Link-by-Link Feedback Mechanism for Intra-Session Random Linear Network Coding in Wireless Sensor Networks

Vahid Shah-Mansouri and Vincent W.S. Wong
Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, Canada
e-mail: {vahids, vincentw}@ece.ubc.ca

Abstract*—In this paper, we study the use of intra-session random linear network coding (RLNC) in wireless sensor networks. In RLNC, intermediate nodes buffer the packets received from upstream nodes. Using intra-session RLNC, intermediate nodes transmit coded packets by performing coding on the packets of various flows. The main challenge in using intra-session RLNC is to determine how many coded packets each node requires to transmit for each flow such that the sink can decode the packets of all the flows. We analytically find the time at which a node can stop transmission of packets for a particular flow without interrupting the decoding process at the sink. Using this analysis, we design a link-by-link feedback mechanism to acknowledge the packets of a particular flow. An intermediate node generates feedback packets for its upstream nodes. The upstream node stops transmission of a flow when it receives an acknowledgment for that flow. Simulation results show that the link-by-link feedback mechanism with intra-session RLNC achieves lower power consumption compared to RLNC without intra-session coding and also the CCACK algorithm [1].

I. Introduction

Wireless sensor networks (WSNs) consist of battery-powered inexpensive wireless devices called sensor nodes. Sensors transmit data via a multihop communication path towards a control center called the sink [2]. Since sensors have limited power supply, it is of interest to design power efficient data gathering techniques for WSNs. The wireless links are prone to failure due to fading and interference and improving the reliability can help to achieve lower power consumption.

It is shown that *random linear network coding* (RLNC) can improve the reliability of end-to-end communication in presence of lossy links [3], [4]. Therefore, it is of interest to study the use of RLNC for WSNs. In RLNC, the intermediate nodes store the received packets and transmit coded packets by linearly combining the buffered packets. The coding coefficients are appended in the header of packets to help the decoding process. To reduce the time for the sink to decode the packets, the source divides the packets into groups called *generation*. The sink can decode the packets of one generation as soon as it has enough packets with independent coding coefficients of that generation. For a multihop single path or multipath connection between a source and the sink, using RLNC with sufficiently large generation size and code size, the packet-level (i.e., max flow min-cut) capacity of a unicast and multicast session can be achieved [4], [5]. Intra-session RLNC is referred to the case where intermediate nodes perform coding operations on the packets of various flows [6]. The sink can decode the packets of a flow if it receives enough packets from all the flows in this case.

The main variant of RLNC which has received attention is network coding based opportunistic routing. Opportunistic routing exploits the broadcast nature of wireless links. The intermediate nodes broadcast the coded packets. The downstream nodes overhearing the packets opportunistically forward them towards the sink. MORE [7] is a network coding based opportunistic routing algorithm. Some other works have improved MORE in different directions [8]–[11]. These algorithms use a credit based system which measures the link loss rates offline and assigns credits to each node. The number of packets each node transmits depends on this credit. This approach is not resilience to dynamic change of loss rates. It also imposes overhead for measuring the loss rates. The cumulative coded acknowledgement (CCACK) algorithm proposed in [1] aims to overcome these challenges. An acknowledgement vector is added to each transmitted packet based on the stored packets at the node. Neighboring nodes use this acknowledgment information to decide how many packets they need to transmit.

Although opportunistic routing with network coding can improve the throughput, it may not be suitable for WSNs. In opportunistic routing, the path for flow of the packets from a source to the sink is determined dynamically hop-by-hop. In this paper, we study the use of intra-session RLNC for WSNs. Nodes transmit coded packets at constant rate independent of receiving new packets. The challenge is to determine how many coded packets a source or an intermediate node should transmit. Since the nodes transmit packets at a constant rate, the problem is equivalent to when a node can stop transmission of a particular flow. As a solution, we develop a link-by-link feedback mechanism used to acknowledge various flows passing a node. An intermediate node stops transmission of packets for a flow as soon as it receives feedback for that flow. The contributions of this paper are as follows:

- We analytically find the time at which a node can stop transmission of packets for a particular flow without

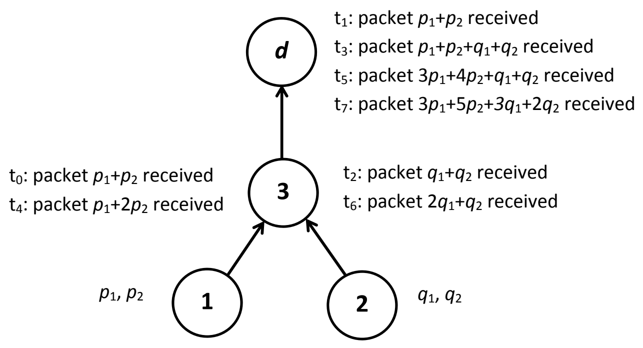
d
t1: packet p1+p2 received
t3: packet p1+p2+q1+q2 received
t5: packet 3p1+4p2+q1+q2 received
t7: packet 3p1+5p2+3q1+2q2 received
t0: packet p1+p2 received
t4: packet p1+2p2 received
3
t2: packet q1+q2 received
t6: packet 2q1+q2 received
p1, p2
1
2
q1, q2

Fig. 1. An example with two sources 1 and 2 sending coded packets to sink d via the intermediate node 3. The generation size is two.

interrupting the decoding process at the sink.

