = 1 \\
C_{t+1,m} + C_{t+1,n} = 2 & \forall l \mid S^*_{t,n} = 1, \; l \leftrightarrow [n,m] \\
C_{t+1,m} = C_{t,m} & \nexists l \mid S^*_{t,n} = 1, \; l \leftrightarrow [n,m] \\
\sum_n C_{row(\mathbf{S}^*),n} = |\mathcal{N}| &
\end{array} \tag{6}$$

Constraints in (6) are similarly defined as ones in (1). 1^{st} constraint in (6) states that links starting from two selected nodes at iteration t cannot be in conflict. 2^{nd} constraint states that each selected node in iteration t must be white. 3^{rd} constraint states that the destination node m and corresponding selected source node n at iteration t must both be white in iteration $t+1$. 4^{th} constraint states that color of a node stays the same at iteration $t+1$ if no nodes with links destined to it was selected at iteration t. 5^{th} constraint states that all the nodes must be white by iteration $row(\mathbf{S}^*)$.

We now present the NP-hardness proof for *CPR-b*.

V. Proof of NP-Completeness: *CPR-b*

We similarly recast *CPR-b* as a decision problem: is there a schedule of nodes with non-conflicting links at each iteration, such that all nodes can be whitened in κ iterations? It is again obvious that the *CPR-b* decision problem is in NP; a solution $(\mathbf{S}^{*o}, \mathbf{C}^o)$ can be checked against constraints in (6), and $row(\mathbf{S}^{*o})$ against κ, for feasibility in polynomial time.

[2] We use square brackets to denote directed links.

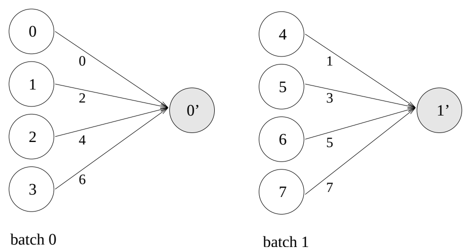


Fig. 3. Example Construction for NP-Completeness Proof for *CPR-b*

We again prove NP-completeness of *CPR-b* via polynomial transformation from *IndS*. The construction of a *CPR-b* instance from an arbitrary *IndS* instance is very similar to *CPR-u*'s. Instead of having $|\mathcal{V}|$ nodes in $\mathcal{N}_l$, we replicate k batches of $|\mathcal{V}|$ nodes each as shown in Figure 3. In this construct, left-node $i \in \mathcal{N}_l$ corresponds to vertex $i \bmod |\mathcal{V}| \in \mathcal{V}$. As done previously, there are k right-nodes in $\mathcal{N}_r$. We draw a link from every left-node in batch i to the i^{th} right-node $\in \mathcal{N}_r$. We label the links as:

$$l = (n \bmod |\mathcal{V}|) * k + m \leftrightarrow [n, m'] \quad n \in \mathcal{N}_l,\ m' \in \mathcal{N}_r \qquad (7)$$

Using the same (4), we create a *CPR-b* conflict matrix $\mathbf{I}$. Since each node has at most out-degree 1, $\mathbf{I}$ satisfies the broadcast conflict property trivially.

The corresponding decision problem is similar: is there a schedule of nodes with non-conflicting links, such that all blue nodes are whitened in 1 iteration? We note that the constructed *CPR-b* instance can again be easily shown to be of polynomial size of the *IndS* instance, as done previously for the constructed *CPR-u* instance.

Theorem 2: The *CPR-b* decision problem is NP-complete.

Proof: Suppose the output of the *CPR-b* decision problem is "yes". Then for each batch i of k, there is a unique left-node $n_i^o \in \mathcal{N}_l$ selected to transmit to right-node $i' \in \mathcal{N}_r$; uniqueness of left-node in each batch is guaranteed by conflicts of links terminating at each right-node $i' \in \mathcal{N}_r$, due to (7) and 2^{nd} `if` statement of (4). For each n_i^o, we select a vertex $\in \mathcal{V}$, where $v_i^o = n_i^o \bmod |\mathcal{V}|$. First, we know these k vertices v_i^o's are distinct due to (7) and 3^{rd} `if` statement of (4). Second, we know they do not conflict due to 1^{st} `if` statement of (4). Hence there exists an independent set of size k in *IndS*.

