Problems in Network coding and error correcting codes

Søren Riis & Rudolf Ahlswede

23rd February 2005

Introduction

In most of todays information networks messages are send in packets of information that is not modified or mixed with the content of other packets during transmission. This holds on macro level (e.g. the internet, wireless communications) as well as on micro level (e.g. communication within processors, communication between a processor and external devises).

Today messages in wireless communication are sent in a manner where each active communication channel carries exactly one "conversation". This approach can be improved considerably by a cleverly designed but sometimes rather complicated channel sharing scheme (network coding). The approach is very new and is still in its pioneering phase. Worldwide only a handful of papers in network coding were published year 2001 or before, 8 papers in 2002, 23 papers in 2003 and over 25 papers already in the first half of 2004; (according to the database developed by R. Koetters). The first conference on Network Coding and applications is scheduled for Trento, Italy April 2005. Research into network coding is growing fast, and Microsoft, IBM and other companies have research teams who are researching this new field. A few American universities (Princeton, MIT, Caltech and Berkeley) have also established research groups in network coding.

The holy grail in network coding is to plan and organise (in an automated fashion) network flow (that is allowed to utilise network coding) in a feasible manner. With a few recent exceptions [5] most current research does not yet address this difficult problem.

The main contribution of this paper is to provide new links between Network Coding and combinatorics. In this paper we will elaborate on some remarks in [8, 9]. We will show that the task of designing efficient strategies for information network flow (network coding) is closely linked to designing error correcting codes. This link is surprising since it appears even in networks where transmission mistakes never happen! Recall that traditionally error correction, is mainly used to reconstruct messages that have been scrambled due to unknown (random) errors. Thus error correcting codes can be used to solve network flow problems even in a setting where errors are assumed to be insignificant or irrelevant.

It should be pointed out that the idea of linking Network Coding and Error Correcting Codes (in a context where networks NOT are assumed to be error-free) was already presented in [4]. In this paper Cai and Yeung obtained network generalisations of the Haming bound, the Gilbert-Varshamov bound, as well as the singleton bound for classical error-correcting codes.

The basic idea and its link to work by Euler.

The aim of the section is to illustrate some of the basic ideas in network coding. To illustrate the richness of these ideas we will show that solving the flow problem for certain simple networks, mathematically is equivalent to a problem that puzzled Euler and was first solved fully almost 200 years later! First consider the following network:

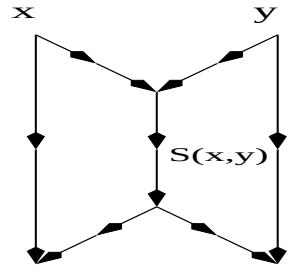

figure 1

The task is to send the message x from the upper left node, to the lower right node labelled $r : x$ (indicating that the node is required to receive x) as well as to send the message y from the upper right node, to the lower left node labelled $r : y$. Suppose the messages belong to a finite alphabet $A = \{1, 2, \ldots n\}$. If the two messages are sent as in ordinary routing (as used on the world wide web or in an ordinary wireless network) there is a dead lock along the middle channel where message x and message y will clash. If instead we send the message $s_{x,y} = S(x, y) \in A$ through the middle channel, it is not hard to show that the problem is solvable if and only if the matrix $(s_{i,j})_{i,j \in A}$ forms a latin square (recall that a latin square of order n is an $n \times n$ matrix with entries $1, 2, \ldots n$ appearing exactly once in each row and in each column). We can now link this observation to work by Euler! Consider the following extension of the previous flow problem:

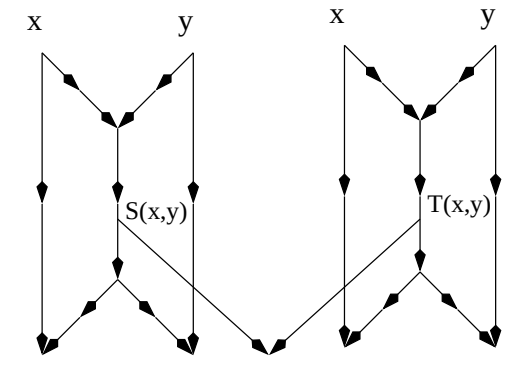

figure 2

Now the task is to send the message x and the message y to each of the five nodes at the bottom. To do this each of the matrices $\{s_{x,y}\}$ and $\{t_{x,y}\}$ must, according to the previous observation, be latin squares. However, the latin squares must also be orthogonal i.e. if we are given the value $s \in A$ of the entry $s_{x,y}$ and the value $t \in A$ of the entry $t_{x,y}$, the values of x and y must be uniquely determined. Thus, we notice that:

Proposition: There is a one-to-one correspondence between solutions to the flow problem in figure 2 with alphabet A and pairs of orthogonal latin squares of order $|A|$.

The problem of deciding when there exist such two orthogonal latin squares has an interesting history. Euler knew (c.1780) that there was no orthogonal Latin square of order 2 and he knew constructions when n is odd or divisible by 4. Based on much experimentation, Euler conjectured that orthogonal Latin squares did not exist for orders of the form $4k + 2, k = 0, 1, 2, \ldots$. In 1901, Gaston Tarry proved (by exhaustive enumeration of the possible cases) that there are no pairs of orthogonal Latin squares of order 6 - adding evidence to Euler's conjecture. However, in 1959, Parker, Bose and Shrikhande were able to construct two orthogonal latin squares of order 10 and provided a construction for the remaining even values of n that are not divisible by 4 (of course, excepting $n = 2$ and $n = 6$). From this it follows:

Proposition (corollary to the solution to Euler's question): The flow problem in figure 2 has a solution if and only if the underlying alphabet does not have 2 or 6 elements.

