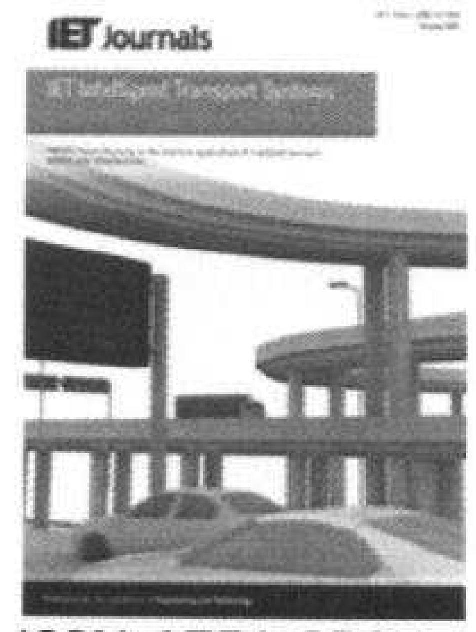
Published in IET Intelligent Transport Systems
Received on 15th November 2007
Revised on 12th November 2008
doi: 10.1049/iet-its:20070052



ISSN 1751-956X

Robust mobility adaptive clustering scheme with support for geographic routing for vehicular *ad hoc* networks

R.T. Goonewardene F.H. Ali E. Stipidis
Engineering and Design, The University of Sussex, Brighton, UK
E-mail: e.stipidis@sussex.ac.uk

Abstract: There are a number of critical problems related to road safety in intelligent transportation systems (ITS) caused by increased vehicle usage, urbanisation, population growth and density, and faster rates of movements of goods and people. It is envisaged that vehicular *ad hoc* networks (VANETs) will bring about a substantial change to the way our road transport operates to improving road safety and traffic congestion. A major challenge in VANETs is to provide real-time transfer of information between vehicles within a highly mobile environment. The authors propose a new clustering scheme named robust mobility adaptive clustering (RMAC) to strategically enable and manage highly dynamic VANETs for future ITS. It employs a novel node precedence algorithm to adaptively identify the nearby 1-hop neighbours and select optimal clusterheads based on relative node mobility metrics of speed, locations and direction of travel. Furthermore, the zone of interest concept is introduced for optimised approach to the network structure such that each vehicular node maintains a neighbour table of nodes, beyond its communications range, that reflects the frequent changes on the network and provides prior knowledge of neighbours as they travel into new neighbourhoods. RMAC predominantly employs more reliable unicast control packets and supports geographic routing by providing accurate neighbour information crucial when making routing decisions in multi-hop geographic routing. It is shown by simulations that RMAC on IEEE802.11 *ad hoc* WLAN protocol is very effective in a highly dynamic VANETs environment, being robust on link failures, and having very high cluster residence times compared to the well known distributed mobility clustering scheme.

1 Introduction

It is perceived that enabling inter-vehicular communications (IVC) for real-time information availability in ITS could be a major enhancement in road safety and passenger comfort. Local communications technologies based on vehicle-to-vehicle and vehicle-to-roadside can offer to the vehicle real-time data for its immediate environment that can assist motorists in many aspects of their driving and facilitate an efficient wider traffic management network. In recent years, much focus has been on VANETs, where each vehicle is a node within a mobile local network. This type of IVC has the potential of becoming an enormous uncontrollable 'beast' purely because of the prospective huge amount of information transmitted over a limited capacity, limited quality of service wireless medium. Furthermore, currently there are number of technologies that would be suitable for VANETs and almost certainly more advanced technologies will appear in the future so as to support newly emerging ITS applications (e.g. predictive traffic management). The real challenge is to provide a reliable IVC strategy-based VANETs that can be applied across different wireless networks (technology independent) and support current and future applications within ITS (application independent). Some of the necessary features would be modularity, reliability, flexibility, enhance ability, scalability, coverage and being dynamic.

In VANETs, and more in general in ad hoc wireless networks, clustering can be used to strategically partition the network into smaller segments. This has many benefits

© The Institution of Engineering and Technology 2009

including optimising bandwidth usage, distribution of resources [1] and resolving scalability issues in combination with routing schemes such as Geographic or Topology-based. In highly dynamic networks, geographic routing normally performs better [2, 3] as all nodes in the network are required to know their 1-hop neighbours. Typically, a location service is used to determine the geographic location of destination nodes [3], an example of such a service is the location server described in [4].

Clustering schemes can be classified into two, clusters with clusterheads (CHs) and ones without. Clusters without CHs avoid overloading a subset of nodes in the network, making the operation of all nodes equal [5]. CHs serve many purposes within a cluster, such as the allocation of resources to member nodes and coordinating transmission events for nodes in the cluster in order to avoid retransmissions by reducing packet collisions [6]. Clusters controlled by CHs can be organised as either 1-hop clusters or multi-hop clusters (also known as *k*-hop clusters). In 1-hop clustering schemes such as [1, 7], cluster members (CM) are within transmission range of the CH, that is, within 1-hop of the CH. In multi-hop clustering schemes such as [8] and [9], the maximum distance between CHs and CM is k hops, such that CM may reside outside the communication range, where intermediate CM relay messages between CHs and those members.

Cluster stability depends on the selection of a suitable CH to ensure greater cluster residence times by reducing cluster change events. There are a number of algorithms for selecting CHs. In [10] the lowest-id criterion is used. Each node in the network has a unique id and during clustering, nodes with the lowest id in the neighbourhood are elected to be CHs. In [11] nodes with the highest number of neighbours are chosen to be CHs. In the Distributed Mobility-Adaptive Clustering (DMAC) algorithm [1], a weight-based criterion is used. Each node is assigned a weight (a real number ≥ 0), the greater the weight the more suited the node is for the role of CH.

A common assumption among most clustering algorithms is that during the clustering phase nodes remain static [1]. In [12], the authors very well defined SCOPE as the local area of interest to support routing taking as de-facto cluster formation, with CHs used as dynamic host configuration protocol (DHCP) servers, which implies that CH selection and cluster formation is more static than adaptive and with unnecessary IVC overhead in allocating IP addresses (Network Layer) via DHCP [13]. However, this is an unreasonable expectation in mobile *ad hoc* networks and is addressed by DMAC, which reacts to variations in the surrounding topology caused by new links and link failures, thus re-assigning CHs as and when required. CHs in DMAC form a dominant independent set, permitting only one CH in the neighbourhood. When CHs come within range, reorganisation of the clusters is required, which may affect surrounding clusters and possibly the entire network, this is known as the ripple effect, and is worsened in highly dynamic conditions. It causes clusters instability and is costly in terms of bandwidth.

In addition, with existing clustering algorithms (e.g. DMAC) and geographic routing algorithms (e.g. Greedy perimeter stateless routing (GPSR) [14]), broadcast beacon packets are utilised for nodes to determine their 1-hop neighbours and to detect changes in their neighbourhood. Wireless networks suffer greatly by the presence of hidden nodes. This is where multiple transmitting nodes with a common receiver are hidden from each other, due to the transmitters being out of range of each other. Hence, when more than one node transmits at the same time to the same receiver, collisions occur at the receiver. This problem is solved by transmitting unicast packets and relying on virtual carrier sensing such as the RTS/CTS mechanism of IEEE802.11, which eliminates the problem of hidden nodes, by allowing transmitters and receivers to reserve the channel prior to transmission of the data frame, with the exchange of the RTS/CTS frames. This technique also significantly reduces the probability of collision where the RTS frame is smaller than the data frame [15]. Furthermore, with broadcast beaconing when nodes constantly move into new neighbourhoods in such dynamic environments, they must await beacon packets in order to learn their 1-hop neighbours.

In this paper we propose the Robust Mobility Adaptive Clustering (RMAC) as a technology independent scheme residing in the Data Link Control sub-layer to provide a cluster-based backbone supporting geographic routing to the next layer above (Network Layer) and managing the dynamic topology and CH weighted selection of highly mobile vehicular networks. It operates autonomously in a decentralised manner at every node. Protocols such as the 'IP address Passing for VANETs' described in [13] can take advantage of this layered architecture capability and significantly improve efficiency, reduce latency and increase IVC. It is a strategic approach where the network is unified and allows certain operational capability defining the communication messages outside the vehicle. In turn these messages are transmitted to the appropriate recipients including ultimately to a traffic management information system. This offers the basis of an enhanced system whereby ITS can operate efficiently within the immediate vicinity of the vehicle in real time with increased information availability used for safety and non-safety related applications. The unified VANET lends itself to automatic collision notification, emergency management, traffic congestion notification and many other applications as simple plug-ins on top of a sustainable (future proof) layered technology independent architecture. The network formed by RMAC is used to disseminate neighbour information using predominantly unicast packets synchronised by the CHs, such that nodes can construct neighbour tables in order to support geographic routing, by providing accurate location information of neighbouring

© The Institution of Engineering and Technology 2009

vehicles. Section 2 describes the principles of the RMAC scheme. Section 3 describes the RMAC algorithms. Section 4 presents the simulation results and analysis, with a comparison to DMAC, an existing mobility adaptive clustering algorithm. And finally, Section 5 concludes the paper with some further work.

