How Network Coding System Constrains Packet Pollution Attacks in Wireless Sensor Networks

Yuanyuan Zhang\textsuperscript{1}\* and Marine Minier\textsuperscript{2}

\textsuperscript{1}Computer Science and Technology Department, East China Normal University
\textsuperscript{2}Université de Lyon, INRIA, INSA-Lyon, CITI laboratory

Abstract. Packet pollution attack is considered as the most threatening attack model against network coding based sensor networks. A widely held belief says that, in a single source multi-destination dissemination scenario, the total number of polluted packets in the network will grow with the length of the transmission path, and the decoding failure (DF) rate at the further destination nodes are relatively lower. In this work, we first obtain an opposite result by analyzing the pollution attack in multicast scenarios, and find out a convergence trend of pollution attack by network coding system, and quantify the network resiliency against the pollution attacks which happen at any place along the source-destination paths. Then, the analysis result is proved by our simulations on two most widely deployed buffer strategies, Random-In Random-Out (RIRO) and First-in First-Out (FIFO). Finally, it is proved that RIRO has an much advanced security feature than FIFO in constraining the pollution attack gradually, and almost vanished in the end.

Keywords: Pollution attack; buffer strategy; network coding security.

1 Introduction

Network coding is a very active field of both information theory and networking for information dissemination. The core idea of network coding technique is "slicing" and "mixing" packets. It consists in encoding a message into several packets and transmitting those packets in an oriented multicast way through the network to the destination. The intermediate nodes could also combine the received packets. It has been shown that network coding could reach the maximum possible information flow in a network.

Network coding is also very interesting for security. Many works have been done in demonstrating the security capacity of network coding. Two security worlds coexist and the border is delimited by the adversary capabilities. Network coding can be used to bring secrecy if the adversary eavesdropping capabilities are bounded (see [4] [13] [9]) and if the attacker is passive.

\* This work is supported by National Natural Science Foundation of China (No. 61103040).
When considering active attackers that act inside the network perimeter, network coding could also be efficient against some particular attacks. For example, selective forwarding attacks where an adversary drops/delays all or a part of the data packets he receives are completely defeated by network coding due to its intrinsic multipath nature as shown in [8, 17]. However, network coding is inefficient against pollution attacks [7] where the adversary attacks the data flow by modifying the messages produced by different sources and decoded by the destinations. In this scenario, pollution attackers inject into the network corrupted packets, which propagate in an epidemic manner. Then the intermediate nodes will push the pollution further and wider in the network by their packet re-encoding and packet forwarding actions. Many works have proposed countermeasures against pollution attacks [16, 15, 3, 6, 1].

Unlike the traditional routing, where the intermediate nodes are just receiving and forwarding packets, in network coding based systems, the intermediate nodes encode packets in the buffer and forward the encoded packets. This "combining" characteristic greatly rises the network throughput, and meanwhile accelerates and expands the dissemination of potentially corrupted packets in the network. The damage caused by pollution can be fast and huge. For example, in our simulation scenario with 400 nodes, a pollution attacker choose one of the next-hop neighbours of the source node, and corrupts one packet in its buffer. When the multicast process completes, there are at least 25% nodes having polluted packets in their buffer. Though the pollution epidemic spread very fast and wide, does it mean its performance against pollution attack is lower than traditional multicast? A novel phenomenon is presented by our analysis and simulations in the following sections.

In this work, we analysis the impact of pollution attack on network coding system, including the pollution dissemination, and polluted packets stay-in time. The decoding ailure (DF) ratio of the destination node is taken into account as the evaluation metric of the negative impact. In simulation part, we first compare different buffer strategies in network coding system. We observe that RIRO has a much better security performance in constraining pollution in the network.

This paper is organized as follows: Section 2 introduces network coding system and pollution attack. The network model, adversary model and buffer models are described in Section 3. Section 4 gives the theoretical analysis and simulation results are given in Section5. Conclusion and future works are presented in Section 6.

