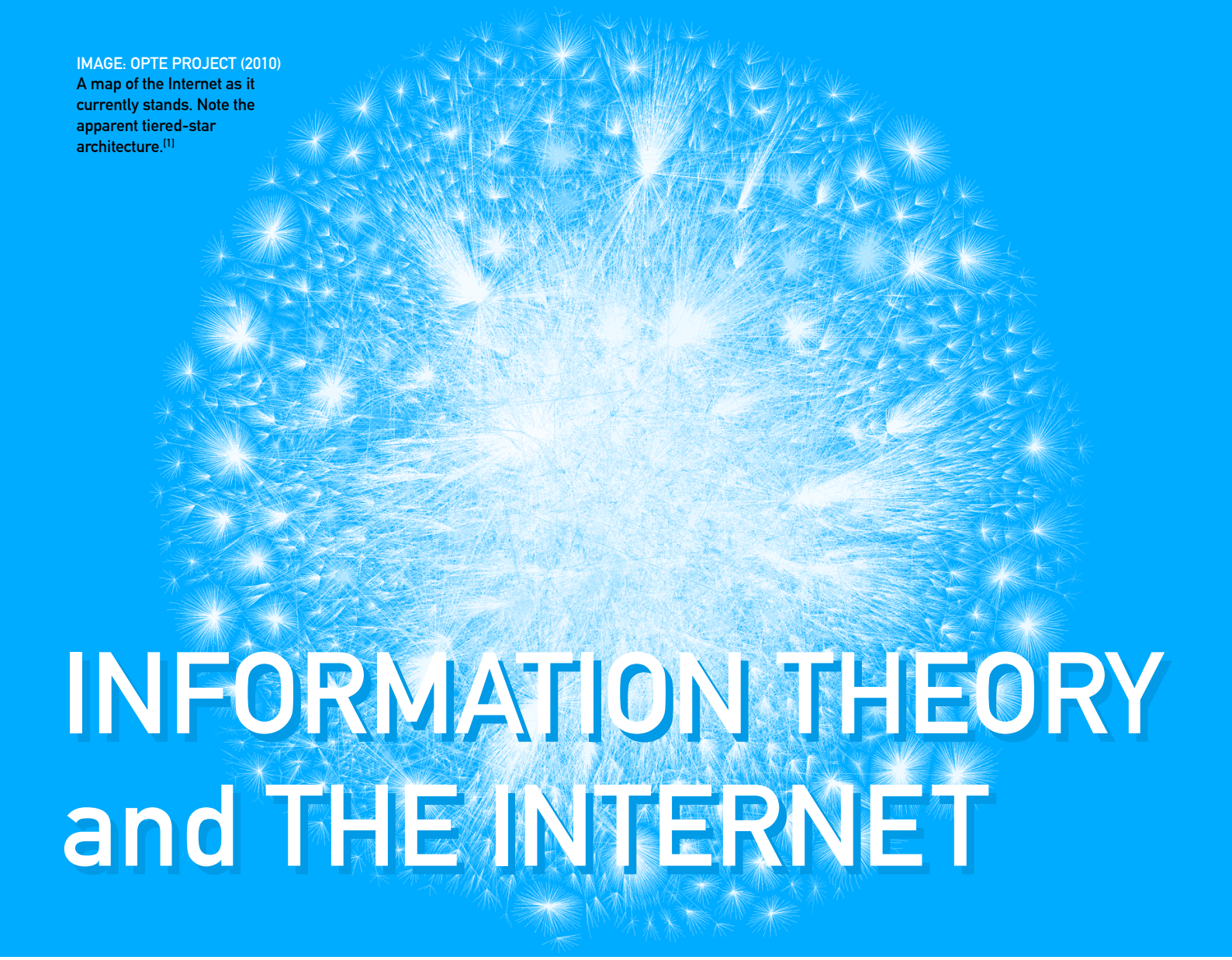


IMAGE: OPTE PROJECT (2010)
A map of the Internet as it currently stands. Note the apparent tiered-star architecture.[1]

INFORMATION THEORY and THE INTERNET

New developments in information theory promise improved online service through novel use of modern network architectures

Joe Cridge

October 2014

THE INTERNET is the name given to the global network of networks which has come to play such an important role in modern day-to-day life. Employing electrical, optical and electromagnetic (wireless) transmission media and connecting together billions of devices, the Internet is now widely relied on by businesses and individuals for instant communication (be it email, voice or video conferencing), social networking, data transfer, financial transactions and commerce, media and news broadcast and access to the wealth of information that is the World Wide Web.

