Reversible and Irreversible information networks

Soren Riis
Queen Mary, University of London
May 2005 (revised March 2006)

***Abstract*— It is shown that there exist information networks where messages can be sent (utilising Network Coding) more easily in one direction than in the opposite direction. This is valid even though each channel is assumed to have the same capacity in both directions.**

It is shown that irreversible information networks only have solutions that use non-linear Network Coding. I argue that this result is more surprising than might appear at first sight and that it follows using ideas resembeling the path integral in Quantum Mechanics.

I. Magic in information networks

Network Coding is a new area of multi-user information theory that has expanded dramatically within the last few years. Network Coding is based on a simple mathematical model of network flow and communication first explicitly stated in its simplicity in [2]. Recently, ideas related to Network Coding have been proposed in a number of distinct areas of Computer Science and engineering (e.g. broadcasting in wireless networks [25], [24], [23], data security [4], distributed network storage [6], [1] and wireless sensor networks [16]). Network Coding has also a broad interface with various Mathematical disciplines (error correcting codes [19], [5], [11], circuit complexity [17], information theory [12], algebra [15], [14] and graph theory).

The basic idea underlying Network Coding has been explained in many papers e.g. [15], [2], [17], [7]. The idea can be illustrated by considering the "butterfly" network in figure 1.

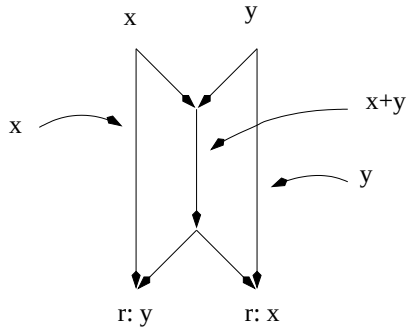

figure 1

The task is to send the message x from the upper left corner to the lower right corner and to send the message y from the upper right corner to the lower left corner. We say the lower left (lower right) node requires x (requires y) and write this requirement as $r : y$ ($r : x$). The messages $x, y \in A$ are selected from some finite alphabet A. Assume that each information channel can carry at most one message at a time. If the messages x and y are sent simultaneously there is a bottleneck in the middle information channel. On the other hand if we, for example, organise A as a commutative group $(A, +)$ and send $x + y \in A$ through the middle channel, the messages x and y can easily be recovered at 'output' nodes at the bottom of the network (since $y = (x + y) - x$ and $x = (x + y) - y$).

It is often convenient to think about each message as a flow of elements from A. Viewed this way we can consider messages a and b as sequences $\ldots a_{-2}, a_{-1}, a_0, a_1, a_2 \ldots$ and $\ldots, b_{-2}, b_{-1}, b_0, b_1, b_2, \ldots$. The solution $a+b$ then consists of the sequence $\ldots, a_{-2}+b_{-2}, a_{-1}+b_{-1}, a_0+b_0, a_1+b_1, a_2+b_2, \ldots$ being sent through the middle channel.

The information network in figure 1 is an example of a *multiple-unicast information network*. In general a multiple-unicast information network $N = (V, E; s_1, t_1; s_2, t_2; \ldots, s_n, t_n)$ is an acyclic graph with source nodes $s_1, s_2, \ldots, s_n$ of in-degree 0 and target nodes $t_1, t_2, \ldots, t_n$ of out-degree 0.

Informally, the idea is (repeatedly) to send messages $m_1, m_2, \ldots, m_n \in A$ from the source nodes to the target nodes. More specifically message m_j has to be sent from source node s_j to target node t_j (node t_j requires m_j). The messages are chosen from an alphabet A that throughout the paper is assumed to be finite, containg at least two letters.

Formally, associate to each source node s_j a variable x_j and associate to the corresponding target node t_j the requirement $r : x_j$. Furthermore associate to each edge $e = (v, w)$ in N with v having in-degree $k(e) \in N$ a $k(e)$-ary function symbol f_e. In the case v is a source node a 1-ary function symbol is associated to (v, w).

For an edge $e = (v, w)$ each of the $k(e)$ incoming edges is associated to one of the $k(e)$ arguments of f_e. Finally, to each target node t_j of in-degree $k(t_j) \in N$ is associated a $k(t_j)$-ary function symbol f_{t_j}. Each of the $k(t_j)$ incoming edges is associated to one of the $k(t_j)$ arguments of f_{t_j}.

A *flow* ρ for the *multiple-unicast network coding problem* N (over the alphabet A) is an assignment that to each function symbol f_e (or f_{t_j}) assigns a map $\bar{f} : A^{k(e)} \to A$ ($\bar{f} : A^{k(t_j)} \to A$). The map assigned to f_e by the assignmet ρ is denoted $\bar{f}^{\rho}_e$; and the map assigned to f_{t_j} by the assignment ρ is denoted $\bar{f}^{\rho}_{t_j}$.

Notice that a flow ρ uniquely determines (inductively) to each edge $e = (v, w)$ (as well as to each target node t_j) a function $\bar{h}^{\rho}_e : A^n \to A$ (or $\bar{h}^{\rho}_{t_j} : A^n \to A$) that expresses the flow through the edge e (target node t_j) as a function of the messages $m_1, m_2, \ldots, m_n \in A$.

The flow through the edge $e = (v, w)$ is (obviously) defined by the equation $\bar{h}^{\rho}_e(x_1, x_2, \ldots, x_n) : \bar{f}^{\rho}_e(\bar{h}^{\rho}_{(u_1,v)}, \bar{h}^{\rho}_{(u_2,v)}, \ldots, \bar{h}^{\rho}_{(u_k,v)})$ where $(u_1, v), (u_2, v), \ldots, (u_k, v)$ are all incoming edges to v. The flow through each edge $(s_j, u) \in$

E is given by $\bar{h}^{\rho}_{(s_j,u)}(x_1, x_2, \ldots, x_n) = x_j$. Finally, the flow arriving at node t_j is given by $\bar{h}^{\rho}_{t_j}(x_1, x_2, \ldots, x_n) := \bar{f}^{\rho}_{e}(\bar{h}^{\rho}_{(u_1,v)}, \bar{h}^{\rho}_{(u_2,v)}, \ldots, \bar{h}^{\rho}_{(u_k,v)})$ where $(u_1, t_j), (u_2, t_j), \ldots, (u_k, t_j)$ are all incoming edges to t_j.

A flow ρ is a *solution* to N (over the alphabet A) if $\bar{h}^{\rho}_{t_j}(x_1, x_2, \ldots, x_n) = x_j$ for each target node t_j. A multiple-unicast information network N is *solvable* over the alphabet A if there exists a flow ρ that is a solution to N (over the alphabet A).

Expressed less formally, a flow is a solution if messages $m_1, m_2, \ldots, m_n$ are sent from the source nodes to their corresponding target nodes. A flow specifies how the messages are transmitted, mixed, and transformed through the network. A flow is a solution if for each choice of "input", the messages are sent correctly to their destinations.

In general instantenous information networks might not be multiple unicast and have different type of requirement where one message, for example, might have more than one destination. It is straightforward to modify the above definitions to include this case. However we do not need this generalisation, but interested readers might, for example, consult [8] for a more general definition. For a multiple-unicast information network N the "dual" information network N^d is obtained by reversing all edge directions of edges in N and reversing the role of source and target nodes. Notice that the dual information network N^d (of a multiple-unicast information network) is a multiple-unicast information network.

In figure 2 (a) a multiple-unicast information network N is given. The information flow problem N is solvable (over any alphabet), and a natural solution (that works for any alphabet A that is organised as an abelian group) is indicated. In figure 2 (b) the dual information flow problem N^d is considered. This problem also has a solution as indicated.

