Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Retransmission in the network-coding-based packet network*

Zheng WANG[1], Hui-fang CHEN[†‡1,2], Lei XIE[1,2], Kuang WANG[1,2]

([1]*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China*)

[†]E-mail: chenhf@zju.edu.cn

Received Aug. 3, 2009; Revision accepted Nov. 9, 2009; Crosschecked May 31, 2010

Abstract: In this paper, retransmission strategies of the network-coding-based packet network are investigated. We propose two retransmission strategies, the packet-loss-edge-based retransmission strategy (PLERT) and the minimum retransmission strategy (MRT), which focus on optimizing the retransmission efficiency without the constraint on the encoding field size. We compared the performances of the proposed retransmission strategies with the traditional automatic repeat-request (ARQ) strategy and the random retransmission strategy. Simulation results showed that the PLERT strategy works well when the packet loss rate is small. Among these retransmission strategies, the performance of the MRT strategy is the best at the cost of the high complexity that is still polynomial. Furthermore, neither of the proposed strategies is sensitive to the encoding field size.

Key words: Generation, Network coding, Packet network, Retransmission
doi:10.1631/jzus.C0910475 **Document code:** A **CLC number:** TN915

1 Introduction

Recently, network coding first proposed by Ahlswede *et al*. (2000) has attracted a great deal of research interest. Due to the encoding ability of intermediate nodes, the bandwidth can be shared among different sinks and the performance of information dissemination over the network is improved by the network coding. It was shown in Ahlswede *et al*. (2000) that the multicast capacity of the network is equal to the size of the minimum cut separating the source and sinks, and that this capacity can be achieved by linear network coding (Li *et al*., 2003).

When the packet loss happens, source messages may be unable to recover in a real network. Thus, if the communication channel between a sender and a sink is not reliable, an appropriate error-control scheme should be used to provide the reliable transmission. Automatic repeat-request (ARQ) and forward error correction (FEC) are usually used to deal with the packet loss problem. On the other hand, many researchers showed that network-coding-based error-control schemes, such as network error correction codes (Cai and Yeung, 2006; Yeung and Cai, 2006; Koetter and Kschischang, 2008; Silva *et al*., 2008; Zhang, 2008) and the retransmission technique (Larsson and Johansson, 2006; Zheng and Sinha, 2007; Nguyen *et al*., 2009), can also be used to solve the packet loss problem.

As we know, retransmission is an effective method to resolve the packet loss problem if feedback is available. Many network-coding-based retransmission strategies for wireless networks have been proposed. In Larsson and Johansson (2006) and Nguyen *et al*. (2009), the XOR-based retransmission strategy for single-hop wireless networks was introduced. Using XOR operations, the source sends an XOR-packet of retransmitted packets instead of sending all of them, and all sinks obtain their required information with XOR operations. Zheng and Sinha (2007) used an XOR-based retransmission strategy to guarantee the end-to-end reliability over wireless

‡ Corresponding author

* Project supported by the Science and Technology Department of Zhejiang Province, China (Nos. 2008C13081 and 2008C01050-2), the Natural Science Foundation of Zhejiang Province, China (No. Y10540720), and the Zhejiang Provincial Foundation for Returnees

networks. Using network coding, the buffer needed in intermediate nodes for retransmission was reduced and the throughput was improved. Ghaderi *et al*. (2008) compared the link-by-link and end-to-end retransmission strategies based on network coding to the traditional ARQ scheme for multicast over wireless networks. However, these strategies utilize the broadcast characteristics of wireless communication, and source packets are not encoded with network coding during the transmission. Hence, these retransmission strategies are not applicable in wired networks.

In wired networks, the reliability of communication using network coding can be simply achieved based on the idea of rateless codes. The source transmits encoded packets until sinks can decode all original packets (Luby, 2002; Maymounkov *et al*., 2006; Shokrollahi, 2006). Lun *et al*. (2008) proposed a link-by-link reliability guarantee scheme with random network coding, which can approach the network capacity. In this scheme, all nodes transmit a random combination of their previously received packets whenever they have a transmission opportunity. Sinks can decode all original packets if they have received enough encoded packets. However, this solution needs a large encoding field size, which leads to a high computational complexity. Another related work is in Sundararajan *et al*. (2008; 2009), where network coding is performed in a completely online manner. A sliding window is maintained and a queue management algorithm called the 'drop-when-seen algorithm' is used. The sliding window is updated according to ARQs from sinks, and transmitted packets are generated by linearly combining packets in the window. Sundararajan *et al*. (2008) gave a deterministic encoding scheme, but only one transmitted packet can be generated according to ARQs from sinks. To generate the next transmitted packet, the next group of ARQs should be waited for. Sundararajan *et al*. (2009) used the randomly linear coding scheme to deal with this problem but a large encoding field size was required.

In this paper, we investigate retransmission strategies of the multicast communication based on network coding in wired networks to achieve the end-to-end reliability. Our motivation is to design the effective retransmission strategy for optimizing the retransmission efficiency without the constraint on the encoding field size. Due to the decoding complexity and the nonzero-delay problem, the network coding is usually performed based on generation, which is called 'generation-based network coding' (Chou *et al*., 2003; Wu *et al*., 2008). Therefore, we focus on the retransmission technique of the generation-based network coding for wired directed networks.

In multicast sessions with linear network coding, intermediate nodes linearly combine incoming packets and forward them. As a result, if one packet is lost during the transmission, it is possible that more than one sink does not receive enough information to recover all of the source messages of that generation. Though received packets are different, the same set of retransmitted packets can meet the requirements of all sinks. Hence, instead of unicasting required packets to each sink, multicasting retransmitted packets utilizing the property of the network coding is a better choice. An efficient retransmission strategy should be carefully designed to make the size of the set as small as possible. In this paper, we address four retransmission strategies of generation-based network coding in the packet network: the traditional ARQ scheme named the 'quasi classical retransmission strategy' (CRT), the rateless-code-based retransmission strategy named the 'random retransmission strategy' (RRT), the packet-loss-edge-based retransmission strategy (PLERT), and the minimum retransmission strategy (MRT). In these retransmission strategies, PLERT and MRT are the main contributions of this paper.

These retransmission strategies adopt different methods to deal with the packet loss problem. In the CRT strategy, receiving ARQs sent by sinks, the source simply retransmits packets according to requests of these ARQs. In the RRT strategy, the source randomly retransmits packets according to the number of packets required by sinks. In the PLERT strategy, all of the nodes detecting the loss of incoming packets send ARQs to the source. Then the source retransmits packets according to these ARQs. In the MRT strategy, sinks send ARQs containing global encoding kernels generated by some rules. With this knowledge, the source minimizes the number of retransmitted packets, which is equal to the maximum number of packets requested by sinks. Furthermore, the CRT, RRT, and MRT strategies are based on local knowledge while the PLERT strategy may need global information.

