Surviving attacks in challenged networks

Jordi Cucurull, Mikael Asplund, Simin Nadjm-Tehrani, Tiziano Santoro

Abstract—In the event of a disaster, telecommunication infrastructures can be severely damaged or overloaded. Hastily formed networks can provide communication services in an ad hoc manner. These networks are challenging due to the chaotic context where intermittent connection is the norm and the identity and number of participants cannot be assumed. In such environments malicious actors may try to disrupt the communications to create more chaos for their own benefit.

This paper proposes a general security framework for monitoring and reacting to disruptive attacks. It includes a collection of functions to detect anomalies, diagnose them, and perform mitigation. The measures are deployed in each node in a fully distributed fashion, but their collective impact is a significant resilience to attacks, so that the actors can disseminate information under adverse conditions. The approach has been evaluated in the context of a simulated disaster area network with a manycast dissemination protocol, Random Walk Gossip, with a store-and-forward mechanism. A challenging threat model where adversaries may attempt to reduce message dissemination or drain network resources without spending much of their own energy has been adopted.

Index Terms—Delay-tolerant networks, manycast, intrusion tolerance, survivability

1 Introduction

Experience tells us that when natural disasters wipe out or severely overload the existing telecommunication infrastructures, the possible means of communication are few and expensive, thereby creating a role for hastily formed networks (HFN) [1]. In those cases where Internet has been prevalent and survived, it serves as a fantastic means of creating situational awareness. However, where there is little infrastructure-based connectivity we are faced with pockets of local connectivity. We believe that there is a potential in tapping into the massive access to handheld devices as a means of establishing ad-hoc communication over wifi links for certain mass dissemination purposes. In these circumstances, there is little room for establishment of mutual trust infrastructures. Any dissemination protocol destined for such environments requires to function in a chaotic context where the node identifiers or even the number of involved nodes cannot be assumed.

The physical aspects of above networks can be characterised by intermittent connectivity, leading to networks in which existence of partitions is a norm. This creates a variation of mobile ad-hoc networks (MANET) with no contemporaneous routes among the nodes, also referred to as intermittently connected MANET (IC-MANET). These networks are complemented with specific routing and dissemination protocols based on store-and-forward approaches [2], [3] that store messages in local buffers of the nodes during periods of no network connection. Experience from the Katrina HFN [4] shows that even in disaster environments there are security threats – actors who try to disrupt operations or use the information for own benefit. However, the application of security measures in these environments is far from trivial.

A node whose aim is to disseminate a message to as many new nodes as possible faces a number of challenges. First, if it tries to deduce the state of communication availability (and the potential threats) by observing its vicinity it will have a local and very restricted view of the world. Due to the nature of IC-MANET, even in absence of attacks, the observed traffic is not easily characterisable (due to mobility and load changes). Second, it is difficult to have a model of an attacker both in space and time. Third, the dissemination protocols, including one that we use for illustration of ideas in this paper, have a tendency that they can spread the impact of an attack in space and in time. Thereby local observation by a node in its vicinity is not necessarily indicative of what is going on around it or right now.

This paper addresses the above challenges by proposing a framework for monitoring adverse conditions. It includes a collection of functions that can be used to detect anomalies, diagnose them (if earlier knowledge for classification exists) and to perform mitigation actions. The measures are deployed in each node in a fully distributed fashion, but their collective impact is that the life of the network is prolonged, so that it can pursue its dissemination goals under adverse conditions.

As an example of a dissemination protocol, we use Random Walk Gossip (RWG) that is designed to run in IC-MANET in a disaster area network [2]. This manycast algorithm tries to disseminate important messages to any k receivers, not relying on knowledge about the network nodes. Its main characteristics are that it exploits opportunistic contacts and uses a store-and-forward mechanism to overcome network partitions. Moreover, to act in an energy-efficient manner it will try to avoid excessive transmissions by decentralised control of number of custodians for a given message.

Our approach is to model the attacker as general as possible. As opposed to the cases where the attacker can be identified by acting in a different way from the

Department of Computer and Information Science, Linköping University SE-581 83 Linköping, Sweden
simin.nadjm-tehrani@liu.se

rest of the nodes, we consider that attackers' behaviours resemble the rest of the nodes in the network, thus almost indistinguishable by pattern matching. We further assume that the adversary too needs to be efficient in its use of energy and bandwidth resources. The adversary may not act normally, but seen from a restricted view, being the local knowledge at each mobile node, it follows fragments of the protocol specification.

Thus, the threat model that we adopt is an adversary that tries (1) to drain the network resources, both at node level (battery life) and network level (available bandwidth), thereby reducing the dissemination flows, or (2) acts as an absorbing patch which reduces some message dissemination in its vicinity, acting as a grey hole at certain times and locations.

The framework that we propose has independent elements that fulfil different functions: detection of anomaly, diagnosis of known attacks if information is available, and response to the known/unknown attack by enabling a mitigation in the own node. The system would work with a subset of the functions, i.e. with/without diagnosis (a kind of misuse detection), and with/without anomaly detection. However, there is no impact on the network behaviour unless *some* mitigation is available. We show extensive evaluations of each of the functions, and when all the functions work together.

Our approach is evaluated in a simulation setting where an implementation of RWG is running with a disaster mobility model adapted from earlier work [5]. The evaluations indicate that the approach indeed creates a resistance to attacks, and show that the network recovers once the attack is over. Moreover, our approach reduces the impact of the attack to a significant extent, i.e. when the attack is overloading the network we reduce the impact by 40-90%, and when the attack reduces the delivery ratio we reduce the impact by at least 40%.

The contributions of the paper are as follows:

- A framework for survivability of information dissemination in presence of scarce resources in IC-MANET scenarios. In particular, characterisation of the functions needed, and implementation of instances of these with a goal to demonstrate survivable communication despite paucity or uncertainty in information, and adverse conditions – both in network connectivity and adversary presence.
- An anomaly detection component based on statistical methods, together with a methodology for how to set the parameters of the component in a plausible context. This includes also a clarification of relevant metrics for evaluation of an anomaly detection function in challenged networks with adversaries, and the role of the classic metrics.
- A diagnosis component that can be trained to recognise known attacks when fed with evidence from network behaviour collected locally.
- Effective mitigation mechanisms that are based on observations from the vicinity and that impact the behaviour of a network in the vicinity of the mitigating node only – thus, being able to deal with uncertainties and inaccuracies of detection.

The survivability framework includes also an adaptation function intended to act as a mechanism to make the overall security approach self-organising. The detailed evaluation of this function is not included in this paper. However, we believe that the methodologies proposed for how to configure each of the other three elements in the framework pave the way for intelligent adaptation leading to higher survivability in changing environments. This is currently explored in ongoing work.

2 Related Work

The proposed framework is devoted to detection, diagnosis and reaction to security threats by using intrusion detection approaches. Neither trust [6] establishment, which is not easy to achieve in HFNs [1], nor cryptographic techniques [7], [8], which are computationally and energy-wise expensive, are required. Several approaches [9], [10], [11], [12] exist for regular MANET, which present many common challenges [13] to IC-MANETs. However, to our knowledge no earlier works present a holistic approach that improves protocol performance in a challenged network in presence of both intermittent connectivity and adversary attacks.

The anomaly detection component detects deviations from the normality of each node from its own observations [14]. The most popular techniques in anomaly detection are classified by Garcia-Teodoro *et al.* [15] into statistical based, knowledge based, and machine learning based. In MANET, depending on the location and interaction of the detectors Xenakis *et al.* [16] classify them as standalone, cooperative, and hierarchical.

In IC-MANET only a few works that address intrusion detection exist. Chuah *et al.* [17] proposes a rule-based misuse detection mechanism targeted towards delay-tolerant networks. It builds on a history-based routing protocol, identifies attacker nodes, and mitigates their effects by keeping them on a black list, i.e. through isolation. In the same context, Ren *et al.* [18] leverage physical properties of the radio transmission to detect wormhole attacks (without any mitigation mechanism).

While we are aware of the general limitations of anomaly detection, such as difficulty of distinguishing good data from bad data, and impact of false positives in a scalable manner, as described in earlier works [19], [20], we believe that detecting deviations from a perceived normality is the only achievable objective in challenged networks. Actors in disaster area networks have a possibility to perform training exercises and train their normality models in terms of levels of dissemination envisaged. However, a new situation will include unforeseen elements and having a perfect detection is inherently not possible. Thus, we aim to show that good results can be achieved despite the presence of detection errors. In particular, we show that given a reasonable accuracy in detection, when combined with existing diagnosis

and mitigation knowledge, the system performance is indeed superior to the undefended case during periods of attacks, and the regime does not cause any significant negative effects during periods without attacks.

