CHAPTER 3

CONTEXT-AWARE ADAPTIVE ROUTING PROTOCOL

3.1 INTRODUCTION

Broadcasting is a common operation in the MANET for route establishment and sending control and emergency messages. A reliable broadcasting in the MANET requires the delivery of messages from different sources to all the nodes in the network within a 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 MANET is characterized by frequent network partitions and so, to achieve a reliable broadcasting is a challenging task.

A new Context-aware Adaptive Routing Protocol (CAR) is designed for broadcasting in MANET based on the idea of exploiting multiple nodes as carriers of messages among network partitions to achieve a guaranteed delivery. A host willing to broadcast the message to all the nodes in the network uses a Kalman filter prediction technique and multi-criteria utility theory to choose the best next hop or carrier for the message 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 (It is assumed that the 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 as other approaches do. For instance, Message Ferrying approach relies on the prior knowledge of the routes of the special hosts called as ferry carrying the information.

3.2 SYSTEM MODEL

The proposed protocol is based on the idea of exploiting multiple nodes as carriers of messages among network partitions to achieve a guaranteed delivery. The choice of the best carrier is made using Kalman filter based prediction techniques and multi-criteria utility theory. The protocol assumes that the only information a host knows its identity and its position related to its logical connectivity. In particular, it is assumed that a host is aware of its absolute geographical location and that of its neighbors with the help of GPS receiver. Another 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 act in a Byzantine manner (a node may perform arbitrary behavior to disrupt the system e.g., processing a request incorrectly or sending an incorrect or inconsistent response to a request).

3.2.1 Overview

The design goal of the proposed protocol is to support the communication in an intermittently connected MANET. The key problem solved by the protocol is the selection of the carrier nodes based on the application of the prediction techniques and multi criteria utility theory for the evaluation of different aspects of the system relevant for taking routing decisions. The key aspect of the protocol is the ability to deliver the messages synchronously i.e., without storing them in buffers of intermediate nodes when there are no network partitions between nodes and asynchronously i.e.,
by means of a store-and-forward mechanism when there are network partitions. The delivery process depends on whether all the nodes are present in the same connected region of the network (cloud) as the sender or not.

If all the nodes are currently in the same connected portion of the network, the message is broadcast with an underlying synchronous routing protocol to determine a forwarding path. If a message cannot be broadcast synchronously, the best carriers for a message are selected which are nodes with the highest delivery probabilities. The message is sent to the host with the highest delivery probability using the underlying synchronous protocol and the message is broadcast by the carrier nodes to those nodes that are in a separate cloud.

In order to understand the operation of the protocol, the following scenario may be considered in which two groups of nodes are connected as in Figure 3.1. Host H1 wishes to broadcast a message to all the hosts in the network. Destination Sequenced Distance Vector (DSDV) routing protocol is used to support the synchronous routing to all the connected hosts and is not possible for the hosts not connected.

![Figure 3.1 Two Connected Clouds with Associated Delivery Probabilities](image)
The delivery probabilities being determined for each connected host, the host with the best delivery probability can be used as a carrier, in this case Host H4 and consequently, the message is sent to H4 that stores it. After some time, H4 moves to the other cloud as in Figure 3.2. Since a connected path between H4 and second cloud exists, the message is delivered to all the hosts in the second cloud. Using DSDV routing protocol, the host H4 is able to broadcast the message shortly after joining the second cloud.

![Figure 3.2 Carrier Node Movements from One Cloud to Another](image)

The delivery probabilities are determined locally from the context information defined as the set of attributes that describe the aspects of the system to be used to drive the process of message delivery. An example of the context information is the change in the degree of connectivity, i.e., the number of connections and disconnections that a host experienced over the last T seconds. This parameter measures relative mobility of the node. Using a proactive routing protocol like DSDV, every host periodically sends the information related to the underlying synchronous routing (routing table having distances, next hop host identifier, and so on.), and a list containing its delivery probabilities for the other hosts.
When a host receives this information, it updates its routing table. Each host maintains the entries in the routing table such as destination, best host and delivery probability for asynchronous routing during the network partition. The routing table for Host H1 is shown in Table 3.1. When a host is selected as a carrier and receives the message, it applies store and forward mechanism and inserts the message into a buffer.

