

---

# Media Coding for Streaming in Networks with Source and Path Diversity\*

Nikolaos Thomos and Pascal Frossard

Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

**Summary.** Novel network architectures such as overlay networks offer significant diversity that can compensate for the lack of strict quality of service in today's communication infrastructures. In order to take advantage of this diversity for delay-sensitive media streaming applications, the network systems can employ efficient mechanisms based on source, channel and even network coding. In particular, fountain codes offer interesting benefits for streaming with server diversity. When they are used independently at each server, they permit to avoid explicit coordination between the senders that only have to provide the receivers with enough innovative packets. In addition, network coding allows for improved throughput and error robustness in multipath transmission where the network nodes participate to increase the symbol diversity in the system. We review in this chapter the most popular rateless codes that enable the deployment of low-cost decentralized communication protocols in self-organized dynamic networks. We then describe their application in distributed multimedia streaming solutions. We further discuss the most popular network coding algorithms in practical media streaming schemes. Finally, we show that hybrid systems based on both rateless coding and network coding can deliver high quality media streams with low computational complexity, as they permit to benefit from both server and path diversity in overlay architectures.

## 1 Introduction

Recent advances on overlay architectures have motivated many research efforts for the development of efficient multimedia streaming systems that are able to exploit the structure of such networks. These overlay networks that can be built on peer-to-peer or wireless mesh networks for example, are characterized by a high diversity in terms of source nodes and transmission paths. In particular, a receiver peer can be connected simultaneously to several senders, which can further use multiple communication paths for media packet delivery. It is therefore important to design efficient streaming solutions that are able to exploit the network diversity for improved media quality at receivers.

---

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

The traditional multimedia communication tools such as error correcting codes, retransmission mechanisms, or packet scheduling are challenged in overlay networks as they usually necessitate coordination between the nodes participating in the transmission process, or good estimation of the network status. Although there exist efficient solutions to improve the video quality in distributed settings [1], these cannot be deployed and maintained easily in real networks suffering from unexpected nodes departures and arrivals as well as non-ergodic link failures. Complex systems combining joint source and channel coding with adaptive network protocols and appropriate routing algorithms provide effective solutions for dealing with the delay-sensitive nature of highly heterogeneous media packet streams in traditional architectures [2]. However, they are usually difficult to be implemented in dynamic ad hoc architectures that provide sources and path diversity. They moreover do not scale well with the number of sources and the size of the network. They require a good knowledge of the end-to-end network characteristics for optimal performance. It is also usually difficult to build distributed implementations of such systems, without high communication costs for coordination or decision about the best streaming strategy.

Appropriate coding of the media streams however permits to flatten the differences in importance of the packets and thus relaxes the need for important coordination. In particular, fountain codes [3] have emerged as a solution that allows to take benefit from the network diversity with decentralized algorithms. The main characteristic of these codes relies on the fact that the receiver only needs to gather a sufficient number of innovative packets for decoding. The actual identity of the packets do not play a role with such codes, only the number of packets matters. Strict coordination is not needed anymore in distributed streaming since the sources only have to generate different packets, which happens with high probability even when the encoding is done independently. Another interesting feature of fountain codes relies on their high adaptivity due to their rateless property. As a potentially limitless set of packets can be generated from a given set of source packets, the sender can easily adapt the coding rate to the status of the communication channel. Another appealing advantage of fountain codes is their relatively low encoding and decoding complexity that enables their use in real-time applications. Fountain codes have been successfully applied in multimedia transmission [4, 5] and distributed storage systems [6] for example.

While fountain codes appear to be an ideal solution for coding at the sources, they are difficult to be implemented in the network nodes within the overlay. Successive decoding and encoding processes in the network nodes [7] might induce delays that are too important for delay-sensitive applications. In dynamic and large scale networks, it is however beneficial to request help from the network nodes as this provides higher adaptivity to exploit fully the network diversity. Such in-network processing is typically performed by network coding algorithms. Network coding [8] provides an interesting solution to exploit the path diversity with gains in terms of achievable throughput and delay time. It typically permits to approach the max-flow min-cut limit of the underlying net-

work graph when the network nodes combine the received packets instead of forwarding them unprocessed. Practical network coding solutions are generally based on random packets combination in the network nodes. Random linear network coding (RLNC) does not require coordination between network nodes so that complex routing and scheduling algorithms can be avoided. Similarly to fountain codes, network coding increases the packet diversity in the network, with a larger overhead however. When the receiver gathers enough innovative packets on multiple transmission paths, it can recover the original information. Network coding is used in various fields like content distribution [9, 10], distributed storage [11] or data dissemination [12]. Network coding can even be combined with fountain codes for efficient systems that take benefit from both source and path diversity [13].

The aim of this book chapter is to present coding techniques that permit to exploit the source and path diversity in overlay networks for efficient media streaming algorithms. First, we provide a brief overview of fountain codes with an emphasis on the encoding and decoding procedures in Section 2. We then show how these codes can be used in distributed streaming solutions with multiple sources. In Section 3, we provide an overview of network coding and its application in streaming systems. We show how network coding permits to exploit path diversity in overlay networks. We also show how network coding can be adapted to support different classes of packets in media streaming. Finally, we describe in Section 4 some hybrid schemes that exploit the advantages from both fountain codes and network codes and appear to be quite appropriate for low-cost real-time multimedia streaming.

## 2 Fountain Codes in Distributed Streaming

### 2.1 Fountain Codes

Fountain codes [3] have interesting properties for adaptive streaming, possibly from multiple sources. They provide rate adaptivity and permit to avoid the need for coordination between the senders. In this section, we first describe the principles of fountain codes and present the main families of fountain codes. We later describe their application to streaming application with a special emphasis on distributed streaming scenarios and unequal error protection algorithms.

The fountain codes are beneficial for multimedia applications that undergo strict deadlines (short playback delays). Some implementations even offer linear encoding and decoding times in contrast to other channel codes such as the Reed-Solomon codes, which have quadratic decoding complexity that grows with the block size. Fountain codes have only probabilistic guarantees for successful decoding and incur a small performance penalty  $\phi$  which means that a client should receive  $(1 + \phi) \cdot k$  packets to decode successfully the  $k$  source packets. Fountain systems eliminate the need for retransmission (ARQ) mechanisms because of their rateless property that permits to easily adapt the coding rate since

a large number of encoded symbols can be generated from the same subset of source packets. Obviously, they utilize the available resources more efficiently as acknowledgment messages (ACK) often worsen congestion problems.

The fountain codes have an implicit structure that depends on the degree distribution function

$$\Omega(x) = \sum_{i=1}^n \Omega_i \cdot x^i,$$

which determines the number of symbols (degree) that should be combined for generating an output symbol. The parameter  $n$  is the maximal symbol degree. The rateless encoded symbols are generated by randomly combining (XOR-ing) the source packets. The number of combined symbols is determined by randomly sampling the degree distribution function  $\Omega(x)$ .

Let's  $x_i$  denote the  $i$ th source symbol and  $y_j$  the  $j$ th transmitted symbol. Then, it holds

$$y_j = \sum_{l=1}^{d_j} \bigoplus x_l$$

where  $d_j$  is the degree of the  $j$ th symbol and  $\sum \bigoplus$  is the bitwise XOR operation. A small header called ESI is appended to each packet. The ESI is usually the seed of the pseudorandom generator used for generating the encoded symbol. Interestingly, Fountain codes are universal and thus near-optimal for any packet erasure channel. Their universal property is attributed to their rateless property that permits to generate large sets of encoded packets with only a small overhead penalty. Note that Fountain codes often need a termination message for signifying the reception of a full rank set of symbols. This mechanism saves significant amounts of bandwidth as otherwise the senders should assume a predetermined transmission rate which may be inaccurate as network conditions generally vary in time. A survey of fountain codes is given in [14].

