Trust Based Monitoring Approach for Mobile Ad Hoc Networks

Nadia Battat[1], Abdallah Makhoul[2(✉)], Hamamache Kheddouci[3], Sabrina Medjahed[1], and Nadia Aitouazzoug[1]

[1] LIMED Laboratory, University of Bejaia, Algeria, France
nadiabattat@yahoo.fr
[2] FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, CNRS, Belfort, France
abdallah.makhoul@univ-fcomte.fr
[3] LIRIS Laboratory, University of Lyon 1, Villeurbanne, France
hamamache.kheddouci@univ-lyon1.fr

Abstract. Mobile ad-hoc networks (MANET) are vulnerable to many types of attacks. Monitoring MANET is then essential to ensure high level performance. Many challenges arise in the MANET self-monitoring. Namely, the limited storage and energy resources of mobile nodes, the high topological dynamism and the unpredictable behaviors, etc. In this paper we propose a new self monitoring scheme that comprises a new multi criteria monitors' election method while integrating a new trust based cooperation technique. This scheme does, not only, elect the trustworthy monitors having a large capacity, but it also can guarantee the continuous participants' control in order to measure their sincerity. We validate our approach through several simulations. The experimental results indicate that the proposed scheme outperforms the cluster-based and CDS-based architectures in terms of the number of exchanged messages, excluded regular monitors and that of detected irregular monitors.

1 Introduction

Self monitoring of MANET consists in evaluating the operational state of its mobile devices, the links between them as well as its quality of service. This is achieved by a subset of mobile nodes (called monitors) which are elected according to several predefined parameters [5]. Each monitor performs its assigned tasks and in the same time is responsible for controlling a subset of mobile nodes in its area called the monitored nodes. In their turn, the monitored nodes are responsible for enforcing the policies they receive from their monitors, collecting the requested information and delivering them to their corresponding monitors. The monitoring evaluation can be performed by analyzing and processing the local collected data by the nodes as well as the information received from their neighbors. To realise high level monitoring, it is vital that each participant (monitored node or monitor) contributes correctly to the election of monitors and the monitoring process. However, this leads to consume more computational and energy

A. Puliafito et al. (Eds.): ADHOC-NOW 2017, LNCS 10517, pp. 55–62, 2017.
DOI: 10.1007/978-3-319-67910-5_5

resources. Actually, not all nodes participate in this process. Selfish nodes can use the resources of the others without participating in the monitoring functions. Malicious nodes can falsify the collected data, modify the distributed policies and make illegal and inappropriate decisions. In order to force mobile nodes to obey the monitoring approach and cooperate with each other, we propose in this paper a new monitor electing method. It is based on two main factors: truthfulness and capability.

In literature, several monitoring approaches were proposed for MANET [5]. These approaches can be classified as follows: ***Unique criterion based election approaches:*** These approaches [5,6] use only one criterion such as Lowest-ID or Highest-Degree to elect monitors in that they are easily achievable. However, these algorithms do not take into account all MANET characteristics and the resources level of the elected nodes. This can lead to reapply the election process which reduces the lifetime of monitoring cycle and increases the network overhead in addition to the consumed energy. Moreover, they do not balance the monitoring tasks uniformly among all the nodes. This can result in electing the same node as monitor frequently. ***Multi criteria based election approaches:*** These ones [4,7] use a diversity of criteria to elect monitors. They aim to increase the lifetime of the monitoring cycle by electing the most cost-efficient nodes as monitors. In this paper, we present a new scheme to guarantee an efficient monitoring in multi-hop mobile ad hoc networks. Furthermore, we define a set of rules in order to detect the malicious and selfish behaviors. We study the performance analysis and evaluation of the proposed architecture through simulations. The obtained results show that the proposed scheme can significantly reduce the overhead and maintain a high level of detection.

2 MANET Self Monitoring Scheme

2.1 Messages Structure

We propose some modifications on the original hello message by adding the following fields: *Weight* ($W(n_i) \in [0,1]$): this field contains the weight of the node n_i that is initialized to 0. Its value is estimated in Sect. 2.3; *Trust value* $T(n_i) \in [0,1]$: this field represents the trust value of the node n_i that is initialized to 0.5; *Energy level* $E(n_i)$: this field indicates the remained energy level of the node n_i; *Role*: this field defines the role of the node n_i: monitor, delegated monitor or ordinary node. $NeighborsList$: this field contains the IDs of the node neighbors and their estimated trust values.

