

# Attack-Resistant Routing : a Framework for Enhancing Routing Robustness in Wireless Ad-hoc Networks

Gi Tae Kim  
Perspecta Labs  
Basking Ridge, NJ  
gkim@perspectalabs.com

Constantin Serban  
Perspecta Labs  
Basking Ridge, NJ  
cserban@perspectalabs.com

Ritu Chadha  
Perspecta Labs  
Basking Ridge, NJ  
rchadha@perspectalabs.com

**Abstract**—Optimizations in wireless ad-hoc network control protocols, such as shortest path routing and reduced topology set operations, are widely employed and recognized to decrease overhead and improve performance in benign conditions. Such optimizations, however, result in vulnerabilities that can decrease the robustness of the network when misconfigured, and can be readily exploited under various adversarial models. In this paper, we study the effects of such optimizations in the presence of misconfiguration and attacks, and we propose an *Attack-Resistant Routing (ARR)* framework that mitigates adversarial effects. While many other methods of combating certain control plane attacks have been proposed, our method is unique in that it assumes no prior knowledge of the mechanisms used to effect the attacks and the type of misinformation affecting the network. Rather, it alleviates the vulnerabilities introduced by various optimizations, thus increasing network robustness when the network is under attack or misconfigured. We show that ARR can withstand multiple and severe attacks and restore network performance to levels close to normal.

**Keywords**—Multi-Topology Routing (MTR), OLSR (Optimized Link State Routing), SMF (Simple Multicast Forwarding), Wireless Ad-hoc Networks, Black hole attack, Wormhole attack.

## I. INTRODUCTION

Control plane protocols implement essential network functionality that enables an otherwise disparate set of devices to function together as a coherent network, capable of transporting data efficiently based on application and user needs. In the context of wireless ad-hoc networks, most control protocols have been designed, implemented, and tuned with the primary goal of optimizing network performance, while other goals, such as security and various aspects of robustness, are considered secondary. For example, routing protocols (subnet unicast routing, multicast routing, gateway election, etc.) optimize network performance using techniques such as control plane overhead reduction via topology control (NHDP[28], OLSR[29]), partial topology maintenance (AODV[30], DSDV[31]), and reduced relay set selection (SMF[32], PIM[33]); and data plane overhead reduction via shortest path routing (with or without weighting criteria), used quasi-universally in this domain. While such optimizations have been shown to be advantageous in numerous situations, a large body of research has shown over the last two decades that they also open up a set of failure models, or vulnerabilities, which, when exploited, can drastically impact network performance.

This paper addresses the robustness of routing protocols in wireless ad-hoc networks when false or inconsistent information is injected into the control plane, as a result of misconfiguration or malicious attacks. Even though such false information may

represent a relatively small portion of the entire information base handled by the control protocols, it can have a disproportionate effect on the overall functionality of the protocols, making them unreliable at best, or completely ineffective at worst. In this paper we will show how this vulnerability is driven to a large extent by various performance-driven enhancements, and how relaxing such enhancements reduces or eliminates the effects of misinformation. The paper introduces *Attack-Resistant Routing (ARR)*, a framework for introducing multi-path diversity in existing routing protocols, aimed at restoring their robustness to misinformation. The introduction of specific objective functions that enables the computation of different topology variations makes the multi-path enabled protocols significantly more resilient against multiple classes of attacks. While state of the art approaches are designed to prevent *specific* types of attacks, the ARR framework presents a novel approach designed to address more generic and unknown attacks without making assumptions about how they were implemented. We evaluate ARR in a variety of conditions, and we show that it performs well under a multitude of attacks, either simultaneous or otherwise, and we show that in most situations it can recover the traffic affected by misinformation, thus increasing the robustness of the network.

Section II introduces the attack model, presents different attack varieties along with their impact on data traffic, and discusses various routing protocol design decisions that exacerbate these impacts. In Section III, we present existing approaches aimed at defending against attacks and their shortcomings. In Section IV we introduce our Attack-Resistant Routing approach, followed by a description of our mechanism for generating routing diversity in Section V, and their application on new topologies and paths computations in Section VI. Section VII presents the results of our evaluation, followed by conclusions.

## II. ATTACK MODELS

A wireless ad-hoc network consists of a set of wireless nodes, each running a link discovery protocol and a set of routing protocols exchanging link information for partial or complete topology discovery for route computation and generation of a forwarding information base that drives the forwarding of unicast and multicast traffic. We consider the following attack models:

**False link advertisement** [11][26]: An attacker injects a fictitious (non-existent) link into the network topology base, at one or more routers. The attack can be carried out by one of the routers on a network node (*in-network attacker*), or by outside actors making a link appear to be genuine to the link discovery protocol via lower-layer mechanisms (*out-of-network attacker*). In-network attacks can be classified as either *misadvertisements* or *link*

*spoofing*, depending on the type of link advertised (self or not). For out-of-network attacks, the false link advertisement might be carried out via physical layer or MAC layer repeating. While a false link introduced in the entire topology base might only represent an insignificant portion of the entire topology base, in certain cases it can have significant effects, *e.g.*, when the false link straddles two topologically distant nodes in the network. In such cases, shortest-path based routing algorithms, present in the vast majority of the ad-hoc routing schemes, will introduce this link in the forwarding base of many data paths, hence inflicting large damage. The loss in robustness in this case is entirely attributable to the use of the shortest path algorithm.

