Efficient Coding for Unicast Flows in Opportunistic Wireless Networks

Muhammad Awais, and Jalaluddin Qureshi*

Department of Electrical Engineering, Namal College,
30 KM Talagang Road, Mianwali 42250, Pakistan.
* Email: jala0001@e.ntu.edu.sg

Abstract: In this paper we consider the scenario of multiple unicast flows intersecting a common router in an opportunistic wireless network. Instead of forwarding packets in each of the flows independently, the router can perform inter-session network coding and transmit codewords to improve the network throughput. Unlike coding for multicast data flow for which an optimal code can be constructed in polynomial time, coding for unicast data flows is a more complicated coding problem and has been shown to be an NP-hard problem. Opportunities for inter-session network coding have also been shown to exist in single-hop wireless data dissemination network such as Wi-Fi and WiMAX networks. In this paper we propose an efficient coding scheme for unicast flows and demonstrate its higher coding gain over previously proposed state of the art coding schemes, validated using simulation results and wireless sensor network testbed packet reception traces. We also show that our proposed algorithm is optimal for all 238 non-isomorphic coding instances for $n \leq 4$, and for 9500 of the 9608 non-isomorphic coding instances for $n = 5$, where $n$ is the number of unicast packets.

1. Introduction

Wireless transmissions are inherently broadcast in nature, where non-intended clients can overhear unicast transmissions. In an opportunistic wireless network, a node can cache such overheard transmissions. Such opportunistically cached packets provide opportunity to perform network coding when multiple unicast flows intersect a relay router. Examples of topologies where inter-session network coding can be performed is illustrated in Figure 1 [1]. In the figure it can be seen that by transmitting $p_1 \oplus p_2$, the relay router $R$ can save one transmission in both the topologies.

The idea of network coding deviates from the traditional routing approach of “store-and-forward” to one where the router can appropriately “mix-and-forward” the data. In network coding, packets from different sources are encoded before being forwarded. Network coding technique offers several benefits, such as increase in network throughput and improvement in robustness to packet losses in wireless networks [2].

Network code can also be used for erasure correction by base station (BS) in WiMAX and access point (AP) in Wi-Fi single-hop networks as multiple unicast flows traverse the common last mile wireless broadcast link [3, 4, 5]. Consider the example illustrated in Figure 2, where the AP is unicasting packets $p_1$ and $p_4$ to $c_1$, $p_2$ to $c_2$, and $p_3$ to $c_3$. Assuming that due to independent packet losses and broadcast nature of wireless transmission, after the AP has transmitted $p_1$, $p_2$, $p_3$
Fig. 1. (a) “X” topology with two path flows, \( s_1 \rightarrow R \rightarrow c_2 \), and \( s_2 \rightarrow R \rightarrow c_1 \). Node \( c_1 \) and \( c_2 \) can overhear \( s_1 \) and \( s_2 \) transmissions respectively. (b) Alice and Bob relay topology with two paths, Alice \( \rightarrow R \rightarrow \) Bob and Bob \( \rightarrow R \rightarrow \) Alice. In both the topologies the relay router \( R \) can encode \( p_1 \) and \( p_2 \), broadcast \( p_1 \oplus p_2 \) and save one transmission.

and \( p_4 \), client \( c_1 \) has overheard packets \( p_2 \) and \( p_3 \) and the requested packets \( p_1 \) and \( p_4 \) are erased, \( c_2 \) has overheard \( p_1 \) and \( p_2 \) is erased, and \( c_3 \) has overheard \( p_4 \) and \( p_3 \) is erased. To correct the erasures, the AP will need to transmit at least four packets using the automatic repeat request (ARQ) protocol, whereas with network coding, the AP can correct the erasures with a minimum of two transmissions given by the codewords \( p_1 \oplus p_2 \) and \( p_3 \oplus p_4 \).

While network code which can achieve the multicast capacity of a network can be constructed in polynomial time, the throughput benefits of coding for unicast flows however has not yet been completely characterized. It is now widely assumed that the theoretical foundation of network coding for multicasting is getting closer to maturity, while many interesting open problem remain in network coding for unicast flows [5].

Unlike the distributed random linear (RL) code which can be used for multicast network coding, network code for unicast flows is a more complicated problem [2]. In RL code, the coding coefficient is randomly selected from a finite field \( GF(q) \). It has been shown that the problem of constructing network code of minimum length for unicast flows is an NP-hard problem as the encoding process needs to generate the minimum possible number of codewords under the constraint of unique packet requests and different overheard packets by each client [6].

