Chapter 6

New Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad-hoc Networks

In this chapter, the researcher proposed a new broadcast protocol which is suitable for a wide range of vehicular scenarios used for real-time Mumbai-Pune Express Highway India, which not only employs local or status information but also employs warning or emergency information acquired via periodic beacon messages, containing Acknowledgment Messages of the circulated broadcast messages. Each vehicle decides whether it belongs to a connected dominating set (CDS), than it calculates multi-hop CDS which will reduce redundancy of transmitted messages. Vehicles in the CDS use a short waiting period/time before it is possible to retransmit. At timeout expiration, a vehicle retransmits, if it is aware of at least one or two neighbor in need of the message. To address intermittent connectivity and appearance of new neighbors, the evaluation timer, can be restarted. Once the CDS is calculated then cluster is formed for Vehicle nodes in communication range on Mumbai-Pune Express Highway. Cluster Head (CH) is elected which will try to broadcast messages to Cluster Member (CM) in the Cluster of vehicle nodes. The protocol resolves propagation at road intersections without any need to even recognize intersections. It is inherently adaptable to different mobility regimes, without the need to classify
6.1. INTRODUCTION

Vehicular Ad-hoc Networks consist of collections of vehicles on Mumbai-Pune Express Highway India. Mumbai-Pune Express Highway is 3-Lane road on each direction equipped with wireless communication capabilities. Vehicles cooperate to deliver different types of messages such as local or status messages and warning or emergency messages through multi-hop CDS connectivity and cluster formation of vehicle on Mumbai-Pune Express Road. To achieve this, V2V communication protocol must cope with the mobility of vehicles on Mumbai-Pune Express Highway Road and the dynamics of wireless signals without Road side Infrastructure. Vehicle movements are restricted by the Mumbai-Pune Express Highway Road layout in bi-directional manner with multi-lane on each side of road. This leads to highly partitioned networks with non uniform distribution of nodes. Furthermore, Mumbai-Pune Express Highway scenario need to be addressed, when studying vehicular ad-hoc networks. Technical challenges in this environment are discussed in Hartenstein and Laberteaux (Munoz) with these other technical challenges are A central challenge of VANET’s is that no central coordination or handshaking protocol can be assumed, and given that many applications will be broadcasting information of interest to many surrounding vehicle on Mumbai-Pune Express Highway Road, the necessity of a single, shared control channel can be derived. The bandwidth of the frequency channels currently assigned or foreseen for VANET applications ranges from 10 to 20 MHz.

With a high vehicular traffic density, those channels easily could suffer from channel congestion. Making use of more than one channel leads to multichannel synchronization problems. Other Challenges are the dynamic network topology based on the mobility of the vehicles and the environmental impact on the radio propagation. The low antenna heights and the attenuation or reflection of all the moving metal vehicle bodies provides for adverse radio channel conditions. All together, VANET’s must work properly in a wide range of conditions, including sparse and dense vehicular traffic. There is a strong need for adaptive transmit power and rate control to
achieve a reasonable degree of reliable and low latency communication. In addition, there is a challenge in balancing security and privacy needs. On the one hand, the receivers want to make sure that they can trust the source of information. On the other hand, the availability of such trust might contradict the privacy requirements of a sender. Socio-Economic Challenges are as follows: Market introduction of direct communication between vehicles is suffering from the network effect: the added value for one customer depends on the number of customers in total who have equipped their vehicle with VANET technology. A key question, therefore, is how to convince early-adopters to buy VANET equipment for their vehicles.

Broadcasting is the task of sending a message from a source node, to all other nodes in the vehicular ad-hoc network on Mumbai-Pune Express Highway Road. It is frequently referred to as data dissemination with a communication range of 0.25 km. The design of reliable and efficient broadcast protocol is a key for the successful deployment of vehicle-to-vehicle communication services. Most of the envisioned services rely on the delivery of broadcast messages to the vehicles inside a certain area of interest [Khabazian and Ali]. This operation is therefore also known as Geocasting. Vehicle-to-vehicle (V2V) communication system can also be used as a distributed platform for 'opportunistic cooperation' among people with shared interests or goals [Lee et al.].

6.2 Related Work

One of the challenges for VANET’s is the dynamic and dense network topology on Highway Road, resulting from the high mobility and high node-density of vehicles. This dynamic topology causes routing difficulties as well as congestion from flooding, and the dense network leads to the hidden terminal problem. A clustered structure can make the network appear smaller and more stable in the view of each node. By clustering the vehicles into groups of similar mobility, the relative mobility between communicating neighbor nodes will be reduced, leading to intra-cluster stability. In addition, the hidden terminal problem can be diminished by clustering.

Another issue generated by the dynamic and dense network, is the Broadcast Storm Problem. The broadcast storm problem describes the congestion resulting from re-broadcasts and flooding in a VANET. The dynamic topology of VANET’s demand
6.2. RELATED WORK

a high frequency of broadcast messages to keep the surrounding vehicles updated on position and safety information or messages. In addition, many routing algorithms necessitate flooding the network to find routes, which in a dynamic network needs to be done frequently to keep routes updated. All of this flooding leads to severe congestion, which can be alleviated by a clustered topology [Hsiao-Kuang et al.]. When the network is clustered, only the cluster-head participates in finding routes, which greatly reduces the number of necessary broadcasts.

An additional challenge for VANET’s is Quality-of-Service (QoS) provisioning. In VANET’s, many different types of data will need to be transmitted, and messages will be both delay-intolerant and delay-tolerant. For example, safety messages will demand high reliability and low delay, whereas non-vital road and weather information will be tolerant to longer delays. These different data types necessitate QoS provisioning, which can be achieved by a clustered network [Hsiao-Kuang et al.].

Clustering is the process of separating the nodes of a network into organized partitions called clusters. The clusters form sub-networks in the overall network, thus forming then hierarchical topology. Nodes in a cluster must be one of the following types Figure 6.1:

- Cluster head (CH) – An elected node that acts as the local controller for the cluster. The cluster-head’s responsibilities may include: routing, relaying of inter-cluster traffic from cluster members, scheduling of intra-cluster traffic, and channel assignment for cluster members.

- Cluster Member (CM) – A normal node belonging to a cluster. Cluster members usually do not participate in routing, and they are not involved in inter-cluster communication.

