

Network Coding and Media Streaming (Invited Paper)

Nikolaos Thomos and Pascal Frossard

Signal Processing Laboratory (LTS4)

Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Email: {nikolaos.thomos, pascal.frossard}@epfl.ch

Abstract—Network coding has recently emerged as an alternative to traditional routing algorithms in communication systems. In network coding, the network nodes combine the packets they receive before forwarding them to the neighbouring nodes. Intensive research efforts have demonstrated that such a processing in the network nodes can provide advantages in terms of throughput or robustness. These potentials, combined with the advent of ad hoc and wireless delivery architectures have triggered the interest of research community about the application of the network coding principles to media streaming applications. This paper describes the potentials of network coding in emerging delivery architectures such as overlay or peer-to-peer networks. It overviews the principles of practical network coding algorithms and outlines the challenges posed by multimedia streaming applications. Finally, it provides a survey of the recent work on the application of network coding to media streaming applications, both in wireless or wired communication scenarios. Promising results have been demonstrated but delay and complexity constraints are still posed as the main challenging issues that prevent the wide-scale deployment of network coding algorithms in multimedia communication systems.

Index Terms—Network coding, error resiliency, overlay networks, p2p streaming, wireless streaming.

I. INTRODUCTION

The traditional multimedia coding and streaming architectures have been challenged in the recent years with emerging applications such as wireless low-power multimedia sensor networks, and portable devices with multimedia coding and communication capabilities. The widespread deployment of efficient communication systems combined with the proliferation of digital media content from numerous sources has fostered the development of a novel media delivery framework built on ad hoc or overlay networks. Such architectures present a high diversity in terms of sources, paths and clients, which could be exploited for improved performance. However, these architectures necessitate appropriate distribution and coding strategies for proper exploitation of the network diversity.

Traditional tools such as source coding, channel coding or routing can be enabled for efficient multimedia streaming in networks with diversity. However, the optimization of these algorithms is often very complex in overlay architectures. End-to-end optimization often requires a global knowledge about the network and might lead to

suboptimal solutions due to conservative channel rate allocation strategies. Clearly, the performance can be improved by requesting the network nodes to perform basic processing operations in order to increase the quality of service. The network nodes in an overlay or peer-to-peer network can implement simple channel coding, packet filtering, or adaptive routing. Joint source and channel coding algorithms could typically lead to efficient solutions when they are combined with adaptive network protocols or appropriate routing [1]. The network nodes could decode and encode packets, and adapt the transmission policy to the state of the network and the content of the packets in order to maximize the overall system performance. However, such solutions are generally quite complex and do not scale well with the number of sources or the size of the network.

Network coding has received increasing interest recently from the research community, as it seems to provide an efficient alternative for delivering data in packet networks with diversity. This has been initiated by the seminal paper of Ahlswede et al. [2], where it is shown that the network throughput can approach the max-flow min-cut limit of the network graph when the network nodes are allowed to combine received packets instead of simply forwarding them. Network coding has great potentials in terms of capacity, delay and resiliency to loss in broadcast or multicast scenarios. It improves the system performance while it avoids the use of complex routing or scheduling algorithms. It also necessitates a reduced control overhead in networks with diversity.

Network coding has been successfully applied for example to content distribution [3], [4], distributed storage [5] or data dissemination [6]. More recently, it has also attracted the attention of the media streaming community since network coding can be seen as a particular form of channel coding that becomes very interesting in decentralized streaming architectures. Network coding permits to avoid the reconciliation or coordination between the network nodes, which is particularly appealing for the design of distributed solutions in ad hoc networks. However, the application of network coding principles in multimedia streaming systems is not a trivial task as streaming applications generally impose strict timing and complexity constraints that limit the coding opportunities.

This paper surveys the recent research efforts that have studied the application of network coding principles to multimedia communication applications. It first provides a brief overview of the network coding (NC) potentials

Manuscript received May 27, 2009; revised July 20, 2009; accepted August 19, 2009.

for data delivery in networks with diversity. The coding algorithms are then studied in more details in III. While complete overviews of the network coding theory can be found [7], [8], [9], [10], [11], we focus here on the design of practical solutions with limited computational complexity. In particular, we discuss in III the main challenges towards the application of network coding in streaming systems, which are the control of the end-to-end delay and the computational complexity in the network nodes. Later, we provide in V an overview of the recent research efforts that have exploited the benefits of network coding for streaming in peer-to-peer architectures or wireless networks.

This paper is similar in spirit to [12], [13] that provide a short overview of network coding in streaming applications. It however provides a more comprehensive description of the recent research in this area. Furthermore, it is complementary to the very good description of practical network coding algorithms that is proposed in [14], as it emphasizes the recent research efforts towards exploiting the theoretical benefits of network coding in media streaming in ad hoc delivery architectures.

II. NETWORK CODING POTENTIALS

A. Overview

The basic idea of network coding is relatively simple. When a network node detects a transmission opportunity, it combines some of the packets it has received previously and sends the result to the next hop network nodes. Intuitively, processing in the network nodes increases the overall system performance as it permits to avoid conservative end-to-end policies and to act close to the bottleneck points in the network. Network coding provides an interesting alternative to routing algorithms in networks with diversity. It also permits to reduce the control overhead in decentralized systems since it avoids the need for reconciliation or coordination between nodes.

The coding operations in the network nodes increase the network throughput and participate to limiting the end-to-end delay. In addition, they provide increased robustness to packet losses. Even if network coding techniques are similar in different architectures such as wired or wireless networks, the underlying challenges might be however quite different. We provide in this section a few toy examples that illustrate the benefits of network coding in different situations. In particular, these examples shed light on the throughput gains induced by the application of network coding techniques for both wireless and wired communication. We also motivate the use of network coding in error prone networks.

B. The butterfly network

The advantages of network coding [2] are usually illustrated by the butterfly network shown in Fig. 1. This toy example represents a simple multicasting scenario where two servers S_i , $i = 1, 2$, transmit two packets X_1 and X_2 to the clients C_1 and C_2 . All links have a

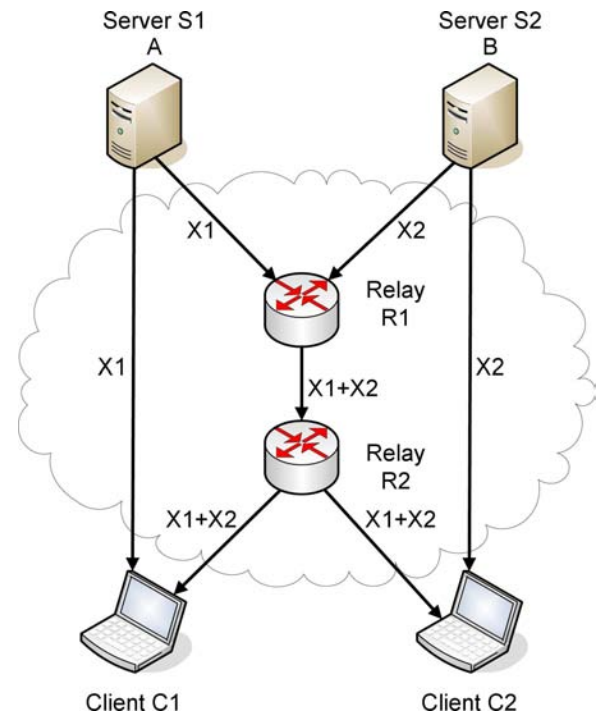


Figure 1. Toy example of a simple multicasting scheme with two clients and two servers. When links have a unit capacity, delay and throughput gains are achieved when the packets are combined at R_1 .

capacity of one packet per time slot. In this topology, there exist several paths connecting clients and servers, either directly or through two relay nodes R_i that provide assistance in the transmission process.

