Journal of Computer Science 8 (5): 721-730, 2012
ISSN 1549-3636

Efficient Broadcasting in Mobile Ad Hoc Networks Using Context Aware Adaptive Routing Protocol during Network Partitions

[1]Bavithiraja, S.V.M.G. and [2]R. Radhakrishnan
[1]Department of CSE,
CMS College of Engineering and Technology, Coimbatore-32, India
[2]Sri Ramakrishna Engineering College, Coimbatore-22, India

Abstract: Problem statement: In mobile ado network, broadcasting is a common operation for route establishment and for sending control and emergency messages. Reliable Broadcasting in mobile ad hoc networks requires delivery of messages from different sources to all the nodes of the network within bounded time. The nodes are highly mobile and the network is highly dynamic and decentralized. Most of the existing routing protocols in MANET have the assumption that a path exists between the sender and the receiver. But the decentralized mobile ad hoc network is characterized by frequent network partitions. Providing reliable broadcasting is a challenging task. **Approach:** In study, a new context aware adaptive routing protocol is designed for broadcasting. The protocol is based on the idea of exploiting nodes as carriers of messages among network partitions to achieve guaranteed delivery. The choice of the best carrier is made dynamically by using Kalman filter based prediction techniques and utility theory. **Results:** Simulation results show that the proposed protocol achieves better packet deliver ratio with reduced control overhead and packet loss during network partition. **Conclusion:** The proposed routing protocol chooses the carrier node dynamically based on the context information, it provides reliable broadcasting to all the nodes with minimum packet loss and control overhead.

Key words: Mobile Ad Hoc Network (MANET), Destination-Sequenced Distance-Vector (DSDV), Routing Information Protocol (RIP)

INTRODUCTION

In wireless communication systems, there will be a need for the rapid deployment of independent mobile users. Significant examples include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts and military networks. Such network scenarios cannot rely on centralized and organized connectivity and can be conceived as applications of Mobile Ad Hoc Networks (MANET).

A Mobile Ad Hoc Network (MANET) (Perkins, 2008) is a set of nodes communicating with each other via multi-hop wireless links. Each node can directly communicate with only those nodes that are in its communication range. Intermediate nodes forward messages to the nodes that are more than one hop distance from the source. Since the nodes are mobile, the topology of the network is constantly changing. Broadcasting is the process in which one node sends a packet to all other nodes in the network. Broadcasting is often necessary in MANET routing protocols. The broadcast is spontaneous. Any mobile host can issue a broadcast operation at any time.

A new routing protocol is proposed for broadcasting in mobile ad hoc networks. The routing protocol uses prediction to allow the efficient routing of messages to the recipient. A host willing to send a message to a recipient, or any host in the multi hop path to it, uses a Kalman Filter prediction and multi-criteria decision theory (Keeney and Raiffa, 1993) to choose the best next hop (or carrier) for the message. The decision is based on the mobility of the host (a highly mobile host is a good carrier as it meets many hosts) and its past collocation with the recipient (we implicitly assume that past collocation indicates that the host will meet the recipient again in the future). The protocol does not assume any previous knowledge of the routes of the hosts like other approaches, such as the Message Ferrying (Zhao *et al*., 2004), that rely on the a priori knowledge of the routes of the special hosts carrying the information. Moreover, the protocol is based on a single copy of the message in the system, instead of

Corresponding Author: Bavithiraja, S.V.M.G., Department of CSE, CMS College of Engineering and Technology, Coimbatore-32, India Tel: 9952208446

having multiple replicas. Any geographical information's are not exploited such as GPS coordinates (Wu and Watts, 2002; Garbinato *et al*., 2009).

MATERIALS AND METHODS

Design of new routing protocol:

Overview: The proposed protocol is based on the idea of exploiting nodes as carriers of messages among network partitions to achieve delivery. The choice of the best carrier is made using Kalman filter based prediction techniques and utility theory.

In particular, the protocol assumes that only information a host knows is, its position which is related to its logical connectivity. Also assumes that a host is not aware of its absolute geographical location and of the location of those to whom it might deliver the message. Although this information could potentially be useful, there might also be battery implications of its use which might be unacceptable (for example, because of the energy requested to operate a GPS device). Another basic assumption is that the hosts present in the system cooperate to deliver the message. In other words, the protocol does not consider the case of hosts that may refuse to deliver a message or that act in a Byzantine manner (Lindgren *et al*., 2003).

The design goal of the proposed protocol is to support communication in intermittently connected mobile ad hoc networks. The key problem solved by the protocol is the selection of the carrier. The solution is based on the application of forecasting techniques and utility theory for the evaluation of different aspects of the system that are relevant for taking routing decisions. Let us now consider the key aspects of the protocol. The protocol is able to deliver messages synchronously (i.e., without storing them in buffers of intermediate nodes when there are no network partitions between sender and receiver) and asynchronously (i.e., by means of a store-and-forward mechanism when there are partitions). The delivery process depends on whether or not the recipient is present in the same connected region of the network (cloud) as the sender. If all nodes are currently in the same connected portion of the network, the message is broadcasted using an underlying synchronous routing protocol to determine a forwarding path (Bantz *et al*., 2003). If a message cannot be broadcasted synchronously, the best carriers for a message are those that have the highest chance of successful delivery, i.e., the highest delivery probabilities.

