Abstract

This paper investigates network coding for multicasting on Multistage Interconnection Networks (MINs). To derive the benefits of network coding, a switching node in the interconnection network is modified to execute coding operations before passing on the messages. Based on Vandermonde matrix over finite fields, we design a decentralized network coding algorithm for single-source multicast on MINs. The algorithm is easily applicable to certain existing MINs such as Beneš networks. The algorithm allows the intermediate nodes to decode incoming messages and encode them again in a distributed fashion, which allows amortization of the global coding burden. We also discuss implications of network coding on the connectivity properties of MINs, such as nonblockingness and rearrangeability. Our results show that, empowered with coding functions, MINs can easily attain the rearrangeability property.

1 Introduction

The past few years have witnessed a strong surge in the research of network coding, thanks to the pioneering work (see e.g. [1, 2, 3, 4, 5, 6, 7, 10, 9]) which surprisingly demonstrated that the throughput of a network could be significantly increased by mixing information flows at the intermediate nodes and recovering the original message at the destination nodes.

The success of network coding (or NC for short) prompted us to consider the feasibility of its application on the Multistage Interconnection Networks (MINs). Since nonblockingness and rearrangeability have always been important issues for these networks, it is also natural to evaluate the benefits of network coding on NC-enabled MINs.

In a nonblocking network, the switching nodes are interconnected in such a way that any unused input-output pair can be connected by a path through unused nodes, independent of the other currently existing paths. The requirement of a rearrangeable network is weaker. A rearrangeable network is capable of realizing any one-to-one connection of inputs to outputs with node-disjoint paths, provided that all the connections to be made are known in advance. In 1950s, Shannon proved that any rearrangeable or nonblocking network with N-inputs and N-outputs must have $\Omega(N \log N)$ edges[11]. Bounded-depth nonblocking networks have subsequently been studied extensively [15, 14]. Bassalygo and Pinsker first proved the existence of a bounded-degree nonblocking network with size $O(N \log N)$ and depth $O(\log N)$[12]. However, in the traditional nonblocking networks, nodes can only forward messages. They cannot mix incoming messages.
Our approach is to enable coding functions on nodes and design network codes on the network topology. Furthermore, we investigate how network coding can improve information transmission on the existing interconnection networks with regard to the features such as non-blockingness and capacity gains.

The paper is organized as follows. Section 2 formalizes the terminologies. Section 3 presents a network code based on Vandermonde matrix. Section 4 investigates the multicast problem on some specific interconnection networks with the new coding scheme. Section 5 briefly concludes with directions for future research.

2 Definitions

Let $V$ denote a set of vertices and $E$ a set of directed edges $e$, where $e = <v, v'> \in V \times V$ denotes the edge from $v$ to $v'$. The tuple $<V, E>$ defines a directed graph.

For a node $v \in V$, let $\Gamma^i(v)$ denote the set of the edges $<v, v'>$ entering $v$ and $\Gamma^o(v)$ denote the set of edges $<v, v'>$ outgoing from $v$. For a graph edge $e = <v, v'>$, $v'$ is said to be the head, denoted by $v' = \text{head}(e)$, and $v$ is said to be the tail, denoted by $v = \text{tail}(e)$.

An ordered set $\{u_1, u_2, ..., u_n\}$ is said to be a path $P(u_1, u_n)$ from $u_1$ to $u_n$ in a directed graph $G$ if $u_j, u_{j+1} \in E$, for all $j \in \{1, 2, ..., n-1\}$. Two paths $P$ and $P'$ are said to be edge-disjoint (resp. node-disjoint) if they do not share edges (resp. nodes) in common. A path $P(u, v)$ is said to be a cycle if $u = v$. A graph $G$ is said to be acyclic if it contains no cycles. By default the graphs mentioned henceforth are all meant to be acyclic directed graphs.

