RESEARCH

Open Access

CrossMark

Systematic network coding for two-hop lossy transmissions

Ye Li[*], Steven Blostein and Wai-Yip Chan

Abstract

In this paper, we consider network transmissions over a single or multiple parallel two-hop lossy paths. These scenarios occur in applications such as sensor networks or WiFi offloading. Random linear network coding (RLNC), where previously received packets are re-encoded at intermediate nodes and forwarded, is known to be a capacity-achieving approach for these networks. However, a major drawback of RLNC is its high encoding and decoding complexity. In this work, a systematic network coding method is proposed. We show through both analysis and simulation that the proposed method achieves higher end-to-end rate as well as lower computational cost than RLNC for finite field sizes and finite-sized packet transmissions.

Keywords: Network coding; Systematic; Two-hop transmission

1 Introduction

Multi-path multi-hop transmission is commonly seen in many communication scenarios, where one or more intermediate nodes may connect the source and destination nodes along a single path and multiple such paths may exist in parallel. Network coding [1–3] can be beneficial in such scenarios. When packets pass through lossy multi-hop links, if intermediate nodes re-encode packets with previously received ones, coding can achieve the max-flow capacities of the links [4] and therefore improve the end-to-end transmission rate while offering protection against erasures. Importantly, this improvement is achieved without the need of either acknowledgements between nodes or estimation of link qualities. Random linear network coding (RLNC) [5], in which the re-encoded packets are random linear combinations of packets received at the intermediate node, can achieve this in a distributed manner. RLNC is known to be asymptotically *rate-optimal*, i.e., it achieves the max-flow capacity when the size of the arithmetic field used for coding goes to infinity.

However, an important drawback of RLNC is complexity. Decoding of RLNC, performed on M source packets each with K symbols from a finite field of size q, $\mathbb{F}_q$, requires the solution of a linear system of equations in

M unknowns, which requires $\mathcal{O}(M^3 + M^2K)$ operations for unstructured matrices, e.g., via Gaussian elimination (GE). For some scenarios wherein nodes have limited computing capability and battery power, this computational cost may be prohibitively high. Moreover, the rate optimality of RLNC relies on a sufficiently large field size, which, as a consequence, requires a larger packet header to carry coding vectors as well as more computational resources when performing finite field arithmetic (for example, larger look-up tables may be needed to speed up multiplication [6]). On the other hand, when the field size is small, for example, $q = 2$ (i.e., binary), the achieved rate using RLNC can be much lower than the max-flow capacity, especially when the number of source packets is also small.

In this paper, we propose a systematic network coding method for networks with one or more parallel paths of two-hop lossy links. The proposed scheme possesses similar rate performance as RLNC but at significantly lower encoding/decoding computational cost. The proposed scheme first sends uncoded packets in their original order and then sends a potentially unlimited number of coded packets using RLNC. Combined with the proposed re-encoding strategy at the intermediate nodes, which forwards received uncoded packets or else performs RLNC, the scheme will be shown to achieve higher rate than RLNC and require less computation. In applications where the field size and the number of source

*Correspondence: y.li@queensu.ca
Department of Electrical and Computer Engineering, Queen's University, K7L 3N6 Kingston, Canada

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 2 of 14

packets are not very large, the improvement of the proposed scheme over using RLNC is significant.

The transmission scheme where a source node sends uncoded packets first and then sends coded packets has recently appeared in the literature for one-hop links. For example, in [7–9], the benefits of reducing decoding complexity and completion time are explored. For the single-path two-hop link, a scheme is presented in [10] (Section 3.2) in which fountain-coded packets (e.g., using LT codes [11]) are transmitted in the first hop; the intermediate node forwards the received packets from the first hop and sends some re-encoded packets from the buffer after forwarding. The method is studied from a coding theory point of view in the asymptotic regime. In [12], a scheme with a similar transmission strategy in the first hop as ours is proposed. However, the method in [12] assumes a direct link between the source and destination nodes and only sends coded packets from the buffer on the second hop after all uncoded packets are transmitted and does not make use of possible transmission opportunities on the second hop before uncoded packets are finished. Therefore, the achievable rate of the scheme in [12] is strictly lower than the max-flow capacity of the link.

A particular application example of the proposed method is WiFi offloading [13]. WiFi offloading is an efficient way to address the rapid increase in data traffic that poses great challenges to current cellular networks. A notable cause is video streaming to smart phones and tablets. An offloading path may be established via a WiFi access point (AP) to transmit packets from the packet data network gateway (GW) to the user equipment (UE), forming a two-hop lossy path. A parallel path going from GW to UE via base-station (BS), which also forms a two-hop lossy path, may or may not be available simultaneously. The multi-path TCP (MPTCP) [14, 15] protocol can be used to seamlessly establish/close the two-hop connections. In [16], it has been shown that incorporating RLNC with MPTCP may achieve higher throughput than MPTCP without network coding when links are lossy. The "seen packet" concept proposed in [17] can be used to address the compatibility issue of network coding with TCP.

This paper considers systematic network coding in two scenarios. First, we consider transmission over a single two-hop link. We prove that the proposed systematic network coding outperforms RLNC in terms of both end-to-end rate and encoding/decoding costs. An analysis based on finite absorbing Markov chains is provided to statistically characterize the end-to-end rate of transmission. Second, we consider the case where two parallel paths can be used to speed up the transmission. In this case, packets are sent from the source node to the destination node over two parallel two-hop links simultaneously. We show that the proposed scheme again outperforms RLNC.

We formulate a packet allocation problem to schedule the transmission of uncoded packets among the two paths such that the expected end-to-end rate is maximized with minimized decoding cost.

We remark that although the proposed method performs better than RLNC, RLNC is rate optimal for multicasting in unknown network topologies. In other words, the benefits of the proposed method in the paper come from exploiting the known topology. To reduce complexity of RLNC in networks with unknown or complex topologies, other approaches such as generation-based network coding [18–24] may be used.
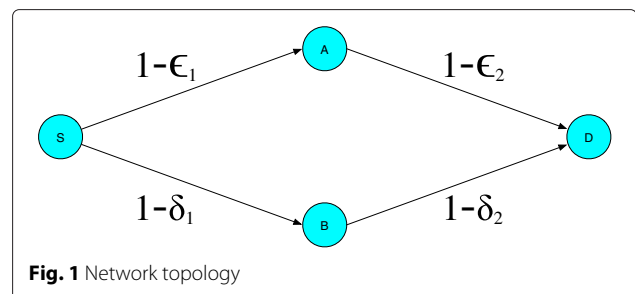
2 System model
2.1 Transmission model
Suppose that a source node, denoted as S, has M data packets $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M\}$ to send to a destination node D. Each packet consists of K symbols from $\mathbb{F}_q$. Separate source and channel coding are assumed. Source packets are independent and identically distributed (i.i.d.) to maximize the information rate. The transmission may involve two paths as shown in Fig. 1; each is a two-hop lossy link. We use S-A-D and S-B-D to refer to the two two-hop links through the two intermediate nodes A and B, respectively.

We assume that A and B operate in the half-duplex mode. Transmission occurs successively on the two hops of each path and the two successive transmissions are referred to as a *use* of the path. Let rate R_1 define the number of packets that are sent on S-A or A-D per use of S-A-D, and let R_2 define the number of packets that are sent on S-B or B-D per use of S-B-D. Given the rates R_1 and R_2, let ϵ_1, ϵ_2, δ_1 and δ_2 denote the corresponding packet loss probabilities on links S-A, A-D, S-B and B-D, respectively. The max-flow capacities of the parallel two paths can then be expressed as $C_1 = R_1 (1 - \max\{\epsilon_1, \epsilon_2\})$ and $C_2 = R_2(1 - \max\{\delta_1, \delta_2\})$ packets per path use, respectively. A *network use* refers to using each path once. The max-flow capacity of the network is then $C_1 + C_2$ packets per network use.

Throughout transmission, we assume that no feedback is available between nodes to indicate loss/reception of packets except that D can immediately inform S, A, and B to stop transmission when D finishes recovering all M

Fig. 1 Network topology

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 3 of 14

source packets. We denote the number of network uses that are needed by D to complete decoding of all packets as the *completion time* of the transmission, T. The end-to-end rate, $R = \frac{M}{T}$.

