Linear network coding in convergecast of wireless sensor networks: friend or foe?

Zhenzhou Tang[1,2], Hongyu Wang[1], Qian Hu[2] and Xiukai Ruan[2]
[1] School of Information and Communication Engineering, Dalian University of Technology
Dalian, Liaoning Province - China
e-mail: mr.tangzz@gmail.com]
[2] College of Physics and Electronic Information Engineering, Wenzhou University
Wenzhou, Zhejiang - China
* Corresponding author: Zhenzhou Tang

Received May 3, 2014; revised July 5, 2014; accepted July 29, 2014; published September 30, 2014

Abstract

Convergecast is probably the most common communication style in wireless sensor networks (WSNs). And linear network coding (LNC) is a promising concept to improve throughput or reliability of convergecast. Most of the existing works have mainly focused on exploiting these benefits without considering its potential adverse effect. In this paper, we argue that LNC may not always benefit convergecast. This viewpoint is discussed within four basic scenarios: LNC-aided and none-LNC convergecast schemes with and without automatic repeat request (ARQ) mechanisms. The most concerned performance metrics, including packet collection rate, energy consumption, energy consumption balance and end-to-end delay, are investigated. Theoretical analyses and simulation results show that the way LNC operates, i.e., conscious overhearing and the prerequisite of successfully decoding, could naturally diminish its advantages in convergecast. And LNC-aided convergecast schemes may even be inferior to none-LNC ones when the wireless link delivery ratio is high enough. The conclusion drawn in this paper casts a new light on how to effectively apply LNC to practical WSNs.

Keywords: Linear network coding; wireless sensor networks; convergecast.

A preliminary version of this paper appeared in IEEE PIMRC 2013, Sept. 8-11, London, UK. This version is reorganized and includes new theoretical evaluations on new performance metrics, more specific analyses and more detailed proofs. This work is supported by the National Natural Science Foundation of China (No. 61303210, No. 61201426 and No. 61303211) and Zhejiang Provincial Natural Science Foundation of China (No. LQ12F02009).

http://dx.doi.org/10.3837/tiis.2014.09.006

1. Introduction

Linear network coding (LNC) [1, 2], proposed on the basis of the theory of network coding (NC) [3], has been proven that it is able to improve a network's throughput, efficiency and reliability. LNC suggests the fundamental idea of linearly combining several packets in an intermediate node for transmission, instead of only replicating and forwarding them as those traditional routing protocols behave. Subsequent studies show that LNC is particularly well-suited for wireless networks due to the broadcast nature of their communications, and has been intensively studied in various networked scenarios such as multicast, broadcast and multi-flow unicast [4-10]. And in recent years, NC/LNC is also applied to cooperative communications to achieve more diversity order [11, 12].

The attractive advantages of wireless LNC also boost LNC's practical application in wireless sensor networks (WSNs). A WSN typically consists of devices which are capable of sensing environmental or physical quantities and communicate with each other over wireless links. They are generally powered by batteries which are difficult and costly to be recharged or replaced. These devices are scattered within the desired area, generate data by sensing the specified objects, and transport them to a common sink, as shown in **Fig. 1**. Hence, the most common communication style in WSNs should be convergecast, i.e., multiple-to-one transmissions.

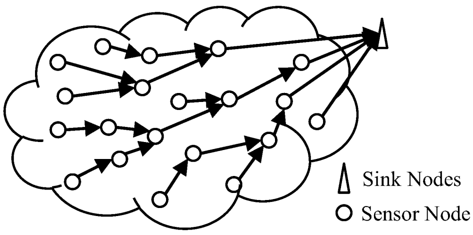


Fig. 1. Convergecast in wireless sensor networks: Sensor devices generate data by sensing the specified objects, and transport them to a common sink.

Although plenty of far-reaching theories and practical solutions on applying LNC to WSNs have been proposed in recent years, which have demonstrated that LNC is beneficial for WSNs in multicast/broadcast or unicast scenarios, it is harder to apply LNC to convergecast. Differing from multicast or unicast, the typical traffic patterns where the LNC benefits are generated, such as the X-topology and butterfly topology, do not typically occur during convergecasting. And most of the existing solutions for LNC-aided convergecast have mainly focused on exploiting some kind of benefits without considering its potential adverse effect. Hong have proposed the Cascading Data Collection (CDC) mechanism to reliably gather data from sensor nodes [13]. CDC can achieve high energy efficiency, while its performance on end-to-end delay has not been presented. Since the original packets cannot be recovered until the sink has received enough number of coded packets, it is believed that CDC most probably suffers long end-to-end delay. Samarasinghe has also proposed a LNC-aided convergecast solution in [14], and have practically applied it to a realistic WSN. And they have showed two key limitations of LNC, i.e., strongly increased delay and high overhead due to limited lack of adaptability [15]. Keller has presented SenseCode, a LNC-aided collection protocol for WSNs [16]. Compared to the best existing alternative, SenseCode improves reliability, however, at the cost of consuming more energy.

So, a natural question is: Is LNC a friend or foe to convergecast in WSNs? The answer is significant for applying LNC to practical WSN applications. However, as far as we know, few works have attempted to give an answer on this issue. Out of this consideration, in this paper, we mainly focus on the role LNC plays in convergecast of WSNs and investigate how LNC affects the performance of convergecast. The issue is discussed within four scenarios, i.e. the LNC-aided and the none-LNC convergecast schemes with and without automatic repeat request (ARQ). Each scheme only involves the most basic topology for performing a complete process of LNC in order to eliminate the influence of other factors as far as possible. And the most essential difference among these scenarios is whether the success of packet delivery is guaranteed or not. For the scenarios without ARQ, data collection rate, energy consumption and energy consumption balance are the most concerned performance metrics. While for those with ARQ, energy efficiency, energy consumption balance and end-to-end delay are the most common performance requirements. Accordingly, two LNC-aided convergecast schemes are proposed in this paper. Theoretical analyses on concerned performance metrics are carried out and simulation are conducted to validate the theoretical analyses.

The rest of this paper is organized as follows. Section II presents the system models. In Section III, we consider the scenario of LNC-aided convergecast without ARQ and offer analyses on packet collection rate and energy consumption. In Section IV, we investigate the LNC-aided convergecast with ARQ, followed by Section V giving the conclusions.

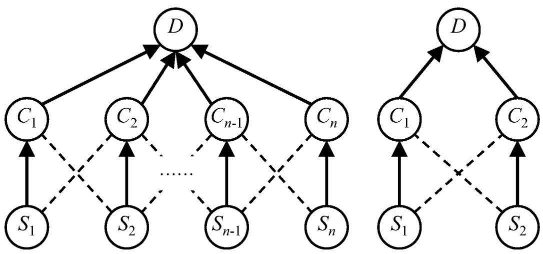
2. System Models and Simulation scenarios

2.1 Network models

In this paper, We assume that a converge tree ($\mathbb{T}$) has been created by existing traditional data collection protocols or routing protocols, such as CTP [17]. $\mathbb{T}$ is a directed tree with the sink as its root. **Fig.** 1 shows an example of $\mathbb{T}$. Then, we give the definition of the converge-structure, which have been introduced in [18].

Definition 1: An *n*-order converge-structure ($\mathbb{CS}_n$) is a three-layer subtree of the $\mathbb{T}$ composed of *n* leaves $S_i (i = 1, 2, \ldots, n)$, *n* interior nodes $C_i (i = 1, 2, \ldots, n)$, and a root *D*, as shown in **Fig. 2**. A $\mathbb{CS}_n$ possesses the following characteristics:

- C_i only has one child S_i;
- C_i can overhear the communications of S_{i-1} and S_{i+1}. While C_1 and C_n can only overhear the communications of S_2 and S_{n-1} respectively.



(a) $\mathbb{CS}_n$ (b) $\mathbb{CS}_2$

Fig. 2. Converge-structures. The nodes and solid lines with arrows form a subtree of converge-tree. The dashed lines represent the overhearing links.

A $\mathbb{CS}_n$ is the most basic structure to perform a complete process of LNC. Original packets are generated by S_i. C_i receives and overhears those original packets and mixes them into a coded packet, which is sent to the decoder D. It should be noted that these $\mathbb{CS}_n$ are abstracted from $\mathbb{T}$, and the probability of forming $\mathbb{CS}_n$ is considerably high during convergecasting, especially for low order $\mathbb{CS}_n$ in a dense network.