As it exists today, the Internet (pictured topologically above) has grown dramatically from its origins as a DARPA experiment in distributed networking. Based on work carried out in the 1960s by Paul Baran and Donald Davies, the project investigated the possibility of a 'survivable' network where multiple connection routes between any two nodes provided resistance to node failure (see diagram, right). Baran envisaged a network where dynamic routing of information would allow optimum use of available network capacity, a concept refined by Davies into what is now known as *packet switching*.

Information theory – which was born in 1948 with Claude Shannon's landmark paper *A Mathematical Theory of Communication* – has played a key role in the development of this and other digital communications systems we take for granted today, but has to an extent become constrained by the limitations present in the telegraph and telephone systems it was originally built to describe. New research is redefining Shannon's work for the Internet Age, with the potential to provide improvements in efficiency and reliability that we can all look forward to.

In particular, a new technique known as *network coding* is challenging whether packet switching is the best way to make use of our global communications infrastructure. Are big changes for the Internet just around the corner?

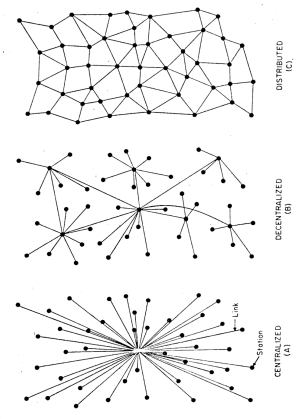


DIAGRAM
Network topologies as described by Baran.[2] Centralised networks (bottom) are susceptible to single point failure, while distributed (top) networks are more resilient. In practise a decentralised (middle) or tiered-star topology is common, as can be seen in the Internet.

EXAMPLE
SOURCE CODING

Many sources generate signals which contain a degree of redundancy – the number of bits in symbols they use to transmit a message is greater than the number of bits of information the message actually contains. Knowledge of the statistical properties of a source allows the choice of a code which removes extra bits.

Consider a source which chooses from the letters A, B, C, D, with respective probabilities 1/2, 1/4, 1/8, 1/8. We could choose to use 2-bit symbols to represent each letter in the channel, say A = 00, B = 01, C = 10, D = 11. Actually, the average information generated per letter choice is less than two bits, since (for instance) A is always most likely to be chosen. We can compress the signal by choosing a code which uses shorter symbols for more common letters: A = 0, B = 10, C = 110, D = 111.

Message: AABADADBDACAABABBBABBAABBBAAABDACBAAADDA

2-bit code: 00000100110011011100100000010001010100010100000101010000000111011001000000111100

Short code: 00100111011110111011000100101010010100010101000010111011010000011111110

Both encoded signals contain exactly the same information (you can recover the message exactly from both of them), but the first signal is 80 bits long while the second is 70 bits. Assigning symbols so that the average information per symbol matches the average information per letter choice allows the shortest signal to be sent. In this case H = 1.75 bits/symbol – not two.

KEY CONCEPTS

CHANNEL – the physical medium through which a message is transmitted

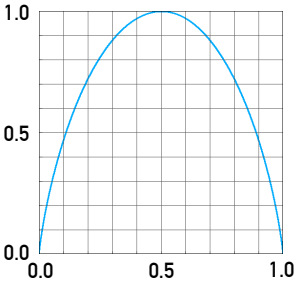
CODE – the particular way the available symbols are used

MESSAGE – the object of meaning to be transmitted

SIGNAL – the message represented in terms of transmittable symbols

SOURCE – the process which selects which message to send

SYMBOLS – the set of distinguishable elements which can be transmitted through a particular channel, such as current or voltage levels



GRAPH

Shannon entropy H (in bits) plotted against probability p for the choice of the first symbol from of a set of two. Entropy is highest when p = q = 0.5 since the outcome is completely uncertain – at all other values of p there is some degree of certainty in the result.