2 Principles of RMAC

Most clustering schemes are generally designed for generic MANETs. However, RMAC is designed specifically for high-speed *ad hoc* vehicular networks, where vehicular mobility metrics, speed, location, and direction of travel are considered for clustering. These parameters are now widely available within vehicles as part of satellite navigation systems incorporating global positioning system (GPS) units. In addition, the proposed scheme is not sensitive to small positional errors and therefore minor inaccuracies of the positioning system are not detrimental. RMAC creates robust 1-hop clusters that are overlapped to form a backbone for nodes to communicate. This backbone is utilised to disseminate neighbour information via unicast packets, in order to support geographic routing protocols by proactively maintaining accurate locations of neighbouring nodes. However, RMAC does not incorporate a geographic routing protocol.

2.1 Design requirements

RMAC has been designed to meet the following requirements.

• *Scalable:* The scheme must be capable of coping with small to extremely large networks. For example, motorway segments can span a few miles with a few hundred vehicles, to a few hundred miles with thousands of vehicles.

• *Mobility Adaptive:* The algorithm must operate in conditions where node mobility is high, on average with speeds in excess of 30 m/s. The clusters should be capable of adapting to cope with such dynamic conditions, maintaining the infrastructure of the network when links fail, and detecting new links when nodes enter the neighbourhood.

• *Synchronised:* The scheme will use CHs to synchronise exchange of control packets.

• *Robust:* Since CHs are responsible for synchronising the clusters, CH failure can cause disarray in that cluster, leaving its members excluded from the network, and possibly causing fragmentation of the network. In the event of a link failure to a CH, members must be capable of a quick recovery to maintain the integrity of the network, therefore eliminating any single point of failure.

• *Distributed and decentralised:* The algorithm could operate in an environment with no fixed infrastructure. Although nodes may assume different roles within clusters over time, their roles are not pre-defined, and constantly change due to mobility. Therefore the network can be considered decentralised and equally distributed.

• *Support for Geographic Routing:* Ultimately, after a communications backbone is established, the scheme must be capable of supporting multi-hop geographic routing by maintaining and providing accurate 1-hop neighbour information.

2.2 Cluster definition

In RMAC a cluster is defined as one or more CM connected to a CH. Hence, clustered nodes can assume the role of CH (the controller of the cluster), CM, or operate in Dual State. A Dual state node is a CH of its own cluster and a CM of one or more other clusters, for example, node f in Fig. 1. RMAC forms 1-hop clusters, where all the CMs are within communication range of their CH. However, not all nodes in range of CH necessarily belong to that cluster. When members leave transmission range of their CH, they are no longer part of that cluster. Connections are sustained by periodic polling, which is part of cluster maintenance, and ensures that link failure is quickly detected. Full connectivity of the network is achieved by overlapping clusters, whereby nodes are permitted to participate in multiple clusters. Furthermore, multiple CHs can operate in close proximity without the need for re-clustering, thus eliminating ripple effects. The CMs of a CH is associated with the set M, and the CHs of a node is associated with H. For example in Fig. 1, for node f, $M_f = \{g, h\}$ and $H_f = \{e\}$.

2.3 The node precedence algorithm (NPA)

Generally with existing clustering algorithms the criterion for selecting a CH is based on values such as node id or computed weight that is applied globally, where the nodes with the greatest weights in their neighbourhoods are elected CHs. This is so that all nodes in the network unanimously agree on the same CHs, in order to meet the criterion that only one CH can preside in a neighbourhood. In a mobile environment a single node cannot necessarily be the CH for all it neighbours, due to the varying mobility of the nodes. Therefore a node should be considered by each of its neighbours with a different weight, based on the relationship

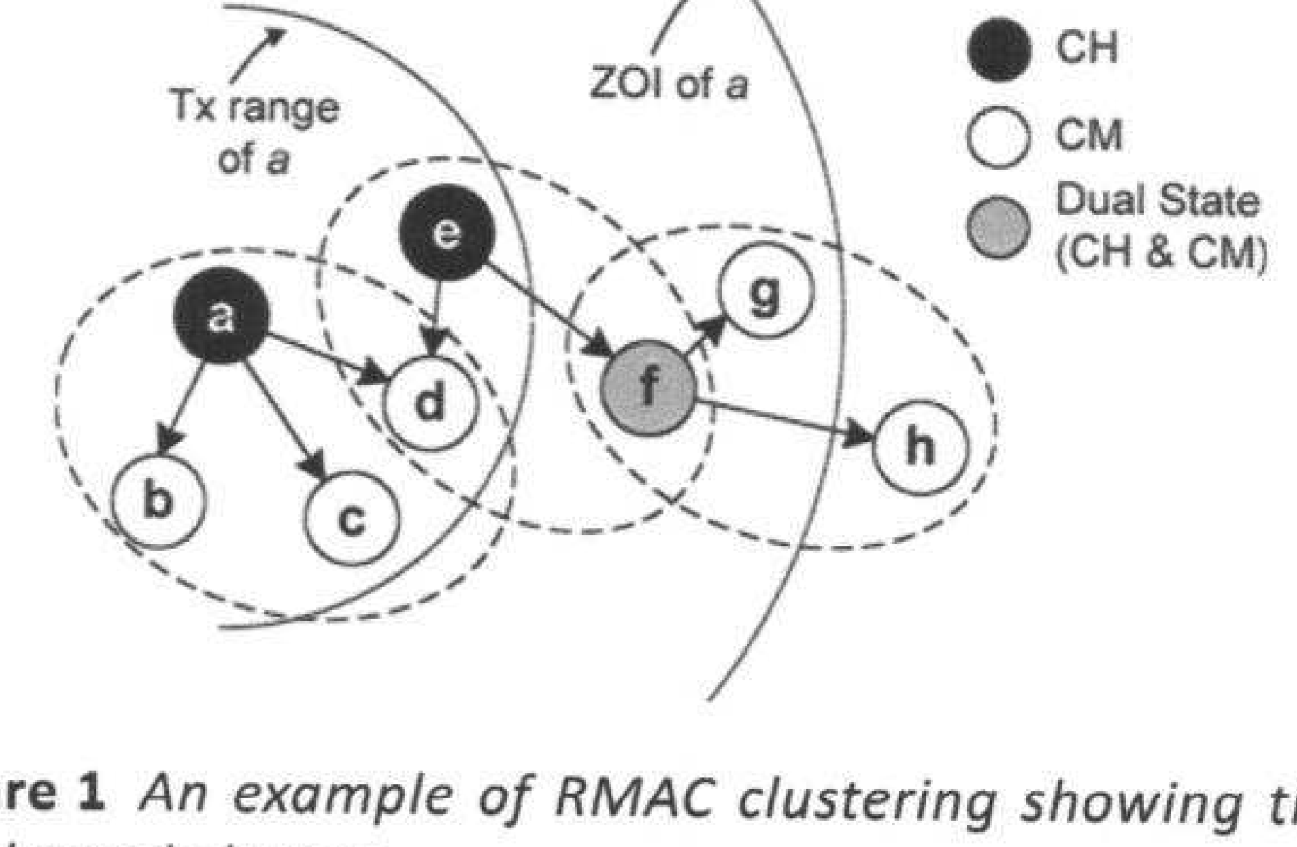


Figure 1 *An example of RMAC clustering showing three overlapped clusters*

© The Institution of Engineering and Technology 2009
IET Intell. Transp. Syst., 2009, Vol. 3, Iss. 2, pp. 148–158
doi: 10.1049/iet-its:20070052

between that node and each individual neighbour. In RMAC a node wishing to join a cluster, first identifies the optimal neighbour to be its CH by assigning precedence to each of its 1-hop neighbours independently of other nodes, and then selects the neighbour with the highest precedence to be its CH. Precedence is assigned with the NPA, which determines the optimal neighbour that would generally yield the longest cluster residence time, and is based on the relative speed, location, and direction of travel of each 1-hop neighbour as a generic approach to the mobility where all nodes can travel in all directions. Neighbouring nodes may view that choice of CH with a lower precedence and therefore may not select it as their CH. However, in the case of high-mobility VANETs, usually regarded as more challenging [16], it is generally accepted that nodes only cluster with neighbours travelling in the same direction to avoid clusters that would inevitably be extremely short-lived [12, 17]. While the concept of the NPA is introduced, the criteria within the NPA, used to identify optimal neighbours may be altered flexibly depending on the environment, or even to meet some other requirements and applications.