In this paper, we simply investigate LNC-aided convergecast in a $\mathbb{CS}_n$, but not an entire $\mathbb{T}$. The reason is that the performance of LNC-aided convergecast in a complete $\mathbb{T}$ is influenced by some other factors than LNC itself. For example, the method of how to recognize $\mathbb{CS}_n$ from $\mathbb{T}$ imposes a significant effect on the overall performance.

2.2 Linear network coding operation

In our previous work, we have investigated the feasibility of acquiring LNC benefits in convergecast, and argued that the reliability benefits can be obtained by applying LNC in the ubiquitous $\mathbb{CS}_n$ [18]. In a $\mathbb{CS}_n$, C_i may receive at most three original packets from $S_{i\text{-}1}$, S_i and S_{i+1} and mix them as follows:

$$X_k = \sum_{i=1}^{3} g_k^i M_i \quad (1)$$

where $M_i (i = 1, 2, 3)$ are the original packets, g_k^i are the coefficients randomly selected from a q-order Galois field $\mathbb{GF}(q)$, and X_k is the coded packet. Since all the operations are computed in $\mathbb{GF}(q)$, X_k is of the same length as the original packets. Totally n coded packets are generated by n coders and are forwarded to D, where they are decoded by means of Gaussian Elimination only if D has received n coded packets [19].

2.3 Simulation scenarios

In this paper, plenty of simulations are conducted to verify our theoretical analyses. All the simulations run on the Network Simulator II (NS2). The topologies are shown in **Fig.** 3. Since our primary objective is to evaluate LNC benefits, we simply assign the converge-tree manually as shown in **Fig.** 2. Each source node generates 100 CBR packets destined to D. And the packet size is defined as 1000 bits. In the simulations, no mobility is assumed.

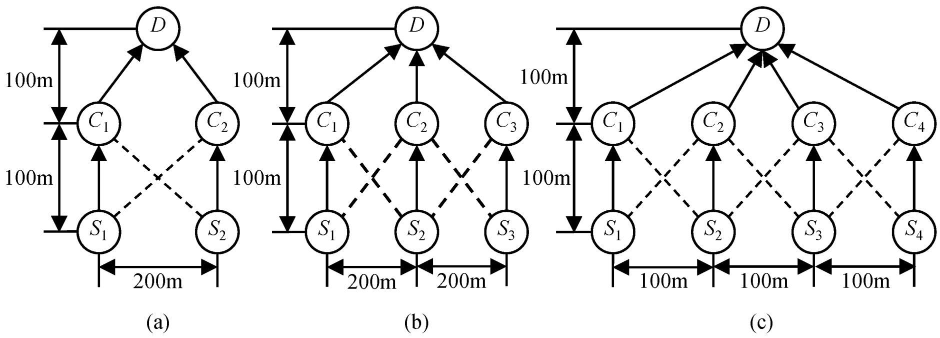


Fig. 3. The network topologies for simulations, (a) for $\mathbb{CS}_2$, (b) for $\mathbb{CS}_3$, (c) for $\mathbb{CS}_4$.

Four convergecast schemes have been implemented in this paper, i.e. the LNC-aided and the none-LNC convergecast schemes with and without ARQ. All the schemes are based on a simple Time Division Multiple Access (TDMA) Medium Access Control (MAC) protocol.

For simplicity, TDMA schedules with global time synchronization are employed. In detail, the first *n* slots of a TDMA frame are allocated to the source nodes, the next *n* slots are allocated to the coders, and the last slot is assigned for the decoder.

In our simulations, any node has four operational modes: transmitting, receiving, idle (idle listening), and sleep, consuming P_t=24.75 mW, P_r=13.5 mW, P_i=13.5 mW, and P_s=15μW, respectively [20]. It should be noted that in idle mode, a node keeps listening to receive the potential traffic that is not sent. And measurements have shown that the energy consumption of idle listening is comparable to that of receiving [21, 22]. The data rate is 250 kbps.

2.4 ARQ for LNC-aided convergecast

For the schemes without ARQ, senders do nothing more after transmissions. And for the none-LNC schemes with ARQ, the most common link-level stop-and-wait ARQ is adopted. While for those LNC-aided ones with ARQ, two different ARQ mechanisms are adopted. On the one hand, for communications between S_i and C_i (C_i is the unique parent of S_i in $\mathbb{T}$), S_i will not start to send another packet unless the current one has been successfully acknowledged. However, unlike the traditional ARQ that only C_i is allowed to give the acknowledgement to S_i, in the LNC-aided convergecast scheme, C_{i-1} and C_{i+1} are also allowed. Specifically, after having received an original packet from S_i, C_{i-1}, C_i and C_{i+1} should acknowledge it one by one. And S_i discards the duplicated ACK frames once it has already got one. On the other hand, the process of Gaussian Elimination starts after *D* has received a coded packet successfully. If all or some of the original packets are successfully recovered, *D* replies with a particular acknowledgement (DACK, Decoder ACK) which carries the IDs of the correctly recovered packets. Once a coder receives a DACK, it removes all the original packets which have been recovered in *D* and are indicated by DACK from the buffer. If a coder fails to get a DACK in time, it re-codes the original packets to a new coded packet and sends it to the decoder again in its next turn. For simplicity, we assume that all the ACK frames can always be received successfully. It is reasonable because an ACK frame is so short that the probability of failure is low enough.

3. Convergecast without ARQ

In these scenarios, no ACK mechanism is adopted. Original and coded packets may be lost due to link failures, which causes that the decoder cannot collect enough coded packet to recover all the original ones. The performances on packet collection rate, energy consumption and energy consumption balance are discussed, since they are the most concerned metrics. The ratios of these concerned performances between the LNC-aided schemes and None-NC schemes, i.e.,

$$\frac{\text{The performance of LNC} - \text{aided schemes}}{\text{The performance of None} - \text{LNC schemes}}$$

are presented in order to emphasize the comparison results.

As mentioned above, plenty of simulations have been conducted to verify the theoretical analyses. Unless otherwise mentioned, theoretical results are adopted in the following discussions, in consideration that the errors between the theoretical calculation and the simulation results are small enough (See Appendix I).

3.1 Collection rate

We have investigated the probability that *D* can recover all the original packets in [18]. However, there are cases that although *D* cannot recover all the original packets, but it can recover some of them. The collection rate (R^n_{col}) of a $\mathbb{CS}_n$ in the LNC-aided scheme is the function of the wireless link delivery ratio (*r*). The link delivery ratio is the probability that a data packet successfully received by the one-hop destination, and is influenced by many factors such as wireless channel fading, noise and interferences from others [23, 24].

According to total probability formula, $R^n_{col}(r)$ can be calculated as

$$R^n_{col}(r) = \frac{1}{n} \cdot \sum_{i=1}^{n} \left[i \sum_{N_{\mathcal{L}}^{C\to D}=i}^{n} P_n^i \left(N_{\mathcal{L}}^{C\to D}, r \right) \right] \tag{2}$$

where $P_n^i(N_{\mathcal{L}}^{C\to D}, r)$ is the probability that the decoder recovers $i(i \leqslant n)$ packets successfully, and $N_{\mathcal{L}}^{C\to D}$ is the number of working links among the coders and the decoder. $P_n^i(N_{\mathcal{L}}^{C\to D}, r)$ can be obtained by method of exhaustion.

And meanwhile, the collection rate of the none-LNC scheme in a $\mathbb{CS}_n$ ($\hat{R^n_{col}}$) equals to the probability of the success of the packet delivery from S_i to *D*. And the sufficient and necessary conditions for the success of the packet delivery from S_i to *D* are: (a) The packet delivery from S_i to C_i is successful; and (b) The packet delivery from C_i to *D* is successful. Obviously, the probabilities of condition (a) and (b) are both equal to *r*. Hence, we have

$$\hat{R}^n_{col}(r) = r^2 \tag{3}$$

Fig. 4 shows the ratios of the collection rates of the LNC-aided and the none-LNC schemes in a $\mathbb{CS}_n(n = 2, 3, 4)$ without ARQ, where it can be observed that when *r* is low, the LNC-aided scheme considerably outperforms the traditional one due to the redundant links introduced by overhearing. And the gain is proportional to the order of $\mathbb{CS}_n$, i.e. *n*, and inversely proportional to *r*. However, things changes when *r* gets higher. The traditional none-LNC convergecast scheme collects more packets than the LNC-aided one. The reason is that, although the redundant links between the sources and the coders help collect more original packets (this benefit is inversely proportional to *r*, as we have discussed in [18]), however, the losses of coded packets result in the recovery failure of *n* original ones directly, i.e., packet loss penalty. And this side effect of LNC trends to play a major role in total packet loss when the link quality is high.