## LT codes

Among the most popular fountain codes, we find the codes based on the Luby Transform (LT) [15]. LT codes are sparse random linear codes with nearly linear decoding time. Their encoding consists in random combinations of the source symbols, while their decoding can be done either by belief propagation or Gaussian elimination. The encoding and decoding procedures are summarized as follows:

---

### Procedure 1. LT encoding

---

- 1: Choose randomly the degree  $d$  of the LT encoded symbol by sampling  $\Omega(x)$ .
  - 2: Choose uniformly  $d$  distinct symbols.
  - 3: Combine the symbols by XOR-ing the selected symbols.
-

**Procedure 2.** LT decoding

---

```

1: while  $\mathcal{S} \neq \emptyset$  do
2:   Select a symbol  $x$  from  $\mathcal{S}$ .
3:   if  $x \notin \mathcal{L}$  then
4:     Include  $x$  in  $\mathcal{L}$ .
5:     XOR  $x$  with its neighbouring symbols  $\notin \mathcal{L}$ .
6:   end if
7: end while

```

---

where  $\mathcal{S}$  in Procedures 1 and 2 denotes the set of symbols in the ripple and  $\mathcal{L}$  the set of the recovered symbols. The ripple is the number of covered input symbols (i.e., symbols that have a unique neighbor in their Tanner graph representation after branches elimination in decoding) that have not been yet processed.

The LT codes employ a degree distribution function known as ideal soliton distribution (ISD) which is given by

$$\rho(d) = \begin{cases} \frac{1}{k}, & \text{for } d = 1 \\ \frac{1}{d \cdot (d-1)}, & \text{for } d > 1 \end{cases}$$

Although ISD performs well in expectation, in practice it is fragile to transmission errors because the ripple often stays empty. Ideally, the ISD ripple contains only one symbol at each decoding step. To enhance ISD's robustness to errors, it is slightly modified. The modified distribution is called robust soliton distribution (RSD)  $\mu(d)$  and it is given by

$$\mu(d) = \frac{\rho(d) + \tau(d)}{\sum_d \rho(d) + \tau(d)} \quad (1)$$

where

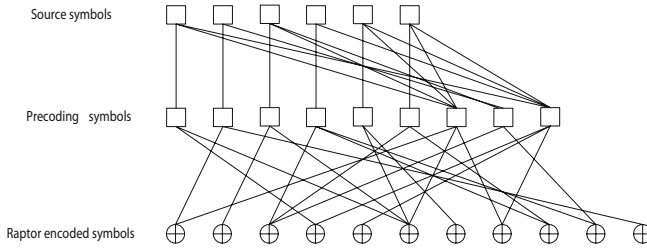
$$\tau(d) = \begin{cases} \frac{S}{d \cdot k}, & d = 1, 2, \dots, \frac{k}{S} - 1 \\ \frac{S}{k} \ln\left(\frac{S}{\delta}\right), & d = \frac{k}{S} \\ 0, & d = \frac{k}{S} + 1, \dots, k \end{cases} \quad (2)$$

The parameter  $\delta$  controls the size of ripple and  $S$  is the average number of symbols of degree one, *i.e.* the number of recovered symbols that have not been processed yet.  $S$  is finally defined as

$$S = c \cdot \sqrt{k} \cdot \ln\left(\frac{S}{\delta}\right)$$

## Raptor codes

Raptor codes [16] are the most successful rateless codes because they have an overhead that asymptotically tends to zero for very large codeblocks. For typical multimedia communication scenarios, the Raptor codes (short codeblocks are



**Fig. 1.** Encoding of non-systematic Raptor codes. The source symbols are first pre-coded and successively LT encoded.

assumed) can recover the data with high probability if they receive a set of symbols slightly larger than the set of source symbols *i.e.* one or two symbols are enough. Their encoding consists of two steps: first, the data are encoded by codes like LDPC, LDGM or Tornado and then the encoded data are fed in the LT encoder. The first step is known as pre-coding and it allows the employment of weakened LT codes, *i.e.* LT codes with very sparse parity-check matrices. Due to the sparse LT codes generator matrices, linear encoding and decoding times are achieved. The pre-coding step makes the decoding procedure more robust to erasures as the errors remaining after LT decoding can be corrected by the pre-coder. Raptor encoding is schematically depicted in Fig. 1. Their degree distribution is a modified ISD distribution

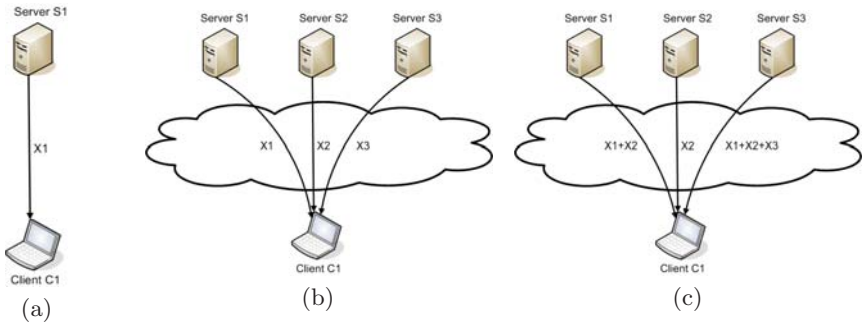
$$\Omega_R(x) = \frac{1}{\mu + 1} \cdot \left( \mu \cdot x + \sum_{i=2}^D \frac{x^i}{i \cdot (i - 1)} + \frac{x^{D+1}}{D} \right) \quad (3)$$

where  $\mu = (\epsilon/2) + (\epsilon/2)^2$ ,  $D = \lceil 4 \cdot (1 + \epsilon)/\epsilon \rceil$ , and  $\epsilon$  is a parameter that drives the decoding failure probability.

Raptor decoding is usually performed by Gaussian elimination of an equation system  $\mathbf{A}$  that is constructed from the received symbols. An efficient implementation of Raptor codes has been presented in 3GPP standard [17]. The pre-coding of 3GPP codes consists of LDPC codes followed by Half codes, which have dense parity check matrices. The Raptor codes equations system is of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{G}_{LDPC} & I_S & O_{S \times H} \\ & \mathbf{G}_{Half} & I_H \\ & & \mathbf{G}'_{LT} \end{bmatrix} \quad (4)$$

where  $\mathbf{G}_{LDPC}$  and  $\mathbf{G}_{Half}$  are respectively the generator matrices of LDPC and Half codes, while  $\mathbf{G}'_{LT}$  corresponds to the generator matrix of the received symbols.  $I_S$  and  $I_H$  are unitary matrices with size  $S \times S$  and  $H \times H$  where  $S$  and  $H$  denote the number of LDPC and Half codes constraints. Finally,  $O_{S \times H}$  is a zero matrix of size  $S \times H$ . It is worth noting that LDPC and Half codes constraints should not be transmitted as they can be reproduced if the clients are aware of the number of source symbols  $k$ . 3GPP codes provide a fast algorithm for solving these equations systems, which consists in a variant of classical Gaussian elimination process.



**Fig. 2.** (a) Direct server-client communication, (b) multiple servers-single client communication with scheduling and (c) multiple servers-single client communication with coding at the servers.

### Other Fountain codes

A few other Fountain codes have been recently presented for specific communication scenarios. Specifically, Growth codes [18] improve the poor performance of Fountain codes when less symbols than the number of source symbols are received by a client. For such cases, most Fountain codes like common channel coding schemes cannot retrieve any data segment. Systematic channel codes can recover more data, but their performance is driven by the transmission of uncoded data. However, the systematic codes are quite restrictive for many applications due to their low symbol diversity. On the other hand, Growth codes offer high symbol diversity while they achieve intermediate performance by means of the recovery of more data symbols compared to systematic codes. This performance comes with an increased overhead, which can still be affordable in catastrophic scenarios and in emergency situations.