- Cluster Gateway Node (CG) – This is an optional node, which is used in some clustering schemes. The gateway node belongs to more than one cluster, acting as the bridge between cluster-heads. When present, the gateway nodes participate in both forwarding of inter-cluster traffic and the routing process. The cluster-heads and gate-way nodes form the backbone network.
6.2. RELATED WORK

The main aim of clustering algorithm is to minimize cluster reconfiguration and cluster-head changes, which are unavoidable due to the dynamic nature of the network. Having a good clustering algorithm requires selecting the cluster-head that will serve most of the vehicles communication on road for the longest possible time. Knowing the traffic flow and the general information of a vehicle, such as speed, direction, location and lane, should lead to better cluster-head selection.

The various cluster-head selection algorithms are as follows:

- Lowest-id clustering algorithm: It has the lowest overhead. In Lowest-ID, each node is assigned a unique ID, and the node with the lowest-ID in its two-hop neighborhood is elected to be the cluster-head \( \text{(Gerla, M., and Tsai)} \). The algorithm works as follows:
  
  - Each node periodically broadcasts its unique-ID, along with the ID of its neighbors.
  - If a node has the lowest-ID of all ID’s it hears, it becomes a cluster-head.
  - The lowest-ID a node hears its cluster-head, unless that node gives up cluster-head status to another lower ID node. In this case, the node will reevaluate lowest-ID status amongst undetermined nodes.
  - A node that hears from more than one cluster-head is a Cluster Gateway node.
In an effort to reduce the frequent re-clustering involved in maintaining the lowest-ID status of all cluster-heads, the Least Cluster Change (LCC) algorithm was suggested (Hsiao-Kuang et al.). In LCC, re-clustering is only performed when two cluster-heads come within range of one another. At this point, the cluster-head with the lower ID remains the cluster-head.

In highest degree based clustering algorithm each node in the network is assigned a degree based on the number of neighbors in the defined range. The node with the highest degree is selected as the cluster head (Gerla, M., and Tsai).

These algorithms do not exhibit cluster stability because they make no attempt to select a stable cluster-head during initial cluster-head election. For highly-mobile networks, mobility must be considered during the clustering process in order to ensure cluster stability.

In this chapter, researcher focus on the problem of broadcasting protocol in Vehicular Ad-hoc Networks without infrastructure support i.e. V2V Communication. Primary goal is to achieve high reliability, while minimizing the total number of re-transmissions using CDS and clustering algorithm with Acknowledgment message. In some safety applications, the delivery latency is critical. However, considering all these goals, appears to be a very challenging task on real time scenario of Mumbai-Pune Express Highway India, and concentrate here on non safety applications only. At the same time, vehicle still may not delay retransmission for too long as the reliability would otherwise suffer.

Topology changes due to mobility, cause frequent and temporary disconnections. Message might require to, be buffered and carried by a given vehicle until a new forwarding opportunity emerges. Several broadcasting protocols have been previously proposed. However, they are designed for either rectilinear highways/roads (Sun et al.), (Biswa, Tatchikou, and Dion), (Tonguz et al.). More surprisingly, only one of them (Tonguz et al.) addresses the issue of temporary disconnections in VANET, which is one of its most salient properties.

Researcher have developed the Broadcast Protocol which is fully distributed adaptive protocol suitable for Vehicular Ad-hoc Networks with all mobility scenarios. This protocol automatically adjusts its behavior without keeping track of the degree of mo-
bility sensed by the vehicle on road. Each node independently decides whether or not to forward a received broadcast message. Such decision is solely based on the local or status information, that vehicles acquire from their neighborhood by means of periodic beacon messages. This guarantees ultimate scalability regardless of the size of the VANET. The set of parameters in Broadcast Protocol is minimal and consists only of few natural choices.

6.3 Proposed Protocol

In Broadcast Protocol, a vehicle on Mumbai-Pune Express Highway receives a broadcast message which does not retransmit it immediately. Instead, the vehicle waits and checks if any retransmissions from other neighbors already available which will cover its whole neighborhood, making its transmission then redundant. To acquire multi-hop neighborhood position information, periodic beacons contain the position of the sender. Such information suffices to compute a connected dominating set (CDS). Nodes in the CDS select a shorter waiting time-out than regular nodes. This allows them to retransmit first if their neighborhood has not been covered already. That is, we combine two different techniques, CDS and Cluster algorithm (Stojmenovic, Seddigh, and Zunic), (Stojmenovic). Once CDS is calculated then cluster is formed for Vehicle nodes in communication range on Mumbai-Pune Express Highway. Cluster Head (CH) is elected which will try to broadcast messages to Cluster Member (CM) in the Cluster of vehicle nodes. The protocol resolves propagation at road intersections without any need to even recognize intersections. It is inherently adaptable to different mobility regimes, without the need to classify network or vehicle speeds. Beacons also include, identifiers of the recently received broadcast messages, which serve as acknowledgments of reception. This way, nodes can check whether all their neighbors successfully received a message from Cluster Head to Cluster Member within Cluster and also check successfully messages received from one Cluster Gateway to other Cluster Gateway which will have inter cluster communication. If this is not the case, a retransmission is scheduled. Otherwise, retransmission would be redundant. In both cases, when a new neighbor emerges, nodes restart their evaluation time-out, if the message being disseminated but not acknowledged. If the message identifier is actually included within the beacon, the
neighbor already has got the message and no retransmission is scheduled. Hence, the use of acknowledgments using RTS/CTS concept, makes the protocol more robust to transmission failures while, at the same time, saves redundant retransmissions.

Temporary disconnection incurs, delivery delay to any protocol. Although the described protocol inherently uses the store-carry-forward paradigm, Broadcast Protocol does not incur large delivery latencies. Vehicles connected to the Cluster Head will receive the message with small delay, due to propagation via CDS.

In a simulation-based study, we analyze the performance of Broadcast Protocol on Mumbai-Pune Express Highway, India scenario. Vehicles movements are generated with a microscopic road traffic simulation package i.e. eWorld and SUMO 12.0 from Google Maps, in order to mimic, common scenario of real vehicular networks on Mumbai-Pune Express Highway, India is considered. Different mobility conditions are simulated between different intersection points on Mumbai-Pune Express Highway like Kharghar, Panvel, Lonvala etc. Under realistic IEEE 802.11p models and AODV and DSR routing protocols.

