NETWORK CODING AND ITS APPLICATION TO CONTENT CENTRIC NETWORKING

Shigeki Miyake[1] *and Hitoshi Asaeda*[2]

[1]Network Innovation Laboratories, NTT,
1-1 Hikarinooka Yokosuka-shi, Kanagawa 239-0847, JAPAN, miyake.shigeki@lab.ntt.co.jp
[2] Network Research Headquarters, NICT,
4-2-1 Nukui-Kitamachi, Koganei-shi, Tokyo 184-8795, JAPAN, asaeda@nict.go.jp

ABSTRACT

Content Centric Networking (CCN) is one of the predominant proposals that have been made for the next generation content distribution platform. In CCN each router in a network is equipped with a large sized cache memory and becomes a content router that can be regarded as a temporary content server in the network. This mechanism can reduce the concentration of data traffic going into the original content server.

To enhance the merits of CCN, we propose applying network coding (NWC) to it. Applying NWC in this way makes it possible to implement distributed data storing of content data and to achieve multipath transmission between content routers and an access router (or an end user). CCN with NWC can provide higher throughput for end users and reduce network load. In addition to proposing the aforementioned application, in this paper we specify issues needing to be addressed to more successfully combine CCN with NWC.

1. INTRODUCTION

In research done by Cisco [3] it is predicted that the ratio of content distribution traffic to total Internet traffic will grow from 34 % in 2012 to 51 % in 2017. Since this large growth may cause overload of current content distribution platforms such as content delivery network (CDN) and peer-to peer (P2P) network, researches on the next generation content distribution platform are now in progress [1]. In these researches, Content Centric Networking (CCN) is one of the predominant proposals that have been made [2].

In CCN each router in a network is equipped with a large sized cache memory and can be regarded as a temporary content server in the network. Since this mechanism appears to provide many content servers and since end user requests for the content can be distributed among these "temporary" content servers, the original content server can avoid traffic concentration. On the other hand, since end users wishing to access particular content in the network do not know the URL of the appropriate server that stores it, they use the name of the content itself to locate it [4]. It can be said that CCN is a content distribution system to improve response from the content server and reduce network load by making each router into a content router that has a large sized cache memory and can pretend to be a content server temporarily.

However, current CCN schemes do not appear to fully utilize the network performance. Cached data may be lost by cache memory updates or storage disk crashes. When the cached data of the desired content is lost from the content router's cache memory, the requests for the content may be concentrated in the origin server. To maintain robustness against cached data loss, the data can be fragmented so that it can be stored in a distributed way in the network.

Although multipath transmission has recently been studied [5] as a means for increasing throughput, CCN usually uses single-path transmission between the end user and the origin server or the content router. Multipath transmission enables network performance to be used more effectively. Network coding can bring both distributed data storing and multipath transmission to CCN.

Network coding (NWC) was devised by Ahlswede et al. in 2000 [6]. They showed that the maximum transmission rate for multicasting on a given network can be obtained by the maxflow bound of the network, and that NWC can attain the maxflow bound. Li et al. [7] demonstrated that a linear network code constructed by a linear code can also attain the maxflow bound. For constructing the optimal linear network code, Jaggi et al. [8] proposed a deterministic construction algorithm, and Ho et al. [9] demonstrated that a randomly constructed linear network code can also attain the maxflow bound. Miyake et al. [10] showed that using low-density-parity-check (LDPC) matrices to construct a linear network code makes it possible to apply efficient algorithms such as the sum-product algorithm to the decoding process. An NWC application that can be easily implemented is distributed data storing, which is also called diversity coding [14].

NWC can implement both distributed data storing and multipath transmission in CCN. Montpetit et al. [12] proposed an application of NWC to CCN, which concerns an efficient method for collecting data fragments that are stored in a distributed way among several nodes in the network. However, the proposal did not clarify what procedure should be carried out to implement distributed data storing in CCN, nor the advantages to using NWC in data transmission, nor the problems needing to be solved to apply NWC to CCN on real networks.

This paper describes some of the advantages to applying NWC to CCN as well as the problems involved in doing so. Outlines of CCN and NWC are presented in Section 2 and Section 3, respectively. Section 4 details the content distribution procedures of CCN with NWC and clarifies the advantages to and the problems involved in implementing CCN with NWC. Section 5 concludes the paper with a summary of key points and a mention of future work.