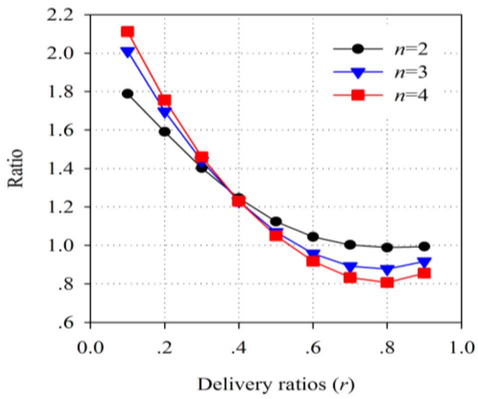


Fig. 4. The ratios of the collection rates between the LNC-aided to the none-LNC convergecast schemes in a $\mathbb{CS}_n(n = 2, 3, 4)$ without ARQ.

3.2 Energy consumption

The following variables are defined:

- Let $\mathcal{E}_{\mathcal{M}}^{\mathcal{T}}(\mathcal{M} \in \{t,r\}, \mathcal{T} \in \{d,a\})$ be the energy consumption per transmission ($\mathcal{M}=t$) or reception ($\mathcal{M}=r$, including overhearing) of a data packet ($\mathcal{T}=d$) or ACK ($\mathcal{T}=a$) in LNC-aided schemes.
- Let $E_{\mathcal{M}}^{\mathcal{R}}(\mathcal{M} \in \{t,r\}, \mathcal{R} \in \{S,C,D\})$ be the total energy consumption of transmitting ($\mathcal{M}=t$) or receiving ($\mathcal{M}=r$) of S_i ($\mathcal{R}=S$), C_i ($\mathcal{R}=C$) or D ($\mathcal{R}=D$) in LNC-aided schemes.
- Let $\mathcal{N}_{\mathcal{M}}^{(\mathcal{T},\mathcal{R})}(\mathcal{M} \in \{t,r\}, \mathcal{T} \in \{d,a\}, \mathcal{R} \in \{S,C,D\})$ be the average number of transmissions ($\mathcal{M}=t$) or receptions ($\mathcal{M}=r$) per data packet ($\mathcal{T}=d$) or ACK ($\mathcal{T}=a$) of S_i ($\mathcal{R}=S$), C_i ($\mathcal{R}=C$) or D ($\mathcal{R}=D$) in LNC-aided schemes.
- Let $\hat{V}$ be the corresponding V in none-LNC schemes. For example, $\hat{E}_t^S$ denotes the total energy consumption of transmitting of S_i in none-LNC schemes.

First, we consider the energy consumption of LNC-aided scheme without ARQ. Assuming that each source node transmits one packet, the total energy consumption (E) is

$$\begin{aligned} E &= nE_t^S + n\left(E_t^C + E_r^C\right) + E_r^D + E_i + E_s \approx nE_t^S + n\left(E_t^C + E_r^C\right) + E_r^D \\ &= n\mathcal{E}_t^d + \left(\mathcal{E}_t^d + \mathcal{E}_r^d\right)\left[2\overline{\mathcal{A}}(2) + (n-2)\overline{\mathcal{A}}(3)\right] + \mathcal{E}_r^d(3n-2) \end{aligned} \tag{4}$$

where $\overline{\mathcal{A}}(x) = 1 - (1-r)^x$. The detailed proof of (4) is presented in Appendix II. The energy consumption of idle listening (E_i) and sleep (E_s) are ignored since they are far less than energy consumed by transmitting and receiving ($E_i = P_i T_i$, where T_i is the amount of time that a node stays in idle mode. In TDMA based MAC protocol, T_i can be extremely short by duly switching nodes to sleep mode when data transmitting and receiving are not required).

And for the none-LNC scheme, the total energy consumption ($\hat{E}$) should be

$$\hat{E} \approx n\hat{E}_t^S + n\left(\hat{E}_t^C + \hat{E}_r^C\right) + \hat{E}_r^D = n(1+r)\left(\hat{\mathcal{E}}_t^d + \hat{\mathcal{E}}_r^d\right) \tag{5}$$

The proof of (5) is presented in Appendix III.

The theoretical and simulation results on the ratios of energy consumption per successfully received packet between the LNC-aided convergecast schemes and none-LNC ones in a $\mathbb{CS}_n(n=2,3,4)$ without ARQ are shown in **Fig. 5**. It can be observed that the LNC-aided schemes are more energy-saving than the none-LNC ones only when r is low enough. While in most cases, the none-LNC schemes consume less energy. And the contrast is even more obvious in a high order $\mathbb{CS}_n$.

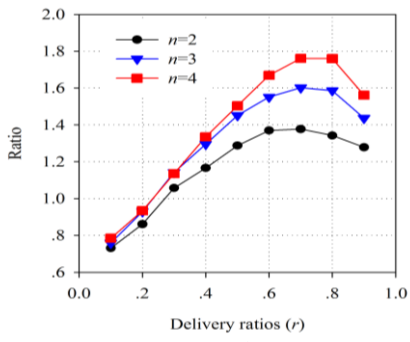


Fig. 5. The ratios of the energy consumption per successfully received packet between the LNC-aided and the none-LNC convergecast schemes in a $\mathbb{CS}_n(n=2,3,4)$ without ARQ.

3.3 Energy consumption balance

We further investigate the performance on the Energy Consumption Balance (ECB) among sources, coders and decoders.

Definition 2: There are k nodes in the network, denoted as $N_i (i = 1, 2, \dots, k)$. $\mathcal{E}_i (i = 1, 2, \dots, k)$ is the energy consumption of N_i. The Energy Consumption Balance Indicator (ECBI) of N_i ($\mathcal{B}_i$) is defined as

$$\mathcal{B}_i \triangleq \frac{\mathcal{E}_i}{\sum_{j=1}^{k} \mathcal{E}_j} - \frac{1}{k} \tag{6}$$

And the ECB of the network can be calculated as

$$ECB = \frac{1}{k} \sum_{i=1}^{k} |\mathcal{B}_i| \tag{7}$$

According to (4) ~ (7), the ECB of the LNC-aided convergecast scheme in a $\mathbb{CS}_n$ without ARQ is

$$ECB_n = \frac{n\left|\frac{E_t^S}{E} - \frac{1}{2n+1}\right| + n\left|\frac{E_t^C + E_r^C}{E} - \frac{1}{2n+1}\right| + \left|\frac{E_r^D}{E} - \frac{1}{2n+1}\right|}{2n+1} \tag{8}$$

Fig. 6 plots the ECB performance of a $\mathbb{CS}_n (n = 2, 3, 4)$ in the LNC-aided and the none-LNC convergecast schemes without ARQ. **Fig. 6(a)** indicates that, in most cases, the energy consumption of none-LNC convergecast schemes is more balanced than that of LNC-aided schemes. Moreover, high-order converge-structures are more balanced than low-order ones, as shown in **Fig. 6(b)**. We can also see that the dashed curves in **Fig. 6(b)**, which illustrate the ECBs of the NC-aided schemes obtained by (8), are concave. And there exists an obvious minimum value for each of them. While the solid curves showing the ECBs of None-NC schemes in **Fig. 6 (b)** are much flatter. Hence, the curves in **Fig. 6(a)** are bound to have peak values since they represent the ratios between the solid curves in **Fig. 6(b)** and the corresponding dashed curves (with the same color) in **Fig. 6(b)**. The positions of these peak values are consistent with the positions of the minimum values of the dashed curves in **Fig. 6(b)**.

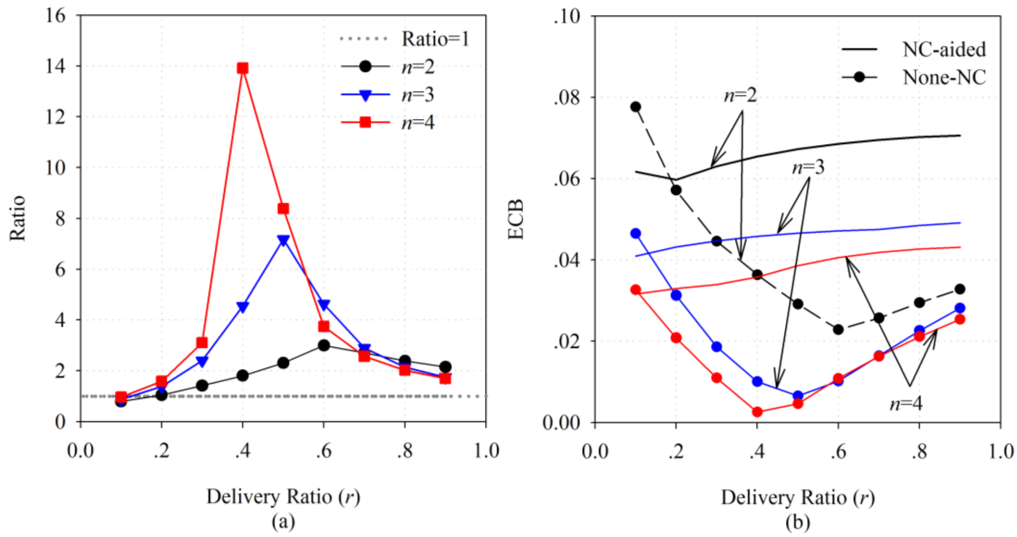


Fig. 6. The energy consumption balance of a $\mathbb{CS}_n (n = 2, 3, 4)$ in the LNC-aided and the none-LNC convergecast schemes without ARQ. (a) Ratio of ECB between the LNC-aided and the none-LNC convergecast schemes; (b) the ECBs.