It's All About Uncertainty

THE INTERNET is all about transmitting information, but until the publication of Shannon's paper it was unclear what exactly information was, let alone how one should go about measuring it. In the 1920s Shannon's predecessors at Bell Telephone Laboratories had made some progress in quantifying information, but their analysis was by no means complete.

In particular, Ralph Hartley put forward an argument in his 1928 paper *Transmission of Information* that for every symbol chosen for transmission, the possibility of any other symbol taking that position is eliminated, and thus as more and more symbols are selected the signal becomes more specific – the number of different messages which it could represent is rapidly reduced. This corresponds to the transmission of meaningful information, and so Hartley reasoned that the information content of a message represented by N symbols in length, where each symbol is chosen from a total set of s symbols, must be related to the total number of messages which *could* have been sent, s^N. Clearly s^N is not in itself a useful measure since, in general, a slight increase in specificity does not cause an exponential increase in understanding (for example, you would expect to store on two hard drives twice – not four times – what you can store on one), but it follows that $\log(s^N)$ will be.[3]

Consider a source which selects symbols from a set of two, with constant probability p for selecting the first symbol and $q = 1-p$ for the second. This could be from recorded speech, for instance, which is being converted into a stream of logical '0's and '1's for transmission. When $p = q = 0.5$ we are completely uncertain about which of the two symbols the source will select, and so we could use Hartley's measure to say that with each symbol choice the source generates log(2) hartleys of information. In fact, this corresponds to the smallest quantity of information that can be stored electronically, and so it is useful to define it as one *bit* (information in bits can then easily be calculated by taking logarithms in base two). But what about when $p \neq q$? Or in the extreme case when, say, $p = 1$? Clearly the source is no longer generating as much information with each choice as it was when $p = q$ because now we (and more importantly, the receiver) have an idea in advance of which symbol to expect.

Shannon refined this idea – that information is defined by the uncertainty it resolves – by more rigorously incorporating uncertainty into information measurement. He defined a measure called *entropy* which improves on Hartley's work by taking into account the probabilities of each symbol being chosen; for a set of n possible symbols where the probability of the i^{th} symbol being chosen is p_i, the Shannon entropy is given by:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

and has been plotted as a function of p for the binary source described above – see graph, left.[4]

This expression is derived uniquely from the mathematical properties expected of uncertainty (continuity in the p_i, continual increase as n is increased if all p_i are held equal, and linear behaviour if one choice is split into two successive choices), but can also be interpreted as follows:

- Our previous measure of information gave the information generated by a choice from n symbols as $\log_2(n)$, though it assumed that the probability of each symbol being chosen, p_i, was equal. It follows that all $p_i = n^{-1}$ in this case and so we can write $\log_2(n) = \log_2(p_i^{-1}) = -\log_2(p_i)$.
- Though the p_i are no longer necessarily equal, it holds that $-\log_2(p_i)$ gives a measure of the information generated in the event of the i^{th} symbol being chosen.
- Hence the expression for H is the sum of the information generated for each possible symbol choice, weighted by the probability of that symbol being chosen.

Entropy is then a measure of the average information content per symbol.

Before applying this mathematical foundation to communication over a physical channel, it is worth considering what is meant when – as in our binary source example with $p \neq q$ – the distribution of symbol probabilities decreases the entropy below that expected by Hartley. In this case, the source is coding the message with a degree of *redundancy*; since each symbol choice is not completely random, the potential for each symbol to carry information is not being fully utilised. Properly coding the message allows this redundancy to be removed – known as *source coding* – and is responsible for all of the data compression techniques such as ZIP and JPEG which we employ to minimise download times of files and Web pages (see example above).

EXAMPLE
CHANNEL CODING

Though it has been shown that removal of redundancy increases transmission rate by shortening signal duration, redundancy can also be added deliberately to allow for the correction of errors when transmitting over a noisy channel. In this case, the redundancy to be added must be carefully chosen to account for the type of noise on the channel.

Shannon showed that a code can always be found which allows for all noise related errors to be corrected, provided that the transmission rate does not exceed the channel capacity. This example – known as the Hamming(7,4) code and presented in Shannon's paper[5] – demonstrates how adding redundancy can achieve this in a simple case.