2. CONTENT CENTRIC NETWORKING

Two content distribution schemes, content delivery network (CDN) and peer-to-peer (P2P) network, have mainly been used up to now. The main problem with CDN is that accesses are concentrated in the master file server and backbone network load will grow due to increased content distribution traffic in the near future. On the other hand, with P2P there is a difficult problem in constructing efficient searching schemes for the vast amount of content in the network. Consequently, it has become increasingly important to develop efficient content distribution platform to reduce network load and provide end users with necessary throughput in light of the rapidly growing content distribution traffic.

Content Centric Networking (CCN) is one of the predominant proposals that have been made regarding the next generation content distribution platform. Its special features are as follows.

- Each router in the network is equipped with a large sized cache memory and becomes a content router. Since content routers distributed over the network can be regarded as temporary content servers, the original content server can avoid traffic concentration.

- Since end users do not know which content router is appropriate for their request, they cannot use a URL to find the location of the desired content. With CCN they can instead use the name of the requested content itself and the network uses the name to find the location.

The content distribution procedure in CCN is as follows.

Step 1. The end user sends a request (called an “interest”) for the desired content to the network (Figure 1, left).

Step 2. The content router receiving the interest packet checks whether the requested data is stored in its cache memory, which is called a “content store”. If it is, it is sent back to the end user through the path on which the interest packet was transmitted (Figure 2). Otherwise, the interest packet is forwarded to the next content router (Figure 1, left).

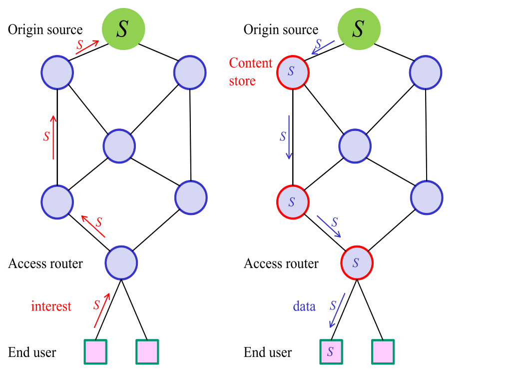

Figure 1. Data distribution procedure in CCN (1).

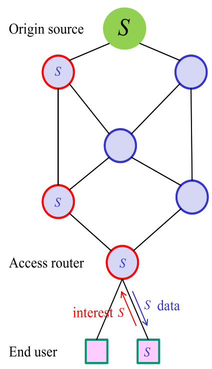

Figure 2. Data distribution procedure in CCN (2).

Step 3. The content router that receives the interest packet repeats Step 2. If no content routers store the requested data and the interest packet reaches the origin server, the origin server sends the requested data back to the end user through the path on which the interest packet was transmitted. Content routers on the path to the end user cache the data in their content store (Figure 1, right).

See [4] for the details of the procedure. By regarding content routers distributed over the network as temporary content servers, it can be expected not only that traffic concentration in any specific server will be avoided but also that the network load will be reduced and user response will be improved. On the other hand, since cached data is stored temporarily it can be lost easily, and requests for specific content may be concentrated in the origin server. When using a cache mechanism it is necessary to consider both its merits and demerits. Using a distributed data storing scheme effectively enhances the robustness of cached data against disk crashes and data evictions.

By fragmenting data and storing it in distributed storages, when data fragments (called "chunks") are lost due to storage disk crashes or data evictions, new chunks can be regenerated by collecting residual chunks over the network. Although there are several ways to implement distributed data storing, the use of NWC (introduced in the next section) makes it possible to achieve both efficient data transmission and distributed data storing at the same time.

3. NETWORK CODING

3.1. Problem Setting and Fundamental Theorem

The network to be considered is given as a directed acyclic graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote node set and link set, respectively. Rate constraint R_{ij} is assigned at each link $(i, j) \in \mathcal{E}$. The network coding problem treats multicast transmission of a message from the origin node $S \in \mathcal{V}$ to the set of terminal nodes $\mathcal{T} \subset \mathcal{V}$ (Figure 3).

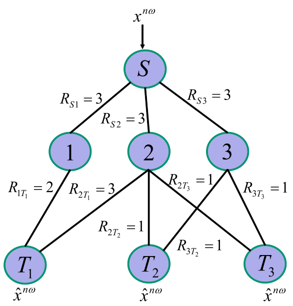

Figure 3. An example of a directed acyclic graph with rate constraints.

