A Bayesian Approach for Disconnection Management in Mobile Ad-hoc Networks

Massimiliano de Leoni, Massimo Mecella, Ruggero Russo
SAPIENZA – Università di Roma
Dipartimento di Informatica e Sistemistica
Via Salaria 113, 00198 Roma, Italy
{deleoni,mecella,russo}@dis.uniroma1.it

Abstract

Mobile Ad-hoc Networks are used in many scenarios (e.g., emergency management) for supporting collaborative work of operators. But this requires either (i) *continuous connections, or at least* (ii) *the possibility to foresee that a device is going out to disconnect (e.g., in order to locally cache important data needed for the following activities, etc.). Therefore a basic problem is how to predict possible disconnections of devices, in order to let upper layers appropriately address connection anomalies (e.g., either taking global remedial actions to maintain the network connected, or local ones to let the disconnecting device to go on for some time with its own work). In this paper we present a bayesian approach to predict disconnections in* MANETs*, and validating experimental results that shows the viability of the approach.*

1. Introduction

A Mobile Ad hoc NETwork (MANET) is a peer-to-peer (P2P) network of mobile nodes capable to communicate with each other without an underlying infrastructure. Nodes can communicate with their own neighbors (i.e., nodes in radio-range) directly by wireless links. Anyway, non-neighbor nodes can equally communicate by using other intermediate nodes as relays which forward packets toward destinations [1]. The lack of a fixed infrastructure makes this kind of network suitable in all scenarios where it is needed to deploy quickly a network but the presence of access points is not guaranteed. Examples are military applications, and more recently, cooperative systems for emergency management [5].

Coordination and data exchange requires nodes are aware on when they are going to disconnect from the remaining of the MANET (i.e., there is not any neighbor within which the node is connected – inside the radio-range). Indeed, if nodes are alerted about probable disconnections, some remedial actions can be enacted:

local disconnection management – the node can take autonomous – not requiring coordination with other nodes – remedial actions, e.g., to cache some data that are needed for the following tasks to be carried out while disconnected, or to disseminate to other nodes critical information he is storing that are crucial for the other nodes (so to let them working);

global disconnection management – the nodes together coordinate, e.g., through an appropriate process management system [8] to be hosted on a "leader device", such that they arrange themselves in a new configuration in which no node is disconnected.

Therefore the basic problem to be addressed for building whichever coordination middleware on top of MANETs is the one of predicting disconnections of devices. In this paper we deal with such a problem, and we propose a novel and effective technique based on a bayesian approach.

The remainder of this paper is as follows. In Section 2 we describe the proposed technique, which is then validated through a set of experiments presented in Section 3. Section 4 describes relevant work, in order to highlight the novelty of our approach, and Section 5 concludes the paper by remarking how this technique is at the basis of the development of a coordination middleware for MANET we are currently developing.

2. Bayesian Filtering for Disconnection Prediction

Our predictive technique is based on some assumptions:

1. Each device is equipped with specific hardware that allows it to know its *distance* from the surrounding connected (i.e., within radio range) devices. This is not a very strong assumption, as either devices are equipped with GPS or specific techniques and methods (e.g.,

TDOA - time difference of arrival, SNR - signal noise ratio, the Cricket compass, etc.) are easily available.

2. There are no landmarks (i.e., static devices with GPS) in the MANET; we are indeed interested in very dynamic MANETs, in which neither the availability of landmarks is supposed.

3. At start-up, all devices are connected (i.e., for each device there is a path - possibly multi-hop - to any other device). The reader should note that we are not requiring that each device is within the radio range of (i.e., one hop connection to) any other device (*tight* connection), but we require only a *loose* connection (guaranteed by appropriate routing protocols, e.g., DSR, AODV, etc.).

4. A specific device in the MANET, referred to as *coordinator*, is in charge of centrally predicting disconnections. As all devices can communicate at start-up and the ultimate goal of our work is to maintain such connection through prediction, it is possible to centrally collect all information from all devices;

The predictive technique is essentially as follows: at a given time instant t_i the coordinator device collects all distance information from the other devices (for assumptions *(1)* and *(3)*); on the basis of such information the coordinator build a probable *connection graph*, that is the probable graph at the next time instant t_{i+1} in which the predicted connected devices are highlighted. On the basis of such a prediction, the coordinator layer will take appropriate actions (which are no further considered in the following of this paper).