The message is sent to the host with the highest one using the underlying synchronous protocol. For the remaining nodes (those nodes that are in a separate cloud), the message is broadcasted by the carrier node. In order to understand the operation of the protocol, consider the following scenario in which two groups of nodes are connected as in Fig. 1. Host H1 wishes to broadcast a message to all the hosts in the network. Dynamic Destination-Sequenced Distance- Vector (DSDV) (Perkins and Bhagwat, 1994) is used to support synchronous routing to all connected hosts. For hosts that are not connected, synchronous routing is not possible. Suppose the delivery probabilities for each connected host can be determined as shown in Fig. 1, the host possessing the best delivery probability can be used as a carrier, in this case Host H4. Consequently, the message is sent to H4, which stores it. After a certain period of time, H4 moves to the other cloud (as in Fig. 2). Since a connected path between H4 and second cloud now exists, the message is delivered to all hosts in the second cloud. Using DSDV, the host H4 is able to broadcast the message shortly after joining the cloud, since this is when it will receive the routing information related with second cloud.

Delivery probabilities are synthesized locally from context information. Context may be defined as the set of attributes that describe the aspects of the system that can be used to drive the process of message delivery. An example of context information can be the change rate of connectivity, i.e., the number of connections and disconnections that a host experienced over the last T seconds (Pasztor *et al*., 2007). This parameter measures relative mobility and, consequently, the probability that a host will encounter other hosts. Assume a proactive routing protocol, every host periodically sends both the information related to the underlying synchronous routing (in DSDV this is the routing tables with distances, next hop host identifier and a list containing its delivery probabilities for the other hosts. When a host receives this information, it updates its routing tables. With respect to the table for asynchronous routing, each host maintains a list of entries, each of which is a tuple that includes the fields (destination, best Host, delivery Probability). When a host is selected as a carrier and receives the message, it inserts it into a buffer. The size of this buffer is fundamental and represents a trade-off between storage overhead and likely performance. If the buffer overflows, messages will be lost (Durbin and Koopman, 2001).

Prediction and evaluation of context information: The routing protocol is optimized by using predicted future values of the context attributes for making routing decisions, instead of using the available current context information as it is, so to have a more accurate estimation of the trend of the time series associated to each context dimension. For example, in the case of patterns of collocation, a host H_A currently not collocated with a host H_B may be considered of scarce utility for acting as a carrier for H_B if evaluated only this instant of time.

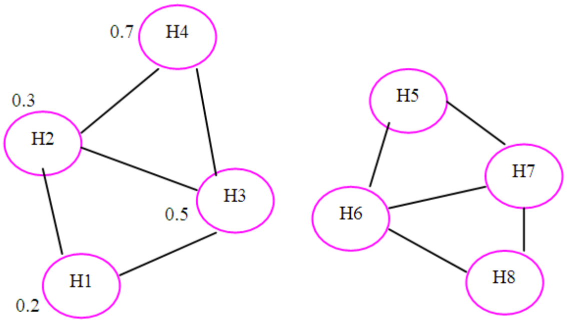

Fig. 1: Two connected clouds, with associated elivery probabilities for message transmission from node H1

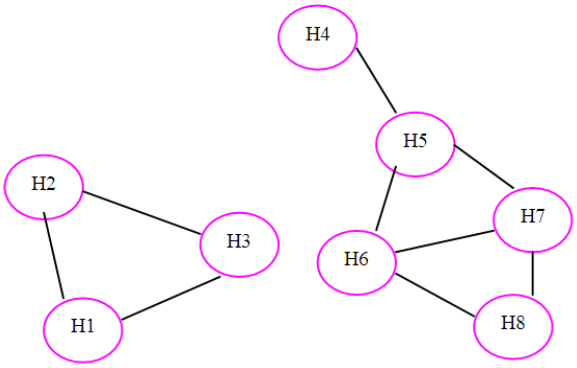

Fig. 2: Host H4 carrying the message, joins the second cloud

However, H_A may have been collocated with H_B for the past 3 h and, therefore, its likelihood of being collocated again, given the assumptions of the model, are high and should be represented accordingly (Musolesi and Mascolo, 2006; Li and Rus, 2000).

The process of prediction and evaluation of the context information can be summarized as follows:

- Each host calculates its delivery probabilities for a given set of hosts. This process is based on the calculation of utilities for each attribute describing the context. Then the future values of these utilities are predicted and composed using multi-criteria decision theory (Keeney and Raiffa, 1993) in order to estimate an overall delivery probability. The calculated delivery probabilities are periodically sent to the other hosts in the connected cloud as part of the update of routing information
- Each host maintains a logical forwarding table of tuples describing the next logical hop and its associated delivery probability, for all known destinations
- Each host uses local prediction of delivery probabilities between updates of information. The prediction process is used during temporary disconnections and is carried out until certain accuracy can be guaranteed

Local evaluation of context information: Each host calculates its delivery probability locally, given observations related to the various context attributes. Therefore, the key problem is to measure and combine the attributes. The delivery probabilities are calculated by evaluating the utility of each host as potential carrier for a message.