6.4 The Broadcast Protocol

6.4.1 Overview

Propose of Adaptive Broadcast Protocol, which is suitable for a wide range of mobility conditions. The main problem that a broadcast protocol faces is its adaptability to the very different vehicular arrangements in real highway road scenarios. It should achieve high coverage of the network, at the expense of, as few transmissions as possible, regardless of whether the network is extremely dense or highly disconnected.

The Broadcast Protocol is localized, and based on applying the CDS and Cluster algorithm concepts on the currently available neighborhood information. In addition, Protocol assumes ideal communication radios to estimate the network connectivity and therefore apply the CDS and cluster algorithm techniques. Since real communication links are far from ideal, the protocol makes use of broadcast acknowledgments to insure the reception of the message or retransmit it. A message is acknowledged during its whole lifetime. At expiration, it is removed from the vehicle’s buffer and no more acknowledgments are issued. Given that broadcast messages are acknowledged,
it is assumed that they can be uniquely identified.

Vehicles are assumed to be equipped with Global Positioning System receivers. Periodic beacon messages are exchanged to update the vehicle’s local topology knowledge. The position of the sender is included within the beacons, which suffices to calculate a CDS backbone after each beacon message round. The source node transmits the message. Upon receiving the message for the first time, each vehicle initializes two lists: list $R$ containing all nodes believed to have received the message, and list $N$ containing those neighbors in need of the message. Then, each receiving node sets a time-out waiting period. If a node is not in the CDS, then it selects longer time-out than the nodes from the CDS, so that the latter reacts first. For each further message copy received, and its own message sent, every node updates $R$, $N$, and the time-out. At the end of the time-out period, it transmits, if $N$ is nonempty. Both ways, the message is buffered until it expires. For each beacon message received, $N$ and $R$ are updated according to the presence or absence of acknowledgment. Nodes that are no longer one-hop neighbors, are eliminated from these lists. Regardless of previous decisions, all nodes that so far, received the broadcast message check whether $N$ becomes nonempty. If so, they start a fresh time-out. In addition, acknowledgments of received broadcast messages are piggy backed to periodic beacons. Nodes that was included in $R$ because they were believed to have received the message, but did not actually get it, are later removed from R and inserted into $N$. This algorithm is executed for each different message. Therefore, the beacon size increases linearly, with the number of simultaneous broadcasting tasks.

Illustrates the protocol behavior on one example. Given the scenario depicted in Figure 6.2, vehicle $a$ generates a broadcast message which is first buffered by $a$, and then received by $b$, $c$, $d$. Receivers set up a waiting time-out which is shorter, if the vehicle belongs to the computed CDS. Let $d$ be in the CDS, thus it retransmits
first. Vehicles $b$ and $c$ cancel their retransmission because all their neighbors have been covered by $d'$s forwarding. Vehicles $e$ and $f$ receive the message. However, none of them have uncovered neighbors, so the retransmission does not take place. The Broadcast Protocol saves these redundant transmissions because the beacons contain the acknowledgment of the message, and therefore the newly discovered neighbors are not covered again.

Vehicle $a$ speeds up and overtakes vehicles $b - f$. In the case of PBSM, new transmissions would occur because new neighbors $e$ and $f$ must be covered by $a$ (and vice versa). However, they are redundant because all the vehicles have already received the message. The Broadcast Protocol saves these redundant transmissions because the beacons contain the acknowledgment of the message, therefore, the newly discovered neighbors are not covered again.

6.4.2 CDS Broadcast Protocol Details

Pseudo-code of The Broadcast Protocol is given in Algorithm 1. Upon receiving the broadcast message, vehicle $x$ includes in $R$ the sender and all its known neighbors (and starts $to$-$ack$ timers), because it is likely they have also received the message (lines 5-14). Accordingly, those vehicles are removed from $N$ (7,12). The remaining neighbors of $x$ which are not connected to the sender (their distance is greater than transmission radius $r$) are inserted into $N$ (15-16). There exists a time-out function $to$-$ev$ which assigns a waiting time to each vehicle before its possible retransmission. $to$-$ev$ is proportional to $1/|N|$, where $|N|$ is the number of elements in $N$, and depends on whether or not the node is currently in the CDS (shorter waiting time if in the CDS). The rationale is to provide vehicles that have more neighbors in need of the message, priority to retransmit first. If several neighbors have the same status and number of neighbors in need of the message, they will obtain the same $to$-$ev$ value. However, this does not mean an increased number of collisions, since The Broadcast Protocol runs at the network layer and these messages still have to contend to access the medium at the link layer (IEEE 802.11p).

Whenever a new neighbor (except the source of a newly received message) is inserted into $R$, $x$ (vehicle under consideration) initializes a time-out $to$-$ack$ attached to such neighbor (line no. 14). It is used to wait for the acknowledgment of reception.
6.4. THE BROADCAST PROTOCOL

Set to-ack to approximately the beacon holding time which is the maximum amount of
time a node waits without receiving beacons from a neighbor, before deleting it from
its neighbor list (line nos 53-60). This allows nodes to still receive acknowledgments
after more than one beacon interval, in case the original message was not initially
received but later it was received, from other retransmitters. That is, it allows saving
some extra retransmissions by just waiting a bit longer for those acknowledgments.

If to-ack expires and the acknowledgment has not been received, the corresponding
neighbor is moved from R to N (Line nos. 49,52), or it is removed from the lists, if its
expected beacons were not received. If N was empty and a new element is inserted,
to-ev is reactivated if it was not already running (line nos 50-51). In case to-ev was
running, it is updated according to the new value of |N| and the elapsed time since
the last schedule (line nos. 50-51). In case N becomes empty (|N| = 0), x cancels
to-ev and decides not to retransmit (line nos. 21-24). When to-ev expires, if N
is not empty, x retransmits the message and moves the content of N to R (causing
the activation of time-outs to-ack) (line nos. 42-49). For each acknowledged message
listed within a beacon from neighbor b, x cancels the associated to-ack (29-30) and
adds or confirms b in R (removing it from N if it was there) (line no. 31). Note
that some acknowledgments can be received, before the message itself, so R may be
nonempty already when the message is received for the first time.