**Black hole attacks** [5][24]: These attacks are generalizations of the in-network false link advertisement attacks, where a (large) number of misadvertised links are used. When an attacker node misadvertises links to all participants in the network (other than its genuine neighbors) the topology base represents a network graph with a diameter of two. As such, any path between nodes that are otherwise at least *three hops away* will traverse a misadvertised link whenever a shortest-path criterion is used. The problem, however, is more severe: for routing protocols based on reduced relay set selection algorithms (SMPR[36], ECDS [37], PIM/DM [38] etc.), any path between nodes that are otherwise *two hops away* will be affected: topology control optimizations will eliminate legitimate two-hop paths from the global topology in the favor of more popular (but false) two-hop paths, hence further reducing network resiliency. Black hole attacks can come in many flavors, such as partial or full; permanent or temporary; stationary or rotating; and single-attacker or coordinated. Black hole attacks may drop data traffic directly, via loops or instability.

**Wormhole attacks** [15][27]: These attacks are false link advertisements coordinated between one or more attackers. In wormholes, the links are not misadvertisements; rather, they are created between two benign nodes (the victims) by neighboring attacker nodes that capture and replay victims' network transmissions, either in-network (at the MAC layer) or out-of-network (at the physical layer). Wormholes can be created by a single attacker, as a repeater node, or by multiple attackers, via tunnels. Judiciously placed wormholes can take advantage of shortest path routing to attract a disproportionate amount of traffic, or can take advantage of relay set selection optimizations to eliminate other competing relays, thereby further increasing their effects. Wormholes can also come in many flavors, such as partial or full (affected victims); permanent or temporary; stationary or rotating; single-party, pairwise, or multi-party (among attackers). Wormhole attacks may rely on network instability or heuristics for choosing the data-plane traffic to drop.

The above attacks can be applied to different protocols, for both unicast and multicast traffic, or changed for targeted effects.

### III. DEFENSE APPROACHES

Traditional approaches for protecting against control plane attacks are largely based on perimeter defense. In such models, nodes are considered trusted as long as they are part of the network, and the onus is implicitly placed on host security to prevent compromise of control plane processes. This approach is clearly not adequate given the pervasive nature of host compromises, and the advent of technology that lowers the bar for conducting out-of network attacks. The past decade has seen a significant number of efforts addressing the problem of control plane attacks, specifically focused on routing vulnerabilities

described in Section II. Many of the efforts ([27][21][1][19] to mention just a few) attempt to secure the control protocols (OLSR [29], AODV [30], and others) using digital signature-based authentication of control plane messages, using either single party and multi-party signatories. While source authentication is a basic and necessary security approach, it only prevents the simplest attacks and does not address insider attacks. Multi-party authentication methods often suffer from high overheads, significantly decrease the response time of the protocols, and provide no solutions against the most crippling forms of attacks, such as wormholes, coordinated false link advertisements, and collusion black holes. The essential problem with these approaches is that they do not account for the many ways in which false information may be injected in the protocol stack, bringing only limited security benefits.

While the above approaches are proactive in nature, a different line of work addresses the compromise of control plane protocols reactively, using various forms of misbehavior detection followed by specific mitigations. Works such as [3] [2] [4] [8] [16] [27], and our own work in [7] use specific detection techniques to identify various pieces of misinformation injected in all relevant control protocols using a variety of approaches (invariant-based, anomaly detection, intrusion detection). While some of the approaches can be effective for specific attacks, such approaches suffer from several drawbacks. First, they most often do not generalize to all aspects of misinformation and are laborious to develop. Second, precise mitigations require precise identification of the misinformation as well as identification of the party responsible for that misinformation. This is an especially challenging problem when the attacker employs dissimulation, and it can have a serious negative impact if not addressed appropriately. Third, in order to be effective, detection/response measures need to be performed rapidly and accurately, which is very hard if possible. These problems are not yet solved in a general sense.

Concepts for implementing multi-topology routing for fault tolerance have been studied, among others, in [20], [9], [23], [17], and [18]. These works aim to improve fault tolerance in different ways. Authors in [13] explored the benefits of using multi-topologies for routing based on device capacities and traffic bandwidth requirements in wireless ad-hoc networks. Multi-topology routing is used in [12] for proactively switching to different routing topologies while convergence activity occurs on a topology. Authors in [14] and [22] propose to manage the multi-topology for effectively supporting traffic in different QoS classes for enhanced throughput. To the best of our knowledge, our work is the first to use multi-topology routing in a general framework aimed at defending against misinformation attacks.