- We propose a novel link-by-link feedback mechanism. In this mechanism, every intermediate node generates feedback packets for its upstream neighboring node.
- Simulation results show that intra-session RLNC achieves lower power consumption compared to the RLNC without intra-session coding and also CCACK algorithm [1].

This paper is organized as follows: The system model along with a description of intra-session RLNC is summarized in Section II. In Section III, link-by-link feedback mechanism is proposed. Performance evaluation and comparison are presented in Section IV. Conclusions are given in Section V.

II. System Model

We consider a WSN with a set of sources and one sink. Without loss of generality, we assume that the sources do not relay information of other flows. Let $\mathcal{V}$ denote the set of sensor nodes. There is a pre-determined unicast route between each source and the sink. Let $\mathcal{F}$ denote the set of all the flows. A flow is referred to the data flow from a source to the sink. The set of nodes relaying packets of flow $f \in \mathcal{F}$ is denoted by $\mathcal{V}^f$. The set of flows relayed by intermediate node $i \in \mathcal{V}$ is denoted by $\mathcal{F}_i$.

Each source divides the generated packets into groups called *generations*. An intermediate node i buffers the received packets of one generation from the upstream nodes. These packets belong to flows in $\mathcal{F}_i$. Node i creates coded packets by combining the buffered packets using random coefficients taken from a Galois field with size q (i.e., GF(2^q)). It appends the coding coefficients in the packet header. The vector of coding coefficients is called the encoding vector.

Fig. 1 shows a WSN with two flows from sources 1 and 2 to sink d through intermediate node 3. The generation size is 2. We consider a system with time slots $t_0, \ldots, t_7$. Nodes 1 and 2 transmit their first coded packet at times t_0 and t_2, respectively. Node 3 transmits two packets containing information of the received packets at times t_1 and t_3. Then, nodes 1 and 2 transmit their second packet at times t_4 and t_6, respectively. Intermediate node 3 transmits the last two packets at times t_5 and t_7. The sink can decode the packets after time t_7.

Let r_f denote the rate at which the packets of flow f are transmitted in the network. The rate at which node i transmits packets to its downstream node j is $\sum_{f \in \mathcal{F}_i \cap \mathcal{F}_j} r_f$ at the beginning. Node i reduces its transmission rate by r_f as soon as it receives acknowledgment for flow $f \in \mathcal{F}_i$. The node stops transmission if it receives acknowledgement for all the flows in $\mathcal{F}_i$. Let $\alpha_{i,j}$ denote the packet success probability of the link from node i to j. The rate at which packets are transmitted on the link from node i to j times the packet success probability $\alpha_{i,j}$ is called the *effective transmission rate* of node i to j. Since various sources may generate packets at different rates, we set the generation size of different flows such that all the flows have the same generation size to transmission rate ratio. This ratio is denoted by Δ. Let K_f denote the generation size of flow f. Then, we have $K_f/r_f = \Delta$ for any flow $f \in \mathcal{F}$. The difference between the time that the sources start transmission of the packets and the time that the sink can decode the packets is called the *decoding delay*.

Let $\Lambda_i(t)$ denote a matrix composed of the encoding vectors of packets in the buffer of node i at time t. Each column of $\Lambda_i(t)$ contains the encoding vector of one packet. Each row in matrix $\Lambda_i(t)$ corresponds to the coefficients of a packet from a generation of a particular flow. The *rank* of this matrix, denoted by $|\Lambda_i(t)|$, represents the number of packets with independent encoding vectors in the buffer of node i.

We define $|\Lambda_i(t)|_F$ as the rank of matrix $\Lambda_i(t)$ with respect to the flows in $F \subseteq \mathcal{F}_i$. It is computed by replacing the rows of matrix $\Lambda_i(t)$ which do not belong to flows in F by zero vectors and then computing the rank of the resulting matrix. A packet W transmitted by node i at time t is an *innovative* packet for downstream node j if it can increase the rank of matrix $\Lambda_j(t)$ by one. For large Galois field sizes, $|\Lambda_i(t)| > |\Lambda_j(t)|_{\mathcal{F}_i}$ would result that W is an innovative packet with probability close to one [4]. The number of innovative packets in the buffer of node j received from node i at time t is $|\Lambda_j(t)|_{\mathcal{F}_i}$.