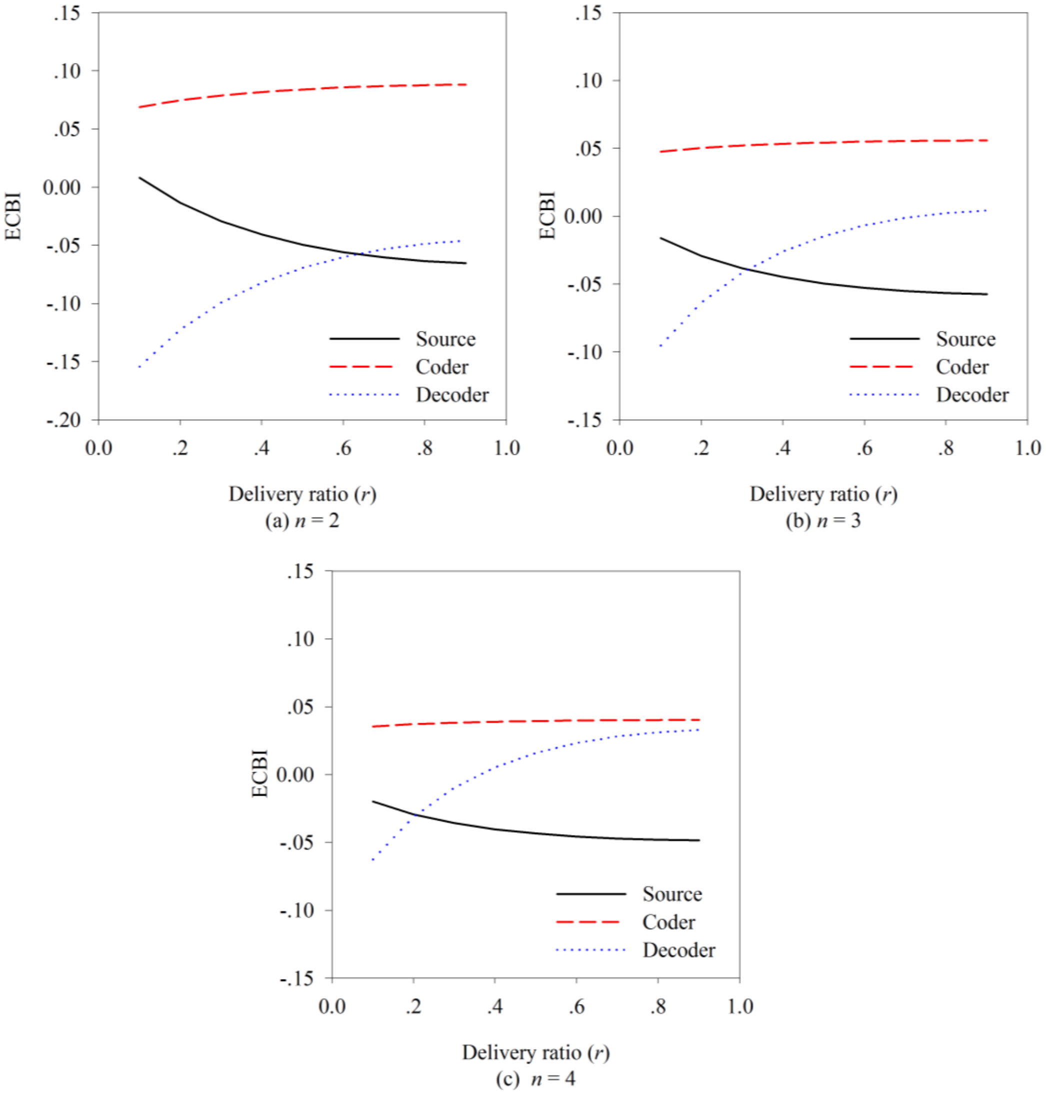


Fig. 7. The ECBIs of different type of node in a $\mathbb{C}\mathbb{S}_n(n = 2, 3, 4)$ in the LNC-aided convergecast schemes without ARQ.

The theoretical analyses also show that in LNC-aided schemes, coders' ECBIs are always positive, while sources' are negative only except the case of $n = 2$ and $r = 0.1$, as shown in **Fig. 7**. This provides a feasible way to improve the ECB of LNC-aided schemes. As we have argued in [18], converge-structures may be overlapped to obtain more LNC benefits, as illustrated in **Fig. 8**. Take the gray node as an example. It serves as a coder in the $\mathbb{C}\mathbb{S}_2$ circled by the ellipse II, and meanwhile, it is also a source in the $\mathbb{C}\mathbb{S}_2$ circled by the ellipse I. And the overall ECBI is very likely to be smaller than both of the ECBIs corresponds to the role of coder and source.

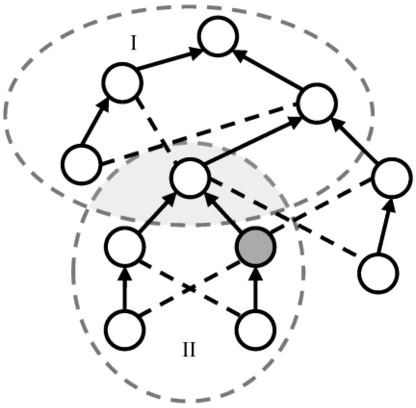


Fig. 8. Overlapped converge-structures may help to reduce ECB in LNC-aided schemes.

4. Convergecast with ARQ

In many practical applications of WSNs, ARQ should be employed to ensure the packet delivery. In these scenarios, all the original packets can be collected by *D* with the help of the ARQ mechanisms mentioned above. Therefore, instead of the collection rate, we care much more about the performance metrics such as energy consumption, energy consumption balance and end-to-end delay.

4.1 Energy consumption

In the LNC-aided convergecast schemes with ARQ, assuming that each source node transmits one packet, the total energy consumption (*E*) is

$$\begin{aligned} E \approx & n\left(E_t^S + E_r^S\right) + n\left(E_t^C + E_r^C\right) + \left(E_t^D + E_r^D\right) \\ = & \mathcal{E}_t^d \left[\frac{2}{\overline{\mathcal{A}}(2)} + \frac{n-2}{\overline{\mathcal{A}}(3)}\right] + \mathcal{E}_r^a \left[\frac{2\overline{\mathcal{N}}(2)}{\overline{\mathcal{A}}(2)} + \frac{(n-2)\overline{\mathcal{N}}(3)}{\overline{\mathcal{A}}(3)}\right] + n\left(\mathcal{E}_t^a + \frac{\mathcal{E}_r^d}{r}\right) \\ & + \mathcal{E}_r^d \left(3 - \frac{2}{n}\right)\left[\frac{2}{\overline{\mathcal{A}}(2)} + \frac{n-2}{\overline{\mathcal{A}}(3)}\right] + \mathcal{E}_t^a \left[\frac{2\overline{\mathcal{N}}(2)}{\overline{\mathcal{A}}(2)} + \frac{(n-2)\overline{\mathcal{N}}(3)}{\overline{\mathcal{A}}(3)}\right] + n\left(\mathcal{E}_r^a + \frac{\mathcal{E}_t^d}{r}\right) \end{aligned} \tag{9}$$

where

$$\overline{\mathcal{N}}(x) = \sum_{k=1}^{x} k \binom{x}{k} r^k (1-r)^{x-k}$$

The detailed proof of (9) is presented in Appendix IV.