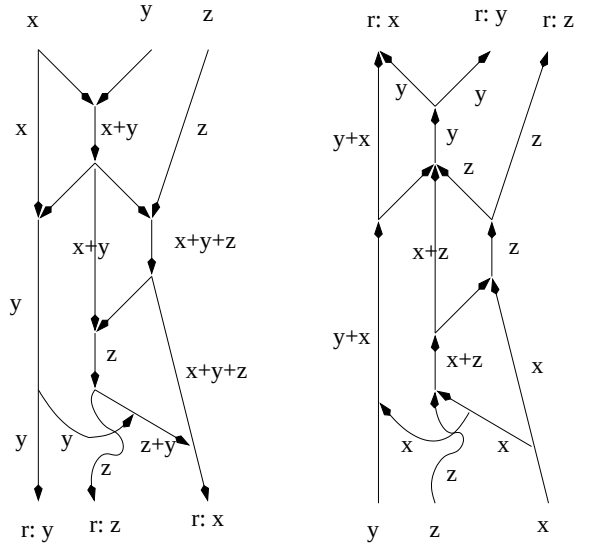

fig 2

Notice that the 'forward' solution in the information network N in figure 2 (a) is quite different from the 'backward' solution in figure 2 (b). Actually any forward solution differs from any backward solution. It can, for example, be shown that **each** forward solution has some channel (edge) that carries an information flow that depends on all 'input' messages (i.e. like $x + y + z$ depends on the messages x, y and z). On the other hand in **none** of the 'backward' solutions is there a channel (edge) that has an information flow that depends on more than two input messages.

In this paper I state and prove two curious theorems concerning multiple-unicast information networks. According to one of the theorems there exists a multiple-unicast information network N which is solvable (over an alphabet of size 2), but where the dual information network N^d not is solvable (over an alphabet of size 2). Expressed in ordinary everyday terms:

There exists an information network configuration such that a set of k users (38 users in my construction) in general can send messages without congestion (or delay) to their k friends (38 friends). On the other hand the friends cannot reply back to the recipients without creating congestion (or delay).

The other theorem states that for linear maps such a situation cannot appear. Maybe this might appear to be what we would expect, but I will try to convince the reader that this result is more surprising that it might appear at first!

To see this it has to be appreciated that for many information networks there seems to be (at least from the superficial level) hardly any relationship between their linear solutions, and their linear solutions to the dual (reverse) problems. We already saw an example of this in figure 2 where each forward solution looks very different from each backward solution.

For some classes of information networks the messages have to flow through completely different regions of the network depending on whether "forward" or "backward" solutions are considered. To illustrate this consider, for example, the network in figure 3.

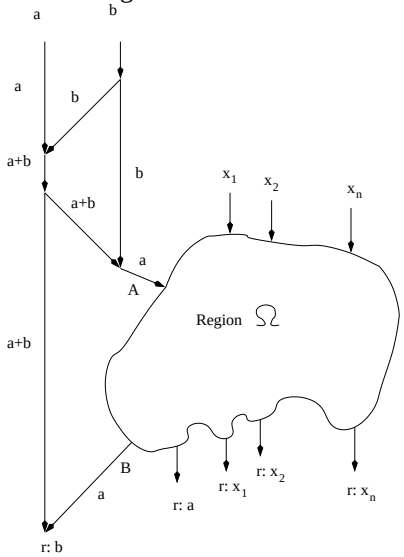

figure 3

As usual assume that each channel can carry at most one message m from some finite set A of potential messages. The task of the network in figure 3, is to send messages a,b as well as messages $x_1, x_2, \ldots, x_n$ from the upper source node to the target nodes at the bottom of the network. Each target node requires one of the messages a,b and $x_1, x_2, \ldots, x_n$ (a node labelled $r : y$ is required to reconstruct message y).

Consider also the reverse problem (the dual problem) where the direction on all edges has been reversed. The information network in figure 3 might (or might not) have a solution over a given fixed alphabet. This depends, of course, on the exact layout of the region Ω. What is certain is that channel A is forced to transmit message a (or a permutation $\pi(a)$) into the region Ω. Thus in any solution, the message b cannot have any influence on any flow inside region Ω.

In other words:

> *Message b does not enter or have any influence on events in region Ω in any downward solution.*

This is in sharp contrast to the situation in the dual problem, where messages have to be sent from the bottom to the top of the network. In this case message b must enter the region Ω through channel B. And message b must be sent through the area Ω such that edge A is able to send a function that depends on both a and b (e.g. $a + b$) In other words:

> *Message b enters region Ω and affects events inside the region as well as the message through edge A in each upward solution.*

This and other examples make it easier to appreciate the fact that linear solutions can always be reversed.

The example can be pushed even further and it is not hard to generalise the construction, and construct information networks where a large set $y_1, y_2, \ldots, y_r$ of messages never enters a region Ω in solutions in one direction, but where all messages $y_1, y_2, \ldots y_r$ have to enter the region Ω for all solutions of the dual flow problem. This observation might suggest that we can get an irreversible information network by simply making sure that a region Ω is getting 'over heated' with information flows in one direction (by forcing many flows into the region), while the flows into the region Ω are keept at a reasonable level in the dual problem. But as it turns out some other region (outside Ω) is then bound to get 'over heated'. So even though the reverse (dual) problem might be solvable in region Ω somehow miraculous (whatever we do) congestion will always appear somewhere else!!

This fact follows from quite a general theorem (Theorem 1) that states that under quite general conditions N and the dual problem N^d in fact have the same number of solutions.

To state Theorem 1 assume that A has a commutative group structure (i.e. $G = (A, +)$). Let R be a space of functions mapping A to A. We say that R *acts* on A. Throughout the paper I assume R is closed under compostion and contains the identity map 1 defined by $1(x) = x$ for all $x \in A$.

A flow is *R-linear* if each function $\bar{f}_e(z_1, z_2, \ldots, z_k)$ is of the form $\bar{f}_e(z_1, z_2, \ldots, z_k) = \lambda_1(z_1) + \lambda_2(z_2) + \ldots + \lambda_k(z_k)$ where $\lambda_1, \lambda_2, \ldots, \lambda_k \in R$.

The space R is a *-algebra if there exists a map $* : R \to R$ which satisfies the identities:

(i) $(a + b)^* = a^* + b^*$

(ii) $(ab)^* = b^* a^*$

(iii) $(a^*)^* = a$

(iv) $1^* = 1$

The definition of R-linarity (when R is a *-algebra) is very general. The definition include many natural examples from linear algebra and ring theory.

Theorem 1

> Let N be a multiple-unicast information network. Let N^d be the dual multiple-unicast information network where all edge directions are reversed, and the role of targets and sources reversed. Let A denote the underlying alphabet, and assume that it is organized as a commutative group $G = (A, +)$.
> Let R be a *-algebra acting on A. Then the number of distinct R-linear solutions to N (over A) is identical to the number of distinct R-linear solutions to N^d (over A).

The theorem has a number of consequences. To state these I introduce the following notions.

Assume that the alphabet is organised as a field $F = (A, +, \times)$ (in which case $|A|$ has p^k elements for some prime number p and $k \in N$). In this case we say a flow is *scalar linear* (or just *linear*) if each function $\bar{f}_e(z_1, z_2, \ldots, z_k)$ is linear i.e. of the form $\bar{f}_e(z_1, z_2, \ldots, z_k) = \lambda_1 z_1 + \lambda_2 z_2 + \ldots + \lambda_k z_k$ with $\lambda_1, \lambda_2, \ldots, \lambda_k \in F$. Notice that F naturally acts on A (since each $r \in F$ defines a map $r : A \to A$ defined by $r(a) = ra$). The set $R_F := F$) of these actions is closed under composition, and contains the identity map (since $1 \in F$). And the identity operation $* : R_F \to R_F$ makes R_F a *-algebra (since R_F is commutative). Thus Theorem 1 applies:

Corollary 2 [13]

> Assume A is organised as a field $F = (A, +, \times)$. Then N has a scalar linear solution over A if and only if N^d has a scalar linear solution over A.

Unknown to me this result was first (as pointed out by R. Koetter) proved in [13] (Theorem 5) where the authors based their proof on Forney's Duality Theorem [10].

Another important case appears when A is a finite dimensional vector space $V = (A, +, F)$ over a (finite) field F. In this case a flow is *matrix linear* (or just *linear*) if each function $\bar{f}_e(z_1, z_2, \ldots, z_k)$ is linear i.e. of the form $\bar{f}_e(z_1, z_2, \ldots, z_k) = \lambda_1 z_1 + \lambda_2 z_2 + \ldots + \lambda_k z_k$ where $\lambda_1, \lambda_2, \ldots, \lambda_k$ are linear maps mapping V to V (or equivalent $d \times d$ matrices with entries from F, where $d = dim(V)$). Notice that each linear map in $R_V = L_F(V, V)$ maps V to V and thus naturally acts on A. The set R_V of linear maps from V to V is closed under composition and contains the identity map (since $1 \in L_F(V, V)$). Matrix transposition defines a *-operation that satisfies axioms (i)-(iv). Thus, R_V can be organised as a *-algebra and Theorem 1 applies:

Theorem 3

> Assume A is organised as a vector space $V = (A, +, F)$. Then the number of matrix linear solutions to N over A is identical to the number of matrix linear solutions to N^d over A.