### IV. ATTACK-RESISTANT ROUTING: CONCEPT

Attack-Resistant Routing (ARR) is a novel approach to routing protocols that achieves robustness through route diversification. Rather than relying on the prevention of injection of misinformation in the control plane, or on the discovery and correction of misinformation after the fact, as performed by other approaches, ARR operates in environments where misinformation is intrinsically present, either due to malicious attacks or misconfiguration errors. The basic idea behind ARR is to use multi-path routing, with different routing paths chosen such that they are unlikely to be affected by the same types of attacks. The most important assumption that enables our solution to work is the existence of some genuine routing paths in the

overall tainted topology base: We do not make assumptions about the ratio of the false information to genuine information in the control plane, the placement of attacker nodes, and of the false links/paths thus created, as long as genuine paths still exist in the network. This broadly fits the model of an attacker with limited footprint (physically and topologically) but unlimited capabilities within the compromised node/area. ARR aims to satisfy the following broad requirements: i) low or no communication overhead: simultaneously running multiple routing protocols was deemed excessively costly; ii) operate within existing protocol stacks, with minor or no modifications: no new protocol or full protocol re-design; and iii) low or no coordination overhead in achieving path diversity: albeit coordination of route diversity is plausible, such coordination would open other attack vectors.

ARR operates in two phases:

- *Inflation*: ARR boosts the topology base generated by the control protocol suite (link discovery, neighborhood discovery, topology control, routing) by disabling all the topology control optimizations used by the existing protocol suite. This is usually achievable via static or runtime protocol configurations. The role of this stage is to prevent the elimination of genuine links from the topology due to the false links. The inflated topology, henceforth called *default topology*, although tainted with false links, will thus carry enough genuine links capable of supporting data plane traffic.
- *Controlled Deflation*: ARR uses a set of objective functions for controlled pruning of links from the default topology, such that the *pruned topology* achieves a certain norm, such as maximally disjoint paths, maximally disjoint nodes, etc., as discussed later (note that “pruning” in this context does not actually delete links but assigns them with high costs to; these links can still be used when encountering a partition.). Each of these pruned topologies constitutes a new topology information base that permits the computation of different routing paths and forwarding routes according to the original implementation of the routing protocol, hence providing multipath routing. This is usually achievable by manipulating the topology information base in the routing protocol, with little or no code modifications.

ARR satisfies requirement i) above by using a single default protocol for exchanging information and creating a comprehensive default topology; it satisfies requirement ii) by reusing most functionality from the existing protocol stack, without requiring a new implementations; and it satisfies iii) by designing the objective functions such that they can be computed independently at each node in the network and arrive at the same globally consistent pruned topology.

We implemented ARR for a control protocol stack consisting of NHDP [28], OLSRv2 [35], and SMF [32]. Fig. 1 shows the processing steps employed in this architecture. First, a default topology is created by enabling full Multi-Point Relay (MPR) [34] network topology (*i.e.*, no MPR optimization). This topology is built in a distributed manner using NHDP [28] and OLSR [29] exchanges (Hello and TC messages). After the topology is built at each node, a set of objective functions  $\langle OF_1, \dots, OF_n \rangle$  are applied, simultaneously producing a set of topologies  $\langle T_1, \dots, T_n \rangle$  each satisfying specific norms (one of the objective functions is a NoOp, producing the default topology and behavior). These topologies are further used by OLSRv2 [35] and SMF [32] independently for computing the unicast forwarding base and the multicast relay set, each getting populated in different forwarding

tables  $\langle FT_1, \dots, FT_n \rangle$ . Data plane traffic is assigned to one or more forwarding tables via a Traffic Flow Mapping Function. The exact implementation of the Traffic Flow Mapping Function is beyond the scope of this paper. Examples of approaches are: source-driven mapping via DiffServ/ToS markings; epoch-based switching of forwarding tables; or performance feedback-driven forwarding table selection. Regardless of the scheme, different flows will use different forwarding tables at different times, hence creating topology diversity. The construction of such topologies from distinct information bases dilutes the amount of misinformation present in the default topology and ensures different or no effects in each forwarding plane. Hence, the ARR approach satisfies many of the design principles of a cyber-resilient system, as defined in [6], via selective exposure of attack surface, controlled visibility, maintained redundancy, managed diversity, etc. Note that although our discussion and implementation takes place in the context of an NHDP/OLSRv2/SMF [28][35][32] stack, the ARR concept can easily apply to other link state and non-link state control protocol stacks and their implementations (M/OSPF [38], PIM [33], RIP [39], AODV [30], etc.) following the same principles.

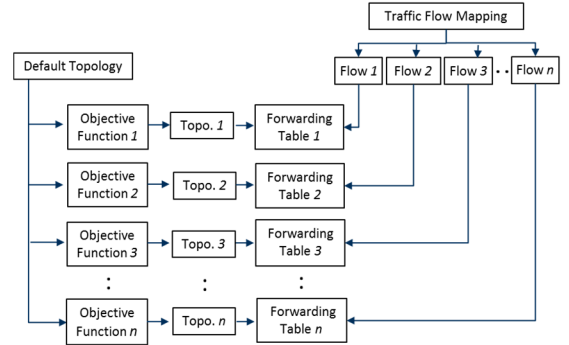


Fig. 1. Multi-topology Routing

## V. OBJECTIVE FUNCTIONS

ARR achieves robustness by generating multiple topologies using objective functions. An objective function performs a specific optimization on the default topology generated by the control plane protocols. The optimized topology includes a reduced set of links to induce paths that are otherwise suboptimal in terms of distance but robust to attacks.