2 Network coding and pollution attack

2.1 Network coding

The seminal work on network coding was done by Ahlswede et al. in [2]. The main aim of network coding is to find optimal information dissemination in a network. It has been shown that network coding can also improve the network resiliency against communication failure, e.g. erasure, (see [11] chapter 1). Wireline and
wireless networks can benefit from network coding. For more details on network coding and on the problems solved by this technique, the readers can consult [11, 14, 5].

An important topic in network coding is linear codes: packets exchanged by the nodes are linear combinations of the data to be transmitted over a given finite field. Random linear network coding [12] has particularly attracted the attention. The coding process is as follows. Let us assume a network viewed as a graph with a source node and some destination nodes. Let us denote \( D = (d_1, d_2, \ldots, d_n) \) a data of \( kn \) bits viewed as a vector of \( n \) fragments \( d_i \in \mathbb{F}_{2^k}, \ i \in [1, n] \). The messages \( m_j = h_j \parallel p_j \) transmitted by the source and the relaying nodes in a scheme using random linear network coding consists in a header \( h_j \) and a payload \( p_j \):

\[
p_j = \sum_{i=1}^{n} \alpha_{i,j} d_i,
\]

where the coefficients \( \alpha_{i,j} \) are chosen randomly over \( \mathbb{F}_q \) with \( q = 2^n \) the favorite choice in the literature. The header \( h_j \) contains all the coefficients \( \alpha_{i,j} \) which describe the payload:

\[
h_j = (\alpha_{1,j}, \ldots, \alpha_{n,j}).
\]

### 2.2 Pollution attack

The “mixing packet” feature makes pollution attack a network coding targeting attack. The adversary can inject a forged packet with total random generated content, or inject a forged packet by modifying any factor in the vector \( \alpha_{i,j} \) over \( \mathbb{F}_q \), or by modifying the encoded message \( p_j \). The latter attack model follows the linear code rules, so the destination nodes are probably decode some messages. Without additional authentication or verification methods, the destination nodes cannot verify the decoded messages from the packets they have received.

### 2.3 Buffer strategy

The sensor nodes maintain a buffer of linearly independent packets they receive. Each intermediate node broadcasts encoded packets generated from the packets stored in its buffer following a pre-designed strategy.

Several common buffer strategies for multi-hop wireless sensor networks are described below.

1. **First-In-First-Out (FIFO)**: Each packet in the buffer is stored in a queue-like data structure. The first packet to be added to the queue will be the first packet to be sent, and when the buffer is full, the packets are dropped sequentially in the same order.

2. **Random-In-Random-Out (RIRO)**: This strategy is to maintain a random pool of packets. The coming packets randomly replace any packets in the buffer, and the randomly chosen packets are sent.
3. Oldest-Out: Each packet is attached with a global time stamp when it is generated. The earliest packet will be the first packet to be sent. When the buffer is full, the newly incoming packets replace the earliest packets. In this work, we only take account of one information flow. The packets are of same generation (even if there's more than one flow, the linear encoding will stay within the same information flow). One of the advantages of network coding system is that the encoded packets can come with no order, therefore, the global time stamp is not necessary in this case.

3 Network models

We may define a generic model for both the network and the adversary. The abstract network model which we are using in the work involves:

3.1 Network description

- **Intra-flow network coding.** Network coding that combines multiple packets from one source, more specifically, one data flow, is known as intra-flow coding. In our case, there is just one source node in the network, which is multicasting one message (native message) to several destinations. Using the linear coding method described in Section 2.1, the native messages are encoded into tens or hundreds of packets. Then, the intermediate nodes will encode the received packets that are generated from the same native message.