2 Generation-based network coding

2.1 Basic definition of network coding (Koetter and Médard, 2003; Yeung *et al*., 2005)

A directed acyclic network can be denoted as $G(V, E)$, where V is the node set and E is the edge set. We assume that the order on E is consistent with the associated partial order of the directed acyclic network G. Each directed edge $e=(i, j)\in E$, represents a channel from node i to node j. The edge e is called an outgoing edge for node i and an incoming edge for node j. For a node i, define $\text{out}(i)=\{e\in E: e$ as an outgoing edge of $i\}$ and $\text{in}(i)=\{e\in E: e$ as an incoming edge of $i\}$. For convenience, we assume that multiple edges between a pair of nodes are possible and that all edges have unit capacity. Let $\{S\}$ and $T=\{t_1, t_2, \ldots, t_N\}$ be two disjointed subsets of V. The source, S, multicasts source messages. Nodes in T named sinks are the destinations of source messages. Let h denote the min-cut of the network, which means there are h disjoint paths from the source to each sink. Edges not belonging to these paths are not considered. Thus, each sink exactly has h incoming edges. The source multicasts h message symbols through one transmission. Let $\mathcal{F}$ denote the base encoding field, and $\boldsymbol{x}=(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_h)^{\mathrm{T}}\in\mathcal{F}^h$, where $\boldsymbol{x}$ is named the source message vector. In the packet network, source messages are also vectors, i.e., $\boldsymbol{x}_i\in\mathcal{F}^L$, where L is the dimension of the vector. Hence, source message vector $\boldsymbol{x}$ becomes a matrix with dimension $h\times L$.

Each node $i\in\{V-T\}$ has a local encoding kernel which is a matrix. With this matrix the node linearly encodes incoming messages and forwards them. The system transfer matrix, $\boldsymbol{F}=(f_{de})_{d\in E, e\in E}$, is an $|E|\times|E|$ matrix, where f_{de} is the local encoding value. Matrix $\boldsymbol{B}$ is the local encoding kernel matrix at the source with dimension $h\times|E|$. The global encoding kernels of network codes, $\overline{\boldsymbol{f}}_e$, are given as column vectors of the matrix $\boldsymbol{B}(\boldsymbol{I}-\boldsymbol{F})^{-1}$, which is an $h\times|E|$ matrix. The global encoding kernel of edge e is an $\mathcal{F}$-valued h-dimensional vector, and $\boldsymbol{U}_e=\boldsymbol{x}^{\mathrm{T}}\overline{\boldsymbol{f}}_e$, where $\boldsymbol{U}_e$ is the message transmitted through edge e.

2.2 Generation-based network coding in packet network

In the packet network, the generation-based network coding operates by partitioning source messages into disjoint packet sets called 'generations'. Network coding performed only within a set can reduce the encoding and decoding complexity. As introduced in Chou *et al*. (2003), source messages are vectors with the same dimension and encoding field $\mathcal{F}$. The set size is equal to the max-flow size of the multicast session, h. Each packet contains a message vector, whose global encoding kernel is included in the packet header. All packets related to the same set of h source vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_h$ are said to be in the same generation. The intermediate node linearly encodes incoming packets with its local encoding kernel and forwards encoded packets to the outgoing edges. It also calculates the global encoding kernel of each forwarding packet and writes it into the packet header. A synchronous mechanism is used to forward packets and to calculate the deadline of a generation. The packet is regarded as lost if it does not arrive before the deadline due to congestion or delay. If an expected incoming packet is lost, both the global encoding kernel and the content of the packet are assumed to be zero when the intermediate node executes the network coding.

3 Retransmission strategies

In wireless networks, intermediate nodes buffer packets and send retransmitted packets when the packet loss happens (Zheng and Sinha, 2007). However, in the wired network, if an intermediate node serves for many sessions, its buffer capacity may be not enough to deal with the retransmission. If there is no extra bandwidth for retransmitted packets, retransmitting by intermediate nodes will destroy the network coding topology. In other words, when the packet loss is temporarily caused by the link overload, the transmission of following generations will be affected by the change of the minimum cut if this link is continued to be used for retransmission. For the same reason, unicasting retransmitted packets to each sink is not applicable.

In this paper, we assume that packets are not buffered by intermediate nodes and that the buffer of the source is large enough for retransmission. When a packet is lost, only the source performs retransmission. The loss of ARQ is not considered for simplicity. If the packet loss happens, either intermediate nodes or sinks (based on the retransmission strategy) send ARQs to the source. Otherwise, if the nodes receive all the expected packets, they keep silent. The source waits a certain time T_{rtt} to collect all ARQs in the current generation. T_{rtt} is set according to the maximum round-trip time (RTT) of paths between the source and sinks which can be determined by the same algorithm as TCP. Based on these ARQs, the source generates retransmitted messages which are linear combinations of source messages in the corresponding generation. Since decoding is performed at sinks, linear combination coefficients are also included in retransmitted packets. Then retransmitted packets are transmitted as a part of the following generation. While sinks receive packets containing retransmitted messages, they decode them and obtain get messages in the corresponding generation and retransmitted messages. With retransmitted messages and their linear combination coefficients, sinks decode messages in the earlier generation which failed to decode because of the packet loss.

We explain the principle of network coding as shown in Fig. 1. The source *S* multicasts message (a, b) to sinks *X*, *Y*, *Z*, and each edge has a unit bandwidth. When all packets are transmitted correctly, the transmission result is shown in Fig. 1a. If packet b through edge *SB* is lost, the transmission result is shown in Fig. 1b. Edges *BC*, *BY*, and *BZ* send zero packet, respectively. Then, the packet through edge *CD* is a instead of (a+b). *X* receives a from two incoming edges instead of a and (a+b). *Y* and *Z* receive a and 0 from incoming edges instead of b and (a+b). Then, *X*, *Y*, and *Z* cannot recover the message (a, b) using received packets. Therefore retransmission is necessary.

Here, we consider four retransmission strategies for the generation-based network coding.

3.1 Quasi classical retransmission strategy

As a classical retransmission scheme, when sinks detect the packet loss, they send ARQs to the source. For example, in Fig. 1b, *Y* and *Z* receive a, but their expected packets are b and (a+b). It is not necessary for the source to retransmit both b and (a+b) since only one more packet is enough. An efficient method is for sinks to randomly generate vectors that can span the full space together with the global encoding kernels of received packets. ARQs contain these vectors. The source collects all ARQs from sinks in one generation and uses Gaussian elimination to obtain linearly independent components of vectors in these ARQs. Then it generates retransmitted packets using these independent vectors as global encoding kernels and sends them during the next generation transmission.