Table 3.1 Routing Table of Host H1

<table>
<thead>
<tr>
<th>Target HostId</th>
<th>Next HopId</th>
<th>dist</th>
<th>bestHop HostId</th>
<th>delivery Prob</th>
<th>Sequence Number</th>
<th>Install Time</th>
<th>Stable Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>H1</td>
<td>0</td>
<td>H1</td>
<td>0.2</td>
<td>406</td>
<td>001000</td>
<td>Ptr_H1</td>
</tr>
<tr>
<td>H2</td>
<td>H2</td>
<td>1</td>
<td>H2</td>
<td>0.3</td>
<td>128</td>
<td>001200</td>
<td>Ptr_H2</td>
</tr>
<tr>
<td>H3</td>
<td>H3</td>
<td>1</td>
<td>H3</td>
<td>0.5</td>
<td>550</td>
<td>001200</td>
<td>Ptr_H3</td>
</tr>
<tr>
<td>H4</td>
<td>H2</td>
<td>2</td>
<td>H3</td>
<td>0.5</td>
<td>588</td>
<td>001200</td>
<td>Ptr_H4</td>
</tr>
<tr>
<td>H5</td>
<td>H2</td>
<td>3</td>
<td>H4</td>
<td>0.9</td>
<td>312</td>
<td>001300</td>
<td>Ptr_H5</td>
</tr>
<tr>
<td>H6</td>
<td>H2</td>
<td>4</td>
<td>H4</td>
<td>0.9</td>
<td>076</td>
<td>001300</td>
<td>Ptr_H6</td>
</tr>
<tr>
<td>H7</td>
<td>H2</td>
<td>4</td>
<td>H4</td>
<td>0.9</td>
<td>710</td>
<td>001300</td>
<td>Ptr_H7</td>
</tr>
<tr>
<td>H8</td>
<td>H3</td>
<td>5</td>
<td>H4</td>
<td>0.9</td>
<td>050</td>
<td>001300</td>
<td>Ptr_H8</td>
</tr>
</tbody>
</table>

3.3 CONTEXT - AWARE ADAPTIVE ROUTING PROTOCOL

The routing protocol is optimized with the help of the predicted future values of the context attributes for making routing decisions to have a more accurate estimation of the context information, instead of using the available current context information as it is. For example, if the attribute future host collocation is considered, it can be observed that the host H₄ is
currently not collocated with a host $H_5$ as in Figure 3.1 and may not be considered for acting as a carrier node for host $H_5$ in case the context information is evaluated based only on the current situation. However, $H_4$ may have been collocated with $H_5$ for the past three hours and therefore its likelihood of being collocated again is high and should be considered accordingly to evaluate the context information.

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 the utilities for each attribute describing the context. Then the future values of these utilities are predicted and composed making use of the multi criteria decision theory to estimate an overall delivery probability. The calculated delivery probabilities are periodically sent to other hosts in the connected cloud as part of the updation of the routing information.

- Each host maintains a logical forwarding table of tuples describing the next logical hop and its associated delivery probability for all the known destinations.

- Each host uses the local prediction of the delivery probabilities between any two updation of routing information. The prediction process is used during the temporary disconnections to calculate an accurate delivery probability.

3.3.1 Evaluation of Context Information

Each host calculates its delivery probability locally by considering only the context attributes excluding the overall topology of the network. It is the key problem 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 for the context attribute and goal programming is one of them. With respect to a single attribute, the goal is to maximize its value 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. This technique is simple and easy to implement. In general, the decision problem involves multiple conflicting objectives, and for example, the attributes like the battery energy level and the change in the degree of connectivity may be considered. The hosts characterized by the highest mobility, in turn, happen to have limited residual battery energy and vice versa. Generally, maximization across all the parameters is not possible and so that of a single attribute against others is an achievable objective.

The context information related to a certain host can be defined with a set of attributes \((x_1, x_2, \ldots, x_n)\). Those attributes denoted with a capital letter (e.g., \(X_i\)) refer to the set of all possible values for the attribute, whereas those denoted with a lower case letter (e.g., \(x_i\)) refer to a particular value within this set. An example of a generic attributes \(X_i\) can be the mobility of the hosts or its battery level. For instance, the value \(x_1\) of the attribute battery level may be 0.99 (i.e., battery almost full).