Given two nodes $s, t \in V$, an $s$-$t$ cut $(U, \bar{U})$ refers to a partition of the nodes $V = U + \bar{U}$ with $s \in U, t \in \bar{U}$. Denote the set of edges going from $U$ to $\bar{U}$ by $\text{cut}(U, \bar{U}) \equiv \{e \in E \mid \text{tail}(e) \in U, \text{head}(e) \in \bar{U}\}$. Therefore the capacity of the cut refers to the sum of the capacity of edges going from $U$ to $\bar{U}$. Denote the minimum capacity of an $s$-$t$ cut in $(V, E, c)$ by $\min\text{cut}(s, t) \triangleq \min_{U \subseteq \bar{U}, U \in \text{cut}(U, \bar{U})} \sum_{e \in U} c_e$, where $c \triangleq \{c_e\}_E$ is a capacity vector on edges.

The channel graph $CG(s, t)$ is the set of all paths from $s$ to $t$. $CG(s, t)$ is said to be edge-disjoint if any two paths in it are edge-disjoint.

3 Vandermonde Network Codes

Network coding is always connected with the simultaneous transmission of multiple messages. Here we consider the multicast problem in a directed acyclic graph $G = (V, E, c)$, where each edge has unit capacity. Let $s \in V$ denote a source node, and $T \subseteq V$ denote a set of destination nodes. The objective is to simultaneously transmit a vector of messages $M = [m_1, ..., m_r]^T$ from the source $s$ to the destinations in $T$. Let $M$ be a set of $r$ independent row vectors on source node $s$, where $m_i \in \mathbb{Z}_q^r$.

The messages are transmitted along the edges of $G$ and may be duplicated or encoded by the intermediate nodes. The edges leaving a node may transmit any function of the messages available at that node or on the edges inbound to that node. A network code is defined as a set of encoding functions for each node. A network code is called a solution if the original messages can be decoded on the destination nodes. A network code is said to be linear if the encoding function for each edge is a linear combination of its input.

Li et al. [5] have already shown that a multicast problem has a solution if and only if for every destination node the minimum cut separating the source from the sink has a size at least equal to the number of source messages. Moreover, they showed that any solvable multicast problem has a linear solution. The minimum cut size between the source and any destination node is said to be the multicast capacity of the given network.

Let $L(e)$ denote the message transmitted on edge $e$. A linear network code is specified by the linear functions on the source node and all intermediate nodes:

**Source $s$:**

$L(e) = f_s(e) \cdot M$, for $e \in \Gamma^o(s)$, where $f_s(e) = [f_{s,e,1}, f_{s,e,2}, ..., f_{s,e,r}]$ is an $r$-dimensional row vector.

**Internal node $v \in V$:**

$L(e) = \sum_{e' \in \Gamma^i(v)} f_{e',e} \cdot L(e')$, for $e \in \Gamma^o(v)$.

Given a linear network code, each edge $e \in E$ will contain an encoded vector which is the linear combination of some other edges.

Given an edge set $A$, let $L(A)$ denote the linear space
generated by message set \{L(e) : e \in A\}, and let \(\text{rank}(A)\) denote the maximum number of independent message vectors in \(L(A)\). Notice that the notion of \(\text{rank}\) is the same as defined in linear algebra. Given two edge sets, \(A\) and \(B\), we have the following rank properties:

1. \(A \subseteq B \Rightarrow \text{rank}(A) \leq \text{rank}(B)\).

2. \(\text{rank}(A) \leq \min(r, |A|)\).

3. \(\text{rank}(A \cup B) \leq \text{rank}(A) + \text{rank}(B)\), the equality holds iff \(L(A) \cap L(B) = \emptyset\).

4. \(\forall v \in V, \forall M, \text{rank}(\Gamma^\text{in}(v)) = r \Rightarrow v\) can decode all the source messages.

Property 4 can be formally stated as follows:

**Lemma 0** Let \(M\) be a test message set which contains \(r\) independent messages, a linear network code solves single-source \(r\)-message multicast problem iff \(\forall v \in T, \text{rank}(\Gamma^\text{in}(v)) = r\).

