Network coding versus traditional routing in adversarial wireless networks

Donghai Zhu\textsuperscript{a}, Xinyu Yang\textsuperscript{a,}\textsuperscript{*}, Wei Yu\textsuperscript{b}, Xinwen Fu\textsuperscript{c}

\textsuperscript{a} Dept. of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
\textsuperscript{b} Towson University, Towson, MD 21252, USA
\textsuperscript{c} University of Massachusetts Lowell, Lowell, MA 01854, USA

\textbf{A R T I C L E  I N F O}

Article history:
Received 2 November 2013
Received in revised form 15 February 2014
Accepted 2 April 2014
Available online 16 April 2014

Keywords:
Network coding
Pollution attack
Wireless network
Performance evaluation

\textbf{A B S T R A C T}

When network coding is used in wireless mesh networks (WMNs), the epidemic effect of pollution attacks can reduce network throughput dramatically. Nevertheless, little attention has been directed toward the performance gain of network coding versus traditional routing in adversarial wireless mesh networks. To address this critical issue, in this paper, we formally model and analyze the impact of pollution attacks on traditional routing and network coding in both unicast and multicast scenarios. With the combination of both numerical and simulation studies, we evaluate the performance of traditional routing and network coding in adversarial wireless networks. Our data is consistent with the theoretical findings. Our results show that network coding is not absolutely better than traditional routing and its performance gain largely depends on various factors. Most importantly, given a network, the threshold of these factors can be derived from numerical solutions given by our developed closed-form formulae. Thus, we can determine whether network coding should be used in the network. Our results contribute to the foundation, providing guidelines for designing and applying network coding into hostile wireless networks.

\textcopyright 2014 Elsevier B.V. All rights reserved.

1. Introduction

In this paper, we tend to investigate the performance gain of network coding versus traditional routing in adversarial wireless mesh networks (WMNs). Network coding [1] is a mechanism that integrates coding and routing schemes, enabling relay nodes to combine information content in packets before forwarding them. Existing research efforts such as [2,3] demonstrated that network coding could improve network throughput and be extended to numerous applications [4,5].

Although network coding can improve network performance, network coding is vulnerable to pollution attacks. Through such attacks, malicious nodes can alter or forge some corrupted packets and inject them into the network. Through the epidemic propagation, corrupted packets can pollute the whole network quickly and network throughput can be impacted significantly [6]. Note that in traditional routing, because downstream nodes only receive packets from their direct last hop, such epidemic effect will not occur.

To deal with pollution attacks in network coding, a large number of defense schemes have been proposed in the past [7–22]. In addition, a number of research efforts have been conducted to secure routing through authentication in traditional routing [23]. Although the performance gain of network coding in non-adversarial networks has been
well established, it is uncertain whether the conclusion still holds in adversarial wireless networks. Hence, before deploying network coding, the following urgent and fundamental issue must be addressed: with a defensive scheme in place to detect and mitigate pollution attacks to some extent (in the condition where some nodes cannot be trusted and may launch a pollution attack), can network coding still achieve a higher performance than traditional routing?

To answer this question, in this paper, we consider and generalize key factors of defensive schemes and inherent networks in traditional routing and network coding rather than focusing on individual defense schemes. We use the Expected Transmission Count (ETX) [24] for successful packet delivery to evaluate the effectiveness of network coding and traditional routing. We consider the abilities of the defender, the adversary, and the network. The ability of the network includes the compromised ratio of nodes in the network to measure its hostile degree, the detection probability of defense schemes deployed in well-behaved nodes, the attack ability of malicious nodes, and the generation size of network coding.

To understand the insightful relationship between metrics and parameters, we then model and analyze pollution attacks on traditional routing and network coding in both unicast and multicast scenarios, according to their fundamental principles (i.e., hop-by-hop in routing and then adding network coding) and the effect of pollution attacks. We derive closed-form formulae to analyze the performance of traditional routing and network coding in adversarial wireless networks. Through comprehensive and realistic modeling, we also obtain the quantitative threshold of parameters. We develop algorithms to compute the ETX of successful packet delivery in pollution attacks in both unicast and multicast scenarios. Through the combination of both numerical and simulation studies, we evaluate the performance of traditional routing and network coding in an adversarial Roofnet [25] network. Our data is consistent with our theoretical findings.

To the best of our knowledge, our work is the first to study and compare the performance of network coding and traditional routing in adversarial wireless mesh networks. The main findings from this research are listed as follows: (i) Network coding is not absolutely better than traditional routing in adversarial networks; (ii) If defense schemes in network coding can achieve a high enough detection probability, network coding can consistently perform better than traditional routing; (iii) Network coding is more suitable than traditional routing for the network in which the adversary’s attack ability is limited to a certain level; and (iv) Network coding is more suitable than traditional routing for the network in which the compromised ratio of nodes is smaller than a certain level. Most importantly, given a network, the threshold of the aforementioned parameters can be obtained through numerical solutions given by our derived closed-form formulae. Then, we can determine whether network coding should be used in the network based on the derived thresholds.

For example, as shown in our results based on Roofnet [25], if the compromised nodes can corrupt all the packets forwarded and the detection probability of the authentication scheme is 0.99, the coding gain varies from 2.01 to 4.53 as the node compromised ratio varies from 0 to 0.5. If the detection probability is only 0.8, the coding gain varies from 2.06 to 0.58. This means that network coding is not always better and it only holds in some conditions. To achieve better performance in this scenario, the detection probability of the well-behaved nodes should be at least 0.847. This means if the detection probability in both network coding and traditional routing is higher than 0.847, network coding performs better. Our work is fundamental and among the first to formally model and compare the effect of pollution attacks in network coding and traditional routing. Our results contribute to the foundation, providing guidelines for designing and applying network coding into hostile wireless networks.

The rest of this paper is organized as follows. In Section 2, we introduce the system and threat models, along with key factors and assumptions. In Sections 3 and 4, we carry out the modeling and analysis of pollution attacks on network coding and traditional routing in unicast and multicast scenarios, respectively. In Section 5, we present our numerical and simulation results. In Section 6, we briefly introduce related work. Finally, in Section 7, we conclude the paper.

2. Preliminary

In this section, we first present the system and threat models and then review key factors and assumptions.