2.2 Random linear network coding

In the RLNC scheme, S always sends random linear combinations of source packets while A and B send random linear combinations of packets in their buffers, respectively. The received packets at D are in the form of $\mathbf{r} = \sum_{i=1}^{M} g_i \mathbf{s}_i$, where g_i is a random coding coefficient chosen from $\mathbb{F}_q$. The vector $[g_1, \ldots, g_M]$ is referred to as the *coding vector* of $\mathbf{r}$. D does not need to distinguish whether the received packets are from A or B.

Suppose that N packets $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N$ are received. The recovery of source packets corresponds to solving the linear system of equations

$$
\begin{bmatrix}
g_{1,1} & \cdots & g_{1,M} \\
\vdots & \ddots & \vdots \\
g_{N,1} & \cdots & g_{N,M}
\end{bmatrix}
\begin{bmatrix}
\mathbf{s}_1 \\
\vdots \\
\mathbf{s}_M
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{r}_1 \\
\vdots \\
\mathbf{r}_N
\end{bmatrix},
\tag{1}
$$

where the matrix on the left-hand side is referred to as the *decoding matrix*, whose rows comprise the coding vectors of the received packets, and the $N \times K$ right-hand side matrix comprises the coded source symbols of the received packets. An *innovative packet* refers to a received packet whose coding vector is not in the span of the coding vectors of all previously received packets. Decoding is only possible if M innovative packets are received, i.e., when the decoding matrix is full rank. As the decoding matrix in (1) is unstructured for RLNC, $\mathcal{O}(M^3 + M^2K)$ operations are required for GE over the finite field, where an *operation* refers to a multiply-and-add finite field operation. If $\mathbb{F}_2$ is used, all operations reduce to taking exclusive-or's (XORs).

RLNC is capacity achieving in the asymptotic regime; it achieves the minimum expected completion time $T_{\min} = \frac{M}{C_1 + C_2}$ when M and q are arbitrarily large [5].

2.3 Systematic 2-hop network coding

The proposed scheme, systematic 2-hop network coding (S2HNC), for the network topology of Fig. 1 refers to where the first M_1 and M_2 packets sent from S to A and B, respectively, are uncoded source packets. After that, RLNC-coded packets (i.e., random linear combinations of packets) are sent until D successfully decodes all M source packets. At both A and B, every received packet is stored in a buffer. If A or B receives an uncoded packet, the packet is forwarded to D; if no packet is received or the received packet is RLNC coded, a RLNC packet is generated from the buffer for transmission to D. The ordered pair (M_1, M_2) denotes an *uncoded packet allocation* of the S2HNC scheme; different choices of (M_1, M_2) result in different performances as we will show. The decoding at D is similar to that of RLNC except that some received packets may be uncoded, which means that some rows in the decoding matrix are singleton, i.e., contain only one nonzero element.

3 S2HNC for two-hop lossy transmission

We now consider using only one path for transmission of M source packets. Packets from S are all sent to D via A where the loss probabilities of the two hops are ϵ_1 and ϵ_2, respectively, as shown in Fig. 1. We analyze the expected end-to-end rates of S2HNC and RLNC and show that the rate of S2HNC is always higher than that of RLNC, while the required decoding cost of S2HNC is lower. For ease of exposition, we analyze the expected completion times of the two schemes. The end-to-end rate is inversely proportional to the completion time.

3.1 Expected completion time of S2HNC

It is known that as M gets large, RLNC achieves the max-flow capacity of the network [5]. In practice, however, where M may not be large, we would experience some loss in the end-to-end rate. In this section, we employ a Markov chain model to calculate the expected completion time of S2HNC in the small M regime for given link erasure rates. In [25], a Markov chain model is proposed to analyze the two-hop network-coded transmission, where RLNC of infinite field size is assumed. In the following, we use a similar approach but applied to finite field sizes.

We use Markov chains C1 and C2 to characterize the process of D collecting innovative packets. The one denoted as C1 is for the period of time when S is sending uncoded packets and the other one, C2, is for when S is sending coded packets. The state spaces of the two Markov chains are the same and can be expressed in a two-tuple state, (k, r), where k and r represent the numbers of innovative packets received by A and D at the beginning of each S-A transmission, respectively. The state transitions that may occur at each step is shown in Fig. 2. It is noted that $r \leq k$ because D cannot receive more information than A at any time. This results in a total of $n = \frac{(M+1)(M+2)}{2}$ two-tuple states.

Fig. 2 Possible state transitions of the Markov chain model

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 4 of 14

When S transmits uncoded packets, the probability that k increases by 1 after a network use is $p_{k \to k+1} = 1 - \epsilon_1$ for $k < M$, i.e., a successful transmission on S-A and 0 for $k = M$. The increment of r is dependent on k's evolution. For $k < M$, we have $p_{r \to r+1|k \to k+1} = 1 - \epsilon_2$ for a successful forward by A or $p_{r \to r+1|k \to k} = (1 - \epsilon_2)\left(1 - \frac{1}{q^{k-r}}\right)$ for a successful transmission of an innovative packet coded from A's buffer. The term $\frac{1}{q^{k-r}}$ is the probability that a uniformly distributed random k-dimensional vector over $\mathbb{F}_q$ lies in the span of r linearly independent k-dimensional vectors. For $k = M$, we have $p_{r \to r+1|k \to k} = (1 - \epsilon_2)\left(1 - \frac{1}{q^{M-r}}\right)$. The evolution of (k, r) for C1 can then be characterized. Denoting $p_{(k,r) \to (k+1,r+1)}$ as the probability that k and r are each increased by 1 after a network use, it is seen that it is equal to $p_{k \to k+1} p_{r \to r+1|k \to k+1}$ for $r \le k < M$. Similarly, we can obtain $p_{(k,r) \to (k,r)}$, $p_{(k,r) \to (k+1,r)}$, and $p_{(k,r) \to (k,r+1)}$.

When S starts sending coded packets, the transition probabilities change as follows: when in state (k, r), k is increased with probability $p_{k \to k+1} = (1 - \epsilon_1)\left(1 - \frac{1}{q^{M-k}}\right)$ for $k \le M$. When $k < M$, the probability that r increases is $p_{r \to r+1|k \to k} = (1 - \epsilon_2)\left(1 - \frac{1}{q^{k-r}}\right)$ and $p_{r \to r+1|k \to k+1} = (1 - \epsilon_2)\left(1 - \frac{1}{q^{k+1-r}}\right)$, where the term $\frac{1}{q^{k+1-r}}$ is due to one more innovative packet received at A before it generates a coded packet. When $k = M$, $p_{r \to r+1|k \to k} = (1 - \epsilon_2)\left(1 - \frac{1}{q^{M-r}}\right)$. With the above probabilities, we can obtain $p_{(k,r) \to (k,r)}$, $p_{(k,r) \to (k+1,r)}$, $p_{(k,r) \to (k,r+1)}$, and $p_{(k,r) \to (k+1,r+1)}$ for C2. Expressions for the transition probabilities of C1 and C2 are provided in Appendix 1.

The calculation of the expected completion time of S2HNC consists of determining the state of C1 after exactly M steps (i.e., M network uses) starting from $(0, 0)$ and the expected number of additional steps that are needed for C2 to transit from that state to (M, M). To simplify the computation, we label each two-tuple state with a single index. We order the states as $(0, 0)$, $(1, 0)$, $(1, 1)$, $(2, 0)$, $(2, 1)$, ..., $(M, M - 1)$, (M, M), and we label state (k, r) using index $i = \frac{k(k+1)}{2} + r$, i.e., we relabel states as $0, 1, 2, 3, \dots$ instead of $(0, 0)$, $(1, 0)$, $(1, 1)$, $(2, 0), \dots$. Below, we will interchangeably use i or (k, r) to refer to a two-tuple state. A consequence of this labeling choice is that state i cannot transit to a state $j < i$. With this notation, we can express the transition matrix of C1 as Π_1, which is an upper-triangular matrix with elements π^1_{ij} denoting the probability of transition from state i to j. Similarly, the transition probabilities of C2 form the state transition matrix Π_2.