Suppose that the alphabet of the message and codewords is a finite field $\mathbf{F}_q = \{0, 1, \cdots, q-1\}$. The message sent from the origin node is represented as an $n\omega$-dimensional row vector $x^{n\omega} \in \mathbf{F}_q^{n\omega}$, and each link is used n times for the transmission of the message, where ω is defined as the transmission rate.

The network code consists of encoder f_{ij}, which is assigned at each link $(i, j) \in \mathcal{E}$, and decoder g_T, which is assigned at each terminal node $T \in \mathcal{T}$, and is defined as mappings

$$f_{ij} : \mathbf{F}_q^{k_i} \to \mathbf{F}_q^{k_{ij}}, \tag{1}$$

and

$$g_T : \mathbf{F}_q^{k_T} \to \mathbf{F}_q^{n\omega}, \tag{2}$$

respectively. The rate constraint at each node $(i, j) \in \mathcal{E}$ can be represented as

$$\frac{k_{ij}}{n} \leq R_{ij}, \tag{3}$$

where

$$k_i \stackrel{\text{def}}{=} \sum_{l:(l,i)\in\mathcal{E}} k_{li} \quad \text{for } i \neq S,$$

$$k_S \stackrel{\text{def}}{=} n\omega \quad \text{for } i = S.$$

Let $\boldsymbol{z}_i$ be the incoming information of a node i. If for a terminal node $T \in \mathcal{T}$ it holds that $x^{n\omega} \neq g_T(\boldsymbol{z}_T)$, then we can say that a decoding error occurs. The multicast problem can be stated as follows.

Multicast Problem:
Suppose that a message $x^{n\omega}$ is multicast from the origin node S to the terminal node set $\mathcal{T}$. Then for what transmission rate ω does there exist the network code

$$\left\{(f_{ij}, g_T) \,\middle|\, (i, j) \in \mathcal{E}, T \in \mathcal{T}\right\}$$

that can attain arbitrarily small decoding error?

The next theorem is fundamental.

Theorem 1 (Maxflow Bound [6]) *The maximum value of the transmission rate ω is given by*

$$\min_{T\in\mathcal{T}} \text{maxflow}(S, T),$$

where maxflow(S, T) *denotes the maxflow between the origin node S and the terminal node T.*

In Figure 3, for example, since it holds that maxflow$(S, T_1) = 5$, maxflow$(S, T_2) = 2$, and maxflow$(S, T_3) = 2$, the maxflow bound of the network becomes

$$\min_{T\in\mathcal{T}} \text{maxflow}(S, T) = 2.$$

3.2. Encoding and Decoding

The network code whose transmission rate attains the maxflow bound can be constructed by linear code [7]. Furthermore, if n or q is sufficiently large, elements of matrices of the linear network code can be taken at random [9]. In the following construction of the code, all matrices are supposed to be constructed by taking their elements at random.

[Construction of Encoder]
For $(i, j) \in E$,

$$f_{ij}(\boldsymbol{z}_i) = \boldsymbol{z}_i \boldsymbol{A}_{ij}, \tag{4}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{A}_{ij}$ denote k_i-dimensional row vector and $k_i \times k_{ij}$ matrix, respectively.

[Construction of Decoder]
Let $\boldsymbol{z}_T$ be the incoming information for the terminal node T. Then the decoder at the terminal node T can be given as

$$g_T(\boldsymbol{z}_T) = \Big\{\text{Solving the linear equation } \boldsymbol{z}_T = x^{n\omega}\tilde{\boldsymbol{A}}_T \text{ by sweep-out method.}\Big\}, \tag{5}$$

where $\tilde{\boldsymbol{A}}_T$ denotes an aggregation of matrices on the paths between the origin node S and the terminal node T. When the message from the origin source is $x^{n\omega}$, the information vector $\boldsymbol{z}_T$ incoming to the terminal node T can be represented by

$$\boldsymbol{z}_T = x^{n\omega}\tilde{\boldsymbol{A}}_T.$$

$\tilde{\boldsymbol{A}}_T$ is called the global encoding kernel [14].