If the set of attributes \(x_1, x_2, \ldots, x_n\) are mutually and preferentially independent characterized by the same degree of significance, the sum of the attributes is equivalent to the overall utility of the context attributes.

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

(3.1)

where, \(U_i\) is a utility function over \(x_i\). Now, the aim is to maximize each attribute to choose the hosts as best carriers for the message delivery. Weights method is applied to choose the best carriers for the message delivery.
The goal function used in the Weights method is defined as,

\[
\text{Maximize } \left\{ \sum_{i=1}^{n} w_i U_i(x_i) \right\}
\]

where, \( w_1, w_2, \ldots, w_n \) are significant weights reflecting the relative importance of each goal.

These results are used to compute the composition of the utilities, \( f(U_1, \ldots, U_n) \) for the CAR protocol based on the different context dimensions (given their mutual independence). The function \( f(U_1, \ldots, U_n) \) is evaluated using the values predicted for each host and the one having the maximum value is selected as the best host.

### 3.3.2 Attributes of the Utility Functions

The knowledge about the current values of the context attributes is useful to a limited extent and the values of the attributes likely to be assumed in the future are very important and predicted using Kalman filter based prediction techniques which do not require the storage of the entire past history of the system and are computationally lightweight making them suitable for a resource constrained mobile network. These predicted values of the attributes are used for the calculation of the utility of each host to select the best one as a carrier for message.

In the implementation of the protocol two attributes are considered; the change in the degree of connectivity and the future host collocation given in the underlying assumption of their mutual independence, as these two attributes are most relevant to the ad hoc scenario. However, the framework is general and open to the inclusion of any other context attribute. Other possible context attributes 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 in the degree of connectivity of a host ‘h’ is defined as the number of connections and disconnections that the host has experienced over the last period \([t-1, t]\) seconds normalized by considering the nodes that have been in reach in this period. This parameter measures the relative mobility and the probability that the host will meet different nodes in a given period of time makes the host as a best carrier. The change in degree of connectivity of a host ‘h’ is computed using the equation,

\[
U_{\text{cdc}_h}(t) = \frac{|n(t-T) \cup n(t)| - |n(t-T) \cap n(t)|}{|n(t-T) \cup n(t)|}
\]

where, \(n(t)\) is the neighbor set of host ‘h’ at time \(t\), \(n(t-T)\) is the number of hosts that disappeared from the host ‘h’ in the time interval \([t-T, t]\).

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 of change in the degree of connectivity means that h recently has changed a large number of its neighbors. The collocation of ‘h’ with a host ‘i’ is calculated using the following equation,

\[
U_{\text{col}_{hi}}(t) = \begin{cases} 
1 & \text{if the host h is colocated with host i} \\
0 & \text{otherwise}
\end{cases}
\]

A value of 1 means that host ‘h’ has been collocated with host ‘i’ at time \(t\). These two values are fed into Kalman filter predictors, yielding the predictions \(\hat{U}_{\text{cdc}_h}\) and \(\hat{U}_{\text{col}_{hi}}\) of these utilities at time \(t + T\). The results are then composed into a single utility value using the multi criteria decision theory.
Multiple criteria decision theory is a sub discipline of operations research that explicitly considers multiple criteria in decision making environments. The common approach to find the solution is to take weighted sum of all attributes.

\[ V(a) = w_1 g_1(a) + w_2 g_2(b) + \ldots \]  \hspace{1cm} (3.5)

where \( g \) is some goal function on the attribute \( a \) and \( w \) is the significant weight reflecting the relative importance of each goal.

The resultant Utility function obtained from multi criteria decision theory is given as follows,

\[ U_{h,i} = W_{cde_i} \hat{U}_{cde_i} + W_{col_{h,i}} \hat{U}_{cde_h} \]  \hspace{1cm} (3.6)

This equation represents the delivery probability of a host ‘h’ with respect to host ‘i’.