6.4.3 Cluster Algorithm for Broadcast Protocol Details

The proposed algorithm is a distributed clustering algorithm. It possesses excellent
cluster stability, where stability is defined by long cluster-head duration, long cluster
member duration, and low rate of cluster-head change. The relative mobility between
vehicle node X and vehicle node Y is then approximated by taking the ratio of time
T taken at vehicle node Y for two successive Hello messages to arrive from vehicle
node X. The relative mobility metric,

\[ M^{rel} Y(X) \] (6.1)

, at vehicle node Y with respect to vehicle node X, is as follows:

\[ M^{rel} Y(X) = 10 \log_{10} \frac{T_{new} X \rightarrow Y}{T_{old} X \rightarrow Y} \] (6.2)
In the above metric, if
\[ T^{\text{new}} X \rightarrow Y \leq T^{\text{old}} X \rightarrow Y, \text{then} \ M^{\text{rel}} Y(X) \leq 0, \tag{6.3} \]
which implies the nodes are moving towards one another. On the other hand, if
\[ T^{\text{new}} X \rightarrow Y \geq T^{\text{old}} X \rightarrow Y, \text{then} \ M^{\text{rel}} Y(X) \geq 0, \tag{6.4} \]
which indicates that the nodes are moving away one another. Therefore, the closer
\[ M^{\text{rel}} Y(X) \text{isto} zero, \tag{6.5} \]
the lower the relative mobility. Vehicle Node Y calculates an aggregate mobility metric by considering the equation 6.1 for each neighbour, Xi. The aggregate mobility metric is found by finding the variance, with respect to zero, for the set of relative mobility values, equation 6.1. This aggregate mobility metric is computed:
\[ M^{\text{rel}} Y(X) = \text{var}\{M^{\text{rel}} Y(X)\}_{j=1}^{m} \tag{6.6} \]

Following is the proposed Cluster Algorithm for Vehicular to Vehicular communication on Mumbai-Pune Express Highway India.

- In the first executed tcl file(message.tcl), all the nodes in the cluster will send messages to others. This will help to capture their speed which will be used for finding the cluster-head for that particular cluster.

- From the trace file(messageout.tr) generated, node id and speed of vehicles are obtained and stored in variance.txt as shown in Figure 6.3 using Cluster-Head.awk file.

- The second executed tcl file(chselection.tcl) will calculate the cluster head.

### 6.4.4 Use Case Diagram for OBU is not Damaged

As shown in figure 6.4. In use case diagram it will check if a vehicle node(an actor) meets with an accident and its OBU is not destroyed then it will send the message to the head of the cluster(an actor). Cluster Head will search for other vehicle OBUs. If found then the message change path will be broadcasted to all nodes(an actor) present in the cluster. cluster head will also unicast the message to the other cluster head.
6.4. THE BROADCAST PROTOCOL

As shown in figure 6.5. In class diagram Cluster head, other clusters head, cluster nodes, accidental node are represented by class. They have attributes like id which uniquely identifies them, x cood and y cood which gives their position, acceleration, deceleration, speed which tells about their movement. They have operations like search for OBU, unicast, receive message which helps in searching OBUs, sending and receiving messages. Cluster heads can broadcast the message and hence they have an additional operation named multicast.

As shown in figure 6.6. In sequential diagram it will check if accidental node will send the message to the cluster head. Cluster head will prepare itself for sending the message. It will then send the message to cluster nodes and other cluster heads. Other cluster heads will further send it to all nodes present in its cluster. After receiving the message, cluster nodes will change their path.

As shown in figure 6.7. In activity diagram it will check When a node meets with an accident, it will send the message to its clusters head. Cluster head will check for OBUs. If OBUs are present then the head will prepare itself for sending the message. It will then send the message to cluster nodes and other cluster heads. Other cluster heads will further send it all nodes present in its cluster. After receiving the message, cluster nodes will change their path.
Figure 6.4: Use Case Diagram when OBU is not damaged
Figure 6.5: Class Diagram when OBU is not damaged
Figure 6.6: Sequential Diagram when OBU is not damaged
Figure 6.7: Activity Diagram when OBU is not damaged
6.4.5 Use Case Diagram when OBU is Damaged

As shown in figure 6.8. In Use case diagram it is mentioned that If a node (an actor) meets with an accident and its OBU gets destroyed then it cannot send the message to the cluster head. In this case, another node (an actor) which is within the range of accidental node will send the message to the cluster head. Cluster head will search for other OBUs. If found then the message change path will be broadcasted to all nodes (an actor) present in the cluster. Cluster head will also unicast the message to the other cluster head.

As shown in figure 6.9. In class diagram it is mentioned that if cluster head, other clusters head, cluster node, accidental node, node within range are represented by class. They have attributes like id which uniquely identifies them, x cood and y cood which gives their position, acceleration, deceleration, speed which tells about their movement. They have operations like search for OBU, unicast, receive message which helps in searching OBUs, sending and receiving messages. Cluster heads can broadcast the message and hence they have an additional operation named multicast. Accidental node depends on node within range for sending message to the cluster head.

As shown in figure 6.10. In sequential diagram it is mentioned that if here the node which is within the range of accidental node will send the message to the cluster head. Cluster head will prepare itself for sending the message. It will then send the message to cluster nodes and other cluster heads. other cluster heads will further send it all nodes present in its cluster. After receiving the message, cluster nodes will change their path.

As shown in figure 6.11. In Activity diagram it is mentioned that if When a node meets with an accident, the other node which is within the range of accidental node will send the message to its clusters head. Cluster head will check for OBUs. If OBUs are present then the head will prepare itself for sending the message. It will then send the message to cluster nodes and other cluster heads. other cluster heads will further send it all nodes present in its cluster. After receiving the message, cluster nodes will change their path.
Figure 6.8: Use Case Diagram when OBU is damaged
Figure 6.9: Class Diagram when OBU is damaged
Figure 6.10: Sequential Diagram when OBU is damaged
Figure 6.11: Activity Diagram when OBU is damaged
6.5 Discussion

Broadcast Protocol is an appropriate solution for VANET. First, the protocol is scalable because it only needs local or status information to perform the broadcasting task. Local or status information is obtained from beacon messages. This does not increase message overhead, because they are needed by safety applications and are mandated by on-going standards like DSRC protocol [ASTM]. The only additional overhead comes from the inclusion of the acknowledgment messages, inside periodic beacons, since the sender’s position is included by default. Acknowledgment message appear the best strategy in broadcast protocol using RTS and CTS, to guarantee delivery to all vehicles on Mumbai-Pune Express Highway. Receivers may malfunction, and physical layer modeling has large randomness component even if made with accurate parameters.