The diagnosis of anomalies in MANETs has not been a priority. Few exceptions exist where misuse detection, which is somewhat related to diagnosis, is used to detect more than one attack. Vigna *et al.* [21] propose a misuse detection approach based on the characterisation of the sequences of actions that attackers perform to compromise the AODV routing protocol. Razak *et al.* [22] propose a collaborative intrusion detector where misuse and anomaly detection are both used for a better detection accuracy. Şen *et al.* [23] propose the use of genetic programming for evolving programs to detect each particular attack. None of the above works evaluates the diagnostic performance or uses it to take a specific action for each attack. Furthermore, our work is in the context of IC-MANET, a special instance of challenged networks.

Some works [17], [22], [24], [25], [26] propose certain actions in the presence of attacks, but only a few evaluate the consequence of mitigation [17], [26]. Furthermore, a common response consists of the isolation of the nodes that are suspected to be attackers [17], [26], [27]. In challenged networks since accurate identification of attackers is hard and false positives are prevalent, an isolation policy can create worse effects than the attack itself. A number of works [26], [28] evaluate the impact of the attacks and the impact of the isolation to intelligently decide when it is worth applying isolation.

Countermeasures based on actions different from node isolation are very few. They are typically preventive mechanisms. For example, Li *et al.* [29] propose an encounter ticket mechanism for probabilistic routing protocols to defeat black hole attacks. The approach is effective, but requires a trusted third party and resource-demanding cryptographic methods. Solis *et al.* [30] propose a technique to reduce the effects of "resource hogs" by establishing different priority classes for different domains. The approach requires node authentication and domain node membership management and does not cope with attackers that belong to the trusted domain.

The evaluation of intrusion detectors is usually based on the accounting for detection rate and false positive rate typically presented as ROC curves. In most cases these metrics reflect the number of attacks detected, but sometimes they show the number of attackers detected [17]. The detection delay is not usually taken into account, but there are a few exceptions [24], [31], [17]. There are approaches that quantify the impact of the detectors, such as network overhead or the CPU speed-up [32], or data delivery ratio to evaluate the impact of attack response [17]. Delivery ratio and total transmissions (as an indicator of overhead) are also chosen as main metrics for evaluation of our system. We use available detection rates and false positive rates from earlier scenarios or off-line studies as an input to the calibration of our algorithms only.

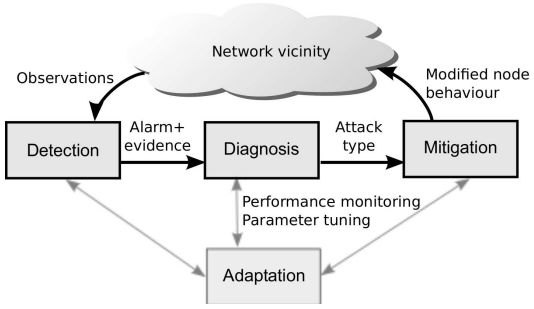

Fig. 1. General Survivability Framework (GSF)

This paper is a substantially extended version of an earlier paper [33]. That work included a preliminary version of the anomaly detector, no overall survivability framework and methodology, and only generic mitigation. The current paper includes a diagnosis component and targeted mitigations for different classified attacks.

3 A Framework for Surviving Attacks

We will now proceed to introduce our General Survivability Framework (GSF) for survivability in IC-MANETS. It is aim is to help the system to survive attacks in extremely challenged environments due to unreliable and intermittent connectivity, lack of trust between nodes, and unknown network characteristics.

Fig. 1 shows the framework consisting of four independent components: detection, diagnosis, mitigation and adaptation to be run in every node in the network. We argue that each of these boxes can be developed and optimised separately and then be combined to form an effective whole. The main loop (illustrated with bold arrows) shows how the characteristics of the network are observed by the system, anomalous behaviour is flagged by the detector, classified by the diagnosis as a possible attack, and how the mitigation changes the behaviour of the node to account for this fact. The adaptation box is intended to monitor the other three boxes and tune their parameters to cope with changes of the network.

The detector raise an alarm if the current traffic deviates from what it considers to be normal. This alarm needs to be coupled with the collected evidence that led to raising the alarm, and fed to the diagnosis box. Obviously, the detection component needs a model of normality to detect deviations from it. Moreover, in a heterogeneous network, with intermittent connectivity, this normality model must be specific for each particular node, since the characteristics of the network will vary considerably depending on the node's location. The information coming from the network must also be filtered and aggregated as it is computationally infeasible to analyse all data which is going through the node.

When an alarm is fed to the diagnosis box together with the collected evidence it should classify it according to previously known attacks. Thus, the diagnosis component needs to have a model for each previously encountered attack. If the attack is new, the alarm cannot be

classified, and the diagnosis should output the attack is unknown, which might still be useful for the mitigation component to enable some generic mitigation.

The mitigation component can decide to change the behaviour of the node to minimise the effects of the attack based on the information given by the diagnosis component. Since this information might be inaccurate and incomplete, the mitigation component must take this into account in deciding *when* to enable or disable a given mitigation scheme. Note that since the mitigation changes the behaviour of the node, this will result in a changed characterisation of the local network. This means that the detection and diagnosis elements need to cope with the effects of mitigations.

The adaptation box will continuously monitor the performance of the other three boxes and has the ability to change parameters and provide training data to account for changing dynamics of the network. In this paper we have focused on the first three boxes and have used a systematic but manual version of the adaptation. Elsewhere, Raciti et al. [34] describe a scheme for adaptation with respect to available energy in the handset.

4 Protocol description and threat model

This section will provide the background on the protocol, and the assumptions on attack scenarios that will be used in later sections.

4.1 Protocol description

The Random Walk Gossip (RWG) is a message dissemination protocol for intermittently connected networks that has been presented earlier [2]. Here we will provide just the information needed to understand the threat model that we have used.

The protocol is designed to cope with the challenges faced in disaster area networks including scarcity of bandwidth and energy, as well as unknown and unpredictable network topologies with partitions. RWG is a manycast protocol, which means that a message is intended to reach a given number k of nodes. When k nodes have been reached, the message is k-delivered and does not need to be propagated anymore, thus not wasting energy. RWG is based on a store-and-forward mechanism, i.e. each node keeps the messages to forward in a local buffer until they have been delivered. This mechanism prevents the loss of messages because of network partitions.

When a message is sent in a connected part of the network, it performs a random walk over the nodes, until all the nodes in the partition are informed of this message. This is controlled by a three-way packet exchange shown in Fig. 2. First a Request to Forward (REQF), that includes the message payload, is sent by the current custodian of the message (grey node in the picture). The neighbouring nodes that hear the REQF reply with an acknowledgement packet (ACK). The custodian chooses one of them randomly and sends an OK to Forward (OKTF) to this node indicating that it will be the next custodian. The other nodes retain the message without actively disseminating it. They keep the message as *inactive* until it expires. Partitions can be overcome by the movement of nodes. Thus, new uninformed nodes will be informed by some node that keeps the message as *inactive* and restarts to disseminate. This process will continue as long as no more uninformed nodes remain in the network or the message is k-delivered.

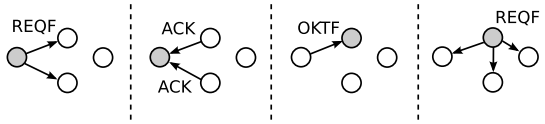

Fig. 2. Random Walk Gossip

All the packet types share the same header structure. In order to keep track of which nodes have seen a given message, each packet header contains a bit vector called the *informed* vector (implemented as a Bloom filter). When a node receives the message it produces a hash of its own address and puts a 1 in the bit vector in the field corresponding to the hash. This allows the protocol to know when a message is k-delivered, and to tell the potential future recipients of the message how far the message has reached towards its dissemination goal (summing the number of 1's indicates the current known local knowledge on this). The *informed* vector also indicates which nodes have received the message. If a node A hears a new neighbour B, then A will go through the messages stored in its buffer to see if B has not yet been informed of any of the messages, in which case those messages will be reactivated and broadcast to node B (and other uninformed nodes in the vicinity).

Finally, when a node realises that a message is k-delivered it sends a Be Silent (BS) packet to its vicinity. This packet will cause all receiving nodes to also realise that the message is k-delivered and thus remove it from their buffers. No new BS packets are sent upon the reception of a BS packet.