2.1. System model

In this paper, we consider the reliable unicast and multicast communications, which are supported by traditional routing and random linear network coding [2]. In traditional routing, source $S$ transmits packets hop-by-hop through a predetermined single path in the unicast scenario or through a multicast tree to a group of receivers in the multicast scenario. We also assume that the ETX metric [24] is used to conduct a routing decision. Note that the ETX metric can effectively measure the expected count of transmissions required to successfully deliver a packet over a lossy wireless link. It can also be easily obtained by pingin other neighbor nodes periodically and estimating the delivery probability on each link. More importantly, the ETX of a route is the sum of the link metrics, which can be easily computed in both traditional routing and network coding. However, other metrics do not have all these benefits. The advantages of ETX compared with hop-count is shown in [24] and there are also several extended metrics (e.g., METX [26]), aiming to improve the throughput of multicast.

Based on the principle of network coding, a source generates a message stream and splits it into generations. Assume that each generation consists of $m$ messages. After prefixing every message with a unit vector of dimension $m$, the source randomly selects the coefficients and sends the linear combination of messages to nodes downstream. Hence, the first $m$ symbols of a coded message, denoted as the global coding vector, will be further used for decoding. In a similar way, forwarders overhear the
transmissions of all upstream nodes in promiscuous mode. Those forwarders then randomly select their coefficients, linearly combine received messages, and then forward the coded messages. The receiver(s) can recover original messages by using Gaussian eliminations. Through this mechanism, after receiving enough linearly independent messages, the original message can be recovered with a high probability. The detailed description and implementation of random linear network coding can be found in [2,3]. Note that in this paper we do not consider the inter-flow network coding presented in [27].

To ensure the reliability of traditional routing, link-by-link ARQ [28] can be used in both unicast and multicast scenarios. In this way, a node retransmits the packet received from its direct last hop (i.e., parent) to its direct next hop (i.e., children) until the packet is correctly received by all of direct next hops (i.e., children) in the unicast (respectively multicast) scenario. Note that in network coding, we use a similar technique adopted by MORE [3] to ensure reliability.

2.2. Threat model

In pollution attacks, adversaries can modify and inject corrupted data packets into the network that enables network coding to disrupt the decoding process [16]. For pollution attacks against traditional routing, we mean that data integrity is under attack [6]. The term “pollution attack” mainly belongs to the network where network coding is enabled and has no pollution effect on traditional routing. More specifically, aiming to corrupt data transmission and degrade network performance, the adversary compromises some nodes in the network first. Then the adversary attempts to inject corrupted packets into the traffic flow or modify packets forwarded by nodes that he/she controls. It is worth noting that if a malicious node is located at the bottleneck of a network with a higher transmission probability (a high frequency of data transmission), it can modify and inject more corrupted packets into the network, leading to a more significant performance degradation. Note that we do not consider the attack that seeks to disrupt the routing process and other attacks such as traffic analysis and eavesdropping, etc.

We now briefly formalize pollution attacks in network coding. Each message transmitted in the network can be designated as a row vector $y_i$ of $m+n+l$ symbols. In this row vector, the first $m$ symbols are designated as the global coding vector $y_i$, the next $n$ are data symbols. The last $l$ are authenticators $t_i$ to assure authentication of the data origin.

Let $X$ be the matrix where the $i$th row is $x_i$, and let $T$ be the matrix where the $i$th row is $t_i$. Then, a correct packet $y_i$ can be derived by $[g_y, g_x, X, g_y, T]$. In addition, a corrupted packet can be denoted as $\tilde{y}_i = y_i + v_i = [g_y + v_m, g_x + v_n, \tilde{X} + v_l]$ where $v_i = [v_m, v_n, v_l]$ is the tampering vector introduced by the adversary and we can derive $\tilde{v}_m, \tilde{X} \neq v_m, \tilde{v}_n, T \neq v_l$.

Then, after receiving messages $y_i$, with a proper key $k$ and verification function $H$, well-behaved intermediate nodes can compute hashes, signatures, or Message Authentication Codes (MACs). The authenticators are then compared with messages to validate the integrity by checking the following condition: $H(y_i) = t_i$. If the above condition returns true, $y_i$ is a correct packet and added into the packet buffer pool, waiting to participate in the combination process; otherwise, $y_i$ is a corrupted packet and is dropped.

We now briefly discuss the impact of polluted packets [8]. Let $V_m$ be the matrix where the $i$th row is $v_m$, and let $V_t$ be the matrix where the $i$th row is $v_t$. The set of packets received by intermediate nodes and destination nodes can be represented as the matrix $Y = Y + V_t$ where the $i$th row is $y_i$. However, if the tampering matrix $V_t$ is not a zero matrix, the defense scheme fails to detect even one of corrupted packets. Then the destination will incorrectly decode packets using Gaussian eliminations by multiplying the matrix $[G + V_m]^{-1}$ with $[Y + V_t]$. This gives $X = [G + V_m]^{-1} \cdot [G \cdot X + V_t]$. As we can see, even if one corrupted packet is undetected and accepted, all correct packets received before that failure will be polluted. More importantly, subsequent output packets will be corrupted as well.

As we can see, pollution attacks could reduce the network performance when network coding is used. Note that although the performance gain of network coding in non-adversarial networks has been well established, it is uncertain whether the conclusion still holds in adversarial wireless mesh networks. This will be our focus in this paper.

2.3. Key factors and assumptions

We now describe some key factors and assumptions. We assume that an end-route filtering scheme can be deployed in the network to ensure data integrity by filtering corrupted packets. With such filtering schemes, after receiving a packet, the well-behaved nodes will verify the packet and determine whether or not it is corrupted. If it is determined to be corrupted, it will be dropped; otherwise, the packet will be inserted into the forwarding buffer queue, waiting for transmission.

Existing defense schemes against pollution attacks in both traditional routing and network coding scenarios can filter corrupted packets to some extent, measured by the detection probability. For symmetric-key schemes (e.g., [13,14,18]), the detection probability is limited because to ensure security and efficiency, these schemes cannot pre-distribute keys to all intermediate nodes. For example, in [13], if half of the codewords in one packet are used to generate a MAC and 10 MACs are attached to a packet, the polluted message can traverse around three hops on average, meaning the detection probability is not high enough.