In this situation, traditional routing schemes would first discover the paths among the clients and servers and then find the optimal transmission schedule for delivering the packets to all clients. However, in the simple butterfly network topology, there is no transmission schedule that permits to serve simultaneously both clients, as the link between the routers R_1 and R_2 is the bottleneck of the network. This link can accommodate either packet X_1 or packet X_2 , but not both at the same time. Thus, if links S_1C_1 and S_1R_1 are allocated for the transmission of packet X_1 and respectively S_2C_2 , S_2R_1 for packet X_2 , one of the clients receives both packets. The other client should however wait for at least one additional time slot (when R_1 has a buffer) until it receives the second packet.

If the relay node R_1 implements network coding, the throughput of the system can however be increased. Equivalently, the delay necessary for both clients to receive all the packets can also be reduced. If R_1 is able to perform coding operations on the received packets, it could transmit the combination $X_1 + X_2$ instead of sending either X_1 or X_2 . With this strategy, both clients receive one of the two original packets from the sources directly and the network coded packet $X_1 + X_2$ through the bottleneck path. This results in a simple equation system whose solution leads to the recovery of both packets at all clients.

The simplistic butterfly network shows clearly that

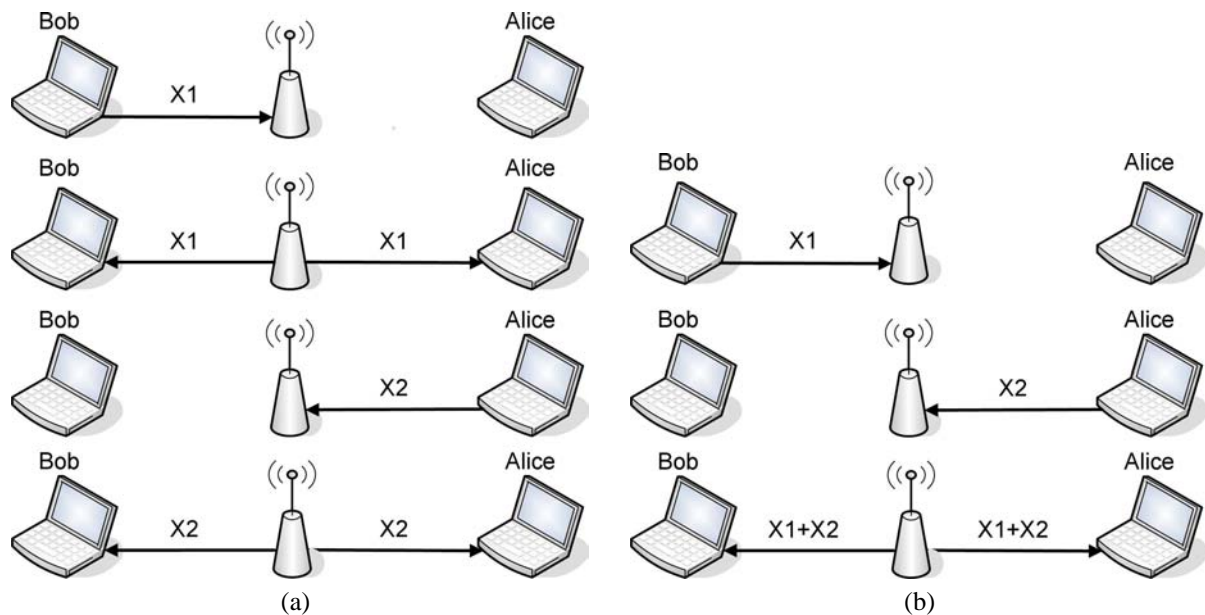


Figure 2. Wireless point-to-point communication. (a) Store and forward approach. (b) Network coding based scheme. The transmission is completed in fewer time slots when network coding is followed.

routing is not always optimal and that network coding may lead to a better exploitation of the network resources. Network coding permits to reduce the number of transmissions that are required for completing the data delivery to all clients. Similarly, it permits to reduce the delay needed for the delivery of all packets. This is quite appealing for streaming applications where the timing constraints are generally strict.

C. Wireless relay networks

Network coding seems to be appropriate also for wireless networks that usually present a significant network diversity due to the multiple transmission paths between servers and clients. In addition, the wireless medium is typically a broadcast channel that represents the appropriate framework for network coding. The benefit of network coding is illustrated by the simple wireless relay network of Fig. 2. In this simple example, Bob and Alice want to exchange packets X_1 and X_2 . In the proposed example, direct communication is not possible because the transmission range is limited and does not permit to reach directly the other client. Bob and Alice therefore use the relay antenna for communication. All links are assumed half-duplex and therefore cannot receive and send data simultaneously. Without employing network coding (Fig. 2(a)) at the relay antenna, the exchange of packets X_1 and X_2 takes four time slots. Each user should first transmit its data to the relay and wait for the relay to transmit it to the other user. However, when the relay node implements network coding, the required time for exchanging both packets can be reduced to three time slots, as illustrated in Fig. 2(b). Bob and Alice successively transmit their data to the relay node. Both packets are combined in the relay node and the resulting network coded packet $X_1 + X_2$ is broadcasted to both

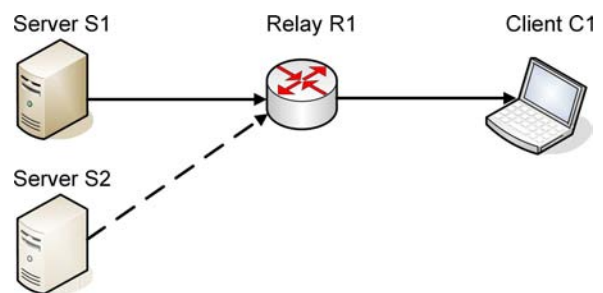


Figure 3. Two-links tandem channel. Network coding at relay R_1 save significant amounts of bandwidth. Furthermore, the communication is more robust.

clients. Bob and Alice can recover all the original packets by solving the simple equation system induced by network coding.

From the above example it is obvious that there are gains regarding the delay as transmission is completed in fewer time slots. There are also gains in terms of bandwidth as the network resources are reserved for a shorter period. Both types of gain are interesting for wireless streaming applications in ad hoc architectures. Finally, the energy consumption is also reduced since the antennas are used less frequently. These benefits are very appealing in wireless networks where relays and antennas are usually low-cost devices with short-time battery life. For a detailed survey about the challenges and the opportunities for network coding in wireless networks, we refer the interested readers to [15].