2.4 The zone of interest (ZOI)

To support geographic routing, each node maintains an up-to-date neighbour table containing all 1-hop neighbours. However, since vehicular nodes are highly dynamic, it is advantageous for nodes to maintain a neighbour table containing information of nodes beyond just their immediate transmission range. However, due to the nature of vehicular networks, the number of nodes maybe very large making it costly for each node to maintain a neighbour table with all the network nodes. Also, it is impossible for all the nodes disseminate information on all other nodes in the network, as these would have severe implications on scalability. In order to resolve these issues, the ZOI is defined. This is a circular area centred on a node and limited by some radius. Each node only needs to maintain neighbour table for nodes within its ZOI and discard information they receive about nodes outside their ZOI. Hence, nodes only manage a small segment of the network at any time. The radius of the ZOI is set to $2Tx_{\max}$, where $Tx_{\max}$ is the maximum transmission range of the node. Therefore effectively each node's neighbour table holds information on its 2-hop neighbours. As RMAC has been strategically designed to be technology independent, largely the ZOI depends on the technology capabilities such as range, bandwidth and so on. Hence, the ZOI radius has been designed as a configurable parameter that can be optimised for performance based on the deployed wireless technology.

By maintaining a ZOI that is greater than their own communication range, nodes have prior knowledge of neighbours as they travel into new neighbourhoods. In Fig. 1, node c will learn of neighbours within the ZOI of a via the polling process. Therefore when c goes out of range of a, becoming unclustered, it will have information about its new neighbourhood allowing it to quickly re-cluster, for example, with f. Also, in terms of geographic routing, as nodes move into new neighbourhoods, they would already have sufficient 1-hop neighbour information to enable packet forwarding, without having to wait for essential neighbourhood updates.

2.5 Neighbour table

A neighbour table holds records containing information of neighbouring nodes that lie within the ZOI. Each record consists of the Vehicle ID, Location, Speed, Direction, Cluster Size and Cluster Membership. Records are time-stamped allowing nodes to determine how up-to-date they are. The vehicle ID is a unique identifier assigned to every vehicle and should be adopted on a global scale, for example this could be the IEEE802.11 MAC address [16] and [13]. The cluster size is the number of CMs that belong to that neighbours cluster. This would be zero if the node was not a CH. Cluster membership is the number of clusters a node belongs to. In order to limit the cluster size and the number of clusters a node can be a member of, RMAC defines two parameters stating the maximum connection limits of a node, they are max-cluster-size, and max-cluster-membership, respectively. These parameters could be configured to offer optimised performance based on selected wireless technology limitations including range, bandwidth, and member nodes (e.g. Bluetooth, WLAN). The neighbour table is defined by set N, and the relationship between N, M, and H is described as $M \subset N$ and $H \subset N$ and also $M \cap H = \emptyset$.

The RMAC scheme consists of three main parts. They are cluster formation, cluster and ZOI maintenance, and cluster unification, each consisting of algorithms that are described in the next section.

3 RMAC algorithms

3.1 Cluster formation

A node that is not affiliated to any cluster is said to be in the unclustered state and it cannot inform its neighbours of its presence, nor can it receive information about neighbouring nodes. A node entering a new environment will initially start off in the unclustered state, and also a node enters the unclustered state if its sole link to another node fails. Cluster formation is the process where an unclustered node becomes clustered, and consists of three steps. These steps are described below and illustrated in Fig. 2*a*.

3.1.1 Step 1: Identifying 1-hop neighbours: Before a node attempts to join a cluster it must gather a list of its 1-hop neighbours, to determine the most suitable neighbour to cluster with. Let this list be associated with the set S. A vehicle that becomes unclustered may already have neighbour records in its table N. Hence, when a node enters the unclustered state, it first checks its neighbour

© The Institution of Engineering and Technology 2009

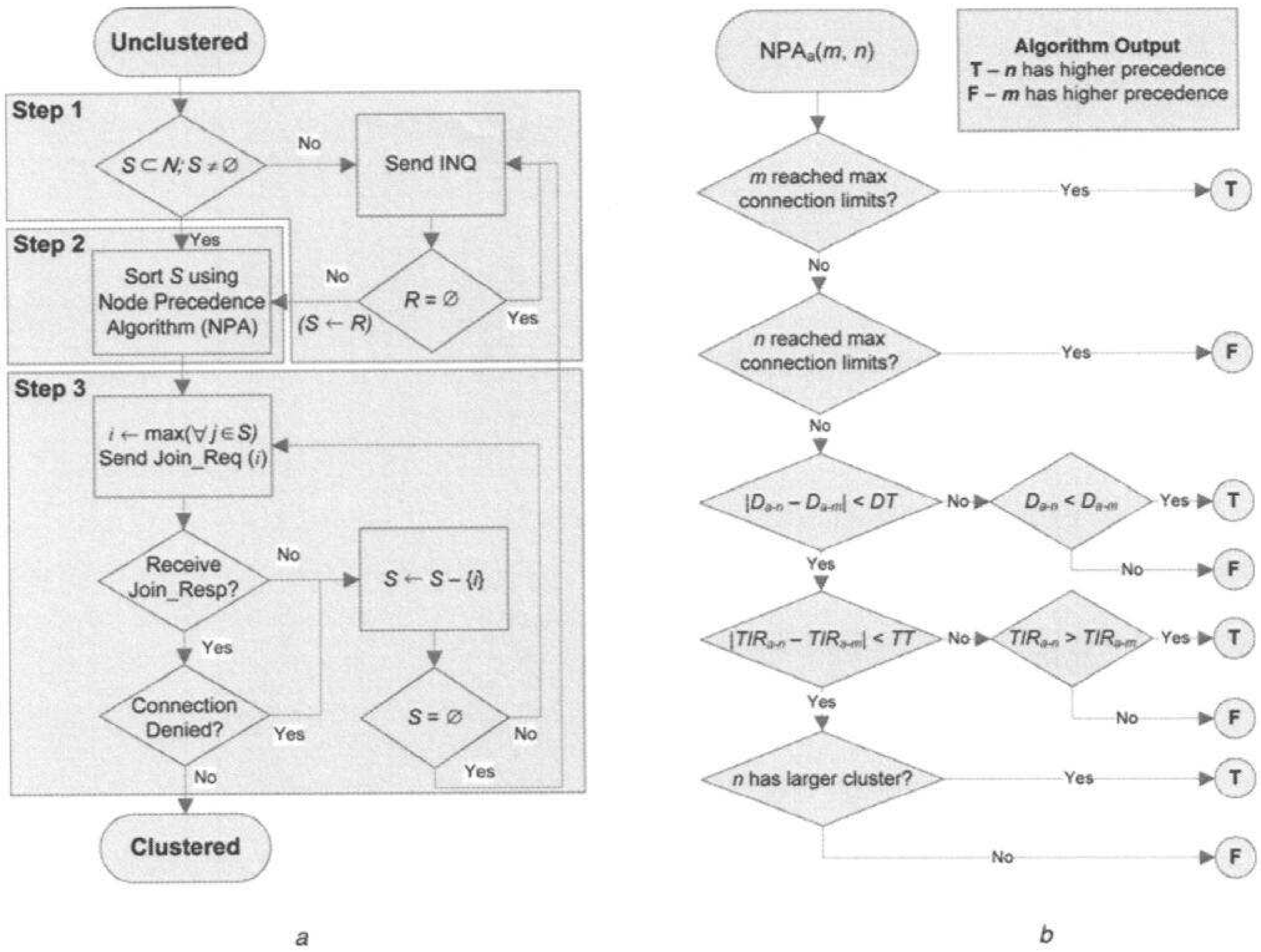


Figure 2 *Cluster formation*

a Cluster formation flow chart
b Node precedence algorithm for node *a* clustering

table to select a list of neighbours based on the following criteria, they are 1-hop neighbours, and their data has not expired (data expires when the time since the last update received exceeds a pre-defined threshold, and is defined to be 5 s). The selected neighbours are placed in the list S, hence $S \subset N$. However, if the neighbour table is empty, or no neighbours met the criteria, then the node enters the inquiry phase. In the inquiry phase the node attempts to discover its neighbours by broadcasting periodic inquiry (INQ) packets. It must be noted that a node rarely enters this phase. Generally a node that becomes unclustered will have sufficient information in its neighbour table to identify its 1-hop neighbours.