In addition, for public-key schemes (e.g., [9–12]), although the detection probability of those schemes is much higher, the computation overhead is too high to validate all packets. As the detection probability is the key attribute of filtering schemes, we define the detection probability from 0 to 1 to quantify the performance of defense schemes. Note that we focus on the effectiveness rather than the overhead of defense schemes as network coding can be applied in a variety of networks. In some scenarios, the overhead may not be the most critical issue in
comparison with the performance gain from network coding.

Similar to the assumptions made in literature related to network coding, we assume that the source and destination nodes will not be compromised. We also assume the destination node can detect all corrupted packets because it can share all the symmetric-keys with the source node (for symmetric-key schemes) or its computation capacity is strong enough (for public-key schemes). When a source node fails to deliver a batch of packets to the destination, it will choose another route to transmit the lost data packets in traditional routing whereas it will keep retransmitting in network coding. We assume that the adversary may choose to compromise intermediate nodes according to the network topology so as to launch efficient attacks. We also assume that the attack ability of compromised nodes in the network is determined by the adversary, according to the network topology, in order to launch efficient attacks. In addition, compromised nodes may cooperate with each other to launch more efficient attacks. In some conditions, the adversary will not alter all forwarded messages in order to avoid being identified as the attack can be easily detected by defense schemes (e.g., [29]) when the corrupted ratio is larger than a certain level.

3. Pollution attacks on unicast scenario

In this section, we first introduce an example to motivate our study. We then describe the model that we use for investigating the impact of pollution attacks on unicast traffic, along with key metrics and system parameters. Based on these, we then model and analyze pollution attacks on unicast traffic and derive closed-form formulae to evaluate the performance of traditional unicast routing and network coding in adversarial wireless networks.

3.1. Example 1

Consider the scenario in Fig. 1 where the value under the arrow is the link quality. In traditional routing, the path is computed before data transmission. In this example, traffic is transmitted along the path “src → R → dest” which has the highest delivery probability. Meanwhile, network coding leads to a larger savings in bandwidth and achieves a higher throughput by exploiting the shared nature of wireless communication media. For example, for every packet from src to dest, two transmissions are needed in traditional routing on average whereas only 1.67 transmissions are needed when network coding is used. Hence, R only needs to transmit 67% of the linear combination of received packets. This is sufficient for dest to recover all packets as dest receives the remaining 33% through the “src → dest” link. The details of how to derive the expected number of transmissions can be found in [3].

What if the network contains malicious nodes? Consider the next hop R is a compromised node, which will randomly manipulate 50% of received packets before forwarding, and assume that dest can detect and drop all corrupted packets. In the network using traditional routing, to assure reliability, 3.5 transmissions are needed on average. That is, besides the two transmissions in the route: src → R → dest. Because 50% of packets forwarded by R are corrupted and need to be retransmitted, an additional three transmissions are necessary for delivering these corrupted packets through a suboptimal route: src → dest. In the same network, when network coding is used, only 2.5 \((1 + 0.67)/(1 \times 0.33 + 0.67 \times 0.5)\) transmissions are needed. That is, in this adversarial network, if src and R only makes one transmission and 0.67 transmissions for one packet on average, respectively, dest can only receive \((1 \times 0.33 + 0.67 \times 0.5)\) correct packets. To decode all packets, extra transmissions are required. Note that in this scenario, the performance of network coding is still higher than that of traditional routing.

Hence, the problem is whether these conclusions still hold in large scale wireless networks. With defensive schemes in place to detect and mitigate pollution attacks, will network coding consistently achieve a higher network performance than traditional routing?

3.2. Metrics and parameters

In terms of the performance of traditional routing and network coding for the unicast scenario, we consider the ETX [24] per successful packet delivery. A successful packet delivery means that if a packet is lost or altered by the adversary during the data transmission, the source or other well-behaved intermediate nodes are responsible for retransmitting until the packet is correctly received by the destination. Recall that the ETX has been used frequently to measure important network characteristics [24,26] (e.g., the consumed power and delay during the transmission). Obviously, the bigger the ETX, the higher the power consumption and the longer the delay, leading to lower performance.

In our modeling, we consider the following system parameters: (i) \(c\): the compromised ratio of nodes in the network to measure the hostile level of the network, (ii) \(p\): the detection probability of defense schemes deployed at well-behaved nodes, (iii) \(q\): the ratio of the number of modified packets sent by a compromised node to the number of all received correct packets, (iv) \(s\): the ratio of the number of injected packets sent by a compromised node to the number of all received packets, and (v) \(b\): the generation size of network coding. Note that the attack ability of malicious nodes consists of \(q\) and \(s\). Without loss of generality, we consider \(c, p, q, s\) as values in \([0,1]\).

3.3. Traditional routing

The main notations used in this paper are listed in Table 1.
Let \( r_i^r \) and \( r_i^c \) be the expected number of correct packets and corrupted packets received by node \( i \), respectively. Let \( X_{ij} \) be the expected transmission count of node \( i \) to successfully transmit a packet to downstream node \( j \), determined by the link quality. We have \( X_{ij} = 1/(1 - \epsilon_{ij}) \) where \( \epsilon_{ij} \) is denoted as the probability of losing link \( e_{ij} \). Let \( t_i \) be the total ETX of forwarder \( i \) routing packets from \( src \) to \( dest \).

The compromised probability of node \( i \) be \( c_i \). The compromised node \( i \) will modify \( q_i \) percentage of correct packets received, and inject corrupted packets \( s_i \) times. In this way, if malicious node \( i \) receives \( N \) correct packets, it will send \( N \cdot (1 + s_i) \) packets, consisting of \( N \cdot (1 - q_i) \) correct packets and \( N \cdot (s_i + q_i) \) corrupted packets.