It also remains an open problem to determine whether the length of such optimal code can be approximated within a multiplicative constant of \( n^{1-\epsilon} \) for some fixed \( \epsilon > 0 \), where \( n \) is the number of unicast packets [7]. In another paper it has been shown that for an index coding instance, even with the a-priori knowledge that the optimal scalar solution of the problem is three index codewords, constructing those three index codewords is an NP-hard problem [8].

Motivated by the complexity of the problem with interesting open questions, in this paper we study the problem of constructing efficient code for unicast flows in an opportunistic wireless networks. We show that our proposed coding scheme outperforms previously proposed state of the art coding schemes for unicast flows [1, 4, 9, 10, 11]. The higher coding gain of our coding scheme is validated using simulation results and TinyOS based wireless sensor network (WSN) testbed packet reception data. We also show that our proposed algorithm is optimal for all the 238 non-isomorphic coding instances for \( n \leq 4 \), and for 9500 of the 9608 non-isomorphic coding instance for \( n = 5 \) [12].

The paper is organized as follow. We first present formulation of the coding problem in Section 2. Bibliography of related works is presented in Section 3. We then present our proposed
scheme along with a motivating example, pseudocode and computational complexity analysis in Section 4. Results of simulation and testbed packet traces based evaluation is presented in Section 5. We then conclude with the main results of our paper in Section 6.

As this paper extensively uses graph theory terms and graph based optimization problems, we refer interested readers to the textbooks [13,14] and the appendix in [10] for introduction of these terms and optimization problems.

2. Problem Formulation

In an index coding with side information (ICSI) problem a server with a set of \( n \) packets \( \mathcal{P} = \{p_1, p_2, ..., p_n\} \) satisfies the transmission requests of \( n \) clients \( \mathcal{C} = \{c_1, c_2, ..., c_n\} \) connected to the server over a common lossless broadcast channel. The unique packets which each client has requested is known as the “want set” of \( c_i \), \( c_i \in \mathcal{C} \), and given as \( W_i \subseteq \mathcal{P} \). The server has the side information knowledge, which is the set of packets each client \( c_i \) has cached and known as the “has set” \( H_i \subseteq \mathcal{P} \setminus W_i \). The objective of the ICSI problem is to minimize the total number of transmissions \( \ell \), while satisfying the requests of all the clients [6].

For the index coding problem it is assumed that the unicast data flow is represented by a singleton want set, \( W_i = \{p_i\} \), this as shown by Lemma 1 does not affect the result for the unicast transmission data flow assuming lossless broadcast channel.

**Lemma 1.** For the index coding problem, without loss of generality we only need to consider the case of a single unique packet being requested by each client (single-unicast) [9, Lemma 1].

While the ICSI problem has traditionally been modelled on a lossless broadcast channel [4], a major contribution of our paper is that we also consider the more practical lossy wireless broadcast channel. In Lemma 2 we show how the singleton want set model can also be applied for index coding problem model on lossy broadcast channel, with independent packet losses.

**Lemma 2.** For the index coding problem on a lossy broadcast channel, without loss of generality...
we only need to consider the case of a single unique packet being requested by each client (single-unicast).

Proof. For the index coding problem on a lossy broadcast channel, for each client $c_i$ with $w$ requested packets, $|W_i| = w$, we create $w$ “sub-clients” such that each sub-client $c_{i,b}$ wants one unique packet $p_j, p_j \subseteq W_i$, with the same hash set as $c_i, H_{i,b} = H_i$. If a client $c_i$ positively acknowledges a codeword, then this implies that all its sub-clients $c_{i,b}$ have also received the codeword. Otherwise if client $c_i$ does not positively acknowledges a codeword, then this implies that none of its sub-clients $c_{i,b}$ have received the codeword. This completes the proof. 

A feasible code which solves the ICSI problem is known as an index code, and the set of codewords in an index code is given by the set $K$, $|K| = \ell$. We call an index code $K^*$ optimal iff $|K^*| = \min(|K|) = \ell^*$. In this paper we only consider scalar linear code over the finite field $GF(2)$. A code is scalar if the data unit, e.g. packet, cannot be split in to smaller data units, e.g. “sub-packets” for encoding. The probability $p_{has}$ is the expected probability that a client has packet $p_i$ in its has set.

In this paper we assume that the encoding relay router has knowledge of opportunistically overheard packets by each of neighbouring nodes. This can be achieved due to one of the many efficient transmission schemes to collect packet reception status as shown in [15, 16, 17].

