AN OVERVIEW OF NETWORK CODING FOR MULTIMEDIA STREAMING

Enrico Magli

Politecnico di Torino (Italy)
Dipartimento di Elettronica
enrico.magli@polito.it

Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory - LTS4
pascal.frossard@epfl.ch

ABSTRACT

The objective of this paper is to survey recent developments of network coding, with specific focus on multimedia streaming. Network coding allows nodes to create and forward "combinations" of incoming messages, which has been shown to increase throughput. While network coding has been invented in the information theory field, its potential benefits are spurring new research on its multimedia applications. We first review the concept of network coding, and briefly describe its potential benefits from a multimedia communication perspective. Then, we discuss the specific issues imposed by media delay constraints on network coding algorithms. Finally, we review some recent works that develop network coding principles in media streaming applications.

1. INTRODUCTION

Recently, a lot of attention has been devoted to applications such as peer-to-peer (P2P) and wireless mesh networks, in which peer nodes can self-organize in order to exploit more efficiently the network infrastructure. A significant advantage of these networks lies in the multiple paths and the multiple forwarding peers between servers and clients. The network diversity can be used to enhance the quality of service in video communication systems with increased bandwidth throughput or improved resilience to packet loss. However, this requires the development of appropriate streaming mechanisms, so that the network resources are exploited and redundancy avoided.

In many cases, it has been found that optimal communication of information over networks requires the intermediate network nodes to perform coding operations. In particular, "network coding" (NC) [1] is a new paradigm that significantly innovates the prevalent model in which the role of intermediate network nodes is only to forward the incoming messages towards the appropriate destination. NC brings the main novelty of allowing "processing" of messages at each hop in the network. Each intermediate node is allowed to mix incoming messages and then to forward the combined packets towards the destination nodes. The encoding ensures that any destination node can receive with high probability enough combinations to recover the original messages. NC has been shown to generally improve throughput, achieving network capacity for single-source multicasting.

While most NC research has been carried out in the field of information theory, its potential benefits for media streaming applications have spurred a lot of interest in the multimedia community. NC multimedia applications however require to overcome the simplistic network models employed in information theory, and to deal with the requirements specific to multimedia streaming, most notably the delay constraints. The objective of this paper is to review the basic concepts of network coding, and the design constraints imposed by practical streaming scenarios. We finally describe recent NC-based solutions in multimedia streaming applications.

2. BASICS OF NETWORK CODING

2.1. Two toy examples

Network coding benefits can be exemplified through the two scenarios depicted in Fig. 1. The first one (Fig. 1-a) considers a multicast setting in which two sources $S1$ and $S2$ want to transmit packets a and b, containing binary information symbols, to receivers X and Y. The intermediate node R does not simply relay a and b. Rather, it creates a new packet $a \oplus b$, where $\oplus$ denotes bitwise exclusive OR, and forwards it to X and Y. As a consequence, using only once the outgoing link of R, X can recover b as $b = a \oplus (a \oplus b)$, and Y can similarly recover a. In a non-coded network, R should have made two individual transmissions of a and b. Therefore, as can be seen, coding in the network allows to increase the throughput (as the saved transmission slot can be employed to communicate a new packet), reduce the delay (because there is no need to wait for two transmissions

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreements 216715 (Newcom++) and 214063 (SEA - seamless content delivery).

from R) and the energy consumption (thanks again to the saved transmission slot). In the second example (Fig. 1-b), wireless nodes X and Y are in the range of a base station S, but cannot communicate directly; they want to exchange information packets a and b. First, X sends a to S, and then Y sends b to S. The base station creates a new packet $a \oplus b$ and broadcasts it to X and Y. As in the previous example, coding allows to save a transmission slot by sending only the "difference" information with respect to what already is in the buffer of each node. These two toy examples illustrate the potential of coding operations in the network nodes.

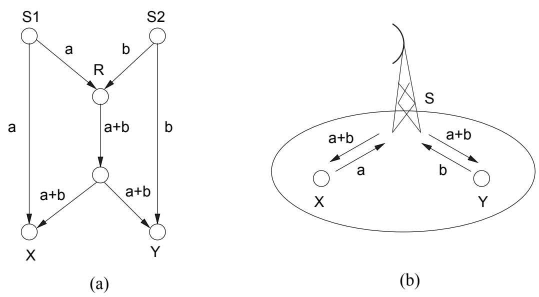

Fig. 1. Examples of network coding: (a) multicast with two sources and two receivers; (b) wireless point-to-point communication.