While in the none-LNC scheme,

$$\begin{aligned} \hat{E} \approx & n\left(\hat{E}_t^S + \hat{E}_r^S\right) + n\left(\hat{E}_t^C + \hat{E}_r^C\right) + \left(\hat{E}_t^D + \hat{E}_r^D\right) \\ = & n\left(\frac{\hat{\mathcal{E}}_t^d}{r} + \hat{\mathcal{E}}_r^a\right) + n\left(\frac{\hat{\mathcal{E}}_t^d}{r} + \hat{\mathcal{E}}_t^a + \frac{\hat{\mathcal{E}}_r^d}{r} + \hat{\mathcal{E}}_r^a\right) + n\hat{\mathcal{E}}_t^a + \frac{n}{r}\hat{\mathcal{E}}_r^d \end{aligned} \tag{10}$$

The detailed proof of (10) is presented in Appendix V.

The ratios of the energy consumption between the LNC-aided and the none-LNC convergecast schemes in a $\mathbb{CS}_n (n = 2, 3, 4)$ with ARQ are shown in **Fig. 9**. It can be observed that when *r* is low, $E < \hat{E}$. However, when *r* gets higher, $E > \hat{E}$. In fact, LNC has a two-fold impact on the energy consumption of convergecast. On one hand, LNC reduces E_t^S with the help of redundant links, but on the other, LNC increases E_r^C greatly due to overhearing.

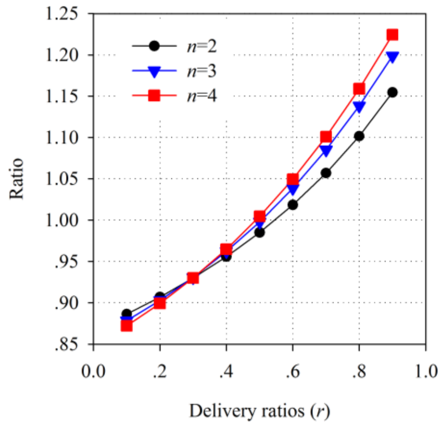


Fig. 9. The energy consumption of the LNC-aided and the none-LNC convergecast schemes in a $\mathbb{CS}_n(n = 2, 3, 4)$ with ARQ.

4.2 End-to-end delay

The average end-to-end delay of the LNC-aided convergecast scheme ($\mathcal{D}_n$) in a $\mathbb{CS}_n$ is

$$\mathcal{D}_n = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} d_n^i(j) \tag{11}$$

where $d_n^i(j)$ is the average end-to-end delay of the packet that is sent by S_i and decoded by *D* when *D* receives the coded packet from C_j in a $\mathbb{CS}_n$. And $d_n^i(j)$ can be calculated as

$$d_n^i(j) = \sum_{k=0}^{\min(n-1,2)} p_n^{i,j}(k)\left[\mathcal{N}_t^{(d,S)}\cdot\mathcal{N}_n + \left(\mathcal{N}_t^{(d,C)} - 1 + k\right)\mathcal{N}_n + (n+j-i)\right] \tag{12}$$

where $p_n^{i,j}(k)$ is the probability that the packet sent by S_i is correctly decoded by *D* when *D* receives the coded packet from C_j at the $(n+j)^{\text{th}}$ slot in the k^{th} subsequent TDMA frame after the first transmission of the packet in a $\mathbb{CS}_n$, and $\mathcal{N}_n = 2n+1$ is the number of nodes of the $\mathbb{CS}_n$. $p_n^{i,j}(k)$ can be obtained by exhaust algorithm.

And that of the none-aided convergecast ($\hat{\mathcal{D}}_n$) in a $\mathbb{CS}_n$ is

$$\hat{\mathcal{D}}_n = \hat{\mathcal{N}}_t^{(d,S)}\cdot\mathcal{N}_n + \left(\hat{\mathcal{N}}_t^{(d,C)} - 1\right)\mathcal{N}_n + n \tag{13}$$

Fig. 10 illustrates the theoretical and simulation results on the ratios of the end-to-end delay (in slots) between the LNC-aided and the none-LNC schemes in a $\mathbb{CS}_n(n = 2, 3, 4)$. Clearly, in the case of poor wireless link quality, LNC is able to reduce the delay by means of its reliability benefits. In these cases, retransmission is the chief reason of the delay. However, when the link quality gets better, none-LNC schemes outperform LNC-aided ones. The reason is that, with the delivery ratio improves, the delay caused by retransmissions decreases rapidly and tends to play a minor role of the total delay. And meanwhile the inevitable waiting period, which is introduced by the decoder for collecting enough coded packets to recover the original ones, significantly increases the end-to-end delay.

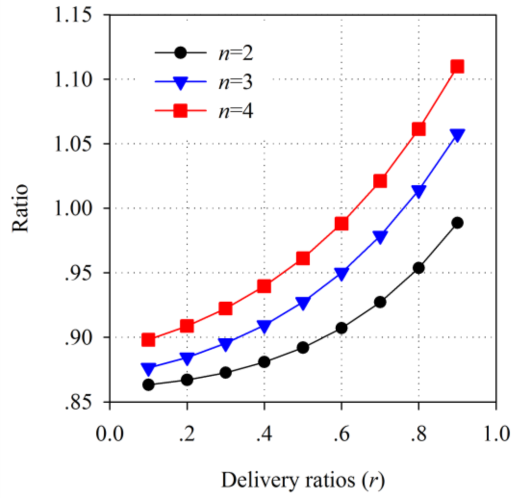


Fig. 10. The ratios of end-to-end delay (in slots) between the LNC-aided and the none-LNC convergecast schemes in a $\mathbb{CS}_n(n = 2, 3, 4)$ with ARQ.

4.3 Energy consumption balance

We also study the ECB performance of the LNC-aided and the none-LNC convergecast schemes with ARQ according to (8). As shown in **Fig. 11**, the results are similar to those presented in Section 3.3. The sources and coders' ECBIs in the LNC-aided schemes are far greater than those in none-LNC schemes, which results in relative high ECBs. For a LNC-aided convergecast scheme in a converge-structure with a given order, ECB reduces along with the improvement of wireless link quality. While given the wireless delivery ratio, ECB is inversely proportion to the order of the converge-structure. It also should be noted that in LNC-aided schemes, coders' ECBIs are positive and sources' are negative, as shown in **Fig. 12**, which means that overlapping converge-structures is also helpful to ECB.

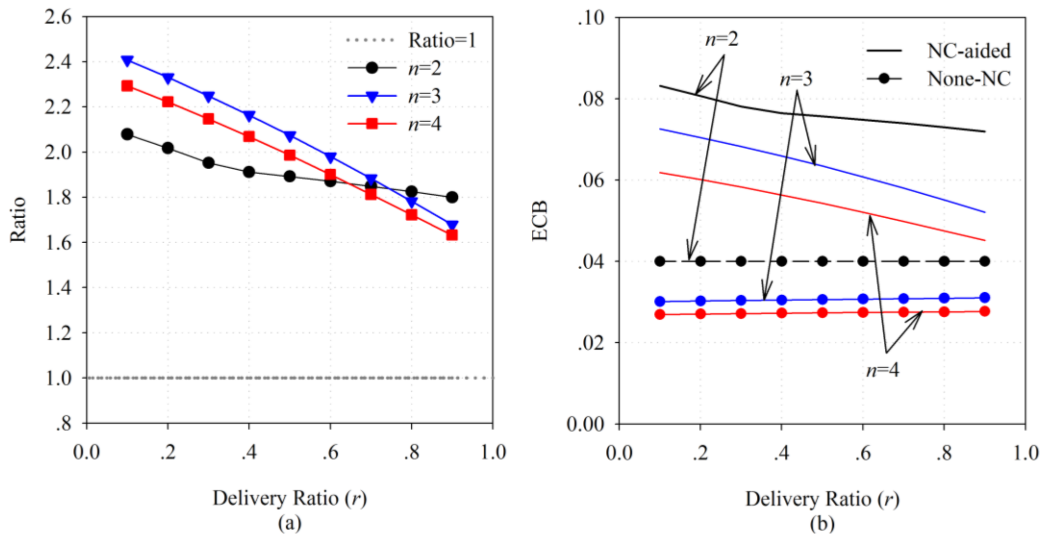


Fig. 11. ECB of a $\mathbb{CS}_n(n = 2, 3, 4)$ in the LNC-aided and the none-LNC convergecast schemes with ARQ.