3.2 Random retransmission strategy

For the network-coding-based transmission, if packet loss occurs, retransmitted packets can be generated by randomly linearly combining source messages. If the encoding field size is large enough, a randomly generated packet is linearly independent of

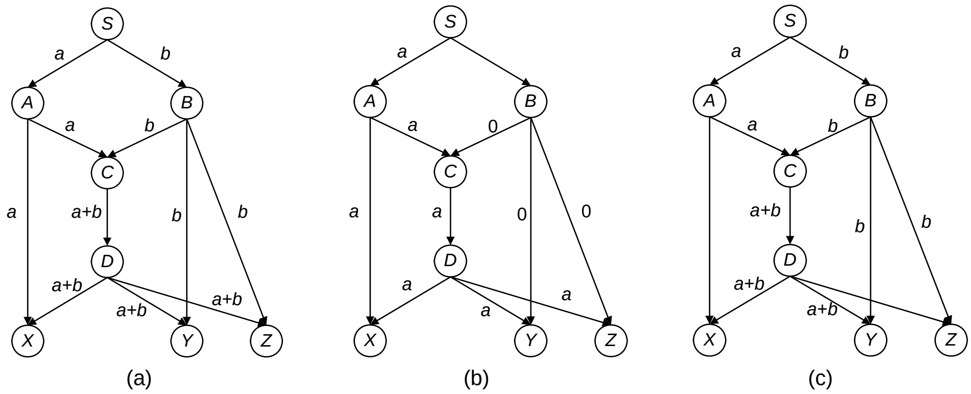

Fig. 1 Examples of network coding

(a) No packet loss; (b) Packet through edge *SB* is lost; (c) Packets through edges *AX* and *DZ* are lost former received packets with a high probability. Hence, the RRT strategy is implemented as follows: when the packet loss occurs, sinks transmit ARQs containing the information regarding how many packets they need. The source collects all ARQs in the same generation and obtains the maximum number of requested packets, *w*. Then it performs a randomly linear combination of the source messages of the corresponding generation to generate *w* retransmitted packets, and sends them during the next generation transmission. If retransmitted packets cannot help sinks recover the source messages, ARQs will be sent again. Obviously, the efficiency of the RRT strategy is low when the encoding field size is small.

3.3 Packet-loss-edge-based retransmission strategy

In the CRT and RRT strategies, for a multicast session with network coding, a lost packet will lead to several sinks sending ARQs. If there are many sinks in the multicast session, the bandwidth consumed by ARQ packets from these sinks is nontrivial. Then sending ARQs by intermediate nodes is an alternative method. The source performs the retransmission based on the feedback from intermediate nodes. The PLERT strategy is proposed based on this motivation.

In Fig. 1b, if the packet through edge *SB* is lost, node *B* sends an ARQ containing the global encoding kernel of packet *b*, i.e., (0, 1). Note that *B* still sends zero packets to edges *BC*, *BY*, and *BZ*. Then *C*, *D*, *X*, *Y*, and *Z* do not send ARQ although their received packets are not the expected ones.

In the PLERT strategy, ARQs are sent by intermediate nodes or/and sinks. Let $\overline{\boldsymbol{f}}_e$ denote the global encoding kernel of the packet transmitted through edge *e* without the packet loss. If a node detects at least one expected incoming packet not arriving on time, it sends an ARQ containing $\overline{\boldsymbol{f}}_e$ of the corresponding edge(s) and the generation number. Similar to the CRT strategy, the source collects all ARQs in the same generation, and uses Gaussian elimination to obtain linearly independent components of the global encoding kernels. Using these independent components as global encoding kernels, the source generates retransmitted packets according to the generation number and sends them in the following generation transmission.

In the traditional network without network coding, since intermediate nodes only forward packets, the lost packets are merely packets required by sinks. But in the network with network coding, since intermediate nodes linearly encode received packets and forward them, received packets are the linear combination of original source packets. Then the lost packets may not be the required ones. However, in Theorem 1, we prove that sinks can recover original source packets using normal received packets and the lost packets retransmitted by the PLERT strategy.

The key problem of the PLERT strategy is how to detect the packet loss and obtain the global encoding kernel. If the global information is available, the problem is easy to solve. From the global information, the node knows which edges have incoming packets and the corresponding global encoding kernels. If incoming packets do not arrive on time, it is claimed that the packet loss has occurred. Otherwise, if the global information is not available and the topology and encoding kernels do not change very frequently, intermediate nodes can infer which edges have incoming packets from previous transmissions. If the packet does not arrive before the deadline, it is assumed to be lost. The node uses the global encoding kernel of the latest incoming packet as the global encoding kernel, and sends it to the source. Because only the global encoding kernels of the packets in the generations without packet loss are used, we set a loss-bit in the packet header to denote whether or not the packet loss occurs. The lost incoming packet is seen as a zero packet with the loss-bit setting. The loss-bit of the packet generated by linearly combining a group of packets including at least one packet with the loss-bit setting will be set. The node buffers only the global encoding kernel of the latest incoming packets without the loss-bit setting.

Theorem 1 In a directed multicast network *G*(*V*, *E*), the min-cut, i.e., the maximum capacity of the multicast session, is achieved using network code. $\overline{\boldsymbol{f}}_e$ denotes the global encoding kernel of the packet transmitting through edge *e* without the packet loss. If packets transmitted through $e_{p_1}, e_{p_2}, ..., e_{p_k}$ are lost, sinks can recover the whole generation's messages with retransmitted packets whose global encoding kernels are $\overline{\boldsymbol{f}}_{p_1}, \overline{\boldsymbol{f}}_{p_2}, ..., \overline{\boldsymbol{f}}_{p_k}$.

Proof We first prove that if packets transmitted through $e_{p_1}, e_{p_2}, ..., e_{p_k}$ are lost, the global encoding kernel $\bar{\boldsymbol{f}}_e$ of edge e becomes $\bar{\boldsymbol{f}}_e'$, which is the linear combination of $\bar{\boldsymbol{f}}_e, \bar{\boldsymbol{f}}_{p_1}, \bar{\boldsymbol{f}}_{p_2}, ..., \bar{\boldsymbol{f}}_{p_k}$. We prove it using the inductive method as follows.