Shifted codes [19] are variants of LT codes that outperform original LT codes when servers are aware about the number of packets that have been already decoded. They are useful for data streaming over wireless sensor networks where a feedback about the number of decoded symbols can be provided. Due to this knowledge, the LT codes distribution can be shifted towards higher degrees. Symbols of higher degrees are typically sent in order to be useful for more clients. Similarly, reconfigurable codes [20] have been proposed for wireless channels. They can be considered as LT codes with time varying degree distribution. Reconfigurable codes adapt their distribution based on a one bit ACK message signaled from clients to servers.

## 2.2 Streaming Systems with Fountain Codes

The traditional server-client paradigm (point-to-point communication) depicted in Fig. 2(a) typically represents a scenario where one client requests data from a single server that dedicates part of its resources to serve this receiver. The paradigm is replaced by a new model in overlay networks, where a client can

request media streams from several servers simultaneously, as illustrated in Fig. 2(b). In this way, the throughput and the error robustness of the system are increased. However, this often necessitates the employment of complex routing and scheduling protocols to minimize the probability of the reception of several identical packets, which represents a waste of bandwidth resources. Fountain codes permit to solve this limitation due to their rateless property where the multiple packet reception probability becomes very low. Fountain codes are applied at the multiple servers that send data simultaneously to the same client, as illustrated in Fig. 2(c). The receiver only needs to gather a sufficient number of different packets in order to decode the media stream. Strict coordination between servers is not required anymore in this case. The importance of the encoded packets becomes quite homogeneous, hence the media delivery is facilitated.

The interesting properties of the Fountain codes have not escaped from the attention of the media streaming community. Many researchers have adopted them in recent multimedia streaming systems for broadcast, one-to-one and many-to-one transmission scenarios. Multimedia broadcast and multicast service (MBMS) delivery over UMTS systems has been investigated in [21]. This system uses Turbo codes at physical layer to cope with bit errors and Raptor codes [16] at the application layer to deal with packet erasures. The Raptor codes permit the system to alleviate the need for coordination and to benefit from inherent network diversity while keeping the complexity low. It becomes clear that the careful balancing of available resources between application and physical layer boosts system's performance. This technique further lowers the power consumption as the error protection can be lighter. Higher packet loss rates can be afforded at the application layer, which can be successively corrected by Raptor codes. MBMS systems based on 3GPP Raptor codes [4] benefit from the partial recovery of packets corrupted by errors at the body of a packet as well as at the overhead. Their advanced performance comes with a slight increment of the computational cost. When a client receives an insufficient number of symbols for successful decoding, two post repair algorithms are used: one that finds the minimum set of source packets to be requested and another that determines the minimum number of consecutive packets to be requested. Transmission of Raptor encoded scalable videos over Mobile Ad Hoc Networks (MANET) has been proposed in [22]. Raptor codes are used to improve the obtained video quality and allow better adaptation to the rapid changing network conditions and network dynamics. A heuristic distributed rate allocation algorithm is also presented, which fairly allocates the available resources to various clients requesting various videos.

Unicast video streaming over packet erasures channel is further examined in [23]. The video is Raptor encoded to combat packet losses. A reliable feedback channel is used for sending a signal that informs the server that enough packets have reached the client. After the reception of such a message the server stops the transmission process. However, in the meanwhile the server continues to stream packets to the client, which wastes valuable resources. To reduce the wasted bandwidth the packet loss probability distribution is taken into account and

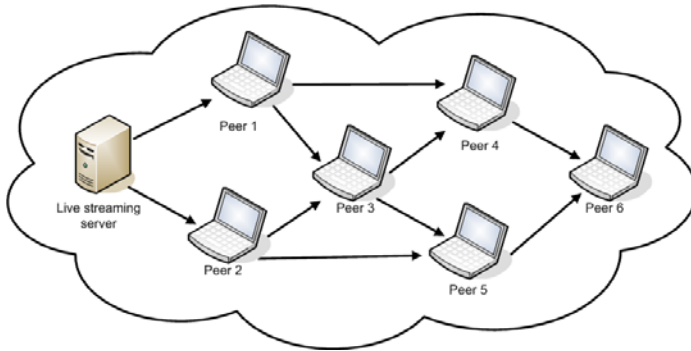


the servers adaptively change the transmission rate starting from an optimistic guess for the channel conditions. The server waits for a given amount of time to receive a message for terminating the transmission process, if the message does not arrive upon the deadline; the transmission rate is repeatedly increased to meet the real channel conditions. The optimal transmission strategies are determined beforehand by a greedy algorithm. However, [23] makes several unrealistic assumptions such as the employed codes have a fixed overhead, the pdf of the channel loss rate is known and the communication is performed through a reliable channel by a single symbol message. These shortcomings are addressed in [24] where the channel conditions and thus the pdf are derived from real time measurements. Raptor codes are replaced by LT codes for utilizing the low cost belief propagation decoder. Moreover, it is assumed that the communication over the feedback channel may face delays. The results show that the on-the-fly update of the channel statistics incurs only a small performance penalty compared to an ideal situation.

Fountain codes have also been used in many-to-one communication scenarios. In [5], Raptor codes are employed to avoid coordination among several servers streaming scalable video to a single client. Independent packet losses are assumed between any pair of server-client. Each video layer of each Group of Pictures (GOP) is independently encoded with Raptor codes. Optimal sending strategies as well as close to optimal heuristics that take into account the importance of each data layer are defined. Later, in [25], correlated packet losses between several pairs of servers-clients are considered and the optimal coding strategies are analyzed. Block-based Raptor codes [16] are also used for video communication between Bluetooth nodes in [26] where the channels are modeled as two-state Markov chains whose states represent the reception or loss of a packet. After a packet erasure some additional Raptor encoded blocks are incorporated into the payload of the next Raptor packets. This leads to low energy consumption compared to that of typical Bluetooth FEC schemes. Furthermore, the added overhead due to the block-based approach is shown to stay low. Finally, Raptor codes have been shown to be appropriate in the streaming of multiview videos in [27]. The transmitted streams consist of large number of packets. With good models for predicting Raptor codes performance, an efficient rate-distortion optimized encoding is shown to lead to a good exploitation of the available resources with good video quality.

### 2.3 UEP and Rateless Codes

Unequal data protection (UEP) of media data is critical in applications with strict decoding deadlines as well as large differences in the importance of the media packets for the reconstruction quality. Fountain codes can be modified to provide unequal error protection of media packets. For example, the LT codes can be re-designed so that they involve more important symbols in more encoding operations (more symbols) [28]. Alternatively, expanding window fountain codes (EWF) [29] have been proposed to protect unequally the data without modifying LT encoding. Specifically, the EWF codes encode successively several expanded



**Fig. 3.** Streaming with path diversity. 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 processing in order to increase the throughput and the robustness of the system.

windows of source symbols with LT codes. These EWF codes have been applied to layered multimedia in [30]. In [31], the LT encoding procedure is modified such as shorter coding paths are assigned to the more important symbols. The decoding probability of these symbols is thus increased. Finally, in [32], the LT codes distribution remains unaltered but the LT encoded symbols of degree one and two are not uniform combinations of symbols from all classes. Therefore, the most important classes have a higher probability to be sent as degree one or combined in order to generate symbols of degree two. This process does not harm the overall performance of LT codes, but rather increases the probability for the most important symbols to be correctly decoded at the receiver.