There are several techniques for assigning an overall utility given the multiple dimensions of the context. A possible method is to use goal programming, exploiting the so-called preemptive methodology. With respect to a single attribute, our goal is to maximize its value. The optimization process is based on the evaluation of one goal at a time so that the optimum value of a higher priority goal is never degraded by a lower priority goal (Taha, 2006).

This technique is too simplistic because, in general, the decision problem involves multiple conflicting objectives (Keeney and Raiffa, 1993). For example, considering both the battery energy level and the rate of change of connectivity, it may happen that the host characterized by the highest mobility has scarce residual battery energy and vice versa. In general, maximization across all parameters will not be possible and, instead, we must trade off the achievement of one objective (i.e., the maximization of a single attribute) against others.

The context information related to a certain host can be defined using a set of attributes (x_1, x_2,.x_n). Those attributes denoted with a capital letter (e.g., X_1) refer to the set of all possible values for the attribute, whereas those denoted with a lower case letter (e.g., x_1) refer to a particular value within this set. Examples of a generic attributes Xi can be the mobility of the hosts or its battery level. For instance, the value x_i of the attribute battery level may be 0.99 (i.e., battery almost full). In the case of mutually preferentially independent attributes x_1, x_2 ….$x_{n,}$ that is to say those charactérized by the same degree of significance, the sum of the attributes is adequate as a means of combining those attributes Eq. 1:

$$U(x_1, x_2, \ldots x_n) = \sum_{i=1}^{n} U_i(x_i) \tag{1}$$

where, U_i is a utility function over x_i.

Our aim is to maximize each attribute, in other words, to choose the host that presents the best trade-off between the attributes representing the relevant aspects of the system for the message delivery. To solve this problem, we apply the so-called Weights method (Keeney and Raiffa, 1993). The combined goal function used in the Weights method can be defined as Eq. 2:

$$\text{Maximize}\left\{ f\left(U\left(x_i\right)\right) = \sum_{i=1}^{n} w_{i U_i}\left(x_i\right) \right\} \qquad (2)$$

where, w_1, w_2, …w_n are significance weights reflecting the relative importance of each goal.

We exploit these results for the composition of the utilities in the protocol related to the different context dimensions (given their mutual independence). In this case, the solution is very simple, since it consists in the evaluation of the function f (U_1…. U_n) using the values predicted for each host and in the selection of the host i with the maximum such value.

Definition of the attributes of the utility functions: Knowledge about the current values of these context attributes is helpful, but only to a limited extent. What really matters are the values the attributes which are likely to assume in the future. We compute these predicted values using techniques based on Kalman filters (Kalman, 1960). These techniques do not require the storage of the entire past history of the system and are computationally lightweight, making them suitable for a resource-scarce mobile setting. It is useful to focus on the use of the predicted values of the attributes for the calculation of the utility of each host as message carrier (Pasztor *et al*., 2007).

In the implementation of protocol, we focus on two attributes; the change degree of connectivity and the future host collocation, because these are the attributes most relevant to the ad hoc scenario taken into consideration. However, the framework is general and open to the inclusion of any other context attribute, given the underlying assumption of their mutual independence. Other possible context dimensions are memory availability, group membership (i.e., two hosts of the same social group are more likely to be collocated), battery level and so on.

The change degree of connectivity of a host h is Eq. 3:

$$U_{cdc_h}\left(t\right) = \frac{\left|n\left(t-T\right) \cup n\left(t\right)\right| - \left|n\left(t-T\right) \cap n\left(t\right)\right|}{\left|n\left(t-T\right) \cup n\left(t\right)\right|} \qquad (3)$$

where, n (t) is h's neighbor set at time t. The formula yields the number of hosts that became neighbors or disappeared in the time interval [t-T, t], normalized by the total number of hosts met in the same time interval. A high value means that h recently changed a large number of its neighbors.

The collocation of h with a host i is calculated as follows Eq. 4:

$$U_{col_{h,i}}\left(t\right) = \begin{cases} 1 \text{ if the host h is colocated with host i} \\ 0 \text{ otherwise} \end{cases} \qquad (4)$$

A value of 1 means that h has been colocated with i at time t. These values are fed into Kalman filter predictors, which yield the predictions $\hat{U}_{cdc_h}$ and $\hat{U}_{col_{h,i}}$ of these utilities at time t + T. These are then composed into a single utility value using results from multi-criteria decision theory described above, as follows Eq. 5:

$$U_{h,i} = W_{cdc_h} \hat{U}_{cdc_h} + W_{col_{h,i}} \hat{U}cdc_h \qquad (5)$$

which, represents how good of a node h is for delivering messages to i.

The choice of using predicted values and not current values of the attributes is evident in the case of collocation. For example, consider two hosts that have disconnected for just 10 sec after being connected for a long period of time. If only considered the current status, the value of the utility function related to collocation would be 0. Instead, since the hosts have been collocated for long time in the past, according to our assumptions, they will be likely collocated again the future. The value 0 does not provide a correct measure of the probability of future collocation of the two hosts. On the contrary, the output of the Kalman filter will be close to 1.