As introduced in Zhang (2008), $\boldsymbol{z}=(z_{e_1}, z_{e_2}, ..., z_{e_{|E|}})^{\mathrm{T}}$ denotes the error message vector, and u_{e_i} is the message transmitted through edge e_i. The extended global encoding kernel of e_i denoted as $\tilde{\boldsymbol{f}}_{e_i}$ is the column of matrix $\boldsymbol{Q}$ indexed by e_i, where

$$\boldsymbol{Q}=\begin{pmatrix}\boldsymbol{B}\\ \boldsymbol{I}\end{pmatrix}(\boldsymbol{I}-\boldsymbol{F})^{-1}=\begin{pmatrix}\boldsymbol{B}\\ \boldsymbol{I}\end{pmatrix}\boldsymbol{K}=\begin{pmatrix}\boldsymbol{BK}\\ \boldsymbol{K}\end{pmatrix}, \quad (1)$$

and $\boldsymbol{K}=(\boldsymbol{I}-\boldsymbol{F})^{-1}$.

When errors are considered, the message transmitted through e_i can be expressed as

$$u_{e_i}=\boldsymbol{x}^{\mathrm{T}}\bar{\boldsymbol{f}}_{e_i}'=(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{z}^{\mathrm{T}})\tilde{\boldsymbol{f}}_{e_i}=\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{I}, \hat{\boldsymbol{f}}_{e_1}, \hat{\boldsymbol{f}}_{e_2}, ..., \hat{\boldsymbol{f}}_{e_{|E|}})\tilde{\boldsymbol{f}}_{e_i}, \quad (2)$$

where $z_{e_j}=\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{f}}_{e_j}$, and $\hat{\boldsymbol{f}}_{e_j}$ is the global encoding kernel of the error component of e_i. If the packet through e_i is transmitted successfully, $\hat{\boldsymbol{f}}_{e_j}=0$. From Eq. (2), we have

$$\bar{\boldsymbol{f}}_{e_i}'=(\boldsymbol{I}, \hat{\boldsymbol{f}}_{e_1}, \hat{\boldsymbol{f}}_{e_2}, ..., \hat{\boldsymbol{f}}_{e_{|E|}})\tilde{\boldsymbol{f}}_{e_i}. \quad (3)$$

Without loss of generality, it is assumed that the topological order of $e_{p_1}, e_{p_2}, ..., e_{p_k}$ satisfies $e_{p_1} \prec e_{p_2} \prec \ldots \prec e_{p_k}$. Obviously, the packet through e_i is affected only by the edges with a smaller topological order than that of e_i. Therefore, Eq. (3) still holds if $\hat{\boldsymbol{f}}_{e_j}=0$ for all $e_j \succ e_i$. If the packet which should be transmitted through e_i is lost, the error component is $z_{e_i}-u_{e_i}$, and $\hat{\boldsymbol{f}}_{e_j}=-\bar{\boldsymbol{f}}_{e_j}'$.

When n=1, we have

$$z_{e_{p_1}}=-u_{e_{p_1}}=\boldsymbol{x}^{\mathrm{T}}(-\bar{\boldsymbol{f}}_{p_1}), \quad (4)$$

$$\bar{\boldsymbol{f}}_{e_i}'=(\boldsymbol{I}, 0, ..., -\bar{\boldsymbol{f}}_{e_{p_1}}, 0, ...)\tilde{\boldsymbol{f}}_{e_i}=\bar{\boldsymbol{f}}_{e_i}-k_{e_{p_1}, e_i}\bar{\boldsymbol{f}}_{p_1}, \quad (5)$$

where $k_{e_{p_1}, e_i}$ is the element of $\boldsymbol{K}$ indexed by (e_{p_1}, e_i). Hence, the assumption holds.

If the assumption holds when n=m, that is,

$$\bar{\boldsymbol{f}}_{e_i}'=\bar{\boldsymbol{f}}_{e_i}-k_{i1}\bar{\boldsymbol{f}}_{e_{p_1}}-k_{i2}\bar{\boldsymbol{f}}_{e_{p_2}}-\cdots-k_{im}\bar{\boldsymbol{f}}_{e_{p_m}}, \quad (6)$$

when n=m+1, the global encoding kernel of the packet which should be transmitted through $e_{p_{(m+1)}}$ is

$$\bar{\boldsymbol{f}}_{e_{p_{(m+1)}}}'=\bar{\boldsymbol{f}}_{e_i}-k_{p_m 1}\bar{\boldsymbol{f}}_{p_1}-k_{p_m 2}\bar{\boldsymbol{f}}_{p_2}-\cdots-k_{p_m m}\bar{\boldsymbol{f}}_m, \quad (7)$$

$$\hat{\boldsymbol{f}}_{e_{p_{(m+1)}}}=-\bar{\boldsymbol{f}}_{e_{p_{(m+1)}}}'. \quad (8)$$

Then,

$$\begin{aligned}\bar{\boldsymbol{f}}_{e_i}'&=(\boldsymbol{I}, 0, ..., 0, \hat{\boldsymbol{f}}_{e_{p_1}}, 0, ..., 0, \hat{\boldsymbol{f}}_{e_{p_2}}, 0, ..., 0, \hat{\boldsymbol{f}}_{e_{p_{(m+1)}}}, 0, ...)\tilde{\boldsymbol{f}}_{e_i}\\ &=(\boldsymbol{I}, 0, ..., 0, -\bar{\boldsymbol{f}}_{e_{p_1}}', 0, ..., 0, -\bar{\boldsymbol{f}}_{e_{p_2}}', 0, ..., 0, -\bar{\boldsymbol{f}}_{e_{p_{(m+1)}}}', 0, ...)\tilde{\boldsymbol{f}}_{e_i}.\end{aligned} \quad (9)$$

Substituting $\bar{\boldsymbol{f}}_{e_{p_i}}'$ in Eq. (6), we have

$$\bar{\boldsymbol{f}}_{e_i}'=\bar{\boldsymbol{f}}_{e_i}-k_{i1}\bar{\boldsymbol{f}}_{e_{p_1}}-k_{i2}\bar{\boldsymbol{f}}_{e_{p_2}}-\cdots-k_{i(m+1)}\bar{\boldsymbol{f}}_{e_{p_{(m+1)}}}. \quad (10)$$

Hence, if the packets through $e_{p_1}, e_{p_2}, ..., e_{p_k}$ are lost, the global encoding kernel of the ith incoming edge of sink t, $\bar{\boldsymbol{\alpha}}_i'$, can be denoted as

$$\bar{\boldsymbol{\alpha}}_i'=\bar{\boldsymbol{\alpha}}_i+k_{t1}\bar{\boldsymbol{f}}_{e_{p_1}}+k_{t2}\bar{\boldsymbol{f}}_{e_{p_2}}+\cdots+k_{tn}\bar{\boldsymbol{f}}_{e_{p_n}}, \quad (11)$$

where $\bar{\boldsymbol{\alpha}}_i$ is the global encoding kernel of the ith incoming edge of sink t without the packet loss. Therefore, the rank of $(\bar{\boldsymbol{\alpha}}_1', \bar{\boldsymbol{\alpha}}_2', ..., \bar{\boldsymbol{\alpha}}_h', \bar{\boldsymbol{f}}_{e_{p_1}}, \bar{\boldsymbol{f}}_{e_{p_2}}, ..., \bar{\boldsymbol{f}}_{e_{p_n}})$ is h.