- **Gradient-based multicast.** In this work, we adopt the gradient-based routing (GBR) protocol. GBR was first proposed in [10]. It uses a natural gradient as a metric to forward the query towards source. The metric can be regarded as physical distance, hops or others. When a source node outwards a packet, it chooses the next-hop nodes which have the smallest gradient. Thus, each forwarder node will choose their next-hops in the same way. Finally, the multicast paths from source to destinations are established ideally. Unlike the common data aggregation scenario in which data comes from multiple sources to the single sink, the scenario in this paper is exactly opposite where one sink is multicasting a native message to several destination nodes. For example, sink node is upgrading a program, or sending new directives, and etc.

- **Delay-tolerant network.** In a delay-tolerant application, data can be delayed by reasonable amount of time before reaching the destination nodes. We can introduce some delay into the scenarios where 'time' is not the top priority. For instance, sensor network reprogramming or updating software, these tasks are not as urgent as real-time data monitoring, but they might have some security requirements, so we can consider to deploy some security protection by bringing in some delay along the routing path.

3.2 Adversary model

The assumption of adversary model involves:
network coding pollution attack

- **Aware of the network deployment.** We assume that the adversary is aware of the network deployment, including the geographic position of the source and destination nodes. The adversary can emerge at any place in the network, and be aware of the local topology.

- **Aware of the linear coding rules and network protocols.** The adversary should know the rule of the linear coding, i.e., the finite field, the number of the message slices, the length of each slice, and etc. Therefore, the adversary can run the linear codes and forward packets just like a normal intermediate node.

- **Able to modify or forge packet.** We assume the adversary launches active attacks by modifying or forging packets, then injects them into the network communication. To be a smarter attacker, the adversary can modify or forge the packets according to the codes of the system.

3.3 Buffer models

The choice of buffer strategy influence the performace of network coding systems. In this work, two most widely adopted strategies are compared in the simulations; they are RIRO and FIFO. In network coding systems, RIRO and FIFO strategy involve a new feature of re-encoding packets.

**RIRO for network coding**

- **Insert.** When the buffer is not full, the incoming packet is inserted into the list. When the buffer is full, randomly choose a packet and drop it from the list.

- **Generate(Re-encode).** When the buffer is not empty, randomly choose $m$ packets to calculate a new linear combination. If the buffer is empty, return an exception.

- **Send.** Each time when the node is going to send packet, it first executes `Generate` to get a new linear combined packet, then send it to the nexthop. If the buffer is empty, return an exception.

- **Drop.** Randomly delete a packet in the queue. When the buffer is empty, return an exception.

**FIFO for network coding**

- **Insert.** When the buffer is not full, the incoming packet is inserted into the end of a queue. When the buffer is full, drop the first packet in the queue.

- **Generate(Re-encode).** When the buffer is not empty, choose $m$ packets from the front of the queue, and calculate a new linear combination. If the buffer is empty, return an exception.

- **Send.** This is the same as in RIRO.

- **Drop.** Delete the first packet in the queue. When the buffer is empty, return an exception.
4 Theoretical analysis

In this Section, we sum up our theoretical analysis when considering a network as described in Section 3. Our criteria are primarily focused on the decoding failure (DF) at all destination nodes. It takes place when the destination node can decode a message which does not match the native one. This ratio is determined by measuring the ratio of the times of DF to the total times of decoding.

4.1 Polluted packet dissemination

The targeting network model is as described in Section 3. We randomly choose a normal sensor node to perform as the adversary denoted ADV, noted as $A_0$, then compromise one of its packets in the buffer. At each time slot, $A_0$ conducts to encode a packet and forwards the packet to all its downstream neighbours who are decided by the considered routing protocol. The set of the first hop neighbours is indicated as $A_1$. Following this, the $n$-hop neighbours are in the set $A_n$, and the neighbour in set $A_n$ will be indicated as $a_n$. For simplicity of analysis, we assume the network uses discrete time-step rather than continuous time. The actions of sending or receiving packets are conducted at the start of a time slot.