Consider a channel using symbols 0 and 1, where a type of random noise exists which causes either one or none of the symbols in every block of seven to be received incorrectly, where each of these outcomes is equally probable.

The capacity of the channel is the maximum rate of information transfer, which we can write $C = \mathrm{Max}[H(y)–H_x(y)]$ using the same notation as below. Each incoming block represents 7 bits of entropy (though not all of it useful information), while the conditional entropy, representing the uncertainty added due to noise, can be calculated from the below expression as $– 8 \times (1/8)\log_2(1/8) = 3$ bits per block. The capacity is then 4 bits per block.

The Hamming code which achieves this capacity works as follows: for each block, use the 1st, 2nd and 7th bits as check bits and the remaining four to carry information. Denoting the i^{th} bit Mi, choose bits M1, M2 and M7 such that

$$A = M1 + M3 + M5 + M7, \quad B = M2 + M3 + M6 + M7, \quad C = M4 + M5 + M6 + M7$$

are all even numbers. The result is that the index of the incorrect bit will be given by the value of $A + 2B + 4C$, and will be zero if there was no error.

Optimising Transmissions

IN THE SAME WAY that knowing the properties of a source allows the choice of a code which minimises transmission length, knowledge of a channel allows optimisation of the rate at which information can be transmitted. This is incredibly important in communications networks such as the Internet and is known as *channel coding*.

Central to Shannon's paper was the idea that – when it comes to speed and reliability – the way in which a message is represented is as important as the physical means through which it is transmitted. Signals transmitted through any communications channel are in general subject to some degree of perturbation that arises (amongst other causes) from the ability of the channel to temporarily store energy.[6] At a time when the only known solution to overcoming this *noise* was to transmit at higher power or, failing that, to send information repeatedly at a reduced rate, Shannon showed that sufficient reliability could instead be achieved solely by the choice of an appropriate message representation or *code*. In fact, Shannon proved that every noisy channel has an intrinsic *capacity* (a limiting rate of information transfer) up to which a properly coded message can be transmitted entirely error-free.[7]

The mathematical proof of this statement is not simple, but the result can be understood in terms of entropies. Suppose we represent the source by a random variable x and the received signal by the random variable y. The probability distribution for the received symbols (after being affected by noise) results in 'incoming' entropy $H(y)$ given by an expression similar to that in previous section; this can be measured as information in bits received per symbol, or, by multiplying by the the average number of symbols transmitted per second, as the rate information is received. Since the channel is noisy, though, y will be some function of x as well as of the random noise on the channel; this means $H(y)$ includes some level of uncertainty which is *not* useful information. The noise will have perturbed the signal so that not all of the transmitted information reaches the receiver, so clearly the rate of useful information transfer is less than $H(y)$; how can we quantify the actual rate?

The solution is to define the *conditional entropy* of y given x; that is, if we did know x but not y, what is the average amount of information to be gained by learning y. This is denoted $H_x(y)$, and corresponds to the amount of uninformative uncertainty added by the noise. If the probability of the i^{th} symbol, when chosen by the source, being perturbed to appear at the receiver as the j^{th} symbol is $p_i(j)$, then this must be given by:

$$H_x(y) = -\sum_{i=1}^{n} p_i \sum_{j=1}^{n} p_i(j) \log_2 p_i(j)$$

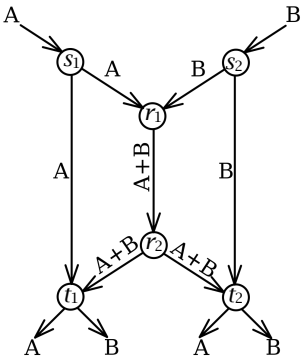
based on our previous expression and weighting according to the probability of the source choosing symbol i. It follows then that the rate of information transfer given noise is $H(y) - H_x(y)$. Shannon's result says that for a channel with capacity C it is *possible* to transmit information error-free at a rate $H(y) - H_x(y) \leq C$, and hence choosing a coding with $H(y)$ too small doesn't fully make use of available resources, while choosing one with $H(y)$ too large must cause an increase in the uncertainty due to noise, $H_x(y)$, with errors lowering the actual transmission rate back to C. The optimal channel code, then, is one which transmits information at or just below the Shannon capacity limit (see box, above).