This shows that not only does N has a (matrix) linear solution if and only if N^d has a (matrix) linear solution, but in fact the two problems have exactly the same numbers of distinct solutions.

In general the alphabet A might also be organised as a finite commutative ring R (with unity). Since R naturally acts on A, is closed under multiplication (i.e. the actions are closed under composition), since R contains the identity map (since $1 \in R$) and since R is commutative (is a *-algebra) Theorem 1 applies:

Corollary 4

> Assume A is organised as a commutative ring $R = (A, +, \times)$ with one element. Then the number of linear solutions to N over A is identical to the number of linear solutions to N^d over A.

In this paper I present two proofs of Theorem 3. The first proof uses ideas I introduced in [18] first to prove two Theorems (Theorem 7 and Theorem 9). These theorems are interesting in their own right. When combined with Lemma 8 the Theorems entail Theorem 3.

The second proof (that was added in the revised version of the paper) in fact proves Theorem 1 that is stronger than Theorem 3. The proof uses an idea that resembles *Feynman's path integral* in quantum physics!

So according to Corollary 2 and 4 as well as Theorem 3, any information network is reversible with respect to linear solutions. However in general (where solutions might be non-linear) there exist irreversible information networks.

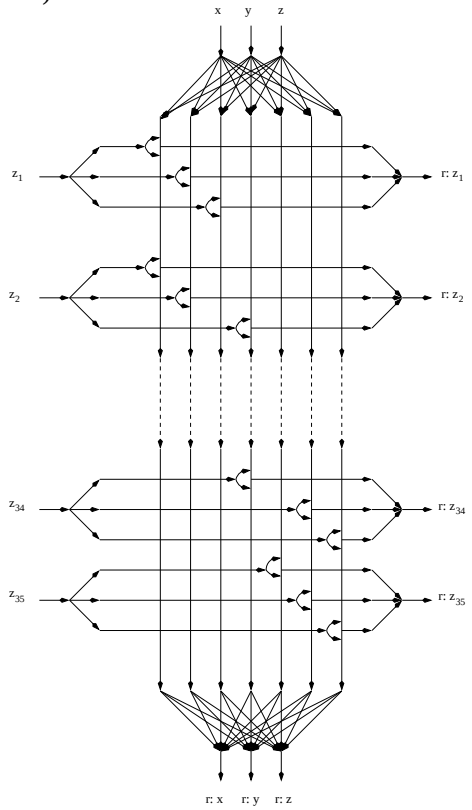

figure 4

Consider the information network N in figure 4. It contains 38 source nodes and 38 target nodes. It has 7 vertical channels which I will refer to as channel $1, 2, 3, 4, 5, 6$ and 7. For each 3-element subset $W \subseteq \{1, 2, 3, 4, 5, 6, 7\}$ I introduce a 3-way crossing as illustrated in figure 4. For each of the 35 of these 3-way crossings I introduce a variable z_j denoting the message (or flow of messages) I want to send along the 3-way crossing.

To define N unambiguously, I fix an ordering of all crossings. Any fixed ordering will work, but I choose the lexographic ordering of the 3-element subsets of $\{1, 2, 3, 4, 5, 6, 7\}$. The variables are enumerated starting with the variables associated the smallest sets first. The reader can check that in this assignment the variable z_1 corresponds to the set $\{1, 2, 3\}$, the variable z_{16} corresponds to the set $\{2, 3, 4\}$, the variable z_{26} to the set $\{3, 4, 5\}$, the variable z_{32} to the set $\{4, 5, 6\}$ and finally variable z_{35} corresponds to the set $\{5, 6, 7\}$.

I will show:

Theorem 5

> The multiple-unicast information network N in figure 4 has a solution over alphabets of size 2.
>
> The "dual" multiple-unicast information network N^d is unsolvable over alphabets of size 2.

The solution for the problem N allows the 38 source messages to be sent through the network along the arrows to their destinations (target nodes) so each of the 38 messages always arrives correctly at their destination.

In the dual problem N^d only 37 out of the total of 38 messages can (expressed in non-technical terms) be guaranteed to arrive 'unscrambled' and without delay at their destinations.

II. Graphs and their Guessing Numbers

The first proof of Theorem 3 follows from two theorems (Theorem 7 and Theorem 9) that are interesting in their own right. The second of these Theorems (Theorem 9) was first introduced in [18], but for completeness I include and prove this Theorem in this paper. The guessing number of a directed graph G is defined by introducing a simple cooperative game that is played on G.

Let $G = (V, E)$ be a directed graph on vertex set $V = \{1, 2, \ldots, n\}$ and let $s \in \{2, 3, \ldots\}$. I define a cooperative game (GuessingGame(G, s)) that is played on the graph G. The vertex set $V = \{1, 2, \ldots, n\}$ represents n players. Each player is assigned randomly (and independently) a die value from a fixed set $A = \{1, 2, \ldots, s\}$ of s elements. No player knows the value of their own die. Each player $v \in \{1, 2, \ldots, n\}$ sends the value of their die $\in \{1, 2, \ldots, s\}$ to each player $w \in \{1, 2, \ldots, n\}$ with $(v, w) \in E$. In other words, each node w receives dice' values from a set $A_w := \{v \in V : (v, w) \in E\}$.

Each player has to guess the value of their own die. We want to calculate (assuming the players have agreed in advance on a guessing strategy ρ) the probability $p = p(G, s, \rho)$ that all the players (nodes) simultaneously guess their own dice values. Formally, a guessing strategy ρ is a collection of functions - one function f_j for each vertex $j \in V = \{1, 2, \ldots, n\}$ and with one argument for each incoming edge. If the nodes $1, 2, \ldots, n$ in G are assigned values $a_1, a_2, \ldots, a_n \in A$ node j guess that their own die value is $f_j(a_{i_1}, a_{i_2}, \ldots, a_{i_d})$. An optimal guessing strategy ρ is a strategy for which the value of $p(G, s, \rho)$ is maximal. Since there is only finitely many guessing strategies there always exist one or more optimal guessing strategies. The maximal value of $p(G, s, \rho)$ - the value being achieved when ρ is optimal - is denoted by $p(G, s)$.

At first glance it might seem that the players can never do better than pure (uncoordinated) random guessing (where players randomly and independenly each making a random guess in $\{1, 2, \ldots, s\}$). In other words, it might seem that $p(G, s, \rho)$ is always independent of G (and ρ) and given by $(\frac{1}{s})^n$. However, a little reflection shows that $p(G, s, \rho)$ in general can be much higher than $(\frac{1}{s})^n$ for many graphs. Let, for example, G be the complete graph K_n. In this graph each player has access to all dice values except the value of their own die. If the players, for example, assume (as their collective guessing strategy ρ) that the sum of all dice values is 0 modulo s, one player guesses correctly his/her own die value if and only if all players guess correctly the value. In other words $p(K_n, s, \rho) = p(K_n, s) = \frac{1}{s}$ and ρ is an optimal guessing strategy. The *guessing number* $k = k(G, s)$ of a directed graph G is defined as the unique number k that satisfies the equation $(\frac{1}{s})^{|V|-k} = p(G, s)$. The complete graph k_n on n nodes has guessing number $n - 1$. Notice that the players all correctly guess the value of their own dice with a factor s^{n-1} better than pure uncoordinated random guessing.

In general, a directed graph G has guessing number $k(G, s)$ if the players have a strategy so they all correctly guess the value of their own dice with a factor $s^{k(G,s)}$ better than pure uncoordinated random guessing.

The guessing number $k(G, s)$ of a directed graph depends in general on the direction of the edges in G. Theorem 5 (combined with Theorem 7) can be used to show that there exists a graph G such that $k(G, 2) \neq k(G^d, 2)$ where G^d is the dual graph that appears from G by reversing all edge directions. However, I will show that if the players use linear guessing strategies (in a wide sense) then the corresponding guessing numbers are the same for G and G^d.