In traditional routing, a node \( i + 1 \) (as the next hop of node \( i \)) only receives packets from its last hop node \( i \). If node \( i \) is a well-behaved node (the probability of being a well-behaved node is \( 1 - c_i \)), it will check the received packets, forward correct packets and undetected corrupt packets, and drop detected corrupt packets. If node \( i \) is a malicious node (the probability of being a malicious node is \( c_i \)), it does not conduct verification. On the contrary, it will modify \( q_i \) percent of correct packets and inject corrupted packets \( s_i \) times as assumed. Note that while conducting pollution attacks, the malicious node randomly alters the received packets according to a given probability as assumed. It also need not tell which packets are correct or not because a corrupted packet will not be altered to a correct packet. We obtain the following recursive formulae for deriving the expected number of correct packets \( r_i^r \) in Eq. (1a) and the corrupted packets \( r_i^c \) in Eq. (1b) received by node \( i \).

\[
\begin{align*}
r_i^{r+1} &= (1 - c_i) \cdot r_i^r + c_i \cdot r_i^c \cdot (1 - q_i), \quad \text{(1a)} \\
r_i^{c+1} &= (1 - c_i) \cdot r_i^c \cdot (1 - p) + c_i \cdot (q_i \cdot r_i^r + r_i^c) \\
&\quad + s_i \cdot (r_i^r + r_i^c). \quad \text{(1b)}
\end{align*}
\]

Then, the expected total number of transmissions of node \( i \) can be derived from \( r_i^r \) and \( r_i^c \). Then, we have:

\[
t_i = ((1 - c_i) \cdot (r_i^r + r_i^c \cdot (1 - p)) + c_i \cdot (1 + s_i) \cdot (r_i^r + r_i^c)) \cdot X_{i,i+1}. \quad \text{(2)}
\]

As assumed, we have the initial condition listed below. We denote \( src \) as the first node, the next hop of \( src \) as the second, etc.

\[
r_1^r = r_2^r = 1, \quad r_1^c = r_2^c = 0, \quad c_1 = 0. \quad \text{(3)}
\]

Note that the result of Eq. (2) corresponds to the shortest path from \( src \) to \( dest \). As packets may be altered during the transmission, the source is responsible for retransmitting dropped packets to ensure reliability. However, if the adversary is in the shortest path, it is important to select another route for these dropped packets. To compute the total ETX of a successful packet delivery, let \( R_j \) be the \( j \)th shortest path from \( src \) to \( dest \), \( R \) be the set of \( R_j \), and \( rd_j \) be the number of correct packets, which is received by \( dest \) through \( R_j \). Then, the total expected number of successful packet deliveries \( T \) is given by:

\[
T = \sum_{R_j \in R} \left( 1 - \sum_{i=1}^{j-1} rd_i \right) \cdot T_j.
\]

Here, \( T_j = \sum_{k \in R_j} t_k \) is the total ETX in \( R_j \).

### 3.4 Network coding

Different from traditional hop-by-hop routing schemes, network coding fully uses the broadcast nature of wireless communication. Hence, the broadcast transmission of an upstream node may be overheard by more than one downstream node. Here, downstream nodes are referred to as nodes that are closer (measured by the ETX along the shortest path) to the destination. We sort nodes by their distance to \( dest \) in ascending order. That is, \( src \) is the \( num \)th node, \( dest \) is the first, and \( num \) is the number of nodes.

We then classify nodes in network coding into the following three categories:

- **Malicious nodes**: These nodes are compromised by the adversary and their behavior is totally controlled. They aim to degrade network performance as much as possible while trying not to be detected.
- **Well-behaved nodes**: These are non-malicious nodes in the network. They validate received packets, filter corrupted ones, and forward correct packets to the destination.
- **Polluted nodes**: These nodes belong to well-behaved ones and are polluted by some corrupted packets because the detection probability is not high enough to detect all received corrupted packets. Their

The following table contains the notation used in the text:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_i^r )</td>
<td>Expected number of correct packets received by node ( i )</td>
</tr>
<tr>
<td>( r_i^c )</td>
<td>Expected number of corrupted packets received by node ( i )</td>
</tr>
<tr>
<td>( t_i )</td>
<td>Total ETX that forwarder ( i ) routes one packet from source to destination</td>
</tr>
<tr>
<td>( c_i )</td>
<td>Probability of node ( i ) being a malicious node</td>
</tr>
<tr>
<td>( q_i )</td>
<td>Ratio of modified packets sent by a compromised node ( i ) to all correct packets received</td>
</tr>
<tr>
<td>( b_i )</td>
<td>Generation size of network coding</td>
</tr>
<tr>
<td>( X_{ij} )</td>
<td>Expected transmission count of node ( i ) to conduct successful packet transmission to ( j )</td>
</tr>
<tr>
<td>( Pr[\cdot] )</td>
<td>Probability that the event ( \cdot ) occurs</td>
</tr>
</tbody>
</table>

\( p \): Detection probability of authentication schemes

\( T \): Total ETX of a successful packet delivery

\( Q \): Ratio of corrupted packets sent by polluted node \( i \) to all packets in network coding

\( C_i \): Expected probability that node \( i \) sends corrupted packets in network coding

\( s_i \): Ratio of injected packets sent by a compromised node \( i \) to all packets received

\( g \): Multicast group size
behavior is similar to the well-behaved node, but these nodes do not realize that they are damaging the network.

The assumption we make and notations we use in this subsection are the same as the ones used in Section 3.3. As defined, the polluted nodes will also send out corrupted packets. However, the behavior of polluted nodes differs from that of malicious nodes. They will not inject extra corrupted packets and their ratio of corrupted packets is also different from that of malicious nodes. For a polluted node $i$, we assume $Q_i$ is the expected ratio of corrupted packets transmitted by polluted node $i$ to all packets. Assume that $N_p$ is the number of corrupted packets detected before the node fails to detect corrupted packets, meaning the node is polluted. Consider that corrupted packets and correct packets arrive independently and the transmission opportunity occurs uniformly. Then, we have:

$$Pr[N_p = n] = (1 - p)^n \cdot p^n, \quad 0 \leq n \leq \lceil r_i^- \rceil, \quad r_i^- \geq 0.$$  

Here, \(\lceil \cdot \rceil\) is the floor operation. If node $i$ receives $r_i^-$ corrupted packets, when $N_p = n$, the ratio of output corrupted packets is $\left(1 - \frac{n}{r_i^-}\right)$. Hence, $Q_i$ can be derived by

$$Q_i = \sum_{n=0}^{\lceil r_i^- \rceil - 1} \left(Pr[N_p = n] \cdot \left(1 - \frac{n}{r_i^-}\right)\right),$$  

$$= 1 - \frac{1}{r_i^-} \cdot \frac{p \cdot \left(1 - p^{\lceil r_i^- \rceil}\right)}{1 - p}, \quad i \leq num - 2.$$  

To understand $Q_i$ better, we show a simple example here. Consider a node that receives nine corrupted packets in total and fails to detect the 6th corrupted packet. Before the 6th corrupted packet is accepted, the node is a well-behaved node, meaning the packets it sends out are all correct. Once polluted, the packets transmitted later are corrupted because the node takes the accepted corrupted packet as the input of a linear combination. As a result, we can say that the ratio of corrupted packets sent by this node is one third (i.e., $1 - 6/9$).