Remark 1 The PLERT strategy does not achieve the most efficient retransmission in some cases. As shown in Fig. 1c, if packets through *AX* and *DZ* are lost, two packets, *a* and *b*, will be retransmitted according to the PLERT strategy. In fact, only one packet such as *a*+2*b* can satisfy all requests. With this motivation, the MRT strategy is proposed.

3.4 Minimum retransmission strategy

Obviously, the source can perform a most efficient retransmission strategy if sinks send ARQs containing all global encoding kernels of received packets. However, extra bandwidth will be used to send ARQs and the whole computation procedure is performed by the source.

Here we present a relatively practical algorithm to minimize the number of retransmitted packets of the generation-based retransmission. As shown in Algorithm 1 and Theorem 2, each sink generates orthogonal global encoding kernels of the packets satisfying its requirements and sends them to the source. The source performs a polynomial deterministic algorithm to generate the minimum number of retransmitted packets. The number of these packets is the maximum number of packets required by sinks.

The MRT strategy is executed according to Algorithm 1, which is much like the linear information flow (LIF) algorithm proposed in Jaggi *et al*. (2005). Let $N=|T|$ denote the number of sinks and r_i denote the rank of $(\bar{\boldsymbol{\alpha}}'_{i1},\bar{\boldsymbol{\alpha}}'_{i2},...,\bar{\boldsymbol{\alpha}}'_{ih})$, where $\bar{\boldsymbol{\alpha}}'_{i1},\bar{\boldsymbol{\alpha}}'_{i2},...,\bar{\boldsymbol{\alpha}}'_{ih}$ are the global encoding kernels of the received packets of t_i with the packet loss.

Lemma 1 (Lemma 8 in Jaggi *et al*. (2005)) Let $N\leq|\mathbb{F}|$. Consider pairs $(\boldsymbol{x}_j, \boldsymbol{y}_j)\in\mathbb{F}^h\times\mathbb{F}^h$ with $\boldsymbol{x}_j\cdot\boldsymbol{y}_j\neq0$ for $1\leq j\leq N$. There exists a linear combination $\boldsymbol{u}$ of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ such that $\boldsymbol{u}\cdot\boldsymbol{y}_j\neq0$ for $1\leq j\leq N$. Such a vector $\boldsymbol{u}$ can be found in time $O(N^2h)$.

Algorithm 1 Minimum retransmission algorithm

1. Each sink t_i generates $q_i=h-r_i$ orthogonal vectors $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$, where $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$ span the subspace being orthogonal with the subspace spanned by $\bar{\boldsymbol{\alpha}}'_{i1},\bar{\boldsymbol{\alpha}}'_{i2},...,\bar{\boldsymbol{\alpha}}'_{ih}$. Then, these orthogonal vectors are transmitted to the source.

2. The source collects ARQs in the same generation, and initializes a set C_i for each ARQ. C_i is defined as $C_i=\{\boldsymbol{c}_{i1},\boldsymbol{c}_{i2},...,\boldsymbol{c}_{iq_i}\}$, where $\boldsymbol{c}_{ij}=\boldsymbol{\beta}_{ij}$.

3. Let q equal the maximum of q_i. For m from 1 to q, perform the following steps (if $q_i<m$, sink t_i is ignored):

(1) Applying Lemma 1 to $\{(\boldsymbol{c}_{im}, \boldsymbol{c}_{im}), 1\leq i\leq N\}$, a vector $\boldsymbol{\gamma}_m$, which is the linear combination of $\boldsymbol{c}_{1m}, \boldsymbol{c}_{2m}, \ldots, \boldsymbol{c}_{nm}$, is obtained.

(2) For each t_i, $\boldsymbol{c}_{in}$ is updated as

$$\boldsymbol{c}_{in}:=\boldsymbol{c}_{in}-(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{in})(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{im})^{-1}\boldsymbol{c}_{im}, \tag{12}$$

where $m+1<n<q_i$.

Since $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$ are linearly independent and $\boldsymbol{c}_{in}$ is the linear combination of $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$, $\boldsymbol{c}_{in}\neq0$. Based on Algorithm 1, q vectors, $\gamma_1, \gamma_2, \ldots, \gamma_q$, are generated. As shown in Theorem 2, $\gamma_1, \gamma_2, \ldots, \gamma_q$ are the global encoding kernels of retransmitted packets.

Lemma 2 The following properties hold for all sinks throughout step 3 of Algorithm 1:

1. $\gamma_u\cdot\boldsymbol{c}_{iu}\neq0$ and $\gamma_u\cdot\boldsymbol{c}_{iv}=0$ for $u\neq v$.

2. $\bar{\boldsymbol{\alpha}}'_{iu}\cdot\boldsymbol{c}_{iv}=0$ for $u=1, 2, \ldots, h$ and $v=1, 2, \ldots, q_i$.

Proof Obviously, these properties hold before step 3 starts. In the mth loop, only $\boldsymbol{c}_{iv}$ with $v>m$ are updated. Therefore, only $\boldsymbol{c}_{iv}$ with $v>m$ need to be verified. Let $\boldsymbol{c}'_{iv}$ denote the value of $\boldsymbol{c}_{iv}$ after the update.

1. That $\gamma_u\cdot\boldsymbol{c}_{iu}\neq0$ is obvious because $\boldsymbol{c}_{iu}$ is not changed since $\boldsymbol{\gamma}_u$ has been generated.

In the mth loop, if $v>m$, we have

$$\boldsymbol{\gamma}_m\cdot\boldsymbol{c}'_{iv}=\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{iv}-(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{iv})(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{im})^{-1}(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{im})=0.$$

For $u<m$ and $v>m$

$$\boldsymbol{\gamma}_u\cdot\boldsymbol{c}'_{iv}=\boldsymbol{\gamma}_u\cdot\boldsymbol{c}_{iv}-(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{iv})(\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{im})^{-1}(\boldsymbol{\gamma}_u\cdot\boldsymbol{c}_{im})=0.$$

2. $\boldsymbol{c}_{iv}$ is the linear combination of $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$ throughout the whole loop. From the generation method of $\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},...,\boldsymbol{\beta}_{iq_i}$, we have $\bar{\boldsymbol{\alpha}}'_{iu}\cdot\boldsymbol{c}_{iv}=0$.