**Proof**

Consider one destination node \(v\). Since \(\text{rank}(\Gamma^\text{in}(v)) = r\), we can choose \(r\) independent incoming messages from \(\Gamma^\text{in}(v)\), denoted as \(F_v\), which is also a \(r \times r\) matrix. Therefore the decoding function on node \(v\) is set to be matrix \(M \times F_v^{-1}\).

For any message set \(M'\), let \(M' = M \times M_2\). Since the network code is linear, when \(M'\) reaches node \(v\) from incoming edges, it will turn to \(F_v \times M_2\). Applying the decoding function, node \(v\) can get \(M \times F_v^{-1} \times F_v \times M_2 = M \times M_2 = M'\). Therefore node \(v\) can decode any messages \(M'\). This completes the proof. \(\square\)

The nodes owning property 4 are called **decodable points** in the paper.

To distinguish each edge \(e \in E\), we assign each edge a unique identification number \(\text{UID}(e) \in \mathbb{Z}_q\), where \(\text{UID}(e) \notin \{0, 1\}\). \(\text{UID}(e)\) is denoted as \(e\) where there is no confusion. We now discuss the decentralized encoding algorithm.

**Encoding Algorithm (VA):**

\textbf{begin}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Example of Algorithm VA, \(r = 2, M = \{a, b\}\)}
\end{figure}

For each node \(u \in V\), check the incoming edges.

Let \(p = \text{rank}(\Gamma^\text{in}(u))\).

If \(p = r\)

Reconstruct \(\{m_1, m_2, ..., m_r\}\) by \(\Gamma^\text{in}(u)\).

Let \(A\) denote the transfer matrix, i.e.,

\[|m_1, m_2, ..., m_r| = A \times \Gamma^\text{in}(u),\] where \(A\) is non-singular.

For each outgoing edge \(e = (u, v) \in \Gamma^\text{out}(u)\), set

\[\text{L}(e) = [1, e, e^2, ..., e^{r-1}] \cdot [m_1, m_2, ..., m_r]^T.\]

Else

Assume the \(p\) vectors are \(\{x_1, x_2, ..., x_p\}\).

For each outgoing edge \(e = (u, v) \in \Gamma^\text{out}(u)\), set

\[\text{L}(e) = [1, e, e^2, ..., e^{r-1}] \cdot [x_1, ..., x_p, 0, ..., 0]^T.\]

\textbf{end.}

In Figure 1, source node is \(s\) and there are two destination nodes, \(t_1\) and \(t_2\). Node \(s\) wants to send message \(\{a, b\}\) to both destination nodes. According to VA algorithm, node \(s\) sends encoded messages \(a + e_1 \ast b\) and \(a + e_2 \ast b\) on edges \(\{e_1, e_2\}\) respectively. Node \(C\) receives the two messages and re-encodes them to one outgoing message \(a + e_7 \ast b\) on out-edge \(e_7\). Finally node \(t_1\) receives two independent messages \(\{a + e_1 \ast b, a + e_7 \ast b\}\) and can decode them. So does node \(t_2\).

Recall that a Vandermonde matrix is of the following form:

\[V_m(a_1, a_2, ..., a_r) = \begin{pmatrix}
1 & a_1 & \cdots & a_1^{r-1} \\
1 & a_2 & \cdots & a_2^{r-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & a_r & \cdots & a_r^{r-1}
\end{pmatrix}\]

For distinct \(a_i\), the determinant of Vandermonde matrix is computed as follows.
Lemma 1 [16] \( \det V_m(\alpha_1, \alpha_2, \ldots, \alpha_r) = (-1)^{\binom{r}{2}} \prod_{1 \leq i < j \leq r} (\alpha_i - \alpha_j). \)

**Theorem 2** If the multicast capacity for the acyclic directed graph \( G \) is no less than \( r \), and \( \forall u \in V \setminus s, \text{rank}(T^u(t)) = 1 \) or \( r \), then the network code provided by Algorithm VA can solve the multicast problem, achieving capacity \( r \).