After M steps starting from $(0, 0)$ with transition matrix Π_1, the probability that the system is in state i is equal to the ith element of $\mathbf{e}_{1n} \Pi_1^M$, where $\mathbf{e}_{1n}$ is a length-

n $\left(n = \frac{(M+1)(M+2)}{2}\right)$ row vector with all-zero elements except the first element being a 1. The probability vector $\mathbf{e}_{1n} \Pi_1^M$ is the "input" of C2, which is an absorbing Markov chain in which (M, M) is the only absorbing state. It is easy to verify that Π_1 and Π_2 can be expressed in the form $\begin{bmatrix} Q_i & R_i \\ 0 & 1 \end{bmatrix}$, $i = 1, 2$, respectively. Q_1 and Q_2 are each of size $t \times t$; $t = n - 1$ is the number of transient states. The expected number of steps, Δ_M, that is needed by C2 to enter the absorbing state given $\mathbf{e}_{1n} \Pi_1^M$ as the initial probability vector of states is [26]

$$\Delta_M = \mathbf{e}_{1t} Q_1^M (I_t - Q_2)^{-1} \mathbf{1}_t, \tag{2}$$

where I_t is a $t \times t$ identity matrix and $\mathbf{1}_t$ is a length-t column vector whose entries are all 1. The expected completion time of S2HNC is then

$$\mathrm{E}\{T\} = M + \Delta_M. \tag{3}$$

We note that the cost of computing the expected completion time from (2) could be high for large M, even though that Q_1 and Q_2 are very sparse. For example, the computation of Δ_M for $M = 500$ involves matrices Q_1 and Q_2 that each of dimension 125,750. However, since S2HNC asymptotically achieves the max-flow capacity for large M, we only require the above calculation of expected completion time for small M, which is feasible.

Note that the above analysis includes the calculation of expected time for RLNC scheme as a special case, in which no uncoded packets are sent and C2 with initial state $(0, 0)$ characterizes the whole transmission process. For RLNC, the expected number of steps before the process entering the absorbing state is then equal to $\mathbf{e}_{1t}(I_t - Q_2)^{-1} \mathbf{1}_t$.

The variances of the completion time can also be obtained. Note that the initial state probabilities of C2 are $\mathbf{e}_{1n} \Pi_1^M$ for S2HNC and are $\mathbf{e}_{1n}$ for RLNC. Let $N = (I_t - Q_2)^{-1}$; we have [26]

$$\mathrm{Var}\{T\} = \mathbf{v}(2N - I_t)N\mathbf{1}_t - (\mathbf{v}N\mathbf{1}_t)^2, \tag{4}$$

where $\mathbf{v}$ is $\mathbf{e}_{1t} Q_1^M$ for S2HNC and $\mathbf{v}$ is $\mathbf{e}_{1t}$ for RLNC.

3.2 Advantage in end-to-end rate

Based on the above analysis, we next establish that S2HNC has a shorter expected completion time and therefore achieves higher end-to-end rate than RLNC. We achieve this by comparing the probability that a received packet is innovative for D using S2HNC or RLNC. S2HNC will be shown to have a larger such probability at any time than RLNC.

Since S2HNC has the same behavior (which is characterized by the Markov chain C2) as RLNC after M transmissions from S, we only need to compare the two schemes during the first M transmissions from S. We consider S2HNC first. Recall that for the state (k, r), k and r denote the numbers of innovative packets that have been received by A and D at the beginning of each S-A transmission, $r \leq k$. During the first M transmissions from S, the probability that D receives an innovative packet after two successive transmissions on S-A and A-D is equal to

$$p_{\text{S2HNC}}(k, r) = (1 - \epsilon_2) \left[(1 - \epsilon_1) + \epsilon_1 \left(1 - \frac{q^r}{q^k} \right) \right], \tag{5}$$

which corresponds to when an uncoded packet is successfully forwarded (which is innovative for D with probability 1) or when the coded packet from the k innovative packets at A is innovative for D.

Now, consider the RLNC scheme. For the same k and r, the probability that a new innovative packet is received by D after two successive transmissions on S-A and A-D is equal to

$$p_{\text{RLNC}}(k, r) = (1 - \epsilon_2) \left\{ (1 - \epsilon_1) \left[\left(1 - \frac{q^k}{q^M} \right) \left(1 - \frac{q^r}{q^{k+1}} \right) \right. \right.$$
$$\left. \left. + \frac{q^k}{q^M} \left(1 - \frac{q^r}{q^k} \right) \right] + \epsilon_1 \left(1 - \frac{q^r}{q^k} \right) \right\}, \tag{6}$$

where the bracketed term $\left(1 - \frac{q^k}{q^M} \right) \left(1 - \frac{q^r}{q^{k+1}} \right) + \frac{q^k}{q^M} \left(1 - \frac{q^r}{q^k} \right)$ corresponds to when the coded packet sent from A is innovative for D when a packet was successfully received by A.

From (5) and (6), we have $p_{\text{S2HNC}}(k, r) > p_{\text{RLNC}}(k, r)$ because

$$\left(1 - \frac{q^k}{q^M} \right) \left(1 - \frac{q^r}{q^{k+1}} \right) + \frac{q^k}{q^M} \left(1 - \frac{q^r}{q^k} \right) < 1.$$

3.3 Advantage in decoding complexity

For M source packets, it is expected that $M_u = (1 - \epsilon_1) \times (1 - \epsilon_2)M$ uncoded packets are received by D using S2HNC. Therefore, S2HNC only needs to decode the remaining $M - M_u$ packets from RLNC-coded packets using GE. For $K \gg M$, the number of operations that are needed using GE is approximately $M^2 K$ for RLNC and is $(M - M_u)^2 K$ for S2HNC. This corresponds to a savings of a factor of $\frac{M^2 K}{(M - M_u)^2 K} = \frac{1}{(\epsilon_1 + \epsilon_2 - \epsilon_1 \epsilon_2)^2}$ computations. This is a considerable saving compared to RLNC when ϵ_1 and ϵ_2 are small as in many system scenarios encountered in practice.

4 S2HNC for parallel two-hop transmission

In this section, we extend the results from a single-path to a parallel-path scenario. We quantify the S2HNC scheme's savings in the expected completion time and the amount of computation over that of RLNC, as a function of given packet loss probabilities.

4.1 Completion time

Lemma 1. *A necessary condition for S2HNC to achieve a shorter expected completion time than RLNC is that the transmitted uncoded packets on the two paths be distinct.*

Proof. Let M_1 and M_2 be the numbers of packets transmitted on the two paths, respectively. Any $M = M_1 + M_2$ received RLNC packets at D are mutually linearly independent as $q \to \infty$. Therefore, the RLNC scheme is expected to finish at $T_{\min} = \frac{M}{C_1 + C_2}$ with $(R_1 + R_2)T_{\min}$ packets sent from S as $q \to \infty$ [5]. Now, assume that the same number of packets are sent from S using S2HNC, among which there are $M_1 + M_2$ uncoded packets. At D, this results in $M_u = (1 - \epsilon_1)(1 - \epsilon_2)M_1 + (1 - \delta_1)(1 - \delta_2)M_2$ expected number of uncoded packets and $M - M_u$ coded packets. As $q \to \infty$, decoding is successful if and only if the M_u uncoded packets are distinct. However, if any uncoded packet is sent more than once, then there is a constant probability (depending on packet loss probabilities but not on q) that would not decrease to zero as $q \to \infty$ that an uncoded packet is received more than once at D, i.e., the probability that the M_u packets are distinct does not approach zero as $q \to \infty$, resulting in a non-zero decoding failure probability at $T_{\min}$. The expected completion time of such a S2HNC scheme cannot be shorter than that of RLNC. □

Theorem 1. *Any S2HNC-uncoded packet allocation with distinct uncoded packets transmitted on the two paths has shorter expected completion time than the RLNC scheme.*

Proof. See Appendix 2. □

In the sequel, S2HNC is fashioned to send distinct uncoded packets on the two paths. Note that Theorem 1 holds regardless of the values of M, q and the packet loss probabilities of the links. RLNC's minimum completion time is achieved asymptotically as $q \to \infty$. In the finite regime when q may be small, S2HNC has a shorter expected completion time.