D. Error-prone tandem network

Network coding can also be beneficial in error prone networks where it permits to exploit efficiently the network diversity for robust data transmission, without the

need of large overhead for node coordination. The advantages of network coding for error resiliency can be illustrated by the two-link tandem network of Fig. 3. The server S_1 wants to transmit some packets to the client C_1 . The communication is performed on lossy channels through the relay node R_1 . Traditional schemes usually employ channel coding at the sources with erasure correcting codes like rateless [16] or Reed-Solomon [17] codes in order to cope with the packet erasure. The channel protection is determined so that the end-to-end performance is maximized. If $\epsilon_{S_1R_1}$ and $\epsilon_{R_1C_1}$ are the loss rates over the links S_1R_1 and R_1C_1 respectively, the client can communicate with the server at rate of $(1 - \epsilon_{S_1R_1}) \cdot (1 - \epsilon_{R_1C_1})$ packets per time slot when the links have a capacity of one packet per time slot. However, if the relay node can decode the packets and then re-encode them, the communication rate increases and becomes equivalent to the minimum of the capacities, i.e., $\min \{(1 - \epsilon_{S_1R_1}), (1 - \epsilon_{R_1C_1})\}$.

Decoding and re-encoding is a form of network processing that can increase the performance in the case of error-prone tandem networks. However, this solution has important limitations in real systems as the end-to-end delay increases rapidly with the size of the network. In addition, decoding in the network nodes may prove to be too complex. Finally, there is actually no guarantee that the relay node can decode the received packets. If the relay R_1 performs packet combinations without priori decoding, it forwards useful information to the client even if R_1 cannot decode it. However, this cannot increase the symbol diversity since all network coded packets are actually combinations of only the received by R_1 packets. Fortunately, by exploiting the network diversity and adapting network coding algorithms it is possible to provide communication rates close to the capacity [18] without significant delay penalty. If a second server S_2 is added to the network and streams the same multimedia content through R_1 , then the communication rate is in general higher and equals to $\min \{\max \{(1 - \epsilon_{S_1R_1}), (1 - \epsilon_{S_2R_1})\}, (1 - \epsilon_{R_1C_1})\}$. Obviously, there are potential gains by jointly exploiting network coding and network diversity. If the number of network coded packets is large enough, and if the packet combinations have been selected properly, network coding permits to recover from packet losses on the successive network links. In addition, network coding interestingly tends to distribute the information evenly among the different packets. As all the network coded packets have the same importance, there is no need for expensive routing or scheduling algorithms in this case. The benefit of network coding becomes even more apparent when the number of source peers or the number of paths between the server and the client increases. The network diversity is obviously useful in error prone networks. Network coding becomes therefore quite appealing for media streaming in best effort packet overlay networks.

We can note that network coding in packet erasure networks shares many concepts with rateless channel

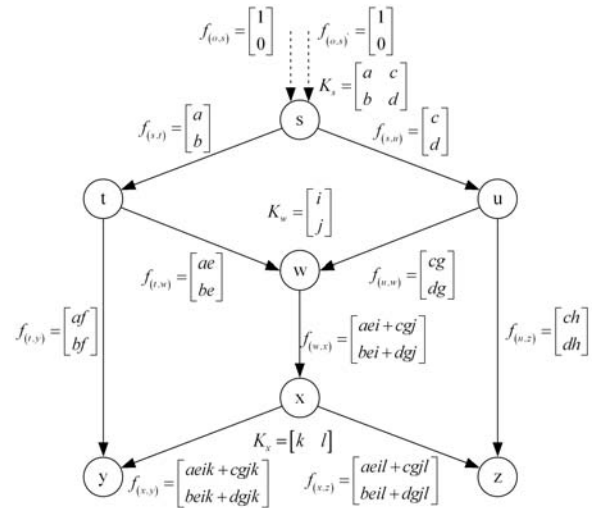


Figure 4. Two dimensional linear network code.

coding techniques such as Raptor codes [19] and LT codes [20]. Rateless codes combine by XOR operations a number of packets that are randomly selected by a pseudo-random generator while following a given degree distribution. Since the structure of these codes is implicit a small header is appended to each rateless encoded packet in order to provide the information necessary for decoding. This is similar to practical network coding schemes [18] that will be described in more details in Section IV.

III. LINEAR NETWORK CODING

In the previous section, we have shown that network coding has very appealing characteristics for data delivery in networks with diversity. We present here a brief theoretical description of the network coding principles. The theoretical limits of network coding are studied in more details in the network information theory and network coding literature [10].

While many coding schemes could a priori be implemented in the network nodes, linear network coding [21] is probably the most successful network coding algorithm due to its relatively low complexity and ability to achieve network capacity in multicast problems. It owes its name from the linear operations performed with the received packets. Elements of graph theory are used for modeling the behavior of linear codes. The network is represented as a graph $G = (V, E)$, where V and E are respectively the set of nodes and edges in G . Only directed graphs are considered, where parallel links of unit capacity are assumed between any pair of nodes. A message of ω symbols in a base field F is sent from a source node s to a collection of nodes. The base field is typically a Galois Field $GF(q)$ of size q in which the coding operations are performed. The network nodes receive parts of the message and map the received symbols into other symbols in F by linear combinations. They finally forward the encoded symbols to the downstream nodes.

Linear network coding is illustrated in Fig. 4, where the message generated at source s consists in $\omega = 2$ symbols. Following the development in [10], we define the local encoding kernel of a ω -dimensional network code at node t as a matrix $K_t = [k_{d,e}]_{d \in In(t), e \in Out(t)}$ of size $In(t) \times Out(t)$, where $In(t)$ and $Out(t)$ are respectively the set of incoming and outgoing links of node t . The local encoding kernel is described by $k_{d,e}$ for every pair of adjacent links (d, e) of the node t . The global description of a linear network code is determined by a column ω -vector f_e for every outgoing link e , as follows:

$$f_e = \sum_{d \in In(t)} k_{d,e} \cdot f_d,$$

for $e \in Out(t)$. It is important to note that the information content does not increase with network coding. In addition, the rate at which information is transmitted from the source node s to any node t or any set of nodes T cannot exceed the value of the maximum flow from s to t , determined by the minimum cut of the network graph. It can further be shown that there is a dependency between the performance of the code, the code's dimension ω and the network topology. In particular, if V_t denotes the linear span of the set of vectors $\{f_d : d \in In(t)\}$ and V_T denotes the union of the spans corresponding to all the nodes $t \in T$, we have [10]

$$\dim(V_T) \leq \min\{\omega, \text{maxflow}(T)\}.$$

The maximum flow of a ω -dimensional linear network code is limited by the minimum between the code's dimension and the maximum flow that could be achieved when routing is applied in the same situation. A ω -dimensional linear network code can further be classified into three main classes that are (i) linear multicast, (ii) linear broadcast and (iii) linear dispersion codes if respectively

- 1) $\dim(V_t) = \omega$ for every non source node t with $\text{maxflow}(t) \geq \omega$.
- 2) $\dim(V_t) = \min\{\omega, \text{maxflow}(t)\}$ for every non source node t .
- 3) $\dim(V_T) = \min\{\omega, \text{maxflow}(T)\}$ for any set T of non source nodes.

