Vulnerability-aware architecture for a tactical, mobile cloud

Anne-Laure Jousselme[1], Kevin Huggins[2], Nicolas Léchevin, Patrick Maupin and Dominic Larkin

Abstract Currently light infantry soldiers do not have access to many of their cyber resources the moment they depart the forward operating base (FOB). Commanders with recent combat experience have reported on the dearth of computing abilities once a mission is underway [14]. To address this, our group seeks to develop a tactical, mobile cloud implemented on a swarm of semi-autonomous robots. We provide two contributions with this work. First, provide a formal definition of the problem followed by a description of our approach to vulnerable state identification based on pattern recognition techniques. Second, we present an awareness definition as it pertains to our domain.

1. Problem statement and formalization

This is in essence a coverage problem. A robot is responsible for providing communication coverage to the set of clients in its area. Additionally, each robot must maintain communications with at least one other robot to ensure that the global network remains connected, see Figure 1.

First, we consider the elements of our domain. Let $R=\{r_1, \ldots, r_N\}$ be the set of robots and $C=\{c_1, \ldots, c_M\}$ the set of clients. The set C combined with their spatial location is a *configuration*. We denote ρ as a robot's unique communications range, which could be adjusted based on environmental demands. Next, $E=\{e_1, \ldots, e_N\}$ is the set of communication links between the robots and $G_c=(R, E)$ is the corresponding communication graph. Let N_c, be the set of node coordinates. Two robots r_i and r_j are separated by a distance of d_{ij} and are connected if their distance is less than ρ[3]. In later sections of this work, d_{ij} will denote the distance between two robots, two clients, or one robot and one client. We assume that (i) indirect links

[1]Anne-Laure Jousselme, Nicolas Léchevin, Patrick Maupin
Defence R&D Canada–Valcartier, 2459, Pie XI North, Quebec, G3J 1X5, Canada
{Anne-Laure.Jousselme,Patrick.Maupin,Nicolas.Lechevin}@drdc-rddc.gc.ca

[2]Kevin Huggins, Dominic Larkin
US Military Academy, West Point, NY 10996, USA
{Kevin.Huggins,Dominic.Larkin}@usma.edu

[3]For simplicity, obstacles that modify the communication links will not be considered.

are possible through intermediate nodes acting as relays, and (*ii*) at least one of the nodes is connected to an external communication node such as a satellite or UAV. In other words, we assume that there exists a communication resource capability within the network to ensure that clients' messages are handled properly through the mobile cloud via an external wide range communication relay.

The environment is represented by a navigation graph, where each node represents a possible position for the robots or clients and each edge between nodes is a possible path [15].

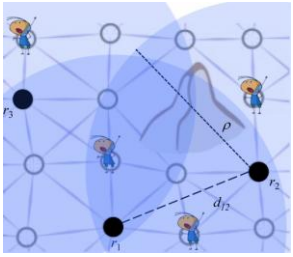

Fig. 1 A tactical mobile cloud for communication coverage. Black circles are robot nodes r_i and light blue circle represent their coverage. ρ is the communication range and d_{ij} is the distance between robot r_i and r_j.

Our formal objective is to provide continuous communications coverage for all clients while simultaneously maintaining sufficient connectivity within the network of robots. Accordingly, our research hypothesis is the following: an early identification of network vulnerabilities will prevent catastrophic events.

1.1. Coverage and connectivity

Central to our work is a precise definition of *coverage*. Clients need to have coverage in the mobile communications architecture in order to access the tactical cloud. Similarly, robots need to provide global coverage to ensure all clients are fully connected to the cloud. For simplicity, we provide a binary definition of coverage: covered or not covered. However, this definition can be easily extended to other models, such as probabilistic ones. We now consider a robot r_i with a communications range ρ and a client c_j separated by a distance d_{ij}.