Corollary 1. *The allocation (M_1, M_2) that results in the greatest expected number of uncoded packets received at D has the shortest expected completion time among all S2HNC-uncoded packet allocations.*
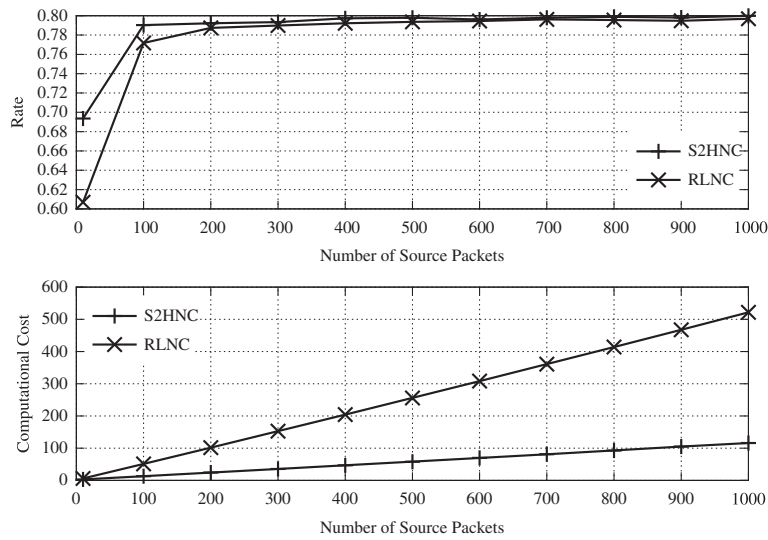
Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 6 of 14

Fig. 3 End-to-end rate and decoding complexity for $\epsilon_1 = 0.05$, $\epsilon_2 = 0.2$

Proof. The proof follows straightforwardly from the proof of Theorem 1, in which it is shown that any received uncoded packet at D is innovative with a higher probability compared to that if a coded packet were received. □

4.2 Computational cost and uncoded packet allocation

Depending on how many uncoded packets are received by D, S2HNC exhibits a different computational cost compared to that of RLNC. Assume that M_u uncoded packets are received, then D needs to decode only the remaining $M - M_u$ RLNC-coded packets. This corresponds to a computational savings by a factor of $\frac{M^2}{(M-M_u)^2}$ in solving the linear system of equations.

To reduce computational cost, we can maximize M_u. Based on Corollary 1, maximizing M_u results in an uncoded packet allocation that achieves the minimum expected completion time. The maximization can be formulated as an optimization problem in the variables M_1 and M_2. The optimization, however, requires knowledge of the link parameters R_1, R_2, ϵ_1, ϵ_2, δ_1, and δ_2. The allocation problem is as follows:
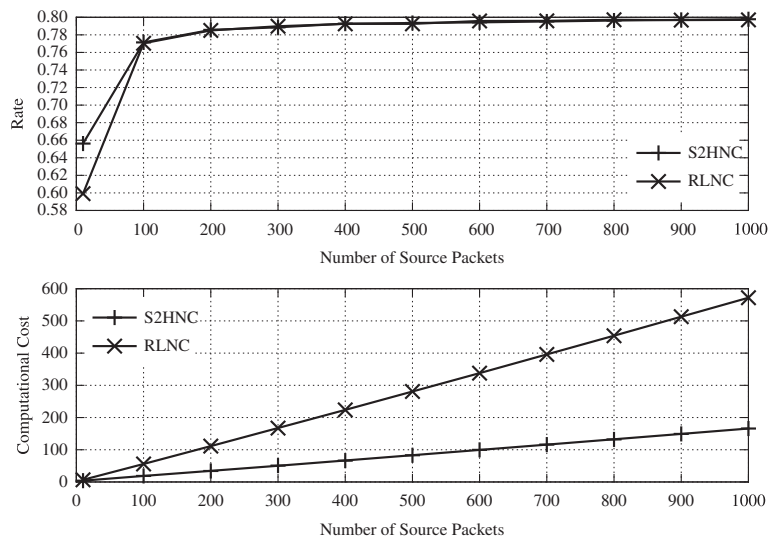

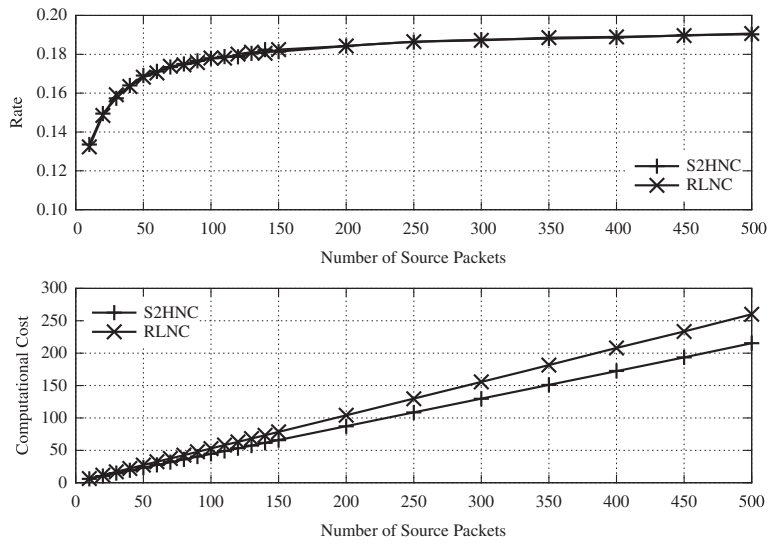**Fig. 4** End-to-end rate and decoding complexity for $\epsilon_1 = 0.2$, $\epsilon_2 = 0.05$

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 7 of 14

Fig. 5 End-to-end rate and decoding complexity for $\epsilon_1 = \epsilon_2 = 0.8$

$$\underset{M_1,M_2}{\text{maximize}} \quad (1-\epsilon_1)(1-\epsilon_2)M_1 + (1-\delta_1)(1-\delta_2)M_2$$

$$\text{subject to} \quad M_1 + M_2 \leq M$$

$$\frac{M_1}{R_1} \leq \frac{M}{C_1 + C_2}$$

$$\frac{M_2}{R_2} \leq \frac{M}{C_1 + C_2}$$

$$M_1, M_2 \in \mathbb{Z}^*,$$

(7)

where the first constraint ensures that at most M distinct uncoded packets are sent and the next two constraints ensure that uncoded packets on either path be transmitted within the first $T_{\min}$ network uses. These constraints

arise to avoid uncoded packets being "over-allocated" to either path and ensure that all distinct uncoded packets be transmitted before D finishes decoding.

We observe that $M_1 + M_2 = M$ always holds according to Corollary 1, because S has to send at least M packets and that sending distinct uncoded packets is always superior to sending coded ones. We can therefore solve (7) and obtain M_1 in closed form as

$$M_1 = \begin{cases} \lfloor \frac{MR_1}{C_1+C_2} \rfloor & (1-\epsilon_1)(1-\epsilon_2) > (1-\delta_1)(1-\delta_2) \\ M - \lfloor \frac{MR_2}{C_1+C_2} \rfloor & \text{otherwise,} \end{cases}$$
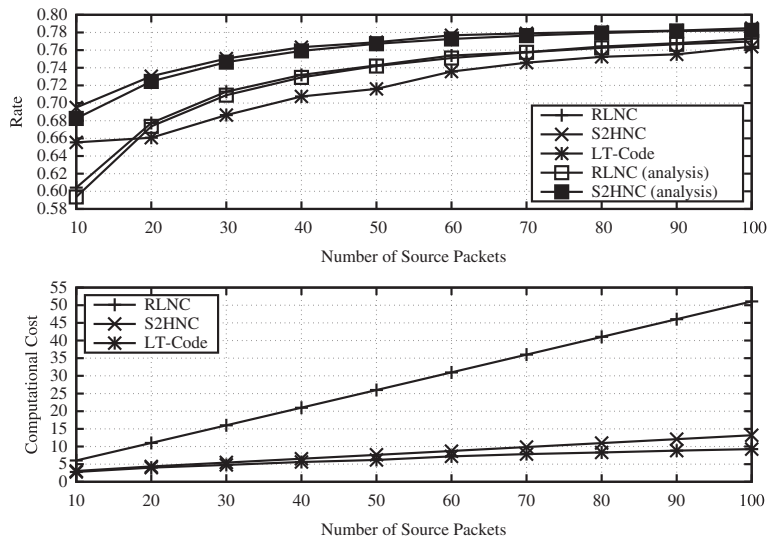
(8)

Fig. 6 Comparison of S2HNC and RLNC for small M, $\epsilon_1 = 0.05$, $\epsilon_2 = 0.2$

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93
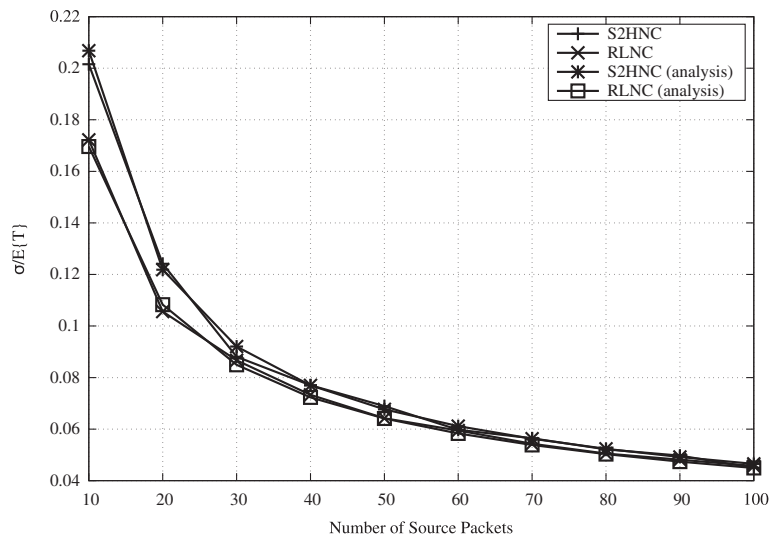
Page 8 of 14

Fig. 7 Expected completion times scaled by standard deviations of S2HNC and RLNC for small M, $\epsilon_1 = 0.05$, $\epsilon_2 = 0.2$

and $M_2 = M - M_1$, where $\lfloor \cdot \rfloor$ is the floor function. The solution corresponds to that one of the second and the third inequalities of (7) achieves equality.