Theorem 2 For each sink t_i, the rank of $(\bar{\boldsymbol{\alpha}}'_{i1},\bar{\boldsymbol{\alpha}}'_{i2},...,\bar{\boldsymbol{\alpha}}'_{ih},\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2,...,\boldsymbol{\gamma}_{q_i})$ is h, where $\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2,...,\boldsymbol{\gamma}_{q_i}$ are vectors generated by Algorithm 1.

Proof Let R_i denote the received global encoding kernels of sink t_i. R_i is initialized as $\{\bar{\boldsymbol{\alpha}}'_{i1},\bar{\boldsymbol{\alpha}}'_{i2},...,\bar{\boldsymbol{\alpha}}'_{ih}\}$. In the mth loop of step 3, $\boldsymbol{\gamma}_m$ is generated. From Lemma 2, it is inferred that for each vector $\boldsymbol{\lambda}\in R_i$, $\boldsymbol{\lambda}\cdot\boldsymbol{c}_{in}=0$ while $n\geq m$ and $\boldsymbol{\gamma}_m\cdot\boldsymbol{c}_{im}\neq0$. It means that subspace Φ spanned by vectors in R_i is orthogonal with the subspace spanned by $\boldsymbol{c}_{im},\boldsymbol{c}_{i(m+1)},...,\boldsymbol{c}_{iq_i}$ and $\boldsymbol{\gamma}_m\notin\Phi$. After $\boldsymbol{\gamma}_m$ is added into R_i, the rank of Φ increases by 1, i.e., from r_i+m-1 to r_i+m. After q_i loops, each R_i is full rank.

Remark 2 For four retransmission strategies mentioned above, the CRT, RRT, and MRT strategies are suitable for any kind of network coding, regardless of whether the network coding is centralized or distributed, the global information is available or unavailable, and encoding functions are fixed or randomly chosen before every transmission. When the global information is not available, the PLERT strategy is applicable, but there is a restriction for this strategy. It is not suitable if the network coding is executed randomly in every transmission. Moreover, for the CRT, RRT, and PLERT strategies, because the encoding field size has no constraint, the size of the encoding field having a feasible network coding solution is enough. For the MRT strategy, the encoding field size, which is not smaller than the number of the sinks, is feasible.

Then, we analyze the overhead of retransmission strategies, such as the decoding delay, the buffer size, and the computational complexity.

The drawback of generation-based retransmission strategies is that if the packet loss rate is high, the packet loss occurs continuously for a large number of generations. In this case, the decoding delay is large, which leads to a large buffer size of sinks. Obviously, in all of our proposed retransmission strategies, if the packet loss does not occur during one generation transmission, all former generations related to it can be decoded.

Hence, the expected decoding delay is

$$D = (T_{\mathrm{rtt}} + M/R) \Big/ \prod_{e \in E} (1 - p_e)\,, \tag{13}$$

where T_{rtt}, used to wait for all ARQs in the same generation, is the maximum RTT of paths between the source and sinks, M is the size of the generation, R is the bit rate of the source, and p_e is the packet loss rate of edge e.

The expected buffer size of the sink is

$$\mathrm{Buf} = RD. \tag{14}$$

Since all packets that failed to be decoded by the sink should be buffered at the source for retransmission, the expected buffer size of the source is also RD.

The complexity of the PLERT strategy is calculated simply. Since the number of lost packets, k, during one generation transmission is at most $|E|$, the complexity of the PLERT strategy is $O(|E|h^3)$, which is used to perform the Gaussian elimination. However, the average complexity is much smaller than $|E|h^3$ since k is smaller than $|E|$. In addition, k is usually smaller than h in the wired network. If $k<h$, the complexity is $O(h^3)$.

The computational complexity of the MRT strategy can be calculated as follows. To generate $\boldsymbol{\beta}_{i1}, \boldsymbol{\beta}_{i2}, ..., \boldsymbol{\beta}_{iq_i}$ in Algorithm 1, we need to solve q_i systems of linear equations whose dimensions are at most $h\times h$. As each system of linear equations is solved in time $O(h^3)$ and $q_i<h$, $\boldsymbol{\beta}_{i1}, \boldsymbol{\beta}_{i2}, ..., \boldsymbol{\beta}_{iq_i}$ are generated in time $O(h^4)$ at sink i. In step 3 of Algorithm 1, applying Lemma 1 takes time $O(N^2h)$ and updating each $\boldsymbol{c}_{in}$ takes time $O(h)$. In each loop, Lemma 1 is applied once, and no more than Nh $\boldsymbol{c}_{in}$ need to be updated. Furthermore, the number of the loops is not larger than h. Hence, the complexity at the source is $O(Nh^2(N+h))$.

4 Simulation results and discussions

In this section, we evaluate the performance of the four retransmission strategies of the generation-based network coding in the packet network with randomly generated network topologies, such as $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, and $G_3(V_3, E_3)$. We use the average number of transmissions per packet as the performance metric. Figs. 2a and 2b depict $G_1(V_1, E_1)$ with $|V_1|=15$, $|T_1|=3$, $h_1=4$ and $G_2(V_2, E_2)$ with $|V_2|=22$, $|T_2|=3$, $h_2=6$, where S is the source and $\{t_i\}$ are sinks. $G_3(V_3, E_3)$ with $|V_3|=40$, $|T_3|=6$, $h_3=8$ is not shown due to the limited space and the complication of the topology. The packet loss rate of each edge is independent and is set to be p, and the bandwidth of each edge shown in Fig. 2 is assigned randomly. The linear network code is constructed with the random approach proposed in Ho *et al*. (2006).

Simulation results about the performance of four retransmission strategies in G_1, G_2, and G_3 are shown in Figs. 3a–3c, where the packet loss rate p of each edge is randomly assigned ranging from 1/1000 to 1/100. From Fig. 3, we observe that the PLERT and MRT strategies are effective for various scales of network topology. Unlike the RRT strategy, these two strategies work well with various encoding field sizes. However, the performance of the PLERT strategy degrades when the size of the network becomes large. The reason for this phenomenon is that inefficient cases as shown in Fig. 1c are much likely to occur with a large scale network.

