When Watchdog Meets Coding

Guanfeng Liang, Rachit Agarwal and Nitin Vaidya
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA
Email: {gliang2, agarwa16, nhv}@illinois.edu

***Abstract*—We consider the problem of misbehavior detection in wireless networks. A commonly adopted approach is to exploit the broadcast nature of the wireless medium, where nodes monitor their downstream neighbors locally using overheard messages. We call such nodes the *Watchdog*s. We propose a lightweight misbehavior detection scheme which integrates the idea of watchdogs and error detection coding. We show that even if the watchdog can only observe a fraction of packets, by choosing the error detection code properly, an attacker can be detected with high probability while achieving throughput arbitrarily close to optimal. Such properties reduce the incentive for the attacker to attack.**

We then consider the problem of locating the misbehaving node and propose a simple protocol, which locates the misbehaving node with high probability. The protocol requires exactly two watchdogs per unreliable relay node.

I. Introduction

Information dissemination in wireless ad hoc and sensor networks is usually done using multihop paths, where data packets are relayed in several wireless hops from the source to the destination. This multihop nature makes wireless networks subject to tampering attack: a compromised/misbehaving node can easily ruin data communications by dropping or corrupting packets it is supposed to forward.

There are two main approaches to mitigating such misbehavior. An end-to-end misbehavior detecting scheme codes the data packets into an error detection code, allowing the destination to detect the authenticity of the received packets. Such end-to-end schemes suffer from two shortcomings; first, the throughput of the network is reduced even when there are no misbehaving nodes in the network, and, second, even if the misbehaving node is detected, it has partially been successful in disrupting the network. Moreover, with such end-to-end schemes, the source and the destination have no knowledge about the location of the misbehaving node(s).

A technique to overcome these shortcomings of end-to-end detection is the *Watchdog* mechanism [1], a monitoring method used for ad hoc and sensor networks which is the basis of many misbehavior detection algorithms and trust or reputation systems. The basic idea of the watchdog mechanism is that of nodes (called watchdogs) policing their downstream neighbors locally using overheard messages in order to detect misbehavior. If a watchdog detects that a packet is not forwarded within a certain period or is forwarded but altered by its neighbor, it deems the neighbor as misbehaving.

This research was supported in part by Army Research Office grant W-911-NF-0710287.

Such a monitoring mechanism is interesting because it allows the intermediate nodes in the network to actively participate in the misbehavior detection, and detect the misbehaving node in the network. However, the main challenge for most watchdog mechanisms is the unreliable wireless environment. Due to possible reasons such as channel fading, collision with other transmissions, or interference, even when the source node and the misbehaving node are both within the communication range, the watchdog may not be able to overhear every transmission. Hence, a misbehaving node may render the whole transmission useless by corrupting just a single packet.

In order to efficiently mitigate the misbehavior of the malicious nodes, a watchdog mechanism must achieve the following two goals. First, malicious behavior in the network should be detected with high probability despite adverse channel conditions and interference, and, second, the throughput with the detection mechanism should be comparable to the throughput without detection in the absence of any attack. Intuitively, these two goals are conflicting; on one hand, more redundancy is required to improve the probability of detection and on the other hand, higher throughput requires redundancy to be reduced.

In this paper, we show that both goals can be achieved simultaneously by introducing error detection coding to the watchdog mechanism. The main contributions of this paper are as follows:

- We propose a computationally simple scheme that integrates source error detection coding and the watchdog mechanism. In particular, we show that by choosing the error detection code properly, a misbehaving node can be detected with high probability while the throughput approaches optimal, even in the case when the watchdog can only overhear a fraction of the packets and the attacker is omniscient, *i.e.*, the attacker knows what code is being used and no secret is shared only between the source and destination.
- We also propose a simple protocol that identifies the misbehaving node using exactly two watchdog nodes per unreliable relay node. We show that our protocol can be interpreted as a maximum likelihood decision making scheme. We discuss that with multiple rounds of detection, the probability of correctly locating the malicious node can be made arbitrarily close to one.
- We illustrate the effectiveness of our schemes with some small example topologies, and show that these results naturally generalize to multihop networks.

The remainder of the paper is organized as follows. We discuss related work in Section II. Section III-A and Section III-B illustrate the ideas using simple networks, for the case of single flow and for the case of two flows respectively. We also design error detection codes that allow achieving a throughput arbitrarily close to optimal and present simulation results to support our claims. In Section IV, we present a protocol for locating the misbehaving node for the single flow and two flow network cases. Section V shows that the results of single and two flow network case can not be improved for multihop routing networks, and showing that the scheme generalizes to multihop networks. We discuss some issues related to implementation of the scheme and improving the performance in Section VI and close the paper with some future directions in Section VII.

II. Related Work

The foundation of dynamic trust systems in ad hoc and sensor networks is the node behavior monitoring mechanism, most frequent discussion being on the watchdog mechanism [1]. The main idea of watchdog was promiscuous monitoring, as discussed in Section I. Once a node is deemed to be misbehaving, the source would choose a new route free of misbehaving node with the aid of a "pathrater". However, as discussed in Section I, such watchdog mechanisms do not perform well in the presence of adverse channel conditions and interference allowing the misbehaving node to corrupt a single packet while being undetected with high probability.

A variant of watchdog mechanism is proposed in [2] where next-hop's behavior is measured with the local evaluation record, defined as a 2-tuple: packet ratio and byte ratio, forwarded by the next-hop neighbor. Local evaluation records are broadcast to all neighbors. The trust level of a node is the combination of its local observation and the broadcasted information. Trust level is inserted to the RREQ (Route REQuest). Route is selected in the similar way to AODV (Ad hoc On Demand Distance Vector) [3]. Although many ad hoc trust or reputation systems such as [4], [5], [6] adopt different trust level calculation mechanism, the basic processes are similar to [2], including monitoring, broadcasting local observation, combing the direct and indirect information into the final trust level.