With M_1 and M_2 as above, S can choose any disjoint subsets of source packets of sizes M_1 and M_2 as the initially transmitted packets on S-A and S-B, respectively.

5 Numerical and simulation results

In this section we simulate the performance of S2HNC and compare it with that of RLNC in the one-path and two-path scenarios. The simulations use $K = 8192$ bits per packet and coding is performed in $\mathbb{F}_2 = \{0, 1\}$. Each simulated curve represents averaging over 10,000 Monte Carlo trials. We count the number of operations, N_{ops}, that are needed to recover all the data in the trials. The computational cost measure we use is the average number of operations per bit for successful decoding, which is equal to $\frac{N_{\text{ops}}}{MK}$.

We first consider the single-path case. We compare S2HNC with RLNC in the cases where $\epsilon_1 = 0.05$, $\epsilon_2 = 0.2$ and $\epsilon_1 = 0.2$, $\epsilon_2 = 0.05$, which correspond to where the
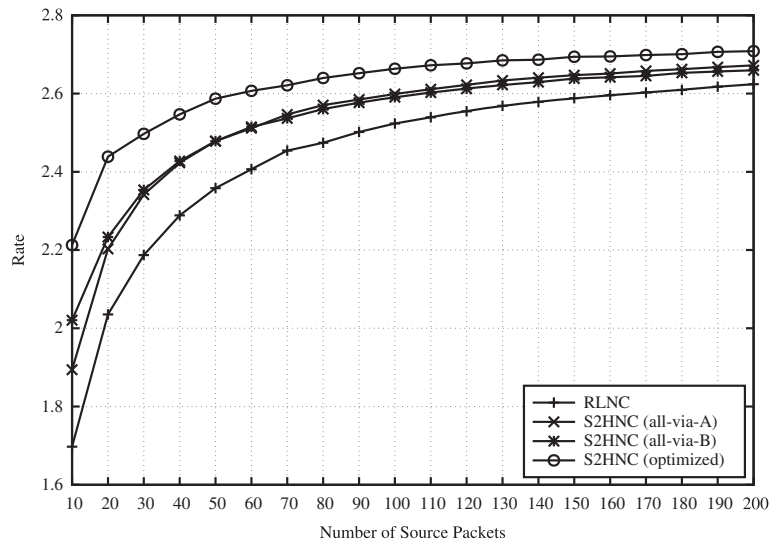

Fig. 8 Rate of using RLNC and S2HNC as a function of the number of source packets, M, for several uncoded packet allocation schemes, and $R_1 = 1$, $R_2 = 2$, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.05$, $\delta_1 = 0.01$, $\delta_2 = 0.1$
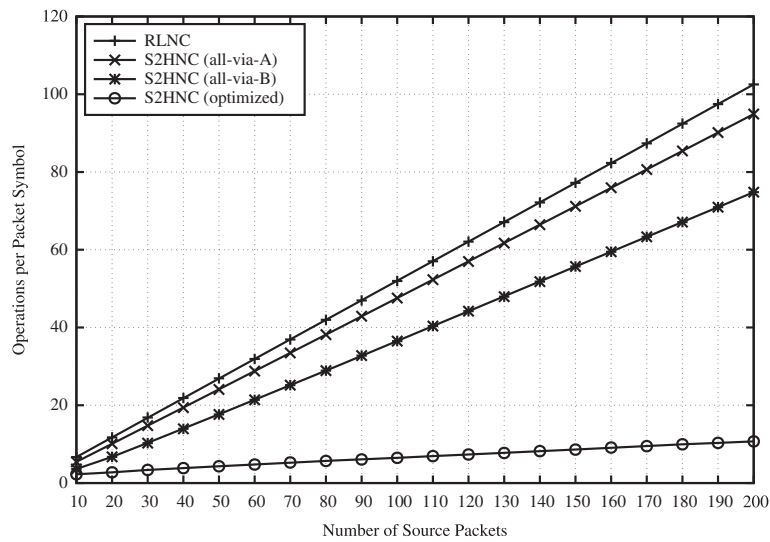
Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 9 of 14

Fig. 9 Computational costs of using RLNC and S2HNC as a function of number of source packets, M, for several uncoded packet allocation schemes, and $R_1 = 1, R_2 = 2, \epsilon_1 = 0.01, \epsilon_2 = 0.05, \delta_1 = 0.01, \delta_2 = 0.1$

quality of the first hop is better and worse than that of the second hop, respectively. The max-flow capacity of the two-hop link is 0.8 packets per network use for both cases. The number of packets varies from $M = 10$ to 1000.

The rate and complexity performances are shown in Figs. 3 and 4. Clearly, the rate of S2HNC is always higher, even though the gap narrows and both schemes approach capacity as M increases. The lower decoding complexity of S2HNC is quite obvious. It is seen that the rate improvement of S2HNC is more significant in Fig. 3. This is not surprising since more uncoded packets are received at A when the first hop link is better and packets are always innovative for D when using S2HNC. On the other hand, if the first hop is worse, S2HNC resembles RLNC at A because more coded packets would be sent. In practice, the former case is commonly found in applications such as WiFi offloading, where the first hop is the link from a packet gateway to either a WiFi access point or base station.

We note that when the link quality is poor, i.e., the erasure rates of S-A and A-D are high, the performance gap between S2HNC and RLNC shrinks. An example of this is shown in Fig. 5, where the two hops have equal erasure probability 0.8 and the performances of S2HNC and

RLNC are almost identical. This is anticipated because the advantage of S2HNC comes from its use of uncoded packets, which are innovative for downstream nodes with probability one while requiring no decoding. As an extreme case when links are highly lossy, very few uncoded packets would be received and S2HNC reduces to RLNC.

We observe above the expected result that both S2HNC and RLNC approach link capacity as M increases. However, when M is small, they perform far below capacity. We further investigate this in Fig. 6, where the rates and computational costs of the two schemes at small numbers of packets $M = 10$ to 100 are compared. As a comparison point with an established sparse coding method, in this scenario, we also include a simulation of the LT code proposed in [10]. A simulation of the LT-coded scheme proposed in [10] is included, where LT-coded packets are transmitted from S, and A either forwards successfully received packets or generates RLNC-re-encoded packets from its buffer. Here, the decoding of LT-coded packets is done by inactivation decoding [27], which finishes at the first instance that a full-rank decoding matrix is received, i.e., the achievable rate of the LT-coded scheme is maximized in the plot. In the low M and q regime, the rate improvement of S2HNC over the other two schemes is quite obvious. The LT-coded scheme has the lowest complexity due to its sparse nature but has the lowest rate.