4.2 Threat model

Routing and dissemination protocols for MANET are usually based on cooperation of the network nodes and assume fair play. RWG is not an exception and an attacker can take advantage of it. There are many ways a malicious node can attempt to disrupt the dissemination activity in a network. This paper focuses on the mitigation of low-cost attacks consistent with the protocol specification. We study how a disrupting node try to impact the dissemination and resource drain the network without investing too much of its own energy resources.

We assume that the adversaries will have a complete knowledge of the protocol and that will act according to the RWG specifications. For example, by changing the

contents of the informed vector in the header, an attacker can fool the other nodes into believing a message has been delivered to more nodes than is actually the case.

5 Detection, diagnosis and mitigation

This section describes our instantiation of GSF described in Section 3. We have implemented the Detection, Diagnosis, and Mitigation components (see Fig. 1).

5.1 Detection

The detection component implemented is a statistical anomaly detector. It will raise an alarm if a given set of observations deviate too much from what is considered normal. The statistical approach has a small footprint in the resource-constrained context of mobile devices.

Like many other anomaly detection techniques our approach requires a learning phase in order to amass a model of normality. Obtaining traces of a running network with no attacks may be considered as unrealistic. However, we believe that networks can be deployed in learning environments, e.g. exercises prior to live testing, and this is a good start for accumulating preliminary normality models. The normality during deployment might be different to the learnt normality, but our approach does not rely on "perfect" normality, and it will be able to contribute to higher survivability even in presence of inaccuracies (see Section 7).

5.1.1 Detection logic

The detector represents the state of the network as perceived by a node i at a given point of time with a status vector x_i of numerical values for selected features. The basic idea of the algorithm is to calculate the Euclidean distance $D(\mathbf{x}_i(t))$ between a given observation $\mathbf{x}_i(t)$ and the normality model local to a node. The distance is compared with a node-specific threshold T_i^d (generating an alert if $D(\mathbf{x}_i(t)) > T_i^d$). This threshold is part of the normality model of the node, and specifies how far from the average a normal observation can be.

Statistical anomaly detection requires a certain time to detect an anomaly within the system. As alerts cannot be mapped to the specific packets causing the attacks, the alarms must be raised after an interval of suspicion. This is the reason why the alerts raised by the detector are processed and aggregated during the interval I_a of aggregation. In each of these periods the number of packets evaluated and the number of alerts registered are counted. Then an alarm is raised if the number of alerts within that period exceeds a certain threshold (T^a). The threshold is a tunable parameter of the system which is defined in terms of proportion of alerts registered over the number of packets evaluated during I_a.

5.1.2 Training

We now proceed to explain how the normality model of the system is automatically generated by training the system. The model is composed of three elements: the average feature vector ($\bar{\mathbf{x}}_i$), the distance threshold T_i^d, and two vectors ($\mathbf{x}_i^H, \mathbf{x}_i^L$) representing the maximum and minimum values observed for each feature.

Calculation of normality vectors: During a period of time with a set N of observations, the average ($\bar{\mathbf{x}}_i$), maximum ($\mathbf{x}_i^H$), and minimum ($\mathbf{x}_i^L$) vectors are calculated. The minimum and maximum vectors are simply the extreme values recorded for each feature in the vector during that period of time. $\mathbf{x}_i^H$ and $\mathbf{x}_i^L$ are used for normalisation, i.e. to equalise the magnitude of the different features in the vector. Given a vector $\mathbf{v}$ at node i, the normalised vector $\mathbf{v_n} = (\mathbf{v} - \mathbf{x}_i^L)/(\mathbf{x}_i^H - \mathbf{x}_i^L)$.

Calculation of the threshold: After calculating the normality vectors, the threshold (T_i^d) is determined by characterising the distribution of the distances $D(\mathbf{x}_i(t))$ given a set of M different observations. The idea is to set the threshold using the three-sigma rule [35] so that only a small tail of the distribution falls outside the threshold. First the mean distance (μ_i) and standard deviation (σ_i) of this distribution are determined. Then the threshold $T_i^d = \mu_i + 3\sigma_i$ is defined as the mean of the distances plus three times their standard deviation. For a normal distribution the three-sigma rule states that the range $[\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$ covers 99.7% of the observations. Since the distribution of the observed distances is close to the normal distribution we can expect that the three-sigma rule will cover most of the normal observations.

5.1.3 Features

The features of an anomaly detector are the variables which are believed to characterise the behaviour of the monitored system. In our case this consists of characterising the RWG protocol behaviour at routing layer and some notion of normality in the network that a node is operating. Hence, we use a number of derivatives from the four packet types that exist in the RWG protocol (REQF, OKTF, ACK and BS) to capture the protocol operation over time, and a few features that describe typical scenario dynamics, e.g. dynamics of spreading. These features are at the routing layer and are mostly based on statistical measurements, thus not geared to a particular attack.

- **Packet rates**: Number of packets of each type received during the last I_1 seconds. There are four of these features, one for each packet type.
- **Packet distances**: Distance, measured in number of packets received, between the reception of two specific types of packets. E.g., number of packets received between the reception of a REQF and the next ACK. There are sixteen of these features that cover all the possible packet type combinations.
- **Packet rate differences**: Relative difference in the packet rates calculated for each type of packet. There are six features, one for each relevant combination.
- **Number of different source addresses**: Number of different source addresses counted in the packets received during the last I_2 seconds.

- **Packet ratios**: Quotient of the number of packets received of a specific type compared to another packet type among the last I_3 packets. Three features are used: ACK/REQF, ACK/OKTF, ACK/BS.
- **Summation of informed vectors**: Summation of all the positions of the *informed* vectors received in the last I_4 packets.

The features that provide information about the last packets received are implemented as sliding windows over the intervals I_1, I_2, I_3, and I_4. In Section 6.5 we come back to how these intervals are selected based on analysis from the training phase.

5.2 Diagnosis

When an alarm is raised by the anomaly detector, the diagnosis component is engaged, in order to identify the nature of the attack. The diagnosis is based on a geometric interpretation of the features that describe the status of the node at a given time. This assumes that the effects of a particular attack in the m-dimensional space are always of the same nature, irrespective of the location of the nodes and the conditions of the network.

The method applied to diagnose the attack is based on the status vector $\mathbf{x}_i(t)$ and the average feature vector $\bar{\mathbf{x}}_i$ provided as *evidence* by the anomaly detector along with the alarm. The component calculates a unit length vector $\hat{\mathbf{d}}_i(t)$, for node i, which is the normalised difference between these two vectors (see Eq. 1 and 2). The diagnosis is done by matching the attack vector with the smallest angle to the evidence vector $\hat{\mathbf{d}}_i(t)$.

$$\mathbf{d}_i(t) \equiv \mathbf{x}_i(t) - \bar{\mathbf{x}}_i \tag{1}$$

$$\hat{\mathbf{d}}_i(t) \equiv \mathbf{d}_i(t)/\|\mathbf{d}_i(t)\| \tag{2}$$

All the possible attacks cannot be characterised during the training of the system (some of them might not be known). Therefore, the diagnoser may return an output indicating an attack is not modelled and thus unknown.

5.2.1 Attack model generation

The attack model is composed of a number of vectors called *exemplar vectors* that represent the effect that a particular attack will have on the different features on the status vector. For instance, if an attack is associated with a sharp increase of a specific feature, this will be the only non-null component in the exemplar vector.

An exemplar vector for a particular attack is calculated by running a simulation or real exercise in which an (emulated) attack is applied. All the observed differences across the network $\mathbf{d}_i(t)$ where the status vectors were classified as anomalous are averaged and normalised to form the exemplar vector $\hat{\mathbf{e}}_j$, where j is the associated attack's status. The resulting model is a matrix $\mathbf{E} = \begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 & \cdots & \hat{\mathbf{e}}_k \end{bmatrix}$, with k columns. Note that an attack can be characterised by more than one exemplar vector. For example, when an attack is mitigated, but it is still carried out by the attacker, the state of the system is neither similar to the normal state nor to the state of the unmitigated attack, hence a separate exemplar vector can be devised to represent this case.

In order to catch non-modelled attacks, a threshold ϕ_j is determined for each exemplar vector $\hat{\mathbf{e}}_j$. The idea with this threshold is to determine the degree of (angular) closeness of a matching with an attack vector in order to classify a given state as an attack. The threshold is determined using the following methodology. First, all the observations used to create $\hat{\mathbf{e}}_j$ are projected against the vector. The distribution of the projections is studied and the threshold ϕ_j is chosen as the range between ϕ_j (< 1) and 1 that contains most of the projections.

The proposed implementation of the diagnosis component, based on the difference from the normal behaviour allows us to use the same exemplar vectors over the entire network's feature space, without requiring an additional, specific training for every node. The effect of attacks is considered to be approximately uniform regardless of the normality model generated for a node.