We can observe that a linear dispersion code is a linear broadcast code, and a linear broadcast code is a linear multicast code too. From these relations, it appears that the construction of a linear code has to consider both the value of ω and the network topology, along with the size of the base field F . It can be noted that the linear codes can also be studied from an algebraic perspective [22] since coding operations can be represented as matrix operations.

Several works have addressed the problem of the construction of good linear network codes. The design of network codes involves in particular the selection of proper coefficients such that all local encoding kernels are full rank. A polynomial time algorithm for constructing ω -dimensional linear network codes on an acyclic network

with edges of unit capacity is presented in [23]. A randomized version of the code, which is more flexible but increases the computational complexity, is proposed in the same work. The results are further generalized to topologies with links of larger (integer) capacity, by assuming the existence of multiple parallel unit capacity edges between the nodes. The proposed construction algorithms are valid for networks suffering from non-ergodic link failures. A code construction method that is robust to link failures is presented in [24]. It utilizes a single encoding kernel with the observation that the local encoding kernel of dimension $\omega - 1$ at every non-source node is equivalent to the original ω -dimensional linear network code. The proposed construction is simpler than the method proposed in [23] and necessitates smaller memory. Finally, the work in [25] translates the network code design problem into vertex coloring problem of reduced complexity.

While linear codes are optimal for the multicast problem, they are not optimal in general communication settings, especially when several flows are coded jointly. To cope with the inefficiencies of linear network codes the relation between channel coding and network coding has been investigated. For example, the relation between network coding and convolutional codes in fixed topology networks has been studied in [26]. The proposed network coding scheme decomposes the network graph into sub-graphs for constructing convolutional codes. LDPC codes [27] have been used for network coding in wireless relay networks [28], [29]. The network nodes broadcast information and serve as relays for the other nodes. In the proposed adaptive network coding algorithm, each relay processes the received information and generates LDPC parity information in an attempt to match the behavior of codes on graph (i.e., LDPC codes).

IV. TOWARDS PRACTICAL MULTIMEDIA NETWORK CODING

The theoretical works on network coding have made apparent that linear network codes are valuable for multicasting. Unfortunately, linear network coding described in [22], [23] necessitates the use of computationally complex algorithms for defining the coding coefficients. In addition, the design methods generally assume that the servers have a full knowledge of the network topology. Linear network coding is therefore not practical in large scale dynamic (ad hoc) networks. In order to alleviate these problems, several works [30], [31], [32] have proposed to implement network coding with a random selection of the coefficients. If the coefficients are chosen in a sufficiently large Galois Field $GF(q)$, random linear network coding can achieve the multicast capacity with a probability that asymptotically approaches one for long codelengths (high number of source packets). The probability that all receivers are able to decode the source message is larger than

$$(1 - d/q)^r,$$

where d is the number of receivers and q represents the size of the field [31]. The parameter r corresponds to the maximum number of links receiving signals with independent randomized coefficients in any set of links constituting a flow solution from all sources to any receiver. This lower bound on the decoding probability holds for independent or linearly correlated sources and for networks with or without delays. Randomized linear network coding provides therefore a low complexity alternative towards the design of practical algorithms, since it permits to relax the requirements about the full knowledge of the network topology. It permits to implement distributed solutions with independent coding decisions in each node. Such a distributed algorithm is particularly interesting in ad hoc networks

Motivated by the results of randomized network coding, the authors in [18] propose a practical network coding system for streaming applications. They define a proper format that can be used in random network graphs without the need for a hypernode that is aware of all coding coefficients and the overall topology. A header is assigned to each packet and contains the coding coefficients. As the packets travel through the network they are subject to successive coding operations that modify both the message and header parts of every packet. All packets therefore contain encoded symbols along with the coefficients that have been used for their computation. The header part defines a global encoding vector that can be used at any decoder to recover the original message by Gaussian elimination, typically. It is shown that the equation system built by the successive network coding operations is full rank with probability 99.6% when the computations are performed on $GF(2^{16})$. Smaller fields like $GF(2^8)$ are sufficient in practice [18].

Even if network coding permits to combine any packets in the network nodes, the coding choices clearly have an influence on the end-to-end delay of the transmission system. Delay could be caused by the asynchronous receptions of packets through the incoming links because of the different propagation and queuing delays that the packets experience in the network. This in turn can create encoding delays if the network coding nodes has to wait for the reception of a given subset of packets before it could combine them. Delay is however a very important issue in streaming applications that often impose stringent timing constraints. In order to cope with the buffering delay problem, the authors in [18] introduce the concept of *generation*. A generation is a group of packets with similar decoding deadlines, which can be combined together by the network coding operations. The generation of every packet is identified by a small header of one or two bytes that is added to each packet. At the network nodes, the packets are stored into the buffer upon their reception. Whenever there is a transmission opportunity, the network coding node linearly combines the available packets and transmits an encoded packet. As coding becomes constrained to the packets of the same generation, the resulting delay is limited. Obviously, the

generation size affects the coding efficiency. Specifically, when generations consist of small number of packets, the nodes' ability to generate independent packets is reduced and the overall system's redundancy is increased. In order to maximize the robustness of the system, the packets that are not innovative with respect to the information that has been previously received, are simply discarded, and not forwarded nor encoded [18]. The clients finally implement a progressive decoding strategy based on Gaussian elimination. The decoding becomes successful if the number of received packets is equal to the size of the generation. The delay in the system is mostly driven by the time that is needed for each client to collect enough packets.

One can feel from the above discussion that there is a trade-off between delay and coding efficiency in practical systems. If the delay is highly constrained, the number of packets that can be combined together decreases, and the coding efficiency or the robustness of the system is affected. On the other hand, if the generation size increases, network coding becomes more efficient due to the enhanced capability of generating independent network coded packets. However, the system's performance is affected by buffering delays at each coding stage in the network. The generation size also influences the computational complexity of the system. If the number of symbols that are encoded together becomes large, the computational complexity augments in the system. For example, the decoding complexity increases when the generation size is large, as it directly influences the size of the equation system that has to be solved at the receiver. Practical network coding systems have therefore to meet an appropriate trade-off between coding efficiency that increases with the size of a generation, and both the computational complexity and the end-to-end delay that augments with the generation size [33].

The design trade-offs are generally not easy to optimize. Interestingly enough, packets in multimedia communication applications have typically different importance in regards to the quality of the decoded streams. The unequal importance of data is another characteristic that should be used for the design of efficient network coding solutions. It influences the choice of an appropriate trade-offs between size of the generation, delay and system robustness. For example, the packets can be organized into classes of importance, and the coding operations can be adapted to each class, similarly to the concept introduced in Prioritized Encoding Transmission (PET) [34]. Unequal protection permits to recover in priority the most important packets, even if the number of received packets is insufficient to decode all the data. The adaptation of the coding strategy to the particular characteristics of the multimedia packet streams can lead to efficient design trade-offs with graceful quality degradations due to losses and reasonable end-to-end delay and complexity.