In case of collocation, the choice of using the predicted future values is more important than the current values of the attributes. For example, two hosts that have got disconnected for just ten seconds after being connected for a long period may be considered. If only the current status is considered, the value of the utility function related to the collocation will be 0. Instead, if the hosts have been collocated for a long time in the past, they are likely to be collocated again in the future.

The value 0 does not provide a correct measure of the probability of the future host collocation of the two hosts. Hence the output of the Kalman filter is close to 1. The weights \( w \) denote the relative importance of each attribute. Their value depending on the application scenario in which they are used, the values of these weights are calculated dynamically for every host.
3.3.3 Adaptive Weight Calculation

If the weights of the utility function are fixed in advance, it will reflect the relative importance of the different context attributes but it is too static, since it fails to consider the dynamic values of the attributes. For example, a small drop in battery voltage may be indicative of the imminent exhaustion of the battery and so, it would be useful to reduce the weight of this attribute non-linearly to reflect this exhaustion.

Generally, the weight of each parameter is adapted dynamically and dependent on the values of those parameters. The weights are self adapted automatically as a runtime function. To incorporate the automatic adaptation of the weights, Equation 3.2 is modified by including one more parameter $a_i$ as adaptive weights to modify the utility function according to the variation of the context.

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

(3.7)

where, $a_i(x_i)$ is a composite parameter having the following three parameters.

- Ranges of values of context information, $a_{\text{range}_i} \left( x_i \right)$
- Predictability of the context information, $a_{\text{predictability}_i} \left( x_i \right)$
- Availability of the context information, $a_{\text{availability}_i} \left( x_i \right)$

The adaptive weights $a_i$ is computed using the following equation.

$$a_i \left( x_i \right) = a_{\text{range}_i} \left( x_i \right) a_{\text{predictability}_i} \left( x_i \right) a_{\text{availability}_i} \left( x_i \right)$$

(3.8)
Ranges of Values of the Context Information

The attributes change in the degree of connectivity of a certain host and collocation of a host at time \( t \) can be described using values in a given range \([0...1]\). For example, collocation can be represented using 1 to indicate that a certain host is in the same transmission range, 0 otherwise. Hence, the adaptive weights \( a_{\text{range}_i}(x_i) \) are modeled as a function in the domain \([0, 1]\).

Predictability of the Context Information

The attribute change in degree of connectivity of a certain host may not be predicted at some time. E.g. host may be unavailable or not reachable at some time and its delivery probability cannot be calculated. Now, the predictability of the context information is set as 0. The predictability of the context information is computed using the following Equation 3.9. The approach is 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 based only 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 \( a_{\text{predictability}_i}(x_i) \).

\[
a_{\text{predictability}_i} = \begin{cases} 1 & \text{if the context information is currently predictable} \\ 0 & \text{if the context information is not currently predictable} \end{cases}
\] (3.9)

Availability of the Context Information

The degree of the availability of the context information is calculated similar to the predictability as a time varying set of available attributes whose values are found using the following equation. It is unreasonable to assume that all context attributes have the same degree of availability. Thus, the degree of availability can be viewed as a time-varying
set whose values are known or predictable. Attributes may drop out of this set if meaningful values can no longer be predicted for them, since the information on which the prediction would have been based is too old. i.e. the missing context information carries an adaptive weight $a_i$ equal to 0.

$$a_{\text{availability}_i} = \begin{cases} 1 & \text{if the context information is currently available} \\ 0 & \text{if the context information is not currently available} \end{cases}$$  \hspace{1cm} (3.10)

### 3.3.4 Routing Table

Each host calculates its delivery probability with the help of Kalman filter prediction technique and multi criteria utility theory and this information is circulated in the network using the routing table. The delivery probability information is piggybacked along with the synchronous routing table information. Each host maintains a routing and context information table used for asynchronous routing using CAR protocol and synchronous routing using DSDV protocol. Each entry of this table has eight fields as shown in Figure 3.3.