The canonical network topologies which we consider in this paper are the “X” and wireless local area network (WLAN) topologies [1, 4]. In an “X” topology, data flows between all source-client pairs traverse a common relay router, and the client is outside the transmission range of the source. An illustration of “X” topology with two source-client pairs is given in Figure 1(a).

2.1. Side Information Graph

The ICSI problem is mapped to the side information digraphs as follow. For the side information digraph $\mathcal{D} = \{V(\mathcal{D}), E(\mathcal{D})\}$, vertex $v_i \in V(\mathcal{D})$ corresponds to packet $p_i \in W_i$. There exist a directed edge (arc) from vertex $v_i$ to $v_j$ iff $D_i$ has $p_j$, i.e. $p_j \subseteq H_i$.

In an undirected side information graph $\mathcal{G} = \{V(\mathcal{G}), E(\mathcal{G})\}$, the set of vertices is given as $V(\mathcal{G}) = V(\mathcal{D})$, and there exist an edge between $v_i$ and $v_j$ iff $p_j \in H_i$ and $p_i \in H_j$. Side information graph $\mathcal{G}$ represents the restricted class of ICSI problem given by symmetric cache content, i.e. if $D_i$ has $p_j$ then $D_j$ has $p_i$. Therefore in this paper we consider the side information digraph $\mathcal{D}$, which can represent any arbitrary instance of the ICSI problem.

2.2. ICSI and Inter-session Network Coding

Rouayheb et al. has shown that there exist a polynomial time mapping scheme such that it can be shown that the ICSI problem and inter-session network coding problem for multiple flows over multiple hop network are equivalent [18]. Therefore without loss of generality we interchangeably use the terms index coding and inter-session network coding in this paper to remain consistent with the terms used in the references cited.

3. Background and Related Work

Yossef et al. have shown that the minrank$_2(\mathcal{D})$ problem and ICSI problem are equivalent [6]. As the minrank$_2(\mathcal{A})$ has been shown to be an NP-hard problem, they hence showed that the ICSI problem is also an NP-hard problem. The objective of the minrank$_2(\mathcal{D})$ problem is to minimize the
A matrix $A$, $A \in GF(2)^{n \times n}$, fits graph $D$ if all the diagonal elements are equal to one and the element $a_{ij}$ is equal to zero if $v_i$ and $v_j$ are adjacent in $D$. Bounds on ICSI problem with symmetric cache content (represented by an undirected graph $G$) was shown to be [6],

$$\omega(\overline{G}) \leq \minrk_2(G) = \ell^* \leq \varphi(G),$$

where $\omega(\overline{G})$ is the size of maximum clique in the complementary graph $\overline{G}$, and $\varphi(G)$ is the minimum clique partition size of $G$. For an undirected graph $G$ mapped from the side information digraph $D$, it can be shown intuitively,

$$\minrk_2(D) = \ell^* \leq \minrk_2(\overline{G}) \leq \varphi(G).$$

The hardness of the ICSI problem has motivated an ongoing research to study classes of ICSI problems where $K^*$ can be constructed in polynomial time [19, 20], and approximation and heuristic algorithms to construct efficient index codes [1, 4, 9, 10, 11].

In our paper we cover both these aspects. We show that our polynomial time algorithm constructs $K^*$ for all the 238 non-isomorphic coding instances for $n \leq 4$, and for 9500 of the 9608 non-isomorphic coding instance for $n = 5$, the highest number of instances with optimal scalar solution to the best of our knowledge for $n \leq 5$. And we show that our proposed algorithm outperforms previous state-of-art coding algorithms.

### 3.1. ICSI heuristic algorithms

Minimum clique partition (also an NP-hard problem) based coding algorithms have been proposed in references [1, 4, 9] to solve the ICSI problem. In the minimum clique partition based coding class of algorithm, the ICSI problem is mapped as an undirected side information graph $G$. The vertex $v_i$ of $G$ corresponds to the packet $p_i$ requested by each client, and the adjacent vertices of $v_i$ correspond to the packets cached by $c_i$.

Codewords generated by XOR addition of packets corresponding to the vertices of a clique can be decoded by all clients corresponding to the vertices of the clique. By minimizing the number of cliques, the number of codewords can also be minimized. Note that the minimum clique partition problem on graph $G$ is equivalent to the minimum graph coloring problem on the complement graph $\overline{G}$.

Birk and Kol proposed the Least Difference Greedy (LDG) heuristic minimum clique partition algorithm [9]. In LDG, two or more rows of $A$ are merged if they form a clique in $D$. The decision about selecting two or more such rows for merging is done using a greedy search algorithm.