The weights w denote the relative importance of each attribute. Their value depends on the application scenario in which they are used. The values of these weights are the same for every host; in other words, the utility composition function is the same for all the nodes of the system (Zhang, 2006).

Automatic adaptation of the utility functions: As it stands, the utility function weights are fixed in advance, reflecting the relative importance of the different context attributes. However, such a formulation is still too static, since it fails to take into account the values of the attributes. Thus, for example, a small drop in battery voltage may be indicative of the imminent exhaustion of the battery; consequently, it would be useful to reduce the weight of this attribute non-linearly to reflect this (Musolesi and Mascolo, 2007).

In general, the weights of each parameter are adapted dynamically and in ways that are dependent on the values of those parameters. In other words, a runtime self-adaptation of the weightings is used for this evaluation process that could be categorized as a typical autonomic mechanism (Jain *et al*., 2004). A simple solution to this problem is the introduction of adaptive weights a_i into the previous formula, in order to modify the utility function according to the variation of the context Eq. 6:

$$\text{Maximize}\left\{ f\left(U\left(x_i\right)\right) = \sum_{i=1}^{n} a_i\left(x_i\right) w_i U_i\left(x_i\right) \right\} \qquad (6)$$

where, $a_i(x_i)$ is a parameter that may itself be composite. This parameter can be defined to have three important aspects that help to determine its value, though the model could easily be expanded to incorporate other aspects deemed to be of importance:

- Criticality of certain ranges of values, arange_i (x_i)
- Predictability of the context information, apredability_i (x_i)
- Availability of the context information, apredability_i (x_i)

We now compose the a_i weights as factors in the following formula Eq. 7:

$$a_i(x_i) = a_{\text{range}_i}(x_i).a_{\text{predictability}_i}(x_i).a_{\text{availability}_i}(x_i) \qquad (7)$$

Adaptive weights related to the ranges of values assumed by the attributes: We can model the adaptive weights arange_i (x_i) as a function in the domain [0, 1]. For example, with respect to the battery energy level (modeled using the percentage of residual battery energy), we would use a monotonically decreasing (though not necessarily linear) function to assign a decreasing adaptive weight that is, in turn, used to ensure that the corresponding utility function decreases as the residual energy tends towards zero.

Adaptive Weights Related to the Predictability of the Context Information: It may happen that the forecasting model is not able to provide accurate predictions for a certain time series related to a given attribute. There are many different methods to evaluate the predictability of a time series Eq. 8 (Sarafijanovic-Djukic *et al*., 2006):

$$a_{\text{predictability}} = \left\{ \begin{array}{l} 1 \text{ if the context information is currently predictable} \\ 0 \text{ if the context information is not currently predictable} \end{array} \right\} \quad (8)$$

We prefer to adopt an approach based on two discrete values (0 and 1) rather than one based on continuous values (i.e., an interval between 0 and 1), since the latter would be only based on a pure heuristic choice and not on any sound mathematical basis. In other words, it is very difficult to map different scales of predictability into the values of apredictability_i Eq. 9:

$$a_{\text{availability}_i} = \left\{ \begin{array}{l} 1 \text{ if the context information is currently available} \\ 0 \text{ if the context information is not currently available} \end{array} \right\} \quad (9)$$

It is unreasonable to assume that all context attributes have the same degree of availability. Thus, we expect to have a time varying set of attributes available whose values are known (Keeney and Raiffa, 1993; Chatfield, 2004; Brockwell and Davis, 2002).

Formally, to date, it is implicitly assumed that a static set of attributes is defined. However, using this approach, a new attribute values can be dynamically incorporated, simply by assuming that they were always there, but had zero weight for $a_{\text{availabilityi}}$. For example, if an operating system is not able to provide information about the current battery level of a device, the value of $a_{\text{availability}}$ is set to 0. This may also be due to an erroneous reading of a parameter (for example, in the case of the change degree of connectivity, because the wireless interface has been switched off temporarily).

Routing tables: We have seen how each host calculates its delivery probability by assembling predictions related to different context attributes. We now describe how this information is circulated in the network.

Format of the routing table entries: The delivery probability information is piggybacked on the synchronous routing table information. Each host maintains a routing and context information table used for asynchronous and synchronous (DSDV) routing. Each entry of this table has the following structure:

(targetHostId, nextHopId, dist,
bestHopHostId, deliveryProb)

The first field is the recipient of the message, the second and the third are the typical values calculated in accordance with the DSDV specification, whereas the fourth is the identifier of the host with the best delivery probability, the value of which is stored in the last field. These routing tables are used both for synchronous and asynchronous delivery. They store information used for routing messages inside a cloud (i.e., the field's nextHopId and distance) and for the selection of the best carrier (i.e., the field's bestHopHostId and deliveryProb). A distance equal to 16 is considered infinite and the host is treated as unreachable using

DSDV. We choose 16 since this is the classic Routing Information Protocol (RIP) infinite (Brockwell and Davis, 2002). However, this is a parameter that can be tuned according to user requirements. In a scenario characterized by high average host speed a lower value may be used, since the probability that the route will be broken or stale is potentially high.