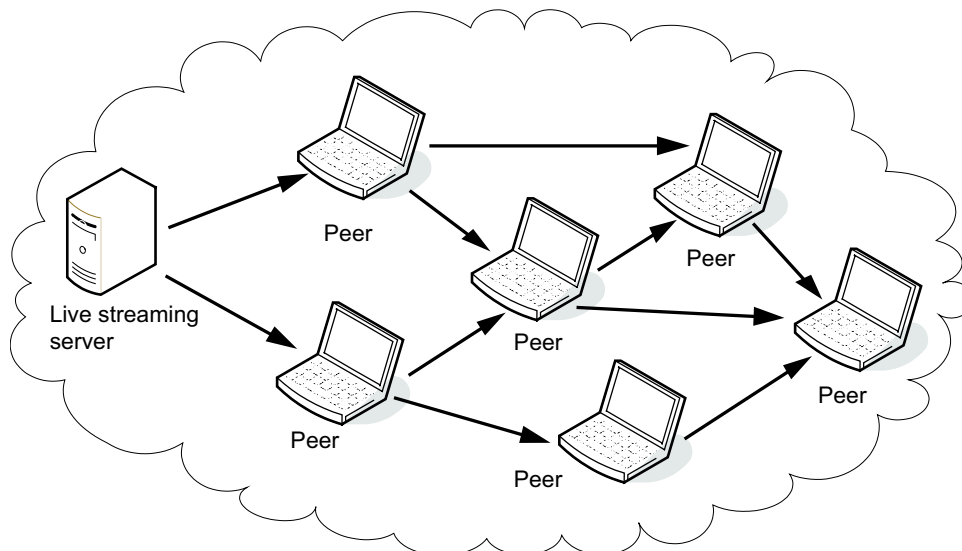


Figure 5. Peer-to-peer streaming. The stream originated from the server is forwarded by the different network peers so that it can reach all the clients. The network nodes can implement network coding in order to increase the throughput and the robustness of the system.

V. APPLICATION OF NETWORK CODING TO MEDIA STREAMING

The application of network coding algorithms to media streaming has to properly consider the specificities of multimedia communication applications, such as strict delay constraints, high bandwidth requirements, as well as the unequal importance of the data that further presents some tolerance to packet losses. The design of the system has to take all these parameters into consideration in order to produce efficient practical solutions with reasonable complexity.

When properly designed, network coding is able to take advantage of the network diversity that consists in multiple source peers and multiple transmission paths. Network coding can be used to improve the throughput of a streaming system, to reduce the end-to-end delay, or to increase the robustness to packet loss, for example. It also provides an efficient solution that reduces the control overhead and avoids the need for reconciliation in distributed systems. We overview below the recent research works that have mostly focused on peer-to-peer multicast scenarios or wireless streaming applications. While this overview is certainly not fully comprehensive, it describes the most relevant research works in the spirit of the present paper.

A. Peer-to-peer streaming

Network coding finds a perfect application in peer-to-peer multicast systems. Such systems have become recently very popular, as they rely on the bandwidth contributions from peers in order to reduce the load on the main streaming server. Multicast streaming is implemented by forwarding the media packets from the servers to all the clients via other peers that are grouped in an overlay or ad hoc configuration (see Fig. 5). The packet distribution is mostly organized in two modes, which are

the push or pull strategies. In the first case, the packets are simply pushed through the different peers in a way that is determined by the senders. In the pull scenario, the clients request specific packets or group of packets from the source peers. Network coding can be beneficial in both cases, as it helps to cope with the network dynamics.

Multimedia streaming systems employing network coding techniques have been motivated by the success of Avalanche [3], [4] which has been proposed for large scale content distribution in peer-to-peer networks. It envisions the deployment of huge overlay networks that allow fast downloading. This distributed architecture improves significantly the file download time of BitTorrent [35]. Avalanche does not necessitate random and local rarest packets policies of BitTorrent and other current protocols since all packets have equal importance due to randomized network coding. It employs a simple system that enables users to locate neighboring users with innovative information for them. For penalizing selfish users a tit-for-tat approach inspired by BitTorrent is utilized. The scheme is extremely robust to random nodes arrivals and departures. It also improves significantly the downloading times compared to traditional schemes that do not perform coding operations with packets.

One of the first works that has studied the performance of network coding in peer-to-peer (p2p) streaming has been proposed in [36]. Randomized linear network coding is implemented in a system called "Lava" in order to evaluate the tradeoffs and benefits of network coding in live p2p streaming. The system offers network coding as an option in a pull-based p2p streaming solution that allows for multiple TCP connections for multiple upstream peers. Prior to transmission, the streams are divided into segments of specific duration, similar to the idea of generation proposed in [18]. These segments are further divided into blocks that undergo network coding operations in the

different peers. The peers periodically exchange messages to announce the availability of segments in a pull-based manner. At any time, peers make concurrent requests for segments that are missing in their playback buffer by addressing randomly one of the peers that possess the segment of interest. The peers then decode the segments from their playback buffer in a progressive manner using Gauss-Jordan elimination. The evaluation shows that the network coding scheme is resilient to network dynamics, maintains stable buffering levels and reduces the number of playback freezes. Network coding is shown to be most instrumental when the bandwidth supply barely meets the streaming demand.

Based on the encouraging results of [36], the same authors redesign the peer-to-peer streaming algorithm and propose the R^2 architecture in [37]. In R^2 , randomized linear network coding is combined with a randomized push algorithm to take full advantage of coding operations at peer nodes. The peers periodically exchange buffer maps that indicate the segments that have not been fully downloaded yet. The R^2 system sends the buffer maps together with the data packets whenever it is possible, otherwise they are transmitted separately. The frequency of these information's exchanges has to be chosen high enough, in order to avoid the transmission of redundant segments. Whenever a coding opportunity is detected, a peer randomly chooses a video segment that the downstream peer has not completely received and generates a network coded block. The segment selection is inspired from [38]. The system also uses large segment sizes in order to avoid the transmission of too much overhead information by buffer map exchanges. The streams are progressively decoded by Gauss-Jordan elimination, similarly to the Lava system described above [36]. The R^2 system provides several advantages in terms of buffer level and delay, as well as resilience to network dynamics. The scalability of the system is also increased. Most of these advantages are due to the combination of push-based methods with randomized linear network coding.