III. Link-by-Link Feedback Mechanism Design

In link-by-link feedback mechanism, the downstream nodes of a node i are responsible to acknowledge a particular flow at this node. In network coding based opportunistic routing, since the broadcast nature of the wireless link is employed, all the downstream nodes require to cooperate to decide when node i can stop transmission of a particular flow f. However, the cooperation requires the downstream nodes to exchange information upon receiving any new packet. This overhead increases the power consumption dramatically. Our goal is to send one feedback packet per generation per flow. We assume that the feedback packets are not lost.

Assume that node j is the downstream node of node i such that $f \in \mathcal{F}_j$. In link-by-link feedback mechanism, node j is responsible to transmit the feedback packet of flow f to node i. Since node i stops transmission of packets of flow f (i.e., reduces its transmission rate by r_f) upon receiving the feedback, the feedback should be transmitted such that it does not affect the transmission rate of other flows or it should not interrupt the decoding process at the sink. Therefore, to design the feedback mechanism, we need to determine the time that node i can stop transmitting the packets of one generation for flow f. We notice that the transmitted packets by node i contain information of all the flows in $\mathcal{F}_i$. Therefore, one cannot distinguish to which flow a transmitted packet belongs to. This is the main challenge that we address in this section.

A. When Can Node i Stop Transmission of Flow f?

Among all the flows in set $\mathcal{F}$, we only study one particular flow f in this section. We notice that the analysis can be used for other flows similarly. For the sake of simplicity, we denote the set of nodes in the path set $\mathcal{V}^f$ by $\{1, 2, \ldots, |\mathcal{V}^f|\}$, where node 1 is the source of flow f and node $|\mathcal{V}^f|$ is the sink.

Let $\mathcal{F}_{i,i+1}$ denote $\mathcal{F}_i \cap \mathcal{F}_{i+1}$. If flow f is the only flow shared between node i and node $i+1$ (i.e, $\mathcal{F}_{i,i+1} = \{f\}$), node i can stop transmitting the packets of flow f as soon as it successfully transmits the K_fth innovative packet of flow f. After this time, the rate at which node i transmits the innovative packets of flow f to node $i+1$ is zero. Therefore, node i can reduce its transmission rate by r_f [3], [12].

Now, consider the case that the number of flows relaying from node i to node $i+1$ is more than one. Node i can stop transmission of flow $f \in \mathcal{F}_{i,i+1}$ (i.e., reduce its transmission rate by r_f) if the rate at which innovative packets of flows $\mathcal{F}_{i,i+1}\backslash\{f\}$ are transmitted does not alter after reducing the rate. This guarantees that reducing the rate does not interrupt the decoding process at the sink. Every transmitted packet from node i contains information of all the flows in $\mathcal{F}_i$. Thus, one cannot distinguish to which flow a transmitted packet belongs to. Consequently, it is not feasible to determine the time that node i can stop transmission of flow f by counting the number of transmitted innovative packets of this flow.

Consider the end-to-end path of flow f from the source to the sink. We start from the source of flow f and move along this path to the sink to determine when each of the nodes in set $\mathcal{V}^f$ can stop transmission of flow f. Since the source of flow f does not relay the packets of other flows, it can stop transmission after successfully transmitting K_f packets with independent encoding vectors to node 2.

Let τ_1^f denote the time that the source of flow f successfully transmits the K_fth innovative packet. Since after time τ_1^f node 2 does not receive any innovative packet from the source of flow f, it can stop transmission of the flow as soon as it can transmit all the packets of that flow after time τ_1^f. Note that the packets transmitted by node 2 to node 3 contain information of all the packets in the buffer of node 2. The first time the rank of the packets for flows $\mathcal{F}_{2,3}$ in nodes 2 and 3 are equal after τ_1^f is the first time that we can ensure all the information of flow f are transmitted from node 2 to node 3. We can extend this concept for any intermediate node i. Let τ_i^f denote the first time after τ_{i-1}^f at which the two ranks are equal. We can define time τ_i^f for $i = 1, \ldots, |\mathcal{V}^f| - 1$ as

$$\tau_i^f = \min\left\{t \;\middle|\; t > \tau_{i-1}^f,\ |\Lambda_i(t)|_{\mathcal{F}_{i,i+1}} = |\Lambda_{i+1}(t)|_{\mathcal{F}_{i,i+1}}\right\}, \tag{1}$$

where $\tau_0^f = 0$. First, we show how this time can be calculated in practice. Then, we show that reducing the rate by r_f at time τ_i^f does not change the rate at which node i transmits the innovative packets meaning that node i can reduce its transmission rate at τ_i^f. The following lemma shows how time τ_i^f can be determined.