The security issues in systems that employ network coding have recently drawn much attention. Due to the *mixing* nature of network coding, such systems are subject to a severe security threat. Several techniques in order to make the system secure have been proposed; in particular, using carefully designed digital signatures [7], [8], [9], [10] or hash functions [11], [12], which allow intermediate nodes to verify the integrity of combined packets. Such cryptographic solutions largely rely on either the private key being kept secret from the adversary or the difficulty to reverse the hash function. Several non-cryptographic solutions have also been proposed [13], [14] for systems that employ network coding. However, the overhead associated with these schemes could be prohibitive for several applications in wireless networks (for a study of transmission overhead in these schemes, see [15]). In the context of wireless networks, [16] proposes two practical schemes to address pollution attacks against network coding in mesh networks without requiring complex cryptographic functions and incur little overhead.

In the context of monitoring, two similar schemes have recently been proposed independently, [17] and our earlier work [18]. The two-hop network investigated in [17] similar to the single flow example in section IV of [18] and section III-A of this paper. Both schemes introduce redundancy at the source of data to improve the detection at the monitoring node, in the form of a polynomial hash function and MDS (maximum distance separable) code, respectively. While [17] considers general network codes, we focus on a particular network code – forward and compare. Both works show that as the amount of redundancy increases, the probability that the malicious node being undetected approaches zero. However, this paper also studies the tradeoff between security and throughput when one node is monitoring more than one flow, and also proposes a scheme to identify the malicious node when the watchdog node can also be malicious. To the best of our knowledge, [18] was the first work that identified the insufficiency of linear network codes in achieving secure unicast capacity.

III. Detecting Misbehavior

In this paper, we focus on multihop wireless networks in which data packets are transmitted from source to destination through multiple relay nodes. We consider networks in which relay nodes do not perform any coding and the data packets are forwarded as they are received at the relay nodes. In such a network, a node W can be assigned as a watchdog for a relay node R if W can overhear both incoming and outgoing transmissions to/from R. More specifically, a node W that can overhear the data packets being transmitted by R and by R's upstream neighbor can compare the two copies of the (overheard) packet, and report an attack to the source or destination if there is a mismatch.

We are particularly interested in detecting tampering attacks, when the data transmitted by the source is corrupted and/or dropped by a misbehaving node. We focus on a single node adversary model, *i.e.*, the adversary can compromise at most one node in the network (except for the source(s) and destination(s)). We also consider the cases when the adversary does not corrupt any packet but does accuse other node(s) in the network as misbehaving node. In particular, if a watchdog is misbehaving, the only way to attack is to accuse a relay node of forwarding corrupted data even though the relay node is well-behaving. Note that misbehavior detection is trivial in this case since the source/destination always knows some node is misbehaving upon receiving the accusation report from the watchdog. Hence, in the case of misbehavior detection, it is more interesting to look at the case when a relay node misbehaves. However, in the case of locating the misbehaving node, we consider both the cases: when the relay node may forward the corrupted data or when the watchdog node may accuse a well-behaving relay node of forwarding corrupted data.

We consider an omniscient adversary model, where the adversary is computationally unbounded and has complete knowledge of the misbehavior detection scheme being employed in the network. We assume that there is no secret between the source and the destination hidden from the adversary. Under such an adversary model, the most pressing difficulty in designing an efficient monitoring mechanism is the unreliable nature of wireless broadcast channel caused by channel fading and/or interference. Since a watchdog node may only be able to overhear a fraction of the transmissions to/from the node it is monitoring, an adversary may be able to avoid being detected by the watchdog with high probability by keeping the fraction of packets it tampers lower than a certain threshold δ_w. In order to overcome this drawback of watchdog mechanisms, we integrate source error detection coding with watchdogs: the source node encodes the data packets with some error detecting code and sends the coded packets through the multihop network with watchdogs. By applying error detecting codes, the destination can detect an attack during the decoding process with high probability if the fraction of packets tampered by the adversary is lower than a certain threshold δ_c. Intuitively, if $\delta_w < \delta_c$, even an *omniscient* adversary will be detected with high probability no matter how many packets it corrupts.

In the following two subsections, we design efficient error detection codes which when combined with the watchdog mechanism allow misbehavior detection with high probability while achieving a throughput arbitrarily close to optimal.

A. Single Flow Case

To illustrate the idea, let's look at the example of a single flow network as in Fig. 1. There are 4 nodes in the network: the source node S, destination node D, attacker R, and the watchdog node W. The thick (directed) lines denote a link from the tail node to the head node, a dashed line denotes the overhearing and a blue line denotes a secure asymptotically negligible rate channel between the two nodes. We assume that all links (except for the blue one) have the same transmission rate of 1 packet per unit time. We also assume an optimal centralized schedule is enforced and the watchdog W knows what to compare. Moreover, we assume all transmissions along the path S-R-D are reliable while W can only overhear both transmission of a packet with probability q, which we call *observe probability* [1].

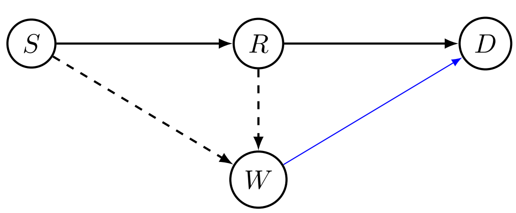


Fig. 1. A single flow network.

[1]Transmissions along the data path is usually protected by channel coding or/and retransmission mechanisms, while the watchdog can only overhear packets opportunistically.