The RIRO and the network structure decides how the packets enter and leave the buffer. Assume the buffer size is $N_b$, and there are $N_p$ polluted packets in the buffer, and re-encoded packet is generated from $N_e$ packets, then the probability of generating a non polluted packet is,

$$Pr_h(N_p) = \frac{(N_b-N_p)}{N_b}.$$

The average number of polluted packets in $a_1$’s buffer at time slot $M$ is:

$$PMI_1(M) = M \times (1 - Pr_h(1)) \tag{1}$$

where the subscript $i$ in $PMI_i$ indicates the hop distance from $ADV$.

The number of polluted packets in $a_1$ linearly depends on $Pr_h$ by a factor $M$, because no matter how many time slots have passed, the probability of receiving a polluted packet from $ADV$ is the same. This is not the case in $A_2$. For example, at time slot 2, $PMI_2(2)$ is:

$$PMI_2(2) = d \times (1 - Pr_h(PMI_1(1)))$$
$$PMI_2(3) = d \times (1 - Pr_h(PMI_2(1))) + PMI_2(2)$$

where $d$ is the average degree in multicast tree at level $n$, and $M \geq n$.

So, we have

$$PMI_2(M) = d \times \sum_{i=1}^{M-1} (1 - Pr_h(PMI_1(i)))$$
From Eqn. (1) and above, it’s easy to obtain that,

$$PMI_n(M) = d \times \sum_{i=1}^{M-1} (1 - Pr_h(PMI_{n-1}(i))),$$  

(2)

At the same time, the downstream neighbours not only receive polluted packets, they also have probability of receiving a healthy packet $HMI$. It’s direct to have,

$$HMI_1(M) = M \times Pr_h(1)$$
$$HMI_2(2) = d \times Pr_h(PMI_1(1))$$
$$HMI_3(3) = d \times Pr_h(PMI_2(1)) + HMI_2(2)$$
$$\cdots$$
$$HMI_2(M) = d \times \sum_{i=1}^{M-1} Pr_h(PMI_1(i))$$

Then,

$$HMI_n(M) = d \times \sum_{i=1}^{M-1} Pr_h(PMI_{n-1}(i)),$$  

(3)

where $M \geq n$.

Therefore, a node $a_i$ has received $PMI_i(m)$ polluted packets and $HMI_i(m)$ healthy packets at time slot $m$.

### 4.2 Polluted packet stay-in time (ST)

We define a stay-in time (ST) to indicate a period of time for which a packet could have stayed in buffer. In an RIRO buffer, the average stay-in time (by slot) at distance $n$, starting from time slot $M$, can be calculated by,

$$ST_{RIRO} = \frac{1 - Pr_h(PMI_{n-1}(M))}{2} \times N_b.$$  

But in an FIFO buffer, the average ST is,

$$ST_{FIFO} = (1 - Pr_h(PMI_{n-1}(M))) \times N_b,$$

which is twice longer than $ST_{RIRO}$. A longer ST means that more polluted packets are received at a certain period of time, so there is a bigger probability of generating and disseminating a polluted new packets.

### 4.3 Decoding failure ratio

The decoding failure ratio $R_f$ is calculated as:

$$R_f = 1 - Pr_h(\frac{PMI_i(m) \times N_b}{PMI_i(m) + HMI_i(m)})$$  

(4)
By Eqn. (4), we have the numerical results shown in Fig. 1.

As the figure depicts, the larger $M$ has higher curve, which indicates the longer time the network coding system has run, the higher the DF ratio at the destinations, which implies severer pollution occurred in the network. This result accords what we assumed.

At distance of $n = 2$, $n = 3$ and $n = 4$, the pollution keeps growing. Then the interesting result appears that, different from the traditional multicast pollution, the network coding helps depress the pollution along the path. Starting from $n = 4$, the peak value of the curve, the pollution falls down rapidly with the distance. Further we can observe that, the curve turning is smoother with smaller $M$, which means the shorter time the network has run, the further destinations has better resiliency against pollution attack. It implies that, in order to avoid receiving more polluted packets, the destination node should decode once it has received enough packets.