5.2.2 Run-time evaluation

At deployment time, for each interval I_a in which the detector raises an alarm, the observations considered anomalous are provided to the diagnoser as attack's evidence. Each observation is diagnosed and the attack type associated with the largest number of observations for this interval is selected as output of the diagnoser.

For each observation considered anomalous, the difference vector $\hat{\mathbf{d}}_i(t)$ is evaluated against the exemplars for the known attacks. The exemplar that most closely resembles $\hat{\mathbf{d}}_i(t)$ is chosen as indicator of the possible attack. The similarity is determined in terms of the angular distance between $\hat{\mathbf{d}}_i(t)$ and the exemplar vector.

The dot product between $\hat{\mathbf{d}}_i(t)$ and the transposed matrix ($\mathbf{E}^{\mathrm{T}}$) of exemplar vectors is calculated, giving as a result a *projection vector* $\theta_i(t)$:

$$\mathbf{E}^{\mathrm{T}} \cdot \hat{\mathbf{d}}_{\mathbf{i}}(t) = \theta_i(t)$$

These vector's components are the scalar projections of the observation along the direction of each of the k exemplar attack models. A higher projection value for a given attack model means that the observation resembles that attack most closely. The dot product between two vectors can be geometrically interpreted as the scalar projection of one vector on the other. Since we have unit length vectors this means that the projection will be:

1	if the two vectors point to the same direction
0	if the two vectors are orthogonal
-1	if the two vectors point to opposite directions

Let $J_i(t)$ be the attack whose exemplar vector has the highest projection value $\theta_{i,j}(t)$ at node i during observation t: $J_i(t) = j : \max_{1 \le j \le k} (\theta_{i,j}(t))$. In order to identify a possible non modelled attack, after selecting an exemplar vector $\hat{\mathbf{e}}_j$, the projection $\theta_{i,j}(t)$ is evaluated against the threshold ϕ_j. If $\theta_{i,j}(t) \ge \phi_j$ the output is $J_i(t)$, otherwise is unknown.

Finally, all the observation diagnostics in the interval I_a are aggregated and the attack type with the largest number of observations is provided to the mitigation component. If the largest number of observations corresponds to the unknown case, this is notified as well.

5.3 Mitigation

The mitigation component receives the results from the diagnosis component. With this information the component selects a suitable action as a response to the suspected attack. The component includes a number of mitigation actuators and a mitigation manager. The mitigation manager decides the type of mitigation to apply and when to apply it. If the attack is classified as unknown a generic mitigation may be applied.

5.3.1 Mitigation actuators

The mitigation actuators are the actual actions applied in response to the attacks. In the current implementation there are three different mitigation actuators, one for each known attack, which are described in more detail in Section 6.5.2. Here it suffices to say that all the mitigation types are of the nature that affect the own node's behaviour. They do not identify a given attacker or try to affect the attacker node's impact. The latter would need extra communication and attackers are very hard to precisely identify in challenged networks. Our mitigation mode will often take the detecting node into a *careful* mode which implies it will reduce the network performance; though restricted in time and space.

5.3.2 Mitigation manager

The mitigation manager is responsible for deciding when to enable/disable mitigation. Because of the network dynamics and detection accuracy, the alarms received from the detector are not accurate in time and space, or indeed in value. Thus, there may exist some non-detected attack intervals while an attack is ongoing.

A mitigation management policy is applied to obtain a more stable mitigation. The policy uses the detection rate of the diagnosed attack, which is calculated during the modelling of the attacks, to extend the mitigation during a period ϵ of time after an alarm. We next describe how the duration of the mitigation is computed.

The detection rate can be expressed as $P(D|A_j)$, which is the probability of detection given that an attack j is present. Hence, the probability of no detection is $P(\neg D|A_j) = 1 - P(D|A_j)$. Given a window ω of a finite number of intervals during which the detector evaluations have taken place, the expected number of intervals γ in which attacks are detected is $E[\gamma] = \omega * P(D|A_j)$ and ϵ in which attacks are not detected is $E[\epsilon] = \omega * (1 - P(D|A_j))$. From these equations it is possible to obtain the expected number of non-detections expressed as a function of the expected number of detected intervals:

$$E[\epsilon] = E[\gamma] * \frac{(1 - P(D|A_j))}{P(D|A_j)} \quad (3)$$

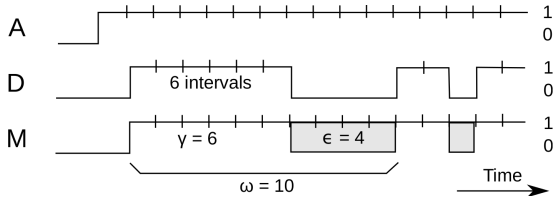

Fig. 3. Example mitigation manager policy (A=attack present, D=detected attack, M=mitigation present)

At run-time, this information can be used to extend the duration of the mitigation actions after the first interval in which no anomalies are detected. Given a number of *observed* consecutive detection intervals γ, the mitigation period will be extended with ϵ intervals of mitigation even if no attack is detected during this time. The rationale for the policy is that the estimated false negative rate should be related to the maximal duration for applying a mitigation when no alarms are present. [1]

An example of the policy is shown in Fig. 3. Assume an estimated detection rate of 60% and 6 consecutive intervals (γ) are detected as anomalous. When an interval without anomaly is detected, an extension of 4 additional intervals (ϵ) with mitigation is calculated, covering the hypothetical 40% undetected remaining intervals.

The adaptive mitigation mechanism above has two benefits: (1) it will not mitigate for unnecessarily long periods, thus avoiding reduced performance, (2) it will mitigate for long enough periods when the attack is persistent. Note that when the latest attack is classified as unknown, the mitigation actions are not extended (i.e. discontinued when the alarm disappears).

6 Evaluation Methodology

This section describes our evaluation framework in the complex setting of challenged networks. While evaluating on real testbeds and attackers would be ideal, we find that simulation platforms create a first proof of concept and are useful due to reproduceability of their results. Thus, GSF will be evaluated using Network Simulator 3.6 (ns-3) running together with a RWG protocol implementation at the network layer.

Our evaluation of the GSF is conducted as follows. First we measure network performance in presence of no attacks. Then we introduce attacker nodes in the simulator and collect similar data. Finally, we activate the GSF and observe its impact on network survivability. The details of the settings of these three scenarios (no attack, no defence, defence) are described in Sections 6.3, 6.4, and 6.5 respectively. The overall evaluation of the GSF will be presented in Section 7.

Before evaluating the GSF as a whole we test each component individually and tune certain parameters. The tuning of the components is described in Section 6.5,

1. The length of the extension period is based on the number of extension intervals which has the same proportion to the number of detected intervals as the corresponding expected entities in Eq. 3.

and the evaluation results for isolated components, as well as GSF as a whole, is presented in Section 7.

6.1 Evaluation setup

Since NS-3 performs packet level simulations, each simulation may take a long time. For example, in the longer scenarios where the whole GSF is evaluated we experience ten hours of real time for 8800 seconds (146 minutes) of simulation. Therefore, in the following sections, we use shorter simulations when we are testing a GSF component in isolation (e.g. 3000 seconds) and longer simulations where the combined GSF is evaluated with several attacks introduced in sequence. In all the cases we rely on a 1400 seconds initial training duration. This duration was found experimentally to suffice for establishing normality models. Within all evaluation intervals, prior to the training interval we used 200 seconds as start-up period to get the dissemination protocol going. All evaluations were done with ten simulation runs per scenario and averages were computed over these runs.

6.2 Performance measurements

For each scenario data has to be collected to establish the impact of attacks and defences. The ground truth in our case can be construed in two ways, a local view observed at one node or a global view observed at network level.

In mobile networks running a dissemination protocol, the concept of being under attack for a particular node is unclear. An attacker can be mobile (its impact varying in space) and the attack lasting over a period can continue to impact other nodes after the duration of the attack (this is specially true for flooding attacks). In networks with intermittent connectivity and store-carry-forward dissemination protocols, a message will typically have a long life time in the network thus accentuating this effect. Just considering an attack interval for measuring the impact on all the nodes is meaningless, since attacks do not impact the network in a well-determined time interval and confined space. Nodes can be isolated from the attackers during some periods or can be too far from the attackers to be significantly affected by them.

When we evaluate individual components in isolation, we use data collected per node as a ground truth, and when we evaluate the GSF as a whole we measure network-wide performance.