The INQ packet contains the vehicle ID and also the direction it is travelling in, which is used by the receiving vehicles to filter packets originating from vehicles travelling in the opposite direction as the available IVC time in a highly mobile VANET would be inadequate to transfer any meaningful data. Neighbouring nodes receiving the INQ packet check that they have not reached their maximum connection limits and respond with unicast inquiry response (INQ_RESP) packets, containing their information. This information includes their mobility parameters, that is, location, speed and direction of travel. On receiving the first INQ_RESP packet, the node initiates a timer and gathers all responses from its neighbours into a list R until the timer expires after a period known as the inquiry response wait period (irw_period). This allows all neighbouring nodes time to find a clear channel and respond. To put this into perspective, by simulating RMAC over IEEE802.11 wireless technology it was empirically found that an irw_period of 0.5 s offered optimal performance results. The inquiry process is repeated while $R = \emptyset$. Otherwise the responses are placed into list S, where $S = R$.

3.1.2 Step 2: Selecting optimal CH from 1-hop neighbours with NPA: After the unclustered node identifies its 1-hop neighbours (e.g. node a), such that $S_a \neq \emptyset$, the neighbours in S_a are sorted using the Bubble sort algorithm, according to the NPA (illustrated in Fig. 2*b* for node a clustering). The sorted list S_a will contain the neighbours in order of precedence, with the most suitable node to be CH at the top. The NPA uses mobility metrics of nodes to determine precedence based on relative mobility, they include location, speed, and direction.

Initially the NPA checks to see if either neighbour has reached their maximum connection limits. If node m has and node n has not, then n is given higher precedence, and vice versa. At the next level, node a calculates the distance between itself, and each of its neighbours, where the Euclidean distance D_{a-i} between itself and an arbitrary neighbour i is given by

$$D_{a-i} = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2} \tag{1}$$

where (x_a, y_a) and (x_i, y_i) are the location coordinates of nodes a and i, respectively. As shown in Fig. 2*a*, node a then

© The Institution of Engineering and Technology 2009
IET Intell. Transp. Syst., 2009, Vol. 3, Iss. 2, pp. 148–158
doi: 10.1049/iet-its:20070052

compares the distances of the two 1-hop neighbours (m and n) with distance threshold (DT) as follows

$$|D_{a-n} - D_{a-m}| < \text{DT} \tag{2}$$

where DT is a diminutive value that has been chosen to be 10 m, such that if the absolute difference in distances of the two neighbours is less than DT, they are considered to be approximately the same distance from a. If the distances are not similar, the closest node is given precedence. Otherwise, the NPA compares the approximate time-in-range (TIR) of each neighbour (m and n), to estimate which neighbour would stay in range longer. The TIR of any two nodes is based on the estimated locations of the two nodes after time t, where the estimated location (x'_i, y'_i) of the ith node after time t is given by

$$x'_i = s_i \times t \times \cos(\theta_i) + x_i \tag{3}$$

and

$$y'_i = s_i \times t \times \sin(\theta_i) + y_i \tag{4}$$

where s_i is the speed of the ith node, and θ_i is its direction of travel. The Euclidean distance between the estimated locations of node a and an arbitrary neighbour i (D'_{a-i}) is given by

$$D'_{a-i} = \sqrt{(x'_i - x'_a)^2 + (y'_i - y'_a)^2} \tag{5}$$

The TIR between nodes a and i (TIR_{a-i}) is calculated iteratively for t, using (3), (4) and (5) until $D'_{a-i} \geq Tx_{\max}$, where $Tx_{\max}$ is the maximum communication range. As shown in Fig. 2b, node a then compares the TIR of the two 1-hop neighbours (m and n) with time threshold (TT) as follows

$$|\text{TIR}_{a-n} - \text{TIR}_{a-m}| < \text{TT} \tag{6}$$

where TT is a diminutive value that has been chosen to be 5 s, such that if the absolute difference in TIR of the two neighbours is less than TT, they are considered to be in range of a for approximately the same amount of time. If the TIR of the two neighbours are not similar, the neighbour with the greater TIR is given precedence, otherwise the cluster sizes of the neighbours are compared, and the neighbour with the larger cluster is given higher precedence. Calculating the TIR may suggest that a neighbour close to the maximum transmission limit would be in range longer than a neighbour closer to node a, however a slight change in mobility may put that neighbour out of range quickly, hence, higher precedence is automatically given to the closer neighbour if the difference in distance between the two neighbours is greater than DT.

3.1.3 Step 3: Joining optimal CH: After sorting list S, the node sends a join request (JOIN_REQ) packet to the 1-hop neighbour at the top of S, requesting that a new link is formed (see Fig. 2a). The neighbour receiving the JOIN_REQ packet replies with join response (JOIN_RESP) packet, assigning the role of CM to the node that sent the JOIN_REQ, or denying the connection. If the neighbouring node is already a CH, the initiating node joins that cluster. However if the neighbouring node is a member of another cluster, or is in the unclustered state, a new cluster is formed with the neighbour acting as the CH. The neighbouring node receiving the JOIN_REQ packet will deny the connection attempt if it has reached its cluster size limits. If a connection is denied, or a JOIN_RESP packet is not received after a timeout period, that neighbour is removed from S, and the node attempts to join the neighbour that has the next highest precedence in the list S. This is repeated each time a join request fails until $S = \emptyset$, after which the vehicle enters the inquiry phase.

Cluster Formation Timings: The time taken to form clusters depends on a number of factors. Let T_{INQ}, $T_{\text{JOIN\_REQ}}$ and $T_{\text{JOIN\_RESP}}$ be the time taken to process and transmit the corresponding packets. When a node enters the unclustered state, if it does not have sufficient information in its neighbour table, then the time taken to conduct neighbour discovery ($T_{\text{N\_DISC}}$) to find its 1-hop neighbours via the inquiry procedure is given by

$$T_{\text{N\_DISC}} = T_{\text{INQ}} + \text{irw\_period} \tag{7}$$

Therefore the total time taken to form clusters (T_{CLUS}) is given by

$$T_{\text{CLUS}} = T_{\text{N\_DISC}} + T_{\text{JOIN\_REQ}} + T_{\text{JOIN\_RESP}} \tag{8}$$

However, when entering the unclustered state, if a node has sufficient information in its neighbour table such that $S \neq \emptyset$, then effectively $T_{\text{N\_DISC}} = 0$. Generally when an inquiry procedure is carried out, it adds a significant amount of time when forming clusters, as the irw_period is much longer than packet transmission times. By utilising information nodes have about their neighbourhood at the time they became disconnected, nodes can quickly re-cluster to join the network and avoid the costly neighbour discovery procedure.

3.2 Cluster and ZOI maintenance with polling

CHs maintain their clusters by periodically polling their CMs, where a POLL packet is sent every poll_interval to each CM. CMs respond with a poll acknowledgement (POLL_ACK) packet. For example in Fig. 3, CH a polls its members b and

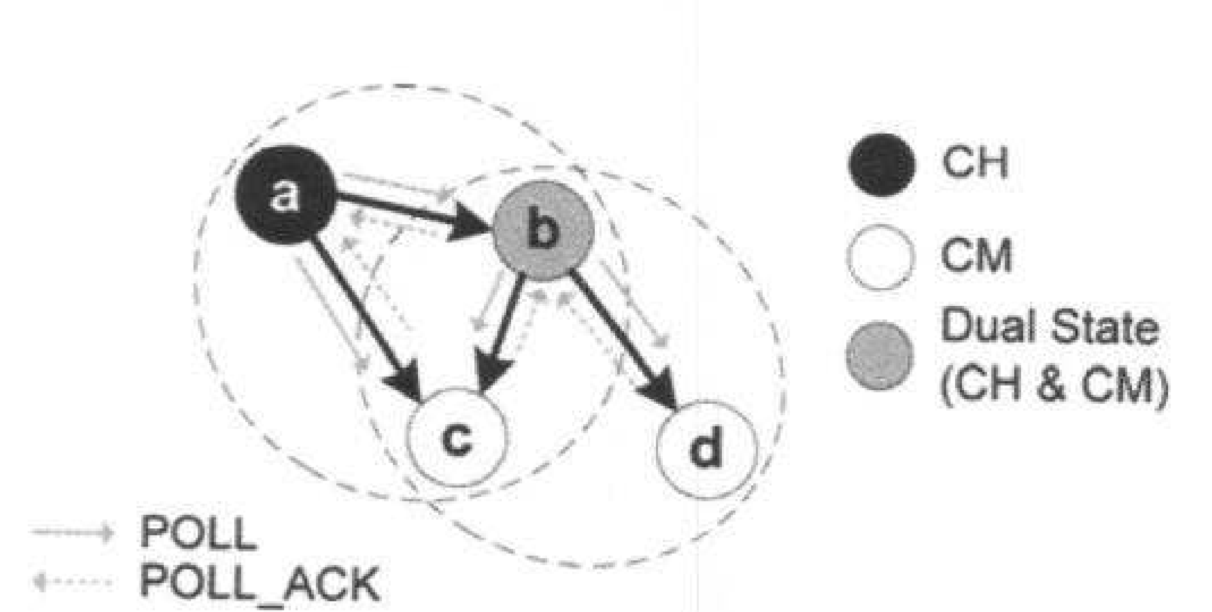


Figure 3 *Polling scheme*

© The Institution of Engineering and Technology 2009

c and awaits a POLL_ACK packet from each CM. CMs that do not respond to a POLL packet for some time are timed out and removed from the cluster, that is from M_a. A CH determines if a CM has timed out by calculating the time since the last received POLL_ACK, if this time is greater than the poll_threshold, it is assumed that the CM is no longer in range. Similarly members also check validity of links to their CHs every poll_interval. CHs that have failed to POLL for a period of time are removed from H. On events where a CM or a CH is removed, if $H = \emptyset$ and $M = \emptyset$ the node enters the unclustered state.