2.2 Trust Computation Method

Each node must observe the behaviors of its neighbors to detect their malicious or selfish comportments. Consequently, it can observe and trace their behaviors by the continuous updates of the trust values. Initially, we assign to each

node a trust value equal to 0.5[1]. Furthermore, we define the following rules for identifying the selfish or the malicious behaviors.

- The contribution level of each mobile node can provide falsified evaluation about its collaborations in order to raise its trust value. Therefore, a monitor can distribute a part or the full report to its controlled node to confirm its honesty. A monitored node can either select a route containing a maximum number of its neighbors for forwarding its data and/or the local report, or divide this quantity of data into N packets (N represents the number of its neighbors). Then it sends each one through each neighbor. When one neighbor drops packets and that this behavior is observed by a sender, the latter will decrease its trust value. A local analysis is needed to avoid monitors to act maliciously and to detect their selfish behaviors. Misbehaved or selfish nodes will be penalized by decreasing their trust values.
- As mobile nodes use limited storage capacities, they can discard not only their collected data but also data of other nodes, in order to exploit its resources for further interesting uses. Therefore, the monitored node (resp. the monitor) can periodically ask its monitors (resp. the data holder) to send a randomly selected piece of its collected data at a specific time. Once receiving this requested data, the monitored node (resp. the monitor) compares it to its stored data hunk and then increases or decreases the corresponding node trust value.
- The participants' contributions of mobile nodes can indicate the existence of malicious or selfish nodes. For instance, if a node exchanges its opinions on neighbors periodically and performs a local analysis without participating in forwarding data or data storage, it will be considered as malicious.
- A monitor can compare the received data within its radio range to detect the malicious or selfish behaviors of its neighbors. For instance, if more than one neighbor indicate that two nodes X and Y are neighbors and the neighbors' list of X does not contain any information about Y, a monitor can conclude that X is either selfish or malicious.

For updating the value of confidence, we use the activity rate (AR), which is calculated according to the number of positive realized tasks including the packet forwarding rate and the realized monitoring tasks. If we consider two nodes i and j, the node i calculates the $AR(j)$ as follows: the node i should record the number of positive interactions ($pos(i, j)$) with the node j, and the total number of interactions ($total(i, j)$), over a given interval of time, and then it calculates the activity rate as follows:

$$AR = pos(i, j)/total(i, j) \tag{1}$$

The trust value is estimated over time to reflect changes in the activity rate. Nevertheless, local estimation on each mobile node might not be enough to detect any node bad behavior. It should have information from other nodes.

[1] To not consider a node in advance as being selfish, malicious or confident.

Moreover, in some cases, a mobile node can monitor only the behavior of its direct neighbors. As a result, not all neighbors at $n - hops$ will honestly share the real values. Consequently, we propose that each node calculates the trust values based on the combination of direct and indirect estimations that derive from neighbors. Therefore, we consider also the two following cases: **(a)** A neighbor does not report his accurate trust value about the corresponding monitor (resp. monitored node) in case of hardware or software failures held by this node; **(b)** A neighbor can provide a false trust value about the corresponding monitor (resp. monitored node). It may provide a negative (or positive) value to misbehaved/trusted monitor (resp. monitored node): *false accusation attack* [3] (or *false praise attack* [1]).

Algorithm 1. Locally detection of regular, irregular and normal nodes

Constant $MaxNbrF = 3$;
$T(n_i)$: Trust value of the node n_i; $NbrF$: number of node' faults ;
A: Last activity that must be realized by the node n_i;
$The functionK$: $K(A) = 1$, if A is correctly realized by n_i, otherwise $K(A) = 0$;
$The functionB$: $B(n_i) = M$, if n_i acts maliciously *or* $B(n_i) = S$ if n_i acts selfishly;
SL: List of detected selfish nodes;
ML: List of detected malicious nodes;
$E(n_i)$: Energy level of the node n_i; Et: Necessary energy level for realizing A ;
$The functionS$: $S(n_i) \in$ {Irregular, Regular, Normal}; Dt: Penalty period;
Begin
if $(((n_i \notin SL) and (n_i \notin ML)) or (S(n_i) = Normal))$ **then**
  if $(K(n_i) = 1)$ **then**
    if $(T(n_i) \prec 1)$ **then**
      Recompute $T(n_i)$;
      if $(T(n_i) \succ 0.5) and (S(n_i) = Normal))$ **then**
        $S(n_i) =$ Regular;
  else
    if $(T(n_i) \succ 0)$ **then**
      Recompute $T(n_i)$;
      $NbrF = NbrF + 1$;
  if $(T(n_i) \leq Bt)$ **then**
    if $(((n_i \notin SL) and (n_i \notin ML)))$ **then**
      if $(E(n_i) \succ Et)$ **then**
        if $(NbrF \geq MaxNbrF)$ **then**
          $S(n_i)=$ Irregular;
          $Dt =$ CurrentTime $+ Tb$;
          if $(B(n_i) = S)$ **then**
            add n_i to SL;
          else
            add n_i to ML;
End