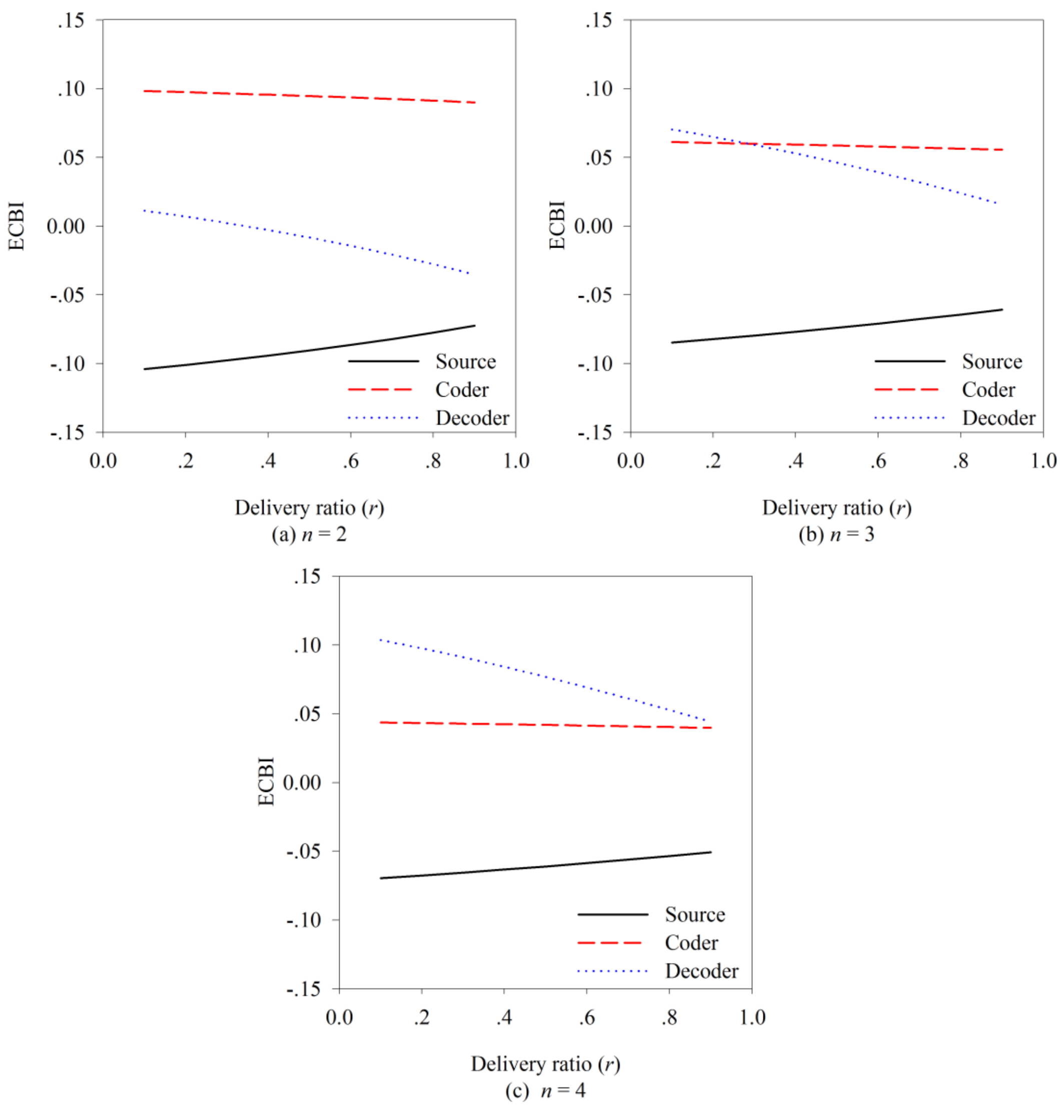


Fig. 12. The ECBIs of different type of node in a $\mathbb{CS}_n(n = 2, 3, 4)$ in the LNC-aided convergecast schemes with ARQ.

5. Conclusion

In this paper, we discuss about whether LNC can always benefit convergecast in a WSN. We implement four various convergecast schemes, i.e., the LNC-aided and the none-LNC convergecast schemes with or without ARQ. By theoretical analyses and simulations, we reach the following conclusions:

- The LNC-aided convergecast schemes surpass the none-LNC ones in the case of low link delivery ratio. However, due to the considerable overhearing required by coders, the packet loss penalties and the prerequisite of successfully decoding, the LNC-aided convergecast schemes fall behind none-LNC ones almost in all respects if the link delivery ratio is high enough. In other words, LNC is particularly fit for WSNs with awful link quality.

- $\mathbb{CS}_2$ outperforms higher order $\mathbb{CS}_n$ in most performance metrics. Although $\mathbb{CS}_2$ introduces the less redundant links, it suffers less side effects of LNC.
- LNC-aided schemes are more unbalanced in energy consumption. However, they are expected to be improved by means of overlapping converge-structures

Appendix I: Theoretical calculation vs. simulation results

The errors between the theoretical calculation and the simulation results are presented in the following tables.

Table A. 1. Errors between the theoretical calculations and simulation results: Collection rate

N	2	3	4	2	3	4
Error (%)	NC-aided			None-NC		
	1.6902307	0.979037	1.3792149	3.226158	1.4760215	1.5316324

Table A. 2. Errors between the theoretical calculations and simulation results: Energy Consumption

(a). Without ARQ

N	2	3	4	2	3	4
Error (%)	NC-aided			None-NC		
	1.393282	1.203558	1.138056	1.171104	2.3569	2.3401

(b). With ARQ

N	2	3	4	2	3	4
Error (%)	NC-aided			None-NC		
	0.311099	0.243567	0.4212891	1.265123	1.025033	1.337107

Table A. 3. Errors between the theoretical calculations and simulation results: Delay

n	2	3	4	2	3	4
Error (%)	NC-aided			None-NC		
	2.570246	2.69713	4.781804	0.285133	0.254775	0.300837

Appendix II: Proof of (4)

Proof: Assuming that each source node transmits one packet, the total energy consumption of the LNC-aided convergecast scheme without ARQ should be

$$E \approx nE_t^S + n\left(E_t^C + E_r^C\right) + E_r^D$$

where E_t^S, E_t^C, E_r^C and E_r^D can be calculated as follows.

- E_t^S: The only thing a source node needs to do is to transmit one data packet in scenarios without ARQ. And no retransmission is required. Therefore, $\mathcal{N}_t^{(d,S)} = 1$ and $E_t^S = \mathcal{E}_t^d \mathcal{N}_t^{(d,S)} = \mathcal{E}_t^d$.
- E_t^C: A transmission of C_i occurs if it has correctly received at least one original packet from S_{i-1}, S_i and S_{i+1} (exceptionally, S_1 and S_2 for C_1, S_{n-1} and S_n for C_n) before its slot comes. The probability that C_1 or C_n can correctly receive at least one original packet is $1-(1-r)^2$, and that for $C_i (i \neq 1, n)$ is $1-(1-r)^3$. Hence, we have

$$\mathcal{N}_t^{(d,C)} = \frac{2\left[1-(1-r)^2\right]+(n-2)\left[1-(1-r)^3\right]}{n} = \frac{2\overline{\mathcal{A}}(2)+(n-2)\overline{\mathcal{A}}(3)}{n}$$

and $E_t^C = \mathcal{E}_t^d \mathcal{N}_t^{(d,C)}$.

- E_r^C: C_i performs receiving or overhearing when $S_{i\text{-}1}$, S_i or S_{i+1} is transmitting (exceptionally, S_1 and S_2 for C_1, $S_{n\text{-}1}$ and S_n for C_n), no matter whether the transmission is successful or not. Therefore, $\mathcal{N}_r^{(d,C)} = \frac{2\times 2+3(n-2)}{n} = 3-\frac{2}{n}$ and $E_r^C = \mathcal{E}_r^d \mathcal{N}_r^{(d,C)}$.
- E_r^D: A receiving operation of D occurs only when a coder transmits a coded packet. Therefore, $\mathcal{N}_r^{(d,D)} = n\mathcal{N}_t^{(d,C)}$ and $E_r^D = \mathcal{E}_r^d \mathcal{N}_r^{(d,D)}$.

It is the end of proof.

Appendix III: Proof of (5)

Proof: Assuming that each source node transmits one packet, the total energy consumption of the none-LNC convergecast scheme without ARQ should be

$$\hat{E} \approx n\hat{E}_t^S + n(\hat{E}_t^C + \hat{E}_r^C) + \hat{E}_r^D$$

where E_t^S, E_t^C, E_r^C and E_r^D can be calculated as follows.