Kwak et al. proposed a coding algorithm based on the “row-merging” operation of LDG, and introducing the “column-merging” greedy heuristic and cycle-of-three vertices detection on $D$, which they called extended-LDG [11]. The “column-merging” heuristic is motivated by the matrix property that the row rank of matrix $A$ is equal to the column rank of $A$. For a cycle of three vertices, they used codeword which saves one transmission.

Chaudhry and Sprintson used an approximation color saving minimum clique partition algorithm, which is guaranteed to be within a factor of $\frac{2}{3}$ of the optimal minimum clique partition solution to minimize the number of index codewords [4].

Qureshi proposed a coding scheme, which they called updated clique index coding (UCIC) [10], in which a piggyback packet is added to the codeword generated by LDG (or color saving) coding scheme, such that in addition to satisfying the transmission request of some clients, the codeword
Fig. 3. Multi-hop WMN with multiple flows. A router can encode packets from different flows and broadcast the codeword, if the codeword can be decoded using some linear combination of packet cached by nodes in the downstream flow from source \( s_i \) such that the client \( c_i \) can decode its requested packet.

also increases the has set of other clients. Increasing the has set of some clients increases the coding opportunities which leads to higher coding gain. Such a coding approach was shown to outperform the LDG and color saving algorithms. The UCIC variant designed using LDG algorithm is known as UCIC-LDG, and the variant designed using the color saving algorithm is known as UCIC-color saving.

Muhammad and Schwarz study the problem of constructing efficient code for the ICSI problem where the request of all clients may not necessarily be unicast [21]. In their paper they show that the problem can be modelled using a directed bipartite graph, and an efficient algorithm is proposed to solve the problem. Dai et al. extended the ICSI problem to include instances where the transmitter and clients may have coded packets, and packet requested by a client can also be coded packet [22]. An efficient heuristic algorithm was proposed to solve the problem considered in their paper.

3.2. **Inter-session network coding aware routing**

Katti et al. in their seminal work proposed a network coding based wireless architecture for unicast flows which they called completely opportunistic coding (COPE), and practically implemented COPE on a 20 nodes 802.11 testbed [1]. It was shown that COPE significantly improves the network throughput. In COPE the coding decision is based on finding maximum clique on the undirected graph \( \mathcal{G} \), so that the request of maximum number of clients is satisfied from the transmitted codeword. In their paper description of implementing a heuristic maximum clique (NP-complete problem) algorithm has not been provided, we therefore use a greedy search algorithm in our paper.
to find solution of the maximum clique problem until the request of all clients is satisfied.

In COPE relay router uses knowledge of which packets its neighbours have overheard to perform coding, such that each of the next-hop routers can immediately decode the codeword based on the packets it has overheard. In COPE, a relay router can only perform coding based on packet overhearing data of its neighbours and hence such coding is limited for a “2-hops scenario” as illustrated in Figure 1(a). Motivated by the results of COPE, many coding aware routing protocols have been proposed for unicast data flows, a survey of which can be found in [23].

Recently Chen et al. generalized the result of COPE for multiple flows beyond the 2-hops scenario [24]. An illustration of coding multiple flows beyond the 2-hops scenario for wireless mesh network (WMN) is shown in Figure 3. Pascho et al. derived the throughput region of a single-hop “X” topology with two receivers where the encoding node has partial information of the packets each of the receivers has overheard [25].

4. Proposed Algorithm

In this section we present our proposed coding algorithm, matrix exponentiation with clique detection (MECD). In MECD we piggyback a packet which is “sparsely” connected to cycles of length \( k \) on the digraph \( D \). By adding a piggyback packet with packets corresponding to vertices which are sparsely connected to the rest of the graph, our proposed algorithm leaves the “densely” connected vertices to form cliques which minimizes transmissions. And uses sparsely connected vertices to increase the has set of clients which can form bigger cliques and lead to efficient codeword encoding for future transmissions. The foundation of our proposed algorithm is based on the following theorem.

**Theorem 1.** Let \( S \) be the adjacency matrix of a connected graph with \( n \) vertices. The \( ij^{th} \) entry of the matrix \( S^k \), \( 2 \leq k \leq n - 1 \), gives the number of paths of length \( k \) from \( v_i \) to \( v_j \) [13, Theorem 8.5].

The \( ii^{th} \) diagonal entry of \( S^k \) indicates the number of cycles of length \( k \) traversing vertex \( v_i \). In MECD we first construct an adjacency matrix \( S \) of the digraph \( D \). We then find the minimum value of \( k \) such that there exist a non-zero element in the diagonal entries of \( S^k \). The vertex \( v_\eta \) which is an element of minimum number of cycle in \( D \) of length \( k \) can then be found from the minimum value of the diagonal entries in \( S^k \).