We define coverage provided by robot r_i to client c_j as

$$\begin{aligned} cov(r_i, c_j) &= 1 \quad \text{if } d_{ij} < \rho \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{1}$$

The global coverage for the network of robots at a given client c_j is

$$cov(R, c_j) = \sum_{i=1}^{N} cov(r_i, c_j) \tag{2}$$

which is the number of robots that cover client c_j. In (3) we describe the inverse, *i.e.*, a definition of set of clients covered by robot r_i.

$$cov(r_i, C) = \sum_{j=1}^{M} cov(r_i, c_j) \tag{3}$$

Given the coverage definitions from the perspectives of both the client and the robot, equation (4) describes the *global coverage* of the network relative to the set of clients.

$$cov(R,C) = \sum_{i=1}^{N} cov(r_i, C) = \sum_{j=1}^{M} cov(R, c_j) = \sum_{i=1}^{N} \sum_{j=1}^{M} cov(r_i, c_j) \quad (4)$$

The other notion crucial to our work is that of network connectivity. We say that two robots are connected if they are in their respective range of communication. We define then:

$$\begin{aligned} con(r_i, r_j) &= 1 \quad \text{if } d_{ij} < \rho \\ &= 0 \quad \text{otherwise} \end{aligned} \quad (5)$$

A value of 1 means then that a link exists between the two robots.

We say then that the global network connectivity holds if for each pair of robots (r_i,r_j) there exists a path linking r_i to r_j:

$$con(R) = 1 \text{if } \forall (r_i, r_j) \in R^2, \exists (r_1,\ldots,r_m) \in R^m; (r_i, r_1), (r_1, r_2)\ldots,(r_m, r_j) \in E \quad (6)$$

Alternative definitions could easily replace these binary definitions and define different connectedness indices such as the algebraic connectivity [6]. Indeed, allowing real values for the connectivity would increase our flexibility in the definition of vulnerable states and lead to different cost functions (see Section 3.3). This kind of approach would lead to a multi-class problem rather a binary one and will be considered in future work.

1.2. Vulnerability

The vulnerability V of a system S can be understood as a mapping, V_S: $T \mapsto \mathcal{C}$, between an initiating threat T, whether intended or not, and a resulting consequence $\mathcal{C}$ characterized by a degree of loss [16] and related to system inoperability or state unreachability. Depending on how the threat uncertainty is characterized, the cost function may be aggregated, giving rise to an expected cost, or equivalently to a risk function [1]. Vulnerability thus corresponds to the susceptibility of a system or to the manifestation of the inherent state of a system, which can be severely affected when threatened [7]. Following the classification proposed by Klibi et al. [10], uncertain initiating events such as threats can be classified as either random, hazardous, or deeply uncertain events. Depending on the event model adopted, various approaches can be used to deal with uncertainties arising in a decision-making problem. For instance, the approach advocated by Brown et al. [2], which is based on worst-case bi-level or simplified tri-level programming, is to be contrasted with risk analysis involving probability and event trees [5], [1]. Indeed, the fault trees used to locate the single point of failure or the minimal cut set[4] that

[4] A set of edges of a graph which, if removed (or "cut"), disconnects the graph (*Wolfram, Mathworld*, http://mathworld.wolfram.com/CutSet.html).

maximizes the probability of disruption relies on the probabilistic modeling of events such as random failures [22].

In this paper, a mobile network *vulnerable* state can be defined as an instance of the network's state that may evolve in time until it affects the network's functions and the completion of its goals. Endogenous and exogenous threats to the network include the robots' inability to precede as intended, possibly due to hardware-software failures or malevolent acts, electronic warfare, obstacles, or unexpected client moves that cause some robots to move beyond their neighbors' communication range. Consider a sample set of possible clients configurations C_0 and a corresponding robot deployment represented by graph $G_0=(R, E_0)$. Include also the set $N_{c,0}$ of nodes' coordinates at time instant t_0 (encoded as attributes of the nodes). Various experiments can be conducted by triggering the loss of a robot or a subset of robots or by repositioning clients. The occurrence at t_1 of this triggering event may give rise to an adaptive robot deployment, whereby communication links can be either permanently lost or re-established, depending on the relative distance to neighboring robots. This hybrid dynamical system is characterized by switching time instants $\{t_1, t_2, \ldots, t_m\}$, where $t_{i+1}>t_i$. At t_i, the edge set jumps from E_i to E_{i+1}. An edge (i,j) is lost whenever the distance between robots r_i and r_j is greater than the communication range. It is assumed that the node set R remains invariant whether or not a robot is able to operate. The final time instant t_m is defined by the absence of any future triggering events such as a robot failure or a client move.

As further explained in Section 2.3, a component of the network (i.e., edge, node, or sub-network) is classified as *vulnerable* when a graph-connectedness-related cost associated to this component is above a prescribed threshold.