### 3 Network Coding in Multipath Streaming

#### 3.1 In-Network Processing

Streaming in overlay networks introduces a new transmission model, where the network nodes can be used advantageously for improved performance, as illustrated in Fig. 3. Processing in the network nodes and not only at the ends of the transmission channels permits a better adaptation to network dynamics and a better exploitation of the diversity of the network. We describe in this section the benefits of network coding and its application to practical streaming applications, with an emphasis on unequal error protection strategies that are specific to media streaming problems.

One of the first attempts to realize some type of coding in nodes of an overlay network is presented in [33, 34]. The nodes are organized in multicast trees. Some of them implement Reed-Solomon (RS) channel coding operations to increase the robustness of the system. These are called network-embedded FEC (NEF) nodes and perform RS decoding on the packets they receive. They encode them

again with RS codes, before passing them to the children peers. NEF nodes permit to increase the resiliency of the system, while avoiding waste of resources with a 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 which results into a high video quality.

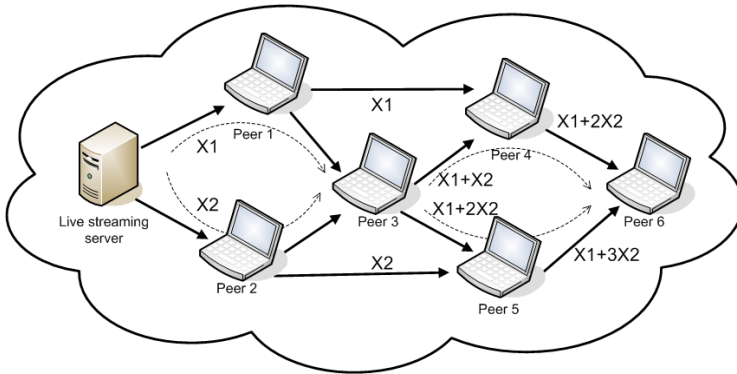
Similarly, decoding and encoding based on fountain codes is performed in the network nodes in [7]. The LT codes [15] are used in this work since they perform close to perfect codes and eliminate the need for reconciliation among network peers and packet scheduling. The intermediate network nodes wait 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 allows 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 and recoding in the network nodes are difficult to implement in large networks, since they introduce quite important delays. However, when the network provides path diversity, the packet diversity in the system can be maintained by effective network coding algorithms. Network coding basically consists in combining packets in the network nodes. The encoded packets are then forwarded to the next nodes. When the receiver eventually gathers enough innovative packets, it can invert the full coding system and recover the original data. Network coding permits to increase the network throughput and the error resiliency in streaming systems with path diversity. The benefits of network coding methods are illustrated in a simple streaming scenario in the peer-to-peer network depicted in Fig. 4 where all links have capacity of one packet per time slot. When all intermediate nodes follow the store and forward approach, there is no guarantee that Peer 6 receives packets  $X_1$  and  $X_2$  because of the bottlenecks at Peers 4 and 5. If, however, Peers 3, 4 and 5 are network coding points, the probability of receiving an undecodable set of packets by Peer 6 is approximately zero when coding operations are in a large galois field.

We refer the interested readers to [35, 36, 37, 38, 39, 40] for detailed descriptions of network coding theory and principles. The theoretical works in network coding have made apparent that network coding can be beneficial for various applications like streaming or distributed storage. We focus, in the rest of this section, on the application of network coding to practical multipath streaming scenarios.

### 3.2 Practical Network Coding

Early network coding schemes [41, 42] necessitate the use of computationally complex algorithms for defining the coding coefficients. In addition, the network



**Fig. 4.** Multimedia streaming in peer-to-peer networks with network coding techniques.

coding design methods generally assume that the servers have a full knowledge of the network topology, which is quite unrealistic in large scale networks. In order to simplify the design of network codes and improve the robustness in network dynamics, several works [43, 44, 45] have proposed to implement network coding with a random selection of the coefficients. If the coefficients are chosen in a sufficiently large Galois Field  $\text{GF}(q)$ , random linear network coding can achieve the multicast capacity with a probability that asymptotically approaches one for long code-lengths (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 [44]. 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. RLNC provides therefore a low complexity alternative towards the design of practical algorithms, since it permits to relax the requirements on 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 RLNC, the authors in [46] 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, which 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 thus 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 with probability 99.6% decodable when the computations are performed on  $\text{GF}(2^{16})$ . Smaller fields like  $\text{GF}(2^8)$  are sufficient in practice [46].

Even if network coding a priori permits to combine any packet in the network nodes, the coding choices clearly have an influence on the end-to-end delay of the transmission system. In order to cope with the buffering delay problem, the authors in [46] 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 transmit the encoded packet. As coding becomes constrained to the packets of the same generation, the resulting delay is limited. However, the redundancy might become rapidly large since the number of packets in a generation stays limited. 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 [46]. The clients finally implement a progressive decoding strategy by Gaussian elimination typically. 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.

### 3.3 Network Coding in Multipath Media Streaming

Due to its benefits in terms of increased throughput and robustness, practical network coding has been applied to diverse multipath streaming applications. It first finds a perfect application in peer-to-peer multicast applications. Such applications 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. Live multicast streaming can therefore be implemented by transmitting the media packets from the server to all the clients via other peers that are grouped in an overlay or ad hoc configuration. 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. It permits to achieve a sustainable throughput with reduced transmission delays.

One of the first works that has studied the performance of network coding in peer-to-peer (p2p) streaming has been proposed in [47]. 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 generations proposed in [46]. 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 [47], the same authors redesign the peer-to-peer streaming algorithm and propose the  $R^2$  architecture in [48]. 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 is possible, otherwise they are transmitted separately. The frequency of this information exchange 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 [49]. 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 [47]. 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 [50]. The overlay construction imposes that all the peers have the same number of parents nodes, which are the nodes that forward 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 increasing throughput and improving system robustness. In order to avoid limiting the performance of the system by the smallest capacity peers, one could organize the overlay into several layered meshes. Heterogeneous receivers can then subscribe to one of several meshes, depending on their capacities [51]. The data are similarly organized into layers, and network coding is performed on packets of the

same layers. The practical network coding scheme of [46] 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 it has to subscribe to. The network throughput is finally increased by network coding combined with appropriate peer organization. In the same spirit, the work in [52] 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 [53].

### 3.4 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) [54] principles has been initially proposed in [46], where data of high importance receive a high level of error protection by a proper arrangement of the data blocks in the encoding matrix. Unequal error protection is also proposed in [55]. The PET algorithm is however replaced by a MD-FEC scheme [56], 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 [57]. Data are segmented and interleaved in the coding matrix, in such a way that base layer typically receives more redundancy bits than the enhancement layers. Classical network coding is then performed on packets within the same generation. The proposed scheme is shown to outperform other solutions based on either routing or routing with replication policies.

The above methods face several problems that limit their application in real settings. Specifically, the MD-FEC based schemes often overprotect the information since they require the end-to-end network statistics and full knowledge of network topology to find the optimal protection. The application of MD-FEC based network coding methods in a distributed manner is possible only when the unequal amounts of protection are determined in hop-by-hop basis. Specifically in such a setting, each peer node first decodes the received network coded packets and then successively re-encodes the recovered video packets. Unfortunately, this procedure is computationally expensive and pronounces latency issues. Furthermore, it is questionable whether MD-FEC based NC methods are appropriate for overlay networks since ideally all received packets by a peer should be generated by the same MD-FEC matrix. Otherwise significant rate is lost as each node can decode only when it has received a decodable set of packets from the same MD-FEC matrix. Without such coordination the nodes receive packets from multiple MD-FEC codes that cannot be jointly decoded.



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 [58] 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. Prioritized coding can also be achieved by modifying the network coding operations in practical streaming applications. For example, the work in [59] addresses the problem of streaming H.264/AVC encoded video content in MBMS networks. 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.