The value of the field deliveryProb is updated using the last received value. However, the values received by the neighbors are also used to update a corresponding Kalman Filter predictor, one for each entry of the table. The state of the filter is updated using the last received utility from the host bestHopHostId. The filter is used if one or more updates are not received, due for example to a temporary disconnection, to transmission errors (for example interference) or simply because the host has moved away. If an update is not received in a given refresh interval of the filter (that is equal to the routing table transmission interval), the previous output of the filter is used as input (i.e., the filter is, in a sense, short-circuited). The entries are removed after a certain number of updates are not received, since the accuracy of the prediction is clearly decreasing.

Local utilities and update of routing tables: Each node keeps local utilities related to the collocation with other nodes. The routing tables are exchanged periodically with a given transmission interval. When a host receives a routing table, it checks its entries against the ones stored in its routing table. The update of the information related to the synchronous protocol is the standard one of every table-driven protocol: an entry in the routing table of the host is replaced if one related to the same targetHostId and a lower or equal distance is received. It is important to note that we also replace the entry if the distance is the same in order to have fresh information about the route. Instead, as far as the asynchronous delivery protocol is concerned, an entry is replaced only if one related to the same targetHostId and higher or equal delivery probability is received. The entry is removed after a number of missing updates. This also avoids the problem that entries with high probabilities persist in the routing table even if they are stale. When the routing table is full, the entries are replaced starting from the one corresponding to the nodes that are not in reach any more (i.e., that have a value of the distance field equal to 16). Among these entries, the one with the lowest delivery probability is selected. The size of the routing table is limited, since to every entry there is a associated Kalman filter based predictor that has to be updated periodically.

Routing table transmission interval: The routing table transmission interval is another fundamental parameter of the protocol. In fact, routing tables are not only used to exchange routing information, but also for discovery. Routing tables are employed as a sort of beaconing mechanism at application level to keep information about the presence of neighbors. In fact, a host is considered collocated (i.e., the input of the Kalman filter is set to 1), if a routing table related to that host has been received in the last routing table transmission interval; we assume that the frequency of the transmission of routing tables is relatively high in order to provide correct information to the collocation predictor (D'Souza and Jose, 2010).

The update interval of the Kalman filter (i.e., its sample interval) is another fundamental parameter of the protocol. This value should be carefully selected in order to detect changes in the observed context attribute. For example, in the case of host collocation, a low sampling interval in a very dynamic mobile scenario may lead to the fact that hosts passing by will not be detected. For instance, if the relative speed of the two hosts is 20 m sec^{-1} and the transmission range is 200 m sec^{-1}, an update interval greater than 20s may lead to the fact that some hosts will not be discovered. The update interval of the Kalman filter is set to the routing table transmission interval.

Message delivery: The message delivery is done in two ways based on whether the recipient is in reachable area (synchronous delivery) or there is no connected path to the recipient (asynchronous delivery).

Synchronous delivery: When a message has to be sent, if the recipient is reachable synchronously (i.e. an entry with the field TargetHostId exists in the routing table and the associated distance is less than 16), the message is forwarded to the next hop indicated by nextHopId. This forwarding mechanism is the typical one of distance vector protocols. It may happen that the path to a certain host is broken, but, at the same time, the routing table has not yet been updated with the information related to this change, given the propagation delay of routing tables. In this case, the message is forwarded until it reaches the host that has been already notified about the disconnections. This host will then check if the message can be sent using the asynchronous delivery mechanism (i.e., an entry for the selection of the best carrier exists in its routing table).

Asynchronous delivery: If a connected path to the recipient does not exist (i.e., the value of distance is equal or greater than 16), the message is forwarded to the host with the highest value of delivery probability (expressed by deliveryProb). In order to reach the carrier, DSDV is used. In other words, the entry having the value of the key targetHostId equal to bestHopHostId is used to forward the message. As the network is dynamic, it may happen that the carrier is unreachable, since, in the meanwhile, it has left the connected cloud. In this case, if the information about the disconnection has reached the sender, the entry related to the best carrier is removed (set to an invalid state designated by 0). In order to avoid the propagation of stale routes, we use sequence numbers for the routing tables like in DSDV. If this information has not been propagated yet to the sender, the intermediate host aware of the topology change will try to resend the message.

(Re-) transmissions: Periodically, for each message in its buffer, a host checks its routing table. The message is then forwarded synchronously to the recipient or to a carrier if a corresponding entry is present in the routing table. If no entry is present, the message stays in the buffer. The number of the retransmissions is another key configuration value that we have measured and tested during the performance evaluation of the protocol.

Each node also maintains a list of its utilities for a certain set of hosts. In particular, each node keeps a list of the local utilities related to the collocation with other hosts and one related to its change degree of connectivity. Periodically, these utilities are composed and the resulting ones are checked against the utilities stored in the local routing table. If the utility of the host is higher of the one currently maintained in the table, the latter is replaced. The value of the utilities is updated before comparing it with the entries of the local routing table.

Prediction of the context information attributes using kalman filter prediction techniques: Kalman filter forecasting techniques (Kalman, 1960) are used for the prediction of the future values of the context attributes and of the delivery probabilities in the local routing tables, if updates are not received.