Let  $P$  be the optimization problem of finding a new topology. An objective function is defined as:

$$P \text{ maximize } (\sum_{i=1}^N X^i)^j,$$

where  $X^i$  is a property of the path for the  $i^{th}$  node-pair (*i.e.*, the  $i^{th}$  source and destination nodes) in the new topology, with respect to the default topology (*i.e.*, shortest path for the  $i^{th}$  node pair),  $N$  is the number of all node pairs in the network, and  $j$  is the number of trials. The function  $X$  is associated with a path metric that indicates how close the path is to the objective. The optimization process, in each iteration, generates a pruned topology and it computes its metric. It then takes the topology with the highest metric. Below we discuss some of the objective functions that we developed for ARR:

**Maximally Disjoint Paths**: The underlying routing scheme establishes routes based on the shortest path assumption, and induces traffic onto the shortest paths. The goal of this objective function is to induce traffic away from default shortest paths

when attacks are present on the default paths. To this end, it finds a secondary robust path that is most disjoint from the default shortest path for all node pairs. The path disjointness for a pair of paths is the portion of routes that are not commonly shared by the two paths. Here, the path disjointness is considered in terms of links, not in terms of nodes, as shown in Fig. 2 below, where the sharing of node “a” in both the default shortest path and robust secondary path does not affect the disjointness metric: the two paths are fully disjoint. This objective function is particularly effective in maintaining routing robustness under out-of-network wormhole attacks.

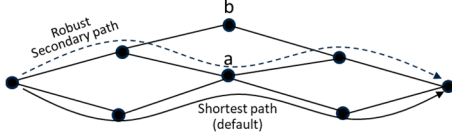


Fig. 2. Max. Disjoint Paths in Terms of Links

To measure the disjointness, the path disjoint ratio metric  $D$  is defined as follows:

$$D_{AB} = \frac{|L_{default}| - |L_{default} \cap L_{second}|}{|L_{default}|}, \text{ where}$$

$D_{AB}$  = disjointness ratio for path from  $A$  to  $B$

$L_{default}$  = links in the default paths from  $A$  to  $B$

$L_{second}$  = links in the robust secondary paths from  $A$  to  $B$

The average of  $D_{AB}$  for all source ( $A$ ) and destination ( $B$ ) node pairs in the network is the overall disjointness ratio. The objective function finds a sub-graph of the default topology that returns smallest average  $D$ .

$$MAX [ D = avg(\sum_{A \in N, B \in N} D_{AB}) ]$$

**Maximally Disjoint Nodes:** This objective function is similar to the “Maximally Disjoint Paths” discussed above, except that the disjointness is measured in terms of shared nodes rather than links. As shown in Fig. 3, no node is shared between the default shortest and the secondary paths. This objective function can be especially effective in maintaining robustness under in-network black hole attacks.

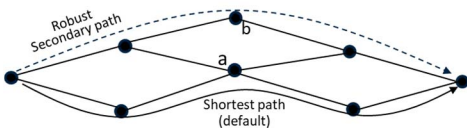


Fig. 3. Max. Disjoint Paths in Terms of Nodes

To evaluate the disjointness ratio, the metric  $D$  is defined as follows:

$$D_{AB} = \frac{|N_{default}| - |N_{default} \cap N_{second}|}{|N_{default}|}, \text{ where}$$

$D_{AB}$  = disjointness ratio for path from  $A$  to  $B$

$N_{default}$  = the nodes in the default paths from  $A$  to  $B$

$N_{second}$  = the nodes in robust secondary paths from  $A$  to  $B$ .

The average of  $D_{AB}$  for all source ( $A$ ) and destination ( $B$ ) node pairs in the network is the overall disjointness ratio. The objective function finds a subgraph of the default topology that returns the smallest average  $D$ .

$$MAX [ D = avg(\sum_{A \in N, B \in N} D_{AB}) ]$$

**Red Area:** This objective function assumes a set of avoidable nodes (red area) are known a priori to the entire network. All

links associated with the known nodes will be pruned and therefore the robust secondary paths will bypass these nodes. (Fig. 4) This objective function can be effective in maintaining routing robustness when attackers or victims are known. We mostly use this objective function as an ideal case for comparison purposes, *i.e.*, when a defender has close to perfect knowledge.

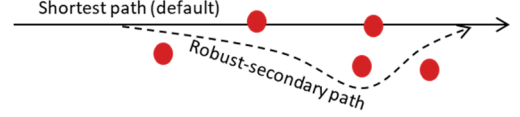


Fig. 4. Example of Red Area

The metric  $R$  is defined as follows:

$$R_{AB} = \frac{|L_{second}|}{|L_{red}|}, \text{ where}$$

$R_{AB}$  = violation ratio

$N_{second}$  = the nodes on the robust secondary path from  $A$  to  $B$  that pass through the red area

$N_{red}$  = the nodes in the red area.

The average of  $R_{AB}$  for all source ( $A$ ) and destination ( $B$ ) pairs in the network is the overall ratio,  $R$ . The objective is to find the routing topology that returns smallest average  $R$ .

$$MIN [ D = avg(\sum_{A \in N, B \in N} R_{AB}) ]$$

**Additional objective functions:** One can imagine different objective functions that target different properties, failure, or performance models. Additionally, a new objective function can be derived by composing multiple existing objective functions in arbitrary ways. An example of such composition is finding most disjoint paths that at the same time are passing through a Blue area.