In Eq. (6), $r_i^-$ is the total number of corrupted packets that node $i$ receives from all malicious upstream and polluted nodes. If an upstream node $j$ is malicious, its ratio of corrupted packets to all the transmitted packets is $1/3$, otherwise, the ratio is $Q_j$. The probability of node $i$ being a well-behaved node is $1 - C_i$ (note that $C_i$ is the probability that node $i$ is a malicious node or polluted node) and $t_{ji} = t_{ji} + (1 - e_{ji})$ is the total number of packets received by node $i$, which were sent from node $j$. It is easy to know $r_{num} = r_{num - 1} = 0$ because the src is assumed to be not compromised. Hence, $r_i^-$ can be derived as follows:

$$r_i^- = \sum_{j=1}^{num-1} \left(C_i \cdot \frac{S_j + q_j}{1 + S_j} + (C_j - C_i) \cdot q_j\right) \cdot t_{ji}, \quad i \leq num - 2.$$  

If a node $i$ is initially a non-malicious node and detects all corrupted packets that it has received, it is a well-behaved node. Because this probability can be computed by $(1 - c_i) \cdot p_{i/2}^-$, we have $1 - C_i = (1 - c_i) \cdot p_{i/2}^-$. The probability that node $i$ will send corrupted packets can be derived by:

$$C_i = c_i + (1 - c_i) \cdot (1 - p_{i/2}^-), \quad i \leq num - 1.$$  

Let $r_i^+$ be total number of correct packets that node $i$ receives from all upstream nodes. It comes from three parts: (i) the total packets sent from well-behaved nodes, (ii) the $1/3p$ of packets sent from malicious nodes, and (iii) the $1 - Q_i$ of packets sent from polluted nodes. Let $r_{num} = b$ where $b$ is the generation size of network coding. Then, when $i \leq num - 1$, we have:

$$r_i^+ = \sum_{j=i+1}^{num} \left(1 - C_j + c_j \cdot \frac{1 - q_j}{1 + S_j} + (C_j - c_j) \cdot (1 - q_j)\right) \cdot t_{ji}.$$  

Consider a well-behaved node $i$, for each correct packet $i$ received from $j$, $i$ should forward it only if no well-behaved node $k$ that is closer to dest than $i$, denoted as $k < i$, overhears the packet. This occurs with a probability of $\prod_{k<i}(1 - e_{jk} + (1 - e_{jk}) \cdot C_k)$. Thus, when $i \leq num - 1$, the average number of packets that $i$ must forward, denoted as $f_i$, can be derived by:

$$f_i = \sum_{j=i+1}^{num} \left(1 - C_j + c_j \cdot \frac{1 - q_j}{1 + S_j} + (C_j - c_j) \cdot (1 - q_j)\right) \cdot t_{ji}.$$  

As a well-behaved node, $i$ should transmit each packet until a well-behaved node with a lower ETX receives it. It is obvious that this event occurs with a probability of $(1 - \prod_{k<i}(1 - e_{jk} + (1 - e_{jk}) \cdot C_k))$. Note that the behavior of polluted nodes is the same as that of well-behaved nodes. On average, $t_i$ can be derived by:

$$t_i = (1 - c_i) \cdot f_i \left/ \left(1 - \prod_{k=1}^{j-1}(1 - e_{jk} + (1 - e_{jk}) \cdot C_k)\right)\right. + c_i \cdot X = (1 + c_i \cdot S_i) \cdot f_i \left/ \left(1 - \prod_{k=1}^{j-1}(1 - e_{jk} + (1 - e_{jk}) \cdot C_k)\right)\right..$$  

Here, $X$ is the number of transmitted packets from malicious in Eq. (11) comes from the assumption that the ratio of injected packets from the malicious nodes is $s$. When computing $t_i$, it is impossible to know the exact value of $C_i(k < i)$ as this value is determined by the received corrupted packets. Thus, in computing $t_i$, the approximate value of $C_i$ is replaced by $c_i$. As we can see, Eq. (10) is a summation of the number of correct packets from upstream nodes. If node $i$ is located at the bottleneck of the network, $f_i$ will be large. Subsequently, $t_i$, as the expected transmission count of node $i$, will also increase. If node $i$ is malicious with a higher transmission probability (a high frequency of data transmission), it can modify and inject more corrupted packets into the network, leading to a more significant performance degradation.
Algorithm 1. Compute $t_i$ – the ETX that node $i$ needs to deliver one packet from src to dest for network coding (unicast)

1: Sort nodes in network by their distance to dest in ascending order;
2: for $i = 1 : num$ do
3:  $C_i \leftarrow c$; $Q_i \leftarrow 0$; $r_i^2 \leftarrow 0$; $r_i^1 \leftarrow 0$;
4: end for
5: $C_{num} \leftarrow 0$; $Q_{num} \leftarrow 0$; $r_{num}^2 \leftarrow b$;
6: for $i \leftarrow num : 2$ do
7:  if $i \neq num$ then
8:  Calculate $r_i^2$ from Eq. (7)
9:  $Q_i \leftarrow 0$; $C_i \leftarrow c$;
10:  if $|r_i^2| = 0$ then
11:  $Q_i \leftarrow 0$; $C_i \leftarrow c$;
12:  else
13:  Calculate $Q_i$ from Eq. (6)
14:  Calculate $C_i$ from Eq. (8)
15: end if
16: end if
17: Calculate $t_i$ from Eq. (11)
18: if $f_j$ in Eq. (11) have been computed already and $C_k$ is approximately replaced by $c$.
19: $M_i \leftarrow 1$
20: for $j = 2 : num - 1$ do
21:  Calculate $r_j^1$ from Eq. (9)
22:  $Q_j$, $C_j$ and $t_j$ ($j > i$) in Eq. (7) have been computed already
23:  $M_i = M_i \times (c_{ij-1} + (1 - c_{ij-1}) \ast C_{j-1})$
24:  $f_j = f_j + t_{ij} \ast M_i$
25: end for
26: end for

Using a bottom-up technique, we can compute all the recursive equations listed above. The detailed algorithm is shown in Algorithm 1. Then, the total ETX of a successful packet delivery can be computed by:

$$T = \left( \sum_{i=2}^{num} t_i \right) / b. \quad (12)$$

By solving Eqs. (4) and (12), we can obtain the threshold of parameters to determine the condition where network coding performs better. However, it is hard to obtain the analytical solution because Eqs. (4) and (12) are both complex recursive formulae. We have tried to find the analytical solution, but have come to the conclusion that it is too complicated to obtain. Hence, we obtain the numerical solution shown in Section 5.