Classic metrics, Detection Rate (DR) and False Positive Rate (FPR) [33], are collected as follows. First we determine whether during the period of alarm a node was under attack. The packets sent by an attacker and their responses are tagged. Thereby, a node is considered under attack if at least one of the tagged packets is received during its aggregation interval I_a (see Section 5.1.1).

Given the chaotic nature of challenged networks the success of the approach is not measurable only with these metrics neither on a per node basis nor on a network-wide (average) basis. The locality of the attackers, the nature of the partitions, and the mobility of the nodes, all affect the results so that DR and FPR are not meaningful across time and space. Therefore, we also adopt other metrics that show the global impact of the attacks in terms of network performance:

- **Packet Transmission Rate (PTR)**: Denotes the number of data and signalling packets transmitted during the interval of study. Besides being an indicator of the usage of bandwidth as a resource, PTR is an indicator of the energy spent by the nodes, since the more transmissions the more energy is consumed.
- **K-Delivery Rate (KDR)**: Depending on the network connectivity, the load, and the dynamics, only a proportion of the messages sent are finally k-delivered. Thus, a good metric to evaluate the effects of an attack and its mitigation is the number of messages which are k-delivered over the interval of study.

In the network-wide evaluation we will use the actual message k-delivery rate in the absence of attacks as one baseline in the evaluation. Also, the transmission volume will indicate the attack-induced unnecessary transmissions compared to the no-attack scenario.

6.3 Base scenario

This section describes the settings of RWG and the mobility model (for all the nodes in the network). The disaster area simulations include the mobility traces from Aschenbruck *et al.* [5], based on a large training manoeuvre in preparation of the FIFA world cup in Germany in 2006. The original traces include 150 mobile nodes. To induce partitions and create an intermittently connected network we have selected 25 of the nodes, chosen uniformly over the locations in the area, while maintaining the trace for that node. This creates a similar network with lower density. All the nodes use the 802.11a protocol, at 6Mb/s maximum link capacity with a radio range of 24 meters. The speed of the nodes varied in the range 1-2 m/s in an area of 350m x 200m. The load is generated by introducing a total of 15 messages to disseminate every second from randomly chosen nodes in the network. Each message has set to be delivered to a minimum number of 10 nodes ($k = 10$).

6.4 Attack scenario

In addition to the 25 fair nodes mentioned above, five other nodes are chosen as attackers with a uniform distribution among the fair nodes. The attacker nodes do not create normal traffic (data) in the network, but produce packets that are compatible with the protocol specification as described in Section 4.1 (in accordance to the threat model). In all the cases the adversaries do not participate in the normal operation of the network, but can listen and send packets as any other node.

The attack scenario is created by the injection of one of three attacks that fall into the threat model described in Section 4.2. While an anomaly-detection-based scheme should ideally be tested for numerous attack types, we

confine this paper to three attack types namely: drain, grey hole and flooding attacks. These attacks were chosen due to their disruptive nature of the network's main activity (that has the goal of disseminating information) effectively targeting the routing layer. We believe that our diagnosis component can be configured to learn other attack characteristics too (e.g. sinkhole or variations of the routing attacks). In the attack scenario, we introduce one attack type at a time, in order to manage the complexity of evaluation of the impact of the attacks in time, with a distribution in space as mentioned above.

Drain attack: It causes the nodes around the attacker to transmit more packets than usual in order to drain their batteries and waste bandwidth. The effect, that exploits the RWG node discovery mechanism, is achieved by regularly sending ACK packets with different fake identities. Then the nodes around the attacker retransmit the messages they have in their buffers to forward them to the apparent (but bogus) new node. This attack is performed by the 5 attacking nodes each sending 10 ACK packets/second with different identities.

Grey hole attack: It exploits the propagation of the message delivery status through the bit settings of the Bloom filter in the packet header. It makes the nodes around the attacker believe the messages they disseminate have already reached k nodes resulting in a reduction of the k-delivery ratio. The grey hole attack, aiming to reduce the chances of successful message dissemination, is performed by 5 nodes each one answering to all the REQF packets they receive with forged ACK packets.

Flood attack: It consists of introducing bogus messages to the network with a large group size k and time to live (TTL). The larger the k value, the more nodes are affected, and the longer the TTL, the longer the bogus message will be in the network. The result is a broadcast effect where a number of useless messages are propagated along the whole network, producing high quantities of unnecessary packets. The consequence is a high quantity of bandwidth and energy consumed. The attack is performed by 5 nodes each sending 10 REQF packets/second without a payload. Each of these REQFs originates a number of ACK packets from the neighbours and the attacker sends an additional OKTF packet per REQF. A chain of retransmissions is produced along the network until the message is propagated to all the nodes.

6.5 Defence configuration

Whenever a system is used it must be calibrated for the environment in which it is applied. This section explains how each of the GSF's components have been calibrated for the scenario described. In particular, we describe how the anomaly detector is tuned regarding its internal parameters, how the diagnosis component is instantiated to recognise the 3 attack types that presented above, and which mitigations were devised to act whenever any of those attacks were deemed in operation.

6.5.1 Parameter tuning

Calibrations were performed using simulation runs of 3000 seconds of which the first 200 seconds were discarded due to start-up time. The following 1400 seconds were used for training the system (half of it for calculating $\bar{\mathbf{x}}_i$, $\mathbf{x}_i^H$, and $\mathbf{x}_i^L$, and the rest for the threshold T_i^d).

Detection component: The detector has two main parameters that must be adjusted for a particular system, the alert aggregation interval I_a and the alert aggregation threshold T^a. Both parameters influence the final DR, FPR and detection delay. In a deployment scenario these parameters need to be set based on studies performed in exercise sessions (or using simulations).

Calibration starts by studying the different combinations of the DR and FPR for the detector in presence of the different conceived attacks. Running the detector over the traces of the simulations, I_a has been tested with values between 1 and 100 seconds, and T^a with values between 1% and 40%. The ROC curves shown in Figs. 4a, 4b and 4c depict the DR against the FPR obtained for each attack with the I_a at 10 and 50 seconds (the rest are not shown for brevity). The study confirms that the DR increases as the I_a increases. Also, with a fixed I_a and just varying T^a, the lower the T^a, the higher is the DR and the FPR. Naturally, a longer I_a also leads to a longer detection delay. Based on these studies, the chosen aggregation interval I_a for the evaluation scenarios has been set to 50 seconds to maximise the DR. The threshold T^a has been set to 1%, which corresponds to a FPR around 18%.

The intervals used to calculate the features (I_1, I_2, I_3, and I_4) have been set up to 5 seconds, 10 seconds, 50 packets, and 100 packets, respectively, and follow the same logic as parameter I_a, i.e. the longer they are the more sensitive to system changes, but with longer delay.

Diagnosis component: The diagnosis component can act on independent evidence collected from the network. However, when anomaly detection is enabled the idea is that diagnosis takes place when an alarm is generated by the detector, and uses the evidence for the alarm. For our experiments, which use the two components cooperatively we have used the same features monitored by the detector as the evidence for creating the exemplar vectors and for the diagnosis.

The diagnosis component has two main parts to configure. The exemplar vectors that characterise the impact of given attacks on the network, and the set of thresholds to distinguish unmodelled attacks. Six exemplar vectors have been created, two for each of the attacks defined in Section 6.4. The vectors have been characterised running two simulations for each attack, one without mitigation and another one with the specific mitigation related to the attack enabled. In addition, one threshold ϕ_j for each of the exemplar vectors j has been defined.

The intuition for creating the exemplar vectors with a given attack in parallel with a mitigation in progress is that an attack does not impact the network in the same way with/without the mitigation in progress. Thus,

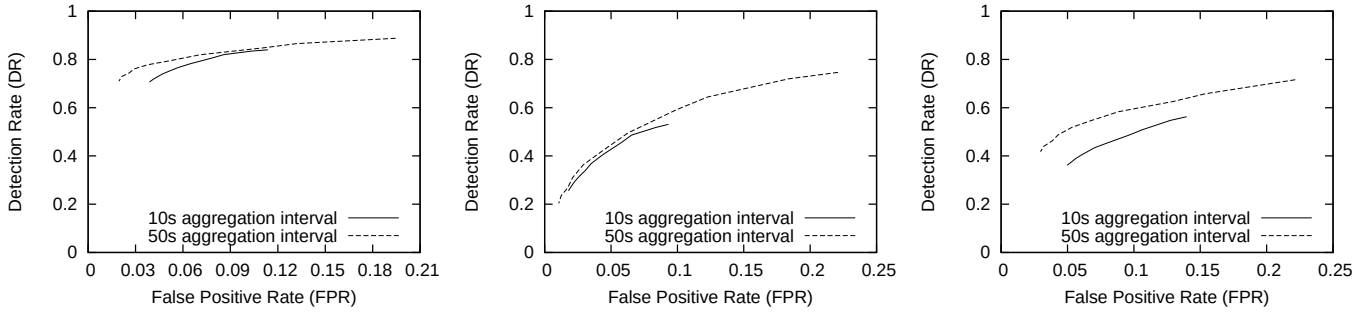

Fig. 4. ROC curves with different aggregation interval (I_a)

adding the knowledge - in the form of observed behaviours during an early exercise/simulation - improves the classification accuracy.