Lemma 1: The time τ_i^f for $i = 1, \ldots, |\mathcal{V}^f| - 1, \quad f \in \mathcal{F}_i$ is

$$\tau_i^f = \min\left\{t \;\middle|\; |\Lambda_{i+1}(t)|_{\mathcal{F}_{i,i+1}} - |\Lambda_{i+1}(t)|_{\mathcal{F}_{i,i+1}\backslash\{f\}} = K_f\right\}.$$

The proof of Lemma 1 can be found in Appendix A. Now, we show that reducing the transmission rate after time τ_i^f does not change the rate at which node i transmits innovative packets of flows $\mathcal{F}_{i,i+1}\backslash\{f\}$. Our model analyzes the system on a *long term operation* period and the transmission rates are *expected* values. Let $z_{i,i+1}(F)$ denote the rate at which node i transmits the innovative packets of flows in set F to node $i+1$. We first focus on flows $f \in \mathcal{F}_i$ while flow f has the lowest value of τ_i^f among all the flows in $\mathcal{F}_{i,i+1}$ (i.e., $\tau_i^f = \min_{s\in\mathcal{F}_{i,i+1}} \tau_i^s$). For the sake of simplicity, we assume that τ_i^f is not equal to τ_i^s for $s \in \mathcal{F}_{i,i+1}\backslash\{f\}$ in the rest of the paper. The following theorem shows that node i can reduce its transmission rate at time τ_i^f without changing the rate at which innovative packets of flows $\mathcal{F}_{i,i+1}\backslash\{f\}$ are transmitted to node $i+1$.

Theorem 1: The value of $z_{i,i+1}(\mathcal{F}_{i,i+1}\backslash\{f\})$ does not change if node i reduces its transmission rate to node $i+1$ from $\sum_{s\in\mathcal{F}_{i,i+1}} r_i$ to $\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} r_i$ (i.e., reduces by r_f) at time τ_i^f, where τ_i^f is equal to $\min\limits_{s\in\mathcal{F}_{i,i+1}} \tau_i^s$.

The proof of Theorem 1 can be found in Appendix B. Based on Theorem 1, after time τ_i^f, the rate at which node i receives innovative packets from flows $\mathcal{F}_{i,i+1}\backslash\{f\}$ is less than the effective rate of node i to node $i+1$. Since the rate of receiving innovative packets is not increased over time, the input rate is always less than $\alpha_{i,i+1}\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} r_s$ for $t > \tau_i^f$. We notice that the transmission rate of node i is not reduced such that it falls below the input rate of innovative packets after time τ_i^f. Therefore, the rate at which node i transmits innovative packets is always greater than the rate at which this node receives the innovative packets after time τ_i^f. Let $\varphi(t) \subseteq \mathcal{F}_{i,i+1}$ denote the set of flows that node i receives innovative packets from them at time t. The following lemma characterizes the rate at which node i transmits innovative packets of flows $\varphi(t)$ to node $i+1$.

Lemma 2: The rate at which node i transmits the innovative packets of flows $\varphi(t) \subset \mathcal{F}_{i,i+1}$ to node $i+1$ from time 0 to t is the weighted sum of the rates of these flows. That is,

$$z_{i,i+1}(\varphi(t)) = \sum_{\rho\in\varphi(t)} r_\rho \beta_\rho^i(\varphi(t)), \tag{2}$$

where $\beta_\rho^i(\varphi(t))$ is a non-negative coefficient for node i and flow ρ corresponding to the set of flows $\varphi(t)$.

The proof of Lemma 2 can be found in Appendix C. Using Lemma 2, we generalize Theorem 1 as follows:

Theorem 2: The value of $z_{i,i+1}(\varphi(t))$ does not change if node i reduces its transmission rate to node $i+1$ by r_f at time τ_i^f for any $\varphi(t) \subset \mathcal{F}_{i,i+1}\backslash\{f\}$.

Algorithm 1 Link-by-Link Feedback Mechanism for Intra-Session RLNC at Node j.

```
1:  input: $\mathcal{F}_j, K_f, \forall f \in \mathcal{F}_j$
2:  for each generation
3:      $\mathcal{A} := \emptyset$
4:      while $\mathcal{A} \neq \mathcal{F}_j$ and a new packet is arrived at time $t$
5:          for flow $f \in \mathcal{F}_j \backslash \mathcal{A}$
6:              $i :=$ upstream node of node $j$ relaying packets of $f$
7:              if $|\Lambda_j(t)|_{\mathcal{F}_{i,j}} - |\Lambda_j(t)|_{\mathcal{F}_{i,j} \backslash \{f\}}$ is equal to $K_f$
8:                  Send feedback control packet for flow $f$ to node $i$
9:                  $\mathcal{A} := \mathcal{A} \cup \{f\}$
10:             end if
11:         end for
12:     end while
13: end for
```

The proof of Lemma 2 can be found in Appendix D. Theorem 2 shows that τ_i^f is the time that node i can reduce the transmission rate by r_f.