1.3. Vulnerability awareness

The notion of awareness considered in this work is derived from the concept of limited system resources [9],[8]. Intuitively, awareness is an epistemic state, close to knowledge, referring to a limited view and a limited capacity of the agents to reach a perfect state of knowledge, the one that would be reached by perfect logically omniscient reasoners. When defining situational awareness, one must consider the concepts of attention, vigilance, intelligence and stress within the context of resource-bounded agents. Therefore, we adopt the following definition of awareness: "an agent is aware of a proposition y if it can compute the truth value of y before time t". The vulnerability awareness of one robot r_i [5] is thus directly linked to its ability to come with an answer the question y ="*Am I vulnerable*?" by means of an algorithm (to be detailed in the upcoming section) given its limited resources (memory, computation, move, etc). Particularly challenging is the feature selection process as (1) the more features, the higher the computation and memory costs; and (2) some feature may require a complete map of the swarm involving higher memory needs while other may be evaluated locally.

[5] This is a local definition but a global definition would concern a central instance having access to the global swarm's state.

2. A pattern recognition approach to vulnerability assessment

2.1. Principle

In [13], the authors proposed a toolbox with the goal of detecting and predicting the vulnerabilities in complex networks. These principles rely on pattern recognition techniques that leverage structural, dynamical, and functional features selected to sensitize the classifier to potential vulnerabilities in abnormal situations. Such an approach is expected to yield fast vulnerability prediction when compared with a simulation using a first-principle-based model of the network. The problem of complex systems vulnerability assessment has already been interpreted as a classificatory problem, which includes such applications as disease surveillance systems [20] and the crisis recognition [11]. Our approach integrates pattern recognition techniques applied to a time series and a network's structural and dynamical properties.

To determine vulnerability, we reason over the network using pattern recognition. With it, we design by training a mapping ψ such that:

$$\begin{aligned} \psi : \mathcal{G} &\rightarrow \{y; \bar{y}\} \\ x &\mapsto \psi(x) = \hat{y} \end{aligned} \tag{7}$$

where x is a representation of an element of $\mathcal{G}$ (e.g. a node, link or sub-graph) and $\hat{y}$ is an estimate of the detrimental effect of that element on the network, either 1 if vulnerable or 0 otherwise. Note that we use the term "vulnerable" for qualifying a node although this is extended to the network. Typically, x is a vector of k network features identified as relevant by feature selection pre-processing. As mentioned in Section 1.3, one of the crucial tasks consists in identifying the set of candidate features for the problem.

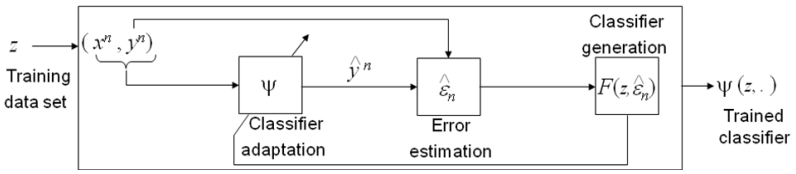

Fig. 2: Training of the network vulnerability classifier. F aims to yield a classifier ψ that minimizes the error estimation given the training data set $z=(x^n,y^n)$.

The training data set $z^n=(x^n,y^n)$ consists of n instances of R, that is a set of n samples x^n of k features together with a label y (see Table 1). z^n feeds a classifier generation mechanism F, as shown in Fig. 2, which seeks a classifier that minimizes the error estimate between the set of estimated class labels and their corresponding ground truth.

2.2. Features

In [1], the authors proposed four network feature categories. The first pertains to the structural properties of a labeled, weighted graph. These include centrality, similarity connectivity, shortest path metrics, clustering coefficient, spectral properties, vertex coreness, graph density, average nearest neighbor degree, among others [3]. The second category considers flow dynamic changes by exploiting information on signals and systems such as Fourier transform, spectral monitoring [12], bifurcation analysis [4] and efficiency measures [17]. Indicators pertaining to complex system science and statistical physics described the third category of features with examples including the exponent of the power-law distribution of failure occurrences at a crossover [18], the local shape factor of a sand pile adapted to networks [19], the Cavahlo-Rodrigues entropy, the spatial entropy, the fractal dimension, the symmetropy, the Hurst coefficient, and the self-similarity parameter [21]. The final feature group concerns functional information on key components of the network. This fourth category includes the notions of coverage introduced in Section 2.2. Our example features were drawn from this latter group as well as the structural features from the first category.