5 Simulation results

In this section, we describe the packet pollution results in a network simulation environment.
**Network coding configuration** In this work, we adopt random linear network coding to implement. We choose factors from the finite field $F_{2^q}$, $q = 7$. The native message is split into 10 slices, which means the destination node must collect at least 10 packets to recover the native message.

**Sensor node configuration** Each sensor node has a small memory space for buffering received packets which stores at most 40 packets. The buffer strategy works as RIRO, i.e., the received packet replace a packet at a random place in the buffer. When sending, only one packet is sent at each time slot. The sensor node randomly chooses 5 packets from its buffer, calculate a new linear combination, and disseminate into the network.

**Network configuration** We adopt two dimension static sensor network. All the sensor nodes are uniformly randomly distributed onto the field. Each sensor node has a radio range equal to 20m. As in real applications, the sensor nodes are placed by a fixed density, so we use the average density as one sensor node in every 50 square meters. Our measurements were taken from simulations on a wireless sensor network simulator. The simulations were based on a static non-directed multicast tree, which is built by underlying routing protocol GBR on a non-directed random graph. The simulations are run with a large topology from 500 nodes to 800 nodes.

In the interest of fairness, the DF ratio at each distance are evaluated at different time slot. The further destination nodes are measured later than closer ones to the attacker node. For example, the destination nodes at $n = 8$ are evaluated at the $M = 14$ time slot, the nodes at $n = 9$ will be evaluated at the next time slot $M = 15$. This can make sure that the nodes at $n = 8$ have experienced $M - (n - 1) = 7$ times of receiving from their upstream neighbours, and nodes at $n = 9$ have received also 7 times (consider that it is a discrete time simulation).

Fig. 3 and Fig. 2 reveals the simulation results under buffer strategy RIRO and FIFO, respectively.

Fig. 2 exhibits the packet pollution damage in network coding system with FIFO buffer. As analyzed in Section 4.2, the time the polluted packets stay in FIFO buffer is twice as in RIRO buffer, so FIFO causes larger damage to network coding system, as depicted in the figure. This result is just as what the common opinion says: under FIFO buffer strategy, the further the pollution goes along the paths, the higher DF ratio at the destinations.

The interesting results are shown in Fig. 3. It is well supported by our theoretical analysis: the multicast pollution is constrained by network coding system under RIRO strategy. Similar to Fig. 1, the peak value emerges at $n = 4$, then the DF ratio drops afterwards. Moreover, in a large network, the source-destination path is necessarily longer than in smaller sized network, so it costs more time.
slots to multicast a native message to the destinations. According to our analysis, the DF ratio curve in larger network should be above the one in smaller network, and that’s what we observe in Fig. 3.

6 Conclusion

Pollution attack is a network coding targeting attack, the damage caused by which can be fast and huge. The pollution caused by different buffer strategies can show different forms. This work evaluates two buffer strategies, RIRO and FIFO. In FIFO, the further the pollution goes along the paths, the higher decoding failure ratio at the destinations, which implies severer pollution in the network. The pollution is widely spread in the network if FIFO strategy is deployed.

However, the pollution epidemic has a trend of convergence after it reaches a peak value if RIRO strategy is used. We prove this by theoretical analysis on a K-ary tree (in this work, we choose $K = 5$) and verify the analysis by simulation results. It means the pollution will gradually shrink and approaching to dying out in the end.

The deployment of FIFO or RIRO is not the vital key item in network configurations in common wireless networks, because the performance is not sig-
network coding pollution attack

Fig. 3. Buffer strategy RIRO with different network size. X-axis indicates the distance between the attacker and the destinations, and Y-axis indicates the decoding failure ratio of destinations.

significantly different from each other. However, in network coding system, RIRO shows a far advanced security feature than FIFO.

The future work is to carry out more comparison between different buffer strategies, deduce the convergence of the pollution attack, and establish other theoretical models, e.g., based on epidemic models.

References