The most clear case appears if A is organised as a field F (i.e. $F = (A, +, \times)$). However to get more generality assume A is organised as a vector space $V = (A, +, F)$ (d dimensional vectorspace over a finite field F). A *linear guessing strategy* is a strategy where each player assumes that their own die value is given by $\lambda_{u_1,v}(a_{u_1}) + \ldots + \lambda_{u_d,v}(u_{a_d})$ where $\lambda_{u_j,v}$ for each incoming edge (u_j, v) is a linear map $\lambda_{u_j} : A \to A$ and $a_{u_j} \in A$ is the die value assigned to the node u_j. Let $L_F(V, V)$ denote the space of linear maps from V to V.

Notice that each linear guessing strategy is uniquely determined by assigning to each (directed) edge in E an element in $L_F(V, V)$ i.e. a linear map $r : V \to V$ (i.e. a $d \times d$ matrix with entries in F). Conversely, each assignment that assigns a linear map in $L_F(V, V)$ to each edge (i.e. a $d \times d$ matrix with entries in F) corresponds to a linear guessing strategy. This natural one to one correspondance shows that the number of linear guessing strategies is $|L_F(V, V)|^{|E|}$. The dual graph G^d has, of course, $|L_F(V, V)|^{|E|}$ guessing strategies since $|E| = |E^d|$. Actually, there is a natural one-to-one correspondance between the linear guessing strategies of G and G^d via the labelling of E that naturally can be viewed as a labelling for E^d. More specifically, if $(v, w) \in E$ has assigned value $t \in L_F(V, V)$, then $(w, v) \in E^d$ has assigned the adjoint value t^* (given by the transposed matrix of the $d \times d$ matrix associated to t).

As already pointed out, for each fixed strategy ρ, each node makes a guess given $\lambda^\rho_{u_1,v}(a_{u_1}) + \ldots + \lambda^\rho_{u_d,v}(u_{a_d})$. This can be displayed as a row vector $(\lambda^\rho_{1,v}, \lambda^\rho_{2,v}, \ldots, \lambda^\rho_{n,v})$ that is multiplied by the coloumn vector $(a_1, a_2, \ldots, a_n)$ in order to get the value guessed by node v. The matrix $M^\rho := (\lambda^\rho_{ij})_{1 \leq i,j \leq n}$, with entries being $d \times d$ matrices over F uniquely determine (when it is multiplied by the vector $\vec{a} := (a_1, a_2, \ldots, a_n)$ of the actual dice values of the n nodes) the guessed vector $\vec{a}^{guess} := (a_1^{guess}, a_2^{guess}, \ldots, a_n^{guess})$.

The players all guess correctly their own die value exactly when $\vec{a} = \vec{a}^{guess}$ i.e. whenever $M^\rho \vec{a} = \vec{a}$, or equivalently, whenevever $(M^\rho - I)\vec{a} = 0$ where I is the $n \times n$ identity matrix (with $d \times d$ identity matrices as entries). Now this product can be written as $(M^\rho - I)\vec{a} = 0$ where we view M^ρ and I as $nd \times nd$ matrices over F and view $\vec{a}$ as a nd-dimensional vector over F.

From this we get:

Lemma 6

> Let G be any directed graph, and let A be a d-dimensional vectorspace over a finite field F. For each linear guessing strategy ρ the set of dice values $\vec{a}$ where each player guesses correctly their own die value is a linear subspace (of the nd-dimensional vectorspace over F). Its vectorspace dimension (over F) is given by $nd - \mathrm{rank} M^\rho$.
>
> The probability that all players are correct is given by $p(G, s, \rho) = \frac{|F|^{nd - \mathrm{rank} M^\rho}}{|F|^{nd}}$.
>
> The linear guessing number $k(G, |A|, \rho)$ is given by $k(G, |F|, \rho) = \frac{nd - \mathrm{rank} M^\rho}{d}$.

Given a linear guessing strategy ρ for G, the corresponding linear guessing strategy for G^d is (with slight abuse of notation) denoted ρ^*. From this we get:

Theorem 7

> Let G be any directed graph, and let A be a d-dimensional vectorspace over a finite field F. Let G^d be the dual graph of G where all edge directions have been reversed. For each guessing strategy ρ for G let ρ^* denote the corresponding guessing strategy for G^d.
>
> The probability that all players are correct in G using the linear strategy ρ is identical to the probability all players are correct in G^d using the linear strategy ρ^* (i.e. $p(G, |A|, \rho) = p(G^d, |A|, \rho^*)$). Thus especially $p(G, |A|) = p(G^d, |A|)$. Furthermore $k(G, |A|, \rho) = k(G^d, |A|, \rho^*)$ and $k(G, |A|) = k(G^d, |A|)$.

Proof: Depends essentially on the algebraic characterisation of the linear guessing numbers in Lemma 6, and the fact that the rank of a matrix is preserved under matrix transposition. ♣

III. Information network problems as cirtuit information problems

A Circuit is an acyclic graph with input nodes $i_1, i_2, \ldots, i_n$ and output nodes $o_1, o_2, \ldots, o_m$. Each input node has indegree 0, and each output node has outdegree 0. Usually, (in circuit complexity) each input is 0 or 1, and each node (except the input nodes) computes a Boolean function if its incomming edges, The function value (0 or 1) is then passed on, along each outgoing edge, to the successor nodes. In the setting of

Boolean circuits, nodes are unually referred to as *(boolean) gates*. In general, there is no reason only to restrict the computatinal model to the case where $A = \{0, 1\}$ and in general A can be any set with at least two elements.

The computational model used in network Coding - the instantanaous information network - is very similar to the circuit model. However, there is an difference. In the circuit each gate computes one specific function valuebthat is then passed on to all successor nodes. In the instantaneous information network more than one function (in fact one for each outgoing edge) is computed at each node.

The circuit has the very convinient feature, that each linear map $f : A^{d_v} \to A$ (into a node v), can be specified uniquely by labelling each edge e by a linear map $g_e : A \to A$. And conversely, each labelling of each edge e by a linear map $g_e : A \to A$ uniquely determine linear maps $f : A^{d_v} \to A$ for each node v.

The next figure shows how any instantainous information N network by "blowing up each node" can be viewed as a circuit N_B.

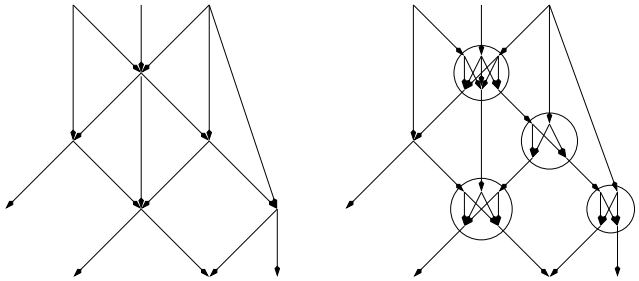

figure 5

In general each node with indegree d_1 and outdegree d_2 is replaced by the graph K_{d_1,d_2} containing $d_1 \times d_2$ edges. Nodes with in-degree 1 (or out-degree 1) does not need "blowing up". This includes all source nodes and all target nodes.

This translation is very convinient (for considering linear flows) for instantanious information networks since each assignment of a linear function $f : A^{d_1} \to A^{d_2}$ in N to a node v uniquely corresponds to an assignment of linear maps $g_e : A \to A$ to the edges (in K_{d_1,d_2}) in N_B that were constructed when blowing up the node v.

Notice if a multiple-unicast information network N has corresponding circuit N_B (by blowing up its nodes), the dual multiple-unicast information network N^d has corresponding circuit $(N^d)_B$ identical to $(N_B)^d$.

Some of the edges in N_B are not needed and actually would make the next lemma invalid! The reason for this is trivial and boil down to the fact that two edges (u, v) and (v, w) when compared to the single edge (u, w) has more assigments (e.g. an assignment of rs to (u, v) and r^{-1} to (v, w) is equivalent to an asignment of s to (u, w)). To resolve this "problem" the above conversion is modified (and simplified) as follows:

Given an multiple-unicast network N. For each source node of N add incomming edge of in-degree 1. For each target node of N add an outgoing edge of degree 1. This new multiple-unicast network is denoted N' (the added edges are viewed as source edges and target edges). The Circuit information problem N_C is defined as follows:

Each edge in N' is a node in N_C. Any two nodes (v_1, v_2) and (v_3, v_4) (that are edges in N') belongs to an edge (in N_C if and only if $(v_2 = v_3)$.