B. The Feedback Mechanism

In link-by-link feedback mechanism, each intermediate node is responsible to generate feedback control packet for its upstream neighboring nodes. Based on Theorem 2, node i can stop transmission of flow f at time τ_i^f. If node j is the downstream neighboring node of node i relaying packets of flow f, node j can generate the feedback control packet of flow f for node i as soon as $|\Lambda_j(t)|_{\mathcal{F}_{i,j}} - |\Lambda_j(t)|_{\mathcal{F}_{i,j} \backslash \{f\}} = K_f$.

Algorithm 1 shows the link-by-link feedback mechanism at node j. $\mathcal{A}$ denotes the set of flows which have already been acknowledged by node j. Upon receiving a packet from an node i, node j checks whether it has enough packets for any of the flows (Step 7). If there is any flow with enough number of packets, node j sends the feedback control packet to node i (Step 8). The algorithm continues until node j sends feedback control packet for all the flows in $\mathcal{F}_j$.

IV. Performance Evaluation

In this section, we present the performance analysis and comparison for the proposed feedback mechanism. We consider a topology with 130 sensors randomly deployed in a 100 m $\times$ 100 m square area. 30 sensors are randomly chosen as sources. The sink is placed at the upper-right corner. We construct the minimum hop single path connection from each source to the sink. For the packet success probability, we follow the model in [13]. For output link of node i to node j, the packet success probability $\alpha_{i,j}$ is randomly chosen with probability $\lambda \alpha_{i,j}^{(\lambda-1)}$, where λ is a tunable design parameter. The results are averaged over 1000 simulation runs. The power consumption of node i is the number of packets transmitted by node i times the power consumed to transmit one packet. To normalize the power consumption, we divide it by the number of flows, the generation size, and the power consumed to transmit one packet. We assume that various sources generate and transmit packets at the same rate. Therefore, the generation size of various flows are equal in our simulations.

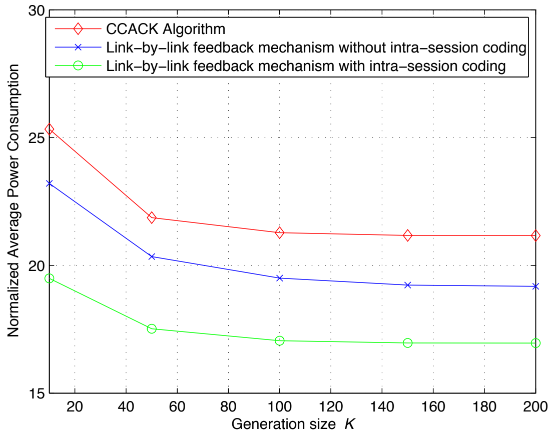

Fig. 2. Normalized average power consumption versus generation size K.

We compare our proposed scheme with RLNC without intra-session coding and CCACK algorithm [1]. In RLNC without intra-session coding, the packets of various flows are not combined at the intermediate nodes. For the feedback mechanism in this case, an intermediate node j transmits feedback of flow f to its upstream node i if it receives K_f innovative packets of flow f. CCACK is a link-by-link acknowledgement mechanism for network coded opportunistic routing with intra-session network coding. Source and intermediate nodes append an acknowledgement vector to each transmitted packet. It allows downstream nodes to cooperatively acknowledge the packets transmitted by an intermediate node.

Fig. 2 compares the normalized power consumption of different mechanisms when generation size increases. The normalized power consumption decreases by increasing the number of packets in each generation. The RLNC with/without intra-session network coding performs better than CCACK in terms of power consumption because CCACK uses acknowledgment packet for every transmitted packet while the other two algorithms use only one feedback packet for each generation. For values greater than 100, the improvement in the power consumption is negligible. Therefore, for the rest of simulations, we set $K_f = 100$ for all the flows.

Fig. 3 compares the link-by-link feedback mechanism with/without intra-session network coding and CCACK algorithm. We vary λ and measure the power consumption. The power consumption decreases as λ increases since a higher packet success probability leads to a lower number losses. For large values of λ (i.e., $\lambda \geq 7$), since the loss rate of the links approach zero, the benefit of using intra-session coding is negligible. The results show that intra-session network coding can achieve a lower power consumption. Moreover, CCACK has a higher power consumption because it relies on per transmission acknowledgment which imposes a high overhead.

Next, we present the *decoding delay* performance. The decoding delay is the difference between the first time that the sink decodes all the packets of a generation and the time that the sources start transmission of that generation. We divide the total delay by the generation size in order to obtain the normalized decoding delay. In RLNC without intra-session

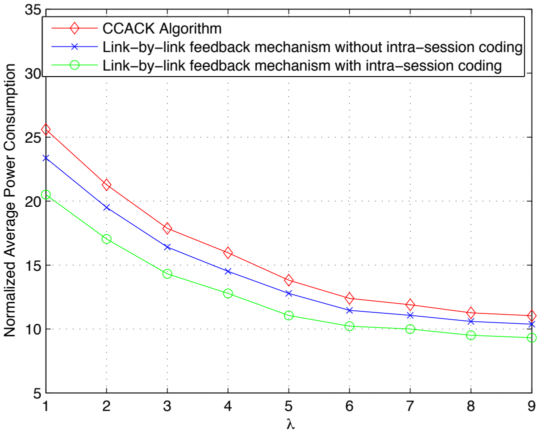

Fig. 3. Performance of the normalized average power consumption for RLNC with and without intra-session network coding and CCACK algorithm.