## VI. TOPOLOGY & FORWARDING TABLE GENERATION

The overall ARR procedure for topology generation and forwarding table computation is given below. The procedure is executed each time the default routing protocol updates the default forwarding table.

- 1) **Select a set of links:** From the default topology select ‘ $p$ ’ links to be pruned, based on the objective function used. For example, when the ‘Maximally Disjoint Path’ objective function is used, the process selects  $p$  most popular links, where the link popularity is represented by a link use count; the use count for a link is incremented each time the link appears on individual  $n(n-1)/2$  default shortest paths, where  $n$  is the number of nodes in the network.
- 2) **Prune selected links:** Prune the  $p$  selected links from the default topology. The reduced topology now contains links with the updated costs (for pruned links) and with original costs (for unpruned links).
- 3) **Run shortest path algorithm on all node pairs:** Using the reduced topology generated in the previous step, compute a new set of paths using Floyd-Warshall algorithm [10]. A single execution of the algorithm finds shortest paths for all node pairs in the graph. This is necessary in order to compute the metric for the objective function in the next step.
- 4) **Compute metric:** Compute a metric corresponding to the resulting path configurations as per the goals of the objective

function. For example, when using the ‘Maximally Disjoint Paths’ objective function, an average ‘path disjoint ratio’ is computed that measures the disjointedness of paths in the pruned graph as compared to the default shortest paths for all node pairs.

- 5) **Evaluate metric for generating forwarding table:** Check if the metric (*e.g.*, average disjoint ratio) computed in the previous step is optimal. This can be done in various ways; a simple stop criterion is to check if the ratio is above a given threshold  $TH$  ( $0 < TH \leq 1.0$ ). If true, the algorithm stops and uses the new topology for generating entries in the forwarding table. Otherwise, go back to step 1.

An important requirement for the ARR process is that all nodes execute exactly the same procedure, computing the same reduced topology in a distributed manner. For this reason, we do not employ non-deterministic behavior, as the resulting topology may diverge across nodes. The overheads for ARR takes  $O(N^3)$  in time for running the Floyd-Warshall algorithm for computing forwarding table, and requires  $O(N)$  in space for maintaining the ARR forwarding table, where  $N$  is the number of nodes in the network. There is no control message exchanges required among nodes for running the ARR.

## VII. PERFORMANCE EVALUATION

We have evaluated the ARR under various networking conditions using NHDP/OLSR/SMF [28][29][32] protocol stack executed in an EMANE [25] network testbed. The main performance metric used in this paper is the packet delivery ratio (PDR)<sup>1</sup> resulting from the use of various objective functions, which are compared against the PDR of the default protocol stack. To assess robustness, we use two types of control plane attacks: in-network wormhole and rotating black hole attacks with multiple attackers. We first discuss the experimental setup, and then analyze the performance of ARR.

### A. Testbed setup

Table 1. Traffic Flows.

| Flow Group   | Transport | Source          | Dest.     | Pkt Gen. Rate (pkt/sec) | Pkt Payload Size (byte) | Aggr. Flow Rate (kbps) |
|--------------|-----------|-----------------|-----------|-------------------------|-------------------------|------------------------|
| #1 unicast   | TCP       | All nodes       | Node 41   | 0.1                     | 512                     | 19.7                   |
| #2 unicast   | UDP       | All Nodes       | Node 44   | 0.1                     | 512                     | 19.7                   |
| #3 unicast   | UDP       | All Nodes       | Node 45   | 0.1                     | 512                     | 19.7                   |
| #4 unicast   | UDP       | 11 Sensor nodes | Node 48   | 1                       | 512                     | 13.2                   |
| #5 multicast | UDP       | 11 Sensor Nodes | All Nodes | 0.1                     | 100                     | 1.1                    |

The physical network attributes, traffic, and other experimental settings are as follows: Network consists of 48 Wi-Fi nodes in a single ad-hoc subnet, with the topology shown in Fig. 5. Default protocol stack uses NHDP/OLSRv2/SMF [28][35][32]. The scenario runs for 700 seconds. Attacks start at time 100 seconds and last for 400 seconds. Details of the attacks are described later. A total of 163 traffic flows are used in the experiment; 47 TCP flows, 105 UDP flows, and 11 multicast flows from five flow groups as summarized in Table 1.

### B. Experiments with wormhole attacks

In this experiment, we activate five pairs of wormholes, with two colluding attackers each, as shown in Fig. 5.

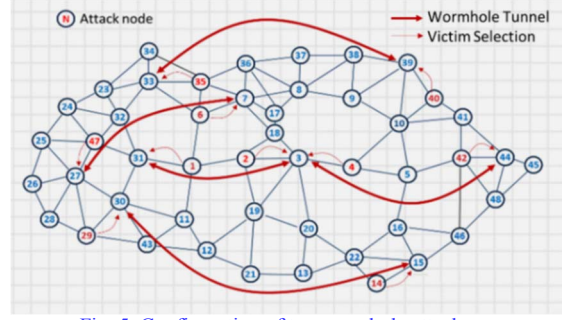


Fig. 5. Configuration of ten wormhole attackers.