Overview: Kalman filter prediction techniques were originally developed in automatic control systems theory. These are essentially a method of discrete signal processing that provides optimal estimates of the current state of a dynamic system described by a state vector. The state is updated using periodic observations of the system, if available, using a set of prediction recursive equations. Kalman filter theory is used in the routing protocol both to achieve a more realistic prediction of the evolution of the context of a host and to optimize the bandwidth usage. As discussed above, the exchange of context information that allows the calculation of delivery probabilities is a potentially expensive process and unnecessarily so where such information is relatively predictable. If it is possible to predict future values of the attributes describing the context, the delivery probabilities stored in the routing tables can be updated, even if fresh information is unavailable. Fortunately, this prediction problem can be expressed in the form of a state space model. Starting from a time series of observed values that represent context information, a prediction model is derived based on an inner state that is represented by a set of vectors and to add to this both trend and seasonal components (Spyropoulos *et al*., 2005). One of the main advantages of the Kalman filter is that it does not require the storage of the entire past history of the system, making it suitable for a mobile setting in which memory resources may potentially be very limited.

Context predictability: Dealing with the variability and uncertainty is one of the major issues in many networked systems such as mobile ad hoc and delay tolerant networks (Ke and Nenghai, 2010). The decentralization of the control and the movement of the hosts have a great impact on systems topology and, more generally, on their conditions. The protocol heavily relies on the accuracy of the prediction model. There are situations, however, where context cannot be predicted. In these cases, using any prediction based techniques to improve performance of the system is completely ineffective. For this reason, a predictability component is designed that is used to measure the accuracy of the prediction of context information presented in (Musolesi *et al*., 2005). The technique that we adopted is predicated on the analysis of the time series representing the context information and, more specifically on residual analysis (Spyropoulos *et al*., 2005). Given a certain number of measurements of the predictability of the time series, predictability level of a context attribute is defined as the percentage of samples for which the component returns true, in other words, the percentage of samples for which the prediction model is sufficiently accurate given a predefined acceptable error. In our experiments, we consider the predictability of the time series of the collocation between pair of hosts; every sample of the time series is evaluated for the calculation of the predictability level. If the predictability level is under a given threshold, alternative protocols can be used, for example epidemics- inspired approaches (Jones *et al*., 2007).

RESULTS

Performance evaluation:

Simulation scenarios: The simulation of the protocol has been performed using ns2. The various Simulation Parameters are given in Table 1.

Table 1: Simulation parameters

Network size	1000×1000 m
Bandwidth	1Mb
Synchronous routing protocol	DSDV
MESSAGE_PORT	42
BROADCAST_ADDR	-1
Nam Animation Speed	250 Micro seconds
Node Velocity	5 to 40 Milli seconds
Transmission Probability	50
Broadcast Probability	50
Broadcast Delay	0.01 Micro seconds
Hello Reply Delay	0.01 Micro seconds
Nam Animation Speed	250 Micro seconds
Message Size	1500 bytes
Interface queue type	Queue/DropTail/PriQueue
Antenna model	Antenna/OmniAntenna
Max packet in ifq	50

Choice of the parameters: The protocol is simulated using a utility function based on the evaluation of two attributes: (i) the change rate of connectivity and (ii) the probability of being connected with the other cloud in case of partition. All possible values which are assumed in the range had the same importance (i.e., $a_{rangei} = 1$) and that the values of attributes are always available during the simulation (i.e., $a_{availabilityi} = 1$). The values of w_{cdch} and $w_{odch\ i}$ for all the pairs of hosts (h, i) are set to 0.25 and 0.75, respectively. These values ensure the best performance in terms of delivery ratio in the scenario (Johnson and Maltz, 1996; Musolesi and Mascolo, 2006).

Each message has a time to live field that is decreased each time a message is transferred to another host (the initial value being 10). Moreover, in this case, a split horizon mechanism is introduced to prevent messages from being retransmitted unnecessarily. The buffer for each node was set to 50 messages, unless otherwise specified. The number of retransmissions for the 20 host’s scenario was set to 10; instead, for the 50 hosts this was set to 20. The values of the message retransmission and the routing table transmission intervals were set to 30 sec. The local utilities and the routing tables are updated every 30s. The routing table size was set to 8 and 20 for the 20 and 50 hosts scenarios respectively (i.e., it is equal to 40% of the number of the hosts and sufficient to store information about all the hosts of two initial communities). This limited size of the routing table is used to study the replacement mechanisms in the buffer and to reproduce possible limitations in terms of memory of small devices.

Protocol used for performance comparison: In order to evaluate the performance of CAR, the protocol is compared with Destination Sequenced Distance Vector (DSDV) routing protocol. The DSDV is a table driven routing protocol, where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network. The tables are exchanged between neighbors at regular intervals to keep up-to-date view of the network topology. It incorporates table updates with increasing sequence number tags to prevent loops, to counter the count to infinity problem and for faster convergence. This protocol suffers from excessive routing overhead that is proportional to the number of nodes in the network. Hence it is not scalable for ad hoc environment.