2.2. The network coding concept

We now formalize NC by looking at the operation of a given node X of the network. We assume that X has M incoming and Q outgoing connections; for simplicity, we assume that each link can transport one packet per instant of the time scale and that nodes are able to perform linear coding operations. At a given instant, node X receives a set of M packets denoted as r_j, $j = 1, \ldots, M$, and combines them to produce Q output packets as $p_l = \sum_{j=1}^{M} \alpha_{j,l} r_j$, $l = 1, \ldots, Q$. This is a symbol-by-symbol linear combination of co-located symbols of all incoming packets using coefficients $\alpha_{j,l}$, with all operations performed on a finite field F of given size. That is, for every output link, a set of M coefficients are chosen and used to compute the symbol-by-symbol combinations. In general, the input packets are not the original information packets, but packet combinations themselves. Therefore, each packet typically contains information about *all* original packets, but is not sufficient to decode any of them. Received packets are of the form $r_j = \sum_{i=1}^{N} \beta_{i,j} p_i$. That is, each of the M received packets is a linear combination, over finite field F, of the set of N original source packets p_i. The final combination coefficients $\beta_{i,j}$ are assumed to be known, and depend on the various $\alpha_{j,l}$'s that have been used by each intermediate node of the network. This can be conveniently written in matrix notation as $r = Gp$, where $r = [r_1 \ldots r_M]^T$, $p = [p_1 \ldots p_N]^T$, and G is the matrix of coefficients $\beta_{i,j}$. The set of original packets can be recovered if G is invertible over F, i.e., if the received packets are linearly independent combinations of the original packets; this in particular requires that $M \geq N$.

Information-theoretic results have shown that NC can increase network throughput compared to traditional networks in both point-to-point and multicast scenarios. The linear NC scheme described above achieves network capacity for multicasting from a single source [1]. Linear coding is however not optimal in general communication settings, in particular when several sessions are transmitted jointly. While coefficients $\alpha_{j,l}$ can be chosen optimally when the network if fully characterized, they are in practice picked randomly from a uniform distribution over F. This has been shown [2, 3] to be asymptotically optimal as the finite field size becomes large. For more details, we refer the reader to the several excellent tutorials already published, e.g. [4].

Besides increasing throughput, NC also provides other advantages. Interestingly, routing in a NC setting turns out to be significantly simpler than in the general case, as it can be carried out in polynomial time. On a similar note, random linear NC algorithms tend to blindly "spread" the information across the network, so that the receiver only needs to make sure it receives a sufficient number of packets. In many applications, and notably P2P, this can be leveraged to design very effective information distribution systems where data management becomes easier. For similar reasons, NC also provides error resilience. The receiver is indeed not sensitive to loss of packets on a particular network segment, since it only needs to receive a given number of independent packets irrespectively of the sending node or transmission path. In other terms, global NC permits to exploit the diversity of the network, if multiple paths are available. Moreover, NC permits to relax the need for tight synchronization/coordination in packet delivery. There exist plenty of other studies related to several aspects of NC, including security, error correction and joint channel/network coding. At the time of this writing, the network coding bibliography, which can be found at `http://www.ifp.uiuc.edu/~koetter/NWC/`, lists 275 papers.

3. DESIGN CONSTRAINTS IN STREAMING APPLICATIONS

Despite the huge attention dedicated so far to NC, most papers consider ideal network settings. However, employing NC for multimedia streaming requires to adapt NC techniques to realistic networks, and to cope with streaming-specific requirements such as delay constraints.

3.1. Network coding and the OSI layers

While NC has been developed for ideal networks represented as directed graphs, practical networks employs protocols to manage communications. Thus, the question arises of where NC should take place in the OSI protocol stack. The answer to this question is not unique, but depends on what results is expected.

Most NC techniques over graphs, when applied in a realistic context, would be implemented as a shim layer between IP and MAC. The advantage of this approach is that, at each node, the data do not have to be sent all the way to the transport or application layer for decoding and re-encoding packets, avoiding the risk of introducing latency, which could outweigh the NC throughput benefits. However, this kind of applications are not compatible with the existing range of IP-based applications.