2.3. Training

The purpose of feature extraction is to build a representation that is particularly suited to vulnerability recognition problems specific to networks. Indeed, features are naturally geared to the modeling of classificatory problems. Once this model is derived, fast and efficient recognition is expected when compared to physics-based models of large interconnected networks. A parallel between pattern-recognition-based and game-theoretic approaches is proposed in [13].

Given an initial clients configuration C_0 and a corresponding robot deployment G_0, (including the robots' coordinates $N_{c,0}$), the training data set z used to derive the classifier in (7), is obtained from the disturbance sample set $\mathscr{D}=\{D_1,\ldots,D_n\}$. D_i stands for a set of sequence of disturbances $\{\delta_{ij1}, \delta_{ij2},\ldots,\delta_{ijm}\}$. This sequence is in a one-to-one correspondence with the occurrence time set $\{ t_1 ,t_2, \ldots, t_m\}$, $t_{i+1}>t_i$. δ_{ijk} represents a random realization of the jth disturbance of the sequence applied to the network of robots at time t_k and for experiment i. The first disturbance δ_{ij0} corresponds to a robot failure and is followed by a series of new client configuration C_j, $\{\delta_{ij1},\ldots,\delta_{ijm}\}$. It assumed that the first triggering event (first disturbance) δ_{ij1} is a robot failure occurring at t_1 and that all δ_{ij1} span R, the set of robots. The following disturbances are a series of new client configuration C_j, $\{\delta_{ij1},\ldots,\delta_{ijm}\}$ representing new client configurations arising from clients adjusting their positions. In order to evaluate each robot's vulnerability, they will be removed successively. D_i is then defined as the following union $\cup_{j\in R}\{\delta_{ij1}, \delta_{ij2},\ldots, \delta_{ijm}\}$. D_i is instrumental in defining experiment i. Indeed, sequence D_i generates the sequence of edge sets $\{\{E_{11},\ldots, \mathrm{E}_{1m}\},\ldots,\{E_{|R|1},\ldots, E_{|R|m}\}\}$, where $|R|$ denotes the total number of robots. The state of G_m is used to determine the cost $\mathscr{C}_{ij}(con_i,cov_i)$ associated with a disturbance sequence j of experiment i. This cost depends on the connectedness of

the graph through a disconnectedness index, $con_i(G_m)$, derived from equation (6), and the coverage $cov_i(G_m,,C_m)$ of the set of clients by the robot network at time t_m. In the binary case, when G_m is disconnected, con_i is equal to 1 and when G_m is connected, con_i is set to zero. Sequence j of experiment i is thus associated with the following mapping

$$\{d_{ij1},d_{ij2},\ldots,d_{ijm}\} \rightarrow \mathscr{C}_{ij} \rightarrow y_{ij}$$

Each experiment i is characterized by the set of labels $\{y_{i1},\ldots, y_{i|R|}\}$. The classifier used to analyze the vulnerability of the robot network's dynamics in response to the occurrence of possible contingencies is obtained by exploiting the data of Table 1 established for each experiment $i \in \{1,\ldots,N\}$.

Table 1 Table of features excerpt from Experiment *i*. Features are divided into two classes depending on whether each robot is able or not to compute the features from local information, which is sent by adjacent nodes.

		Features						Label
		Local			Global			
	...	...			...			...
Experiment *i*	Robot 1	f_{11}	f_{12}	...	...	f_{1p-1}	f_{1p}	y_{i1}
	Robot 2	f_{21}	f_{22}	...	...	f_{2p-1}	f_{2p}	y_{i2}
	⋮	⋮				⋮		⋮
	Robot $\|R\|$	$f_{\|R\|1}$	$f_{\|R\|2}$	...	...	$f_{\|R\|p-1}$	$f_{\|R\|p}$	$y_{i\|R\|}$
	...	...			...			...

3. Conclusion

In this paper, we have described a formal definition of a vulnerability-aware mobile, tactical cloud architecture designed to support dismounted soldiers. We developed a problem formalization that described the network as well as a definition of coverage and vulnerability. In addition, we provided a formal definition of awareness as it pertains to our domain. We applied a novel pattern-recognition approach to vulnerability assessment, which enables robot nodes in the network to efficiently detect when they are in a vulnerable state. Future work consists of implementing and benchmarking the architecture as well as exploring multi-class problems for more robust connectivity definitions.