The source node S encodes every k data packets into a block of n coded packets with an (n, k) Maximum Distance Separable (MDS) code. We assume the packet size is large enough so that an MDS code always exists for the desired value of n and k. With an (n, k) MDS code, an attack will always be detected at the decoder as long as no more than $n-k$ packets are altered. As a result, R has to alter at least $n-k+1$ packets in a block in order to avoid being detected by the decoder. However, the more packets R tampers, the more likely it will be caught by W. Hence, it is of R's interest to just attack the minimum number of packets per block: $n-k+1$. In this case, it is easy to show that the probability of R not being caught is

$$P_{miss}(n, k, q) = (1-q)^{n-k+1}. \qquad (1)$$

If we construct a (n, k) encoder such that

$$k = n + 1 - \frac{f(n, q)}{q}$$

From Eq. 1 we have

$$\begin{aligned} P_{miss}(n, k, q) &\leq e^{-q(n-k+1)} \\ &= e^{-f(n,q)} \end{aligned}$$

We can then choose the function $f(n, q)$ appropriately so that we can make P_{miss} arbitrarily small while the coding rate k/n approaches arbitrarily close to optimal. For example, by making $f(n, q) = \beta \ln n$ for any positive constant β, we have

$$\begin{aligned} P_{miss}(n, k, q) &\leq e^{-\beta \ln n} \\ &= n^{-\beta} \to 0 \quad \text{as} \quad n \to \infty \end{aligned}$$

Using the above error detection code, the coding rate becomes

$$\begin{aligned} \frac{k}{n} &= \frac{n + 1 - \frac{\beta \ln n}{q}}{n} \\ &= 1 + \frac{1}{n} - \frac{\beta}{q}\frac{\ln n}{n} \to 1 \quad \text{as} \quad n \to \infty \end{aligned}$$

So we can reduce the incentive for R to attack by making n large and choosing β appropriately.

Since the delay to verify a block equals the time it takes to transmit n packets in the block, tradeoff between probability of miss-detection and n is of interest. Fig. 2 and Fig. 3 show the probability of miss-detection with the observe probability q and with the number of packets n respectively. We can see that by integrating a watchdog and error detection coding, we can reduce the incentive for the attacker to attack by allowing longer delay.

Notice that by making n large, the coding/decoding complexity increases. In the case complexity is a concern, the source can scramble coded packets of multiple (n, k) encoded blocks and transmit these packets in a random order. By doing so, the attacker will have to corrupt more packets in order to destroy a particular block, which makes it easier to be detected by the watchdog.

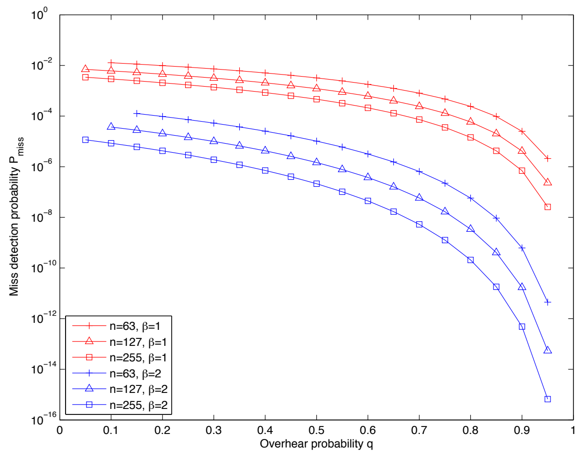


Fig. 2. Miss detection probability versus observe probability q for the single flow case of Fig. 1.

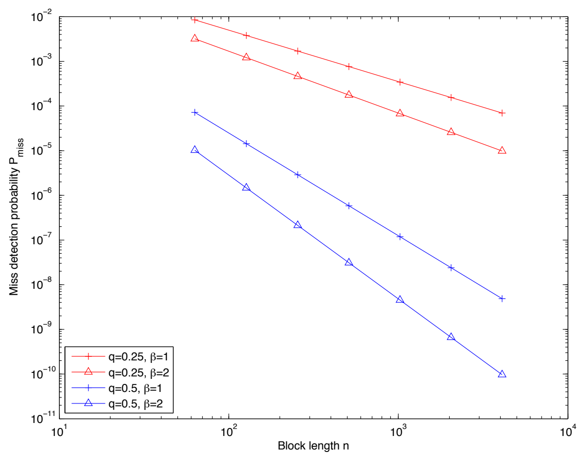


Fig. 3. Miss detection probability versus code block length n with $k = n+1-\frac{\beta \ln n}{q}$ for the single flow case of Fig. 1

B. Two Flows Case

In Section III-A, we have illustrated the effectiveness of integrating source error detection coding with the watchdog mechanism for a single flow example with a centralized optimal scheduler. In this subsection, we will study the trade-off between throughput and security in a more practical setting: there are multiple data flows in the network and a distributed random access MAC protocol is used. In the following example, we show that the proposed scheme achieves a high level of security while maintaining a reasonably good throughput.

Consider the network shown in Fig. 4 with two flows: S_1-R_1-D_1 and S_2-R_2-D_2. Suppose that the flows are far enough away from each other so there is no inter-flow interference, but the watchdog W is between the two flows and can overhear transmissions on all the four links. So even though a transmission is successful along its path, it may collide with packets from the other flow received at W. We assume a slotted aloha access protocol with access probability α is used. To simplify the analysis, we further assume that a node will access the

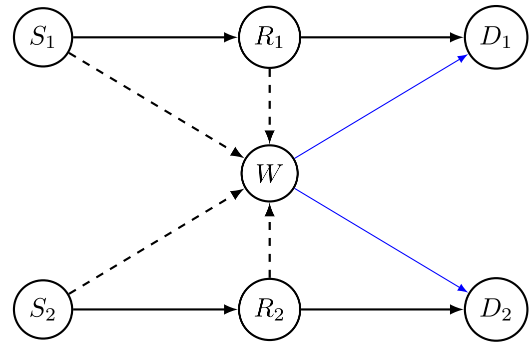


Fig. 4. A two flow network. The thick (directed) lines denote a reliable connection from the tail node to the head node, a dashed line denotes the overhearing and a blue line denotes a secure asymptotically negligible rate channel between the two nodes.

channel by transmitting dummy packets when it has no data packet to send. Under these assumptions, we can compute the throughput of each flow as:

$$T = \alpha(1-\alpha),$$

and the probability with which the watchdog W can successfully compare a particular packet:

$$q = (1-\alpha)^5. \tag{2}$$

The exponent in Eq. 2 is 5 because given that the transmission from S_1 to R_1 is successful, W can overhear it if neither S_2 nor R_2 transmit which occurs with probability $(1-\alpha)^2$. To compare this packet, W should overhear the transmission from R_1 to D_1 too, which happens with probability $(1-\alpha)^3$ for S_1, S_2 and R_2 to remain silent.