The attackers only repeat link messages (NHDP Hellos), to support the creation of the false link, but do not repeat data plane messages, hence causing these packets to be lost. Since the wormholes in these scenarios span longer topological distances, they become preferred links, creating very effective attacks. We evaluated robustness against different temporal wormholes as follows:

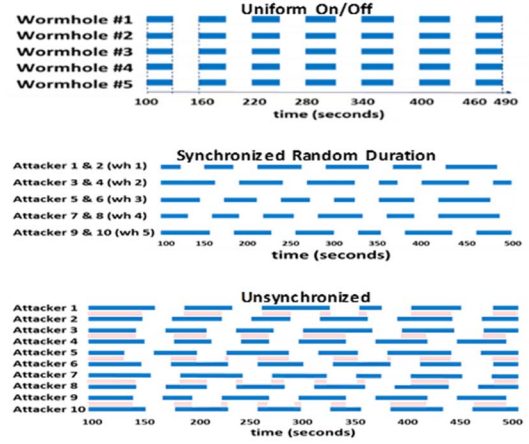


Fig. 6 Wormhole Attack Patterns

1. **Uniform On/Off:** During the attack each of the ten wormhole attackers repeats the same 30-second ‘On’ period followed by a 30-second ‘Off’ period. The ‘On/Off’ patterns are synchronized. The five wormhole tunnels activation time (blue) are shown in Fig. 6 top (Exp #1).
2. **Pair-wise Synchronized Random Duration:** Five pairs of wormhole attackers independently repeat ‘On/Off’ pattern. A pair of attackers choose a non-uniform random ‘On’ duration between 20 and 60 seconds, and the duration of ‘Off’ phase is a fixed 30-second. The ‘On’ phases of the five wormholes are randomly overlapping as shown in Fig. 6 middle (Exp #2).
3. **Unsynchronized Random Duration:** Ten wormhole attackers independently repeat the ‘On/Off’ pattern. Each attacker chooses a uniform random duration between 20 and 60 seconds for the ‘On’ phase, and a fixed duration for the ‘Off’ phase (30 seconds). Fig. 6 bottom (Exp #3) shows the attack patterns: a wormhole is effective only when the two attackers are ‘On’ simultaneously (light shaded).

Each of the above three experiments compare performance under the following four different conditions:

<sup>1</sup> PDR is computed by dividing the total number of the received packets by the total number of sent packets. For TCP traffic, duplicate packets and pending

packets that are not actually received at the end of simulation are not considered as the received packets.

1. **Baseline:** Traffic in this scenario is routed using an unmodified protocol stack. No attack is present.
2. **Baseline with wormhole:** Five wormholes are executed. Traffic is routed via an unmodified protocol stack.
3. **ARR with wormhole and Max Disjoint Paths:** Five wormholes are created as above. Traffic is routed through ARR's using *Maximally Disjoint Paths* objective function.
4. **ARR with wormhole and Red Area:** The same as above, except that it uses the "Red Area" objective function.

Fig. 7 shows the per-flow packet delivery ratios (PDRs) for each of the four scenarios in the three experiments: Experiment #1 (upper), Experiment #2 (middle), and Experiment #3 (lower). Each experiment shows PDRs (as a percentage) for each traffic flow, with the average PDRs across flows shown on the right. All PDRs are captured during the attack period only.

The results are as follows: ARR shows consistently high PDRs across different attack patterns for all five flows, with an overall PDR increase of ~65%. Additionally, ARR shows a performance comparable to the baseline scenario, suggesting that the ARR is highly effective in neutralizing these attacks.

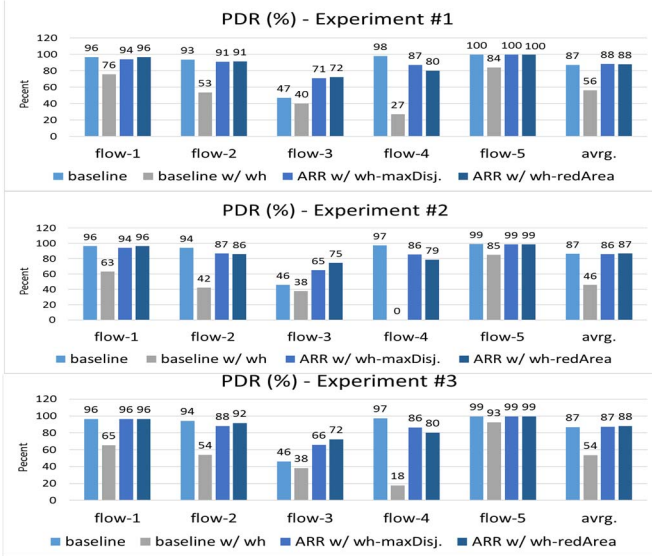


Fig. 7. Packet Delivery Ratios (PDRs) w/ wormhole Attacks

### C. Experiments with black hole attacks

This section compares the performance of individual scenarios when black hole attacks are present. The experiments were repeated at three different attack levels: 2, 5, and 10 attackers. The attackers are located in the following nodes:

- 1) Two black holes: Nodes 1 and 9
- 2) Five black holes: Nodes 1, 7, 9, 14, and 21
- 3) Ten black holes: Nodes 1, 3, 5, 6, 7, 9, 11, 14, 21, and 37.