NC can also be applied above IP, and specifically in overlay networks, which is the main focus of this paper. E.g., in [5] a P2P protocol is proposed, in which each node creates packet combinations and forwards them to the appropriate peers. This can be seen as an application-layer NC stage. Given the large latency of typical P2P live streaming protocols, the additional latency introduced by NC is not an issue, but it might be for other applications. On the other hand, NC in overlay networks does maintain compatibility. Moreover, the availability of different network paths from a server to a peer, and the possibility of choosing different peers from whom to download the data, are a perfect match to NC. Indeed, as all packet combinations are equally informative, the receiver is not constrained to wait for a specific packet, but can receive useful information from any peer involved in a given multimedia session.

3.2. Bandwidth and delay issues

A first step towards practical NC schemes has been presented in [3]. We describe here bandwidth and timing issues in the NC-based delivery of continuous streams, although we do not explicitly address delay-constrained NC in this paper.

A packet format is proposed in [3], which allows each receiver to construct the matrix G necessary for decoding. In particular, the header of the j-th outgoing packet includes a *global encoding vector*, that is, the set of coefficients $\beta_{i,j}$, with $i = 1, \ldots, N$. This generates a rate overhead, as the size of the global encoding vector is N times the size of the finite field. However, the overhead can be kept small if the packet size is large enough. Each node receives a set of incoming packets with their global encoding vectors, and can easily compute the encoding vectors $\beta_{i,j}$ of the outgoing packets, by combining the received encoding vectors and the coefficients $\alpha_{j,l}$.

As real networks transmit packets asynchronously, and certain applications (e.g., streaming) continuously transmit new packets, the NC protocol has to identify which packets can be combined together. In [3] this is done by dividing the source packets into so-called *generations*, i.e., sets of packets that belong to the same NC session. Thus, each packet must also be tagged with the generation number, and each node may have to keep multiple buffers in order to separately process packets of different generations.

It should be noted that the NC operation of taking linear combinations of packets can be seen as a nonsystematic erasure correction code in which the parity symbols are not computed by the server in a centralized way, but rather by each node. This paves the way for the use of efficient erasure correction coding techniques in the NC scenario. The scheme in [3] also allows to differentiate the number of received packets required to decode different symbols of the packets, providing a form of unequal protection that can be coupled with layered source coding.

Finally, NC increases the overall latency of the streaming system. It further adds extra encoding and especially decoding complexity. While [5] shows that this is not an issue on a personal computer, this may not be the case of other applications with very tight constraints on computation power.

4. NETWORK CODING BASED STREAMING APPLICATIONS

NC principles have been used recently in multimedia streaming applications, targeting efficient resource allocation, increased throughput and resiliency to transmission errors. NC has been mostly applied in peer-to-peer or wireless broadcast scenarios, under different forms. Extensions have recently been proposed that account for the different packet importance in media streams.

In a first attempt of pushing channel coding into network nodes, Reed-Solomon codes or digital fountain codes have been implemented in network-embedded FEC nodes [6] and in network peers [7]. Both schemes show that the network throughput can be significantly improved with NC. However, in both cases, the packets are decoded and re-encoded in the network nodes, before transmission towards the streaming client. Decoding and re-coding operations in network nodes unfortunately augments the latency of the streaming system. A new streaming algorithm called R^2 [5] incorporates random NC along with a random pushing algorithm to smooth the latency problems, and enable live peer-to-peer streaming. Alternatively, the latency could also be reduced by avoiding the decoding in the network nodes. Packet re-encoding with rateless codes has been proposed in [8], which has been shown to be robust to erroneous channel estimations, especially at high loss rates and with limited network diversity.

NC has also been proposed as an efficient algorithm for multicasting multimedia streaming in overlay or peer-to-peer networks [9–11], where it can take advantage of path diversity. The benefit of NC in such scenarios has been analyzed in [5], which shows that NC is very useful in peer-to-peer networks, since it provides resiliency to network dynamics and leads to better bandwidth exploitation.

Research has been also conducted in the application of NC to wireless broadcast scenarios. Once losses happen in this context, one could rapidly face a feedback implosion effect if all the receivers expect to receive missing packets by retransmission. NC has been shown to efficiently combat the implosion problem [12].