Similar to the single-flow example, we can make P_{miss} arbitrarily small by choosing

$$k = n + 1 - \frac{\beta \ln n}{(1-\alpha)^5}. \tag{3}$$

And the effective throughput is

$$\begin{aligned} T_E &= T \times \frac{k}{n} \\ &= \alpha(1-\alpha)(1+\frac{1}{n}) - \frac{\alpha\beta \ln n}{(1-\alpha)^4 n}. \end{aligned}$$

In Fig. 5 and Fig. 6, we plot the miss-detection probability and effective throughput when the error detection code is chosen according to Eq. 3. We only plot the result for $\alpha \le 0.5$ because further increasing α will only reduce the throughput. Notice in Fig. 5 the probability of miss-detection increases as α increases and converges to roughly $n^{-\beta}$. Since the higher the α is, the fewer packets the watchdog can observe, the source has to sacrifice coding rate in order to maintain a certain probability of missing an attack as α increases.

Notice in Fig. 6 that as α increases, the effective throughput increases up to a certain level then drops to zero as α gets larger. We can also see the optimal access probability changes according to the value of n and β: the larger n is, the higher α should be; the larger β is, the smaller α should be.

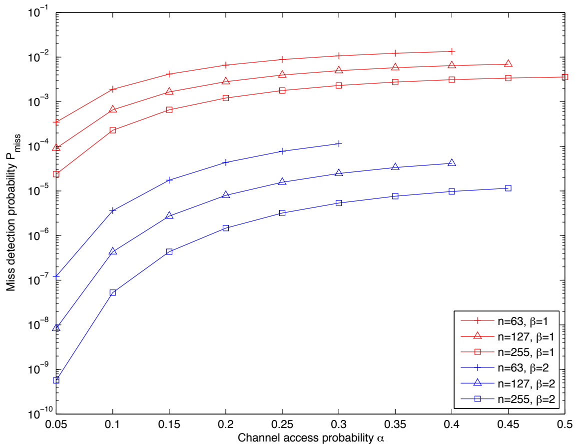


Fig. 5. Miss detection probability versus observe probability for the two flow case of Fig. 4.

For instance, if the source does not perform any coding (which is not plotted here), it is well known that the optimal α is 0.5 and the per-flow throughput is 0.25 packet per slot. In the case $n = 255$ and $\beta = 1$, the optimal α is about 0.35 and the throughput is about 0.19 packets per slot. Although the throughput is higher without source coding, it comes with the cost of not being able to provide any security guarantee. On the contrary, our scheme guarantees by upper bounding P_{miss} by $n^{-\beta}$. Our scheme provides a method to optimize the balance among throughput, delay, and security.

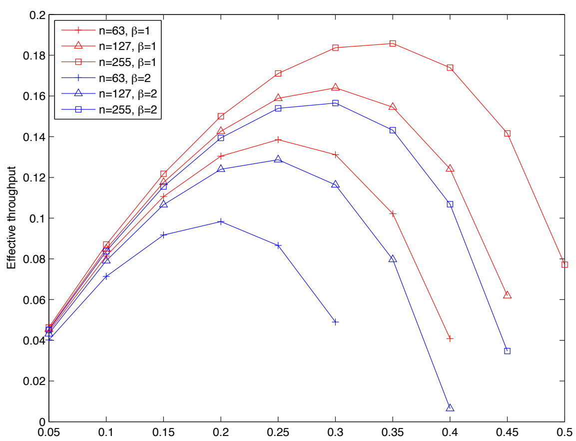


Fig. 6. Miss detection probability with $k = n + 1 - \frac{\beta \ln n}{q}$ for the two flow case of Fig. 4

IV. Identifying the Misbehaving Node

While misbehavior detection is essential in some applications, it is also important to identify the node that is misbehaving in order to avoid that node in future transmissions. The scheme discussed in the previous section cannot determine which node is misbehaving. In particular, if the watchdog node informs the source/destination that the relay node is misbehaving, the scheme in the previous section does not allow to detect whether the watchdog node is accusing the (well-behaving) relay node or whether the relay node is indeed sending corrupted data.

In this section, we present a simple protocol that identifies the misbehaving node with just a single extra watchdog, including the cases when the watchdog node is misbehaving. An interesting aspect of the proposed protocol is that given the protocol, an attacker has no incentive to attack the watchdog. In particular, we show that if the watchdog node misbehaves, our protocol locates the misbehaving node deterministically (with probability equal to one). However, if the relay node is misbehaving, our scheme is guaranteed to locate the misbehaving node with a probability that quickly approaches to unity with increasing number of packets transmitted.

The protocol in the following subsection can be viewed as several nodes making a decision on the correctness of the message transmitted by the relay node. The protocol can be visualized as the maximum likelihood decision scheme, and as we show in the following subsection, gives an optimal decision based on the decisions of the watchdogs.

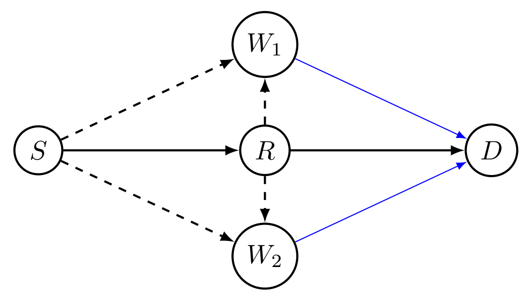


Fig. 7. Single Flow network of Fig. 1 with an extra watchdog.