Evaluation metrics: The metrics used in this evaluation are defined as follows:

Packet delivery ratio: It is the number of data packets received by the destination nodes divided by the number of data packets transmitted by the source nodes.

Routing overhead: The routing overhead is defined as the total number of routing control packets normalized by the total number of received data packets.

Packet loss ratio: It is the number of data packets lost divided by number of lost packet and number of packets received successfully.

DISCUSSION

Comparison between DSDV and CAR protocol on packet delivery ratio is given in Fig. 3. When compared with DSDV protocol, CAR protocol produces constant delivery ratio during network partition. As number of nodes increases, packet delivery ratio decreases drastically in DSDV. In CAR protocol, packet delivery ratio remains constant and during network partitions, delivery ratio experience slight delay as given in Fig. 3. The delay is due to buffering at carrier node and carrier node movement from one cloud to another.

Comparison between CAR and DSDV protocol on Normalized Routing Load is given in Fig. 4. Number of routing packets drastically increases with increasing nodes. This is because as the number of nodes increases, more nodes will be flooding the network with RREQs and consequently more nodes will send RREPs as well. Also source node will have to generate more RREQs to find a fresh enough route to destination. DSDV protocol uses more control packets during network partition and routing overhead increases as the number of nodes increases. When compared with DSDV, CAR reduces routing overhead by minimizing control packets.

Comparison between CAR and DSDV protocol on Packet Loss is given in Fig. 5. In DSDV protocol, as the node speed increases, the position of a node will clearly change more rapidly.

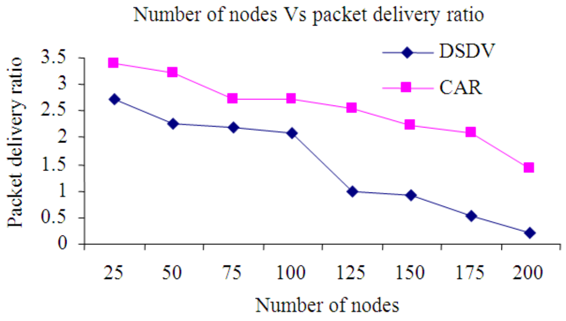

Fig. 3: Comparison between CAR and DSDV protocol on packet delivery ratio

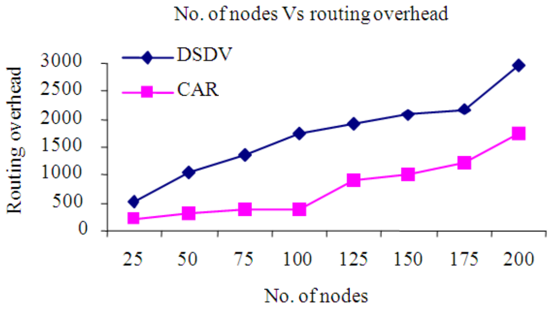

Fig. 4: Comparison between CAR and DSDV protocol on routing overhead

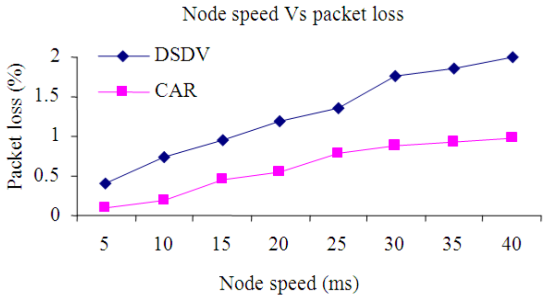

Fig. 5: Comparison between CAR and DSDV protocol on packet loss

The source node will still use the last route it has for a destination (if it didn't expire yet), but due to the fast mobility pattern, this route will frequently be invalid which causes the packet to be dropped. This will cause more and more packets to time out before reaching their destinations. This was also noticed during our simulation, as almost all of the packets were dropped because they exceeded their maximum TTL (Time to Live). When compared with DSDV protocol, Context Aware Adaptive Routing Protocol minimizes packet loss by buffering the undelivered packets in the Carrier Node.

CONCLUSION

We have presented the design, the evaluation and the implementation of a new context aware routing protocol for broadcasting in mobile ad hoc networks. Prediction techniques can be used to design store-and-forward mechanisms to deliver messages in intermittently connected mobile ad hoc networks, where a connected path between the source and all destination nodes may not exist. A generic framework is designed for the evaluation of multiple dimensions of the mobile context in order to select the best message carrier. Kalman filter based forecasting techniques can be applied effectively to support intelligent message forwarding.

The simulation experiments have shown that the proposed protocol is able to guarantee good performance with a limited overhead in terms of number of messages sent, in comparison to the other single copy and multiple-copy protocols taken into consideration. For future work, the protocol can be extended to work under error-prone conditions, i.e., location errors, imperfect time synchronization and message losses

REFERENCES

Bantz, D.F., C. Bisdikian, D. Challener, J.P. Karidis and S. Mastrianni *et al*., 2003. Autonomic personal computing. IBM Syst. J., 42: 1. DOI: 10.1147/sj.421.0165

Brockwell, P.J. and R.A. Davis, 2002. Introduction to Time Series and Forecasting. 2nd Edn., Springer, New York, ISBN: 0387953515, pp: 434.