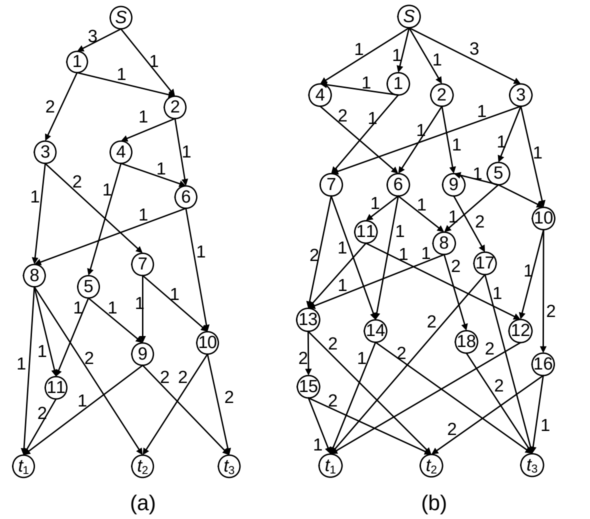

Fig. 2 Network topologies for simulations
(a) $G_1(V_1, E_1)$ with $|V_1|$=15, $|T_1|$=3, and h=4; (b) $G_2(V_2, E_2)$ with $|V_2|$=22, $|T_2|$=3, and h=6. S is the source and $\{t_i\}$ are sinks

In the following, we select G_1 to evaluate the performance of four retransmission strategies under different encoding field sizes and packet loss rates.

The performance of four retransmission strategies along with the encoding field size increasing from 3 to 257 is shown in Fig. 4, where the packet loss rate of each edge is set to 1/100. From Fig. 4, we observe that the MRT strategy achieves the best performance regardless of the encoding field size while the performance of the CRT strategy is the worst. As the encoding field size increases, the performance of the RRT strategy improves until it converges to that of the MRT strategy with a large enough encoding field size. The performance of the PLERT strategy is worse than that of the MRT strategy and better than that of the RRT strategy with a small encoding field size. The reason for this phenomenon is that inefficient cases as shown in Fig. 1c occur. The performances of the PLERT strategy and the MRT strategy are not sensitive to the increasing of the encoding field size.

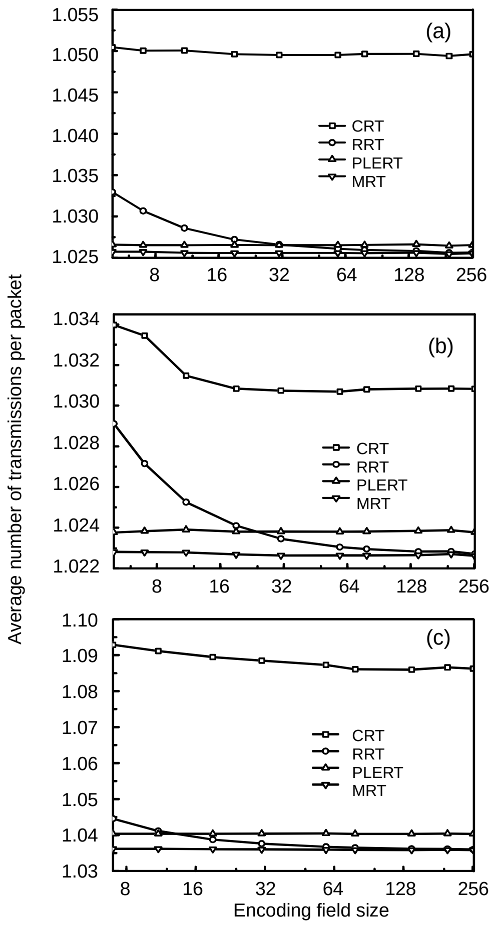

Fig. 3 The average number of transmissions per packet with a randomly assigned packet loss rate under different encoding field sizes
(a) $G_1(V_1, E_1)$; (b) $G_2(V_2, E_2)$; (c) $G_3(V_3, E_3)$

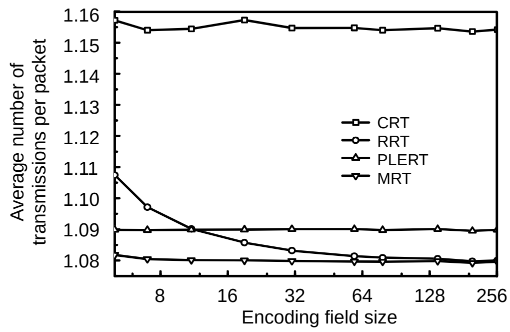

Fig. 4 The average number of transmissions per packet under different encoding field sizes (packet loss rate p=1/100)

The performance along with the encoding field size increasing from 3 to 257 is shown in Fig. 5, where p is set to 1/1000. From Fig. 5, we observe that the performance of the MRT strategy is the best and the performance of the CRT strategy is the worst. The performances of the PLERT strategy and the MRT strategy are very similar because inefficient cases as shown in Fig. 1c seldom occur in the case of low packet loss rate. Meanwhile, the performances of these two strategies are stable when the encoding field size increases. The performance of the RRT strategy is close to that of the MRT strategy when the encoding field size increases.

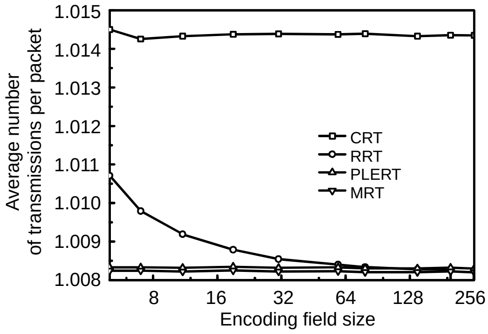

Fig. 5 The average number of transmissions per packet under different encoding field sizes (packet loss rate *p*=1/1000)

Fig. 6a shows the performance of four retransmission strategies with the packet loss rate increasing from 1/2000 to 1/50, where the encoding field size is 5. From Fig. 6a, we observe that the performance of the MRT strategy is the best, while the CRT strategy is the worst. The RRT strategy needs a higher number of transmissions per packet than the MRT strategy because it requires a large encoding field size. Although the performance of the PLERT strategy is close to that of the MRT strategy with a small packet loss rate, it degrades when the packet loss rate becomes high. The reason for this phenomenon is that the bad effect of inefficient cases as shown in Fig. 1c becomes serious if the packet loss rate is high.

Fig. 6b shows the performance of four retransmission strategies with the packet loss rate increasing from 1/2000 to 1/50, where the encoding field size is 257. From Fig. 6b, we observe that when the encoding field size is large, the performance of the MRT strategy is still the best while the CRT strategy is the worst. The performance of the RRT strategy is closer to the MRT strategy no matter how large the packet loss rate is. However, when the encoding field size is large, the performance of the PLERT strategy rapidly decays with the increase of the packet loss rate.