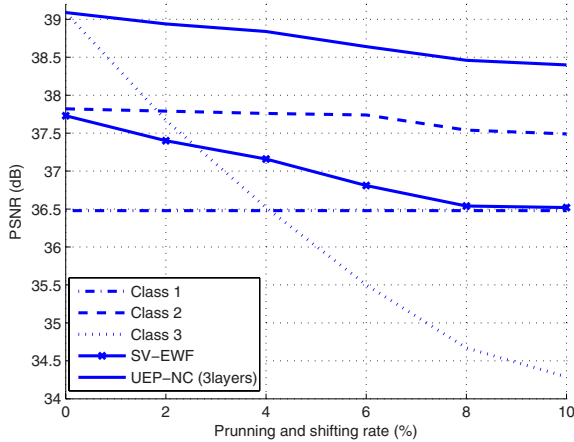
However, the distributed method of [59] imposes frequent packet exchanges to determine the optimal coding operations. Obviously, this increases significant the computational cost. Moreover, the application of this distributed algorithm for video streaming in overlay mesh networks is not straightforward as for MBMS broadcast networks the extended communication between peer nodes is reasonable, however, in mesh networks it may worsen congestion problems and lead to an explosion of erased packets. Techniques based on UEP rateless codes like EWF codes [29] are inefficient for scalable video multicasting due to network dynamicity which results in poor estimation of the expected EWF codes overhead when central estimation is considered. EWF codes can work efficiently for large codeblocks which is not typical for video streaming where strict timing constraints exist. Smaller codeblocks penalize significant EWF codes performance as they are associated with large overheads.

Differently from the works described thus far a receiver-driven network coding technique for prioritized video transmission in overlay networks is presented in [60]. The specific importance of media packets is taken into account to prioritize the delivery of the most important packets. The overlay nodes perform RLNC on incoming video packets in order to improve the robustness to failures or erasures in the delivery process without the need for any centralized control. As media packets are grouped into classes of different importance, prioritized transmission is achieved by varying the number of packets from each class that are used in the network coding operations at a node. The mixing operations are not uniform across all packets arriving at a node, but instead packets with higher importance are involved in more coding operations. Thus, each peer is interested in selecting the vector of coefficients  $\mathbf{w} = [w_1, \dots, w_L]$  such that the network coding strategy employed by its parent nodes will maximize the peer's video quality.  $L$  stands for the number of video layers and  $w_i$  is a weighted coefficient determining the number of packets from class  $i$ <sup>1</sup>. An optimization problem solved by each node tries to determine the number of packets it should request from each class,

---

<sup>1</sup> Class  $i$  is a set of video packets from first  $i$  video layers and the combinations of these packets.





**Fig. 5.** Influence of the network regularity (pruning and shifting probabilities) of the proposed UEP scheme with baseline NC solutions and the *SV-EWF* scheme.

so that the total video distortion is minimized or alternatively the cumulative distortion reduction, as contributed by the requested packets, is maximized. This optimization problem is actually log-concave and can be solved by a low-cost greedy algorithm [60]. The algorithm starts from a pivotal distribution and then examines single packet exchanges between consecutive classes. This procedure is repeated till there is no beneficial exchange that further maximizes the distortion reduction.

The influence of the network topology on the performance of the above method termed *UEP-NC* is investigated in [60] and we illustrate here the performance of several UEP scheme based on network coding. The *UEP-NC* is evaluated for irregular networks generated following the process described in [61] where network link's are pruned and shifted according to two probabilities  $P_{pr}$  and  $P_{sf}$  respectively. The *UEP-NC* scheme of [60] is compared with basic network coding schemes called *Class- $i$*  and a UEP scheme based on *EFWF* codes (*SV-EWF*). *Class- $i$*  scheme assumes video packets from first  $i$  video layers that are uniformly combined through RNC. Three video layers are assumed and the coding operations are restricted into GOPs to deal with timing issues and the frame rate is set to 30 *fps*. The pruning and the shifting probabilities are set to be equal, however their values vary in the range  $[0, \dots, 10]$ %. For each value of  $P_{pr}$  and  $P_{sf}$ , a seven-stage irregular network was generated. The packet loss rate on a link is fixed to 5% and the link capacity value equals to 360 kbps. The results of the evaluation are illustrated in Fig. 5. It can be seen that, when  $P_{pr}$  and  $P_{sf}$  are low then *Class-3* scheme performs equally well with *UEP-NC* as both are able to exploit the sufficient network resources. However, when  $P_{pr}$  and  $P_{sf}$  increase, the performance of *UEP-NC* degrades gracefully while *Class-3* scheme exhibits a significantly lower performance. The other two basic network coding schemes seems also to be robust to network variations, but are limited by the smaller

number of video layers that they consider. The performance of *SV-EWF* for low pruning and shifting probabilities is comparable to that of *Class-2* scheme, however as network topologies become more random its performance approaches that of *Class-1* scheme. This is because bandwidth variations force *SV-EWF* to generate symbols from the first class only as the available bandwidth is insufficient to transmit data from other classes. Furthermore, the second class can often not be retrieved and the source symbols diversity in the network degrades quickly. For detailed comparisons interested readers are referred to [60].

## 4 Streaming with Rateless Coding and Network Coding

The practical network coding scheme presented in [46] has proposed efficient solutions to cope with inefficiencies of early network coding systems for deployment in streaming applications. However, there are still many issues that should be resolved before network coding can be applied to real time communication. One of the major drawbacks of today's network coding schemes is their relative high decoding computational cost as a dense equation system should be solved by Gaussian elimination. Another limiting factor is the significant coding overhead that grows linearly with the generation size. The above issues have been addressed in [13] where network coding is combined with Raptor codes [16] to take benefit from linear encoding and decoding times of the latter. Packets are encoded with non-systematic Raptor codes at the servers, as illustrated in Fig. 6. The network nodes then selectively combine packets when they have to compensate for packet losses and bandwidth variations. This scheme requires a smaller network coding header than that of typical network coding schemes. It is simply a concatenation of the combined Raptor coded packets headers. This saves significant amounts of rate which can be used for transmitting video streams of higher quality. Compared to implementations of fountain codes with in network processing [7], decoding operations in the network nodes are avoided

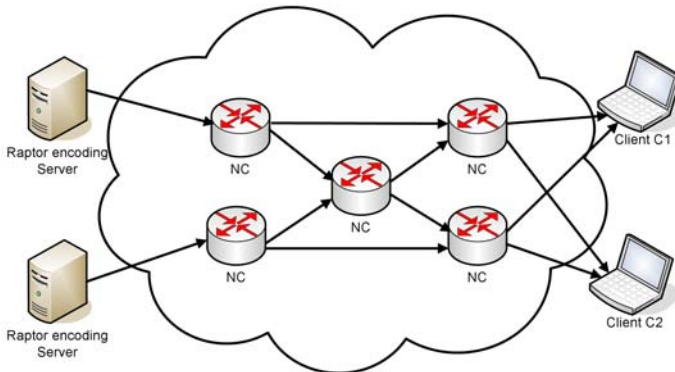
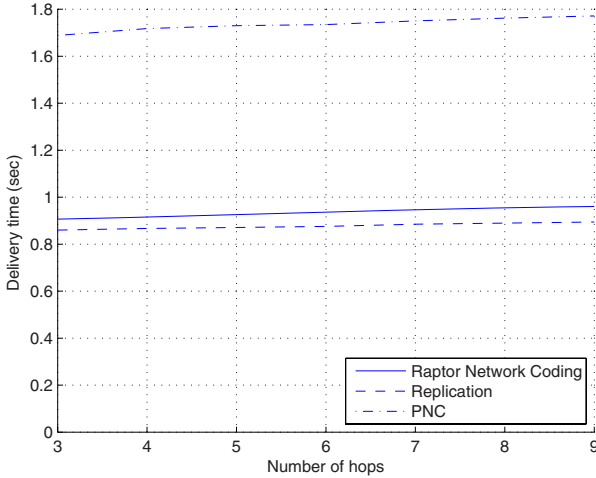


Fig. 6. Raptor network coding.