For example, for the network in Figure 4 $\tilde{\boldsymbol{A}}_{T_1}$ is a $n\omega \times k_{T_1}$ matrix and represented by

$$\tilde{\boldsymbol{A}}_{T_1} = [\boldsymbol{A}_{S1}\boldsymbol{A}_{1T_1}, \boldsymbol{A}_{S2}\boldsymbol{A}_{2T_1}].$$

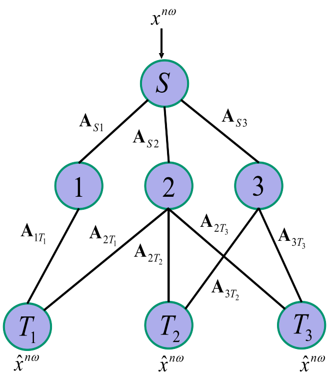

Figure 4. Assignment of code matrices in NWC.

3.3. Diversity Coding

Using NWC enables distributed data storing to be easily executed. Figure 5 represents the model of distributed

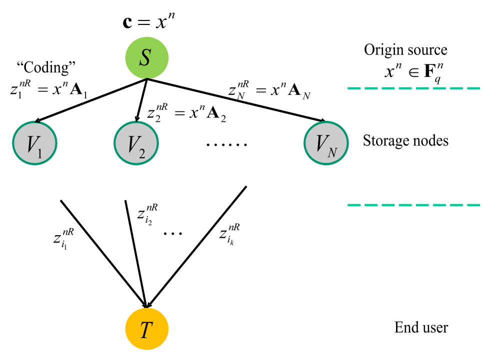

Figure 5. Distributed data storing using network coding.

data storing using linear NWC. N storage disks V_i $(i = 1, .., N)$ are connected to the origin source S. In each storage disk V_i, data fragment (chunk) of the message, z_i^{nR}, is stored, where R is a coding rate and represents distribution degree (see (7)). Users want to reproduce the message $x^n \in \mathbf{F}_q^n$ by accessing any k storage disks. The obtained information consists of k chunks $z_{i_1}^{nR}, .., z_{i_k}^{nR}$, and the reproduction of the message is carried out by solving linear equation

$$(z_{i_1}^{nR}, .., z_{i_k}^{nR}) = x^n \left(\boldsymbol{A}_{i_1}, .., \boldsymbol{A}_{i_k}\right), \quad (6)$$

which corresponds to the decoding process of NWC described in (5). In Theorem 1, by setting $\omega = 1$ and maxflow bound$= k$ we can see if

$$kR > 1 \quad (7)$$

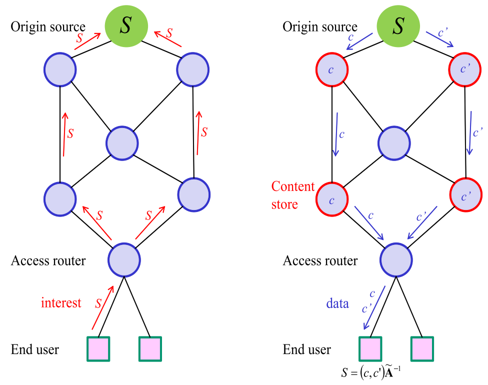

Figure 6. Data distribution procedure in CCN with NWC (1).

holds and n is sufficiently large, Equation (6) can be solved using the sweep-out method. The distributed data storing scheme mentioned here is called diversity coding (DC) [14].

A conventional distributed data storing scheme is constructed using Reed-Solomon code. Diversity coding is advantageous for a network in which a transmission using NWC is carried out. Since at each storage node the codeword of DC is the same as that of NWC, no extra processing other than NWC is needed. When chunks are lost in some of the N storage nodes, using chunks in the residual k nodes enables new chunks to be regenerated [11]. Furthermore, when each $\boldsymbol{A}_i$ is constructed by an LDPC matrix, since the sum-product algorithm can be applied to the decoding process instead of the sweep-out method, more efficient decoding can be achieved [10].

4. CCN WITH NWC

Using CCN makes it possible to deploy multiple content routers that function as temporary content servers by caching content data at each content router between the origin source and end users, and to perform efficient transmission by avoiding concentration of the traffic in any specific server. However, there is a risk that stored content data will be lost when a cached data eviction or storage disk crash occur. In these cases, requests for content may concentrate in the origin server and the network load will increase. By storing data in a distributed way over the network, it can be expected that the content distribution network will become more robust against cached data evictions or storage disk crashes. Moreover, adopting distributed data storing, allows multipath transmission to be implemented since an end user obtains chunk data from multiple content routers. This means effective throughput for end users is improved.