**[Proof]**

Starting from the source node \( s \) by induction on the number of steps, it is easily shown that for each \( e \in E \), it holds true that \( L(e) \) is in the form of \( [1, e, e^2, \ldots, e^{r-1}] \cdot [m_1, m_2, \ldots, m_r]^T \), where \( e \) is the unique ID of an edge in \( E \).

By Lemma 0, given an independent message set \( \{m_1, m_2, \ldots, m_r\} \), we only need to prove \( \forall t \in T, \text{rank}(T^s(t)) \) = \( r \). Since \( G \) has a multicast capacity no less than \( r \), by max-flow-min-cut theorem, \( r \) edge-disjoint paths \( P^i(s, t) \) can be found, \( 1 \leq i \leq r \). Denote \( P_t := \{P^i(s, t) : 1 \leq i \leq r \} \).

Choose \( r \) edges \( \{e_1, e_2, \ldots, e_r\} \) from \( P_t \), each one from \( P^i(s, t), 1 \leq i \leq r \).

\[
\begin{pmatrix}
L(e_1) \\
\vdots \\
L(e_r)
\end{pmatrix}
=
\begin{pmatrix}
e_1 \cdots e_1^{r-1} \\
\vdots \\
e_r \cdots e_r^{r-1}
\end{pmatrix}
\begin{pmatrix}
m_1 \\
\vdots \\
m_r
\end{pmatrix}
\]

Notice that \( e_i \) is exactly on the path \( P^i(s, t) \), which implies that all \( e_i \) are different. Therefore, the transfer matrix shown as a Vandermonde matrix is invertible, by using Lemma 1. This completes the proof. \( \square \)

Theorem 2 indicates that if the ranks of network edges are either 1 or \( r \), the proposed encoding algorithm can solve the multicast problem on the networks. The described algorithm can be done in a decentralized manner for specific networks in that each node need not know the overall topology of the network. Each node need only know the connections with its neighbors, and the encoding/decoding operations are done distributively. In practice, however, the requirement on the rank of edges might be difficult to achieve since the concept of rank is not directly related to the degree of nodes. In the next section we will discuss the structure of multistage networks and show that certain networks can actually satisfy the requirement of VA algorithm.

## 4 Multistage Network with Coding Functions

A multistage network \( G \) is a network in which the nodes are partitioned into \( L + 1 \) stages such that:

(a). The inputs are the nodes on stage 0.
(b). The outputs are the nodes on stage \( L \), and
(c). Every edge links a node on stage \( i \) to a node on stage \( i + 1 \) for some \( 0 \leq i < L \).

The multistage network has been well studied in the context of parallel computers. Well-known examples include Clos 3-stage network [15], Butterfly and Beneš networks etc. [14]. However, network coding suggests a new view on the function of switches in the multistage networks. Consider a crossbar (switch block) with \( N \) inputs and \( M \) outputs. A crossbar in the conventional multistage networks context is only capable of either forming point-to-point connections or simple fan-out capability. For our interest, the modified crossbar will be able to decode the input messages, re-encode before passing to the output. Intuitively, because network coding enables the sharing of connection, the whole network should offer more multicast traffic than without network coding.

We start with the Butterfly and Beneš networks. We found that Beneš networks actually satisfy the requirement of our VA algorithm. An \( N \)-input Butterfly has \( \log N + 1 \) stages, each with \( N \)-nodes. The Beneš network is a \((2 \log N + 1)\)-staged network consisting of two copies of Butterflies back-to-back.

**Corollary 3** VA algorithm can solve multicast problem on the Beneš networks with \( r = 2 \).

**[Proof]**

The min-cut between any input-output pair is 2 in Beneš networks. Therefore multicast capacity is 2 at most. Since the in-degree of any node in the network is 2, the rank of any edge involved in transmission is 1 or 2. Applying Theorem 2 yields the result. \( \square \)

In the multicast traffic, if the number of receivers is at most \( f \), the multicast is called an \( f \)-cast. A switch is said to have the fan-out capability if the switch itself can connect any inlets to any number of idle outlets. Kirkpatrick-Klawe-Pippenger considered the Clos network \( C(n, r, m) \) [15] with unconstrained fan-out capability, and showed that it is multicast rearrangeable if \( m \geq (n - 1) \floor{\log_2 r} + 2n - 1 \) [13]. However, currently
there is no precise low bound on the feasible $m$. With coding, certain cases of multicast rearrangeability problem become straightforward.