The expected rates, $M/\mathrm{E}\{T\}$, of S2HNC and RLNC are also provided in Fig. 6, where the expected completion times are calculated using the analysis results. The analytical rates match simulations closely. Figure 7 plots $\sigma/\mathrm{E}\{T\}$ for the M's, where σ is the standard deviation

Table 1 Optimized uncoded packet allocations for $R_1 = 1, R_2 = 2,$ $\epsilon_1 = 0.01, \epsilon_2 = 0.05, \delta_1 = 0.01, \delta_2 = 0.1$, as a function of the number of source packets, M

M	20	40	60	80	100	120	140	160	180	200
M_1	7	14	21	29	36	43	50	58	65	72
M_2	13	26	39	51	64	77	90	102	115	128

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93
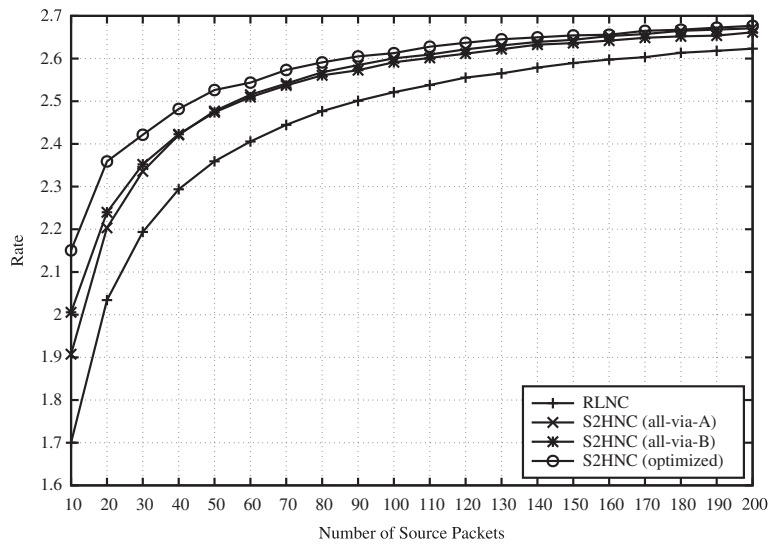
Page 10 of 14

Fig. 10 Rate of using RLNC and S2HNC as a function of the number of source packets, *M*, for several uncoded packet allocation schemes, and $R_1 = 1, R_2 = 2, \epsilon_1 = 0.01, \epsilon_2 = 0.05, \delta_1 = 0.1, \delta_2 = 0.01$

(i.e., square root of the variance) of the completion time. We show both analytical results calculated using (4) and the simulated results. From Fig. 7, it is seen that the standard deviations of the completion times of the different methods are comparable and small relative to the completion times themselves, confirming that S2HNC's rate does not vary significantly and is at least that of RLNC.

We now compare the performances of RLNC and S2HNC for the parallel-path scenario. The first set of network parameters for simulations are $R_1 = 1$, $R_2 = 2$, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.05$, $\delta_1 = 0.01$, and $\delta_2 = 0.1$. In Figs. 8 and 9, we show the averaged end-to-end rate and computational cost of the two schemes, respectively. The uncoded packet allocation solutions of (7) for some *M*'s are given in Table 1. In addition to the optimized allocation obtained by solving (7), for comparison, we also include two other allocations, namely all-via-A and all-via-B in which uncoded packets are only sent to A and B, respectively. Achieving the minimum completion time
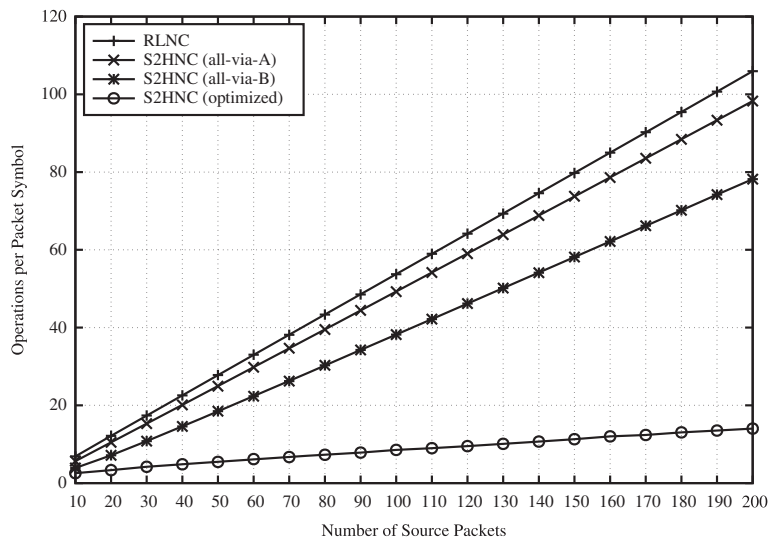

Fig. 11 Computational costs of using RLNC and S2HNC as a function of the number of source packets, *M*, for several uncoded packet allocation schemes, and $R_1 = 1, R_2 = 2, \epsilon_1 = 0.01, \epsilon_2 = 0.05, \delta_1 = 0.1, \delta_2 = 0.01$
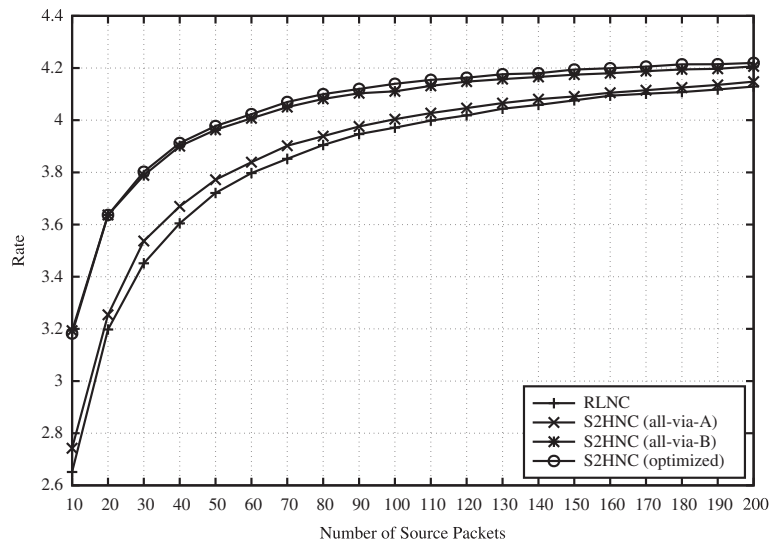
Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 11 of 14

Fig. 12 Rate of using RLNC and S2HNC as a function of number of source packets, *M*, for several uncoded packet allocation schemes, and $R_1 = 1$, $R_2 = 4, \epsilon_1 = 0.01, \epsilon_2 = 0.5, \delta_1 = 0.01, \delta_2 = 0.01$

is equivalent to achieving the max-flow capacity between S and D, which is equal to $C_1 + C_2 = 2.75$ packets per network use for the parameters used.

It is clear that S2HNC when using any of the allocations achieves a higher end-to-end rate than RLNC. Although the gap decreases as *M* increases, the rate improvement is considerable for small *M*. As expected, RLNC requires the most computations in decoding. Different allocations of

uncoded packets result in different computational costs. The optimized allocation results in the least computational cost and a significant saving in computational cost compared to RLNC is obtained. In Figs. 10 and 11, we show the results for the same set of parameters except that we exchange erasure probabilities of the two hops of S-B-D. The optimized allocation in this case is the same as Table 1 according to (8). We see that the improve-

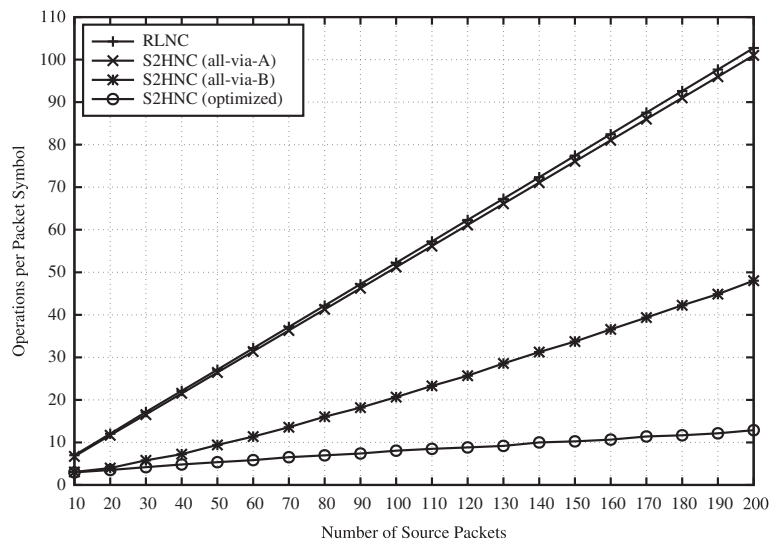**Fig. 13** Computational costs of using RLNC and S2HNC as a function of number of source packets, *M*, for several uncoded packet allocation schemes, and $R_1 = 1, R_2 = 4, \epsilon_1 = 0.01, \epsilon_2 = 0.5, \delta_1 = 0.01, \delta_2 = 0.05$

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 12 of 14

ment slightly shrinks because the first hop of S-B-D is worse.