Chatfield, C., 2004. The Analysis of Time Series: An Introduction. 6th Edn., Chapman and Hall CRC, Boca Raton, ISBN: 1584883170, pp: 333.

D'Souza, R.J. and J. Jose, 2010 Routing approaches in delay tolerant networks: A survey. Int. J. Comput. Appli., 1: 8-14. DOI: 10.5120/370-557

Durbin, J. and S.J. Koopman, 2001. Time Series Analysis by State Space Methods. 1st Edn., Oxford University Press, Oxford, ISBN: 0198523548, pp: 253.

Garbinato, B., A. Holzer and F. Vessaz, 2009. Context-aware broadcasting approaches in mobile ad hoc networks. Comput. Netw., 54: 1210-1228. DOI: 10.1016/j.comnet.2009.11.008

Jain, S., K. Fall and R. Patra, 2004. Routing in a delay tolerant network. Proceedings of the 2004 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 30-Sept. 03, ACM, Portland, OR, USA., pp: 145-158. DOI: 10.1145/1015467.1015484

Johnson, D.B. and D.A. Maltz, 1996. Dynamic source routing in ad hoc wireless network. Kluwer Academic Publishers.

Jones, E.P.C., L. Li, J.K. Schmidtke and P.A.S.W. Aug, 2007. Practical routing in delay tolerant networks. IEEE Trans. Mobile Comp., 6: 943-959. DOI: 10.1109/TMC.2007.1016

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng., 82: 35-45. DOI: 10.1109/ICASSP.1982.1171734

Ke, M. and Y. Nenghai, 2010. AEBP: A new adaptive efficient broadcast protocol for delay tolerant network. Proceedings of the IEEE International Conference on Multimedia Technology, Oct. 29-31, IEEE Xplore Press, Ningbo, pp: 1-4. DOI: 10.1109/ICMULT.2010.5629673

Keeney, R.L. and H. Raiffa, 1993. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. 1st Edn., Cambridge University Press, Cambridge, ISBN: 0521438837, pp: 569.

Li, Q. and D. Rus, 2000. Sending messages to mobile users in disconnected ad-hoc wireless networks. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Aug. 06-11, ACM, Boston, MA, USA., pp: 44-55, DOI: 10.1145/345910.345918

Lindgren, A., A. Doria and O. Schelen, 2003. Probabilistic routing in intermittently connected networks. Mobile Comput. Commun. Rev., 7: 19-20. DOI: 10.1145/961268.961272

Musolesi, M., S. Hailes and C. Mascolo, 2005. Adaptive routing for intermittently connected mobile ad hoc networks. Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Jun. 13-16, IEEE Xplore Press, pp: 183-189. DOI: 10.1109/WOWMOM.2005.17

Musolesi, M. and C. Mascolo, 2006. A Community based mobility model for ad hoc network research. Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality, May 26-26, ACM Press, Florence, Italy, pp: 31-38. DOI: 10.1145/1132983.1132990

Musolesi, M. and C. Mascolo, 2007. Designing mobility models based on social network theory. ACM SIGMOBILE Mobile Comput. Commun. Rev., 11: 59-70. DOI: 10.1145/1317425.1317433

Pasztor, B., M. Musolesi and C. Mascolo, 2007. Opportunistic mobile sensor data collection with SCAR. Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, Oct. 8-11, IEEE Xplore Press, Pisa, pp: 1-12. DOI: 10.1109/MOBHOC.2007.4428679

Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Proceedings of the Conference on Communications Architectures, Protocols and Applications, (SIGCOMM'94), ACM, New York, USA., pp: 234-244. DOI: 10.1145/190314.190336

Perkins, C.E., 2008. Ad Hoc Networking. 1st Edn., Addison-Wesley, Boston, ISBN: 0321579070, pp: 384.

Sarafijanovic-Djukic, N., M. Pidrkowski and M. Grossglauser, 2006. Island hopping: Efficient mobility-assisted forwarding in partitioned networks. Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, Sept. 28-28, IEEE Xplore Press, Reston, VA., pp: 226-235. DOI: 10.1109/SAHCN.2006.288427

Spyropoulos, T., K. Psounis and C.S. Raghavendra, 2005. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, Aug. 22-26, ACM Press, Philadelphia, PA, USA., pp: 252-259. DOI: 10.1145/1080139.1080143

Taha, H.A., 2006. Operations Research: An Introduction. 1st Edn., Prentice Hall, Upper Saddle River, N.J., ISBN: 0131889230, pp: 813.

Wu, J. and D.J. Watts, 2002. Small worlds: The dynamics of networks between order and randomness. ACM SIGMOD Record, 31: 74-75. DOI: 10.1145/637411.637426

Zhang, Z., 2006. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. IEEE Commun. Surveys Tutorials, 8: 24-37. DOI: 10.1109/COMST.2006.323440

Zhao, W., M. Ammar and E. Zegura, 2004. A message ferrying approach for data delivery in sparse mobile ad hoc networks. Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, May 24-26, ACM Press, Tokyo, Japan, pp: 187-198. DOI: 10.1145/989459.989483