<table>
<thead>
<tr>
<th>Target HostId</th>
<th>Next HopId</th>
<th>dist</th>
<th>bestHop HostId</th>
<th>delivery Prob</th>
<th>Sequence Number</th>
<th>Install Time</th>
<th>Stable Data</th>
</tr>
</thead>
</table>

**Figure 3.3 Different Fields in the Routing Table**

The first three fields and the last three fields are mandatory in any routing protocols, the first field being the recipient of the message, the second, the identifier of neighboring host, the third, the distance between the two hosts, the sixth, the sequence number, the seventh, the installation time, and the eighth, the stable data. The fourth field is the identifier of the best host (carrier node) and its delivery probability is stored in the last field. The routing table information is used for both synchronous and asynchronous
delivery. The field’s nextHopId and distance are used during synchronous message transmission and the field’s bestHopHostId and deliveryProb are for the selection of the best carrier during the network partition. A distance equal to 16 in the third field considered to be infinite is chosen as this is the infinity value of the classical Routing Information Protocol (RIP) and the host is treated as unreachable. The protocol structure of RIP is given in Appendix 2. However, this parameter can be tuned according to the user requirements. In a scenario characterized by a high average host speed, a lower value may be used since the probability that the route may be broken or stale is potentially high.

The value of the field delivery Prob is updated using the last received value. The values received by the neighbors are used to update the 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 because of a temporary disconnection or transmission errors due to interference or the movement of the host. 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 an input to the filter.

### 3.3.5 Routing Table Updation Interval

The routing tables being exchanged periodically with a given transmission interval, each node keeps the local utilities related to the collocation with other nodes. When a host receives a routing table, it checks its entries against the ones stored in its routing table. The updation 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.
The entry is replaced even if the distance is the same to have a fresh information about the route. As far as the asynchronous routing 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 avoiding 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 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 there is an associated Kalman filter based predictor that has to be updated periodically in every entry.

### 3.3.6 Routing Table Transmission Interval

The routing table transmission interval is another fundamental parameter of the protocol. Routing tables are used not only to exchange the routing information but also for the discovery of new routes. They are employed as a sort of beaconing mechanism at the application level to keep the information about the presence of the neighbors.

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. The frequency of the transmission of the routing tables is relatively high to provide the correct information to the collocation predictor. The update interval of the Kalman filter is another fundamental parameter of the protocol that is used to detect the changes in the observed context attribute. For example, in case of the 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/s, the transmission range is 200 m/s, and an update interval greater than 20s, some of the hosts may not be discovered. Hence, the update interval of the Kalman filter is set as the value equal to the routing table transmission interval.

3.3.7 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, (i.e. an entry with the field Target HostId exists in the routing table and the associated distance is less than 16), it is forwarded to the next hop indicated by nextHopId if the recipient is reachable synchronously. This forwarding mechanism is typical one of any distance vector routing protocols.

Sometimes the path to a certain host may be broken but, at the same time, the routing table may not have been updated with the information related to this change. In such cases, the message is forwarded until it reaches the host that has already been notified about the disconnections. This host then checks if the message can be sent using the asynchronous delivery mechanism (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 equal to or greater than 16), the message is forwarded to the host with the highest value of the delivery probability (expressed by deliveryProb).
DSDV routing protocol is used to reach the carrier node. 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, the carrier may happen to be unreachable leaving the connected cloud. In such cases, the entry related to the best carrier is removed (set to an invalid state designated by 0) if the information about the disconnection has reached the sender. In order to avoid the propagation of stale routes, the sequence numbers are used for the routing tables as in DSDV protocol. If this information has not been propagated yet to the sender, the intermediate host aware of the topological change tries to resend the message.

### 3.3.8 Retransmissions

Periodically, a host checks its routing table for each message in its buffer. The message is then forwarded synchronously to the recipient or to a carrier if a corresponding entry is present in the routing table and in the event of no entry being present, the message stays in the buffer.

Each node maintains a list of its utilities for a certain set of hosts particularly that of the local utilities related to the collocation with other hosts and the one related to its change in the 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 than the one currently maintained in the table, the latter is replaced. The value of the utilities is updated before being compared with the entries of the local routing table.

### 3.3.9 Kalman Filter Prediction Techniques

Kalman filter prediction techniques are used for the prediction of the future values of the context attributes and the delivery probabilities in the
local routing tables, if updates are not received. The Kalman filter, also known as Linear Quadratic Estimation (LQE) is an algorithm that uses a series of measurements observed (past values) over time and produces estimates of unknown variables (future values) that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The Kalman filter has numerous applications in technology. A common application is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics.