The polling scheme is also used to maintain an up-to-date neighbour table containing neighbours within the ZOI by including neighbour information in the POLL and POLL_ACK packets. Each record held in the neighbour table is associated with a data provider, this being the node that provided the latest update of that node's information. A CH polling a member includes its neighbour records in the POLL packet with the exception of records whose data provider is the recipient of the packet. This avoids the same information looping back and forth between two nodes. Likewise, a member replying with a POLL_ACK packet sends its neighbour records excluding records that were provided by that CH.

Fig. 3 illustrates how neighbour location updates occur through the polling system. In Fig. 3, node d receives details of a via b, after two poll cycles. In this manner all nodes will soon learn of all neighbouring nodes. However, node a receives information of c from b and also directly from c. Inevitably information arriving from b would not be as up-to-date as that received directly. Hence, nodes receiving information compare the timestamps of the received records against the timestamps of the records in the neighbour table, and if information received is older than the existing information it is discarded, else the neighbour table is updated accordingly. The maximum time taken for information to be propagated between nodes is given by $((n-1) \times \text{poll\_interval})$, where n is the number of hops the between the respective nodes, that is, 2 between d and a in Fig. 3.

3.3 Cluster unification

Owing to mobility, in time independent clusters could come within range creating disjoint clusters (see Fig. 4a). This in turn implies that clustering initialisation process including new neighbour tables and possible IP addressing would have to be reassigned [13], which imposes unnecessarily high delays at the overall system. With RMAC, all neighbouring clusters are overlapped and hence, transparent transition (e.g. because of cluster deletion or merging) of a member from one cluster to another is achieved with no unnecessary overheads. Hence, RMAC importantly identifies disjoint clusters and unifies them in three steps.

3.3.1 Step 1: Cluster presence advertisements:

In order to unify disjoint clusters, such clusters must become aware of each other's presence. Hence, clusters periodically broadcast their presence with a cluster presence (CLUS_PRES) packet, which is sent with a minimum interval of cp_interval, where cp_interval $= 4 \times$ poll_interval. CHs identify the two edge nodes of the cluster, and instruct them via a special field in the POLL packet to broadcast the presence of that cluster. Edge nodes are nodes in a single cluster whose distance from each other is the greatest. If the CH deems itself to be one of the edge nodes, it too broadcasts the cluster presence. CLUS_PRES packets contain the sender's information, including a list holding the addresses of nodes in its neighbour table.

3.3.2 Step 2: Identifying presence of new clusters:

Assuming node a receives a CLUS_PRES packet from an arbitrary neighbour i, a determines if i is a node within a disjoint cluster based on the decision process illustrated in Fig. 4b. It first checks if i is contained in its neighbour table, where $i \in N_a$. If it is, it implies that either i belongs to the same cluster as a, or that their clusters are connected via overlapped clusters, such that a learned of i via

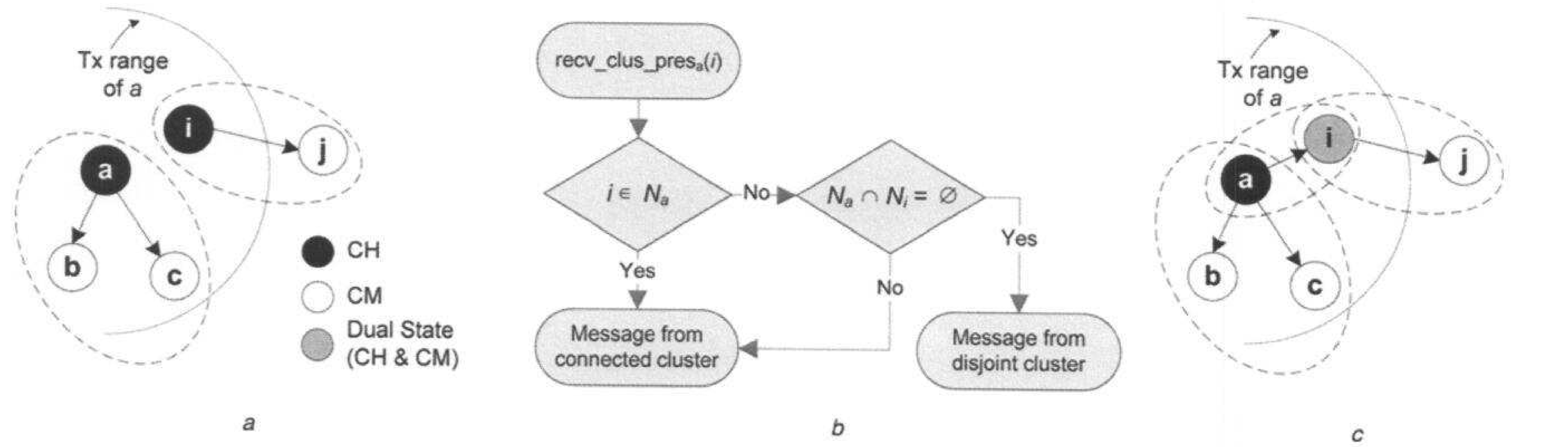


Figure 4 *Disjoint clusters*

a Example of disjoint cluster
b Identifying disjoint clusters
c Resulting unified clusters

© The Institution of Engineering and Technology 2009

doi: 10.1049/iet-its:20070052

intermediate nodes. If i is not contained in the neighbour table, a compares the neighbour list of i against its own neighbours. If there are common nodes, it is assumed their clusters are connected, but neighbour tables have not yet propagated fully, where $i \in N_a$. However, if there are no common nodes, that is, $N_a \cap N_i = \varnothing$, i is deemed to a node belonging to a disjoint cluster Fig. 4*a*.

3.3.3 Step 3: Unifying disjoint clusters: It is possible that many nodes from a set of overlapping clusters will receive the same broadcast CLUS_PRES packet from a disjoint cluster. Therefore, each node receiving the CLUS_PRES packet determines the closest node in its cluster structure to the disjoint cluster based on information in their neighbour tables, such that the closest node will perform cluster unification. In Fig. 4*a*, for example, if a CLUS_PRES sent by i is received by both a and c, both nodes would identify a is the closest, based on knowledge of each others' locations. Node a then sends a cluster unify request (CLUS_UNIFY_REQ) packet to node i, who replies with a cluster unify response (CLUS_UNIFY_RESP) packet, containing the *role* of the node a and also a copy of the local neighbour table, thus unifying the clusters. The node with the larger cluster becomes CH. In case their sizes are equal, the node that initiated the connection with the CLUS_UNIFY_REQ packet is assigned the role of CM.

CLUS_PRES and CLUS_UNIFY_REQ packets contain a list of the transmitting node's neighbours. Nodes responding to these packets with CLUS_UNIFY_REQ and CLUS_UNIFY_RESP packets, respectively, first check the neighbour list against the lists of previous recipients of these packets ensuring there are no common nodes between them. If there are, it implies that a node is attempting to unify simultaneously with multiple nodes from the same disjoint cluster, in which case transmission of the packet is aborted.

4 Simulation results and analysis

RMAC is simulated within the NS-2 [18] simulator environment, which supports the use of mobile nodes and the IEEE802.11 wireless standard operating in *ad hoc* mode. The wireless interface has a communication range of approximately 250 m, and a bandwidth of 2 Mbits/s. Node mobility is controlled by a scenario file modelled on a 2 km long four-lane motorway, where all vehicles move freely at varying speeds, travelling in the same direction. It was generated using the IMPORTANT Freeway mobility generator developed by USC [16].