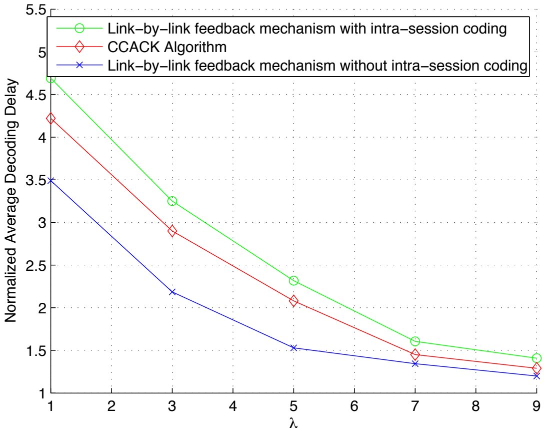

Fig. 4. Performance of the normalized decoding delay between RLNC with and without intra-session network coding and CCACK algorithm.

network coding, the sink decodes packets from different flows independently. Therefore, the decoding delay is different for various flows. We consider the average delay of various flows in the network. Fig. 4 compares the normalized decoding delay for RLNC with and without intra-session network coding. In RLNC with intra-session network coding and CCACK, all the flows are decoded at the same time. Therefore, they have higher decoding delays compared to RLNC without intra-session network coding. However, since CCACK uses opportunistic routing, it can use the broadcast nature of the links and achieves better decoding delay compared to RLNC with intra-session coding. As λ increases, the decoding delay of both techniques approaches since the number of lost packet diminishes and the effect of using network coding disappears.

V. Conclusions

In this paper, we studied the use of intra-session random linear network coding in wireless sensor networks. Through analytical modeling, we showed when an intermediate node can stop transmission of the packets for a particular flow without changing the average rate at which that node transmits innovative packets. Then, we proposed a link-by-link feedback mechanism in which an intermediate node transmits feedback control packet to its upstream neighboring node and acknowledges each flow individually. The upstream node reduces its transmission rate when it receives a feedback packet. Simulation results show that the link-by-link feedback mechanism with intra-session RLNC achieves a lower power consumption compared to RLNC without intra-session coding and also the CCACK algorithm [1]. This is achieved at expense of a higher decoding delay.

Appendix

A. *Proof of Lemma 1*

For simplicity of the proof, let $g^f_{i+1}(t) = |\Lambda_{i+1}(t)|_{\mathcal{F}_{i,i+1}} - |\Lambda_{i+1}(t)|_{\mathcal{F}_{i,i+1}\backslash\{f\}}$. The function $g^f_{i+1}(t)$ is an increasing function in time since it is not possible that node $i+1$ receives a packet which is innovative for all the flows except f while this packet is not innovative. $g^f_{i+1}(t)$ shows the number of packets with exclusive information of flow f at the buffer of node $i+1$ at time t. Nevertheless, the number of packets with exclusive information of flow f is less than or equal to the number of innovative packets of flow f. Therefore, function $g^f_{i+1}(t)$ is upper bounded by K_f. Now, we proof the lemma by mathematical induction. At time τ^f_1, node 2, which is the downstream node of node 1 (i.e., the source node of flow f), has received the K_fth innovative packet from the source.

Since the source does not relay packets of other flows, we have $g^f_2(\tau^f_1) = K_f$. This is the basis of our induction. Now, as the hypothesis of the induction, assume that $g^f_{i+1}(\tau^f_i)$ is equal to K_f while $g^f_{i+1}(t) < K_f$ for $t < \tau^f_i$. We need to prove that $g^f_{i+2}(\tau^f_{i+1}) = K_f$, while $g^f_{i+2}(t) < K_f$ for $t < \tau^f_{i+1}$. Since $\tau^f_{i+1} > \tau^f_i$ and $g^f_i(t)$ is an increasing function and is upper bounded by K_f, we have $g^f_{i+1}(\tau^f_{i+1}) = K_f$. According to equation (1), at time τ^f_{i+1}, node $i+2$ has the same information that node $i+1$ has about flows in set $\mathcal{F}_{i+1,i+2}$, which implies $g^f_{i+2}(\tau^f_{i+1})$ is equal to K_f. To complete the proof, we need to show that $g^f_{i+2}(t) < K_f$ for $t < \tau^f_{i+1}$. Since $g^f_{i+2}(t) < g^f_{i+1}(t)$ and $g^f_{i+1}(t) < K_f$ for $t < \tau^f_i$, therefore, $g^f_{i+2}(t) < K_f$ for $t < \tau^f_i$. Now, assume that there exists a time ζ such that $\tau^f_i < \zeta < \tau^f_{i+1}$ and $g^f_{i+2}(\zeta) = K_f$. Thus, at time ζ, node $i+2$ has received a packet with exclusive information of flow f. It means node i has no packets with information of other flows except flow f. However, this is the last packet sent by node $i+1$ with exclusive information of flow f. Therefore, after time ζ, node $i+1$ has no innovative packets for node $i+2$ while this contradicts to the definition of τ^f_{i+1}. Therefore, $g^f_{i+2}(t) < K_f$ for $t < \tau^f_{i+1}$. ■