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current variables. Once the outcome of the future measurement is observed, these estimates are updated using a weighted average. Because of the recursive nature of the algorithm, it can run in real time using only the present input measurements and the previously calculated state. No additional past information is required to predict the future values.

Overview

Kalman filter prediction techniques are developed in an automatic control systems theory that is the discrete signal processing method providing the optimal estimates of the current state of a dynamic system described by a state vector. The state is updated using the periodic observation of the system with a set of prediction recursive equations. Kalman filter theory is useful in the routing protocol to achieve a more realistic prediction of the evolution of the context of a host and optimize the bandwidth usage. As discussed above, the exchange of the context information that allows the calculation of the
delivery probabilities is potentially an expensive process and unnecessary as such information is relatively predictable.

If it is possible to predict the future values of the attributes describing the context so that the delivery probabilities stored in the routing tables can be updated, even if fresh information is unavailable. This prediction problem can be expressed in the form of a state space model. Starting from the time series of the observed values that represent the context information, a prediction model is derived based on an inner state represented by a set of vectors. 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 network in which memory resources may potentially be very limited.

**Context Predictability**

It is one of the major issues to deal with the variability and uncertainty in many networking systems like the mobile ad hoc and delay tolerant network. The decentralization of the control and dynamic movement of the hosts have a great impact on the topology of the system. The protocol relies on the accuracy of the prediction model to a greater extent but there are situations where the context cannot be predicted and using any prediction based techniques to improve the performance of the system is completely ineffective. So, a predictability component is designed to measure the accuracy of the prediction of the context information. The technique adopted is predicted by the analysis of the time series representing the context information. Given a certain number of measurements of the predictability of the time series, the 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 has given a predefined acceptable error. The algorithm used by the
prediction component to calculate the predictability of the time series is given below. In the experiments, the predictability of the time series of the collocation between a pair of hosts is considered. Every sample of the time series is evaluated for the calculation of the predictability level and if it is below the threshold value, alternative protocols such as epidemic approach can be used.

**Algorithm 3.1 Verifying the predictability of the time series**

Variables

- $Z(t)$ -> time series of residuals (Prediction errors)
- $R_k$ -> autocorrelation coefficients
- $K_{max}$ -> max log
- $N$ -> number of samples
- $\text{maxError}$ -> max acceptable prediction error

Boolean $\text{isPredictable} (U(t-NT)\ldots U(T), \bar{U}(t-NT)\ldots \bar{U}(t))$

1: for all $\tau \in [t-nT,t]$ do

2: calculate $z(T) = \bar{U} (\tau) - U (\tau)$

3: for all $k \in [1\ldots K_{max}]$ do

4: calculate $R_k$ of $z(t)$

5: if $(90\% \ of \ { R_k } \in [-2/\sqrt{N}, 2/\sqrt{N}]) \ and \ (0.1z(t)_{\leq \text{maxError}})$ then

6: return true;

7: else

8: return false;
3.4 SIMULATION

The simulation of the CAR routing protocol has been performed using Network Simulator - 2 and the performance of DSDV and CAR routing protocols are analyzed. The channel capacity of the mobile hosts is set as a constant value of 2 MBPS. The DCF of IEEE 802.11 is used for the wireless LANs as the MAC layer protocol which notifies the network layer about link breakage.

3.4.1 Choice of the Parameters

The protocol is simulated using the utility function based on the evaluation of two attributes:

(i) The change in the degree of connectivity.

(ii) The probability of being connected with the other cloud in case of partition i.e, future host collocation.

All the possible values assumed in the range have equal importance (i.e., $a_{\text{range}} = 1$) and the values of attributes are always available during the simulation (i.e., $a_{\text{availability}} = 1$). The values of $w_{\text{cdc}}$ and $w_{\text{col}}$ for all 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. The various Simulation Parameters are depicted in Table 3.2.
Table 3.2 Simulation Parameters for CAR and DSDV Routing Protocols