2.1. Bayesian Filtering

Bayes filters [3] probabilistically estimate/predict the current state of a system from noisy observations. Bayes filters represent the state at time *t* by a random variable Θ_t. At each point in time, a probability distribution $Bel_t(\theta)$ over Θ_t, called *belief*, represents the uncertainty. Bayes filters aim to sequentially estimate such beliefs over the state space conditioned on all information contained in the sensor data. To illustrate, let's assume that the sensor data consists of a sequence of time-indexed sensor observations $z_1, z_2, ..., z_n$. The $Bel_i(\theta)$ is then defined by the posterior density over the random variable Θ_t conditioned on all sensor data available at time t:

$$Bel_t(\theta) = p(\theta | z_1, z_2, ...z_t) \tag{1}$$

Generally speaking the complexity of computing such posterior density grows exponentially over time because the number of the observations increases over time; to make the computation tractable is necessary to make the following two assumptions:

```
timer : a timer expiring each T seconds.
iBuffer[x,y] : a bi-dimensional squared matrix storing distance among
   couples of nodes X and Y
bayesianBuffer[x,y]: a bi-dimensional square matrix storing a triple
   (alpha, beta, distance) for each couple of nodes X and Y

upon delivering of a tuple (i,j,dist) from i
   ibuffer[i,j] := dist

upon expiring of timer
   copy ibuffer to localbuffer
   foreach (i,j) in ibuffer /* empty intermediate buffer */
      ibuffer[i,j] := RADIO_RANGE
   foreach i,j in localbuffer
      if (localbuffer[i,j] = RADIO_RANGE)
         observation :=1
      else
         observation := ( iBuffer[i,j] - bayesianBuffer[i,j].distance ) / RADIO_RANGE
         observation := (observation + 1) /2
      end if
      bayesianBuffer[i,j].distance := localbuffer[i,j]
      bayesianBuffer[i,j].alpha := u * bayesianBuffer[i,j].alpha + observation
      bayesianBuffer[i,j].beta := u * bayesianBuffer[i,j].beta + (1 – observation)
   end foreach
```

Figure 1. The pseudo-code of proposed algorithm.

1. the system's dynamic is markovian i.e., the observations are statistically independent;

2. the devices are the only subjects that are capable to change the environment.

On the basis of the previous two assumptions the equation in a time instant *t* can be expressed as the combination of a prediction factor $Bel_{t-1}(\theta)$ (the equation in the previous time instant) and a update factor that on the basis of the observation in the time instant *t*, realizes the update of the prediction factor.

In our approach the random variable Θ_t belongs to [0,1] and we use the Beta(α,β) function as a *belief* distribution to model the behaviour of the system, according to the following equation:

$$Bel_t(\theta) = Beta(\alpha_t, \beta_t, \theta) \tag{2}$$

where α and β represent the state of the system and vary according to the following equations:

$$\begin{cases} \alpha_{t+1} = \alpha_t + z_t \\ \beta_{t+1} = \beta_t + z_t \end{cases} \tag{3}$$

In our approach, the observation z_t represents the variation of the relative distance between nodes *(i,j)* normalized with respect to radio range in the period *[t-1,t]*. It is used to update the two parameters α and β of the Beta function according to equation 3. The evaluated $Beta(\alpha, \beta)$ function predicts the value of $\theta_{t+1}^{(i,j)}$ estimating the relative distance that will be covered by the nodes *(i,j)* in the next time period *[t,t+1]*.

2.2. Prediction of the Distances

Our approach relies on clock cycles whose periods are T. The pseudo-code for the monitor is described in Figure 1. We assume the access of shared data structures to be appropriately synchronized.

For each ordered couple (i, j) of nodes, in the n-th cycle, monitor stores two float parameters, $\alpha_n^{(i,j)}$ and $\beta_n^{(i,j)}$, and the last observed distance $d_{n-1}^{(i,j)}$. If a node k comes in the MANET in the m-th clock cycle, $\forall$ node j already in MANET we initialize $\alpha_m^{(k,j)} = \beta_m^{(k,j)} = 1$ so to get uniform distribution in $[0, 1]$, i.e., every distance $d_m^{(k,j)}$ gets the same probability.