B. *Proof of Theorem 1*

Since τ^f_i is less than τ^s_i for $s \in \mathcal{F}_{i,i+1}\backslash\{f\}$ (i.e., $\tau^f_i = \min_{s\in\mathcal{F}_{i,i+1}} \tau^s_i$), $z_{i,i+1}(\mathcal{F}_{i,i+1}\backslash\{f\})$ is constant in time period $[0,\ \tau^f_i]$ if node i receives innovative packets from flows in $\mathcal{F}_{i,i+1}\backslash\{f\}$. At time τ^f_i, node i has transmitted all the packets it has received from flows $\mathcal{F}_{i,i+1}$ so far. The number of innovative packets transmitted from node i to node $i+1$

up to time time τ_i^f is $\left|\Lambda_i\left(\tau_i^f\right)\right|_{\mathcal{F}_{i,i+1}}$, which is equal to $g_i^f\left(\tau_i^f\right)+\left|\Lambda_i\left(\tau_i^f\right)\right|_{\mathcal{F}_{i,i+1}\backslash\{f\}}$. Based on Lemma 1, we have $g_i^f\left(\tau_i^f\right) = K_f$. The transmission rate of innovative packets for flows $\mathcal{F}_{i,i+1}\backslash\{f\}$ is $z_{i,i+1}(\mathcal{F}_{i,i+1}\backslash\{f\})$. The average number of innovative packets transmitted from node i to node $i+1$ up to time τ_i^f is $z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right)\tau_i^f+K_f$. The average number of successful transmissions of node i up to time τ_i^f is $\tau_i^f\alpha_{i,i+1}\sum_{s\in\mathcal{F}_{i,i+1}} r_s$. The average number of innovative packets transmitted is less than or equal to the number of successful transmissions of node i. Therefore, we have

$$z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right)\tau_i^f + K_f \leq \tau_i^f\alpha_{i,i+1}\textstyle\sum_{s\in\mathcal{F}_{i,i+1}} r_s, \quad (3)$$

which can be re-arranged as

$$z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right) \leq \alpha_{i,i+1}\textstyle\sum_{s\in\mathcal{F}_{i,i+1}} r_s - \frac{K_f}{\tau_i^f}. \quad (4)$$

If $K_f/\tau_i^f > \alpha_{i,i+1}r_f$, then we obtain $z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right) < \alpha_{i,i+1}\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} r_s$ from (4). Now, we consider the case that $K_f/\tau_i^f \leq \alpha_{i,i+1}r_f$, which is equal to $\Delta/\tau_i^f \leq \alpha_{i,i+1}$. Since the flows in $\mathcal{F}_{i,i+1}\backslash\{f\}$ are still active, the total number of innovative packets transmitted by node i from these flows up to time τ_i^f is less than the total number of packets for one generation of these flows (i.e., $z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right)\tau_i^f < \sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} K_s$). Therefore, we have

$$\begin{aligned} z_{i,i+1}\left(\mathcal{F}_{i,i+1}\backslash\{f\}\right) &< \frac{\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} K_s}{\tau_i^f} \\ &= \frac{\Delta}{\tau_i^f}\textstyle\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} r_s \leq \alpha_{i,i+1}\sum_{s\in\mathcal{F}_{i,i+1}\backslash\{f\}} r_s. \end{aligned}$$

Hence, the rate at which this node transmits the innovative packets does not alter by reducing the transmission rate. ■