- $\hat{E}_t^S$: The only thing a source node needs to do is to transmit one data packet in scenarios without ARQ. And no retransmission is required. Therefore, $\hat{\mathcal{N}}_t^{(d,S)} = 1$ and $\hat{E}_t^S = \hat{\mathcal{E}}_t^d \hat{\mathcal{N}}_t^{(d,S)} = \hat{\mathcal{E}}_t^d$.
- $\hat{E}_t^C$: A transmission of C_i occurs if it has correctly received its child's original packet. The probability that C_i can correctly receive the packet is r. Hence, we have $\hat{\mathcal{N}}_t^{(d,C)} = r$ and $\hat{E}_t^C = \hat{\mathcal{E}}_t^d \hat{\mathcal{N}}_t^{(d,C)} = r\hat{\mathcal{E}}_t^d$.
- $\hat{E}_r^C$: C_i performs receiving when S_i is transmitting, no matter whether the transmission is successful or not. Therefore, $\hat{\mathcal{N}}_r^{(d,C)} = \hat{\mathcal{N}}_t^{(d,S)} = 1$ and $\hat{E}_r^C = \hat{\mathcal{E}}_r^d \hat{\mathcal{N}}_r^{(d,C)} = \hat{\mathcal{E}}_r^d$.
- $\hat{E}_r^D$: A receiving operation of D occurs only when C_i transmits a coded packet. Therefore, $\hat{\mathcal{N}}_r^{(d,D)} = n\hat{\mathcal{N}}_t^{(d,C)} = nr$ and $\hat{E}_r^D = \hat{\mathcal{E}}_r^d \hat{\mathcal{N}}_r^{(d,D)} = nr\hat{\mathcal{E}}_r^d$.

It is the end of proof.

Appendix IV: Proof of (9)

Proof: Assuming that each source node transmits one packet, the total energy consumption of the LNC-aided convergecast scheme with ARQ should be

$$E \approx n\left(E_t^S + E_r^S\right) + n\left(E_t^C + E_r^C\right) + \left(E_t^D + E_r^D\right)$$

where E_t^S, E_r^S, E_t^C, E_r^C, E_t^D and E_r^D can be calculated as follows

- E_t^S: According to the ARQ scheme introduced in Section II.D, the average transmission number required by S_1 or S_n to successfully delivery a data packet is $\frac{1}{1-(1-r)^2}$, and that of $S_i(i \neq 1, n)$ is $\frac{1}{1-(1-r)^3}$. So, we have $E_t^S = \mathcal{E}_t^d \mathcal{N}_t^{(d,S)}$ where

$$\mathcal{N}_t^{(d,S)} = \frac{1}{n}\left[\frac{2}{1-(1-r)^2} + \frac{n-2}{1-(1-r)^3}\right] = \frac{1}{n}\left[\frac{2}{\overline{\mathcal{A}}(2)} + \frac{n-2}{\overline{\mathcal{A}}(3)}\right]$$

- E_r^S: In this scenario, all of C_{i-1}, C_i and C_{i+1} are allowed to acknowledge to S_i. So, $E_r^S = \mathcal{E}_r^a \mathcal{N}_r^{(a,S)}$ where

$$\begin{aligned}\mathcal{N}_r^{(a,S)} =& \frac{1}{n}\left[(n-2)\sum_{j=0}^{+\infty}(1-r)^{3j}\sum_{k=1}^{3}k\binom{3}{k}r^k(1-r)^{3-k}\right.\\ &\left.+2\sum_{j=0}^{+\infty}(1-r)^{2j}\sum_{k=1}^{2}k\binom{2}{k}r^k(1-r)^{2-k}\right]\\ =&\frac{1}{n}\left[\frac{2\overline{\mathcal{N}}(2)}{\overline{\mathcal{A}}(2)}+\frac{(n-2)\overline{\mathcal{N}}(3)}{\overline{\mathcal{A}}(3)}\right]\end{aligned}$$

- E_t^C: A coder transmits coded data packets to D and ACKs to those source nodes from which it has received or overheard original packets. Hence, $E_t^C = \mathcal{E}_t^d \mathcal{N}_t^{(d,C)} + \mathcal{E}_t^a \mathcal{N}_t^{(a,C)}$ where $\mathcal{N}_t^{(d,C)} = \frac{1}{r}$ and $\mathcal{N}_t^{(a,C)} = \mathcal{N}_r^{(a,S)}$.
- E_r^C: A coder receives or overhears original data packets from the corresponding source nodes. It also receives DACK from D. Therefore, $E_r^C = \mathcal{E}_r^d \mathcal{N}_r^{(d,C)} + \mathcal{E}_r^a \mathcal{N}_r^{(a,C)}$. We first consider $\mathcal{N}_r^{(d,C)}$. One transmission initiated by S_1 or S_n is bound to trigger two receptions (C_1 and C_2 hear S_1, C_{n-1} and C_n hear S_n), and one transmission initiated by $S_i(i \neq 1, n)$ triggers three receptions. So,

$$\mathcal{N}_r^{(d,C)} = \mathcal{N}_t^{(d,S)}\frac{2\times 2+3(n-2)}{n} = \left(3-\frac{2}{n}\right)\frac{1}{n}\left[\frac{2}{\overline{\mathcal{A}}(2)}+\frac{n-2}{\overline{\mathcal{A}}(3)}\right]$$

  And as for $\mathcal{N}_r^{(a,C)}$, $\mathcal{N}_r^{(a,C)} = \mathcal{N}_t^{(d,C)}\cdot r = 1$ since D responses with a DACK to each correctly received coded packet.
- E_t^D: D only transmits DACK, one for each correctly received coded packet. So, $E_t^D = \mathcal{E}_t^a \mathcal{N}_t^{(a,D)}$ and $\mathcal{N}_t^{(a,D)} = \mathcal{N}_t^{(d,C)}\cdot n\cdot r = n$
- E_r^D: D only receives coded data packets from coders, that is, $E_r^D = \mathcal{E}_r^d \mathcal{N}_r^{(d,D)}$. As mentioned above, any transmission towards D triggers a receiving operation regardless whether the transmission is successful or not. Therefore, $\mathcal{N}_r^{(d,D)} = n\mathcal{N}_t^{(d,C)} = \frac{n}{r}$.

It is the end of proof.

Appendix IV: Proof of (10)

Proof: Assuming that each source node transmits one packet, the total energy consumption of the none-LNC convergecast scheme should be

$$\hat{E} \approx n\left(\hat{E}_t^S+\hat{E}_r^S\right)+n\left(\hat{E}_t^C+\hat{E}_r^C\right)+\left(\hat{E}_t^D+\hat{E}_r^D\right)$$

where $\hat{E}_t^S$, $\hat{E}_r^S$, $\hat{E}_t^C$, $\hat{E}_r^C$, $\hat{E}_t^D$ and $\hat{E}_r^D$ can be calculated as follows.

- $\hat{E}_t^S$: $\hat{E}_t^S = \hat{\mathcal{E}}_t^d \hat{\mathcal{N}}_t^{(d,S)}$. $\hat{\mathcal{N}}_t^{(d,S)} = \sum_{k=1}^{+\infty} k(1-r)^{k-1}r = \frac{1}{r}$.
- $\hat{E}_r^S$: $\hat{E}_r^S = \hat{\mathcal{E}}_r^a \hat{\mathcal{N}}_r^{(a,S)}$. $\hat{\mathcal{N}}_r^{(a,S)} = 1$ because each source node gets one and only one ACK.
- $\hat{E}_t^C$: C_i acknowledges the data packet sent by S_i and forwards it to D. Thus,

$\hat{E}_t^C = \hat{\mathcal{E}}_t^d \hat{\mathcal{N}}_t^{(d,C)} + \hat{\mathcal{E}}_t^a \hat{\mathcal{N}}_t^{(a,C)}$ where $\hat{\mathcal{N}}_t^{(d,C)} = \sum_{k=1}^{+\infty} k(1-r)^{k-1} r = \frac{1}{r}$ and $\hat{\mathcal{N}}_t^{(a,C)} = 1$.