Notice, that N_C appears from N by essentially first "blowing up points" (as defined above) and then removing "unnessary" edges. Notice that $(N^d)_C$ is identical to $(N_C)^d$.

Lemma 8

The linear solutions to the multiple-unicast problem N is in a one-to-one correspondance to the linear solutions to the Circuit information problem N_C. The number of distinct linear solutions to N and N_C is the same.

IV. Guessing numbers and their link to Network Coding

Given a Circuit Information Problem N with source nodes $s_1, s_2, \ldots, s_n$ and target nodes $t_1, t_2, \ldots, t_n$. The graph G_N appears by identifying nodes s_j and t_j for $j = 1, 2, \ldots, n$.

The following theorem (that was first introduced in [18]) shows an interesting link between multiple-unicast network problems and guessing games:

Theorem 9

A Circuit information problem N with n input and n output nodes has a solution over an alphabet A if and only if the directed graph G_N has guessing number $k(G, s) \geq n$ if and only if $k(G, s) = n$.

Furthermore, the number of distinct solutions to N is identical to the number of distinct guessing strategies ρ for G_N that achieve guessing number n.

And the number of distinct linear solutions to N is identical to the number of distinct linear guessing strategies ρ for G_N that achieve guessing number n.

Proof: Consider the graph $G_N = (V, E)$. The set V of nodes can be divided into two disjoint sets: the set I of nodes in G_N that corresponds to the inner nodes in N (i.e. nodes that are not input or output nodes in N), and the set J of n nodes in G_N that corresponds to the n input and n output nodes in N. The set I consists of $|G_N| - n$ nodes. The sub-graph of G_N restricted to I is an acyclic graph (since N is acyclic). Thus as we already noticed for any strategy by the players (it does not matter which) the nodes in I all guess correctly their own die value with probability $(\frac{1}{s})^{|I|}$. But, this shows that the probability all players in G_N guess correctly their own die value is at most $(\frac{1}{s})^{|I|}$. Theorem 1 follows because this probability can be achieved if and only if the players in J (corresponding to the output nodes in N) are able to work out their own die value with probability 1 (given that all players in I correctly worked out their own die values).

To prove Theorem 9 consider a guessing strategy (i.e. a set of specific functions assigned to the nodes in G_N).

If we assign the same functions to the information network N (the output nodes $o_1, o_2, \ldots, o_n$ get assigned the specific functions assigned to the nodes J in G_N).

Conversely, any attempted solution to N can be converted to a guessing strategy by the same assignment. Thus the space of coding functions for N is in a natural 1-1 correspondence with the space of guessing strategies to the graph G_N. Furthermore,

a coding function for N solves the information problem for N if and only if the conditional probability that the n nodes in J guess correctly their own die values (*given* that all "inner" nodes (i.e. all the nodes in I) guess correctly their own die values) is 1. ♣

Now Theorem 3 follows by combining Theorem 7, Lemma 8 and Theorem 9.

V. Proving Theorem 1 using "path integrals"

Now I prove of Theorem 1, based on ideas akin to Feymann Integrals!

A very nice introduction to Feymans paths integrals can be found in [9]. Roughly the idea behind the path integral is that the probability that a Quantum Mechanical system in state v in a later mesurement is in state w, can be found by summing over all possible ways (all possible paths) the system can move from state v to state w. A key feature is that the contribution of different paths might cancel out if the paths have different phases. It should however be emphasised that my proof does not presupose any results or knowledge about Quantum Physics or the path integral. In fact the proof can - if one please - be presented as a simple pice of algebra and graph theory.

To begin with a simple example that illustrates how ideas akin to the path integral can be applied in network coding, consider the butterfly network in figure 1 (figure 6a). Each edge is assigned numbers 1 and -1 as illustrated in the next figure:

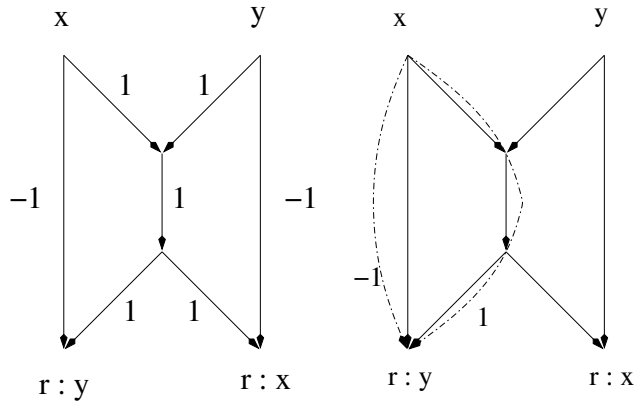

figure 6

Two edges assigned value 1 and -1 are out of phase. In agreement with Feynman's treatment the phase of a path is the product of the phases of edges along the path. If we consider ordinary (classical) routing, the message x can move to the lower left corner in two distinct ways (see Figure 6b). One path have phase -1 while the other have phase 1. Thus the two paths are out of phase and they cancel out (since they add up to 0), which implies that message x does not arrive at this node. On the other hand message x can, using ordinary (classical) routing also arrive at the lower right node. This can only happen in one way and the phase of the path is 1 which ensure that the message x arrive at this node (with its original phase).

Similar for message y. The two paths to the lowere right node have phase 1 and -1 and thus cancel out, while the single path to the lower left node have phase 1 so y only arrive at this node.

In general (towards proving Theorem 1) we assume that A is organised as an abelean group while R is a *-algebra acting on A. This covers the case where A is the additive structure of a field $F = (A, +, \times)$ and $R = A$ denote the set of actions, where $r \in R = A$ is multiplication by r.

To illustrate the idea with one more example consider again the information network from figure 3 (now represented as a circuit) and assume that $A = \{0, 1\}$.

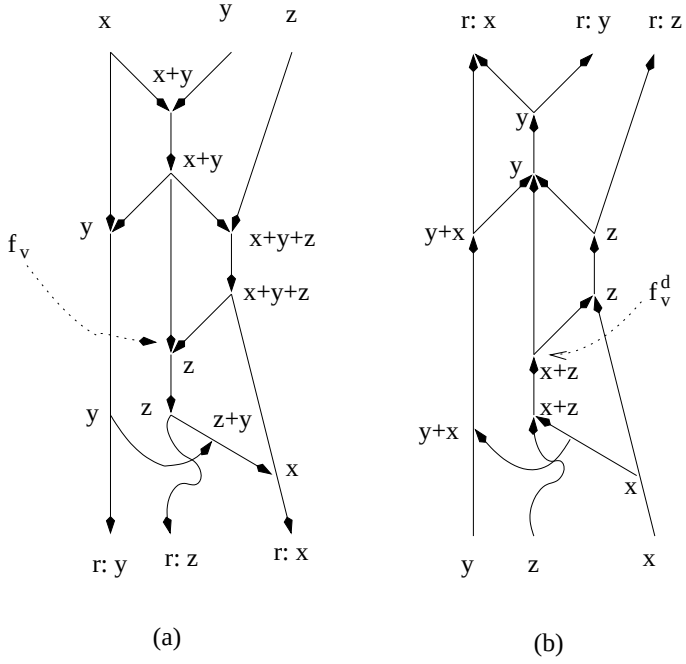

figure 7

To each node v in N is assigned a function $f_v(x, y, z) = a_v x + b_v y + c_v z$ with $a_v, b_v, c_v \in \{0, 1\}$.

Notice that each coefficient a_v is 1 if and only if there is an odd number of paths from the input node for x to v. And each coefficient b_v (c_v) is 1 if and only if there is an odd number of paths from the input node for y (z) to v.

To each node v in the dual network N^d is assigned a function $f_v^d(x, y, z) = a_v^d x + b_v^d y + c_v^d z$ with $a_v^d, b_v^d, c_v^d \in \{0, 1\}$.

For this network notice that each coefficient a_v^d is 1 if and only if there is an odd number of paths from the source node in N^d (target node in N) for x to v. And each coefficient b_v^d (c_v^d) is 1 if and only if there is an odd number of paths from the source node (target node in N) for y (z) to v.

The underlying graph N (and N^d) are both labelled so all edges have phase 1 and this labeling determine all the coding functions f_v and f_v^d as already explained. Since the labelling of N and N^d are the same, the two solutions indicated in figure 3 are infact representing the **same** underlying "reality":

There is an odd number of paths between a source node and a target node if and only if the source node corresponds to the same variable required by the target node.