**Fig. 7.** Time delivery for regular networks with three nodes per hop with respect to the network size (number of hops).

in [13] and replaced by linear packets combinations. 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 both delay and complexity. The combination of Raptor coding and network coding permits to benefit from both source and path diversity in overlay networks, with controllable delay and complexity. We describe this algorithm in more details in the rest of this section, where we focus on the analysis of the delay and the coding strategies in Raptor network coding.

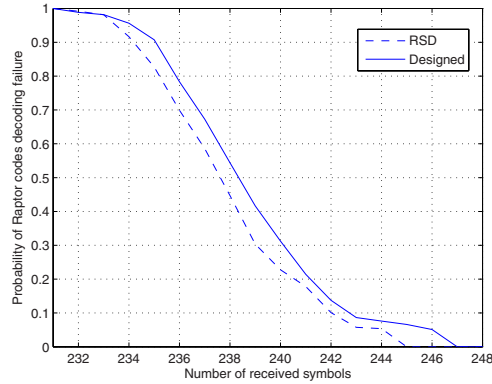
The Raptor based network coding system of [13] resembles RLNC based systems in that it performs simple linear operations with packets for generating network coded packets. The implicit coding structure of the Raptor network coding schemes is communicated as a header that is appended to each Raptor encoded packet. Raptor codes as RLNC systems are endowed with the rateless property that allows easy adaptation to bandwidth variations. The servers encode the video with non-systematic Raptor codes in order to provide initially high symbol diversity to the network, while the intermediate nodes perform encoding with the received packets. Specifically, the packets are XOR-ed before the peers forward them. Unlike other network coding systems, the peers do not replace the received packets with random combinations of them, but they perform selectively network coding to cope with bandwidth variations and packet erasures. This selective network coding keeps high the symbol diversity as all symbols are not involved in every coding operation. This approach also reduces the complexity since the equation system constructed at clients with the received packets is less dense.

The Raptor network coding scheme does not need large buffers. The packets are combined as soon as they arrive to the peer node, hence limiting end-to-end delays. Each node maintains a list with the transmitted packets in order to avoid transmission of a network coded packet over the same link with the packets combined to generate this packet. Then, these packets are removed from buffer to avoid future re-combinations of the same packets. This strategy assists the system to maintain a high symbol diversity in the network. The delivery times of the Raptor network coding scheme with respect to various network sizes are presented in Fig. 7. This scheme is compared with the practical network coding scheme of [46] and a scheme that applies Raptor codes in an end-to-end fashion. The intermediate nodes just replicate randomly some of the received packets to replace missing packets. Regular mesh networks, where the nodes are organized into different transmission stages that depend on the hop distance to the sources are examined. Three nodes per stage are considered. The link's capacity is set to 400 kbps. The packet erasure rate is fixed to 5% for each network link and the buffer size equal to 32 packets. As it can be seen from Fig. 7 results in larger delivery times as each node forwards packets when the buffer is full. It is worth noting that there are not significant performance variations as the network size grows. The advanced performance of the Raptor network coding scheme is attributed to fact that network coded packets are generated upon request. Since the packet loss rate is rather low and the buffer size sufficiently large, the probability of not finding orthogonal packets is also very low.

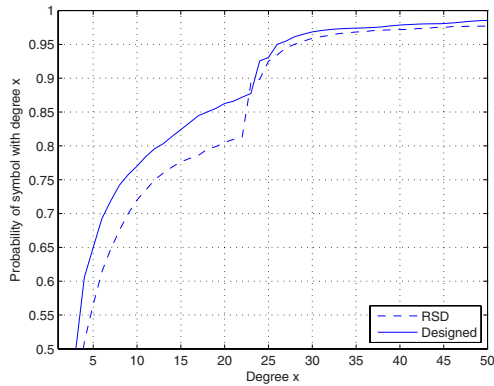
Although the computational cost of the Raptor network coding scheme is relative low, it increases with the successive coding operations. Specifically, due to losses and network coding in successive stages, the distribution of the degree of the symbols received at the decoder is altered and might differ significantly from the degree distribution used in the source Raptor encoders. In particular, the generator matrix becomes denser due to combinations of Raptor symbols by network coding, which leads to increased decoding complexity. Apparently, the received symbols by a client do not follow anymore the original Raptor codes degree distribution<sup>2</sup>. In order to guarantee linear decoding times, in [62] the re-design of the degree distribution of the Raptor codes  $\Omega(x)$  used at the senders can be studied. The optimization goal is the design of a  $\Omega(x)$  that grows as packets travel through the network towards clients and ends up to a distribution similar to that of well performing Raptor codes. The design optimization problem can be formulated as a geometric programming (GP) [63] problem. When the servers are aware of the network statistics (network losses and topology), the optimal source degree distribution can be determined so that linear decoding times are preserved. This method is more generic than [64] where the design of LT codes degree distribution for simple relay topologies has been investigated and the RSD [15] is decomposed into two component distributions prior to deconvolution of RSD. Although the algorithm in [64] ensures that clients receive symbols whose degree distribution is close to RSD, it imposes rather complicated encoding rules.

---

<sup>2</sup> The degree stands for the number of symbols combined for generating an Raptor coded packet.



(a)

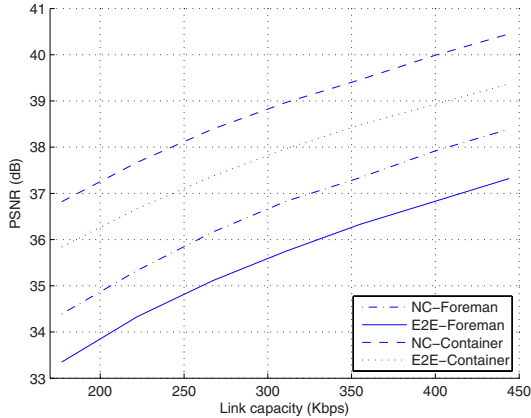


(b)

**Fig. 8.** Performance evaluation of  $(231, 249)$  Raptor codes for regular network topologies with three intermediate nodes per stage. The compared network coding schemes employ: a 3GPP variant with RSD distribution and the low-complexity codes. (a) Raptor decoding probabilities at clients with respect to the number of received symbols. (b) Cumulative distribution of symbols degree received by each client.

The extension of this method to complex network topologies is furthermore not trivial.

The Raptor decoding probabilities with respect to the number of received packets for nine-stage regular network topologies with three nodes per stage is shown in Fig. 8(a). It is obvious that the designed distribution performs slightly worse, but it remains close to the performance of a scheme employing 3GPP codes with RSD distribution. This performance degradation is due to the inefficiency of the design method to preserve the exact value of the spike which guarantees the convergence of the decoding process. However, when the methods are compared in terms of the cumulative degree distribution function (cdf) as



**Fig. 9.** PSNR performance evaluation of Raptor network coding (NC) and end-to-end Raptor coding (E2E), for regular topologies with six coding stages and 3 nodes per stage. The PSNR is measured as a function of the average link bandwidth in the network, the packet loss ratio is 5 %.

depicted in Fig. 8(b), it is clear that the proposed codes outperform the 3GPP variant with RSD as they correspond to sparser generator matrices and thus decoding is faster.

Finally, the optimal determination of source and channel rate allocation for the Raptor network coding system is investigated in [65]. The optimization algorithm is centralized and seeks for the optimal allocation based on the network statistics gathered backwards. The algorithm estimates the rank decrement of the equation system built on the received network coded packets. The centralized optimization algorithm is formulated as a *minmax* problem. Therefore, the algorithm seeks for the channel rate allocation which minimizes the maximal distortion among the clients. The optimization is performed under the constraints that the number of packets sent over a link can not surpass link capacity.