The flow problem in figure 2 might be considered somewhat 'unnatural' however the link to orthogonal latin squares is also valid for very natural families of networks. The multi-cast problem $N_{2,4,2}$ defined below has for example recently been shown to be essentially equivalent to Eulers question [6].

Network coding and its links to error correcting codes

The task of constructing orthogonal latin squares can be seen as a special case of constructing error correcting codes. There is, for example, a one-to-one correspondence between orthogonal latin squares of order $|A|$ and $(4, |A|^2, 3)$ $|A|$-ary error correcting codes. [1]

Next consider the flow problem:

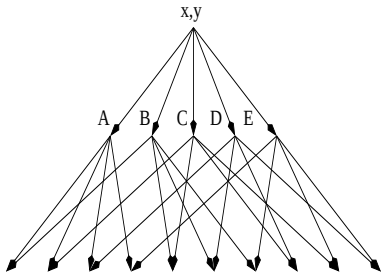

figure 3

Assume each channel in this multi-cast network has the capacity to carry one message (pr. unit time). Assume that the task is to send two messages $x, y \in A$ from the top nodes to each of the 10 bottom nodes. It can be shown that this flow problem has a solution over the alphabet A if and only if there exist an $(5, |A|^2, 4)$ $|A|$-ary error correcting code. It has been shown that there exit such codes if and only if $|A| \notin \{2, 3, 6\}$. The flow-problem in figure 3 can be generalised. Consider a network $N_{k,r,s}$ such that it consists of k messages $x_1, x_2, \ldots x_k \in A$, that are transmitted from a source node. The source node is connected to a layer containing r nodes, and for each s element subset of r (there are $\left(\begin{array}{c} r \\ s \end{array} \right) = \frac{r!}{(r-s)!r!}$ such) we have a terminal node. The task is to insure that each message $x_1, x_2, \ldots, x_k \in A$ can be reconstructed in each of the terminal nodes. Notice the previous network flow problem is $N_{2,5,2}$. In general it can be shown [9, 8]:

Proposition [2]: The flow problem $N_{k,r,s}$ has a solution if and only if there exists an $(r, |A|^k, r-s+1)$ $|A|$-ary error correcting code.

Essentially, there is a one-to-one correspondence between solutions to the network flow problem $N_{2,4,2}$ and $(4, 4, 3)$ 2-ary error correcting codes, i.e. orthogonal latin squares. Thus despite of the fact that the flow problem in figure 2 has a topology very different from the $N_{2,4,2}$ problem, the two problems essentially have the same solutions!

Next, consider the famous Nordstrom-Robinson code: This code is now known to be the unique binary code of length 16, minimal distance 6 containing 256 words. The point about this code is that it is non-linear, and is the only $(16, 256, 6)$ 2-ary code. Again we can apply the proposition to show that the multi-cast problem $N_{8,16,11}$ has no linear solution over the field F_2, while it has a non-linear solution. Are phenomena like this just rare isolated incidences or much more widespread?

The classical theory for error correcting needs extensions

The previous sections indicate how it is possible to recast and translate network flow problems into the theory of error correcting codes (thus, using standard results in coding theory, it is possible to translate network flow problems into questions about finite geometries). Another approach is outlined in [7].

In [9, 8] the first example with only non-linear solutions was constructed. Unlike other examples this construction seems to go beyond standard results in error correcting codes. The construction is based on the following network:

[1] Recall that a (n, c, d) r-ary error correcting code C consists of c words of length n over an alphabet containing r letters. The number d is the minimal hamming distance between distinct words $w, w' \in C$.

[2] The fact that known bounds on maximum distance separable codes can be applied to bound the required alphabet-size was shown in [10]

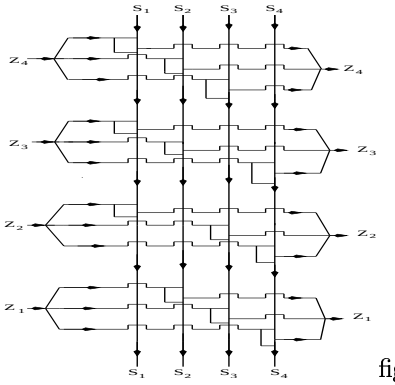

figure 4

The network N in figure 4 has the curious property (like $N_{8,16,11}$) that the maximal through-put can only be achieved if non-linear flows are allowed (i.e non-linear boolean functions are needed in any solution). Furthermore it turns out that any code optimising the vertical flows has to be a "minimal distance code" [9, 8]. This phenomena is interesting since a minimal distance code from a traditional perspective is very bad (as it essentially has the worst possible error correcting capability).

This example is one of a collection of examples that suggests that the classical theory of error correcting codes needs to be extended and developed in order to serve as a basis for network coding. See also [3, 1, 2] more results pointing in this direction.

References

[1] R Ahlswede. Remarks on shannon's secrecy systems. *Probl. of Control and Inf. Theory*, 11(4):301–308, 1982.

[2] R Ahlswede and G Dueck. Bad codes are good ciphers. *Probl. of Control and Inf. Theory*, 11(5):337–351, 1982.

[3] R Ahlswede and L.H Khachatrian. The diametric theorem in hamming spaces – optimal anticodes. *Advances in Applied Mathematics*, 20:429–449, 1996.

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

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

[6] R Dougherty, C Freiling, and K Zeger. Linearity and solvability in multicast networks". In *Proceeding of CISS*, 2004.

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

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

[9] S Riis. Linear versus non-linear boolean functions in network flow (draft version). Technical report, November 2003.

[10] M. Tavory, A. Feder and D. Ron. Bounds on linear codes for network multicast. Technical Report 33, Electronic Colloquium on Computational Complexity, 2003.