It is illustrative to present simulation results for a different set of parameter values $R_1 = 1$, $R_2 = 4$, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.5$, $\delta_1 = 0.01$, and $\delta_2 = 0.05$, which corresponds to having one lower quality path in terms of both transmission rate and packet loss rate. The max-flow capacity of the scenario is $C_1 + C_2 = 4.3$ packets per network use. The achieved rates and required computational costs of using RLNC and S2HNC with different uncoded packet allocations are shown in Figs. 12 and 13, respectively. Again, all S2HNC schemes achieve higher rates and lower computational costs than RLNC.

Since the S-B-D path has a higher rate and better quality than S-A-D in this scenario, we expect the all-via-B allocations to have performances close to that of the optimized allocation. The optimal allocation solution is provided in Table 2, where most of the uncoded packets are allocated to the S-B-D path. In Fig. 12, the achieved rates of all-via-B and optimized allocations are almost the same. However, it is seen in Fig. 13 that the two allocations have differences in computational costs at the receiver.

6 Conclusions

In this paper, we proposed a systematic network coding method for transmission over single or parallel two-hop lossy links. For these network topologies, we showed that it is not necessary to perform the high-complexity random linear network coding in order to approach the max-flow capacity. Sending all the uncoded packets first is shown to be advantageous compared to sending randomly coded packets all of the time.

Compared to random linear network coding, which is capacity achieving for large numbers of source packets and sufficiently large finite field sizes, the proposed scheme is observed to provide better performance in terms of both end-to-end rate and decoding complexity. The improvement is appreciable when the number of source packets is small and the binary field is used. A particular interesting application of this operating condition is WiFi offloading, where frequent transmissions of small numbers of data packets are common.

It is noted that the proposed technique may be directly extended to networks that contain more parallel paths and/or more than two hops. The extension, however, is left for future work.

Table 2 Optimized uncoded packet allocations for $R_1 = 1, R_2 = 4$, $\epsilon_1 = 0.01, \epsilon_2 = 0.5, \delta_1 = 0.01, \delta_2 = 0.05$, as a function of number of source packets, M

M	20	40	60	80	100	120	140	160	180	200
M_1	2	3	5	6	7	9	10	12	13	14
M_2	18	37	55	74	93	111	130	148	167	186

Appendix

Appendix 1: transition probabilities of the Markov chains modeling S2HNC

For the period of time when S is sending uncoded packets, we have the following transition probabilities for different combinations of (k, r) in Markov chain C1. We have

$$p_{(0,0)\to(0,0)} = \epsilon_1, \tag{9}$$

$$p_{(0,0)\to(0,1)} = 0, \tag{10}$$

$$p_{(0,0)\to(1,0)} = (1 - \epsilon_1)\epsilon_2, \tag{11}$$

$$p_{(0,0)\to(1,1)} = (1 - \epsilon_1)(1 - \epsilon_2), \tag{12}$$

for $k = r = 0$.

For $k \geq 1$ and $r < k < M$,

$$p_{(k,r)\to(k,r)} = \epsilon_1 \left(\epsilon_2 + (1 - \epsilon_2)\frac{q^r}{q^k} \right), \tag{13}$$

$$p_{(k,r)\to(k,r+1)} = \epsilon_1(1 - \epsilon_2) \left(1 - \frac{q^r}{q^k} \right), \tag{14}$$

$$p_{(k,r)\to(k+1,r)} = (1 - \epsilon_1)\epsilon_2, \tag{15}$$

$$p_{(k,r)\to(k+1,r+1)} = (1 - \epsilon_1)(1 - \epsilon_2). \tag{16}$$

For $k \geq 1$ and $r = k < M$,

$$p_{(k,r)\to(k,r)} = \epsilon_1, \tag{17}$$

$$p_{(k,r)\to(k,r+1)} = 0, \tag{18}$$

$$p_{(k,r)\to(k+1,r)} = (1 - \epsilon_1)\epsilon_2, \tag{19}$$

$$p_{(k,r)\to(k+1,r+1)} = (1 - \epsilon_1)(1 - \epsilon_2). \tag{20}$$

and for $r < k = M$,

$$p_{(k,r)\to(k,r)} = \epsilon_2 + (1 - \epsilon_2)\frac{q^r}{q^k}, \tag{21}$$

$$p_{(k,r)\to(k,r+1)} = (1 - \epsilon_2) \left(1 - \frac{q^r}{q^k} \right), \tag{22}$$

$$p_{(k,r)\to(k+1,r)} = 0, \tag{23}$$

$$p_{(k,r)\to(k+1,r+1)} = 0. \tag{24}$$

For the period of time when S is sending coded packets, i.e., for Markov chain C2, we have

$$p_{(0,0)\to(0,0)} = \epsilon_1 + (1 - \epsilon_1)\frac{1}{q^M}, \tag{25}$$

$$p_{(0,0)\to(0,1)} = 0, \tag{26}$$

$$p_{(0,0)\to(1,0)} = (1 - \epsilon_1)\left(1 - \frac{1}{q^M}\right)\left(\epsilon_2 + (1 - \epsilon_2)\frac{1}{q}\right) \tag{27}$$

$$p_{(0,0)\to(1,1)} = (1 - \epsilon_1)\left(1 - \frac{1}{q^M}\right)(1 - \epsilon_2)\left(1 - \frac{1}{q}\right), \tag{28}$$

for $k = r = 0$.

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 13 of 14

For $k \geq 1$ and $r < k < M$,

$$p_{(k,r)\to(k,r)} = \left(\epsilon_1 + (1-\epsilon_1)\frac{q^k}{q^M}\right)\left(\epsilon_2 + (1-\epsilon_2)\frac{q^r}{q^k}\right),$$
(29)

$$p_{(k,r)\to(k,r+1)} = \left(\epsilon_1 + (1-\epsilon_1)\frac{q^k}{q^M}\right)(1-\epsilon_2)\left(1 - \frac{q^r}{q^k}\right),$$
(30)

$$p_{(k,r)\to(k+1,r)} = (1-\epsilon_1)\left(1 - \frac{q^k}{q^M}\right)\left(\epsilon_2 + (1-\epsilon_2)\frac{q^r}{q^{k+1}}\right),$$
(31)

$$p_{(k,r)\to(k+1,r+1)} = (1-\epsilon_1)\left(1 - \frac{q^k}{q^M}\right)(1-\epsilon_2)\left(1 - \frac{q^r}{q^{k+1}}\right).$$
(32)

For $k \geq 1$ and $r = k < M$,

$$p_{(k,r)\to(k,r)} = \epsilon_1 + (1-\epsilon_1)\frac{q^k}{q^M},$$
(33)

$$p_{(k,r)\to(k,r+1)} = 0,$$
(34)

$$p_{(k,r)\to(k+1,r)} = (1-\epsilon_1)\left(1 - \frac{q^k}{q^M}\right)\left(\epsilon_2 + (1-\epsilon_2)\frac{q^r}{q^{k+1}}\right),$$
(35)

$$p_{(k,r)\to(k+1,r+1)} = (1-\epsilon_1)\left(1 - \frac{q^k}{q^M}\right)(1-\epsilon_2)\left(1 - \frac{q^r}{q^{k+1}}\right).$$
(36)

and for $r < k = M$,

$$p_{(k,r)\to(k,r)} = \epsilon_2 + (1-\epsilon_2)\frac{q^r}{q^k},$$
(37)

$$p_{(k,r)\to(k,r+1)} = (1-\epsilon_2)\left(1 - \frac{q^r}{q^k}\right),$$
(38)

$$p_{(k,r)\to(k+1,r)} = 0,$$
(39)

$$p_{(k,r)\to(k+1,r+1)} = 0.$$
(40)

Appendix 2: proof of Theorem 1

We prove the theorem by examining the probability that a received packet at D is innovative with respect to (w.r.t.) previously received packets. S2HNC will be shown to have a larger such probability at any time than RLNC. The completion time is inversely proportional to this probability because the process of D collecting innovative packets is Markovian, i.e., the probability whether a received packet is innovative only depends on the number of innovative packets that D has received so far.

Consider a S2HNC with fixed M_1 and M_2. In the first $T_1 = \min\left(\frac{M_1}{R_1}, \frac{M_2}{R_2}\right)$ network uses, packets transmitted on paths S-A and S-B are all distinct uncoded packets. The information flows on the two paths are therefore independent. Using the results from Section 3, it is clear that on

either path, using S2HNC will result in more innovative packets at D than using RLNC after T_1 network uses.