4. Pollution attacks on multicast scenario

4.1. Example II

Similar to Fig. 1, we consider a multicast scenario shown in Fig. 2. In this example, src sends packets to two receivers dest_1 and dest_2 through a predetermined multicast tree based on traditional routing, and two transmissions are necessary to deliver a packet to receivers whereas 1.67 transmissions are needed in network coding approach. For the details of computing the expected number of transmissions in network coding, please refer to [3].

However, if $R$ randomly modify 50% of packets received as assumed in multicast scenario, the result is 5.33 and 2.5, respectively. For the traditional multicast routing in this example, we consider that the link quality from src to dest_1 and dest_2 are 0.5. Then, the expected transmission count of src and $R$ can be computed as $\sum_{i=0}^{\infty} \left( 1 - (1 - 0.5^2) \right)^2$. When network coding is used in this case, the computation and the result are the same as the ones in Example I. The detailed computation will be discussed in next subsection. As we can see here, network coding achieves a higher throughput than traditional routing, while the coding gain is also larger than that of Example I.

4.2. Traditional routing

To transmit information from one source to multiple destinations, traditional multicast routing first sets up a multicast tree $\mathcal{M}$. Then, the information will be transmitted hop-by-hop over the tree. In this way, the network performance can be significantly improved rather than sending the information to individual destinations using unicast.

In terms of the performance of traditional routing for multicast scenario, we consider the multicast expected transmission count (METX) [26] per successful packet delivery. For the consistency, we will still use ETX as the notation.

As the downstream children nodes only receive packets from their parent node, the recursive formulae for multicast will be the same as Eqs. (1a) and (1b). However, not all multicast group members are the leaves of tree, the initial condition of recursive formulae is listed below:

$$\begin{align*}
    c_i &= \begin{cases} 
    0, & \text{if } i \in \mathbb{R} \\
    c_i, & \text{others}
    \end{cases}, & p_i &= \begin{cases} 
    1, & \text{if } i \in \mathbb{R} \\
    p_i, & \text{others}
    \end{cases}, & i \geq 2 \\
    r_i^1 &= 1, r_i^2 = 0
\end{align*} \quad (13)$$

Here, $\mathbb{R}$ is the set of multicast group members.
With the initial condition, we have
\[ t_i = \left((1 + c_i \cdot s) \cdot (r_i^+ + r_i^-) - p \cdot (1 - c_i) \cdot r_i^- \right) \cdot MX_i. \]  
(14)

If link-by-link ARQ (i.e., intermediate nodes are responsible for packet retransmission rather than the source) is used, \( MX_i \) can be derived by
\[ MX_i = \sum_{n=0}^{\infty} \left( 1 - \prod_{k \in \text{child}(i)} \left( 1 - e_{ik}^n \right) \right). \]  
(15)

In terms of the retransmission of corrupted packets, there are the following three ways: (i) The first is to set up a new multicast group to include receivers which did not receive the packet. It is infeasible as the unacceptable latency will be involved. (ii) The second is to unicast original packets that are corrupted during the transmission to destinations. It will lead to large bandwidth consumption when the compromised ratio is high. (iii) The third is that some group members retransmit packets to their descendant group members. This is the simplest and most effective strategy [30]. Hence, the total ETX of a successful packet delivery can be derived by,
\[ T = \sum_{k \in \text{dest}} \left( t_i + \sum_{n=0}^{\infty} \left( 1 - \prod_{k \in \text{desc}(i)} \left( 1 - (1 - r_k^-)^n \right) \right) \right). \]  
(16)

Here, \( k \in \text{desc}(i) \) means node \( k \) is a direct descendant group member of node \( i \) and the second summation is to compute the expected transmission number that each node makes to deliver one packet to all of its direct descendant group members.

To make the packet delivery model on multicast scenario more understandable, we would like to give a simple example here. In Fig. 3, the source needs to transmit packets to four multicast group members. First, a multicast tree is built and \( t_i \), the total ETX that forwarder \( i \) routes one packet, is computed for each node \( i \) using Eqs. (14) and (15). Then, to retransmit the corrupted packets modified by the malicious nodes, extra transmissions are needed. As stated, group members are responsible for retransmitting packets to their descendant group members. In this example, \( \text{dest}_1 \) rather than the source shall retransmit the packets to \( \text{dest}_4 \). The expected transmission number that \( \text{dest}_1 \) makes to deliver one packet to \( \text{dest}_3 \) and \( \text{dest}_4 \) can be computed using the second summation in Eq. (16).

4.2. Network coding

Network coding in multicast scenario is a natural extension of unicast scenario. Equations derived from Section 3.4 can be generally applied with a little modification. The basic workflow is listed below. First, we compute all unicast transmitting nodes \( i \) and their \( t_i \) from \( \text{src} \) to all \( \text{dest} \) belong to the multicast group. Second, multicast transmitting nodes set is the union of unicast transmitting nodes, and their \( t_i \) is the maximum value among that of corresponding unicast transmitting nodes. The detailed algorithm is similar to Algorithm 1 and can be found in our technical report [31] because of the space limitation.

5. Performance evaluation

In our evaluation scenario, we use real-world link quality measurements from Roofnet [25], which is an experimental 802.11b/g mesh network developed by MIT. The topology and link qualities are shown in Fig. 4 and the raw bandwidth is 2 Mbps. The simulated network consists of 35 nodes.