- $\hat{E}_r^C$: C_i performs a reception for i) each transmission of S_i, and ii) the ACK from D (only one ACK). So, $\hat{E}_r^C = \hat{\mathcal{E}}_r^d \hat{\mathcal{N}}_r^{(d,C)} + \hat{\mathcal{E}}_r^a \hat{\mathcal{N}}_r^{(a,C)}$ where $\hat{\mathcal{N}}_r^{(d,C)} = \hat{\mathcal{N}}_t^{(d,S)} = \frac{1}{r}$ and $\hat{\mathcal{N}}_r^{(a,C)} = 1$.
- $\hat{E}_t^D$: D needs to send, in total, n ACKs. Consequently, $\hat{\mathcal{N}}_t^{(a,D)} = n$ and $\hat{E}_t^D = \hat{\mathcal{E}}_t^a \hat{\mathcal{N}}_t^{(a,D)} = n\hat{\mathcal{E}}_t^a$.
- $\hat{E}_r^D$: n coders initiate $n\hat{\mathcal{N}}_t^{(d,C)}$ data transmissions, and each transmission trigger a reception of D. Hence, $\hat{\mathcal{N}}_r^{(d,D)} = n\hat{\mathcal{N}}_t^{(d,C)} = \frac{n}{r}$ and $\hat{E}_r^D = \hat{\mathcal{E}}_r^d \hat{\mathcal{N}}_r^{(d,D)} = \frac{n}{r}\hat{\mathcal{E}}_r^d$

It is the end of proof.

References

[1] S. Y. R. Li, R. W. Yeung, C. Ning, "Linear network coding," *IEEE Transactions on Information Theory,* vol.49, no.2, pp. 371-381, 2003. Article (CrossRef Link)

[2] R. Koetter, M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking,* vol.11, no.5, pp. 782-795, 2003. Article (CrossRef Link)

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, *et al.*, "Network information flow," *IEEE Transactions on Information Theory,* vol.46, no.4, pp. 1204-1216, 2000. Article (CrossRef Link)

[4] P. Pahlavani, V. Derhami, A. M. Z. Bidoki, "FENC: Fast and Efficient Opportunistic Network Coding in wireless networks," *KSII Transactions on Internet and Information Systems,* 2011.1, vol.5, no.1, pp. 52-67, 2011.1, 2011. Article (CrossRef Link)

[5] D. Wei, C. Chun, L. Xue*, et al.*, "A Lightweight and Density-Aware Reprogramming Protocol for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing,* vol.10, no.10, pp. 1403-1415, 2011. Article (CrossRef Link)

[6] D. Wang, Q. Zhang, J. Liu, "Partial network coding: Concept, performance, and application for continuous data collection in sensor networks," *ACM Transactions on Sensor Networks,* vol.4, no.3, 2008. Article (CrossRef Link)

[7] L. Wang, Y. Yang, W. Zhao, "Network coding-based multipath routing for energy efficiency in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking,* vol.2012, no.1, p. 115, 2012. Article (CrossRef Link)

[8] J. Wang, X. Wang, S. Zhang*, et al.*, "An Efficient Reliable Communication Scheme in Wireless Sensor Networks Using Linear Network Coding," *International Journal of Distributed Sensor Networks,* vol.2012, no.Article ID 605494, p. 11 pages, 2012. Article (CrossRef Link)

[9] A. Keshavarz-Haddad, R. Riedi, "Bounds on the Benefit of Network Coding for Wireless Multicast and Unicast," *IEEE Transactions on Mobile Computing,* vol.PP, no.99, pp. 1-1, 2012. Article (CrossRef Link)

[10] S. Chachulski, M. Jennings, S. Katti*, et al.*, "Trading structure for randomness in wireless opportunistic routing," in *Proc. of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 169-180, August 27-31, 2007. Article (CrossRef Link)

[11] Z. Yulong, Z. Jia, Z. Baoyu, "A fully distributed opportunistic network coding scheme for cellular relay networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2937-2942, 7-10 April 2013, 2013. Article (CrossRef Link)

[12] J. L. Rebelatto, B. F. Uchoa-Filho, L. Yonghui*, et al.*, "Multiuser Cooperative Diversity Through Network Coding Based on Classical Coding Theory," *IEEE Transactions on Signal Processing,* vol.60, no.2, pp. 916-926, 2012. Article (CrossRef Link)

[13] L. Hong, Y. HongYi, L. Linhai*, et al.*, "On the Cascading Data Collection Mechanism in

Wireless Sensor Networks," in *Proc. of 2007 International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007).* pp. 2479-2482, 21-25 Sept. 2007, 2007. Article (CrossRef Link)

[14] Kasun Samarasinghe, Thiemo Voigt, Luca Mottola, *et al.*, "Network Coding with Limited Overhearing," in *Proc. of 8th European Conference on Wireless Sensor Networks (EWSN2011)*, February 23-25, 2011, 2011. Article (CrossRef Link)

[15] Thiemo Voigt, Utz Roedig, Olaf Landsiedel, *et al.*, "Practical Network Coding in Sensor Networks: Quo Vadis?," in *Proc. of 3rd International Workshop on Networks of Cooperating Objects*, April 16, 2012. Article (CrossRef Link)

[16] L. Keller, E. Atsan, K. Argyraki, *et al.*, "SenseCode: Network coding for reliable sensor networks," *ACM Transactions on Sensor Networks,* vol.9, no.2, pp. 1-20, 2013. Article (CrossRef Link)

[17] O. Gnawali, R. Fonseca, K. Jamieson, *et al.*, "Collection tree protocol," in *Proc. of 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 1-14, November 4-6, 2009. Article (CrossRef Link)

[18] Z. Tang, H. Wang, Q. Hu, *et al.*, "How network coding benefits converge-cast in wireless sensor networks," *KSII Transactions on Internet and Information Systems,* vol.7, no.5, pp. 1180-1197, 2013. Article (CrossRef Link)

[19] H. Tracey, M. Muriel, S. Jun, *et al.*, "On randomized network coding," in *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, pp. 11-20, October 1-3, 2003.

[20] J. Hill, R. Szewczyk, A. Woo, *et al.*, "System architecture directions for networked sensors," *SIGPLAN Not.,* vol.35, no.11, pp. 93-104, 2000. Article (CrossRef Link)

[21] X. Zhang, K. G. Shin, "E-MiLi: energy-minimizing idle listening in wireless networks," in *Proc. of the 17th ACM Annual International Conference on Mobile Computing and Networking*, pp. 205-216, 2011. Article (CrossRef Link)

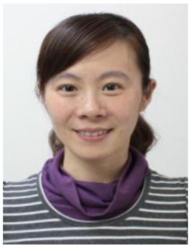
[22] W. Ye, J. Heidemann, D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking,* vol.12, no.3, pp. 493-506, 2004. Article (CrossRef Link)

[23] N. Baccour, A. Koubaa, L. Mottola, *et al.*, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Transactions on Sensor Networks,* vol.8, no.4, p. 34, 2012. Article (CrossRef Link)

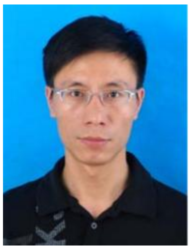
[24] Y. Zou, Y.-D. Yao, B. Zheng, "Opportunistic distributed space-time coding for decode-and-forward cooperation systems," *IEEE Transactions on Signal Processing,* vol.60, no.4, pp. 1766-1781, 2012. Article (CrossRef Link)

Zhenzhou Tang received his B.S. degree in Electronic Engineering and M.S. degree in Communications and Information System from Zhejiang University, Hangzhou, China, in 2001 and 2004 respectively. He is currently an associate professor of the College of physics and electronic information engineering, Wenzhou University, Wenzhou, China, and is pursuing his Ph.D. in Wireless Communication at Dalian University of technology. His research interests include cooperative communications, network coding, and wireless sensor networks.

Hongyu Wang received his B.S. degree from Jilin University of Technology, Changchun, China, in 1990, and M.S. degrees from Graduate School of Chinese Academy of Sciences, Changchun, China, in 1993, both in Electronic Engineering. He received the Ph.D. in Precision Instrument and Optoelectronics Engineering from Tianjin University, Tianjin, China, in 1997. Currently, he is a Professor in the institute of Information Science and Communication Engineering, Dalian University of Technology, China. His research interests include algorithmic, optimization, and performance issues in wireless ad hoc, mesh and sensor networks.



Qian Hu received her B.S. degree in Communication Engineering from Nanjing Institute of Posts and Telecommunications, Nanjing, China, in 2001, and M.S. degree in Circuits and Systems from Zhejiang University, Hangzhou, China, in 2005. She is now an associate professor of the College of physics and electronic information engineering, Wenzhou University, Wenzhou, China. Her research interests are in the areas of wireless communications, including cooperative communication, network coding, and MAC and network layer protocols for wireless sensor networks.



Xiukai Ruan received the M.S and Ph.D. degrees from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006 and 2011, respectively. He is currently working as the vice director in Institute of Laser and Optoelectronics Technology, Wenzhou University. His research interests are broadly in the areas of wireless communications, signal and information processing, pattern recognition, machines learning, computational neuroscience, and complicated nonlinear dynamics.