In order to minimize the number of message transmissions while preserving reliability, Protocol creates a broadcast delivery backbone based on a CDS heuristic and cluster formation algorithm. Vehicles in the CDS choose a shorter time-out, to give them higher priority to retransmit messages. In addition, cluster algorithm is employed to further reduce the number of redundant transmissions messages. Cluster algorithm will form cluster of vehicles on road with same speed in same direction etc. then elect the cluster head from the speed capture on Mumbai-Pune Express Highway. This cluster head will communicate with the Cluster member for transmission of messages related local information or warning information. This cluster formation will be long time and information will be shred among all members which reduces redundancy. This approach is appropriate for vehicular scenarios such as Mumbai-Pune Express Highway layouts with different intersections points on Highway road. Vehicles located at junctions which are the only ones with connectivity, with other vehicles at converging streets, will be selected as dominating, therefore, will retransmit sooner to propagate the broadcast message along those streets (see Figure 6.2). This is achieved by means of the own CDS selection mechanism for multi-hop, without ever dealing directly with the notion of ‘intersection’ in the protocol description. Note that VANET-specific protocols (including those designed for safety applications) in which the forwarder selection is based on the concept of progress from the trans-
6.5. DISCUSSION

Figure 6.12: (Intersections in vehicular scenarios. Dotted lines represent connectivity between the subset of vehicles surrounding the intersection. Vehicle $s$ initiates the broadcasting task. In PBSM, $b$ is used as relay and therefore the message propagates for every converging street. Forwarding progress-based approaches would select $c$ as relay, and the message would only propagate through the current street.)

mitter (Sjoberg) (Munoz), fail to support this scenario. In Figure 6.2, if vehicle $c$ receives $s$ transmission and forwards first (since it is farther from $s$ than $b$), vehicles $a$, $f$ located at converging streets would not receive the message. Other approaches (Korkmaz et al.), (Korkmaz, Ekici, and Uner) need to explicitly handle the case of intersections by starting new directional broadcasts.

In the unit disk graph (UDG) model, two nodes $u$, $v$ are neighbors can directly communicate if distance $(u, v) \leq r$, where $r$ is the radius of the communication range. We demonstrate that CDS concept used here is effective in VANET. Actual CDS definition in realistic physical layer is complicated because physics is complicated: the link between any two vehicles is probabilistic so it is not even clear, when to declare them neighbors. CDS was indeed here defined using UDG as approximation, but then it shows that such use of simple approximated CDS is just enough for satisfactory performance of Broadcast Protocol under realistic VANET physics. Computing a CDS in a VANET environment comes for free, since beacons with geographic information are periodically triggered. The use of acknowledgments makes the protocol more suitable to the VANET fading environment. If a message is not received by a theoretical
neighbor, latter does not announce its reception in subsequent beacons and the vehicles with the message will issue new transmission. If the message is received by a theoretical non-neighbor, there will be no retransmission later, if that node suddenly becomes a neighbor. The superiority of Broadcast Protocol over PBSM is explained by this correction of UDG-based initial estimate. PBSM updates lists $R$ and $N$, implicitly assuming the UDG model. The inclusion of acknowledgments in ABSM protects the protocol against this assumption, since message losses are expected to happen. This allows Broadcast Protocol to perform well in real environments.

Finally, and contrary to protocols like DV-CAST, researcher solution does not need to determine the traffic regime that is sensed by the vehicle. This simplicity is a great advantage: since there are, no different internal states, flaws due to unexpected situations are less prone to appear. Nowhere in the Broadcast Protocol, it matters what is the speed of a vehicle or if the vehicle is at an intersection. It therefore provides smooth adaptation to network dynamics including intersections, without changing its behavior. For comparison, GPCR protocol (Lochert et al., “Geographic Routing in City Scenarios”) changes when vehicle is at intersection. Also, determining which nodes, are located at intersections requires downloading maps in addition to position information.

### 6.6 Evaluation Setup

Researcher have performed different tests, to assess the performance of Broadcast Protocol. The simulation work has been done with the Network Simulator NS-2, version 2.34. Along with Broadcast Protocol, researcher also implemented competing algorithm DV-CAST and two variants of PBSM: PBSM-2t, which uses two-hop topology information as described in [Khan, Stojmenovic, and Zaguia]; and PBSM-1p, employing one-hop position information. PBSM-1p, PBSM-2t, and Broadcast Protocol implement the CDS heuristic described in [Stojmenovic, Seddigh, and Zunic] and [Stojmenovic]. Researcher have used vehicles unique identifiers as keys. In all PBSM variants, the time-out $to - ev$ is computed as in equation 6.1, while $to-ack$ is fixed to a constant value in Broadcast Protocol. The effect of parameters $W$ and
6.6. EVALUATION SETUP

to-ack is studied in later section.

\[
\text{to-ev} = \begin{cases} 
\frac{W}{|N|} \times 1, & \text{if in CDS within Cluster;} \\
W(1 + \frac{1}{|N|}), & \text{otherwise (CDS outside Cluster).} 
\end{cases}
\] (6.7)

For DV-CAST, implementation employs the weighted p-persistence algorithm as the broadcast suppression technique, with parameters \( W = 0.25 \text{ sec} \) and \( \delta = \frac{W}{10} \text{ sec} \). The other slot-based approaches were not chosen because, as recognized by the authors, they depend on parameters which may be hard to tune in practice [Wisitpongphan et al.]. To determine the vehicle status, DV-CAST uses concepts such as, the message forwarding direction, the position inside a cluster of vehicles, and the presence of neighbors in the same or opposite direction [Tonguz et al.]. The position of the sender and its direction is included in periodic beacons. In addition, data packets are augmented with a network header that indicates the position and direction of the source, as well as the position of the last forwarder (previous hop). Such information suffices to derive the status of each vehicle.

Researcher consider, specific vehicular scenarios and movements for setups, namely, highway. The former consists of a 4 km long rectilinear highway with two lanes per direction. The former consists of a 4 km² with two crossing streets that converge at the center of the square. Each street has two lanes in opposite direction. Vehicles must stop at intersections when others are crossing, so that traffic jams are longer here. DV-CAST has not been included in this set of simulations, because it is not designed for highway scenarios with intersections.