We first discuss the pseudocode of MECD in the next section and then illustrate the steps of MECD with a simple ICSI problem instance using a motivating example.

4.1. Pseudocode

A pseudocode of our proposed algorithm MECD is given in Table 1. In MECD the algorithm first run the Kosaraju’s algorithm [14, Section 22.5] on the digraph. Kosaraju’s algorithm finds the strongly connected components (SCCs) of a graph. As shown in [20, Lemma 4.2], the \( \ell^* \) of an ICSI problem represented by digraph \( D \) is equal to the summation of optimal index code for each of the SCCs, assuming \( D \) is a connected digraph. Therefore to reduce the computation cost, it is preferable to run the MECD algorithm on smaller subgraph given by the SCCs.

For each SCC we construct the adjacency matrix \( S \), and calculate \( S^k \) such that there exist a non-zero element in the diagonal entries of \( S^k \) and \( k \) is minimized. For such \( S^k \), we then find the \( j^{th} \) index of the diagonal entry with minimum value, when two or more such diagonal entry exist.
Table 1: MECD Pseudocode.

<table>
<thead>
<tr>
<th>Input</th>
<th>Side information $H_i$ and packet requested $W_i$ by each client.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Index code $K$.</td>
</tr>
</tbody>
</table>

Find strongly connected components of $D$ using Kosaraju’s algorithm. **Output:** Strongly connected components $SCC_h$.

for $\forall SCC_h$

% The do while loops runs until the packet request for all clients % corresponding to vertices in the $SCC_h$ is not satisfied.

**do while** $W_i \neq \emptyset$, $\forall W_i : v_i \in SCC_h$

construct adjacency matrix $S$ of $SCC_h$
calculate the diagonal entries of $S^k$

$k = 2$

while $(s_i^k = 0, \forall s_i^k), 1 \leq i < |V(SCC_h)|$

$k \leftarrow k + 1$
calculate $S^k$

$j \leftarrow \arg\min_i (s_i^k)$,  % arg max$_t$($H_t$) used as a tie-breaker.

$x_a = p_i$ % $x_a$ denotes the generated codeword.

if $s_{jj}^2 = 0$ or $k > 2$

% In this if condition we do not search for a clique as $v_j$ does % not form a clique with any vertex. Search piggyback packet % which increases the has set of maximum unsatisfied clients.

$p_{pb} = \text{search piggyback packet ( )}$

$x_a \leftarrow x_a \oplus p_{pb}$

else if $s_{jj}^2 > 0$

for $1 \leq g \leq 3$

Exhaustive search of $g$ vertices given by set $B$ which form

(g+1)-vertices clique with $v_j$ such that $g$ is maximized.

% All the $g$ packets corresponding to the vertices of the clique % are XOR coded with $p_j$ (where $p_j = x_a$).

$x_a \leftarrow \bigoplus_{p_f : v_f \in B} p_f \oplus x_a$

$p_{pb} = \text{search piggyback packet ( )}$

$x_a \leftarrow x_a \oplus p_{pb}$

% The transmission channel can be lossy broadcast channel or % lossless broadcast channel.

transmit $x_a$

Add arc $(v_x, v_y)$ if $c_x$ can decode $p_y$ from $x_a$.

$H_z \leftarrow H_z \cup p_y$

$W_j = \emptyset$  % If the packet request of $c_j$ is satisfied.

Prune the arcs adjacent to vertices of clients $c_j$ and/or $c_f$ whose requests has been satisfied.

% Packets corresponding to vertices which are not element of any % SCCs are transmitted uncoded.

for $\forall v_r : v_r \notin SCC_h, \forall SCC_h$

transmit $p_r$.
with minimum value, then as a tie-breaker we select the \( t^{th} \) index such that \( H_t \) has the maximum value of all minimum diagonal entries \( s_{ii} \).

If \( s_{jj}^k \) is equal to zero then this implies that vertex \( v_j \) is not an element of any cycle of length two in \( D \), which implicitly also implies that \( v_j \) does not form clique of size at least 2-vertices with any vertex. Similarly if \( k > 2 \), then this also implies that there does not exist any clique of size at least 2-vertices in the graph. Therefore it is sufficient to only search for a piggyback packet \( p_{pb} \), \( p_{pb} \in H_j \), using greedy search algorithm, which when XOR added with \( p_j \) will increase the has set of maximum number of clients.