We illustrate the performance of the Raptor network coding scheme with efficient rate allocation in in Fig. 9. The average quality is given as a function of the average link bandwidth is for the transmission of “Foreman” and “Container” CIF sequences over a regular topology with six encoding stages between servers and decoders and three nodes per coding stage. The packet loss ratio is set to 5% while the link bandwidth varies between 170 kbps and 450 kbps. The Raptor network coding scheme is compared with a scheme that applies Raptor codes as a form of end-to-end error protection. The Raptor network coding algorithm performs better than the end-to-end solution due to its improved adaptivity that permits a better exploitation of the path diversity. We also observe that the performance gap remains unaltered as the link capacity increases. This is due to the fact that the schemes are not very sensitive to bandwidth variations, but rather to the overall packet loss rate. A more detailed analysis of Raptor

network video coding system performance for a variety of transmission scenarios can be found in [65].

## 5 Conclusions

In this book chapter, we have shown that fountain codes and network coding algorithms can be interesting solutions for building efficient streaming systems in networks with diversity. These techniques are similar in some concepts as they are based on random operations, do not target at a predetermined rate and try to maintain a high packet diversity in the streaming system. Although both types of algorithms have appealing characteristics, their use is not trivial for multimedia applications with strict deadlines. For example, rateless codes may still require some coordination especially in large scale networks, while network codes use computationally expensive decoding algorithms such as gaussian elimination. Hybrid schemes combining network coding with rateless codes enable the use of low-cost decoding algorithms without imposing any requirements for coordination of the servers. Hence, they probably represent an interesting solutions for streaming in overlay networks with proper exploitation of source and path diversity.

## References

1. Zhang, X., Liu, J., Li, B., Yum, Y.S.P.: CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In: Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 3, pp. 2102–2111 (2005)
2. Chakareski, J., Frossard, P.: Adaptive systems for improved media streaming experience. *IEEE Communications Magazine* (2007)
3. Byers, J., Luby, M., Mitzenmacher, M., Rege, A.: A Digital Fountain Approach to Reliable Distribution of Bulk Data. In: Proc. ACM SIGCOMM 1998, Vancouver, BC, Canada, pp. 56–67 (1998)
4. Gasiba, T., Xu, W., Stockhammer, T.: Enhanced System Design for Download and Streaming Services Using Raptor Codes. *European Trans. on Telecommunications* 20(2), 159–173 (2009)
5. Wagner, J.P., Chakareski, J., Frossard, P.: Streaming of Scalable Video from Multiple Servers using Rateless Codes. In: Proc. of IEEE Conf. on Multimedia and Expo, ICME 2006, Toronto, Ontario, Canada, pp. 1501–1504 (2006)
6. Aly, S.A., Kong, Z., Soljanin, E.: Raptor Codes Based Distributed Storage Algorithms for Wireless Sensor Networks. In: Proc. of IEEE Int. Symp. on Information Theory, ISIT 2008, Toronto, Ontario, Canada, pp. 2051–2055 (2008)
7. Wu, C., Li, B.: rStream: Resilient and Optimal Peer-to-Peer Streaming with Rateless Codes. *IEEE Trans. Parallel and Distributed Systems* 19(1), 77–92 (2008)
8. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network Information Flow. *IEEE Trans. Information Theory* 46(4), 1204–1216 (2000)
9. Gkantsidis, C., Miller, J., Rodriguez, P.: Comprehensive View of a Live Network Coding P2P system. In: Proc. ACM SIGCOMM/USENIX IMC 2006, Brasil (2006)

10. Gkantsidis, C., Rodriguez, P.R.: Network Coding for Large Scale Content Distribution. In: Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 4, pp. 2235–2245 (2005)
11. Dimakis, A.G., Godfrey, P.B., Wainwright, M.J., Ramchandran, K.: Network Coding for Distributed Storage Systems. In: 26th IEEE Int. Conf. on Computer Communications, ICC 2007, Anchorage, Alaska, USA (2007)
12. Deb, S., Médard, M., Choute, C.: On Random Network Coding Based Information Dissemination. In: IEEE Int. Symp. on Information Theory 2005, ISIT 2005, Adelaide, Australia, pp. 278–282 (2005)
13. Thomos, N., Frossard, P.: Raptor Network Video Coding. In: Proc. of the 1st ACM Int. Workshop on Mobile video (in conjunction with ACM MM 2007), Augsburg, Germany (2007)
14. Mitzenmacher, M.: Digital Fountains: A Survey and Look Forward. In: Proc. of the Information Theory Workshop, ITW 2004, San Antonio, Texas, USA, pp. 271–276 (2004)
15. Luby, M.: LT codes. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002), Vancouver, Canada, pp. 271–280 (2002)
16. Shokrollahi, A.: Raptor codes. *IEEE Trans. Information Theory* 52(6), 2551–2567 (2006)
17. 3GPP TS 26.346 V7.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs (2005)
18. Kamra, A., Feldman, J., Misra, V., Rubenstein, D.: Growth Codes: Maximizing Sensor Network Data Persistence. In: Proc. of ACM SIGCOMM 2006, Pisa, Italy (2006)
19. Agarwal, S., Hagedorn, A., Trachtenberg, A.: Adaptive Rateless Coding Under Partial Information. In: Proc. Information Theory and Applications Workshop, ITA 2008, San Diego, CA, USA, pp. 5–11 (2008)
20. Bonello, N., Zhange, R., Chen, S., Hanzo, L.: Reconfigurable Rateless Codes. In: Proc. VTC 2009-Spring, Barcelona, Spain (2009)
21. Luby, M., Gasiba, T., Stockhammer, T., Watson, M.: Reliable Multimedia Download Delivery in Cellular Broadcast Networks. *IEEE Trans. on Broadcasting* 53(1), 235–246 (2007)
22. Schierl, T., Johansen, S., Perki, A., Wiegand, T.: Rateless Scalable Video Coding for Overlay Multisource Streaming in MANETs. *Journal on Visual Communication and Image Representation* 19(8), 500–507 (2008)
23. Ahmad, S., Hamzaoui, R., Akaidi, M.A.: Robust Live Unicast Video Streaming with Rateless Codes. In: Proc. of 16th Int. Workshop on Packet Video, PV 2007, Lausanne, Switzerland, pp. 77–84 (2007)
24. Ahmad, S., Hamzaoui, R., Akaidi, M.A.: Practical Channel-adaptive Video Streaming with Fountain Codes. In: Proc. of IEEE Int. Conf. on Pervasive Computing and Applications, ICPCA 2008, Alexandria, Egypt (2008)
25. Wagner, J.P., Frossard, P.: Adaptive and Robust Media Streaming over Multiple Channels with Bursty Losses. In: Proc. European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland (2007)
26. Razavi, R., Fleury, M., Ghanbari, M.: Block-based Rateless Coding for Energy-efficient Video Streaming over Bluetooth. In: Proc. of IEEE Symp. on Computers and Communications, ISCC 2008, Marrakech, Morocco (2008)
27. Tan, A.S., Aksay, A., Akar, G.B., Arikan, E.: Rate-distortion Optimization for Stereoscopic Video Streaming with Unequal Error Protection. *EURASIP Journal on Advances in Signal Processing* 2009(632545) (2009)