In order to create Mumbai-Pune Express Highway scenario, as well as, to generate the mobility traces of the vehicles at different time slot, researcher have employed the SUMO microscopic road traffic simulation package. This allows to simulate, common vehicular situations such as overtakes and stops at intersections points on Highway. This leads to intermittent connectivity and uneven distribution of vehicles. In this scenario, researcher defined several routes which are followed by the vehicles. SUMO injects vehicles in each route according to a given traffic rate, measured in injected vehicles per second. In order to get a wide range of network connectivity, researcher have varied the traffic injection rate per route from \( \frac{1}{175} \) to \( \frac{1}{5} \) vehicles per second. The higher the traffic injection rate, the higher the network density. Some figures and
6.6. EVALUATION SETUP

Tables in this section are labeled with the reciprocal of this rate, i.e., with the interval between the injection of consecutive vehicles (from 75 to 5 sec). Two types of vehicles have been defined, with maximum speeds of 50 km/h and 80 km/h. Refer Figure 6.12 to Figure 6.15.

Table 6.1 summarizes the main simulation parameters used for Mumbai-Pune Express Highway Scenario. Beacon interval refers to the time between consecutive beacons. The information acquired is considered valid during the beacon hold time. Each run consists of one broadcasting task that is started by a random source, chosen from a cluster of vehicles that meet some requirements. Namely, the vehicle must be active when the steady state of the network is reached, and it must have at least 30 sec remaining before reaching its destination. The broadcasted message contains 500 bytes of payload and has a lifetime of 120 sec, afterwards it is discarded. Results show the average value of 20 independent runs, along with the 95% confidence interval.

List of metrics are as follows:

- **Reliability**: Defined as the ratio between the number of vehicles which receive the broadcast message and the total number of them that could have received it: \( \text{Rel} = \frac{N_{\text{recv}}}{N_{\text{total}}} \); \( \text{Rel} \in [0, 1] \). Note that, probably, not every simulated node, can receive the broad-casted message because some vehicles may remain partitioned...
Figure 6.14: Vehicle Parameters Captured for Mumbai-Pune Highway Screen Shot in SUMO
6.6. EVALUATION SETUP

Figure 6.15: Vehicle Graph for Mumbai-Pune Highway Screen Shot in SUMO

Figure 6.16: Cluster Head Selection using CDS Screen Shot on Mumbai-Pune Express Highway in VANET
### Table 6.1: Simulation Parameters for the Mumbai-Pune Highway and Nerul-Vashi City Vehicular Scenario

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Time</td>
<td>120, 1200, 2000 sec</td>
</tr>
<tr>
<td>Area</td>
<td>$6780 \times 7800$ meter, 120 km</td>
</tr>
<tr>
<td>Traffic rate</td>
<td>$(1/75, 1/60, 1/45, 1/30)$ veh/sec/route</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>$(50, 80, 120)$ km/h</td>
</tr>
<tr>
<td>Beacon Interval</td>
<td>0.5, 0.75, 1 sec</td>
</tr>
<tr>
<td>Beacon Hold Time</td>
<td>1.5, 2.0 sec</td>
</tr>
<tr>
<td>W</td>
<td>$(0.1, 0.25, 0.5, 1, 1.5, 1.75, 2)$ sec</td>
</tr>
<tr>
<td>to-ack</td>
<td>$(0.6, 1.1, 1.6, 2.1, 2.6)$ sec</td>
</tr>
<tr>
<td>Transmission Power</td>
<td>1.52, 2.0 mW</td>
</tr>
<tr>
<td>Carrier sense Threshold</td>
<td>$802.11p : -94, -96$ dBm</td>
</tr>
<tr>
<td>Contention Window</td>
<td>$802.11p : [15 - 1023]$</td>
</tr>
<tr>
<td>RTSThreshold</td>
<td>2346</td>
</tr>
<tr>
<td>SlotTime</td>
<td>0.000013</td>
</tr>
<tr>
<td>frequency</td>
<td>$5.85e + 9$</td>
</tr>
<tr>
<td>bandwidth</td>
<td>70e6</td>
</tr>
<tr>
<td>Data</td>
<td><em>Local/EmergencyMessage</em></td>
</tr>
<tr>
<td>Transmission Range</td>
<td>100m – 600m</td>
</tr>
<tr>
<td>Traffic direction</td>
<td><em>TwoWay</em></td>
</tr>
<tr>
<td>Number of Vehicle Nodes</td>
<td>11, 60, 232, 1218, 2000</td>
</tr>
<tr>
<td>Intersection points</td>
<td>4</td>
</tr>
</tbody>
</table>
from the source. In order to overcome this issue, researcher measure $N_{total}$ on each simulation as follows: we have implemented and simulated with ideal MAC and PHY layers a variant of hyper flooding [Viswanath and Obraczka]. The number of covered nodes $N_{recv}$ obtained on such simulations, becomes the upper bound $N_{total}$ for the remaining protocols.

- **Number of message transmissions per involved vehicle**: This measures the efficiency of the protocol. Given the same reliability, a protocol is said to be more efficient than another, if it needs fewer transmissions to complete the broadcasting task. The number of involved vehicles $N_{total}$ has been computed as explained before.

- **Control overhead per vehicle**: Since the protocols are localized, the overhead comes from the periodic exchange of beacon messages. Our DV-CAST implementation also adds information, as an extra header within data packets. The total number of bytes devoted to protocol information per simulated vehicle, during every run, has been measured.

- **Delivery latency**: Measured as the time, in seconds, since the data source issues the message until it arrives at every receiver. For this metric, focus on one specific run.

- **Message Delivery Ratio (PDR)**: It is the fraction of packets generated by received packets. That is, the ratios of packets received at the destination to those of the packets generated by the source. As of relative amount, the usual calculation of this system of measurement is in percentage (%) form. Higher the percentage, more privileged is the routing protocol.

- **Average End-to-End Delay (E2E Delay)**: It is the calculation of typical time taken by message (in average packets) to cover its journey from the source end to the destination end. In other words, it covers all of the potential delays such as route discovery, buffering processes, various in-between queuing stays, etc, during the entire trip of transmission of the message. The classical unit of this metric is millisecond (ms).
6.7 Results

Researcher implemented CDS for multi-hop using NS2 version 2.35 and OTCL using C++ language. Researcher consider Mumbai-Pune Express Highway scenario and collected result for number of cluster heads i.e. cluster quality, reliability, PDR and throughput for reliable and efficient vehicle-to-vehicle communication on Mumbai-Pune Express Highway road without infrastructure. During experiments different parameters like number of nodes, mobility and density of network is considered. In the tests as the surface area decreases the density off the graph increases, it means that nodes will have more neighbors in lesser area. Speed is determined randomly by SUMO simulation within specified velocity limits. Observed the number of cluster-heads in a graph of size 20 to 50 for varying densities from 4 to 13. Typically in a graph, expect to have less cluster-heads as density increases. This can see the decrease in the cluster-head numbers in the graph as the degree value increases in Figure 6.16 and Figure 6.17 tell the throughput on Mumbai-Pune Highway. Broadcast protocol performs better in given scenario. Algorithm has less number of cluster-heads using CDS calculation for multi-hop network as compared to previous CDS algorithm, then better throughput and better reliability, PDR and E2E results as shown in below results.