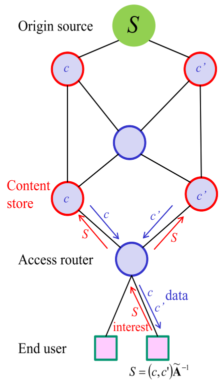

Figure 7. Data distribution procedure in CCN with NWC (2).

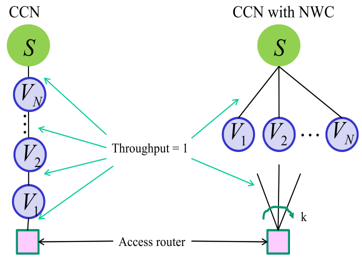

Figure 8. Simple models of CCN and CCN with NWC.

4.1. Proposed Scheme

The content distribution procedure of CCN with NWC is as follows.

Step 1. The end user sends an interest for the desired content to the network through multiple paths (Figure 6, left).

Step 2. The content router receiving the interest packet checks whether a chunk of the requested content is cached in its content store. If it is, it is sent back to the end user through the path on which the interest packet was transmitted (Figure 7). Otherwise, the interest packet is forwarded to the next content router (Figure 6, left).

Step 3. The content router that receives the interest packet repeats Step 2. If no content routers on the path store the chunk of the requested content and the interest packet reaches the origin server, the origin server sends the chunks back to the end user through the paths on which the interest packets were transmitted. Note that at each node after incoming data is network coded, it is forwarded to the next node and at the same time cached as a chunk in the content store of the node (Figure 6, right).

It should be noted that in the above procedure, the transmission using NWC enables distributed data to be stored naturally.

Advantages over CCN are:

1. By adopting distributed data storing, the content distribution scheme becomes robust against cached data evictions or storage disk crashes.

2. When a cache miss occurs, network load can be reduced by reducing download data.

3. Multipath transmission between content routers and the access router provides higher effective throughput to the end user (Figure 7).

4.2. Analysis of Toy Model

In this subsection we show that network load can be reduced by combining distributed data storing and a chunk regeneration scheme. This corresponds to the second of the advantages of CCN with NWC mentioned in the previous subsection.

The left of Figure 8 shows a simple CCN model that has N content routers in series, and the right shows a simple CCN with NWC model that has N content routers in parallel and makes a layer of content routers for the regeneration of chunks. Throughput of each link assumed to be unit. In the CCN with NWC model, after the access router obtains k chunks from any k content routers, desired content can be reconstructed from the obtained chunks. Suppose that the CCN with NWC model adopts a chunk regeneration scheme [11] under which, for the loss of less than $N - k + 1$ chunk data, using residual chunk data in the network enables new chunk data to be regenerated without downloading data from the origin server. Let p be the hit probability of cached data in each content router, λ be the rate of interest generation at the end user or access router side, and D be the content size. Note that the size of a stored chunk becomes D/k. When a data with size D is transmitted through j links, then the network load is defined by $D \times j$.

Under the above premise, the expected network load of the CCN model can be described by

$$\begin{aligned}&\lambda D\left\{p+(1-p)p2+\cdots(1-p)^{N-1}pN\right.\\&\quad\left.+(1-p)^{N}(N+1)\right\}\\&=\lambda D\left\{\frac{1-(1-p)^{N}(Np+1)}{p}+(N+1)(1-p)^{N}\right\}\end{aligned} \tag{8}$$

by considering that data is downloaded from the node at which cache hit occurs for the first time from the access router. On the other hand, in the CCN with NWC model, data is downloaded from the origin source only when l cache misses occur at the same time where $l \geq N - k +$

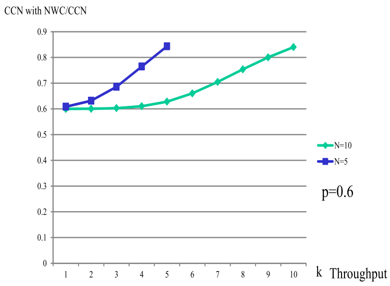

Figure 9. Network load and throughput.