The publication of Shannon's paper sparked an international search for these optimum coding techniques, and recent decades have seen some groundbreaking discoveries which allow practical transmission at rates fast approaching the capacity limit. One such example is the family of 'turbo codes' which have made 3G (and now 4G) high-speed mobile data protocols possible. In fact, recent developments have seen useable codes which reach within fractions of a decibel of the theoretical limit.[8] No doubt channel coding has been instrumental in developing the Internet as we know it today, but as we get so close to fundamental limits on capacity, it's worth asking: where do we turn next to improve our online communications?

A New Approach

USERS NOW EXPECT the ability to stream high quality audio to their mobile devices during daily commute, and watch high definition video on demand without having to endure any noticeable buffering. Rising expectations on connection speed require continual improvement in the way we use the Internet, but with traditional coding techniques fast approaching their potential (and physical updates to the well-established infrastructure being unimaginably costly), it looks like a roadblock is nearing.

One solution to this might be provided by a new development in information theory known as *network coding*. Where traditional coding techniques optimise the reliability and speed at which information can be transmitted through each channel in a network, network coding aims to make optimal use of the available capacity of *all* channels on the network. This difference was something that Shannon never needed to consider – telephone and telegraph lines merely connect together two end nodes – but can allow for drastic improvements in networks where each node is connected to many others.



DIAGRAM[10]
Simple 'butterfly' network. Network coding doubles the overall throughput compared to traditional packet switching.

This may sound similar to the intent of packet switching as mentioned in the introduction – it is – but network coding realises Baran's vision more successfully and on a lower level. Packet switching – which was standardised in 1982 as the Internet Protocol Suite (TCP/IP) and has experienced only relatively minor changes since[9] – is centred on the idea that splitting messages up into small packets and sending them via different routes through the network will maximise the use of available capacity. This is certainly true when packets are thought of as physical objects moving through some sort of narrow piping (for instance, simultaneously dropping three marbles down three separate tubes will allow them all to reach the bottom before three marbles which are dropped one after the other down the same tube), but this is an unnecessary limitation. Network coding takes advantage of the vector-like properties of message packets by simultaneously sending different linear combinations of them through different network channels, leaving the receiving nodes (known as *sinks*) with a system of linear equations to solve for the individual packets it desires.

This concept is perhaps best illustrated with an example; consider the simple 'butterfly' network shown to the left. Nodes s_1 and s_2 are two information sources, each generating at a rate of 1 bit/s; r_1 and r_2 are routers and t_1 and t_2 are sinks. Suppose s_1 is sending a message A to t_2 and s_2 is sending a message B to t_1, and that all channels have capacity 1 bit/s. Under a packet switching system, the routers simply repeat forward any incoming messages, queuing them if necessary; in this case the messages have to be transmitted one at a time through the middle channel, which for n-bit messages would take $2n$ seconds. If network coding is used, r_1 can calculate the value of $A+B$ and forward that instead; this can be transmitted in n seconds only, and the sinks can subtract the value of A or B simultaneously received from the side channel to calculate the message they desire. Effectively, network coding has squashed multiple packets into a congested channel (in a way which obscures their meaning) and utilised underused channels to send instructions which allow the original packets to be recovered: the information has been 'spread' around the network.

In the case of a larger network – where many message packets are transmitted through a decentralised arrangement of nodes – network coding can be considered in terms of linear algebra. Each transmitted message $\mathbf{M}_i$ can be expressed as a vector over the field F of used symbols; say the symbols used are two-bit binary values {00,01,10,11} then a message 11011011 would be written $(11,01,10,11)^{\mathrm{T}}$. Each router on the network constructs a linear combination $\mathbf{X}$ of the n messages it receives and forwards this:

$$\mathbf{X} = \sum_{i=1}^{n} a_i \mathbf{M}_i$$

where the a_i are all members of F (since F is a finite field, closed addition is carried out by the bitwise XOR operation; scalar multiplication is carried out by a slightly more complex but equally closed operation).[11] Each router r_k will generally receive a different set of messages and use a different set of coefficients, hence each *information vector* $\mathbf{X}_k$ forwarded should be different.