For each period T, each generic node i sends to the monitor a set of tuples (i, j, d^j) where j is the unique name of a neighboring node and d^j is the distance to j. Coordinator collects continuously such tuples (i, j, d^j) coming from nodes in an intermediate buffer. Please observe that clocks of generic nodes are not synchronized but they can shift at most $T/2$.

Monitor performs prediction according to the same clock T: at the generic n-th clock cycle upon timer expiring, it copies the tuples (i, j, d_n^j) from the intermediate buffer to another one and, then, it empties the former buffer to get updated values. In the clock cycle, for each collected tuple (i, j, d^j) monitor updates the parameters as follow by a bayesian filter:

$$\begin{cases} \alpha_{n+1}^{(i,j)} = u\alpha_n^{(i,j)} + o_n^{(i,j)} \\ \beta_{n+1}^{(i,j)} = u\beta_n^{(i,j)} + (1 - o_n^{(i,j)}) \end{cases}$$

where $o_n^{(i,j)}$ is an observation and $u \in [0, 1]$ is a constant value in order to weight observations so less as they are old. The value for observation can be computed from the relative distance variation between i and j, scaled with radio-range:

$$\Delta dr_n^{(i,j)} = \frac{d_n^{(i,j)} - d_{n-1}^{(i,j)}}{radio\_range} \tag{4}$$

where $radio\_range$ is the maximum distance where two nodes can communicate with each other.

It is straightforward to prove $\Delta dr_n^{(i,j)}$ to range in $[-1, 1]$ interval. This range is not suitable for bayesian filter since observations should be between 0 and 1. So we map the value in Equation 4 onto the suitable range [0,1] as follow[1]:

$$o_n^{(i,j)} = \frac{\Delta dr_n^{(i,j)} + 1}{2} = \frac{d_n^{(i,j)} - d_{n-1}^{(i,j)}}{2 * radio\_range} + \frac{1}{2} \tag{5}$$

In sum, our bayesian approach estimates the relative percentage of radio-range which nodes walk in a time T about a node i with respect to j. This percentage is mapped onto a [0,1] range so that values greater than 0.5 mean nodes drift apart and smaller values mean nodes move closer. If the value 0.5 is estimated, then node i is estimated not to move with respect to j.

Possibly $o_n^{(i,j)}$ can miss in the cycle n. It could miss because the distance between i and j is greater than radio-range or the packet sent by i is lost or delivered late(MANETs are quite unreliable). In these cases we assume $o_n^{(i,j)} = 1$, meaning nodes will not be in radio-range in the next clock cycle.

The parameters α and β are the inputs for Beta-distribution $Beta(\alpha, \beta)$ where the expectation $\theta_{n+1}^{(i,j)} = \mathbb{E}\Big(Beta(\alpha_{n+1}^{(i,j)}, \beta_{n+1}^{(i,j)}) + 1\Big)$ is the variation of the distance between i and j in radio-range percentage which will be estimated in $(n + 1)$-th clock cycle.

By taking into account Equation 5 we can estimate distance between nodes i and j at the beginning of $(n + 1)$-th cycle. That can be done from such an equation where the observation term $o_n^{(i,j)}$ is replaced with the value $\theta_{n+1}^{(i,j)}$ estimated from observations. So, distance is estimable as follow:

$$\begin{aligned} \widetilde{d}_{n+1}^{(i,j)} &= d_n^{(i,j)} + \widetilde{\Delta d}_n^{(i,j)} \\ &= d_n^{(i,j)} + (2\theta^{(i,j)} - 1) * radio\_range \end{aligned} \tag{6}$$

From a theoretical viewpoint $o_n^{(i,j)} = o_n^{(j,i)}$ and $d_n^{(i,j)} = d_n^{(j,i)}$; so, it should be $\widetilde{d}_{n+1}^{(i,j)} = \widetilde{d}_{n+1}^{(j,i)}$. For all practical purposes, we have to consider $\widetilde{d}_{n+1}^{(i,j)} \neq \widetilde{d}_{n+1}^{(j,i)}$ since clocks are not synchronized and, therefore, information can miss or differ (because collected in different moments).