A. The Protocol

Consider a relay node R that is being observed by two watchdogs W_1 and W_2 and relays the information from a source node S to destination node D (see Fig. 7). Assume that the source node employs an (n, k)-MDS code, and each packet contains a unique *generation number* that identifies the generation to which a particular packet belongs to. Each watchdog in the network decides whether or not the relay node is misbehaving based on all the overheard packets that belong to the current generation . If R is misbehaving (one of the n packets transmitted by R does not match the corresponding packet transmitted by S), it transmits a *decision bit* 1 to the judge node [2], else it transmits a decision bit 0 to the judge node. We assume that if the watchdog is misbehaving, it may transmit a 0 or a 1 for any particular relay node (same watchdog may transmit different decisions for different relay nodes). Denote the bits received from W_1 and W_2 by w_1 and w_2.

[2] A judge node may be a destination node or the source node or both the nodes. In case of the destination node, it may decide to treat the information as authentic if it infers the relay node of not misbehaving. In case of the source node, it may decide to consider the path $S \rightarrow R \rightarrow D$ secure if it infers the relay node to be not misbehaving.

The judge node collects the decision bits and make a decision as follows:

- $w_1 w_2 = 11$: R is misbehaving;
- $w_1 w_2 = 10$: W_1 is misbehaving;
- $w_1 w_2 = 01$: W_2 is misbehaving;
- $w_1 w_2 = 00$: none of the nodes is under attack.

We remark that our scheme gives a decision based on maximum likelihood probability of a particular node misbehaving. To see the protocol as a maximum likelihood decision making scheme, first consider the two simple cases of the decision bits being 11 and 00; in the former, the relay node must be misbehaving, else W_1 and W_2 can not both detect a misbehavior at R (note that one of them can, if that particular watchdog is misbehaving, it could pretend that the relay node is actually misbehaving), and, in the latter, there is no way to detect which node is misbehaving; indeed there may be no misbehaving node in such a case. For the case of 01 (10), note that if the attacker is at W_1 (W_2), W_2 (W_1) will never send a decision bit 1.

Hence, assuming that each node can be misbehaving with equal probability and the miss-detection probability for W_1 and W_2 are both P_{miss}, we can compute the probability of a particular node misbehaving, given $w_1 w_2 = 01$, as:

$$
\begin{aligned}
P_{W_1|01} &= 0 \\
P_{R|01} &= \frac{P_{miss} \times (1 - P_{miss})}{1 + (P_{miss} \times (1 - P_{miss}))} \\
P_{W_2|01} &= \frac{1}{1 + (P_{miss} \times (1 - P_{miss}))}
\end{aligned}
$$

Recall that the protocol in such a scenario decides that the watchdog sending a decision bit 1 is misbehaving, which is precisely the maximum likelihood decision given such a configuration (note that $P_{W_2|01} > P_{R|01}$).

We show in the following subsections that the misbehaving node can be located with a very high probability using just a single extra watchdog per unreliable relay node. We finally comment on how to bring the probability of correct location detection arbitrarily close to unity.

Let $P_{L|N}$ denote the probability of correctly locating the misbehaving node in the network given the adversary is at node N (where N may be R, W_1, or W_2); $P_{F|N}$ denote the probability that a node other than N is accused to be misbehaving while in fact N is the adversary; and $P_{U|N}$ denote the probability when the adversary at node N operates undetected.

B. Performance – Single Flow Case

For the single flow case, only one extra watchdog is required to locate the adversary in the network (see Fig. 7). We employ the protocol discussed above at destination D. Given this scheme, we have the following lemmas characterizing the performance of the protocol:

Lemma 1: In single flow case of Fig. 7, if any of the watchdogs is misbehaving, it will be located, *i.e.*,

$$
\begin{aligned}
P_{L|W_1} &= P_{L|W_2} = 1 \\
P_{F|W_1} &= P_{U|W_1} = P_{F|W_2} = P_{U|W_2} = 0
\end{aligned}
$$

Proof: Let us assume, without loss of the generality, that W_1 is misbehaving. In such a scenario, W_2 will always send a decision bit 0 to D since it will never overhear any incorrect packet being transmitted by R. A misbehaving W_1, on the other hand, will accuse the relay node of misbehaving. Then, the received decision bits at node D are 10. Given our protocol, D will decide that R is a reliable node and hence, the node W_1 sending a 1 must be misbehaving. Hence, D will always be able to locate the misbehaving node. ∎

The above lemma implies that the adversary has no incentive to attack either of the watchdogs in the network. Using the results of previous sections, this further restricts the capabilities of the attacker: it is not only restricted to corrupt a large number of packets, but is restricted to attack the relay node(s) only. The following lemma, characterizes the performance of the protocol when the relay node misbehaves (corrupts more than $(n-k)$ packets out of n packets):

Lemma 2: In single flow network of Fig. 7, if R is misbehaving, then:

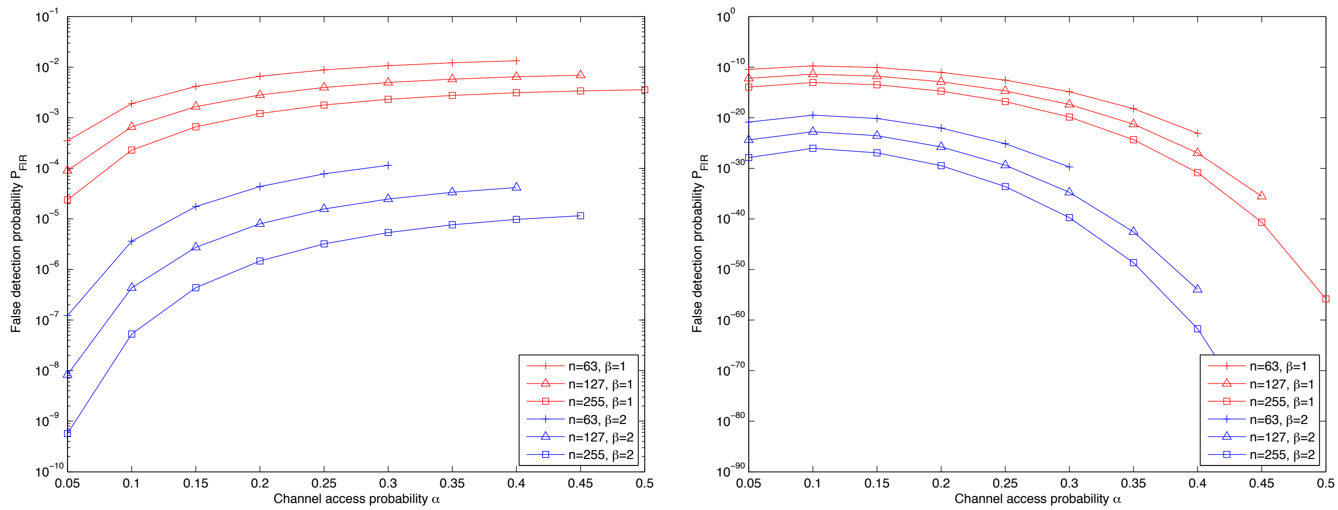
$$
\begin{aligned}
P_{L|R} &= (1 - P_{miss})^2 \\
P_{F|R} &= 2 \times P_{miss} \times (1 - P_{miss}) \\
P_{U|R} &= P_{miss}^2
\end{aligned}
$$

Proof: Let R is misbehaving and the decision bits sent by W_1 and W_2 are w_1 and w_2 respectively. Then, R goes undetected if and only if $w_1 w_2 = 00$, *i.e.*, when both the watchdogs miss all the packets corrupted by the attacker. Hence, the probability of R operating undetected is $P_{U|R} = P_{miss} \times P_{miss}$. On the other hand, R will be detected if and only if none of the watchdogs miss any of the packets corrupted by R, *i.e.*, $w_1 w_2 = 11$, leading to the fact that $P_{L|R} = (1 - P_{miss}) \times (1 - P_{miss})$.

Finally, the case of false detection is when exactly one of the watchdogs miss all the packets corrupted by R, *i.e.*, when $w_1 w_2$ is either 10 or 01, in this case W_1 or W_2 is detected as bad (not R). This gives $P_{F|R} = P_{miss} \times (1 - P_{miss}) + P_{miss} \times (1 - P_{miss})$. Notice that $P_{F|R} = 1 - (P_{L|R} + P_{U|R})$. ∎

The probabilities $P_{F|R}$ and $P_{U|R}$ are plotted in Fig. 8(a) and Fig. 8(b) as a function of channel access probability for $k = n + 1 - \frac{\beta \ln n}{(1-\alpha)^5}$.

In Lemma 2, we have assumed that both the watchdogs have the same probability P_{miss}. This might not be the case since different nodes might observe different channel conditions due to being at different locations. We consider this case in the following subsection but the results of Lemma 2 can be modified easily to incorporate such a difference in probability of W_1 and W_2 missing the detection of packet modification by the relay node.



(a) False detection probability versus channel access probability

(b) Miss-detection probability versus channel access probability

Fig. 8. False detection probability and miss-detection probability in the single flow example with $k = n + 1 - \frac{\beta \ln n}{(1-\alpha)^5}$. Where the curves stop means no code is available.

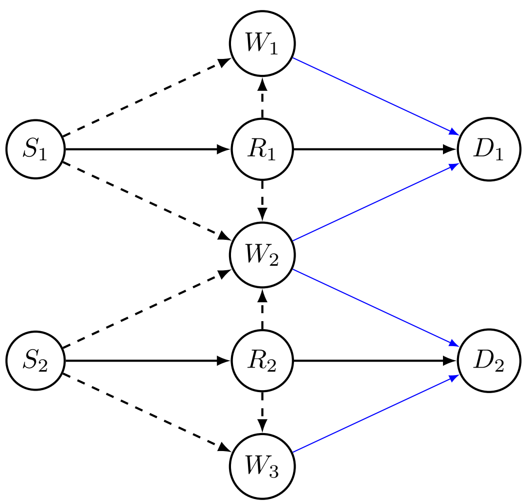


Fig. 9. Two Flow network of Fig. 4 with extra watchdogs. The thick (directed) lines denote a reliable connection from the tail node to the head node, a dashed line denotes the overhearing and a blue line denotes a secure asymptotically negligible rate channel between the two nodes.

C. *Performance – Two Flows Case*

In this subsection, we study the location detection of the misbehaving node for the two flow case of Section III-B. We first consider the case when the destination nodes may collaborate among themselves to locate the misbehaving node and show that such a collaboration does not necessarily reduce the connectivity requirement and/or improve the detection probability as long as the misbehaving node is not oblivious to the attack detection mechanism. We then show that the case of two flow network reduces to the case of multiple single flows with appropriate modifications to the probabilities of missing an attack at the watchdog nodes.

Consider the two flow network shown in Fig. 9, which is an extension of the network shown in Fig. 4 with one extra watchdog per unreliable relay node. Assume that the two destinations D_1 and D_2 collaborate among themselves (share a few bits in order to locate the misbehaving node) and that the misbehaving node is oblivious to any attack detection mechanism in the network. Hence, if watchdog W_2 is the misbehaving node, it will send decision bits 1 to both D_1 and D_2. However, since there is a single misbehaving node in the network, R_1 and R_2 cannot be both misbehaving. If D_1 and D_2 both receive 1 from W_2 they will (collaboratively) decide that W_2 is the misbehaving node. On the other hand, if R_1 (R_2) is misbehaving, W_2 sends a 1 to D_1 (D_2), which will certainly imply that the corresponding relay node is under attack (assuming that W_2 is oblivious to the attack detection mechanism).