During the time period from T_1 to $T_2 = \max\left(\frac{M_1}{R_1}, \frac{M_2}{R_2}\right)$, one of the paths will be transmitting RLNC packets coded over the M source packets while the other one is transmitting uncoded packets. Without loss of generality, we assume that S-A is the one transmitting RLNC packets, i.e., $\frac{M_1}{R_1} < \frac{M_2}{R_2}$. Now, the proof reduces to showing that transmitting uncoded packets on S-B while transmitting RLNC-coded packets on S-A is indeed beneficial compared to transmitting RLNC-coded packets on both S-B and S-A.

Let $r_1^{(T_1)}$ and $r_2^{(T_1)}$ denote the numbers of innovative packets that have been received by D through S-A-D and S-B-D after T_1 network uses, respectively. Let $r_1^{(t)}$ and $r_2^{(t)}$ denote the numbers of innovative packets that are received through S-A-D and S-B-D during (T_1, t), $T_1 \leq t \leq T_2$. A newly received packet after t network uses is innovative if and only if it is linearly independent w.r.t. all the $r_1^{(T_1)} + r_2^{(T_1)} + r_1^{(t)} + r_2^{(t)}$ packets. For a given number of innovative packets at B, if S-B was unsuccessful, the probability that D receives an innovative packet from B in the next transmission is the same using S2HNC or RLNC. If S-B was successful when using S2HNC, however, a new uncoded packet would be received by B. This packet is innovative with probability 1 w.r.t. the previously received innovative packets at B (which are uncoded) and is also innovative w.r.t. the $r_1^{(T_1)}, r_2^{(T_1)}$, and $r_2^{(t)}$ innovative packets that had been received at D because those $r_1^{(T_1)} + r_2^{(T_1)} + r_2^{(t)}$ packets are in the span of the other distinct uncoded packets. The forwarded uncoded packet is non-innovative if and only if it is in the span of the $r_1^{(t)}$ RLNC packets. The probability of such event is no larger than that when using RLNC because a RLNC-coded packet may also be in the span of the other $r_1^{(T_1)} + r_2^{(T_1)} + r_2^{(t)}$ packets. Again, uncoded packets are innovative for B with probability 1 and therefore B collects more innovative packets than the RLNC scheme. Therefore, transmitting uncoded packets on S-B is advantageous.

Therefore, in the first T_2 network uses, S2HNC always results in more innovative packets at A, B, and D compared to RLNC. After T_2, S2HNC and RLNC have the same behavior. Altogether, S2HNC achieves a shorter completion time.

Li *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:93

Page 14 of 14

References

1. R Ahlswede, N Cai, S-YR Li, RW Yeung, Network information flow. IEEE Trans. Inf. Theory. **46**(4), 1204–1216 (2000). doi:10.1109/18.850663
2. S-YR Li, RW Yeung, N Cai, Linear network coding. IEEE Trans. Inf. Theory. **49**(2), 371–381 (2003). doi:10.1109/TIT.2002.807285
3. R Koetter, M Medard, An algebraic approach to network coding. IEEE/ACM Trans. Netw. **11**(5), 782–795 (2003). doi:10.1109/TNET.2003.818197
4. RW Yeung, *Information Theory and Network Coding*. (Springer, New York, 2008)
5. T Ho, M Medard, R Koetter, DR Karger, M Effros, J Shi, B Leong, A random linear network coding approach to multicast. IEEE Trans. Inf. Theory. **52**(10), 4413–4430 (2006). doi:10.1109/TIT.2006.881746
6. H Shojania, B Li, in *Proc. 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. Random network coding on the iPhone: fact or fiction? (2009), pp. 37–42. doi:10.1145/1542245.1542255, http://doi.acm.org/10.1145/1542245.1542255
7. J Heide, MV Pedersen, FHP Fitzek, T Larsen, in *Proc. IEEE Int. Conf. Communications Workshops*. Network coding for mobile devices—systematic binary random rateless codes, (2009), pp. 1–6. doi:10.1109/ICCW.2009.5208076
8. B Shrader, NM Jones, in *Proc. IEEE Military Communications Conf. MILCOM*. Systematic wireless network coding, (2009), pp. 1–7. doi:10.1109/MILCOM.2009.5380081
9. DE Lucani, M Medard, M Stojanovic, in *Prof. IEEE International Symposium on Information Theory (ISIT)*. Systematic network coding for time-division duplexing, (2010), pp. 2403–2407. doi:10.1109/ISIT.2010.5513768
10. P Pakzad, C Fragouli, A Shokrollahi, in *Proc. IEEE Int. Symp. Information Theory*. Coding schemes for line networks, (2005), pp. 1853–1857
11. M Luby, in *Proc. 43rd Annual IEEE Symp. Foundations of Computer Science*. LT codes, (2002), pp. 271–280. doi:10.1109/SFCS.2002.1181950
12. G Giacaglia, X Shi, M Kim, D Lucani, M Médard, Systematic network coding with the aid of a full-duplex relay. CoRR abs/1204.0034, 3312–3317 (2013). doi: 10.1109/ICC.2013.6655057
13. 3G LTE WiFi Offload Framework, White Paper (2011). http://www.qualcomm.com/media/documents/3g-lte-wifi-offload-framework
14. A Ford, C Raiciu, M Handley, O Bonaventure, TCP extensions for multipath operation with multiple addresses. IETF, RFC6824 (2013). http://www.ietf.org/rfc/rfc6824.txt
15. MAP Gonzalez, T Higashino, M Okada, in *Information Networking (ICOIN), 2013 International Conference On*. Radio access considerations for data offloading with multipath tcp in cellular/wifi networks, (2013), pp. 680–685. doi:10.1109/ICOIN.2013.6496709
16. J Cloud, F du Pin Calmon, W Zeng, G Pau, LM Zeger, M Medard, in *Proc. IEEE 78th Vehicular Technology Conference (VTC Fall)*. Multi-path TCP with network coding for mobile devices in heterogeneous networks, (2013), pp. 1–5. doi:10.1109/VTCFall.2013.6692295
17. JK Sundararajan, D Shah, M Medard, S Jakubczak, M Mitzenmacher, J Barros, Network coding meets TCP: theory and implementation. Proc. IEEE. **99**(3), 490–512 (2011). doi:10.1109/JPROC.2010.2093850
18. PA Chou, Y Wu, K Jain, in *Proc. 41st Allerton Conference on Communication, Control, and Computing, Vol. 41, No. 1*. Practical network coding, (2003), pp. 40–49
19. P Maymounkov, *Online codes*. (Technical report, New York University, 2002)
20. D Silva, W Zeng, FR Kschischang, in *Proc. Workshop Network Coding, Theory, and Applications (NetCod)*. Sparse network coding with overlapping classes, (2009), pp. 74–79. doi:10.1109/NETCOD.2009.5191397
21. Y Li, E Soljanin, P Spasojevic, Effects of the generation size and overlap on throughput and complexity in randomized linear network coding. IEEE Trans. Inf. Theory. **57**(2), 1111–1123 (2011). doi:10.1109/TIT.2010.2095111
22. A Heidarzadeh, AH Banihashemi, in *Proc. IEEE Information Theory Workshop (ITW)*. Overlapped chunked network coding, (2010), pp. 1–5. doi:10.1109/ITWKSPS.2010.5503153
23. Y Li, W-Y Chan, SD Blostein, in *Proc. International Symposium on Network Coding (NetCod)*. Network coding with unequal size overlapping generations, (2012), pp. 161–166. doi: 10.1109/NETCOD.2012.6261902
24. Y Li, SD Blostein, W-Y Chan, in *Proc. IEEE Globecom International Workshop on Cloud Computing Systems, Networks, and Applications*. Large file distribution using efficient generation-based network coding, (2013), pp. 427–432. doi: 10.1109/GLOCOMW.2013.6825025
25. X Shi, M Medard, DE Lucani, Whether and where to code in the wireless packet erasure relay channel. IEEE J. Selected Areas Commun. **31**(8), 1379–1389 (2013). doi:10.1109/JSAC.2013.130803
26. JG Kemeny, JL Snell, *Finite Markov Chains*. (D. Van Nostrand Company, Inc., Princeton, New Jersey, 1976)
27. A Shokrollahi, M Luby, Raptor codes. Found. Trends Commun. Inf. Theory. **6**(13-4), 213–322 (2009)