Therefore, estimated distance $\widetilde{d}_{n+1}^{i,j}$ is computed by considering both $\widetilde{d}_{n+1}^{(j,i)}$ and $\widetilde{d}_{n+1}^{(i,j)}$, through different weights:

$$\widetilde{d}_{n+1}^{i,j} = rel^{(i,j)} * \widetilde{d}_{n+1}^{(i,j)} + rel^{(j,i)} * \widetilde{d}_{n+1}^{(j,i)}$$

where $rel^{(i,j)}$ is a factor for estimation reliability and it is inversely proportional to $\sigma^{(i,j)} = \sqrt{Var(Beta\big(\alpha^{(i,j)}, \beta^{(i,j)}\big))}$:

$$rel^{(i,j)} = \frac{\frac{1}{\sigma^{(i,j)}}}{\frac{1}{\sigma^{(i,j)}} + \frac{1}{\sigma^{(j,i)}}} = \frac{\sigma^{(j,i)}}{\sigma^{(i,j)} + \sigma^{(j,i)}}.$$

2.3. Connected Components Computation

Disconnection prediction depends on a parameter γ which stands for the fraction of the radio-range the predictive technique doesn't signal a disconnection anomaly[2]. Let be $P(disc_{n+1}^{(i,j)}) = P(\widetilde{d}_{n+1}^{(i,j)} \geq \gamma radio\_range)$; two nodes i

[1] If a node has entered in this cycle we assume $o_n^{(i,j)} = 0.5$, i.e., it is not moving.

[2] As an example, in IEEE 802.11 with 100 meters of radio-range, if γ equal to 0.3 means that for a communication distance of 70 meters the prediction algorithm signals a probable disconnection.

and j are predicted going to disconnect if and only if

$$rel^{(i,j)} * P(disc_{n+1}^{(i,j)}) + rel^{(j,i)} * P(disc_{n+1}^{(j,i)}) > \frac{1}{2} \quad (7)$$

i.e. two nodes i and j are estimated disconnecting if it is more probable their distance to be greater than $\gamma radio\_range$ rather than distance to be smaller than such a value. We have judged useless to let $\frac{1}{2}$ parametric: smaller values would cause prediction to be more conservative. Indeed, we can tune conservativeness by lowering γ (i.e. the fraction of radio-range which disconnections are not predicted in). If we consider Equation 6, then:

$$\begin{aligned} P(disc_{n+1}^{(i,j)}) &= P(\frac{d_n^{(i,j)}}{radio\_range} + (2\theta^{(i,j)} - 1) \geq \gamma) \\ &= P(\theta^{(i,j)} \geq \frac{1+\gamma}{2} - \frac{d_n^{(i,j)}}{2 * radio\_range}) \end{aligned} \quad (8)$$

Where the last term in the Equation 8 is directly computable from the estimated beta distribution:

$$P(\theta^{(i,j)} > k) = \int_k^1 Beta\big(\alpha^{(i,j)}, \beta^{(i,j)}\big)$$

Once the link losses are predicted, we can compute easily connected components (i.e., sets of nodes that are predicted to be connected). On the basis of the connected components, disconnection anomalies are identified by the monitor and notified either to the nodes (if a local management strategy is adopted) or to the upper layers of the coordination middleware (if a global management strategy is adopted). Connected components are computable through "The Mobile Gamblers Ruin Algorithm" described in [9], where an edge between couples of nodes in the connection graph exists if the Equation 7 is false.

If we compare this approach to the one in [9], its advantages are threefold. Firstly, we estimate future distances between couple of devices (and, thus, disconnections) by considering the distance trend (Δd), whereas the other approach estimates distance as the weighted mean of older distances. Secondly, we consider both distances collected by i about j and by j about i where the previous approach takes into account just one, assuming both of them equal (indeed they may be different). Finally, the previous approach considers only the expected value, instead of the overall distribution of possible values. So, it would return, in the extreme case, the same probability for the uniform distribution between [0,1] and for the Beta degenerating in the Dirac impulse centered in 0.5. Clearly, the disconnection probability for uniform distribution is higher than for Dirac impulse. In the bayesian approach introduced here, we improve predictions, since we consider the whole distribution of possible values.