One of the important characteristics of multimedia data lies in the unequal importance of the packets with respect to the quality of the decoded information. Typically, the multimedia information is organized hierarchically by the source coding algorithm, such that it is crucial for the decoder to receive at least the most important packets. This property has to be included in the NC algorithm, in order to ensure an efficient use of the bandwidth resources, and maximize the quality of the decoded stream. The significance of each packets can be considered in the selection of the packet to be encoded in the network, as proposed in [13] and [14] for effective video streaming over wireless mesh networks or broadcasting from a WLAN access points, respectively. The construction of the network codes could also be optimized in order to prioritize the delivery of the most important video frames when the bandwidth or transmission energy is constrained [15].

5. DISCUSSION AND CONCLUSIONS

NC is a new networking paradigm with many prospective streaming applications. However, there are several research problems that warrant further investigation. A representative sample is provided by the IEEE ICME 2009 special session on "network coding for multimedia streaming". The session contains three contributions addressing major issues in this area. [16] presents a system for wireless multi-party videoconferencing which employs a new multi-step network coding algorithm, as well as a dedicated unequal error protection scheme. [17] proposes a rate control method that exploits awareness of NC in wireless networks. Finally, [18] employs rateless codes to implement a NC system that provides reduced startup delay, as well as low complexity.

6. REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, 2000.

[2] S. Jaggi, P.A. Chou, and K. Jain, "Low complexity optimal algebraic multicast codes," in *Proc. of IEEE ISIT*, 2003.

[3] P.A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of 41st Allerton Conference*, 2003.

[4] P. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Processing Magazine*, pp. 77–85, Sept. 2007.

[5] M. Wang and B. Li, "R^2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.

[6] M. Wu, S. Karande, and H. Radha, "Network Embedded FEC for Optimum Throughput of Multicast Packet Video," *EURASIP Journal on Advances in Signal Processing*, vol. 20, no. 8, pp. 728–742, Sep. 2005.

[7] C. Wu and B. Li, "Optimal Peer Selection for Minimum-Delay Peer-to-Peer Streaming with Rateless Codes," in *Proc. of ACM P2PMMS*, Singapore, Nov. 2005, pp. 69–78.

[8] N. Thomos and P. Frossard, "Raptor Network Video Coding," in *Proceedings of the 1st ACM International Workshop on Mobile video (in conjunction with ACM Multimedia 2007)*, Augsburg, Germany, Sep. 2007.

[9] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, "LION: Layered Overlay Multicast With Network Coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1021–1032, Oct. 2006.

[10] Y. Liu, Y. Peng, W. Dou, and B. Guo, "Network Coding for Peer-to-Peer Live Media Streaming," in *Proc of the Fifth International Conference Grid and Cooperative Computing, GCC 2006*, Monticello, IL, USA, Oct. 2006, pp. 149–155.

[11] K. Jain, L. Lovász, and P. A. Chou, "Building Scalable and Robust Peer-to-Peer Overlay Networks for Broadcasting Using Network Coding," *Journal on Distributed Computing*, vol. 19, no. 4, pp. 301–311, Dec. 2006.

[12] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Hybrid arq-random network coding for wireless media streaming," *Communications and Electronics*, Jan 2008.

[13] H. Seferoglu and A. Markopoulou, "Opportunistic Network Coding for Video Streaming over Wireless," in *Proc. of Packet Video 2007*, Lausanne, Switzerland, Nov. 2007.

[14] D. Nguyen, T. Nguyen, and Xue Yang, "Multimedia Wireless Transmission with Network Coding," in *Proc. of Packet Video 2007*, Lausanne, Switzerland, Nov. 2007.

[15] X. Liu, G. Cheung, and C.-N. Chuah, "Rate-distortion optimized network coding for cooperative video stream repair in wireless peer-to-peer networks," *Proceedings of WoWMoM 2008*, pp. 1 – 6, May 2008.

[16] H. Wang, R.Y. Chang, and C.C.-J. Kuo, "Wireless Multi-party Video Conferencing with Network Coding," in *Proc. of IEEE ICME*, 2009.

[17] H. Seferoglu and A. Markopoulou, "Network Coding-Aware Rate Control and Scheduling in Wireless Networks," in *Proc. of IEEE ICME*, 2009.

[18] M. Grangetto, R. Gaeta, and M. Sereno, "Rateless Codes Network Coding for Simple and Efficient P2P Video Streaming," in *Proc. of IEEE ICME*, 2009.