After receiving the indirect estimations, a node i calculates the trust values $T(c)$ (its and that of its neighbors) using the following formula:

$$T(n_j) = (\sum_{k=1}^{n}(T(k,j)) + T(i,j))/(n+1) \quad (2)$$

n is the nodes number having sent their trust values about the node j to the node i. $T(k,j)$ is node k trust value about node j.

When a node does not receive any trust value, it can rely either on its trust values or on the previously gathered ones. The trust value can be increased or decreased by a chosen changing step Stp, according to nodes' behaviors. We assume that the chosen changing step $Stp = 0.1$ [2,8]. Mobile node can behave selfishly or maliciously following to its features or according to the mobile environment characteristics. Nevertheless, environmental conditions can lead to intensively deteriorate trust values. Therefore, we propose to use the maximum authorized faults number $MaxNbrF$ to avoid the inexactitude of trust value estimation. If a node does not participate in *three* successive activities while it has sufficient energy level to perform them and its trust value is equal or less than a predefined threshold $Bt = 0.3$, it will be irregular. A detected node will be added to the selfish or malicious nodes list (see Algorithm 1) according to its last behavior.

A mobile node will be considered as malicious if it: falsifies the monitoring policies; generates unnecessary traffic; advertises non-existing monitors; modifies the monitoring system; provides fake data; broadcasts a false alarm; contributes in some monitoring tasks only. On the other hand, it can be considered as selfish if it: refuses to participate in monitoring process; discards the collected data; drops the exchanged monitoring messages.

2.3 Monitors Election

Our approach is a multi criteria based election method. The network is logically divided into clusters with a single monitor (cluster-head). We assume that only regular nodes can participate in monitors election. Every regular node n_i, aware of its neighbors, performs the following steps:

1. it calculates its weight $W(n_i)$ which indicates its ability to serve as monitor as follows: $W(n_i) = COF1 * T(n_i) + RS(n_i)$, where, $COF1 \in [0,1]$ is the metric trust value coefficient. A monitor can consume more resources than a monitored node because of the monitoring tasks that must be performed. In fact, we also use the weighted parameter $RS(n_i)$ to elect monitors. This weighted parameter can be computed according to: the processing power[2]; the energy level; the storage capacity.
2. it forwards a hello message, containing its weight, its trust value, its neighbors and their estimated trust values list and its energy level, to its neighbors.

[2] Node with little processing power can slow the forwarding or analyzing of collected data.

3. it waits a time period for receiving messages from its neighbors.
4. it compares its weight with those of its neighbors. It becomes monitor, if it has the maximum weight.

A monitor informs its neighbors about its presence by sending hello message, while initializing the field *Role* to 1. Each neighbor selects the nearest monitor based on hop count.

2.4 Maintenance of the Monitoring Architecture

The proposed topologies for the monitoring approaches are usually based on the construction of cluster or CDS (Connected Dominating Set) [5]. Our proposed approach is also cluster-based where each participant (monitor or monitored node) is controlled by its regular neighbors. To detect any mobile node neighbors, periodic hello messages are exchanged. Once these messages are received, each mobile node can update the list of its neighbors, their weights, their trust values and their roles with minimum transmission overhead. When mobile nodes voluntarily/involuntarily disconnect or move, our approach faces these topological changes by applying the following policies. **(a)** When a new regular node joins a network, it exchanges with its neighbors its data, and then chooses the nearest monitor. **(b)** When a mobile node loses connectivity with its monitor, two cases can be considered: **1.** ***Voluntary disconnection of regular monitor:*** the regular monitor can select one of its regular neighbors having the maximum weight to replace it. Then, it informs its neighboring nodes and the other monitors about the new one. **2.** ***Involuntary disconnection of monitor:*** when a mobile node detects the sudden death of its monitor, it launches the election of new monitor.