3. The Experimentation

The test bed setting. We implemented the bayesian algorithm on actual devices since it is the best way to check

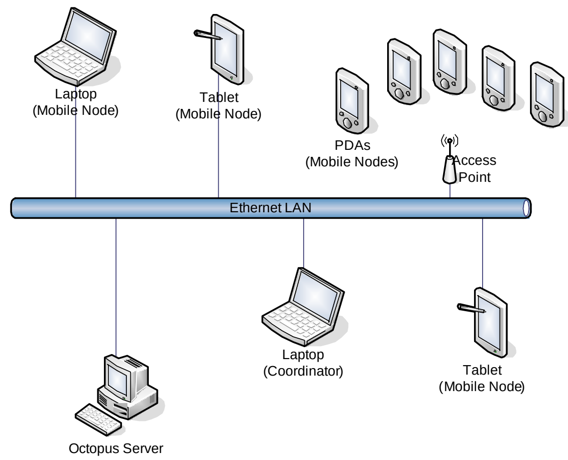

Figure 2. The test bed setting.

and verify whether the algorithm is practically feasible. We coded in MS Visual C# .NET as it enables to write applications once and deploy them on any device for which a .NET Framework exists for (PCs and PDAs included). Of course, field testing may be expensive both in terms of resources and time: several persons are required to deploy in a wide area and this might need time (and, thus, cost) to prepare them. Furthermore, field testings do not provide a controlled environment and it should be used just as final user validation of the system. We used OCTOPUS [6] an emulator specifically targeted for MANET. OCTOPUS keeps a virtual map with virtual nodes which are bound to actual devices. Devices and OCTOPUS are deployed on the same LAN. When a node wants to send packets to another, those are actually captured by OCTOPUS which plays the gateway rule. OCTOPUS analyzes the map and, if the sender and receiver are in radio-range, it forwards to the destination. The good OCTOPUS' feature is that software on board of devices is not aware about the presence of an emulator. The test bed is show in Figure 2 and consists of ten machines (PCs and PDAs). One of them hosts OCTOPUS with the virtual map and, so, it does not represent a real mobile node. All machines, except the first one, are bounded to different virtual nodes of the virtual map.

The virtual map is 400x300 meter wide and the radio range is set up as 100 meters. At the beginning, nodes are located into the virtual map in a random fashion but, anyway, so to form one connected component. Afterwards, each S seconds, every node chooses a point (X,Y) in the map and begin heading towards at a speed of V m/s. Both S and V are Gaussian random variables: the mean and variance are, respectively, 450 and 40 seconds for S and 3 and 1.5 m/s for V. The couple (X,Y) is chosen uniformly at random in the virtual map. Of course, the devices used in tests do not actually move. In our emulated environment nodes move only in the virtual map. For this purpose, devices send particular commands to a specific OCTOPUS

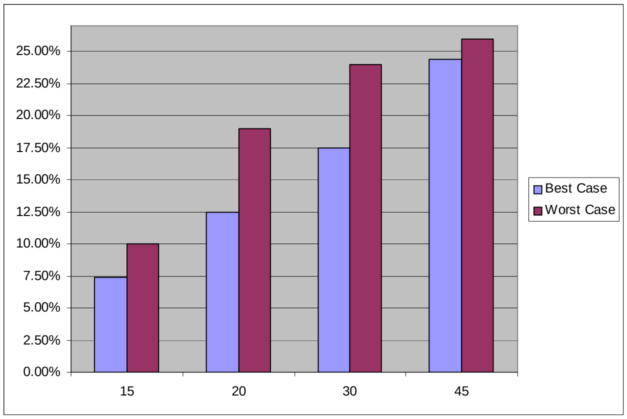

Figure 3. The smallest and largest measured error in percentage, changing clock periods.