If \( s_{jj}^2 \) is greater than zero, then this implies that \( v_j \) forms a clique of size at least 2-vertices with some other vertices. We then run an exhaustive search algorithm which finds a clique of maximum size of up to 4-vertices such that \( v_j \) is an element of the 4-vertices clique. The packets corresponding to the vertices of the clique are then XOR coded. A piggyback packet is then searched which will be in the has set of all clients whose vertices form clique, and which increases the has set of maximum number of clients. The piggyback packet is then XOR added to the codeword.

The algorithm adds the arc \((v_z, v_y)\) if \( c_z \) can decode \( p_y \) from \( x_a \), and update the has set of \( c_z \). The adjacent arcs of client \( c_j \) whose request has been satisfied is pruned from the digraph, and in any subsequent iteration we ignore the \( j^{th} \) row and column of \( S \) in our algorithm. Finally, we transmit uncoded packets corresponding to all vertices which are not an element of any SCCs from the output of Kosaraju’s algorithm. Note that a greedy search algorithm may not necessarily be able to find a piggyback packet in all instances. When a piggyback packet cannot be found, then \( x_a \) is transmitted without XOR adding any piggyback packet with it.

4.2. Motivating Example

Consider the ICSI problem where \( W_i = \{p_i\} \), \( H_1 = \{p_2\} \), \( H_2 = \{p_3, p_5\} \), \( H_3 = \{p_1, p_5\} \), \( H_4 = \{p_2, p_3\} \) and \( H_5 = \{p_4\} \). The adjacency matrix for digraph \( D \) of this instance is given as,

\[
S = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}.
\]

As no clique of size greater than 1-vertex exist in the digraph \( D \), LDG, color saving, and COPE like any other minimum clique partition and maximum clique based index coding algorithm would partition \( D \) in five cliques which correspond to five transmissions.

UCIC will need four transmissions for this problem instance. As the transmission of codeword \( p_1 \oplus p_3 \) or \( p_4 \oplus p_5 \) will both result in satisfying the request of one client, and increase the has set of two clients. Let assume UCIC arbitrarily chooses \( p_1 \oplus p_3 \) as it does not use any tie-breaker when two or more piggyback packet increment the has set of same number of clients. Note that, had UCIC chosen \( p_4 \oplus p_5 \) it would still not lead to reduction in transmissions. Assuming that UCIC transmits \( p_1 \oplus p_3 \), it would then need to transmit \( p_2 \oplus p_4 \), \( p_1 \oplus p_2 \) and \( p_5 \) to satisfy the request of all clients as partially illustrated in Figure 4.

As there is no clique in \( D \) for this problem instance, extended-LDG will search for a cycle-of-three vertices. Assuming it considers cycle-of-three vertices given by path \( v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_1 \), extended-LDG will transmit \( p_1 \oplus p_2 \) and \( p_2 \oplus p_3 \), and for the remaining unsatisfied clients \( p_4 \) and \( p_5 \) will be transmitted, resulting in a total of four transmissions.
**Fig. 4.** Index coding using UCIC algorithm. (a) The original digraph $D$ for the ICSI problem instance in the motivating example. (b) The updated digraph after the transmission of $p_1 \oplus p_3$. (c) The updated digraph after the transmission of $p_2 \oplus p_4$.

We now show the solution constructed by MECD. First, we calculate $S^2$ as follow,

$$S^2 = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 2 \\
0 & 1 & 1 & 0 & 0
\end{bmatrix}.$$

As all the diagonal entries are zero, then this implies that there is no cycle of path length two. Thus, we proceed to calculate $S^3$, given as follow,

$$S^3 = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 \\
1 & 2 & 1 & 1 & 0 \\
0 & 1 & 2 & 0 & 1 \\
1 & 1 & 0 & 2 & 1 \\
1 & 0 & 1 & 0 & 2
\end{bmatrix}.$$