C. Proof of Lemma 2

We prove it by mathematical induction. The rate at which the source of flow $\rho \in \varphi(t)$ transmits the innovative packets is the product of the transmission rate and the packet success probability of its outgoing link. This is the basis of induction. Let N_i denote the set of upstream nodes of node i. As inductive step, assume that node $h \in N_i$ transmits packets of flows in set $\varphi(t)\cap\mathcal{F}_{h,i}$ to node i at rate $z_{h,i}\left(\varphi(t)\cap\mathcal{F}_{h,i}\right) = \sum_{s\in\varphi(t)\cap\mathcal{F}_{h,i}} r_s\beta_s^h(\varphi(t)\cap\mathcal{F}_h)$, where $\beta_s^h(\varphi(t)\cap\mathcal{F}_h)$ depends only on $\varphi(t)\cap\mathcal{F}_h$. Then, the rate at which node i receives innovative packets of $\varphi(t)$ is the sum of the rates at which nodes in set N_i transmit innovative packets. The rate at which node i transmits the innovative packets is the minimum of the rate at which it receives innovative packets and its effective transmission rate. That is,

$$\begin{aligned} z_{i,i+1}\left(\varphi(t)\right) &= \min\left\{\sum_{h\in N_i} z_{h,i}\left(\varphi(t)\cap\mathcal{F}_{h,i}\right), \alpha_{i,i+1}\sum_{\rho\in\varphi(t)} r_\rho\right\} \\ &= \min\left\{\sum_{h\in N_i}\sum_{s\in\varphi(t)\cap\mathcal{F}_{h,i}} r_s\beta_s^h(\varphi(t)\cap\mathcal{F}_{h,i}), \alpha_{i,i+1}\sum_{\rho\in\varphi(t)} r_\rho\right\}. \quad (5) \end{aligned}$$

Both arguments of the minimum on the right hand side of (5) are linear combination of r_ρ for all $\rho \in \varphi(t)$. We notice that $\cup_{h\in N_i}\mathcal{F}_{h,i} = \mathcal{F}_i$ and $\mathcal{F}_i \cap \varphi(t) = \varphi(t)$. Therefore, $z_{i,i+1}\left(\varphi(t)\right)$ is a linear combination of r_ρ for all $\rho \in \varphi(t)$ which can be denoted by $z_{i,i+1}\left(\varphi(t)\right) = \sum_{\rho\in\varphi(t)} r_\rho\beta_\rho^i(\varphi(t))$ where $\beta_\rho^i(\varphi(t))$ depends only on $\varphi(t)$. ■

D. Proof of Theorem 2

If $\varphi(t) = \mathcal{F}_{i,i+1}\backslash\{f\}$, based on Theorem 1, reducing the transmission rate does not change $z_{i,i+1}(\varphi(t))$. Now, assume that flow f is not the first flow whose rate is reduced. Since node i has reduced its rate before time τ_i^f, the rate at which it receives the innovative packets before time τ_i^f is less than its effective transmission rate. Based to Lemma 2, the rate at which it receives the packets of flows in set $\varphi(t)$ is less than $z_{i,i+1}(\varphi(t))$. The average rate at which node i received the packets of flow f before time τ_i^f is K_f/τ_i^f. Thus, we have

$$z_{i,i+1}\left(\varphi(t)\backslash\{f\}\right) + \frac{K_f}{\tau_i^f} \leq \alpha_{i,i+1}\textstyle\sum_{\rho\in\varphi(t)} r_\rho. \quad (6)$$

Equation (6) is similar to (4). We can follow the rest of proof of Theorem 1 after (4) and show that $z_{i,i+1}(\varphi(t)\backslash\{f\}) < \alpha_{i,i+1}\sum_{s\in\varphi(t)\backslash\{f\}} r_s$. This shows that reducing the transmission rate by r_f does not change the rate at which this node transmits the innovative packets to node $i+1$. ■

References

[1] D. Koutsonikolas, C.-C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *Proc. of IEEE INFOCOM*, San Diego, CA, Mar. 2010.

[2] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.

[3] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. of Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2004.

[4] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, Mar. 2008.

[5] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proc. of IEEE INFOCOM*, Phoenix, AZ, Apr. 2008.

[6] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I^{2}NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of IEEE INFOCOM*, Shanghai, China, Apr. 2011.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. of ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[8] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "An optimization framework for opportunistic multipath routing in wireless mesh networks," in *Proc. of IEEE INFOCOM*, Phoenix, AZ, Apr. 2008.

[9] F. Soldo, A. Markopoulou, and A. L. Toledo, "On minimizng network coding resources: An evolutionary approch," in *Proc. of the International Symposium on Network Coding (NetCod)*, Toronto, Canada, June 2010.

[10] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "Core: a coding-aware opportunistic routing mechanism for wireless mesh networks," *IEEE Trans. on Wireless Communications*, vol. 17, pp. 96–103, June 2010.

[11] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "Toward practical opportunistic routing with intra-session network coding for mesh networks," *IEEE/ACM Trans. on Networking*, vol. 18, pp. 420–433, Apr. 2010.

[12] D. Lucai, M. Stojanovic, and M. Médard, "Random linear network coding for time division duplexing: When to stop talking and start listening," in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009.

[13] Y. Mao, F. R. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," *IEEE J. on Selected Areas in Commun.*, vol. 23, no. 4, pp. 820–829, Apr. 2005.