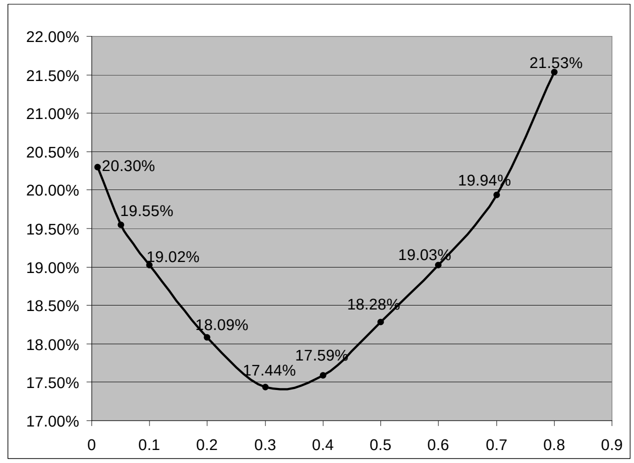

Figure 4. The measured error in percentage, changing the weight of past observations.

socket in order to instruct it to move corresponding nodes in the map.

Results. The first set of experiments has been intended to verify which error in percentage is obtained for different values of clock period T. The error is here defined as the gap between estimated $\widetilde{d_n}$ distances at $(n-1)$-th clock cycle and actual measures d_n at n-th cycle. The value is scaled with respect to radio-range: $\frac{|\widetilde{d_n}-d_n|}{radio\_range}$.

The Figure 3 shows the outcomes for clock periods equal to 15, 20, 30 and 45 seconds. We used $u = 0.5$ and we performed ten running tests for each clock period. Every test has been 30 minutes long. The results show, of course, the error percentage is as larger as clock period increases. Probably the most reasonable value for real scenarios is 30-45 seconds (smaller values are not practically feasible since the MANET would be probably overloaded by “distance” messages). Please consider the greatest clock period we tested: the error ranges between 24.34% and 26.8% (i.e., roughly 25 meters). These values are good: indeed, if we set $\gamma = 0.75$ (i.e., disconnections are predicted when nodes are more than 75 far), we would be sure to predict every actual disconnections. Moreover, for $\gamma = 0.75$ the algorithm does not appear so conservative even if we could get false negatives.

Afterwards, in a second test set, we fixed clock period to 30 seconds, testing for u equal $0.01, 0.05, 0.1, 0.2, \ldots, 0.8$. We even tripled the frequency which nodes start moving with. The outcomes are depicted in Figure 4 where x-axis corresponds to u values and y-axis to the error percentage. The trend is parabolic: the minimum is obtained for $u = 0.3$ where error is 17.44% and the maximum is for $u = 0.8$ where error is 21.54%. Little values for u means that the past is scarcely considered whereas large values means the past is strongly taken into account. This matches our expectation: we get the best result for intermediate values. That is to say the best tuning is obtained whether we consider the past neither too little nor too much.

4. Related Work

Much research on mobility prediction has been carried on (and still it is in progress) for cellular phone systems [11, 2]. These approaches are based on Markov models, which predict a mobile user future on the basis of its current and past locations. The targeted issue of these approaches is to predict whether a mobile user is leaving a cell (crossing cell boundaries) and which new cell it is going into. Such an information is then used for channel reservation in the cell a mobile user is going to enter. Anticipating reservation should lower the probability of a call to be dropped during handoff [3] caused by the absence of free channel for the call in the new cell.

In order to make these predictions, proposed algorithms generally include a combination of timing, speed, movement information, and distance information between base stations and mobile users. All communications between the backbone systems, which calculate the prediction, and mobile users occur through base stations, which are the centers of star topologies. In such systems, the wireless-network topology does not change over time. Communications between a base station and all mobile users in the cell are always one-hop, and communications between mobile users do not occur directly but through base stations.

[3] In cellular telecommunications, the term *handoff* refers to the process of transferring an ongoing call or data session from one channel connected to a core network or cell to another

Our scenario is different: MANETs can be arranged in several network topologies that change over time together with multihop path among mobile devices. Furthermore, MANETs have no fixed infrastructures (i.e., base stations). Our approach takes into account such network peculiarities. It does employ a centralized model to predict connection (and disconnection) of devices from the rest of the MANETs and to overcome disconnections (through the functionalities of the coordinator), but these operations are enacted differently compared to the approaches used in cellular-network systems; indeed, it takes into account the knowledge of all distances among mobile users, not only those between base stations and mobile devices.