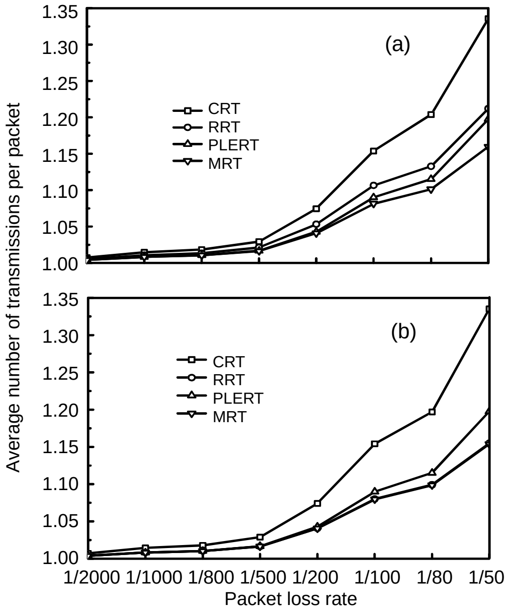

Fig. 6 The average number of transmissions per packet under different packet loss rates

(a) Encoding field size $|F|=5$; (b) Encoding field size $|F|=257$

5 Conclusions

Retransmission is an effective method to resolve the packet loss problem in the packet network if the feedback is available. In this paper, we investigated the retransmission technique of generation-based network coding in the packet network. We addressed four retransmission strategies, CRT, RRT, PLERT, and MRT. Simulation results show that the CRT strategy executed as the classical retransmission scheme has the worst performance. The RRT strategy based on the theory of random network coding does not work well if the encoding field size is small, but it has good performance if the encoding field size is large enough. Moreover, the RRT strategy is not sensitive to the packet loss rate. The PLERT strategy requires any node detecting the loss of expected incoming packets to send ARQ. This strategy is not sensitive to the encoding field size, but it does not work well when the packet loss rate is high. The MRT strategy has the best performance with any packet loss rate and any encoding field size, but its complexity is much higher than that of other three strategies. However, the complexity of the MRT strategy is still polynomial. Our future work will focus on analyzing the decoding delay of these retransmission strategies with numerical results. We will also try to improve the PLERT strategy with a high packet loss rate.

References

Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W., 2000. Network information flow. *IEEE Trans. Inform. Theory*, **46**(4):1204-1216. [doi:10.1109/18.850663]

Cai, N., Yeung, R.W., 2006. Network error correction, part II: lower bounds. *Commun. Inform. Syst.*, **6**(1):37-54.

Chou, P.A., Wu, Y., Jain, K., 2003. Practical Network Coding. 41st Allerton Conf. on Communication, Control, and Computing, p.1-10.

Ghaderi, M., Towsley, D., Kurose, J., 2008. Reliability Gain of Network Coding in Lossy Wireless Networks. 27th IEEE Conf. on Computer Communications, p.2171-2179. [doi:10.1109/INFOCOM.2008.284]

Ho, T., Médard, M., Koetter, R., Karger, D.R., Effros, M., Shi, J., Leong, B., 2006. A random linear network coding approach to multicast. *IEEE Trans. Inform. Theory*, **52**(10): 4413-4430. [doi:10.1109/TIT.2006.881746]

Jaggi, S., Sanders, P., Chou, P.A., Effros, M., Egner, S., Jain, K., Tolhuizen, L.M.G.M., 2005. Polynomial time algorithms for multicast network code construction. *IEEE Trans. Inform. Theory*, **51**(6):1973-1982. [doi:10.1109/TIT.2005.847712]

Koetter, R., Kschischang, F.R., 2008. Coding for errors and erasures in random network coding. *IEEE Trans. Inform. Theory*, **54**(8):3579-3591. [doi:10.1109/TIT.2008.926449]

Koetter, R., Médard, M., 2003. An algebraic approach to network coding. *IEEE/ACM Trans. Network.*, **11**(5):782-795. [doi:10.1109/TNET.2003.818197]

Larsson, P., Johansson, N., 2006. Multi-User ARQ. 66th IEEE Vehicular Technology Conf., p.2052-2057. [doi:10.1109/VETECS.2006.1683207]

Li, S.Y.R., Yeung, R.W., Cai, N., 2003. Linear network coding. *IEEE Trans. Inform. Theory*, **49**(2):371-381. [doi:10.1109/TIT.2002.807285]

Luby, M., 2002. LT Codes. 43rd Annual IEEE Symp. on Foundations of Computer Science, p.271-280. [doi:10.1109/SFCS.2002.1181950]

Lun, D.S., Médarda, M., Koetter, R., Effros, M., 2008. On coding for reliable communication over packet networks. *Phys. Commun.*, **1**(1):3-20. [doi:10.10.16/j.phycom.2008.01.006]

Maymounkov, P., Harvey, N., Lun, D., 2006. Methods for Efficient Network Coding. 44th Annual Allerton Conf. on Communication, Control, and Computing, p.1-10.

Nguyen, D., Tran, T., Nguyen, T., Bose, B., 2009. Wireless broadcast using network coding. *IEEE Trans. Veh. Technol.*, **58**(2):914-925. [doi:10.1109/TVT.2008.927729]

Shokrollahi, A., 2006. Raptor codes. *IEEE Trans. Inform. Theory*, **52**(6):2551-2567. [doi:10.1109/TIT.2006.874390]

Silva, D., Kschischang, F.R., Koetter, R., 2008. A rank-metric approach to error control in random network coding. *IEEE Trans. Inform. Theory*, **54**(9):3951-3967. [doi:10.1109/TIT.2008.928291]

Sundararajan, J.K., Shah, D., Médard, M., 2008. ARQ for Network Coding. IEEE Int. Symp. on Information Theory, p.1651-1655. [doi:10.1109/ISIT.2008.4595268]

Sundararajan, J.K., Shah, D., Médard, M., Mitzenmacher, M., Barros, J., 2009. Network Coding Meets TCP. 28th IEEE Conf. on Computer Communications, p.280-288. [doi:10.1109/INFCOM.2009.5061931]

Wu, X., Zhao, C., You, X., 2008. Generation-Based Network Coding over Networks with Delay. IFIP Int. Conf. on Network and Parallel Computing, p.365-368. [doi:10.1109/NPC.2008.57]

Yeung, R.W., Cai, N., 2006. Network error correction, part I: basic concepts and upper bounds. *Commun. Inform. Syst.*, **6**(1):19-36.

Yeung, R.W., Li, S.Y.R., Cai, N., Zhang, Z., 2005. Network coding theory. *Found. Trends Commun. Inform. Theory*, **2**(4-5):241-381. [doi:10.1561/0100000007I]

Zhang, Z., 2008. Linear network error correction codes in packet networks. *IEEE Trans. Inform. Theory*, **54**(1):209-218. [doi:10.1109/TIT.2007.909139]

Zheng, Z., Sinha, P., 2007. XBC: XOR-Based Buffer Coding for Reliable Transmissions over Wireless Networks. 4th Int. Conf. on Broadband Communications, Networks, and Systems, p.76-85. [doi:10.1109/BROADNETS.2007.4550409]