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Size</td>
<td>1000 x1000 Meters</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>25, 50, 75, 100, 125, 150, 175, 200</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>2 MBPS</td>
</tr>
<tr>
<td>Synchronous Routing Protocol</td>
<td>DSDV</td>
</tr>
<tr>
<td>MESSAGE_PORT</td>
<td>42</td>
</tr>
<tr>
<td>BROADCAST_ADDR</td>
<td>-1</td>
</tr>
<tr>
<td>NAM Animation Speed</td>
<td>250 µs</td>
</tr>
<tr>
<td>Node Speed</td>
<td>5, 10, 15, 20, 25, 30, 35, 40 Micro Seconds</td>
</tr>
<tr>
<td>MAC</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Radio Range</td>
<td>250 Meters</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>100 s</td>
</tr>
<tr>
<td>Traffic Source</td>
<td>CBR (Constant Bit Rate)</td>
</tr>
<tr>
<td>Broadcast Delay</td>
<td>0.01 µs</td>
</tr>
<tr>
<td>Hello Reply Delay</td>
<td>0.01 µs</td>
</tr>
<tr>
<td>Packet Size</td>
<td>512 Bytes (Max 1500 Bytes)</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>Antenna/OmniAntenna</td>
</tr>
<tr>
<td>Interface queue type</td>
<td>Queue/DropTail/PriQueue</td>
</tr>
<tr>
<td>Max packet in Interface Queue</td>
<td>50</td>
</tr>
<tr>
<td>Mobility Model</td>
<td>Random Way Point</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>0.360 Watt</td>
</tr>
<tr>
<td>Receiving Power</td>
<td>0.395 Watt</td>
</tr>
<tr>
<td>Idle Power</td>
<td>0.335 Watt</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>5.1 J</td>
</tr>
</tbody>
</table>

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). A split horizon mechanism is used to prevent the messages from being retransmitted.
unnecessarily. The split horizon rule can be explained with an example. Consider a network scenario given in Figure 3.4 having three nodes, node A routes packets to node B in order to reach node C. The links between the nodes are distinct point-to-point links.

![Figure 3.4 Split Horizon Rule](image)

Node A routes packets to node B in order to reach node C. The links between the nodes are distinct point-to-point links. According to the split-horizon rule, node A does not advertise its route for C (namely A to B to C) back to B. On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C. However, if the link between B and C goes down, and B had received a route from A, B could end up using that route via A. A would send the packet right back to B, creating a loop. When the split horizon rule is used, this particular loop scenario cannot happen, improving convergence time in complex, highly-redundant environments.

The buffer for each node is set to 50. The number of retransmissions for the 25 hosts scenario is set to 10 and for the 50 hosts scenario is set to 20 and so on. The values of the message retransmission and the routing table transmission intervals are set to 30 micro seconds. The local utilities and the routing tables are updated every 30 micro seconds. The routing table size is set to 40% of the number of the hosts and is sufficient to store the information about all the hosts. For example, the size is set to 10 for the 25 hosts and 20 for 50 hosts scenarios and so on. This limited size of the routing table is used to study the replacement mechanisms in the buffer and suitable for small devices having limited memory.
3.4.2  Protocol Used for Performance Comparison

In order to evaluate the performance of CAR, the protocol is compared with 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 that incorporates the table updates with increasing sequence number tags to prevent loops and, solve the count to infinity problem. This protocol suffers the excessive routing overhead proportional to the number of nodes in the network and therefore, it is not suitable during the network partition.

3.4.3  Experimental Setup and Results

The node creation and deployment is shown in Figure 3.5. There are 50 nodes in the simulation. The source node 1 is marked in red and the carrier node is identified as 3 and is marked in merun. The source node starts broadcasting the message to all its neighboring nodes including the carrier node.
Figure 3.5  Node Creation and Deployment

Figure 3.6 shows the initial phase of broadcasting using DSDV routing protocol. The neighboring nodes start broadcasting the message using DSDV routing protocol and the nodes receiving the broadcast message are marked in green.
Figure 3.6 Initial Phase of Broadcasting using DSDV Routing Protocol