As defined, performance of our CDS-based depend on a parameter $to - ev$ that represents the wait time before retransmitting a given message. In addition, Broadcast Protocol incorporates an additional time-out $to - ack$, which is employed to wait for acknowledgments of a forwarded message with respect to cluster formation algorithm. They also rely on two more parameters, namely, the beacon interval and holding time. However, latter related to $to - ev$ and $to - ack$, which are the relevant parameters to study since they are exclusive of the evaluated broadcast protocols. Investigate how parameters $W$ and $to - ack$ influence the behavior of the protocols, fixing the beacon interval and holding time, as shown in Table 6.1.

Figure 6.18 and 6.21 shows the impact of parameter $W$ onto each protocol’s reliability, where $to - ack = 1 : 6 \ sec$ in the case of Broadcast Protocol. Focus on a moderately dense network (30 sec of injection interval) and a moderately sparse one (60 sec of injection interval) on Mumbai-Pune Express Highway. Regardless the
Figure 6.17: Number of Cluster Head on Mumbai-Pune Express Highway in VANET

Figure 6.18: Throughput on Mumbai-Pune Express Highway in VANET
6.7. RESULTS

Figure 6.19: PDR on Mumbai-Pune Express Highway in VANET

Figure 6.20: E2E on Mumbai-Pune Express Highway in VANET
6.7. RESULTS

Figure 6.21: Effect of parameter $W$ onto reliability for different traffic injection intervals for Mumbai-Pune Express Highway Scenario. (Interval = 30 sec)

Figure 6.22: Effect of parameter $W$ onto reliability for different traffic injection intervals for Mumbai-Pune Express Highway Scenario. (Interval = 60 sec)
Table 6.2: Reliability Result (%) on Mumbai-Pune Express Highway Scenario for different Injection Intervals (Sec between Injected Vehicles per Route)

<table>
<thead>
<tr>
<th>Interval</th>
<th>Reliability (Broadcast Protocol)</th>
<th>Reliability (DV-CAST Protocol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>97.1 ± 2.1%</td>
<td>16.7 ± 3.1%</td>
</tr>
<tr>
<td>60</td>
<td>97.0 ± 0.5%</td>
<td>20.8 ± 3.8%</td>
</tr>
<tr>
<td>45</td>
<td>96.3 ± 0.6%</td>
<td>20.2 ± 4.6</td>
</tr>
<tr>
<td>30</td>
<td>100 ± 0%</td>
<td>63.4 ± 8.1%</td>
</tr>
<tr>
<td>15</td>
<td>100 ± 0%</td>
<td>72.6 ± 0.1%</td>
</tr>
<tr>
<td>5</td>
<td>100 ± 0%</td>
<td>85.7 ± 0.3%</td>
</tr>
</tbody>
</table>

protocol under consideration, disconnected networks can benefit from low $W$ values. This makes sense because some vehicles might remain as neighbors during a very short period of time may be in 10 ms. If the evaluation time is too high, the link between those vehicles might not exist any longer and the forwarding opportunity would be lost. For denser networks, each protocol tends to converge at 100% reliability, for every evaluated value of $W$. On the other hand, low $W$ values generally provoke more redundant transmissions, especially in congested Mumbai-Pune Express Highway Road.

The waiting time before retransmission gets higher, the number of needed forwarding decreases. This phenomenon is explained by the use of cluster algorithm, since the neighborhood might receive the message from other retransmissions. However, high $W$ values augment the delivery latency of the broadcasting task, especially in sparse networks. This parameter is not relevant with respect to the protocol overhead. Taking these results into account, recommend low $W$ values for disconnected networks, and slightly higher values for dense ones. Broadcast Protocol behaves very well in all the studied cases when compared to the other approaches. Broadcast Protocol provides high reliability, scalability and efficiency for broadcasting on Mumbai-Pune Express Highway (see Table 6.2) as compared to DV-CAST Protocol.
6.7. RESULTS

It is not surprising because they are based upon the Cluster forwarding framework, which is meant to cover the whole network. Among them, Broadcast Protocol achieves the best results. The lowest reliability offered by this scheme, is the 97.1% of the vehicles that could have received the message within cluster and inter-cluster network. On the other hand, DV-CAST offers a very poor reliability, PDR and throughput as compared to Broadcast Protocol for sparse networks, while it only covers around the 75–85% of vehicles when the highest traffic rates are simulated. The reason is that the protocol does not foresee common vehicular movements such as passing maneuvers. For example, refer Figure 6.2. Assume that vehicle $f$ initiated the broadcasting and the message has been propagated backward up to $a$. All vehicles are in idle state except $a$, which has the forwarding responsibility at that moment. Then, $a$ speeds up and overtakes the remaining vehicles, forwarding the message to them, and going to idle state. According to DV-CAST [Tonguz et al.], the receivers, discard the message as duplicated and the message custody is lost. No one will forward the message again, even when new vehicles $g; h$ emerge. This problem is derived from different states in which DV-CAST operates depending on the traffic regime which is sensed by a vehicle, since it is hard to foresee every possible combination of movements in vehicular setups. Approach does not suffer from this problem.

Focus now on the number of forwarding for each protocol, shown in Figure 6.22. Given the low reliability of DV-CAST, the number of broadcast messages issued by the protocol is also low. Interestingly, Broadcast Protocol obtained the best reliability, throughput, PDR and E2E delay at the expense of almost as few transmissions as DV-CAST provokes. Furthermore, the number of broadcast messages, issued by Broadcast Protocol is almost constant with respect to the simulated traffic flow rate. This indicates the suitability of Broadcast Protocol as a scalable solution for broadcasting on Mumbai-Pune Express Highway Road, India. It takes advantage of the piggy backed acknowledgments, to reduce the protocol redundancy. When a vehicle node contacts a new neighbor for the first time, new forwarding’s are avoided if the latter has already received the message.