This property is of course valid for N if and only if its valid for N^d.

Suppose we want to find all R-linear solutions to N (as well

as to N^d) over an alphabet A. Like already noticed there is a one-to-one correspondance between all R-linear encodings and labellings of the edges of N (as well as N^d) with elements in R. Let ρ be an assignment of N with each edge being assigned an element in R. The element in $r \in R$ associated to the edge is the *phase* of the edge with regards to the assignment ρ. Then a path p, $(v_1, v_2), (v_2, v_3), \ldots, (v_{r-1}, v_r)$ with labels $\lambda_{v_1,v_2}, \lambda_{v_2,v_3}, \ldots, \lambda_{v_{r-1},r} \in R$ has *path integral* (or *phase*) $\lambda_{v_{r-1},r} \ldots \lambda_{v_2,v_3} \lambda_{v_1,v_2}$ and is denoted $\int_p \rho$. Notice that $\int_p \rho$ is an element in R.

The coding function $f_v(x, y, z) = \lambda_{1v}x + \lambda_{2v}y + \lambda_{3v}z$ are determined (using this assignment ρ) such that λ_{1v} equals the sum of all paths integrals from input node x to v, λ_{2v} equals the sum of all paths integrals from input node y to v, while λ_{3v} equals the sum of all paths integrals from input node z to v. Let $P(u, v)$ denote the set of paths from u to v. Formally, we can then write $f_v(x, y, z) = (\Sigma_{p \in P(i_1,v)} \int_p \rho)(x) + (\Sigma_{p \in P(i_2,v)} \int_p \rho)(y) + (\Sigma_{p \in P(i_1,v)} \int_p \rho)(z)$.

The assignment ρ is a solution for N if and only if $f_{o_1}(x, y, z) = x$ $f_{o_2}(x, y, z) = y$ and $f_{o_3}(x, y, z) = z$.

Or equivalently the assignment ρ is a solution for N if and only if $\Sigma_{p \in P(i_j,o_k)} \int_p \rho = \delta_{jk}$ where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$.

The assignment ρ^* is defined such that $r^* \in R$ is assigned to the edge (u, v) (in N^d) if and only if the assignment ρ assigns $r \in R$ to edge (v, u) (in N) the element $r \in R$. In general $\int_{p^*} \rho = (\int_p \rho)^*$ where p^* denote the path p (in N) in opposite direction (in N^d).

But, then by the same argument as above (and since $0^* = 0$ and $1^* = 1$) ρ^* is a solution for N^d if and only if $\Sigma_{p^* \in P(i_j,o_k)} \int_{p^*} \rho^* = \delta_{jk}$ where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$.

Thus *ρ is a solution for N if and only if ρ^* is a solution for N^d*.

So far we only considered special networks like in figure 6 and figure 7.

Now let me consider the general case where N might denote any multiple-unicast network. The argument is essentially the same I just developed, except that in general the network N might have nodes of in-degree ≥ 2 and out-degree ≥ 2. In this case linear maps cannot be represented just be labeling the edges of N. However by "blowing up" such nodes and removing all superfluous edges, the graph N_C has the same set of R-linear solutions as N. The R-linear maps in N_C are in one-to-one correspondance with labellings ρ of edges in N_C with elements in R.

Each R-linear solution ρ determines a R-labelling ρ, and each R-labelling ρ determine a R-linear solution ρ. With slight abuse of notation, the linear solution and the corresponding labelling are both denoted by ρ.

A labelling determine a R-linear flow by letting a node v with incomming edges $(u_1, v), (u_2, v), \ldots, (u_d, v)$ labelled by $\lambda_{u_1,v}, \lambda_{u_2,v}, \ldots, \lambda_{u_d,v} \in R$ transmit the message $\lambda_{u_1,v}(z_1) + \lambda_{u_2,v}(z_2) \ldots \lambda_{u_d,v}(z_d)$ with $z_1, z_2, \ldots, z_d$ are the messages transmitted from $u_1, u_2, \ldots, u_d$. Notice, that node v transmit a message that can be expressed as a R-linear function of the messages transmitted from $u_1, u_2, \ldots, u_d$.

When the R-linear functions in the different nodes in N_C are composed each node transmit a message $f_v(x_1, x_2, \ldots, x_n)$ that can be expressed (using path integrals) as $(\Sigma_j \Sigma_{p \in P(i_j,v)} \int_p \rho)(x_j)$.

Like in the special case, in general a R-linear flow ρ defines a solution if and only if $\Sigma_{p \in P(i_j,o_k)} \int_p \rho = \delta_{jk}$ where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$.

By the same argument the R-linear ρ^* defines a solution if and only if $\Sigma_{p^* \in P(i_j,o_k)} \int_{p^*} \rho^* = \delta_{jk}$ where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$.

This shows:

Lemma 10

> Let N be an multiple-unicast information flow problem. There is a natural 1-1 correspondance ψ between the class of R-linear flows for N and labellings of N_C with elements from R. This map maps the set of R-linear solutions of N onto the set of R-linear solutions for N_C. There is also a natural 1-1 correspondance ψ' between the class of R-linear flows for N^d and labellings of $(N^d)_C = (N_C)^d$ with elements from R. This map maps the set of R-linear solutions for N^d onto the set of R-linear solutions for $(N^d)_C = (N_C)^d$.
> Furthermore, ρ is a R-linear solution for N_C if and only if ρ^* is a R-linear solution for $(N_C)^d$.

Theorem 1 follows directly from Lemma 10.

VI. Proof of Theorem 5

Consider the information network N in figure 4. A related information network was analysed [17]. I claim:

Lemma 11

> The information network N is solvable over the alphabet $\{0, 1\}$.

Proof: The 8 vertical channels need to send the messages x, y and z i.e. 8 messages. This is done be introducing the code (in the sense of error correcting codes) that consists of the 8 words $(0,0,0,0,0,0,0,0)$, $(1,0,0,0,0,0,0,0)$, $(0,1,0,0,0,0,0,0)$, $(0,0,1,0,0,0,0,0)$, $(0,0,0,1,0,0,0,0)$, $(0,0,0,0,1,0,0,0)$, $(0,0,0,0,0,1,0,0)$ and $(0,0,0,0,0,0,1,0)$. These 8 messages are the only messages send along the 7 vertical channels.

The solution is indicated in figure 8.

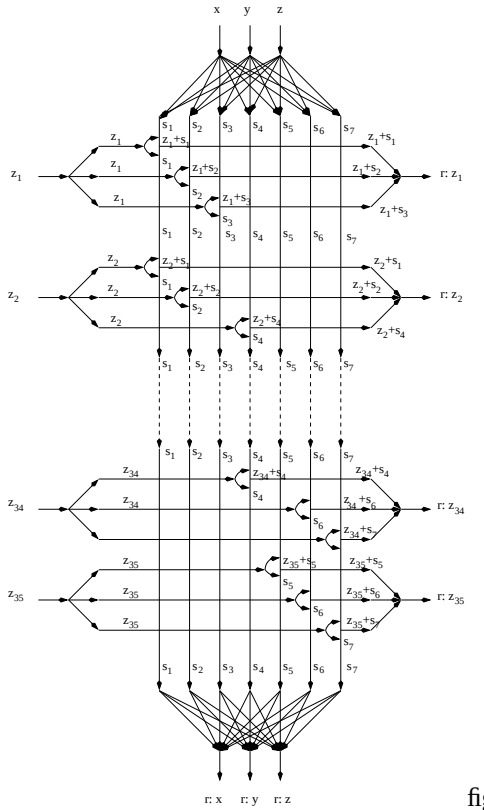

figure 8

For each variable $z_1, z_2, \ldots, z_{35}$ the corresponding message is send accross each of the three horizontal channels. On arrival two of these message are without error. Thus each message z_j can be reconstructed using a simple "majority" decision in the target node that requires z_j. Thus N has a solution. ♣

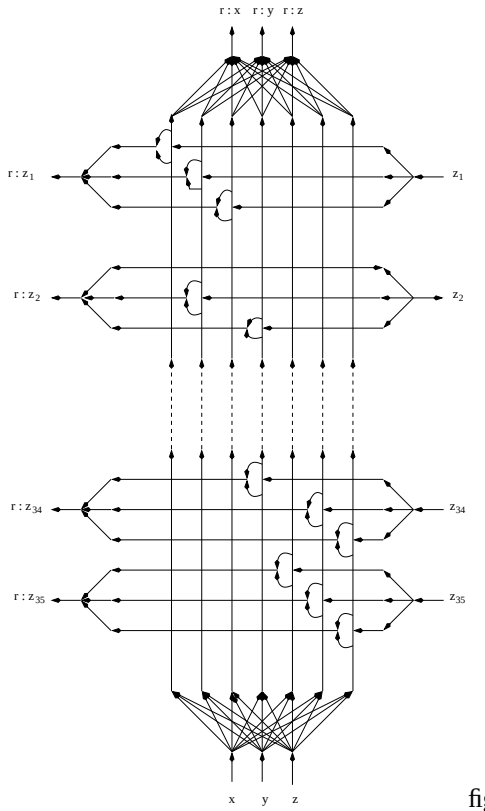

figure 9

Consider the reverse (dual) information flow problem N^d, in figure 8 and fix an information flow (it might or might not be a solution).