For each of the attack levels we ran the following scenarios:

- 1) Scenario 1: Baseline. Traffic in this scenario is routed via an unmodified protocol stack. No attack is present.
- 2) Scenario 2: Baseline w/ BH: Black hole attackers are included. Traffic is routed via unmodified protocol stack.
- 3) Scenario 3: ARR w/ BH: Black hole attacks are exercised. Traffic is routed via ARR using 'Popular Disjoint Nodes' objective function.

Fig. 8 shows the PDRs in these scenarios: two attackers (up), five attacker (middle), and ten attacker (low). PDR is reported for each of the five reference traffic flows as well as for the average across flows. With ARR, the PDRs are maintained high even under the maximum level of attack, compared to that of the unmitigated scenario (i.e., baseline w/BH). The ARR scenario shows an average PDR that represents over 219% improvement over the baseline w/BH case.

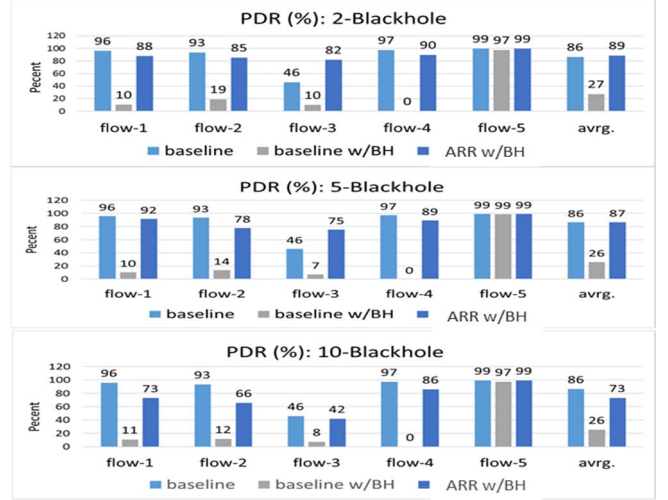


Fig. 8. Packet Delivery Ratios (PDRs) w/ black hole Attacks.

## VIII. CONCLUSIONS AND FUTURE WORK

This work describes a general approach for handling misinformation in control plane protocols using a diversity-based multi-topology routing framework. Our Attack-Resistant Routing (ARR) framework allows the (dynamic) introduction of various objective functions that create a diverse set of sub-graphs of the network, which results in a robust communication medium. The use of the objective functions enables the network manager to tailor the routing topology to effectively neutralize the adverse effects of various faults or attacks in the control plane, without a specific need to identify the exact misbehavior or the culprit. Our experiments have demonstrated that the ARR can effectively mitigate the impacts of the wormhole and black hole attacks, without explicit knowledge of the attackers or attacks. The positive impact of ARR was observed in the following two important categories: (i) ARR maintains a level of performance close to, or sometimes exceeding, that of default OLSR [29] routing in benign conditions, and (ii) ARR shows a sustained level of performance under different attack patterns.

Specifically not addressed in this work are (1) the dynamic selection of objective functions based on the analysis of network conditions, (2) performance analysis under mobile environment, (3) scalability of the ARR for large networks, and (4) comparative performance studies of our work with other fault-tolerant routing schemes - these topics are left for future work.

## ACKNOWLEDGMENT

Supported in part by AFRL and DARPA under contract FA8750-15-C-0038. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. DISTRIBUTION A. Approved for public release: distribution unlimited