In $S^3$ there are non-zero elements in the diagonal entries. We select packet $p_1$ as $s^3_{11}$ has the minimum value of all diagonal entries in $S^3$. We then search for a piggyback packet and select $p_2$. The codeword $p_1 \oplus p_2$ will satisfy the request of $c_1$ and increase the has set $H_2$ as $H_2 = \{p_1, p_5\}$. The adjacency matrix of the updated graph will now be given as follow,

$$S = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}.$$

And the corresponding matrix $S^2$ is given as,

$$S^2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 2 \\
0 & 1 & 1 & 0 & 0
\end{bmatrix}.$
Now we can either choose $p_4$ or $p_5$ as the entries $s_{44}^2$ and $s_{55}^2$ both have minimum value of all diagonal entries. As both $p_4$ and $p_5$ are not an element of any clique of size 2-vertices we search for piggyback packet. For each of the packets $p_4$ and $p_5$ the algorithm greedy searches for a piggyback packet which will increase the has set of maximum number of clients. Codeword $p_4 \oplus p_5$ is generated, as this codeword increase the has sets $H_2$ and $H_3$. In the updated graph there is now a 3-vertices clique for which MECD transmits $p_2 \oplus p_3 \oplus p_4$, resulting in a total of three transmissions. A partial illustration of the steps used by MECD for this problem instance is shown in Figure 5.

4.3. MECD Computational Complexity

The computational complexity of the Kosaraju’s algorithm is given as $O(|V(D)| + |E(D)|)$ [14]. To calculate $S^k$ requires matrix multiplication. The computational complexity of multiplying two $n \times n$ matrix is $O(n^3)$. Performing such multiplication $k$ times results in a worst case complexity of $O(n^4)$. The worst case computational complexity of searching up to three vertices which form a maximum of 4-vertices clique with $v_j$ is given as $\mathcal{O}(n^3)$, as there exist $\binom{n-1}{3}$ combination of three vertices in $D$ excluding $v_j$.

As a maximum of four clients can be satisfied in each iteration, the do while loop will repeat for a minimum of $\lceil n/4 \rceil$ times. The total worst case computational complexity of constructing adjacency matrix, calculating $S^k$, searching for piggyback packet, and exhaustive search of three vertices will be given as $O(n^3)$, $O(n^5)$, $O(n^3)$, and $O(n^4)$ respectively. The computational complexity of our algorithm is hence dictated by the complexity of calculating $S^k$.

The probability that an edge exist between two arbitrary vertices of a graph is given as $p_{has}^2$. The probability that an edge does not exist between two vertices is given as $1 - p_{has}^2$. As there are $n$ vertices in the graph, the total number of vertices pair is given as $\binom{n}{2}$. The probability that an edge does not exist between any pair of vertices in the graph is given as,

$$X = (1 - p_{has}^2)^{\binom{n}{2}}.$$  \hspace{1cm} (3)

Then the probability that at least one edge exist between any two pair of vertices of graph is given as $1 - X$. A plot of the probability that at least a single edge exist in a graph, verified using simulation results, is plotted in Figure 6. The plots shows that it is sufficient to calculate $S^k$, until $k = 2$ with high probability, and hence the worst case computational complexity of MECD will be
(a) Expected maximum value of $k$ using simulations, 1000 runs.

(b) Probability that at least a single edge exist in the graph.

Fig. 6. Plot of the (a) expected maximum value of $k$, and (b) the probability that at least a single edge exist in the graph, so that it sufficient for the MECD algorithm to calculate $S^k$ until $k = 2$. 
Fig. 7. A 10-nodes TinyOS based TelosB mote receivers testbed to collect packets overhearing data in a single-hop WLAN topology. The transmitting mote is placed 16 meters away from the receivers.

given as $O(n^4)$.

5. Performance Evaluation

We verify the higher coding gain of the MECD for the ICSI problem using simulation under the traditional lossless broadcast channel model assumption used in [4, 9, 10, 11]. The expected number of transmissions using MECD coding algorithm compared with other coding algorithms is plotted in Figures 8(a) and 8(b). The coding gain is given by the ratio $\frac{n}{\ell}$.

For low $p_{has}$ given as $p_{has} \leq 0.5$, MECD reduces the number of transmissions by up to 30% for 100 clients compared to the next best performing coding algorithm. For high $p_{has}$, MECD reduces the number of transmissions by up to 40% for 100 clients compared to the next best performing coding algorithm. It can also be seen in Figure 8(b) that for a fixed $p_{has}$, as the number of clients increases, the gap between the coding gain of MECD and the next performing coding algorithm increases.

5.1. Testbed Implementation

The results of simulations we plotted in Figure 8 were performed under the ideal assumption that all clients experience the same packet erasure probabilities, and hence have equal $p_{has}$ values. However for the topologies of interest, i.e. the WLAN and the “X” topologies, this may not
Fig. 8. The expected number of codeword transmissions for (a) 100 clients with varying $p_{\text{has}}$ probability. And Coding gain for (b) $p_{\text{has}}$ of 0.3, for up to 200 clients. In both the simulations singleton want set and lossless broadcast channel was assumed.
necessarily be the case. In reference [26] it has been shown that two 802.11 wireless devices located in immediate proximity of one another do not show significant packet loss correlation due to localized errors at the receiving devices. And different co-located devices also show different packet loss burstiness patterns. Similar results have also been reported in [27].