For each horizontal crossing a number of different things can happen (depending on the function $f(s,w)$). Consider figure 10. The variable w denotes any of the variables $z_1, z_2, \ldots, z_{35}$, while $s = s(x,y,z)$ denotes any message sent through the vertical channel. In general s is a function depending on x, y and z (see figure 9 and figure 10).

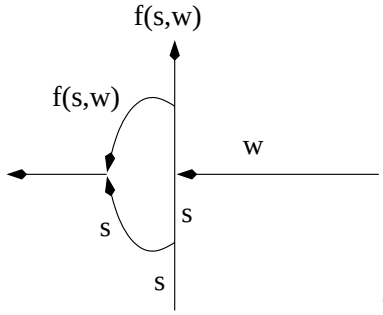

figure 10

A horizontal channel is *active* if the coding function $f(s,w)$ can be affected by w.

A horizontal channel is *inactive* if it not is active (i.e. if the coding function $f(s,w)$ can be expressed as a function of s). If a horizontal channel is active this can happen in two different ways:

One-sided activity: This happens if $f(s,w)$ (or $f(s,w)+1$) is given by one of the expressions $sw, sw, sw+s, sw+w$ or $sw+s+w$. In each of these cases the horizontal channel can force $f(s,w)$ to take a constant value. If $f(s,w) = sw$ $w = 0$ force $f(s,w) = 0$. Similarly for $f(s,w) = sw + w$, $w = 0$ force the function $f(s,w) = 0$. If $f(s,w) = sw + x$, $w = 1$ force $f(s,w) = 0$ and if $f(s,w) = sw + s + w$, $w = 1$ force $f(s,w) = 1$.

Two-sides activity: This happens if $f(s,w) = s + w$. In this case the horizontal channel has a choice. It can either force $f(s,w) = 0$ by letting $w = s$ (i.e. by letting $w(x,y,z) = s(x,y,z)$) or it can force $f(s,w) = 1$ by letting $w = s + 1$ (i.e. by letting $w(x,y,z) = s(x,y,z) + 1$).

In both cases it is possible to construct a horizontal message w that 'blocks' the vertical information flow s (since the horizoltal channel can send a message $w = w(x,y,z)$ such that $f(s,w) = f(s(x,y,z), w(x,y,z)$ is independent of x, y and z).

Based on this observation an horizontal channel (corresponding to a variable z_j) is said to *block* the vertical channel it is linked to. And we say that the variable z_j can be used to block the vertical channel.

Lemma 12

> There is no solution to N^d where five variables $z_{j_1}, z_{j_2}, z_{j_3}, z_{j_4}$ and z_{j_5} can be used to block five distinct vertical channels.

Proof: Assume that N^d has a solution where five variables $z_{j_1}, z_{j_2}, z_{j_3}, z_{j_4}$ and z_{j_5} can be used to block five distinct vertical channels.

Let $s_1 = s_1(x,y,z), s_2 = s_2(x,y,z), \ldots, s_7 = s_7(x,y,z)$ denote the vertical flows. By definition there exists five functions $w_{j_1}(x,y,z), w_{j_2}(x,y,z), \ldots, w_{j_5}(x,y,z)$ such that

five of the functions $s_1, s_2, \ldots, s_7$ are constant functions independent of x, y and z. But then there is only two vertical channel to send 8 messages which is impossible, ♣

To get a contradiction I show the following simple combinatorial lemma:

Lemma 13

Let $S = \{1, 2, 3, 4, 5, 6, 7\}$ be a set with 7 elements and let l be a choice function that for each 3-element subset B selects an element $l(B) \in B$. Then there exists a 5-element subset $W \subset S$ as well as 5 3-element subsets B_1, B_2, B_3, B_4 and B_5 such that $W = \{l(B_1), l(B_2), l(B_3), l(B_4), l(B_5)\}$.

Proof: Pick any 3-element subset B_1 and let $v_1 := l(B_1)$. Let B_2 be any 3-element subset of $S \backslash \{v_1\}$ and let $v_2 := l(B_2)$. Let B_3 be any 3-element subset of $S \backslash \{v_1, v_2\}$ and let $v_3 := l(B_3)$. We continue like this.

Let B_4 be any 3-element subset of $S \setminus \{v_1, v_2, v_3\}$ and let $v_4 := l(B_4)$. Finally, let B_5 be any 3-element subset of $S \setminus \{v_1, v_2, v_3, v_4\}$. The set $S \setminus \{v_1, v_2, v_3, v_4\}$ contains 3 elements so there is such a set. Let $v_5 := l(B_5)$. The set $W = \{v_1, v_2, v_3, v_4, v_5\}$ contains 5 elements and $W = \{l(B_1), l(B_2), l(B_3), l(B_4), l(B_5)\}$.♣

This implies:

Lemma 14

For any solution to N^d (over $A = \{0, 1\}$) there exists five variables $z_{j_1}, z_{j_2}, z_{j_3}, z_{j_4}$ and z_{j_5} that can be used to block five distinct vertical channels.

Proof: Assume that N^d has a solution over $A = \{0, 1\}$. For each variable z_j we have naturally associated a 3 element subset $B_j \subset \{1, 2, 3, 4, 5, 6, 7\}$. Let $l(B_j)$ denote the smallest number of the 3 vertical channels that correspond to an active channel. Using Lemma 8, notice that there exists a set W corresponding to five variables $z_{i_1}, z_{i_2}, \ldots, z_{i_5}$ such that each of the variables can be used to block one vertical channels.♣

This Lemma shows (when combined with lemma 12) that:

Lemma 15

The information network N^d is not solvable over the alphabet $A = \{0, 1\}$.

Theorem 5 now follows by combining Lemma 11 and Lemma 15.

VII. A non-constructive proof of Theorem 5

Finally let me present another proof of Theorem 5 that is based on two facts:

- (i) Each solution to N requires the use of *all* vertical crosssings.
- (ii) If there is a solution to N^d it does *not* need the use of all vertical crossings.

When combined we infer that **either** N^d has no solution (in which case N is irreversible) **or** N^d has a solution, in which case we according to (ii) can remove one crossing from N and obtain N' such that $(N')^d$ still has a solution. But, since according to (i) any solution to N requires the use of all vertical crossings N' has no solution. Thus $(N')^d$ is irreversible.

There are 105 vertical crossings of N so all this argument shows is that one of 106 networks (the 105 networks $(N')^d$ or N itself) is irreversible. Of course since we have already showed that N is irreversible, none of the networks $(N')^d$ are in fact solvable.

Item (i) follows using one of the most fundamental results on forbidden configurations of matrices:

Proposition 16 [20], [21], [22]

Let K_k denote the $k \times 2^k$ matrix of all possible $(0, 1)$-columns on k rows. Then each $m \times s$ $(0, 1)$-matrix A (with no repeated columns) has configuration K_k (i.e. a submatrix of A is and row and column permutation of F) if

$$s = \binom{m}{k-1} + \binom{m}{k-2} + \ldots + \binom{m}{0} + 1$$

The value of s is in general the best possible.

A very basic induction proof of this can be found in [3]. For my application I only need the very special case where $k = 2$ and $m = 6$, which gives $s = \binom{6}{1} + \binom{6}{0} + 1 = 8$.