Mitigation component: The mitigation component has been configured with the DRs obtained during the configuration tests of the detector component as described before. This implicitly determines the length of the mitigation for each attack as described in Section 5.3.

6.5.2 Devised mitigations

A set of mechanisms, implemented as mitigation actuators, are proposed to counteract the effects of the attacks described in Section 6.4.

Drain attack mitigation: The action proposed to reduce the impact of this attack consists of ignoring the ACK packets with fake identity. Of course, in the normal operation of the protocol nodes cannot distinguish between good and fake identities. However, there is a chance to recognise fake IDs by storing a list of known nodes during periods when the mitigation is not enabled. This can be done if a list is updated with identities of nodes that have sent REQF packets (thus creating a white list that is relatively accurate, at least as long as the detector is accurate). A slight increase in the latency for recognising new fair nodes can be expected.

Grey hole attack mitigation: This attack targets the RWG mechanism for propagation of delivery status. The mitigation consists of going into a "suspicious mode" where only ACK, OKTF, and BS packets coming from white-listed nodes are accepted and where the update of the delivery information coming from the ACK packets received from any node is restricted (i.e. do not set zeros to ones in the bit vector). More specifically, if an ACK is received from a known node, the local *informed* vectors are just updated for the position that corresponds to the sender of the ACK, but the *informed* vector contained within the ACK packet is ignored. This mitigation imposes a heavy burden on the network resources. The information about the number of deliveries of each message is propagated slower than usual and the message is kept in the network for a longer time than needed.

Flood attack mitigation: It consists of ignoring and removing packets involved in the propagation of messages with large k group sizes and long TTLs. Two histograms, one for each parameter, are created with the packets received during the node training period. They are used to define the limits over which a packet will be associated to the attack. The limit covers a percentage of the leftmost area of the histogram. This percentage has been set to 95% to discard packets with unusually high k group sizes or TTLs received during the training. When the mitigation is enabled two actions are applied: incoming packets exceeding one of the two limits are ignored; and messages stored in the local buffer exceeding the limits are removed. These two actions are applied to deter the highly contagious effects of the attack.

The application of these techniques would significantly reduce the performance of the network (disseminating information) if enabled indefinitely. This is the reason why they are not an integrated part of the RWG protocol specification. The idea of GSF is to apply the appropriate mitigation when needed but not otherwise.

7 Evaluation Results

This section describes evaluation results for the GSF components and their collective impact running with RWG in the disaster area scenario described in Section 6.1. The network created is far from being stable and homogeneous, which is a challenge for identifying deep insights respect to the performance of the survivability framework. The evaluation individually analyses each component (Sections 7.1 to 7.3) and, finally, the cooperation of the three components in the system (Section 7.4).

For the individual components tested in isolation we used the 3000 second simulations, described in Section 6.5.1, the last 1400 seconds of which were used for testing. For the collective scenario an extension of the simulation to 8800 seconds was utilised.

7.1 Detection component

In order to show why the standard metrics DR and FPR do not make sense as indicators in a survivability context we now proceed with a more detailed analysis. The network wide average results obtained in terms of these metrics, as depicted on Fig. 4a, 4b and Fig. 4c, are computed by averaging the results of all 25 anomaly detectors over the entire test interval of 1400 seconds.

TABLE 1
Performance of detection and classification

(a) Detection

# Adversaries per partition	Drain Attack						Grey Hole Attack						Flood Attack					
	Best		Average		Worst		Best		Average		Worst		Best		Average		Worst	
	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR
2	98%	4%	97%	8%	94%	9%	83%	5%	79%	8%	69%	10%	96%	4%	95%	8%	89%	10%
1	99%	4%	97%	7%	94%	16%	61%	2%	48%	6%	25%	13%	49%	4%	39%	7%	30%	13%
0	63%	3%	51%	6%	44%	10%	30%	2%	22%	4%	15%	8%	51%	6%	41%	12%	34%	16%

(b) Classification: isolated

Attack performed	Attack classified			
	Drain	Grey hole	Flood	Unknown
Drain attack	87%	0%	0%	13%
Grey hole attack	0%	87%	0%	13%
Flood attack	0%	0%	94%	6%

(c) Classification: embedded in GSF

Attack performed	Attack classified			
	Drain	Grey hole	Flood	Unknown
Drain attack	71%	12%	8%	9%
Grey hole attack	0%	80%	6%	14%
Flood attack	0%	9%	72%	19%

We noted that in highly partitioned networks with very different conditions it is not fair to analyse the results of the detection mechanism on an aggregate basis using these metrics. We observed that the traffic flow, the type of attack, and the number of attackers in each partition produce very different detection rates. The network topology in our disaster area is composed of eight partitions more or less stable along the whole simulation, with moving nodes acting as ”bridges” over the partitions. A node by node analysis has confirmed that the parameter with more influence over the detection performance is the proximity of the adversaries to the fair nodes. Table 1a shows the best, average and worst DR and FPR, for the drain, grey hole and flood attacks. Results in each column are categorised into different classes. Each class shows the results aggregated for partitions that have similar number of adversaries, i.e. partitions with no adversaries, partitions with 1 adversary, and so on. There are around 1/3 of the fair nodes in each class. The results, calculated with the alert aggregation threshold T^a at 10%, demonstrate that the less the partition is affected by the attacks the worse is the performance of the detection. That is, the classes with zero and one adversary are the ones that reduce the average detection performance. Note that despite having partitions with no adversaries, some attacks are received by sporadic contacts with other partitions. In Section 7.4 we illustrate that the detection errors on their own are not significant in the context of the longer simulations we performed with the three attack types.

7.2 Diagnosis component

This section evaluates the results of the diagnosis component in isolation from the detector. We found experimentally that learning attack vectors is faster compared to the training period required for learning normality. Hence, the diagnosis component was trained with 900 seconds of traces from one node in an environment where the three described attacks were applied in individual runs (only exemplar vectors without mitigation). The test was performed by classifying traces of a node running the three attacks in three different simulations. The results, shown in Table 1b, are the average of ten different simulation runs. This experiment showed a correct classification performance in over 87% of the cases.

Since in practice the diagnoser is integrated in a larger system, and the results of the detector and mitigation components have an impact on the network state, the performance of the diagnosis component was also studied in this context. In this case the evidence the diagnosis uses to classify the attack is dependent on the detector, and the state of the network also depends on the mitigation triggered. Therefore, each attack was additionally characterised with an exemplar vector in presence of mitigation. This gives two exemplar vectors per attack (one prior to mitigation, and one during mitigation). In addition to classification inaccuracies, the inaccuracy of the detector may also result in alarms/evidence for which the diagnosis has no strong match with a known attack, thus classifying the attack as unknown. The results shown in Table 1c demonstrate that, despite inheriting the detector inaccuracies, the diagnosis function implemented has a relatively good accuracy. In Section 7.4 we come back to the impact of the diagnosis misclassifications in the longer test scenario.

7.3 Mitigation component

The mitigation manager, as explained in Section 5.3, enables a specific mitigation actuator upon a given classified attack. Fig. 5 shows the results of two mitigation studies, by depicting the PTR in a 3000 second simulation test in which a drain attack was introduced. The curves show the effects with and without mitigation enabled applying two different intervals of aggregation (namely 10 and 50 seconds respectively) in the framework. As explained in Section 7.1, the higher the I_a the higher is DR. The figure illustrates the impact of the selected interval on the latency of the response. As expected a long interval will delay the detection and thereby postpone the impact of the mitigation.