For unicast scenarios, we compute the ETX metric of every source–destination pair from the given parameters \( c, p, q, s \) under 1000 attack scenarios and then show the average over all flows on the corresponding figure. Note that in our evaluation, we define the parameters \( c, q, s \) as the average of values associated with these 35 nodes. That is to say, under different attack scenarios (i.e., different \( c_i, q_i, s_i \) ) chosen by the adversary, \( c = \frac{1}{35} \sum c_i \) and so on. While for multicast scenarios, we randomly choose 1000 source–destination pairs and then show the mean value over these 1000 results on the corresponding figure.

5.1. Numerical results

As we stated, as Eqs. (4) and (12) are too complex to derive analytical solutions to compare the performance, we show numerical results below.
Fig. 5 shows the total ETX of successful packet delivery versus the ratio of compromised nodes and the detection probability of well-behaved nodes with $b = 16$, $q = 1$, $s = 0$ in unicast scenario. As we can see, when the detection probability is below 0.847, with the increase of the ratio of compromised nodes, the total ETX of network coding grows much faster than that of traditional routing. The smaller the detection probability is, the faster the ETX of network coding grows. The result is consistent with our theoretical analysis: if the detection probability is not high enough to detect all corrupted packets, not only nodes themselves will be polluted, but also the risk of downstream nodes to be polluted will increase considerably. If the detection probability is high enough (i.e., the threshold is 0.847 in this scenario), network coding always performs better than traditional routing. We observe that when $c = 0$ (i.e., network is highly secured), the coding gain of unicast is 2.01, which matches the result in [3]. With a high enough detection probability and a relatively secure network (the relationship of these two parameters is illustrated in Fig. 6), the performance of network coding is always better than that of traditional routing. For example, when $p = 0.99$, the coding gain varies from 2.01 to 4.53 given $c$ from 0 to 0.5; when $c = 0.2$, the coding gain varies from 1.12 to 1.99 given $p$ from 0.7 to 0.99.

Fig. 6 illustrates the relationship between the compromised ratio and detection probability when the total ETX of network coding and traditional routing are equal for different $q$ and $s$ when $b = 16$ in unicast scenario. The point on the curve represents the detection probability corresponding to the compromised ratio when ETX is the same. The upper area represents that network coding is better. We observe that if the compromised ratio is large (i.e., the network environment is highly hostile), the detection probability should be high enough to count attacks. For example, when $s = 0$, $q = 1$ and the compromised ratio is 0.37, the detection scheme should be designed with a filtering probability of at least 80% to make sure that network coding performs better than traditional routing. We can also observe that the detection probability increases as the attack ability increases, whereas the compromise ratio decreases as the attack ability increases, which is consistent with our expectation. In addition, from the red, green and black curves, we can observe that this relationship is insensitive to the corrupted packets injection. In addition, we would like to note that although the result is based on the Roofnet, our data provides guidelines for designing and applying network coding into wireless networks with different security risk. That is, after obtaining the topology and evaluation of adversarial wireless networks ($c, q$ and $s$) and selecting a defense scheme with the detection probability of $p$, we can use this figure to guide whether network coding should be used in the network or not. We can draw the point $(c, p)$ in the figure and check whether the point is above the curve or not.

Fig. 7 shows the relationship between the total ETX and the generation size of network coding when $c = 0.3$, $q = 1$, $s = 0$, $p = 0.8$ in unicast scenario. The total ETX of network coding increases significantly with the generation size, because the polluted risk, determined by the number of received corrupted packets, also increases when the generation size grows. In this regard, the generation size should be selected as a small value. However, the bandwidth overhead of most defensive schemes will be unacceptable if generation size is too small. Meanwhile,
according to MORE [3], the throughput of network coding is insensitive to the generation size. Hence, the value of generation size should also be considered carefully when the defense scheme and network coding are designed in adversarial wireless networks.

Fig. 8 shows the total ETX of a successful packet delivery versus the ratio of compromised nodes and the detection probability of well-behaved nodes when \( b = 16 \), \( q = 1 \), \( s = 0 \) in multicast scenario. From these figures, we can see that the general growing trend of network coding is similar because network coding supports multicast naturally. When the multicast group size grows from 3 to 7, the total ETX of traditional multicast routing increases from 10 to 22. In comparison with results in unicast scenario, the variation range of routing is smaller because the algorithm to set a multicast tree prefers to select multicast group members as forwarders. In addition, the coding gain in secure networks increases from 2.63 to 3.42 with the increase of the multicast group size. Hence, we conclude that network coding is more suitable for multicast than unicast. Similar to unicast, with a high detection probability in a relatively secured network, the performance of network coding can be always better than that of traditional routing. For example, when \( p = 0.99 \) and group size \( g = 5 \), the coding gain varies from 3.27 to 1.86 given \( c \) from 0 to 0.5; when \( c = 0.2 \), the coding gain varies from 1.06 to 2.78 given \( p \) from 0.625 to 0.99. The detailed relationship of these two parameters in the scenario with the same total ETX is illustrated in Fig. 9.

Similar to Figs. 6 and 9 shows the relationship between the compromised ratio and detection probability when the total ETX of network coding and traditional routing are equal for different \( g \) when \( b = 16 \), \( q = 1 \), \( s = 0 \) in multicast scenario. From this figure, we can determine whether we should use network coding in adversarial multicast networks or not.

5.2. Simulation results

In this subsection, using simulations we compare the performance of traditional routing and network coding in the adversarial scenario in terms of the total ETX as discussed in Section 3. Our simulation data validates our modeling and theoretical analysis in Sections 3 and 4.
In our simulation, we first randomly choose source and destination(s) and then randomly initialize network nodes into well-behaved nodes and malicious nodes according to the predetermined compromised ratio $c_i$. We generate a random number that is uniformly distributed in the interval $[0, 1]$ for each intermediate node. If the random number is smaller than the predetermined $c_i$, it is set as a malicious node. Malicious nodes and well-behaved nodes will act as the one discussed in Section 2.1. Every action is controlled by a random number as well. Each simulation is repeated 1000 times and the results illustrated below show the average over 1000 times. All simulations in this paper are run in Matlab 7.0.