Through different paths in a decentralised or distributed network, each sink will receive a selection of information vectors $\mathbf{X}_k$ along with their corresponding *encoding vectors* $\mathbf{a}_k = (a_1, \ldots, a_n)^{\mathrm{T}}$. The sink can insert the encoding vectors to the left of the information vectors to form rows of an augmented matrix, and perform Gaussian elimination to reduce the matrix to row echelon form. Once n linearly independent rows have been received, reduced row echelon form can be calculated and hence each original $\mathbf{M}_i$ is given by the right hand side of the row beginning with $\mathbf{e}_i^{\mathrm{T}} = (0,\ldots,0,1,0,\ldots,0)$ with the 1 in the i^{th} position. This calculation need only be carried out by the sinks – routers within the network simply calculate linear combinations of incoming messages and forward them along with encoding vectors.[12]

Looking Forward

IN PRACTISE, network coding as described above can be employed to increase the throughput of large, decentralised networks without the need for additional physical channels. Modern networks are becoming increasingly distributed, and the introduction of network coding could allow us to make the best use of them. Even in cases such as the Internet where the architecture remains somewhat tiered, using network coding between nodes of the same tier can only offer improvement: where the network architecture is restrictive, network coding will simply imitate a packet switched protocol.[13]

Though we are unlikely to see network coding replace packet switching as the dominant Internet protocol anytime soon, it may make an introduction in *multicasting* applications – the transmission of the same information from a small number of sources to a relatively large number of sinks. This could be used, for instance, to offer better reliability in the online broadcast of live sports or other events, or for the release of popular software. Network coding is well suited to this because of the way it often allows all packets to arrive simultaneously (in the butterfly network example, both sinks were able to calculate A and B at the same time, even if they only required one). Note that this mixing of messages is not necessarily a security issue: individual messages may still need decryption once they are separated.

Finally, network coding offers other benefits beyond those provided by many traditional network protocols, such as coping well with dropped packets: messages are transmitted through the network in various linear combinations but each sink only requires one linearly independent set to decode all of them. Even if nothing of the network structure is known, enough linearly independent packets can be successfully generated by allowing routers to select coefficients at random, provided the symbol field is reasonably large.[14] As a result, network coding can be used to operate networks that are constantly changing or with no fixed infrastructure.

Though this may sound unnecessary for the Internet today, ad-hoc 'mesh' networks between mobile devices are starting to become popular and could benefit significantly from this robustness.[15] Currently, mesh networks are enabling instant communication in situations where the Internet is inaccessible – after national disasters, or during political protests – but they could soon play a more regular role, such as at sports games or music festivals where external connection if poor, as well as in military operations where the ability to establish an efficient but temporary communications infrastructure would be beneficial. As more and more devices in our lives gain Internet connectivity, perhaps the time for a new networking paradigm is drawing near.

References

[1] Lyon, B. The OPTE Project, http://www.opte.org/about/.

[2] Baran, P. *On Distributed Communications Networks*, RAND, 1964, p. 4.

[3,6] Hartley, RVL. *Transmission of Information*, Bell System Technical Journal, July 1928, pp. 536-540, 535.

[4,5,7] Shannon, CE. *A Mathematical Theory of Communication*, Bell System Technical Journal, July 1948, pp. 388-389, 404-405, 400–403.

[8] Guizzo, E. *Closing in on the perfect code*, IEEE Spectrum, March 2004.

[9] Fairhurst, G. http://www.erg.abdn.ac.uk/~gorry/course/inet-pages/tcp.html .

[10] Diagram modified from http://commons.wikimedia.org/wiki/File:Butterfly_network.svg .

[11,12] Fragouli, Le Boudec and Widmer. *Network Coding: An Instant Primer*, 2005, http://infoscience.epfl.ch/record/58339/files/rt.pdf, pp. 3, 1.

[13,14] Effros, Koetter and Médard, *Breaking Network Logjams*, Scientific American, June 2007, p. 83.

[15] Effros, Goldsmith and Médard, *The Rise and Fall of Instant Wireless Networks*, Scientific American, April 2010, pp. 74-74.