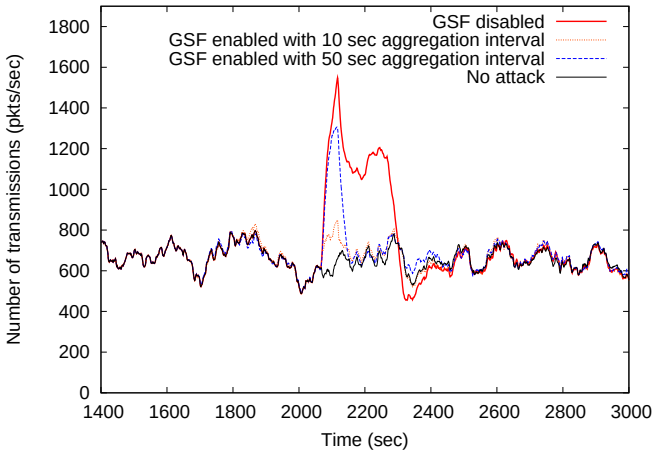

Fig. 5. Drain attack

TABLE 2
Timing characteristics of the simulation

Process	Time period	Process	Time Period
Init	0 - 200 s	Attack 1: Drain	2200 - 3400 s
Train	200 - 1600 s	Attack 2: Grey hole	4300 - 5500 s
Test	1600 - 8800 s	Attack 3: Flood	6400 - 7600 s

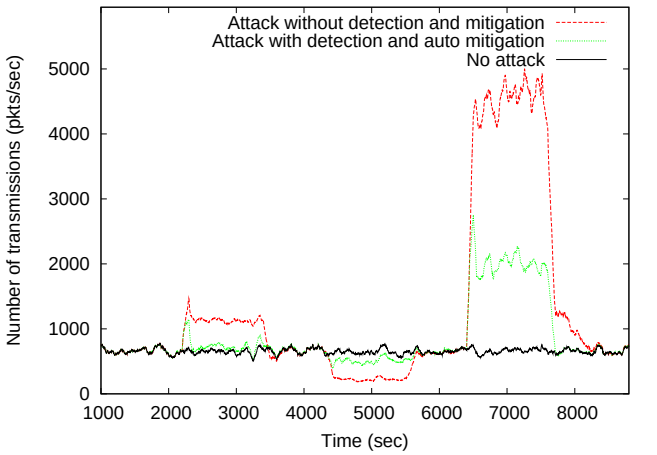

Fig. 6. PTR in simulation with three attacks

7.4 Three cooperating components

The Detection, Diagnosis, and Mitigation components in the survivability framework were then evaluated with a long (8800 second) simulation that includes two hours of testing time. One instance of each attack is included in the scenario, each one during 20 consecutive minutes and with a separation of 15 minutes (see Table 2).

Fig. 6 shows the effects of each of the attacks and the detection and mitigation system in terms of network packet rate (PTR). The drain attack, the first one, sharply increases the PTR by around 100% and then decreases to a steady 70% additional PTR compared to the no attack baseline. The reason is that just at the beginning of the attack there are more inactive messages ready to be forwarded in the buffers of the fair nodes. When the

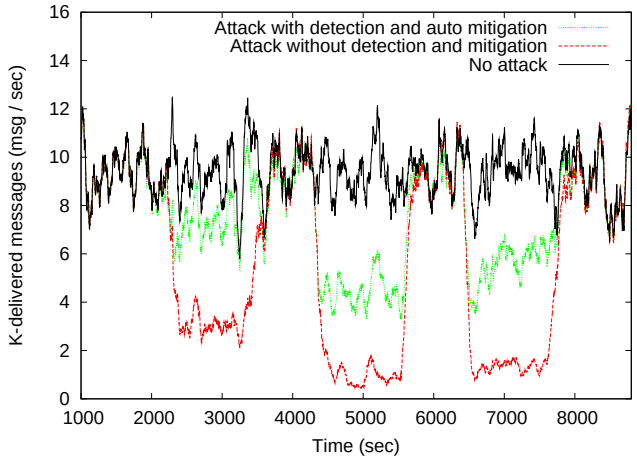

Fig. 7. K-Delivery ratio in simulation with three attacks

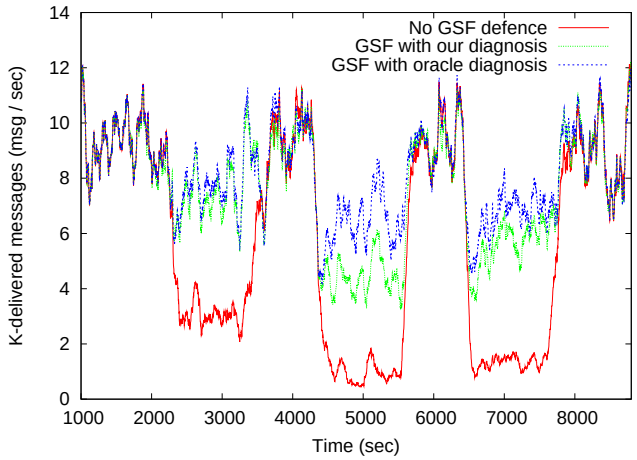

Fig. 8. The three attack scenario with isolation of the impact of diagnosis errors in each case

attack begins, the PTR increases, but as soon as the attack becomes detected in most of the nodes, approximately after 50 seconds, the mitigation actions are taken and the attack impact is almost cleared. The grey hole attack, the second one, has the opposite effect, it reduces the PTR to 35% since its purpose is to stop the dissemination of messages. When the detection and response mechanism is enabled, the PTR is kept around 80% of the normal state. The flood attack, shown as the last one, increases the PTR by a 600% . As the results indicate, the effects of this attack are significant because of its propagation effect. In addition, even after the attack there is a period of 500 seconds during which part of the effects of the attack remain. The detection and mitigation mechanisms reduce the effects of the attack to around 200% over the normal state, i.e. one third of its initial effects and, in addition, they remove the tail effect at the end of it.

Fig. 7 shows the effects of each attack and the detection and mitigation system in terms of k-delivery ratio (KDR). Although only the grey hole attack has as a main purpose to reduce the KDR, all three attacks have an impact

on delivery to some extent. The reason is that the KDR is heavily affected by network adverse conditions, and all the attacks in some way induce them.

The drain attack, the first one, reduces the KDR to 1/3 of the normal ratio. The reason is the overhead produced by the attack, resulting in unnecessary signalling and data packet transmissions that consume resources otherwise used to deliver the rest of the messages. Detection and mitigation reduce the effects of the attack keeping the KDR at 80% of the normal values. The grey hole attack, the second one, drastically drops the KDR to 10%. When the detection and mitigation mechanisms are enabled the KDR is raised by over 5 times, although still slightly under 50% of the normal values. It is worth mentioning that this is a complex attack to mitigate since once the *informed* vector is sent there is a contagious impact on the other partitions whereas the mitigation is not enabled everywhere. The flood attack, the third one, also drops the KDR to similar low levels, since the network bandwidth is consumed by bogus packets. In this case, when the detection and mitigation mechanisms are enabled the KDR increases by over 4 times, and a KDR corresponding to half of the normal one is achieved.

Finally, we show a special scenario created to isolate the effects of the diagnosis errors with the same attack pattern as the long simulation (Table 2). Figure 8 shows the outcome of the test dedicated to diagnosis effects. The scenario consists of the GSF enabled once with the diagnosis engine described in Section 7.2, and once with a perfect diagnoser (an oracle) that exactly identifies an ongoing attack. The significance of the errors can be visualised by the gap between the two curves depicting k-delivery ratio. We see that while the drain attack mitigation is not significantly affected by the diagnosis errors, with the grey hole attack (the second one) the delivery ratio is lower compared to the perfect diagnoser (averaging at 40% compared to 60%), and with the flood attack, the delivery ratio is reduced to 50% on average as opposed to 60%. In all three cases we see GSF creates a significant improvement compared to the no-defence strategy. Similar effects appeared for packet rate, but we ommit the figure because of space restrictions.

Furthermore, Figs. 6 and 7 can be studied with a focus on the impact of the false positives of the detector. They show that despite the presence of false alarms indicated by Table 1a, the GSF as a whole has no or very small impact on the system behaviour during periods without attacks. Overall, the results show that the approach is successful in creating a resistance to the attacks that conform to the given threat model, despite the difficulties that the complexity of IC-MANET bring to the picture.

8 Conclusions

This article has presented a modular framework for attack survivability in IC-MANETS composed of detection, diagnosis, mitigation and adaptation components.

The detector, a standalone statistical-based anomaly detector algorithm, combats the resource constraints in the mobile devices, i.e. CPU power (and implicitly energy) and network bandwidth usage. The diagnosis approach, based on a feature-space geometric interpretation of the normality deviation, allows the use of an identical set of attack signatures for all the nodes irrespective of the environment in which each operates. The mitigation approach, given a non perfect detection and classification input, applies a number of actions to mitigate the suspected attack during a certain period dynamically calculated for a better performance. The use of the framework requires calibration. Extensive guidelines, and a study on calibration have been provided.

The framework has been evaluated with positive results in a hostile environment with different network attacks. Strong resistance to attacks has been demonstrated when the framework is enabled. For example, when the attacker aimed at reducing delivery ratio we recover around half of the lost performance by mitigation.