The organization of the peers in the overlay network has a large influence on the performance of the streaming system. In particular, the delivery has to be organized in such a way that the bandwidth constraints can be respected, and such that the clients with the smallest bandwidth do not penalize the performance of the overall system. A method for constructing peer-to-peer overlay networks for data broadcasting is proposed in [39]. The overlay construction imposes that all the peers have the same number of parents nodes, which are the nodes that send them the data packets. Such a constraint tends to distribute the load over the network. Network coding is then used in the peer nodes for increased throughput and improved system robustness. One could also organize the overlay into several layered meshes. Heterogeneous receivers can then subscribe to some of the meshes depending on their capacities [40]. The data is similarly organized into layers, and network coding is performed on packets of the same layers. The practical network

coding scheme of [18] is adopted in this work due to its low complexity. The construction of the layered meshes takes into consideration the overlapping paths in order to exploit the network coding benefits. Depending on the network state and the clients' requirements, every receiver determines the proper number of meshes that it has to subscribe to. The network throughput is finally increased by network coding combined with appropriate mesh organization. In the same spirit, the work in [41] proposes to split the bitstream into several sub-bitstreams for streaming over peer-to-peer networks. A neighbourhood management algorithm is then used to schedule appropriately the transmission of the different encoded sub-bitstreams. Finally, the problem posed by the heterogeneity of the receivers could also be solved by combining network coding with multiple description coding as proposed in [42].

B. Resiliency to packet losses

Network coding principles can also be used to increase the robustness of the streaming system. Multimedia streaming imposes in general strict timing constraints, which may render some of the packets useless if they are late at the receiver. Overall, the system has to be robust to packet erasures and maintain low delay for improved performance. The increase of network throughput described in the previous section is only beneficial if the clients can decode the media packets.

One of the first attempts to realize some type of coding in the nodes of an overlay network is presented in [43], [44]. The nodes are organized in multicast trees. Some of them implement channel coding operations to increase the robustness of the system. These are called network-embedded FEC (NEF) nodes and perform Reed-Solomon (RS) decoding on the packets they receive. The decoded packets are encoded again with RS codes before transmission to the children peers. NEF nodes permit to increase the resiliency of the system, while avoiding a waste of resources with strong end-to-end protection. A greedy algorithm determines the number of NEF nodes and their location. Only a few well-positioned NEF nodes are sufficient to provide significant network throughput gains that result into a high video quality.

Similarly, decoding and encoding based on fountain codes is performed in the network nodes in [45]. The LT codes [20] are used in this work since they perform close to perfect codes and eliminate the need for reconciliation among network peers and for packet scheduling. The intermediate network nodes wait for receiving a sufficient number of packets to recover the source content. Then the source packets are re-encoded into a new set of LT packets that differ from the packets produced independently in the other nodes. This is made possible by the rateless property of LT codes, which allow for the generation of an infinite number of different packets. Decoding and encoding in the nodes however come at the price of increased complexity and delay. The network topology is however constructed such that minimal delays can be

achieved. The streaming system is shown to be resilient to network dynamics with an increased throughput due to the rateless properties of the LT codes.

Decoding operations in the network nodes could be avoided and replaced by linear packet combinations. The work in [46] proposes to take benefit of the properties of Raptor codes [19] that offer linear encoding and decoding complexity and rateless characteristics. Packets are encoded with Raptor codes at the servers. The network nodes then selectively combine packets when they have to compensate for packet losses and bandwidth variations. Such a system does not necessitate the use of large buffers in the nodes and the coding operations are kept very simple. This solution is advantageous in terms of delay and complexity compared to methods that would implement Raptor decoding in the nodes, in a similar manner to LT decoding in [45]. A rate allocation algorithm [47] further determines the optimal source and channel rates so that the quality is maximized for the smallest capability client. The resulting scheme is shown to be extremely robust to network variations.

Finally, network coding could also be used for recovering from errors in broadcast applications. When clients experience errors, packets retransmissions rapidly lead to bandwidth explosion as every client might request a different packet. Network coding is helpful in limiting the number of retransmissions since it replaces the retransmission of original packets by the transmission of packet combinations that can be decoded at the clients. For example, the work in [48] studies the problem of broadcasting using network coding over one hop WiMAX networks. Network coding is applied whenever several packet losses are reported. Such a strategy is shown to be more efficient than state-of-the-art error resilient transmission schemes.

C. Prioritized network coding

Media streams are generally characterized by packets with different importance with respect to their contribution to the quality at the decoder. Network coding can adapt to this property by handling the packets according to their priority. Network coding based on Prioritized Encoding Transmission (PET) [34] principles has been initially proposed in [18], where data of high importance receives a high level of error protection by means of a proper arrangement of the data blocks in the encoding matrix. Unequal error protection is also proposed in [49]. The PET algorithm is replaced by a MD-FEC scheme [50], which seeks for the distortion-minimal source and channel rate allocation for the given channel conditions. Prioritized network coding is applied to scalable video streams in [51]. Data is segmented and interleaved in the coding matrix, in such a way that the base layer typically receives more redundancy bits than the enhancement layers. Classical network coding is then performed on packets of the same generation. The proposed scheme is shown to outperform other solutions based on either routing or routing with replication policies.

One could also achieve different levels of protection by changing the network coding scheme itself, where the coding operations are adapted to the importance of the packets. Priority random linear codes are proposed in [52] for data dissemination in peer-to-peer and sensor networks. Improved data persistence is achieved due to the fact that the most important video data represents a combination of fewer source packets. In [53], the prioritized encoding problem is casted as inter-session network coding problem [54]. In inter-session network coding, combinations of packets from different information sources are allowed when the clients simultaneously subscribe to all sources. The layered data can then be organized into multiple pipes that convey the network coding packets. The data from the most important layer typically flows into the first pipe. The second pipe transmit packet combinations from the first two layers. The other layers are arranged similarly, and network coding is applied on embedded sets of data. The packets from the most important layers are therefore used more frequently in the coding operations, which lead to a higher recovery probability in a progressive decoding scheme. Competition between the packets of the different layers is avoided in such a scheme, which however requires as many coding buffers as the number of quality levels. Unequal error protection can also be achieved by redefining the global encoding kernel (GEK) as proposed in [55]. This approach decomposes the network graph into connected line graphs with different coding operations, similarly to [25]. It optimizes the level of protection by solving an exhaustive search problem.

Prioritized coding can further be achieved by organizing the packets in multiple classes, depending on their importance. For example, the work in [56] addresses the problem of streaming wirelessly some H.264/AVC encoded video content. Packets are grouped in different classes, and frame dependencies are further taken into account for determining the optimal network coding operations for each class. The coding choices are determined locally in each node by estimating the number of innovative packets received by each client. However, the coding decisions are still complex to compute due to the high number of dependencies between packets. Another network coding algorithm that considers the specific importance of media packets in order to prioritize the delivery of the most important packets is proposed in [57]. Media packets are grouped into classes of different importance and unequal protection is achieved at each node by varying the number of packets from each class that are used in the network coding algorithm. A low complexity greedy algorithm is used locally in each node in order to determine the best coding choice. The proposed scheme outperforms baseline network coding algorithms that do not take into account the importance of the packets for the delivery of layered media streams.