Notice that in the above case, we do not need W_1 and W_3 for locating the misbehaving node. The problem arises when the misbehaving node knows that an attack detection scheme is being employed in the network. In such a case, the misbehaving node (at W_2) may send a decision bit 1 to one destination node (say D_1) and a 0 to the other destination node, making D_1 (incorrectly) think that R_1 is actually misbehaving. In such a case, we need W_1 and W_3 to be able to correctly decide the location of the misbehaving node. Note that the above discussion implies that even if several judge nodes start collaborating, at least one extra watchdog per unreliable relay node is required to correctly locate the misbehaving node. Hence, collaboration of judge nodes does not help in reducing connectivity requirements and/or devising a better attack detection scheme.

We note that the above discussion of collaborating judge nodes also captures the multipath transmission mechanism where a source node might relay the information to the same destination via multiple relay nodes (see Fig. 10). Hence, to (correctly) locate the misbehaving node, the connectivity requirements for the network is every relay node being monitored by at least two watchdogs. We derive below the results for the two flow case when the judge nodes do not collaborate but as discussed above, these results hold even if the judge nodes collaborate among themselves.

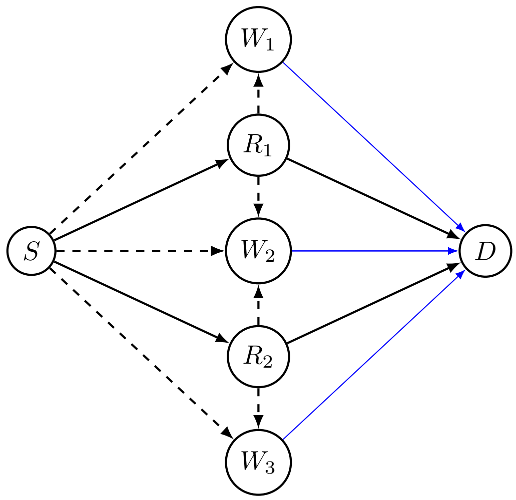


Fig. 10. Two Flow network of Fig. 9 when the judge nodes collaborate among themselves. Also captures the multipath routing case when S relays the information to D via multiple relay nodes.

If the destination nodes do not collaborate, then the decision made by any of the destination nodes, say D_1, is dependent only on the decision bits of the watchdogs observing the corresponding relay node, *i.e.*, W_1 and W_2 for D_1 (similar remarks hold for D_2). This in turn means that each destination node individually behaves as if it is participating in a single flow network. However, as discussed earlier, it might be the case that the watchdogs W_1 and W_3 have probabilities of detection different from that of W_2. The following lemmas hold for the case of two flow network of Fig. 9, where we denote the probabilities of missing an attack at the relay node for watchdogs W_1 and W_3 are $P_{miss,1}$ and that of W_2 is $P_{miss,2}$.

Lemma 3: In the two flow case of Fig. 9, our protocol deterministically locates the attacker if it attacks at any of the watchdogs, *i.e.*,

$$\begin{aligned} P_{L|W_1} &= P_{L|W_2} = 1 \\ P_{F|W_1} &= P_{U|W_1} = P_{F|W_2} = P_{U|W_2} = 0 \end{aligned}$$

Proof: Similar to Lemma 1, collaboration of destination nodes does not play a role. ■

Lemma 4: In the two flow case with our protocol, if the adversary attacks R_1 or R_2, then:

$$\begin{aligned} P_{L|R_1} &= P_{L|R_2} = (1 - P_{miss,1}) \times (1 - P_{miss,2}) \\ P_{F|R_1} &= P_{F|R_2} = P_{miss,1} + P_{miss,2} - 2 \times P_{miss,1} P_{miss,2} \\ P_{U|R_1} &= P_{U|R_2} = P_{miss,1} \times P_{miss,2} \end{aligned}$$

Proof: Similar to Lemma 2, collaboration of destination nodes does not play a role. ■

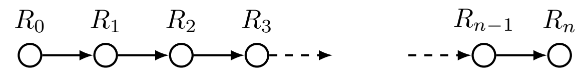


Fig. 11. A multi-hop flow where R_0 is the source, R_n is the destination and each R_i behaves like a watchdog for node R_{i+1}. This network requires at least one more watchdog per unreliable node to locate the misbehaving node.

V. Multihop Routing

In the above sections, we have shown that for each $S \to R \to D$ flow, we need two watchdogs per unreliable relay node to correctly locate the misbehaving node in the network. In this section, we show that this result generalizes to multihop flows. In particular, consider the multihop flow shown in Fig. 11 where R_0 is the source node, R_n is the destination node and information is relayed via relay nodes R_1 to R_{n-1}. We assume the links are bidirectional symmetric such that each relay node R_i behaves like a watchdog for relay node R_{i+1}. We do not loose any generality with such an assumption, since any watchdog watching relay R_{i+1} must listen to both R_i and R_{i+1}. We show that in spite of R_i watching R_{i+1}, we need at least one more watchdog per unreliable path.

Without loss of generality, assume that R_2 is compromised by the adversary and assume that there is no other watchdog other than R_1 that is watching R_2. There are three ways the adversary can attack the data communication:

- R_2 corrupts the packets and claims that R_3 is misbehaving: In such a case both R_1 and R_2 claim their next hop neighbor is misbehaving;
- R_2 only corrupts the packets: In such a case, R_1 claims that R_2 is misbehaving;
- R_2 only claims that R_3 is misbehaving: In such a case, R_1 will not claim that R_2 is misbehaving since R_2 relays all packets correctly.