The classic metrics such as detection rate and false positive rate were shown not to be appropriate to measure the detection performance in highly partitioned networks because they are affected by the node's locality.

This scheme has been implemented on modern smartphones and shown to have a small resource footprint [36]. Future work includes exploration of the adaptive component to dynamically influence the parameters of the components presented in this paper and to show how the system resilience is increased by adding the adaptive element. A starting point has been the adaptation of survivability depending on the energy levels of the device [34]. A further interesting direction to explore is the study of the impact of the mitigation actions, which contribute to a change in the behaviour of the network creating a recursive chain of alarms among the nodes.

Studying distributed attackers in scenarios where only a single attack type is active over any interval of time was only a first step in studying survivability in challenged networks. More work remains in creating resistance to multiple attacks overlapping in time intervals.

Acknowledgements

This work was supported by a grant from the Swedish Civil Contingencies Agency (MSB), supporting the project Hastily Formed Networks [37], and the national Graduate school in computer science (CUGS).

References

[1] P. J. Denning, "Hastily formed networks," *Commun. ACM*, vol. 49, no. 4, pp. 15–20, 2006.

[2] M. Asplund and S. Nadjm-Tehrani, "A partition-tolerant manycast algorithm for disaster area networks," *IEEE Symposium on Reliable Distributed Systems*, pp. 156–165, 2009.

[3] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing with location service in intermittently connected manets," *IEEE Trans. Veh. Technol.*, 2011.

[4] B. Steckler, B. L. Bradford, and S. Urrea, "Hastily formed networks for complex humanitarian disasters after action report and lessons learned from the naval postgraduate school's response to hurricane katrina," Naval Postgraduate School, Tech. Rep., 2005.

[5] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling mobility in disaster area scenarios," in *Proc. 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM)*, 2007, pp. 4–12.

[6] J. Cho, A. Swami, and I. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, 2010.

[7] P. Prasithsangaree and P. Krishnamurthy, "On a framework for energy-efficient security protocols in wireless networks," *Comp. Commun.*, vol. 27, no. 17, pp. 1716 – 1729, 2004.

[8] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Trans. Mobile Comput.*, vol. 5, no. 2, pp. 128 – 143, 2006.

[9] H. Nakayama, S. Kurosawa, A. Jamalipour, Y. Nemoto, and N. Kato, "A dynamic anomaly detection scheme for AODV-based mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2471–2481, 2009.

[10] Y. Liu, Y. Li, H. Man, and W. Jiang, "A hybrid data mining anomaly detection technique in ad hoc networks," *Int. J. Wirel. and Mob. Comp.*, vol. 2, no. 1, pp. 37–46, 2007.

[11] J. B. Cabrera, C. Gutirrez, and R. K. Mehra, "Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks," *Information Fusion*, vol. 9, no. 1, pp. 96–119, 2008.

[12] E. Gerhards-Padilla, N. Aschenbruck, and P. Martini, "TOGBAD An approach to detect routing attacks in tactical environments," *Security and Commun. Netw.*, 2010.

[13] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE Wireless Commun. Mag.*, vol. 11, no. 1, pp. 38 – 47, 2004.

[14] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41.

[15] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[16] C. Xenakis, C. Panos, and I. Stavrakakis, "A comparative evaluation of intrusion detection architectures for mobile ad hoc networks," *Computers & Security*, vol. 30, no. 1, pp. 63 – 80, 2011.

[17] M. Chuah, P. Yang, and J. Han, "A ferry-based intrusion detection scheme for sparsely connected ad hoc networks," in *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*. IEEE, 2007, pp. 1–8.

[18] Y. Ren, M. C. Chuah, J. Yang, and Y. Chen, "Detecting wormhole attacks in delay-tolerant networks," *IEEE Wireless Commun. Mag.*, 2010.

[19] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy (SP), IEEE Symposium on*, 2010.

[20] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 3.

[21] G. Vigna, S. Gwalani, K. Srinivasan, E. Belding-Royer, and R. Kemmerer, "An intrusion detection tool for AODV-based ad hoc wireless networks," in *Computer Security Applications Conference, 2004. 20th Annual*, 2004.

[22] S. Razak, S. Furnell, N. Clarke, and P. Brooke, "Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1151 – 1167, 2008.

[23] S. Şen, J. A. Clark, and J. E. Tapiador, "Power-aware intrusion detection in mobile ad hoc networks," in *Ad Hoc Networks*, ser. LNCS. Springer, 2010, vol. 28, pp. 224–239.

[24] B. Sun, K. Wu, and U. W. Pooch, "Zone-based intrusion detection for ad hoc networks," *Ad Hoc Sens. Wirl. Netw.*, 2004.

[25] A. Deodhar and R. Gujarathi, "A cluster based intrusion detection system for mobile ad hoc networks," Virginia Polytechnic Institute & State University, Tech. Rep.

[26] S.-H. Wang, C. H. Tseng, K. Levitt, and M. Bishop, "Cost-sensitive intrusion responses for mobile ad hoc networks," in *Recent Advances in Intrusion Detection*, ser. LNCS, vol. 4637. Springer, 2007, pp. 127–145.

[27] E. Ayday, H. Lee, and F. Fekri, "Trust management and adversary detection for delay tolerant networks," in *Military Communications Conference - MILCOM*, 2010.

[28] Z. Zhao, H. Hu, G.-J. Ahn, and R. Wu, "Risk-aware response for mitigating manet routing attacks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[29] F. Li, J. Wu, and A. Srinivasan, "Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets," in *IEEE INFOCOM*, 2009.

[30] J. Solis, N. Asokan, K. Kostiainen, P. Ginzboorg, and J. Ott, "Controlling resource hogs in mobile delay-tolerant networks," *Computer Communications*, vol. 33, no. 1, pp. 2 – 10, 2010.

[31] C. H. Tseng, S.-H. Wang, C. Ko, and K. Levitt, "DEMEM: Distributed evidence-driven message exchange intrusion detection model for MANET," in *Recent Advances in Intrusion Detection*, ser. LNCS, vol. 4219. Springer, 2006, pp. 249–271.

[32] Y.-a. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in *Proc. 1st ACM workshop on Security of ad hoc and sensor networks (SASN)*, 2003, pp. 135–147.

[33] J. Cucurull, M. Asplund, and S. Nadjm-Tehrani, "Anomaly detection and mitigation for disaster area networks," in *Recent Advances in Intrusion Detection*, ser. LNCS, S. Jha, R. Sommer, and C. Kreibich, Eds., vol. 6307. Springer, 2010, pp. 339–359.

[34] M. Raciti, J. Cucurull, and S. Nadjm-Tehrani, "Energy-based adaptation in simulations of survivability of ad hoc communication," in *Wireless Days (WD), IFIP*, 2011.

[35] D. S. Moore and G. P. M. Cabe, *Introduction to the practice of statistics*, 5th ed. W. H. Freeman, 2005.

[36] J. Cucurull, S. Nadjm-Tehrani, and M. Raciti, "Modular anomaly detection for smartphone ad hoc communication," in *16th Nordic Conference on Secure IT Systems, NordSec*, ser. LNCS. Springer.

[37] "Hastily formed networks," http://www.ida.liu.se/~rtslab/HFN,.

PLACE PHOTO HERE

Jordi Cucurull is a post-doctoral researcher at the Real-Time Systems Laboratory (RTSLab) of Linköping University, in Sweden, since late 2009. His main research interests are computer security, computer networks, and green computing. He earned his PhD in 2008 from the Autonomous University of Barcelona (UAB). He obtained the BSc and MSc degrees in Computer Engineering with honours in 2002 and 2004 from the same university.

PLACE PHOTO HERE

Mikael Asplund is an assistant professor at the Dept. of Computer and Information Science at Linköping University, and is currently working as a Research Fellow in the Distributed Systems Group at Trinity College Dublin. He received his MSc degree in computer science and engineering in 2005 and his PhD in computer science in 2011 both from Linköping University.

PLACE PHOTO HERE

Simin Nadjm-Tehrani received her BSc degree (with honours) from Manchester University, UK, and her PhD degree in Computer Science at Linköping University, Sweden, in 1994. In 2006-2008 she was a full professor at the University of Luxembourg, and is currently back at Department of Computer and Information Science at Linköping University where she has led the Real-time Systems Laboratory since 2000. Her research interests are in dependable distributed systems with resource constraints.

PLACE PHOTO HERE

Tiziano Santoro was a visiting undergraduate student at Linköping University in 2010, and is currently employed by Google in UK.