Figure 2: Example of Algorithm VA, $r = 4, M = \{a, b, c, d\}$

As in Section 2, one set of output edges is said to be $r$-recoverable if there are $r$ independent messages being transmitted on it.

**Theorem 4** Given $m \leq N$ independent messages at the inputs of an $N \times N$ crossbar, let $R$ (resp. $Q$) denote the number of different $m$-recoverable set without network coding (resp. with network coding). Then $Q/R \geq \left(\frac{(N-1)m}{N^m}\right)^{m-1} \geq 1$.

**[Proof]**

First consider the case without coding. Let $i_j$ be the number of outputs which receive message $j$, $1 \leq j \leq m$. Then $\sum_{j=1}^{m} i_j = N$.

The number of recoverable output sets

$$R = \prod_{j=1}^{m} i_j \leq \left(\frac{N}{m}\right)^m.$$  

Now consider the case with coding. In terms of the proof of Theorem 2, any $m$ of output edges can fully decode the messages. Therefore the number of recoverable output sets

$$Q \geq \binom{N}{m}.$$  

Therefore, $Q/R \geq \prod_{k=0}^{m-1} \left(\frac{(N-k)m}{N(m-1)}\right) \geq \left(\frac{(N-1)m}{N(m-1)}\right)^{m-1} \cdot \square$

**Theorem 5** Consider Clos network $C(n, r, m)$ with coding. If the set of destination crossbars for any f-cast is the same, and message set is $M$, where $|M| \leq f$, then $m \geq f \Leftrightarrow f$-casts are arrangeable.

**[Proof]** The first $m$ of middle crossbars can be chosen for the f-cast purpose. The $i$-th $(1 \leq i \leq m)$ middle crossbar collects messages from all the source crossbars, and decodes the $i$-th message, finally sends this message to all the receiving crossbars.

On the other hand, clearly the Clos network with $n < f$ of middle crossbars cannot transmit $f$ independent messages. □

**Corollary 6** Consider Clos network $C(n, r, m)$ with coding. If the maximum number of output crossbars for all f-cast is $f_1$, and the message set is $M$, $|M| \leq f_1$, then $m \geq \max(f, f_1) \Rightarrow f$-casts are arrangeable. □

The above results suggest that by using coding on switching nodes, the network cost (in terms of the number of edges) might be decreased. In other words, edges could be shared during multiple source multicastrs.

5 Conclusions

Our research has adapted network coding to the multi-stage interconnection networks.

The VA algorithm in the paper provides a decentralized network coding for multicast problems on the specific networks which require independency check on the incoming edges of the nodes. Although the requirement somewhat restricts networks that can be applied with our coding scheme, the algorithm still shows some promises. The algorithm allows the intermediate nodes to encode/decode distributively, which can amortized the global coding burden. For multistage networks such as Benes network, the algorithm shows its feasibility and low complexity. We also demonstrated certain results concerning multicast rearrangeability.

One useful research direction may be in designing cost-effective network topologies in conjunction with specific coding schemes. For instance, one way to construct a cost-efficient multistage network is to recursively extend the “butterfly” structure given in Fig 1. Fig. 2 show a case where there are 4 destinations and $r = 4$. Without network coding, the maximum number of messages to multicast is 3. Therefore the coding gain is $4/3$. Further study
on this recursive construction shows that the coding gain decreases with the number of destinations. According to Jaggi’s result in [7], the network coding gain on some special networks could be \( \Omega(\log |V|) \). In this sense, more research is needed to fully exploit the power of network coding.

References