D. Adaptivity and opportunistic coding

Network coding can also be applied for media streaming on shared communication medium. When packet transmission can be overheard by multiple nodes, receivers could build up a buffer of packets that can be used to decode the successive packets. The senders can thus use some knowledge about the receiver status to optimize network coding operations and reduce the overall transmission costs. The COPE architecture has been presented in [58] for communication over wireless mesh networks. It introduces the concepts of listening and opportunistic coding. The antennas listen to the broadcast channel and acquire packets that are stored temporarily in their buffers. When the sender is informed about the receiver status, it could determine packets' combinations that maximize the probability of decoding for a maximum number of clients. For video transmission, the selection of network coding operations is based only on the maximization of the number of clients that can decode the packets. It is however suboptimal, since packets typically have different importance for the reconstructed video [59]. The video quality is significantly improved by selecting the proper network codes that take into account the importance of the data as well as the timing deadlines. Different adaptive coding schemes are presented in [59]. A baseline scheme considers the importance of each packet and selects the most important packet for transmission from the top of the queue, ignoring packets that have been already transmitted. A more advanced scheme uses the rate-distortion optimization framework proposed in RaDiO [60] and incorporate it into the design of the network coding algorithm such that the expected distortion is minimized. Such a solution is shown to outperform simpler schemes that do not consider jointly packet deadlines and packet importance.

VI. CONCLUSIONS

Network coding is an interesting paradigm that requests the network nodes to perform basic processing operations on packets in order to improve the throughput or the robustness of communication systems with network diversity. We have shown in this paper that the theory of network coding finds application in media streaming. Multimedia communication could typically benefit from network coding in overlay networks or wireless mesh networks. Coding decisions could also be adapted to the packet importance or the state of the receivers in order to maximize the quality of service.

There are still a few open issues to solve before network coding algorithms could be widely deployed in streaming applications. For example, the decoding complexity is still pretty high in most of the schemes proposed in the literature due to the length of the coding blocks that is necessary for designing efficient coding solutions. Also, distributed algorithms require to transmit coding information in the packet header, which might lead to large overhead in large networks. Then, network coding might also present security issues, as overheard packets could be

used to reconstruct parts of the signal without permission. Malicious nodes could also introduce corrupted packets and therefore penalize the performance of the system [61].

Nevertheless, content-aware network coding and prioritization techniques have surely a strong potential and may improve the quality of the multimedia services and increase the robustness of the transmission. The choice of the right trade-off between delay, coding efficiency and complexity is still an open issue in network coding. The use of hybrid methods that exploit both the benefits of channel coding and network coding might provide interesting solutions to this compromise. We finally expect that network coding will become a key technology for multimedia applications where the communication is performed in random networks with diversity.

ACKNOWLEDGMENT

This work has been supported by the Swiss National Science Foundation, under grant PZ00P2-121906.

REFERENCES

- [1] J. Chakareski and P. Frossard, "Adaptive systems for improved media streaming experience," *IEEE Communications Magazine*, vol. 45, no. 1, pp. 77–83, Jan 2007.
- [2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [3] C. Gkantsidis, J. Miller, and P. Rodriguez, "Comprehensive View of a Live Network Coding P2P system," in *Proc. of ACM SIGCOMM/USENIX IMC*, Brasil, Oct. 2006.
- [4] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. of the 24th Annual Joint Conf. of the IEEE INFOCOM*, vol. 4, Mar. 2005, pp. 2235–2245.
- [5] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," in *Proc. of the 26th IEEE Int. Conf. on Computer Communications*, Anchorage, Alaska, USA, May 2007.
- [6] S. Deb, M. Médard, and C. Choute, "On Random Network Coding Based Information Dissemination," in *Proc. of IEEE Int. Symp. on Information Theory 2005*, Adelaide, Australia, Sep. 2005, pp. 278–282.
- [7] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*, ser. Source, Foundations and Trends in Networking archive. Boston, MA: Now Publishers Inc, 2007.
- [8] —, *Network Coding Applications*, ser. Source, Foundations and Trends in Networking archive. Boston, MA: Now Publishers Inc, 2007.
- [9] R. W. Yeung, S. Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*, ser. Foundations and Trends in Communications and Information Theory. Boston, MA: Now Publishers Inc, 2005.
- [10] R. W. Yeung, *Information Theory and Network Coding*, ser. Information Technology: Transmission, Processing and Storage. Springer-Verlag, 2008.
- [11] T. Ho and D. S. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.
- [12] E. Magli and P. Frossard, "An Overview of Network Coding for Multimedia Streaming," in *Proc. of Int. Conf. on Multimedia and Expo*, Cancun, Mexico, June 2009.
- [13] A. Markopoulou and H. Seferoglu, "Network Coding meets Multimedia: Opportunities and Challenges," *IEEE MMTC E-letter*, vol. 4, no. 1, pp. 12–15, Feb. 2009.