Fig. 10 shows the simulation results of the total ETX of successful packet delivery versus the ratio of compromised nodes and the detection probability of well-behaved nodes when $b = 16$, $q = 1$, $s = 0$ in unicast scenario. We can see that, the difference between analytical results and simulation results is relatively small and their trend is same. Because of the randomness of simulation, Fig. 10b shows a relatively heterogeneous result. After polynomial fitting, we obtain Fig. 10c that is consistent with the analytical results in Fig. 6. Fig. 11 shows the simulation result of the total ETX of successful packet delivery versus the ratio of compromised nodes and the detection probability of well-behaved nodes when $b = 16$, $q = 1$, $s = 0$, $g = 5$ in multicast scenario. We can observe that, the difference between analytical results and simulation results of network coding is relatively small whereas their trend goes same. After polynomial fitting, the simulation result of compromised ratio is a bit smaller than that of expected analytical result when the detection probability is lower than 0.75 because we filtered some extreme cases in which there are few packets transmitted to destinations.

To summarize, our main findings are as follows:

- Network coding is not absolutely better than traditional routing in adversarial wireless networks.
- If defense schemes in network coding can achieve a high enough detection probability, network coding can always perform better than traditional routing.
- Network coding is more suitable than traditional routing for the network in which the adversary’s attack ability is limited to certain level.
- Network coding is more suitable than traditional routing for the network in which compromised ratio of the nodes is smaller than a threshold.

We would like to note that all thresholds mentioned above can be derived from closed-form formulae derived in Sections 3 and 4.

6. Related works

To deal with pollution attacks in the network using network coding, a number of authentication schemes have been proposed [7–22]. While there are a number of defense schemes to perform data origin authentication in the network using traditional routing, we focus only on authentication schemes in the network using network coding. The details about data authentication in traditional routing can be found in [23].

The techniques to defend against pollution attacks in the network using network coding can be largely categorized into two groups: (i) information-theoretic based schemes and (ii) cryptography-based schemes. For
For cryptography schemes, based on the type of keys, existing authentication schemes can generally be categorized into public-key cryptographic approaches (e.g., [9–12]), symmetric-key cryptographic approaches (e.g., [13,14,17,19]), hybrid-key cryptographic approaches [18], and others [15,16]. As examples of public-key cryptographic approaches, the Cooperative Security Scheme (CSS) [9] extended on-the-fly verification. Yu et al. [11] developed an RSA-based Efficient Signature-Based Scheme (ESBS), which is based on modular exponentiation. Public-key approaches are extremely time-consuming due to their high computational overhead. To apply them to resource-constrained wireless networks, techniques such as probabilistic verification [9] were introduced.

As examples of symmetric-key cryptographic approaches, the Efficient Scheme for XOR network coding (ESXOR) [13] is the first scheme to use XOR network coding. RIPPLE [17] is a time-synchronized based scheme. The symmetric-key cryptographic approach needs a key pre-distribution scheme unless a time-synchronized based scheme is applied, but time synchronization is not easy to implement in wireless networks. After proposing a polynomial-based authentication scheme, [22] also provided a multicast goodput analysis to assess the impact of pollution attacks on multicast throughput and showed the amount of goodput gained. Nonetheless, the analysis of [22] is elementary to some extent and does not provide a general comparison with the impact of pollution attacks in traditional routing.

Different from existing research efforts of addressing pollution attacks against network coding, [7–22] our research focuses on answering the fundamental question: with a defensive scheme in place to detect and mitigate pollution attacks to some extent, will network coding still achieve a higher performance than traditional routing and in what conditions?

7. Conclusion

In this paper, we addressed the issue of investigating the performance gain of network coding versus traditional routing in adversarial wireless networks. To be specific, we formally modeled and analyzed the impact of pollution attacks on traditional routing and network coding in both unicast and multicast scenarios. Through the combination of both numerical and simulation studies, we evaluated the performance of traditional routing and network coding in adversarial wireless networks. Our evaluation data shows that network coding is not absolutely better than traditional routing and its performance gain largely depends on a number of factors. These factors include the compromised ratio of the nodes, adversary’s attack ability, defender’s detection probability, and others. Our results contribute to the foundation by providing guidelines for designing and applying network coding to hostile wireless networks.

Acknowledgement

The work was supported in part by the following funding agencies in China: the New Century Excellent Talents in Universities of China under grant NCET-09-0649, the Ph.D. Program Foundation of Ministry of Education of China under grant 20100201110027, the Program for Industrial Development of Shaanxi under grant 2011K06-40.
References


Donghai Zhu received the B.S. degree in the Department of Computer science and technology from Xi’an Jiaotong University, in 2010. He is currently a Ph.D. candidate in the Department of Computer Science and Technology from Xi’an Jiaotong University. His current research interests include wireless ad hoc/mesh networks, network coding, and network security.

Xinyu Yang received the Diploma in Computer Science and Technology from the Xi’an Jiaotong University of China in 2001 and the Bachelor, Master, and Ph.D. degrees from Xi’an Jiaotong University in 1995, 1997, and 2001. He has held positions with the Department of Computer Science and Technology at the Xi’an Jiaotong University, where he is a professor teaching in the Department of Computer Science and Technology. His main technical interests lie in the areas of network security, wireless communication and mobile ad hoc networks.

Wei Yu received the B.S. degree in electrical engineering from Nanking University of Technology, Nanjing, China, in 1992, the M.S. degree in electrical engineering from Tongji University, Shanghai, China in 1995, and the Ph.D. degree in computer engineering from Texas A&M University, College Station, in 2008. He is currently an Assistant Professor with the Department of Computer and Information Science, Towson University, Towson, MD. Before that, he was with Chico Systems Inc. for almost nine years. His research interests include cyberspace security, computer networks, and distributed systems.

Xinwen Fu is an associate professor and associate chair in the Department of Computer Science, University of Massachusetts Lowell. He received B.S. (1995) and M.S. (1998) in Electrical Engineering from Xi’an Jiaotong University, China and University of Science and Technology of China respectively. He obtained Ph.D. (2005) in Computer Engineering from Texas A&M University. Dr. Fu’s current research interests are in network security and privacy, network forensics, computer forensics, information assurance, system reliability and networking QoS. Dr. Fu has been publishing papers in conferences such as IEEE S&P, ACM CCS, ACM MobiHoc, IEEE INFOCOM and IEEE ECDCS, journals such as ACM/IEEE Transactions on Networking (TON), IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE Transactions on Computers (TC), IEEE Transactions on Mobile Computing (TMC) and IEEE Transactions on Vehicular Technology (TVT), book and book chapters. His research is supported by University of Massachusetts Lowell and NSF.