28. Dimakis, A.G., Wang, J., Ramchandran, K.: Unequal Growth Codes: Intermediate Performance and Unequal Error Protection for Video Streaming. In: Proc. of Multimedia Signal Processing, MMSP 2007, Pisa, Italy, pp. 107–110 (2007)
29. Sejdinovic, D., Vukobratovic, D., Doufexi, A., Piechocki, R., Senk, V.: Expanding Window Fountain Codes for Unequal Error Protection. In: Proc. of 41st Annual Asilomar 2007 Conf. on Signals, Systems and Computers, Pacific Grove, CA, USA (2007)
30. Vukobratovic, D., Stankovic, V., Sejdinovic, D., Stankovic, L., Xiong, Z.: Scalable Video Multicast using Expanding Window Fountain Codes. *IEEE Trans. Multimedia* 11(6), 1094–1104 (2009)
31. Chang, S.K., Yang, K.C., Wang, J.S.: Unequal-Protected LT Code for Layered Video Streaming. In: Proc. of IEEE Int. Conf. on Communications, ICC 2008, Beijing, China, pp. 500–504 (2008)
32. Woo, S.S., Cheng, M.K.: Prioritized LT codes. In: Proc. of 42nd Annual Conf. on Information Sciences and Systems, CISS 2008, Princeton, NJ, USA, pp. 568–573 (2008)
33. Karande, S., Wu, M., Radha, H.: Network Embedded FEC (NEF) for Video Multicast in Presence of Packet Loss Correlation. In: Proc. IEEE Int. Conference on Image Processing, Genoa, Italy, vol. 1, pp. 173–176 (2005)
34. Wu, M., Karande, S., Radha, H.: Network Embedded FEC for Optimum Throughput of Multicast Packet Video. *EURASIP Journal on Applied Signal Processing* 20(8), 728–742 (2005)
35. Fragouli, C., Soljanin, E.: Network Coding Fundamentals. Source. In: Foundations and Trends in Networking archive. Now Publishers Inc., Boston (2007)
36. Fragouli, C., Soljanin, E.: Network Coding Applications. Source. In: Foundations and Trends in Networking archive. Now Publishers Inc., Boston (2007)
37. Yeung, R.W., Li, S.Y.R., Cai, N., Zhang, Z.: Network Coding Theory. In: Foundations and Trends in Communications and Information Theory. Now Publishers Inc., Boston (2005)
38. Yeung, R.W.: Information Theory and Network Coding. In: Information Technology: Transmission, Processing and Storage. Springer, Heidelberg (2008)
39. Ho, T., Lun, D.S.: Network Coding: An Introduction. Cambridge University Press, Cambridge (2008)
40. Thomos, N., Frossard, P.: Network Coding for Theory to Media Streaming. *Journal of Communications* 4(9) (2009)
41. Koetter, R., Médard, M.: An Algebraic Approach to Network Coding. *IEEE/ACM Trans. on Networking* 11(5), 782–795 (2003)
42. Jaggi, S., Sanders, P., Chou, P.A., Effros, M., Egnér, S., Jain, K., Tolhuizen, L.M.G.M.: Polynomial Time Algorithms for Multicast Network Code Construction. *IEEE Trans. Information Theory* 51(6), 1973–1982 (2005)
43. Ho, T., Medard, M., Shi, J., Effros, M., Karger, D.R.: On Randomized Network Coding. In: 41st Allerton Annual Conf. on Communication, Control, and Computing, Monticello, IL, USA (2003)
44. Ho, T., Koetter, R., Medard, M., Karger, D.R., Effros, M.: The Benefits of Coding Over Routing in a Randomized Setting. In: IEEE Int. Symp. on Information Theory, ISIT 2003, Kanagawa, Japan (2003)
45. Ho, T., Medard, M., Koetter, R., Karger, D.R., Effros, M., Jun, S., Leong, B.: A Random Linear Network Coding Approach to Multicast. *IEEE Trans. Information Theory* 52(10), 4413–4430 (2006)
46. Chou, P.A., Wu, Y., Jain, K.: Practical Network Coding. In: Proc. 41st Allerton Conf. on Communication Control and Computing, Monticell, IL, USA (2003)

47. Wang, M., Li, B.: Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming. In: IEEE INFOCO, Anchorage, Alaska (2007)
48. Wang, M., Li, B.:  $R^2$ : Random Rush with Random Network Coding in Live Peer-to-Peer Streaming. *IEEE Journal on Selected Areas in Communications* 25(9), 1655–1666 (2007)
49. Maymoukov, P., Harvey, N.J.A., Lun, D.: Methods for Efficient Network Coding. In: Proc. 44th Allerton Conf. on Communication, Control, and Computing, Monticello, IL, USA (2006)
50. Jain, K., Lovász, L., Chou, P.A.: Building Scalable and Robust Peer-to-Peer Overlay Networks for Broadcasting Using Network Coding. *Journal on Distributed Computing* 19(4), 301–311 (2006)
51. Zhao, J., Yang, F., Zhang, Q., Zhang, Z., Zhang, F.: LION: Layered Overlay Multicast With Network Coding. *IEEE Trans. Multimedia* 8(5), 1021–1032 (2006)
52. Liu, Y., Peng, Y., Dou, W., Guo, B.: Network Coding for Peer-to-Peer Live Media Streaming. In: Proc of the Fifth Int. Conf. Grid and Cooperative Computing, GCC 2006, Monticello, IL, USA, pp. 149–155 (2006)
53. Shao, M., Wu, X., Sarshar, N.: Rainbow Network Flow with Network Coding. In: Proc. of the fourth Workshop on Network Coding, Theory and Applications, NetCod 2008, Hong Kong, China, pp. 1–6 (2008)
54. Albanese, A., Bloemer, J., Edmonds, J., Luby, M., Sudan, M.: Priority Encoding Transmission. *IEEE Trans. Information Theory* 42, 1737–1744 (1996)
55. Ramasubramonian, A.K., Woods, J.W.: Video Multicast Using Network Coding. In: SPIE VCIP, San Jose, CA, USA (2009)
56. Puri, R., Ramchandran, K., Bharghavan, K.W.L.V.: Forward Error Correction (FEC) Codes Based Multiple Description Coding for Internet Video Streaming and Multicast. *Signal Processing: Image Communication* 16(8) (2001)
57. Wang, H., Xiao, S., Kuo, C.C.J.: Robust and Flexible Wireless Video Multicast with Network Coding. In: IEEE Global Telecommunications Conference, GLOBECOM 2007, Adelaide, Australia, pp. 2129–2133 (2007)
58. Lin, Y., Li, B., Liang, B.: Differentiated Data Persistence with Priority Random Linear Codes. In: 27th Int. Conf. on Distributed Computing Systems of Contents, ICDCS 2007, Toronto, ON, Canada, pp. 47–55 (2007)
59. Liu, X., Cheung, G., Chuah, C.N.: Structured Network Coding and Cooperative Local Peer-to-peer Repair for MBMS Video Streaming. In: IEEE Int. Workshop on Multimedia Signal Processing, MMSP 2008, Cairns, Queensland, Australia (2008)
60. Thomos, N., Chakareski, J., Frossard, P.: Randomized network coding for UEP video delivery in overlay networks. In: International Conference on Multimedia and Expo, ICME 2009, Cancun, Mexico (2009)
61. Thomos, N., Frossard, P.: Collaborative Video Streaming with Raptor Network Coding. In: Inter. Conf. on Multimedia and Expo, ICME 2008, Hannover, Germany, pp. 497–500 (2008)
62. Thomos, N., Frossard, P.: Degree Distribution Optimization in Raptor Network Coding. EPFL Technical report (2010)
63. Chiang, M.: *Geometric Programming for Communications Systems*. Now Publishers Inc., Boston (2005)
64. Puducheri, S., Klierer, J., Fuja, T.E.: The Design and Performance of Distributed LT codes. *IEEE Trans. Information Theory* 53(10), 3740–3754 (2007)
65. Thomos, N., Frossard, P.: Network Coding of Rateless Video in Streaming overlays EPFL Technical report LTS-REPORT-2009-011 (2009)