Similarly for the “X” topology used to perform network coding in wireless networks with unicast flows, a transmitted packet will only be overheard by its neighbouring nodes. And the probability of overhearing transmissions will be dictated by the distance between the transmitting node and the neighbour.

To test the higher coding gain of our proposed algorithm in practical scenario, we collected traces of packet overhearing in the WLAN and “X” topologies TinyOS based TelosB motes, which has an IEEE 802.15.4 2.4 GHz wireless module. All the experiments were conducted at Namal College lawn as illustrated in Figure 7, which provided a line-of-sight communication and relatively low interference (not measured) in the 2.4 GHz frequency band due to the rural location of the college.

For the WLAN topology, the receiver motes were placed in a cluster, 16 meters away from the transmitter mote. For the “X” topology, the nodes were placed approximately 16 metres away from the relay router, and a bidirectional flow was assumed to be traversing all nodes, i.e. an upstream and a downstream paths traverse all nodes. The distance between all pairs of neighbour had been kept approximately equal. All source-client pairs were selected such that the 1-hop packet erasure probability between the source and client is one.

The average packet overhearing probabilities for 10-nodes testbed for 1,000 sets of packet over-
Table 4 Total instances for which MECD algorithm is optimal for $n \leq 5$.

| $|V(D)| = n$ | Total ICSI instances | Instances for which MECD is optimal |
|------------|----------------------|-----------------------------------|
| 1          | 1                    | 1                                 |
| 2          | 3                    | 3                                 |
| 3          | 16                   | 16                                |
| 4          | 218                  | 218                               |
| 5          | 9608                 | 9500                              |

hearing data traces is given in Table 2 and 3 for WLAN and “X” topologies respectively. Matrix element $a_{ij}$ represents the probability that client $c_i$ overheard packet $p_j$. Variable $a_{ii} = W_{i,0}$ in Table 3 means that $c_i$ wants $p_i$ and the probability of overhearing $p_i$ is zero, and the probability $a_{ij} = 1$ represents that client $c_i$ is transmitting packet $p_j$, and hence will have that packet with probability of one.

Simulation result of generating codewords using various coding algorithms for packet overhearing traces is plotted in Figure 9(a) and 9(b) for the WLAN and “X” topologies respectively. The results show that MECD higher coding gain increases with the network size, and such gains are in particular higher in the WLAN topology and increases with the number of nodes in both the topologies. With the number of connected devices expected to increase by 10 to 100 fold in the 5G mobile network [28], MECD coding algorithm can significantly increase throughput in such dense networks.

5.2. Instances of Optimal Results

We also run our algorithm for all the non-isomorphic graphs $D$ with $n = |V(D)| \leq 5$, each graph representing a unique ICSI problem instance [12]. For each of the instances we run our algorithm and compare the generated codeword length $\ell$ with the optimal length $\ell^*$ found using exhaustive search of $\min\text{rank}_2(A)$. For a given number of vertices, the number of instances for which the MECD algorithm is optimal is summarised in Table 4. To the best of our knowledge, our algorithm constructs optimal index codewords for the highest number of instances for $n \leq 5$.

5.3. Computational Complexity Comparison

The computational complexity of the LDG, extended-LDG, and color-saving algorithm is given as $O(n^3)$, and the worst case complexity of UCIC-LDG, UCIC-color saving is given as $O(n^4)$. Therefore the worst-case computational complexity of our proposed algorithm, $O(n^4)$, is of similar scale as previously proposed algorithms.

6. Conclusion

In this paper we studied the problem of constructing efficient codewords for unicast flows in an opportunistic wireless networks. We proposed the MECD coding scheme, which can construct codewords of smaller length, and hence significantly improve the throughput. We verified the higher coding gain of our proposed algorithm using simulation results under the traditional assumption of lossless broadcast channel and using traces of packet overhearing data collected from TinyOS based TelosB motes testbed. We also showed that our algorithm is optimal for $9738$ of
Fig. 9. The coding gain $\frac{n}{\ell}$, for the (a) WLAN topology, and (b) “X” topology, using packet overhearing traces for testbed of various sizes.
the total 9846 non-isomorphic ICSI instances for \( n \leq 5 \). MECD has worst case computational complexity of \( O(n^4) \), which is of similar scale as previously proposed coding algorithms for the ICSI problem.

7. References