For simulation, the range of velocity of the vehicles was set between 22 and 36 m/s (approximately 79–130 km/h). These values represent a speed range around the maximum UK national speed limit of 112.7 km/h. In order to test the scheme under varying conditions 12 scenario files were created with a combination of two parameters, the node density and maximum acceleration. Increasing the maximum acceleration of the nodes makes their mobility more unpredictable, as neighbouring nodes can quickly go out of transmission range, thus testing the capability of the algorithm to adapt its clusters in a rapidly changing environment. For each scenario of 25, 50 and 75 nodes, four maximum acceleration speeds were specified, 1.8, 3.6, 5.4 and 7.2 m/s^2, corresponding to 5, 10, 15 and 20% of the maximum velocity. Each scenario with the above parameters was simulated five times to avoid run time simulation errors and the average was taken. The simulations were each run for a period of 1000 s. Additionally, the DMAC algorithm was also simulated in NS-2 with the same mobility models and results obtained for comparison with RMAC. DMAC reacts to variations in the surrounding topology, caused by new links and link failures, thus re-assigning weighted CHs as and when required, making this algorithm suitable for mobile environments and geographic routing. DMAC assumes that all neighbours receive broadcast control packets, in order for proper operation, which is not entirely feasible as packet collisions are inevitable, especially in larger asynchronous wireless communication networks. Each cluster node maintains a neighbour table, containing its neighbours' id, and their weight. This information is gathered from periodically received beacon messages. In DMAC, the CHs form a dominant independent set. On start-up, a node executes an initialisation procedure. All other procedures described by DMAC are message driven, and are generally invoked when changes in the network are detected (such as link failure and new links) or in response to receiving a packet. For further details on DMAC refer to [1].

4.1 Cluster stability and robustness

Cluster stability can be determined by analysing the cluster residence times, which is the average time each node spends as a member of a cluster. Higher cluster residence times indicate fewer cluster changes, resulting in more stable clusters. Fig. 5 illustrates the average cluster residence times for the increasing maximum acceleration

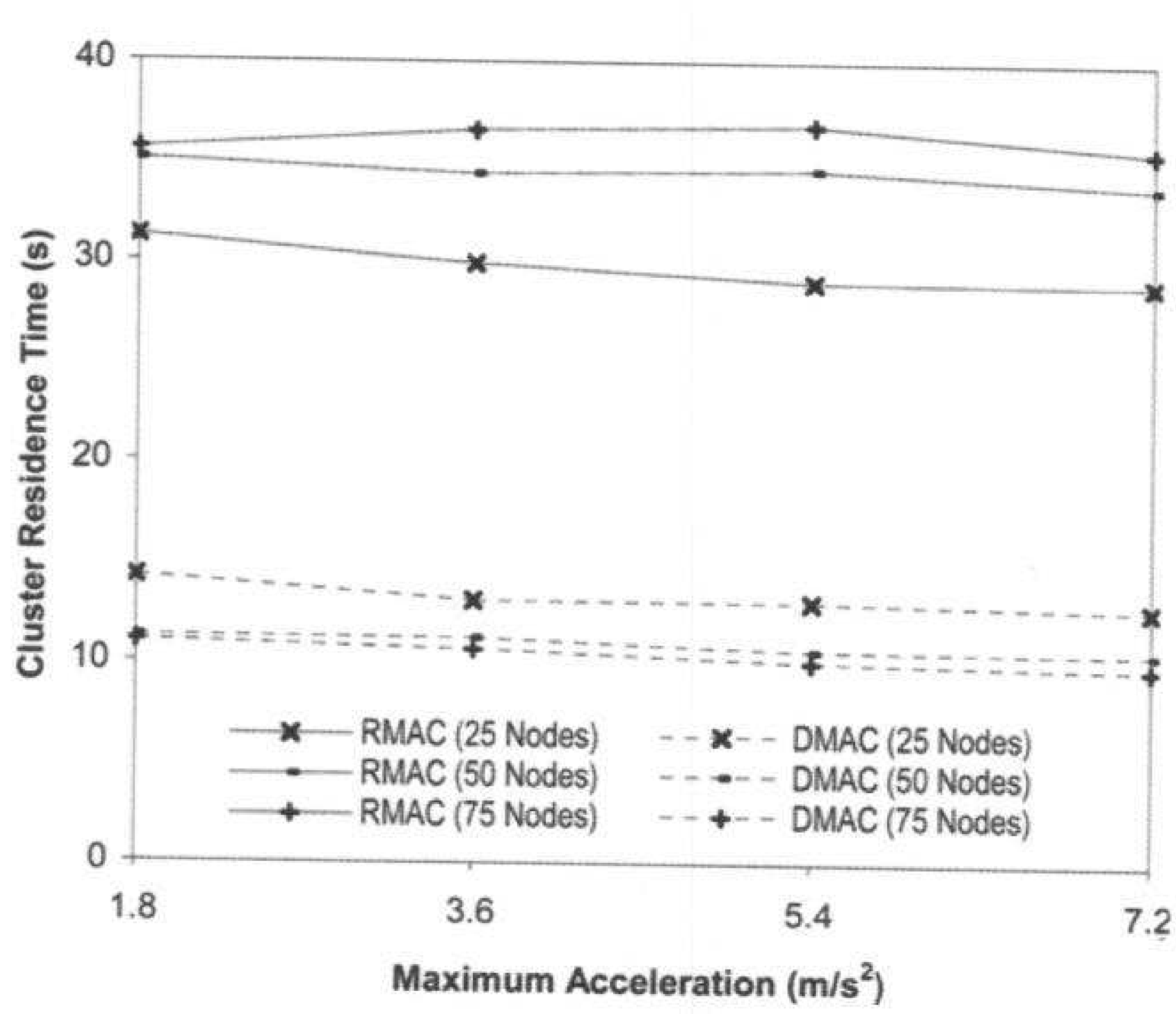


Figure 5 *Comparison of cluster residence times*

© The Institution of Engineering and Technology 2009

speeds. It can be seen that RMAC achieves high cluster residence times, as compared to DMAC, under the varying conditions. RMAC improves slightly with increased node density, whereas DMAC's cluster residence times decrease. In RMAC, when density increases nodes have a larger set of neighbours to select from when joining a cluster, therefore enabling them to select the most optimal neighbour that would yield the longest cluster residence time. However, with DMAC, the existence of a larger number of highly mobile nodes would increase the number re-election events as nodes move in and out range of each other, causing them to re-affiliate with new clusters.

A characteristic of RMAC is the connection oriented nature of its clusters, where nodes can update and receive updates from neighbours only when connected to a cluster, unlike broadcast beacon based clustering. Therefore, when nodes become unclustered, they attempt to quickly re-cluster with a new CH by utilising information in the neighbour tables. Fig. 6 shows the average percentage of nodes that re-cluster within a given time for the different node densities. It can be seen that a high proportion of nodes re-cluster within 0.2 s, and is because of their utilising the information in their neighbour tables to quickly identify 1-hop neighbours and re-cluster. Whereas a small percentage of nodes take in excess of 0.5 s to re-cluster, and is due to them having insufficient information in their neighbour tables, thus resorting to the inquiry process to discover their 1-hop neighbours. At lower densities the re-clustering times are much faster. This is due to the fact that the channel is less congested therefore nodes have to wait for a shorter period of time to send their JOIN_REQ packets. However, with higher densities more nodes can reconnect without having to resort to the inquiry process, this is due to the greater availability of neighbouring nodes.