1. In this case the download data size is $\frac{l-N+k}{k} \times D$. Note that in the case of $l \leq N-k$, since the regeneration scheme [11] is applied, new chunks can be regenerated at the layer of content routers without downloading chunks from the origin server. Then the expected network load of CCN with NWC can be described by

$$\lambda D \left\{ 1 + \sum_{l \geq N-k+1} \binom{N}{l} p^{N-l} (1-p)^l \frac{l-N+k}{k} \right\}. \tag{9}$$

By considering the ratio $(9)/(8)$, the network load ratio becomes

$$\frac{1 + \sum_{l \geq N-k+1} \binom{N}{l} p^{N-l} (1-p)^l \frac{l-N+k}{k}}{\frac{1-(1-p)^N (Np+1)}{p} + (N+1)(1-p)^N}. \tag{10}$$

Figure 9 plots the network load ratio for the case of $N = 5$ and $N = 10$, respectively, where the cache hit probability is set to $p = 0.6$. Since k can be regarded as the number of paths between content routers and access router, the horizontal axes of Figure 9 can be interpreted to show effective throughput between them. It can be said that the network load of CCN with NWC model becomes smaller than that of CCN model over all range of k, although when the effective throughput becomes high, the network load in the CCN with NWC model tends to increase.

4.3. Issues for Consideration

While CCN with NWC has the advantages mentioned in Subsection 4.1, the application of NWC produces some problems. Subsection 4.1 showed the content distribution procedure of CCN with NWC, where the parameters for distributed data storing (N, k) were determined in advance and it was assumed that all routers were content routers. Although these assumptions are unlikely to cause any problems in small or simple networks, the following issues will need to be addressed to implement CCN with NWC in larger or more complex networks.

1. How do we deploy content routers?
   Enlarging the memory of all routers would require considerable investment [13]. However, we want to maximize the advantages shown in Subsection 4.1 with as little cost as possible.

2. How do we determine the parameters (N, k) for distributed data storing for the given network?
   Whether interest and data traffic becomes large or modest will depend on the deployment of content routers and parameters (N, k) for the distributed data storing.

The following subjects should also be considered in moving forward with research on CCN with NWC.

3. Thorough analysis of network load for more complex networks:
   We will need to show the advantages of CCN with NWC over CCN for more general cases than that given in Subsection 4.2.

4. An index for the efficient deployment of content routers:
   It seems logical to assume that when the number of content routers increases, the network load will decrease and the number of end users who can use the network at the same time will increase. Can this number of end users become an index for the efficiency of content router deployment?

5. CONCLUSION

Content Centric Networking (CCN) is one of the predominant proposals that have been made for the next generation content distribution platform. We have shown that the merits of CCN such as enhanced throughput and reduced network load can be enhanced by applying network coding (NWC). We also showed that some issues remain to be addressed in implementing CCN with NWC, among them determining how to effectively deploy content routers and appropriately setting distributed data storing parameters.

In the near future, we plan to implement CCN with NWC on an experimental network and continue researching to verify the merits of this combination and implementing in the real world.

6. REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *IEEE Communications Magazine*, **vol. 50**, no. 7, pp. 26–36, 2012.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proc. 5th ACM International Conference on Emerging Networking Experiments and Technologies*, pp. 1–12, 2009.

[3] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017.

[4] "Named data networking (NDN) project," *Technical Report NDN-001*, 2010.

[5] G. Rossini and D. Rossi, "Evaluating CCN multipath interest forwarding strategies," *Computer Communications*, **vol. 36**, no. 7,pp. 771–778, 2013.

[6] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, **vol. 46**, no. 4, pp. 1204–1216, 2000.

[7] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, **vol. 49**, no. 2, pp. 371–381, 2003.

[8] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inform. Theory*, **vol. 51**, no. 6, pp. 1973–1982, 2005.

[9] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, Jun Shi, and Ben Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, **vol. 52**, no. 10, pp. 4413–4430, 2006.

[10] S. Miyake and J. Muramatsu, "On a construction of universal network code using LDPC matrices," in *Proc. ISIT 2012*, pp. 1301–1305, 2012.

[11] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory* ,**vol. 56**, no. 9, pp. 4539–4551, 2010.

[12] M. J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," in *Proc. the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design– Architecture, Algorithms, and Applications (NoM'12)*, pp. 31-36, 2012.

[13] D. Perino, and M. Varvello, "A reality check for content centric networking," in *Proc. the ACM SIGCOMM workshop on Information-centric networking(ICN'11)*, pp. 44-49, 2011.

[14] R. W. Yeung, *Information Theory and Network Coding*, Springer, New York, 2008.