However, in the scenario, Broadcast Protocol needs around one forwarding per vehicle. This can achieve high reliability in disconnected networks without requiring every receiving node to retransmit the message.
Figure 6.23: Number of data transmissions per vehicle involved on highway

Figure 6.24: Control Overhead (KB) per vehicle on highway
Simulation result shows that Broadcast Protocol can achieve the best reliability, throughput, PDR and E2E delay results, with the number of transmissions. More important, the redundancy trend of the protocol remains almost constant as density of the network changes. Hence, Broadcast Protocol scales with respect to this parameter. Also investigated, the control overhead introduced in periodic beacon messages, by each protocol. Figure 6.23 draws the values of this metric for different injection intervals. Broadcast Protocol overhead is slightly higher, because it also needs to include an identifier for each received broadcast message. Given the huge reduction in data message transmissions (Figure 6.22), Broadcast Protocol is still the most efficient approach, of all the evaluated ones. Our implementation of DV-CAST is heavier because direction information is added to the beacons. Additionally, it also includes control information inside data messages.

Investigated, the delivery latency that is experienced by Broadcast Protocol to check the message propagation delay, focuses on one run of a dense highway scenario (traffic injection interval is set to 15 sec between vehicles per route) and measure the time, since a message is generated until it is successfully decoded by every receiver.

Figure 6.25: Delivery Latency (sec) for every receiver in highway scenario.
6.7. RESULTS

Figure 6.26: Reliability for Highway Scenario.

Figure 6.27: Overhead Message for Highway Scenario.
6.8. SUMMARY

Figure 6.28: Number of Hops for Highway Scenario.

The results of this experiment are shown in Figure 6.24 to Figure 6.27 shows the results of new broadcast protocol for reliability, message overhead for dense network and number of hops for dense network. From the observation it is clearly observed that new broadcast protocol performs better as compared to DV-CAST protocol. The reception time of the message by each vehicle, have formed groups of 40 vehicles, and the average delay of each group is shown along the y-axis. It can be seen that Broadcast Protocol deliver the message faster than the other approaches. With respect to DV-CAST, it incurs larger delays, under common retransmission parameter \( W \). In the connected part of the network, the weighted p-persistence broadcast suppression technique is applied. Hence, the waiting time before retransmission is constant (either \( W \) or \( W + \delta \)), contrary to our adaptive approach, in which this value depends on the local density of the network. Besides, when there are disconnected groups of vehicles that eventually merge, there is an increased latency for every protocol. However, this is higher in DV-CAST because only a subset of vehicles that own the message custody are the ones that can forward it. DV-CAST reaches fewer vehicles, than our solutions, since it cannot reach group of vehicles 161 – 200 and 201 – 240.

6.8 Summary

New Broadcast Protocol, which is a localized broadcast protocol for vehicular ad-hoc networks. It is built upon the Cluster algorithm framework. It implicitly uses the
store-carry-forward paradigm, typical of delay-tolerant networks and cluster formation algorithm. The pseudo-code employs the position information of the multi-hop neighborhood, acknowledgments and cluster formation of the latest received broadcast messages and warning messages, improve protocol reliability, scalability and efficiency. Broadcast Protocol not only calculate reliability but also scalability and efficiency which makes the protocol better than DVCAST protocol and Khalid Abdel Hafeez protocol. Khalid Abdel Hafeez protocol does not resolves hidden terminal problem and broadcast storm problem totally only reduces redundancy but Broadcast protocol resolves problem of hidden terminal and broadcast storm problem using CDS and cluster algorithm which improves reliability, scalability and efficiency on Mumbai-Pune Express Highway India.
Algorithm 1 Pseudo-code of The Broadcast Protocol

1: \( B \leftarrow \) neighbor set of this node;
2: \( r \leftarrow \) communication range;
3: \( r \leftarrow 1000 \) meters;
4: \( R \leftarrow 0 \);
5: \( N \leftarrow 0 \);
6: \( \textbf{Event} \) cluster formation and cluster head selection using relative mobility metric
   for distributed algorithm;
7: \( \textbf{Event} \) message copy received from neighbor \( s \) or generated by this node \( s \);
8: Insert message id in subsequent beacons;
9: \( CM \leftarrow CH \);
10: \( CG \leftarrow CH \);
11: \( R \leftarrow R \cup \{s\} \);
12: \( N \leftarrow N \setminus \{s\} \);
13: \( n \in \{B\} \);
14: \( \textbf{for} \ n \in \{B\} \ \textbf{do} \)
15: \( \textbf{if} \ \text{dist}(n,s) < r \ \textbf{then} \)
16: \( R \leftarrow R \cup \{n\} \);
17: \( N \leftarrow N \setminus \{n\} \);
18: \( \text{Schedule to-ack for} \ n \);
19: \( \textbf{else if} \ n \notin R \ \textbf{then} \)
20: \( N \leftarrow N \cup \{n\} \);
21: \( \textbf{else} \ \text{cancel to-ack} \);
22: \( \textbf{end if} \)
23: \( \textbf{end for} \)
24: \( \textbf{if} \ s = \text{source} \ \textbf{then} \)
25: \( \text{forward message via 802.11} \);
26: \( \textbf{else if} \ N = 0 \ \textbf{then} \)
27: \( \text{cancel to-ev} \)
else

Schedule to-ev;

end if

Event beacon received from neighbor n

Add n to neighbor set with in cluster;

Compute CDS;

if beacon contains ack then

cancel to-ack for n;

R ← R∪\{n\};

N ← N/{n};

else if n /∈ R then

if n /∈ R then

Schedule to-ev;

end if

N ← N∪\{n\};

else if N ≠ 0 then

R ← R∪N;

dead end if

Event to-ev expires;

if N ≠ 0 then

R ← R∪N;

for n ∈ N do

schedule to-ack for n;

end for

N ← 0;

forward message via 802.11;

dead end if

Event to-ack expires for neighbor n and ack from n never received

R ← R/{n}; If n /∈ N

schedule to-ev;

N ← N∪\{n\};
60: **Event** beacon from n not received for last beacon-hold-time

61: if $N = \{n\}$ then

62: \hspace{1em} cancel to $-ev$;

63: end if

64: $N \leftarrow N/\{n\}$;

65: Remove n from neighbor set from cluster;

66: Compute CDS;