3 Simulation Results

To study the effectiveness of our scheme, we compare it with the cluster-based and CDS-based architectures. We use the same metrics as our approach to elect cluster-heads and dominator nodes. We assume that $RS(n_i) = COF2 * EC(n_i)$ where $COF1 = COF2 = 0.5$ and $EC(n_i)$ indicates the remaining energy level of the node n_i. We also assume that the necessary energy for performing a given monitoring task, the mobile node trust value and remaining energy level are randomly selected from the range $[0, 1]$ following a uniform distribution. The settings of our simulations are as follows: $Duration = 5000\,\text{s}$, $Numberofnodes = 100$, $Territoryscale = 100\,\text{m}^2$, $Rangeofnode = 20$, the mobility model is random waypoint, $Pauseinterval = [0, 20]$ (s), $speedinterval = [0, 20]$ (m/s).

Figure 1 indicates the number of the exchanged messages in order to construct topologies through time. From the results, we can observe that our scheme outperforms the cluster and CDS based architectures by attaining low message overhead. This explains that only regular nodes can perform the monitoring plan. Figure 2 illustrates the evolution of the irregular monitors detection rate through

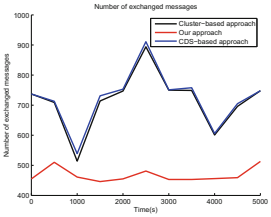

Fig. 1. Number of exchanged message

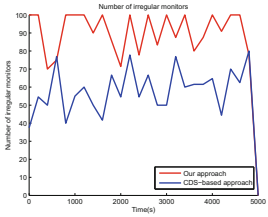

Fig. 2. Number of detected irregular monitors

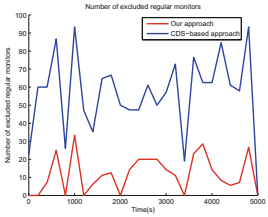

Fig. 3. Number of excluded regular monitors

time. The results indicate that an important detection rate of our scheme. This is interpreted by the fact that in CDS-based architecture, regular monitors can be isolated and consequently, cannot detect the malicious or selfish behaviors of irregular ones. Figure 3 shows that our scheme decreases the number of the excluded regular monitors compared to CDS-based architecture.

4 Conclusion and Perspectives

Monitoring the behavior of each mobile is an essential requirement for developing a robust and reliable monitoring approach. Therefore, we propose to elect only well behaving and honest nodes as monitors. We select monitors based on a weighing factor which uses the trust value. The latter is measured using the rate of contribution in monitoring process of the participants and their neighbors. We evaluated the performance of this scheme compared with the cluster-based and CDS-based architectures. The obtained results demonstrate the effectiveness of our scheme in terms of the numbers of the exchanged messages, the excluded regular monitors and the detected irregular monitors. The proposed scheme also decreases the maintenance time.

References

1. Alzaid, H., Alfaraj, M., Ries, S., Jsangand, A., Albabtain, M., Abuhaimed, A.: Reputation-based trust systems for wireless sensor networks: a comprehensive review. In: IFIPTM, pp. 66–82 (2013)
2. Awad, A.I., Hassanien, A.E., Baba, K. (eds.): Advanced in Security of Information and Communication Networks. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40597-6
3. Banković, Z., Vallejo, J.C., Fraga, D., Moya, J.M.: Detecting bad-mouthing attacks on reputation systems using self-organizing maps. In: Herrero, Á., Corchado, E. (eds.) CISIS 2011. LNCS, vol. 6694, pp. 9–16. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21323-6_2
4. Battat, N., Kheddouci, H.: HMAN: hierarchical monitoring for ad hoc network. In: Embedded and Ubiquitous Computing (2011)

5. Battat, N., Seba, H., Kheddouci, H.: Monitoring in mobile ad hoc networks: a survey. Comput. Netw. **69**, 82–100 (2014)
6. Chen, W., Jain, N.: ANMP: ad hoc network network management protocol. IEEE J. Sel. Areas Commun. **17**(8), 1506–1531 (1999)
7. Shen, C., Jaikaeo, C., Srisathapornphat, C., Huang, Z.: The Guerrilla management architecture for ad hoc networks. In MILCOM, October 2002
8. Xu, X., Gao, X., Wan, J., Xiong, N.: Trust index based fault tolerant multiple event localization algorithm for WSNs. Sensors **11**, 6555–6574 (2011)