References

[1] Al Mannay, W.I. and Lewis, T.G. , Minimizing network risk with application to critical structure protection, Journal of Information Warfare, Vol. 6, No. 2, pp. 52-68, 2007.

[2] Brown, G.G., Carlyle, W.M., Salmeron, J., and Wood, K., Analyzing the vulnerability of critical infrastructure to attack and planning defenses, Tutorial in Operations Research, Informs, pp. 102-123, 2005

[3] Csárdi, G., Nepusz, T., The igraph software package for complex network re-

search, InterJournal Complex Systems, 1695, 2006.

[4] Dobson, I., Distance to bifurcation in multidimensional parameter space: margin sensitivity and closest bifurcations, in *Bifurcation Control, Theory and Application*, Eds Chen, D.J. Hill, X. Yu, Vol. 293, pp. 49-66, Springer-Verlag, 2003.

[5] Garcia, M.L., Vulnerability assessment of physical protection systems, Butterworth-Heinemann, 2006.

[6] Godsile, C. and Royle, G., *Algebraic Graph Theory*, New York: Springer, 2001.

[7] Haimes, Y.Y., On the definition of vulnerabilities in measuring risk to infrastructure, Risk Analysis, Vol. 26, No. 2, pp. 293-296, 2006.

[8] Halpern, J., Moses, Y. and Vardi, M. Y. "Algorithmic knowledge," in Proc. of the 5th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'94). Morgan Kaufmann, 1994, pp. 255–266.

[9] Jousselme, A.-L., Maupin, P., Garion, G., Cholvy, L., Saurel, C., Situation awareness and ability in coalitions, 10th International Conference on Information Fusion, Quebec city, Canada, 9-12 July 2007.

[10] Klibi, W., Martel, A., and Guitouni, A., The design of robust value-creating supply-chain network: a review, European Journal of Operational Research, Vol. 203, No. 2, pp. 283-293, 2010.

[11] Larsen, H.L.and Yager, R.R., A Framework for Fuzzy Recognition Technology, IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol. 30, No.1, pp. 65-76, 2000.

[12] Léchevin, N, Rabbath, C.A., and Maupin, P., Toward a stability monitoring system of an asset-communications network exposed to malicious attacks, American Control Conf., San Francisco, 2011.

[13] Léchevin, N., Jousselme, A.-L., Maupin, P., Pattern Recognition Framework for the Prediction of Network Vulnerabilities, IEEE Network Science Workshop, West Point, NY, June 2011.

[14] Levine, C., Soldiers as Gathers, Analysers, and Users of Situational Information, Workshop on Information Sharing at the Front Line, Indian Wells, CA, April 2010

[15] Maupin, P., Jousselme, A.-L., Wehn, H., Mitrovic-Minic, S., Happe, J., A Situation Analysis Toolbox: Application to Coastal and Offshore Surveillance, Int. Conf. on Information Fusion, Edinburgh, UK, 2010.

[16] McGill, W.L. and Ayyub, B.M., The meaning of vulnerability in the context of critical infrastructure protection, in *Critical Infrastructure Protection: Element of Risk*, Critical Infrastructure Protection Program, George Mason University School of Law, 2007.

[17] Nagurnay, A. and Quiang, Q., A network efficiency measure with application to critical infrastructure networks, Journal of Global Optimization, Vol. 40, pp. 261-275, 2008.

[18] Pradhan, Hansen, A. and Hemmer, P.C., Crossover behavior in burst avalanches: signature of imminent failure, Physical Review Letters, Vol. 95, No. 12, pp. 125501-1(4), 2005.

[19] Ramos, O., Altshuler, E., and Maloy, K.J., Avalanche prediction in a self-organized pile of beads, Physical Review Letter, Vol. 102, No. 7, pp. 078701(1-4), 2009.

[20] Shmueli, G., and Fienberg, S.E., Current and potential statistical methods for monitoring multiple data streams for bio-surveillance, in Statistical Methods in Counter-Terrorism, pp. 109-140, Springer, 2006.

[21] Sprague, K.B. and Dobias, P., Behavior in Simulated Combat – Adaptation and Response to Complex Systems Factors, DRDC CORA TM 2008-044, 2008.

[22] Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., and Railsback, J., Fault tree handbook with aerospace applications, NASA Office of Safety and Mission Assurance, Washington, DC, 2002.