Suppose there exists an independent set $\{v_0^o < \ldots < v_{k-1}^o\} \subset \mathcal{V}$ in *IndS* instance of size k. We select k left-nodes $\in \mathcal{N}_l$, where for each v_i, we pick left-node $n_i^o = i * |\mathcal{V}| + v_i^o \in \mathcal{N}_l$ in batch i. Subsequently, each left-node n_i^o has link $v_i^o * k + i$ connected to right-node $i' \in \mathcal{N}_r$ due to (7). We see that these k links do not conflict: not due to 1^{st} `if` statement of (4), since $\left\lfloor \frac{v_i^o * k + i}{k} \right\rfloor = v_i^o$ and v_i^o's do not conflict by assumption; not due to 2^{nd} `if` statement of (4), since each link goes to a different right-node m' and links are labeled using (7); not due to 3^{rd} `if` statement of (4), since $\left\lfloor \frac{v_i^o * k + i}{k} \right\rfloor = v_i^o$ and v_i^o's are distinct by assumption. Therefore, all k blue nodes in *CPR-b* are whitened in 1 iteration. Therefore, "yes" in the *IndS* instance implies "yes" in *CPR-b* instance. Since both directions of the implication are proven, Theorem 2 is also proven. ■

Corollary 2: The *CPR-b* optimization problem is NP-hard.

VI. Other Variants of Cooperative Peer-to-peer Repair

Given both the unicast mode and the broadcast mode of *CPR* are both NP-hard, it is easy to show more general settings which include one or both of the proven cases as special cases are also NP-hard. We discuss two of these here.

A. CPR: Multicast Mode

One can imagine a *multicast* mode for *CPR*, which is like the unicast mode *CPR-u*, with the generalization that at a given time, a node can *multicast* to any subset of directly connected nodes that are listening simultaneously. In other words, it has the same formulation (1) as *CPR-u* — selecting non-conflicting links at each iteration — without the requirement of unicast conflict property on conflict matrix $\mathbf{I}$. It is obvious that the multicast mode includes the unicast mode as a special case — one where conflict matrix $\mathbf{I}$ satisfies the unicast conflict property, hence the multicast mode is also NP-hard.

B. CPR: Directed Connectivity Case

It is possible to construct a peer-to-peer connectivity graph for the unicast and multicast modes using a *directed* graph instead of the *undirected* graph described in Section II. Note, however, that typical 802.11 MAC unicast behavior dictates that a node n_i can communicate with another node n_j only if n_j can also communicate with n_i (so that ACK packets can be properly returned), and this "$n_i \rightarrow n_j$ implies $n_j \rightarrow n_i$" property for node connectivity is modeled best by an undirected graph. Nevertheless, a more general notion of directed connectivity is in theory possible, as done in [3]. We emphasize that this is the general case, since a undirected link between n_i and n_j can be modeled by two directed links to and from n_i and n_j, but not vice versa. Given the undirected connectivity cases for unicast and multicast modes are NP-hard, the directed connectivity cases for unicast and multicast modes are also NP-hard.

VII. Conclusion

In this paper, we formulate the Cooperative Peer-to-peer Repair problem (*CPR*) as a discrete optimization problem, and show that the problem and its many variants are NP-hard.

References

[1] *Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs (3GPP TS.26.346 version 6.1.0*, June 2005.

[2] P. Sharma, S.-J. Lee, J. Brassil, and K. Shin, "Distributed communication paradigm for wireless community networks," in *IEEE International Conference on Communications*, Seoul, Korea, May 2005.

[3] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom 2003)*, San Diego, CA, September 2003.

[4] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, 1998.