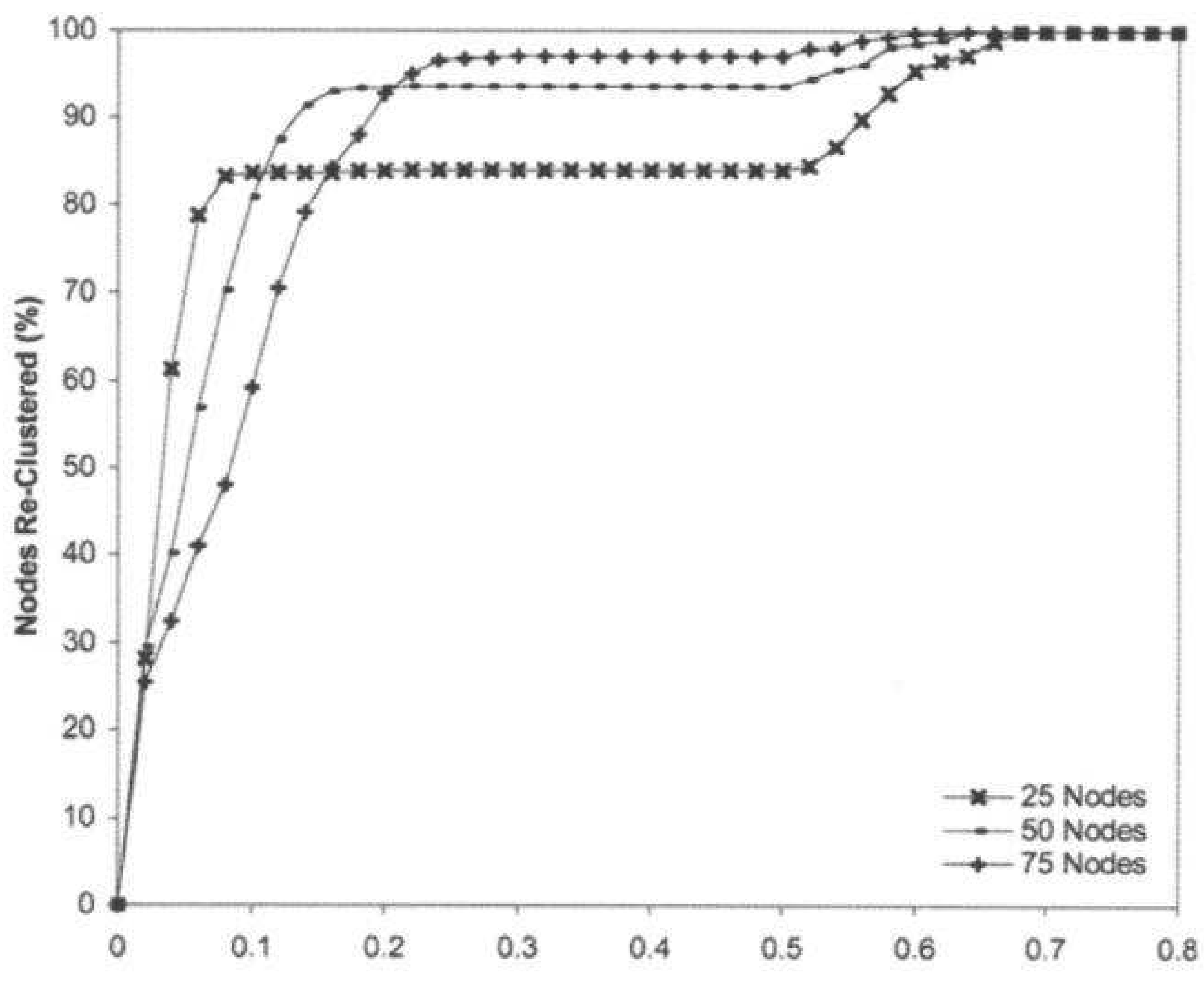


Figure 6 *Node re-clustering time after link failure*

4.2 Effectiveness of neighbour update mechanism

The effectiveness of this scheme can be quantified by analysing the data held in the neighbour tables at each node. Ideally the neighbour tables should contain 100% of a node's 1-hop neighbours. Owing to the constant mobility of nodes, it is not possible to achieve the idealistic state. This is also true for conventional algorithms that use beaconing, nodes will not be aware of 100% of its neighbours at any instant of time. This may be caused when the topology changes and beacon packets are yet to be transmitted, or due to collision of beacon packets, which inherently are not retransmitted.

In Fig. 7 the average percentage of 1-hop neighbours that exist in the neighbour tables are presented for varying node densities, by sampling neighbour tables at 10-s intervals. Varying the acceleration speeds did not have significant effects on the results. It can be seen that across the varying node densities the neighbour tables maintain a large percentage of neighbours within communication range. This is a crucial requirement of geographic routing protocols, to enable selection of a suitable next hop node. With the exception of the initial settling period, the graphs remain consistent throughout the simulation. It can be seen that as node density increases the percentage slightly decreases. This is due to the greater rate of change in topology caused by the existence of a larger number of highly mobile nodes. Considering that these values represent a percentage, this effect is not detrimental. Even though an increase in density causes a slight decrease in the percentage of nodes held in the neighbour table, a sufficiently large number of 1-hop neighbours are available to be utilised by a geographic routing algorithm.

When determining the current location of a 1-hop neighbour, nodes utilise information held in the neighbour

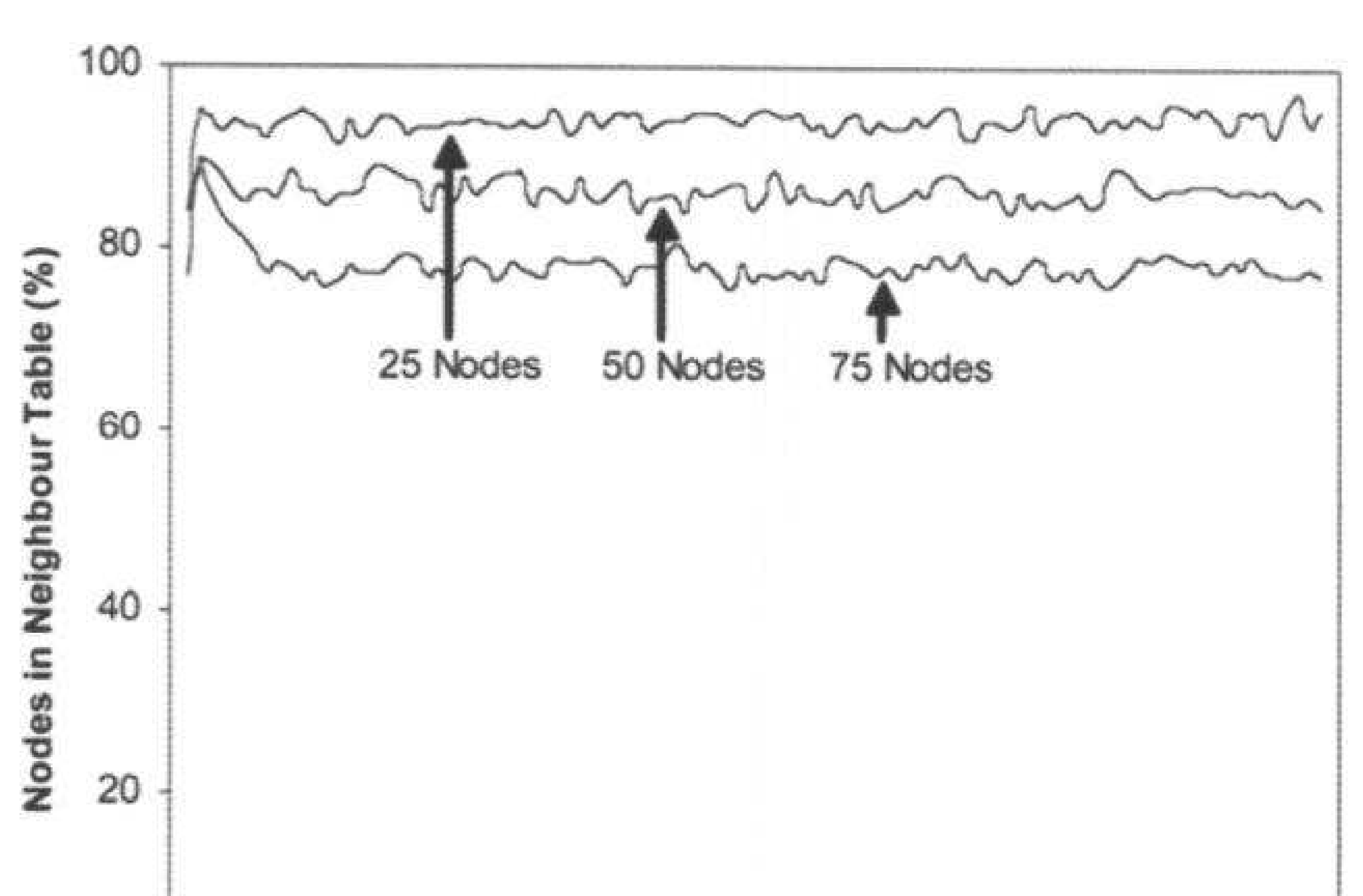


Figure 7 *Effectiveness of neighbour table for 1-hop neighbours*

© The Institution of Engineering and Technology 2009
IET Intell. Transp. Syst., 2009, Vol. 3, Iss. 2, pp. 148–158
doi: 10.1049/iet-its:20070052