In the literature some approaches are investigated to predict the state of connectivity of the nodes of a MANET; in [12] the MANET is considered as a combination of clusters of nodes and it is studied the impact (i.e., the performances) of two well defined mobility prediction schemes on the temporal stability of such clusters; unlike our approach the authors uses pre-existing predictive models while the the novelty of our approach consists in the formalization of a new model based on Bayesian filtering techniques.

Our research group has already worked on disconnections prediction in MANET [9]. The technique described in the present paper relies on a more complete and efficient statistical approach based on Bayes' filters. A deeper discussion of the enhancements of this new approach has been previously presented in Section 2.3.

Bayesian filters can be used in all fields where the entities of a system cannot directly know its (dynamic) state and they are forced to learn it by an estimation from samples. In [4], it is used to predict the behaviour (regarding security aspects) of nodes in a MANET in order to make a reputation system (applyed to peer-to-peer mobile ad-hoc network) both robust against false ratings and efficient at detecting misbehavior. In [10] the bayesian approach is used in its various formalizations (kalman, particles filters, etc.) for the problem of robot location estimation. In particular it assumes no location sensor to take perfect measurements and that enables its use independently from the type of sensors and from the used technology (e.g., GPS, infrared, etc.). Then, for each point p_i and each robot r_i, the technique gives the probability for r_i to stay in p_i. This approach cannot be easily used to compute when nodes are going to disconnect.

5. Conclusions

In this paper we have proposed and experimented a novel techniques for predicting disconnections in MANETs. This is a basic problem when building coordination middleware for mobile scenarios, as we argue that whichever kind of remedial action (either local or global) need to be enforced in advance with respect to the effective disconnection, and therefore such a disconnection needs to be predicted.

This work is the basis of the development of a coordination system for MANETs in emergency management – cfr. the WORKPAD project (`http://www.workpad-project.eu`) we are currently involved, as our plans are to build a global management approach in which, after predicting disconnections, the coordination middleware instructs devices on how to arrange differently their tasks in order to keep the MANET connected. We plan to use executing monitoring techniques [7] for building such a layer.

References

[1] D. P. Agrawal and Q. A. Zeng. *Introduction to Wireless and Mobile Systems*. Thomson Brooks/Cole, 2003.

[2] I. F. Akyildiz, J. S. M. Ho, and Y. B. Lin. Movement-based location update and selective paging for pcs networks. *IEEE/ACM Transactions on Networking*, 4(4):629 – 638, 1996.

[3] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[4] S. Buchegger and J. Y. Le Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *Proc. 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2003.

[5] T. Catarci, F. De Rosa, M. de Leoni, M. Mecella, M. Angelaccio, S. Dustdar, A. Krek, G. Vetere, Z. M. Zalis, B. Gonzalvez, and G. Iiritano. Workpad: 2-layered peer-to-peer for emergency management through adaptive processes. In *Proc. 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2006)*, 2006.

[6] F. D'Aprano, M. de Leoni, and M. Mecella. Emulating mobile ad-hoc networks of hand-held devices. the octopus virtual environment. In *Proc. Third International ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments*, 2007.

[7] G. De Giacomo, R. Reiter, and M. Soutchanski. Execution monitoring of high-level robot programs. In *Proc. KR 1998*, pages 453–465, 1998.

[8] M. de Leoni, F. De Rosa, and M. Mecella. MOBIDIS: A Pervasive Architecture for Emergency Management. In *Proc. 4th International Workshop on Distributed and Mobile Collaboration (DMC 2006) (at WETICE 2006)*, 2006.

[9] F. De Rosa, A. Malizia, and M. Mecella. Disconnection prediction in mobile ad-hoc networks for supporting cooperative work. *IEEE Pervasive Computing*, 4(3):62 – 70, 2005.

[10] D. Fox, J. Hightower, L. Lao, D. Schulz, and G. Borriello. Bayesian filters for location estimation. *IEEE Pervasive Computing*, 2(3):24 – 33, 2003.

[11] B. Liang and Z. J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM Transactions on Networking*, 11(5):718 – 732, 2003.

[12] A. Venkateswaran, V. Sarangan, N. Gautam, and R. Acharya. Impact of mobility prediction on the temporal stability of manet clustering algorithms. In *Proc. 2nd ACM International Workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, pages 144–151, New York, NY, USA, 2005. ACM Press.