Since at most one node can be misbehaving, it is easy to see that the only possible reason for the first case is that R_2 misbehaves. So if two nodes claims their next hop neighbor misbehaving, the judge node can always correctly identify the misbehaving node to be the one with a larger index. However, if only one node declares an attack, there is no way for the judge node to differentiate the last two cases.

Hence, the strategy adopted by the misbehaving node in multihop flows is either to corrupt the packets or claim that the node it is watching is misbehaving, but not both. In such a case, we will need at least one extra watchdog per unreliable path to draw correct inferences about the location of the misbehaving node; for example, if we have one watchdog node that can compare the information transmitted by R_0 (say d_t) and transmitted by R_{n-1} (say d_r). Indeed, if $d_t = d_r$, the relay node that claims another node to be misbehaving is indeed the misbehaving node. On the other hand, if $d_t \neq d_r$, then the relay node which is being accused of misbehaving is indeed misbehaving. In the case there is no such node that can overhear transmissions from both the head (R_0) and tail (R_{n-1}) of the multihop flow, we need more than one watchdog each of which can overhear the incoming and outgoing transmissions of a segment of the path such that the union of all the segments monitored by the watchdogs is the whole path.

VI. Multiple Transmissions: Improving Performance & Confidence

In this section, we discuss the benefits of watchdog mechanisms with source error detection coding over multiple rounds in two contexts: improving the probability of correct location detection, and incentives for watchdog nodes to avoid selfish behavior.

Recall from Section IV-C that $P_{L|R} = (1 - P_{miss,1}) \times (1 - P_{miss,2})$. If the location detection is done over multiple rounds, say m, then $P_{L|R}^{(m)} = (1 - P_{miss,1}^{m}) \times (1 - P_{miss,2}^{m})$. Hence, the probability of correct location detection can be made arbitrarily close to unity by doing location detection over multiple rounds.

Note that in the above discussion, we have assumed that none of the nodes behave selfishly. While the relay nodes have no incentive to behave otherwise, the watchdogs are inferred to be misbehaving even when they are not (with probability $P_{F|R}$). The watchdog nodes, hence, have an incentive to always transmit a decision bit 0 so that they are never deemed misbehaving. Having location detection performed over multiple rounds gives enough incentive for the watchdog nodes to avoid such selfish misbehavior.

VII. Final Remarks

In this paper, we have studied the problem of misbehavior detection in wireless networks. We propose a lightweight misbehavior detection scheme which integrates the idea of watchdogs and error detection coding. We show that even if the watchdog can only observe a fraction of packets, by choosing the encoder properly, an attacker will be detected with high probability while achieving throughput arbitrarily close to optimal. We then propose a simple protocol which, by using just one extra watchdog per relay node, locates the misbehaving node with probability approaching to unity.

There are several possible extensions to the results of this paper. First, our results may not directly apply to networks that have several misbehaving nodes, for example if both the relay node and one of the watchdogs are misbehaving. In such cases, the relay node can alter the packets as much as possible without being detected as long as the faulty watchdog never declares an attack. It would be interesting to extend the ideas in this paper to consider the case of multiple colluding adversaries in two contexts: one, what are the connectivity requirements to detect misbehavior in presence of multiple colluding adversaries, and, second, evaluating the achievable performance in terms of throughput and false/miss detection probabilities in the presence of multiple colluding adversaries.

We have also assumed existence of a *reliable* channel between the watchdogs and the judge nodes which is used to transfer the decision bits. While this assumption is quite acceptable since only one bit is required to be transmitted from the watchdog node(s) to the destination(s), the relay node might intentionally interfere while the decision bit is being transmitted from the watchdogs to the judge node, which might preclude the judge node of receiving the decision bits. It would be interesting to see if a scheduling mechanism could be enforced to limit such an action from the attacker and evaluate the achievable performance in presence of such scheduling mechanisms.

References

[1] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 255–265.

[2] T. Ghosh, N. Pissinou, and K. Makki, "Towards designing a trusted routing solution in mobile ad hoc networks," *Mob. Netw. Appl.*, vol. 10, no. 6, pp. 985–995, 2005.

[3] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pp. 90–100, Feb 1999.

[4] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 226–236.

[5] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "A quantitative trust establishment framework for reliable data packet delivery in manets," in *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2005, pp. 1–10.

[6] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2004, pp. 66–77.

[7] D. C. Kamal, D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *In Proceedings of the fortieth annual Conference on Information Sciences and Systems*, 2006.

[8] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 1409–1417, April 2008.

[9] F. Zhao, T. Kalker, M. Medard, and K. J. Han, "Signatures for content distribution with network coding," in *In Proc. of International Symposium on Information Theory (ISIT*, 2007.

[10] Q. Li, D.-M. Chiu, and J. Lui, "On the practical and security issues of batch content distribution via network coding," *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pp. 158–167, Nov. 2006.

[11] M. N. Krohn, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *In Proceedings of the IEEE Symposium on Security and Privacy*, 2004, pp. 226–240.

[12] C. Gkantsidis and P. Rodriguez Rodriguez, "Cooperative security for network coding file distribution," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[13] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks using randomized network coding," 2004.

[14] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient network coding in the presence of byzantine adversaries," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 616–624, May 2007.

[15] M. Kim, M. Medard, and J. Barros, "Counteracting byzantine adversaries with network coding: An overhead analysis," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008, pp. 1–7.

[16] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in *WiSec '09: Proceedings of the second ACM conference on Wireless network security*. New York, NY, USA: ACM, 2009, pp. 111–122.

[17] M. Kim, R. Kotter, M. Medard, and J. Barros, "An algebraic watchdog for wireless network coding," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 28 2009-July 3 2009, pp. 1159–1163.

[18] G. Liang and N. Vaidya, "When watchdog meets coding," *Technical Report, CSL, UIUC*, May 2009.