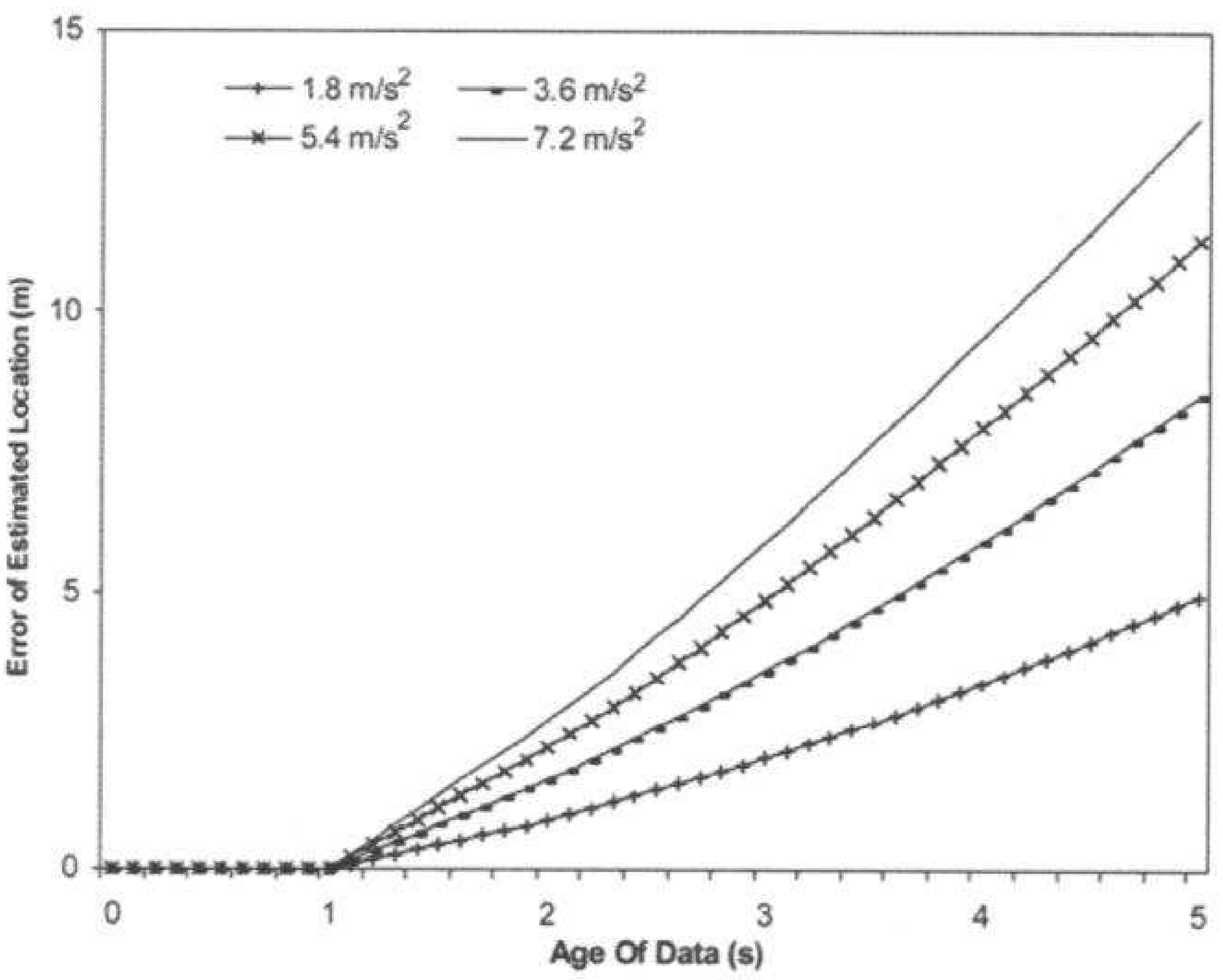


Figure 8 *Average error between real and estimated locations based on age of data*

tables to estimate its current locations (x', y') using (3) and (4), where t corresponds to the data age, which is determined based on the timestamp of the information, and generally corresponds to the number of hops between the source of the information and the holder of the information (e.g. in Fig. 3, information of d will arrive at a after two poll cycles). Fig. 8 shows the error between the estimated and actual locations of 1-hop neighbours, based on their data age. Again, node density did not have a significant effect on the location errors. However, as expected, in simulations with a greater maximum acceleration speed, the errors in the estimated location also increase. It can also be seen that positions of nodes with data less than 1 s old can be estimated accurately, with near zero error. However, as the data ages, the error increases. Nevertheless, data up to 5 s old can be confidently utilised by a routing agent, as errors are below 15 m, which is approximately less than 6% of the transmission range of IEEE802.11, assuming a 250 m range. The errors appear to be consistent and hence, could be modelled by routing algorithms in order to compensate for it, thereby selecting a next hop node whose distance plus the modelled error is less than the transmission range.

5 Conclusions

A new clustering scheme named RMAC is presented for high-speed VANETs networks. The cluster formation, maintenance and unification algorithms are also given. At the heart of the scheme is the NPA, which uses relative node mobility metrics of speed, location and direction of travel for the adaptive selection of optimal CHs from the 1-hop neighbours. By defining the ZOI within the scheme, individual nodes are only required to manage a small segment of the network leading to scalability to large networks. The ZOI also maintains a reliable neighbour table containing nodes beyond its own communication range resulting in a high mobility adaptive scheme. It is shown by simulation that cluster stability is achieved with very high residence times, approximately 35 s for high-density networks (>50 nodes) and in a highly dynamic environment. This is a much better performance than DMAC that offers only about 15 s. A very high percentage of recovery after link failure has been demonstrated with approximately 80% re-clustering within 0.2 s and 100% within 0.7 s even with low number of nodes. It is also shown to maintain high percentage of 1-hop neighbours and very accurate location information essential for geographic routing protocols. In effect, the proposed scheme based on the above, offers a unique capability in enabling and managing a highly dynamic road transport IVC very efficiently. It is important in supporting real time vehicular safety applications essential in ITS such as collision avoidance, safe distance keeping, and traffic control that overall can improve road safety and driving experience [16]. Further work includes the integration of the proposed clustering scheme with geographical routing protocols and to investigate optimisation of the scheme by employing pre-emptive clustering to predict imminent link failures and take proactive measure by forming a secondary link with a neighbouring node, leading to further cluster stability.

6 References

[1] BASAGNI S.: 'Distributed clustering for ad hoc networks'. Proc. of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99), 1999, pp. 310

[2] FÜBLER H., MAUVE M., HARTENSTEIN H., VOLLMER D., KÄSEMANN M.: 'A comparison of routing strategies in vehicular ad hoc networks'. Department of Computer Science, University of Mannheim, Tech. Rep. TR-02-003, Mar. 2002

[3] MAUVE M., WIDMER J., HARTENSTEIN H.: 'A survey on position based routing in mobile ad hoc networks', *IEEE Netw. Mag.*, 2001, **15**, (6), pp. 30–39

[4] LI J., JANNOTI J., DECOUTO D., KARGER D., MORRIS R.: 'A scalable location service for geographic ad hoc routing'. Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), August 2000, pp. 120–130

[5] BABURAM A., ALI F.H., STIPIDIS E.: 'Dynamic clustering for high speed mobile ad hoc networks', *Spec. Issue Mobile Commun. Theory Technol. Appl. Mediterranean J. Electron. Commun.*, 2006, **2**, (2), pp. 72–81

[6] YU J.Y., CHONG P.H.J.: 'A survey of clustering schemes for mobile ad hoc networks', *IEEE Commun. Surv.*, First Quarter, 2005, **7**, (1), pp. 32–48

© The Institution of Engineering and Technology 2009

[7] YU J.Y., CHONG P.H.J.: 'An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs)', *Comput. Commun.*, 2006, **30**, (1), pp. 5–16

[8] XU K.X., HONG X.Y., GERLA M.: 'An ad hoc network with mobile backbones'. Proceedings IEEE ICC2002, August 2002, vol. 5, pp. 3138–3134

[9] OHTA T., INOUE S., KAKUDA Y.: 'An adaptive multihop clustering scheme for highly mobile ad hoc networks'. Proc. The Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03), April 2003

[10] EPHREMIDES A., WIESELTHIER J.E., BAKER D.J.: 'A design concept for reliable mobile radio networks with frequency hopping signalling', *Proc. IEEE*, 1987, **75**, (1), pp. 56–73

[11] GERLA M., TSAI J.T.-C.: 'Multicluster, mobile, multimedia radio network', *ACM/Baltzer J. Wirel. Netw.*, 1995, **1**, (3), pp. 255–265

[12] FAZIO M., PALAZZI C.E., DAS S., GERLA M.: 'Vehicular address configuration'. AutoNet Proceedings, GLOBECOM 2006

[13] ARNOLD T., LLOYD W., ZHAO J., GAO G.: 'IP address passing for VANETs'. 6th IEEE Conference on Pervasive Computing and Communications, PerCom 2008, pp. 70–79

[14] KARP B., KUNG H.T.: 'GPSR: Greedy Perimeter Stateless Routing for wireless networks'. Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom '00), August 2000, pp. 243–254

[15] BIANCHI G.: 'Performance analysis of the IEEE 802.11 distributed coordination function', *IEEE J. Sel. Areas Commun. (JSAC)*, 2000, pp. 535–547

[16] HARTENSTEIN H., LABERTEAUX K.P.: 'A tutorial survey on vehicular ad hoc networks', *IEEE Commun. Mag.*, 2008, **46**, (6), pp. 164–171

[17] IMPORTANT: An evaluation framework to study the 'Impact of mobility patterns on RouTing in ad hoc NetWorks', available at http://nile.usc.edu/important/, last accessed 11 May 2007

[18] NS-2 Network simulator, available at http://www.isi.edu/nsnam/ns/index.html, last accessed 11 May 2007

© The Institution of Engineering and Technology 2009

doi: 10.1049/iet-its:20070052