## REFERENCES

- [1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, D. Raffo, "Securing OLSR Protocol", in Proc. IFIP Med-Hoc-Net. 2003
- [2] Asmaa Adnane, Rafael T. de Sousa, Jr., Christophe Bidan, and Ludovic Mé, "Economic trust reasoning enables misbehavior detection in OLSR," In Proceedings of the 2008 ACM symposium on Applied computing (SAC '08). ACM, New York, NY, USA, 2006-2013. DOI: <https://doi.org/10.1145/1363686.1364173>[1]
- [3] Todd R. Andel and Alec Yasinsac, "Surveying Security Analysis Techniques in MANET Routing Protocols," IEEE COMMUNICATIONS SURVEYS, 4<sup>th</sup> Quarter 2007, VOLUME 9, NO. 4.
- [4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An On-demand Secure Routing Protocol Resilient to Byzantine Failures," Proceedings of the ACM Workshop on Wireless Security, pp. 21-30, 2002.
- [5] Wu B, Chen J, Wu J, Cardei M: A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks. In Wireless Network Security. on Signals and Communication Technology. Edited by: Xiao Y, Shen X, Du D-Z. Springer, New York; 2007
- [6] D. Bodeau, R. Graubart, "Cyber Resiliency Design Principles", MITRE Technical Report, January 2017.
- [7] R. Chadha, A. Poylisher, C. Serban, "Reliability Estimation in Mobile Ad hoc Networks", in CNSM 2013: International Conference on Network and Service Management.
- [8] Jin-Hee Cho, Ananthram Swami, Ing-Ray Chen, "A Survey on Trust Management for Mobile Ad-Hoc Networks", IEEE communications surveys and tutorials, vol. 13, no. 4, pp. 562-583, 2011.
- [9] T. Cicic, "On basic properties of fault-tolerant multi-topology routing," Computer Networks, vol. 52, no. 18, pp. 3325 – 3341, 2008. <http://www.sciencedirect.com/science/article/pii/S1389128608002831>
- [10] Robert W. Floyd (June 1962). "Algorithm 97: Shortest Path". Communications of the ACM. 5 (6): 345
- [11] N. Gabi, K. Alex, G. Dima, "Persistent OSPF Attacks", NDSS, 2012.
- [12] M. Gardner, R. May, C. Beard, and D. Medhi, "Using Multi-Topology Routing to Improve Routing during Geographically Correlated Failures", 10th DRCN, pp. 1-8, April 2014.
- [13] Audun Hansen and Paal Engelstad, "Profiles and Multi-Topology Routing in Highly Heterogeneous Ad Hoc Networks," INFOCOM 2006, Poster and Demo session, pp. 23-29
- [14] M. Hauge, J. Andersson, M. A. Brose, and J. Sander, "Multi-topology routing for improved network resource utilization in mobile tactical networks," IEEE MILCOM, 2010.
- [15] Y.-C. Hu, A. Perrig and D.B. Johnson, Packet leases: a defense against wormhole attacks in wireless ad hoc networks, in: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003) (April 2003) pp. 1976–1986.
- [16] Felix Charles Joseph, Amitabha Das, Bu-Sung Lee, Boon-Chong Seet, "CARRADS: Cross layer based adaptive real-time routing attack detection system for MANETS," Computer Networks: The International Journal of Computer and Telecommunications Networking, v.54 n.7, p.1126-1141, May, 2010 [doi>10.1016/j.comnet.2009.10.012]
- [17] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," IEEE/ACM Transactions on Networking (TON), vol. 17, no. 2, pp. 473–486, 2009.
- [18] Kvalbein, Amund, Fossellie Hansen, Audun, Cicic, Tarik, Gjessing, Stein, and Lysne, Olav., "Fast IP network recovery using multiple routing configurations," in Proceedings of INFOCOM, Apr. 2006. 1 - 11. 10.1109/INFOCOM.2006.227.
- [19] S. Lu, L. Li, K. Lam, L. Jia, "SAODV: A MANET Routing Protocol that can Withstand Black Hole Attack", in 2009 International Conference on Computational Intelligence and Security D. Bodeau, R. Graubart, "Cyber Resiliency Design Principles", MITRE Technical Report, January 2017.
- [20] M. Menth and R. Martin, "Network resilience through multi-topology routing," in The 5th International Workshop on Design of Reliable Communication Networks, 2005, pp. 271–277.
- [21] E. Panaousis, G. Drew, G. Millar, T. Ramrekha, C. Politis, "A Test-Bed Implementation for Securing OLSR In Mobile Ad-Hoc Networks", in International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.4, October 2010
- [22] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault, "Multi-topology (MT) routing in OSPF," RFC 4915, June 2007.
- [23] M. C. Scheffel, C. G. Gruber, T. Schwabe, and R. G. Prinz, "Optimal multi-topology routing for IP resilience," AEU - International Journal of Electronics and Communications, vol. 60, no. 1, pp. 35 – 39, 2006.
- [24] Shurman, Mohammad & Yoo, Seong-Moo & Park, Seungjin. (2004). Black hole attack in mobile Ad Hoc networks. Proceedings of the 42nd Annual Southeast Regional Conference, 2004, April 2-3, 2004
- [25] U.S. Naval Research Laboratory, "Extended Mobile Ad-hoc Network Emulator (EMANE)," <https://www.nrl.navy.mil/itd/ncs/products/emane>
- [26] Wang, Feiyi, Wu, "On the vulnerabilities and protection of OSPF routing protocol" in Computer Communications and Networks IEEE, 1998. (LSA)
- [27] Wang, Weichao & Bhargava, Bharat & Mark, Linderman. (2009). Defending against Collaborative Packet Drop Attacks on MANETs.
- [28] T. Clausen, C. Dearlove, J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", IEEE RFC6130, April 2011
- [29] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC3626, October, 2003.
- [30] Ad hoc On-Demand Distance Vector (AODV) Routing C. Perkins, E. Belding-Royer, S. Das [ July 2003 ] (TXT = 90356) (Status: EXPERIMENTAL) (Stream: IETF, Area: rtg, WG: manet) (DOI: 10.17487/RFC3561)
- [31] Perkins Charles E., Bhagwat Pravin, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," London England UK, SIGCOMM 94-8/94.
- [32] J.P. Macker ; J. Dean ; W. Chao, "Simplified multicast forwarding in mobile ad hoc networks," IEEE MILCOM 2004. Military Communications Conference, October 2004.
- [33] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification," IEEE RFC4601, August, 2006.
- [34] E. Baccelli, P. Jacquet, D. Nguyen, T. Clausen, "OSPF Multipoint Relay (MPR) Extension for Ad Hoc Networks," IEEE RFC 5449, February, 2009.
- [35] C. Dearlove, T. Clausen, "Optimized Link State Routing Protocol Version 2 (OLSRv2) and MANET Neighborhood Discovery Protocol (NHDP) Extension TLVs," IEEE RFC 7188, April, 2014.
- [36] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", IEEE RFC 3626, October 2003,
- [37] Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", IEEE RFC 5614, August 2009.
- [38] A. Adams, J. Nicholas, W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", IEEE RFC 3973, January 2005.
- [39] Malkin, G., "RIP Version 2", STD 56, IEEE RFC 2453, November 1998.