Figure 3.7 shows the network partition resulting in two connected clouds and the carrier node movement from one cloud to another. The figure depicts the condition in which the network is broken into two connected clouds because of the dynamic nature of the ad hoc nodes. The carrier node is highly mobile and starts moving from one cloud to the other, i.e. it detaches itself from one cloud and attaches to the other proving that the carrier node has higher mobility and delivery probability compared with the other nodes in the network.
Figure 3.8 shows the final phase of the broadcasting using CAR protocol. The carrier node attached to the other cloud initially broadcasts using asynchronous CAR protocol since it buffers the undelivered packets. This is accomplished by store and forward mechanism and thereby it ensures asynchronous message transmission in the intermittently connected network. The message gets broadcast to the rest of the nodes in the second cloud using synchronous DSDV routing protocol.
Figure 3.8 Carrier Node Movements from One Cloud to Another

Figure 3.9 shows the final phase of the broadcasting in CAR protocol. The message gets broadcast to the second cloud using DSDV synchronous routing protocol.
3.5 PERFORMANCE METRICS AND ANALYSIS

The following metrics are considered to evaluate the performance of Context-aware Adaptive Routing Protocol and Destination Sequenced Distance Vector (DSDV) Routing Protocol.

- **Packet Delivery Ratio** - It is the ratio of the number of the data packets received by the destination nodes to the number of the 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 ratio of the number of data packets lost to the total number of the lost packets and the number of packets received successfully.

X-graphs are generated to compare the performance of CAR and DSDV routing protocols based on the above three metrics. Each point in the plot is an average of over a hundred simulation runs. The simulation results based on different threshold values are presented to verify and compare the effectiveness of these algorithms.

The comparison between DSDV and CAR protocols on Packet Delivery Ratio is given in Figure 3.10. Compared with DSDV routing protocol, CAR protocol produces a finer packet delivery ratio during the network partition. As the number of nodes increases, the packet delivery ratio decreases drastically in DSDV.

In CAR protocol, the packet delivery ratio remains good. In general, when the number of nodes increases, the packet delivery ratio decreases because more number of the control packets are transmitted by the intermediate nodes for the route establishment and maintenance. CAR protocol produces a good packet delivery ratio as it buffers the data packets at the carrier node.
The comparison between CAR and DSDV routing protocols based on Routing Overhead is given in Figure 3.11. Generally, the number of routing packets being transmitted increases when the number of nodes increases. As the number of nodes increases, more nodes are flooding in the network with Route Requests (RREQ) and consequently more nodes are sending Route Replies (RREP) as well. In addition to this, the source node has to generate more RREQs to find a fresh route to the destination. DSDV protocol uses more control packets during the network partition and the routing overhead increases as the number of nodes increases. Compared with
DSDV, CAR reduces routing overhead by minimizing the number of control packets transmitted during the network partitions.

Figure 3.11 Comparison between CAR and DSDV Protocols on Routing Overhead

Comparison between CAR and DSV routing protocols on Packet Loss Ratio is given in Figure 3.12. As the speed of the node increases, the position of a node clearly changes more rapidly. The source node still uses the last route it has for a destination (if it didn’t get expired), but owing to the fast mobility pattern, this route is frequently invalid causing the packet to be dropped. This causes more and more packets to time out before reaching their destinations that was also noticed during the simulation, as some of the
packets were dropped since they exceeded their maximum Time to Live (TTL). Compared with DSDV protocol, CAR Protocol minimizes packet loss by buffering the undelivered packets in the carrier node.

![Graph comparing CAR and DSDV protocols on packet loss ratio](image)

**Figure 3.12** Comparison between CAR and DSDV Protocols on Packet Loss Ratio

### 3.6 Conclusion

The design, evaluation and implementation of a new Context-aware Adaptive Routing Protocol for broadcasting in the MANET are presented. The prediction techniques are used to design store-and-forward mechanisms to deliver messages in an intermittently connected MANET in which a
connected path between the source and all nodes may not exist. A generic framework is designed for the evaluation of the multiple dimensions of the mobile context to select the best carriers for message transmission. Kalman filter based prediction techniques are applied effectively to support the intelligent message forwarding.

The simulation experiments have shown that the new protocol is able to provide the guaranteed better results with a limited overhead in terms of the number of messages sent compared with other routing protocol like DSDV. The protocol maximizes the packet delivery ratio during the network partitions and minimizes the routing overhead. Compared with DSDV protocol, CAR Protocol is more efficient in minimizing packet loss by buffering the undelivered packets in the carrier nodes.