Lemma 17

Consider again the network N in figure 4. Assume that one of the three horizontal channels that links a variable z_j with the node where it is required is removed. Then the resulting network N' has no solution over $A = \{0, 1\}$.

Proof: Consider the network N'. The 7 vertical channels send messages $s_1, s_2, \ldots, s_7$. Consider the variable z_j for which there is only two vertical crossings (channels) in N'. The message z_j can (if we assume N' is solvable) be reconstructed from two messages that must be on one of the following forms:

(1) $s_i + z_j$ and $s_k + z_j$

(2) $s_i + z_j$ and s_k

(3) s_i and s_k.

Case (3) give no hope of reconstructing z_j. Case (2) only make is possible to reconstruct z_j if s_i or $s_i + s_k$ is a constant (i.e. independent of x, y and z). Finally, z_j can only be derived from 1 if at least one of s_i or s_k are constant. In all cases one of the vertical channels is superfluous (since it is only sending "dummy" information) and is not needed for transmitting the messages x, y and z. Assume, channel 7 is sending dummy messages. For each of the 8 settings of the variables x, y and z consider the flow $(s_1, s_2, s_3, s_4, s_5, s_6)$ throgh the 6 remaining vertical channels. We can list these 8 distinct words as a 6×8 $(0 - 1)$-matrix:

$$\begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} & s_{37} & s_{38} \\ s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} & s_{47} & s_{48} \\ s_{51} & s_{52} & s_{53} & s_{54} & s_{55} & s_{56} & s_{57} & s_{58} \\ s_{61} & s_{62} & s_{63} & s_{64} & s_{65} & s_{66} & s_{67} & s_{68} \end{pmatrix}$$

But then the this matrix contains (according to the special case of Proposition 16 for $k = 2$ and $m = 6$ since $s = 8$), the configuration K_2 i.e.

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

In other words there exists $i, j \in \{1, 2, 3, 4, 5, 6\}$ with $i \neq j$ such that s_i and s_j can send all possible pairs of $(0-1)$-messages. Now let $r \in \{1, 2, \ldots, 6\} \setminus \{i, j\}$ be arbitrary and let z_u denote the variable that corresponds to the 3-element subset $\{i, j, r\} \subseteq \{1, 2, 3, 4, 5, 6\}$. The message z_u has to be constructed from three messages. They might for example be z_u+s_i, z_u+s_j and z_u+s_r (the other cases e.g. (s_i, z_u+s_j, z_u+s_r), (s_i, s_j, z_u+s_r), (z_u+s_i, s_j, z_u+s_r) are treated similary). The task is to derive z_u from these three messages. Since however s_i and s_j take all 4 combinations we conclude that s_r must be uniquely determined by s_i and s_j. But r was chosen arbitrarily in $\{1, 2, \ldots, 6\} \setminus \{i, j\}$, so we conclude that s_r is uniquely determined from s_i and s_r for each $r \in \{1, 2, \ldots, 6\}$. This is a contradiction since s_i and s_j can determine at most 4 words and not the 8 words required. ♣

Item (ii) follows from the slightly stronger statement:

Lemma 18

There is no solution (over $A = \{0, 1\}$) of N^d in which all vertical channels are active.

Proof: Assume that there is a solution to the information flow problem N^d where all $105 = 3 \times 35$ horizontal channels are active. We consider only the variables $z_1, z_{16}, z_{26}, z_{32}$ and z_{35}. The variable z_1 affects channels $1, 2$ and 3, z_{16} affects channels $2, 3, 4$, z_{26} affects channels $3, 4, 5$, z_{32} affects channels $4, 5, 6$ and z_{35} affects channels $5, 6$ and 7. From this we can clearly 'jam' channel $1, 2, 3, 4$ and 5. To see this we choose the value of z_1 (z_1 might be a function of x, y and z) such that any message that passes through channel 1 is independent of x, y and z. Next, choose a value of z_{16} such that any message that passes through channel 2 is independent of x, y and z. Then the variable z_{26} is selected (as a suitable function of x, y and z) such that any message that passes through channel 3 is independent of x, y and z. Finally, after suitable functions $z_{32}(x, y, z)$ and $z_{35}(x, y, z)$ have been chosen, the first five vertical channels have been blocked in such a way that no information about x, y or z can be transmitted through these channels. This leaves open only the vertical channels 6 and 7 and thus only 4 messages can be sent through the vertical channels. To transmit the messages x, y and z successfully through the next work we need to be able to send 8 messages through the vertical channels. This is a contradiction and thus we conclude that not all horizontal channel are active ♣

An anonymous referee pointed out that this lemma in fact shows that one of $5 \times 3 + 1 = 16$ rather that one of $35 \times 3 + 1 = 106$ specific multiple-unicast problems are irreversible. If this can be improved further using the type of argument just given is, of course, somewhat irrelevant since we know that only N and none of the 105 other multiple-unicast problems are irreversible over $A = \{0, 1\}$).

REFERENCES

[1] S. Medard M.-Koetter R. Acedanski, S. Deb. How good is random linear coding based distribued network storage? In *Procedings of the NetCod 2005 conference*.

[2] R Ahlswede, N Cai, Li, and R Yeung. An algebraic approach to network coding. page 104, 2001.

[3] R.P. Anstee and A. Grigss. Small forbidden configurations. *Graphs and Combinatorics*, 13:97–118, 1997.

[4] K.R. Bhattad, K. Narayanan. Weakly secure network coding. In *Procedings of the NetCod 2005 conference*.

[5] N. Cai and R.W. Yeung. Network coding and error correction. In *ITW 2002 Bangalore*, pages 119–122, 2002.

[6] Deb, Choute, Medard, and Koetter. Data harvesting: A random coding approach to rapid dissemination and efficient storage of data. In *INFOCOM*, 2005. Submitted.

[7] R Dougherty, C Freiling, and K Zeger. Insufficiency of linear coding in network information flow. 2004. To appear.

[8] R Dougherty, C Freiling, and K Zeger. Unachievability of network coding capacity. *IEEE Transactions on Information Theory and IEEE/ACM Transactions on Networking (joint issue)*, 2005.

[9] R Feynman. *QED: The Strange Theory of Light and Matter*. Princeton University Press.

[10] G.D. Forney. Codes on graphs: Normal realization". *IEEE Transactions on Information Theory*, 47:520–548, Feb. 2001.

[11] C. Fragouli and E Soljanin. A connection between network coding and convolutional codes. In *IEEE International Conference on Communications*, 2004.

[12] T Ho, M Medard, and R Koetter. An information theoretic view of network management. In *Prooceeding of the 2003 IEEE Infocom*.

[13] R. Koetter, M Effros, T Ho, and M Medard. Network codes as codes on graphs. In *Proceeding of CISS*, 2004.

[14] R Koetter and M Medard. An algebraic approach to network coding. In *Proocedings of the 2001 IEEE International Symposium on Information Theory*.

[15] R Koetter and M Medard. Beyond routing: An algebraic approach to network coding. In *Proceedings of the 2002 IEEE Infocom*, 2002.

[16] K. Rabaey J Petrovic, D. Ramchandran. Overcomming untuned radios in wireless networks with network coding. In *Proocedings of the NetCod 2005 conference*.

[17] S. Riis. Linear versus non-linear boolean functions in network flow. In *Proceeding of CISS 2004*.

[18] S Riis. Utilising public information in network coding. Technical report, Queen Mary, University of London, 2005.

[19] S. Riis and R Ahlswede. Problems in network coding and error correcting codes. In *Procedings of the NetCod 2005 conference*.

[20] N Sauer. On the density of families of sets. *Journal Combin. Th. Ser A*, 13:145–147, 1972.

[21] S Shelah. A combinatorial problem: Stability and order for models and theories in infinitary languages. *Pac. J. Math.*, 4:247–261, 1972.

[22] V.N. Vapnik and A.Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Th. Prob. and Applics*, 16:264–280, 1971.

[23] C. Boudec J-Y. Widmer, J. Fragouli. Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding. In *Procedings of the NetCod 2005 conference*.

[24] Wu, Chou, and Kung. Information exchange in wireless networks with network coding and physical-layer broadcast. Technical Report MSR-TR-2004-78, Microsoft Technical Report, Aug. 2004.

[25] Yeung and Zhang. Distributed source coding for satellite communications. *IEEE Trans. Inform. Theory*, (IT-45):1111–1120, 1999.