- [14] P. A. Chou and Y. Wu, "Network Coding for the Internet and Wireless Networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sept. 2007.
- [15] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless Network Coding: Opportunities and Challenges," in *Proc. of Military Communications Conf.*, Cancun, Mexico, Oct 2007, pp. 1–8.
- [16] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *Proc. of ACM SIGCOMM*, Vancouver, BC, Canada, Jan. 1998, pp. 56–67.
- [17] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge University Press, 2003.
- [18] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. of the 41st Allerton Conf. on Communication Control and Computing*, Monticell, IL, USA, Oct. 2003.
- [19] A. Shokrollahi, "Raptor codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
- [20] M. Luby, "LT codes," in *Proc. of the 43rd Annual IEEE Symp. on Foundations of Computer Science (FOCS '02)*, Vancouver, Canada, Nov. 2002, pp. 271–280.
- [21] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [22] R. Koetter and M. Médard, "An Algebraic Approach to Network Coding," *IEEE/ACM Trans. on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
- [23] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Trans. Information Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.
- [24] S. L. Fong and R. W. Yeung, "Variable-Rate Linear Network Coding," in *Proc. of IEEE Information Theory Workshop*, Chengdu, China, Oct 2006, pp. 22–26.
- [25] C. Fragouli, E. Soljanin, and A. Shokrollahi, "Network Coding as a Coloring Problem," in *Proc. of Conf. Information Science and Systems*, Princeton, NJ, USA, Mar 2004.
- [26] C. Fragouli and E. Soljanin, "A Connection Between Network Coding and Convolutional Codes," in *Proc. of IEEE Int. Conf. on Communication*, Paris, France, June 2004, pp. 661–666.
- [27] R. Gallager, "Low-density parity-check codes," *IRE Trans. Information Theory*, pp. 21–28, Jan 1962.
- [28] X. Bao and J. Li, "Matching Code-on-Graph with Network-on-Graph: Adaptive Network Coding for Wireless Relay Networks," in *Proc. of the 43rd Annual Allerton Conf. on Communication, Control and Computing*, Champaign, IL, USA, Sep. 2005.
- [29] ———, "A Unified Channel-Network Coding Treatment for Wireless Ad-Hoc Networks," in *Proc. of the IEEE Int. Symp. on Information Theory*, Seattle, WA, USA, July 2006.
- [30] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," in *Proc. of the 41st Allerton Annual Conf. on Communication, Control, and Computing*, Monticello, IL, USA, Oct 2003.
- [31] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The Benefits of Coding Over Routing in a Randomized Setting," in *Proc. of IEEE Int. Symp. on Information Theory*, Kanagawa, Japan, July 2003.
- [32] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.
- [33] M. Wang and B. Li, "How practical is network coding?" in *Proc. of the 14th IEEE International Workshop on Quality of Service*, pp. 274–278, Jun 2006.
- [34] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority Encoding Transmission," *IEEE Trans. Information Theory*, vol. 42, pp. 1737–1744, Nov. 1996.
- [35] B. Cohen, "Incentives Build Robustness in Bittorrent," in *Proc. of P2P Economics Workshop*, Berkeley, CA, USA, 2003.
- [36] M. Wang and B. Li, "Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, May 2007.
- [37] M. Wang and B. Li, " R^2 : Random Rush 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.
- [38] P. Maymoukov, N. J. A. Harvey, and D. Lun, "Methods for Efficient Network Coding," in *Proc. of the 44th Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, USA, Sep. 2006.
- [39] K. Jain, L. Lovsz, 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.
- [40] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, "LION: Layered Overlay Multicast With Network Coding," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1021–1032, Oct. 2006.
- [41] Y. Liu, Y. Peng, W. Dou, and B. Guo, "Network Coding for Peer-to-Peer Live Media Streaming," in *Proc of the 5th Int. Conf. Grid and Cooperative Computing*, Monticello, IL, USA, Oct. 2006, pp. 149–155.
- [42] M. Shao, X. Wu, and N. Sarshar, "Rainbow Network Flow with Network Coding," in *Proc. of the 4th Workshop on Network Coding, Theory and Applications, NetCod*, Hong Kong, China, Jan 2008, pp. 1–6.
- [43] S. Karande, M. Wu, and H. Radha, "Network Embedded FEC (NEF) for Video Multicast in Presence of Packet Loss Correlation," in *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, Genoa, Italy, Sep. 2005, pp. 173–176.
- [44] M. Wu, S. Karande, and H. Radha, "Network Embedded FEC for Optimum Throughput of Multicast Packet Video," *EURASIP Journal on Applied Signal Processing*, vol. 20, no. 8, pp. 728–742, Sep. 2005.
- [45] C. Wu and B. Li, "rStream: Resilient and Optimal Peer-to-Peer Streaming with Rateless Codes," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 77–92, Jan. 2008.
- [46] N. Thomos and P. Frossard, "Raptor Network Video Coding," in *Proc. of the 1st ACM Int. Workshop on Mobile video (in conjunction with ACM Multimedia 2007)*, Augsburg, Germany, Sep. 2007.
- [47] N. Thomos and P. Frossard, "Collaborative video streaming with Raptor network coding," in *Proc. of Int. Conf. on Multimedia and Expo*, Hannover, Germany, June 2008, pp. 497–500.
- [48] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless Broadcast Using Network Coding," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, Feb. 2009.
- [49] A. K. Ramasubramanian and J. W. Woods, "Video Multicast Using Network Coding," in *Proc. of SPIE VCIP*, San Jose, CA, USA, Jan 2009.
- [50] R. Puri, K. Ramchandran, and K. W. L. V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for Internet video streaming and multicast," *Signal Processing: Image Communication*, vol. 16, no. 8, May 2001.
- [51] H. Wang, S. Xiao, and C. C. J. Kuo, "Robust and Flexible Wireless Video Multicast with Network Coding," in *Proc. of IEEE Global Telecommunications Conf.*, Adelaide, Australia, Nov. 2007, pp. 2129–2133.
- [52] Y. Lin, B. Li, and B. Liang, "Differentiated Data Persistence with Priority Random Linear Codes," in *Proc. of*

the 27th Int. Conf. on Distributed Computing Systems of Contents, Toronto, ON, Canada, June 2007, pp. 47–55.

- [53] Y. Wu, "Distributing Layered Content using Network Coding," in *Proc. of the 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, San Francisco, CA, USA, June 2008, pp. 1–4.
- [54] —, "On Constructive Multi-Source Network Coding," in *IEEE Int. Symp. on Information Theory*, Seattle, WA, USA, July 2008, pp. 1349–1353.
- [55] A. Limmanee and W. Henkel, "UEP Network Coding for Scalable Data," in *Proc. of the 5th Int. Symp. on Turbo Codes and Related Topics, Turbo-coding*, Lausanne, Switzerland, Sep. 2008, pp. 333–337.
- [56] X. Liu, G. Cheung, and C. N. Chuah, "Structured Network Coding and Cooperative Wireless Ad-hoc Peer-to-Peer Repair for WWAN Video Broadcast," *IEEE Trans. on Multimedia*, vol. 11, no. 4, pp. 730–741, June 2009.
- [57] N. Thomos, J. Chakareski, and P. Frossard, "Randomized network coding for UEP video delivery in overlay networks," in *Proc of Int. Conf. on Multimedia and Expo*, Cancun, Mexico, June 2009.
- [58] S. Katti, H. Rahu, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *Proc of ACM SIGCOMM*, vol. 36, no. 4, New York, NY, USA, May 2006, pp. 243–254.
- [59] H. Seferoglu and A. Markopoulou, "Video-Aware Opportunistic Network Coding over Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 713–728, June 2009.
- [60] P. Chou and Z. Miao, "Rate-Distortion Optimized Streaming of Packetized Media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr 2006.
- [61] S. Jaggi, M. Landberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *IEEE Trans. on Information Theory*, vol. 54, no. 2, June 2008, pp. 2596–2603.

Nikolaos Thomos received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2000 and 2005, respectively. During his studies he held teaching and research assistantship positions with the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki.

Currently he is postdoctoral researcher with Signal Processing Laboratory of Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Previously, he was a postdoctoral and postgraduate researcher with the Informatics and Telematics Institute / Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include network coding, multimedia communications, joint source and channel coding and distributed source coding.

Dr. Thomos received a Swiss NSF researcher award in 2008. He is a member of the IEEE and the Technical Chamber of Greece.

Pascal Frossard received the M.Sc. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000 respectively.

Between 2001 and 2003, he was a member of the research staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media coding and streaming technologies. Since 2003, he has been an assistant professor at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, non-linear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution.

Dr. Frossard has been the General Chair of IEEE ICME 2002 and Packet Video 2007, and a member of the organizing or technical program committees of numerous conferences. He has been an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004-) and of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006-). He is an elected member of the IEEE Image and Multidimensional Signal Processing Technical Committee (2007-), the IEEE Visual Signal Processing and Communications Technical Committee (2006-), and the IEEE Multimedia Systems and Applications Technical Committee (2005-). He has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee (2004-2006) and as a member of the IEEE Multimedia Signal Processing Technical Committee (2004-2007). He received the Swiss NSF Professorship Award in 2003, and the IBM Faculty Award in 2005 and the IBM Exploratory Stream Analytics Innovation Award in 2008.