Journal of Advances in Science and Technology
Vol. 12, Issue No. 24, November-2016, ISSN 2230-9659

Pattern of Dense Manet (PDM) in Replication Strategy

Navneet Garg[1]* Dr. Vijay Pal Singh[2]

[1]Research Scholar, OPJS University, Churu, Rajasthan

[2]Associate Professor, OPJS University, Churu, Rajasthan

--------------------------♦-----------------------------

There are only few proposals which face the challenging issue of resource replication in mobile environments with the goal of increasing accessibility and effectiveness [53]. In particular, a middleware solution is proposed here, called Replication Strategy in Dense MANETs), that transparently disseminates, retrieves, and manages replicas of common interest resources among cooperating nodes in dense MANETs [54]. Replication Strategy has the main goal of improving the availability of data and service components, and maintaining a desired replication degree for needed resources, independently of possible (and unpredictable) exits of replica-hosting nodes from the dense MANET. In this chapter presents how Replication Strategy addresses the primary challenging issues of dense MANETs, i.e., the determination of nodes belonging to the dense region, the sensing of nodes entering/exiting the dense MANET, and the dynamic selection of replica Heads responsible for orchestrating resource replication and maintenance. Here Replication Strategy protocols for dense MANET recognition and Head selection explained, extensive simulation results about the Replication Strategy prototype, whose performance shows the effectiveness and the limited overhead of the proposed solution, by confirming the suitability of the application-level middleware approach even in conditions of high node mobility, notes about potential security issues, related work, and conclusions end the article.

5.1 RECOGNITION OF DENSE MANET

Replication Strategy design a simple protocol where any node autonomously determines whether it belongs to the dense MANET or not. One node is in the dense MANET DM (n) only if the number of its neighbors, i.e., the nodes at single-hop distance, is greater than *n*. Each node autonomously discovers the number of its neighbors by exploiting simple single-hop broadcast discovery messages. By delving into finer details, at any time one Replication Strategy node can start the process of dense recognition/update; in the following, the *starter* node will be called. The starter starts the original Replication Strategy protocol for dense MANET recognition by broadcasting a discovery message that includes the number of neighbors (*NoN*) required to belong to the dense region and the identity of the sender. That number can be autonomously decided depending on the desired degree of connectivity redundancy: typically, a value between 10 and 20 ensures a sufficiently large set of alternative connectivity links. When receiving the discovery message, each node willing to participate replies by forwarding the message to its single-hop neighbors, if it has not already sent that message, and by updating a local list with IP addresses of detected neighbors.

Let us consider the network example in Figure 5.1; the cyclic diagram of the dense MANET recognition protocol triggered by node A is reported in Figure 5.2. The first discovery message (including the number of neighbors required to belong to the dense MANET - 2) reaches nodes I, F, E, and NP1, as illustrated in Figure 5.2 step (a). These nodes, after a random interval, re-broadcast the message unchanged (first NP1, then I, F, and E – step (b) in the figure) and set a timeout to determine whether they belong to the dense region. Since A, I, F, and E receive 4, 6, 3, and 4 messages (step (c) in the figure) before timeout expiration, they realize they either be-long or not to the dense MANET, as shown in step (d) in the cyclical diagram.

Since dense MANET nodes can move after the recognition process, the proposed algorithm includes a lightweight lazy-consistent maintenance phase. Nodes periodically exchange bye-bye packets; each node receiving a bye-bye message records its source in a table entry, with an associated timeout; next bye-bye received from the same source restart a new timeout. Dense MANET nodes periodically check whether their table entries are still valid; if an entry has expired, the node removes it from the table, and verifies whether the condition for dense MANET belonging still holds.

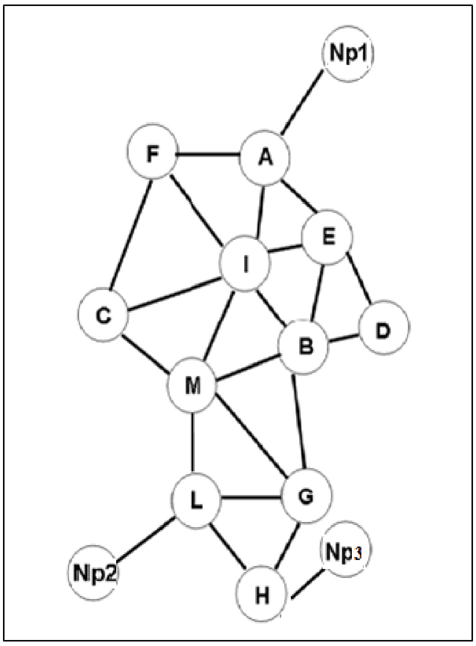

Figure 5.1: Situation of node and there connectivity.

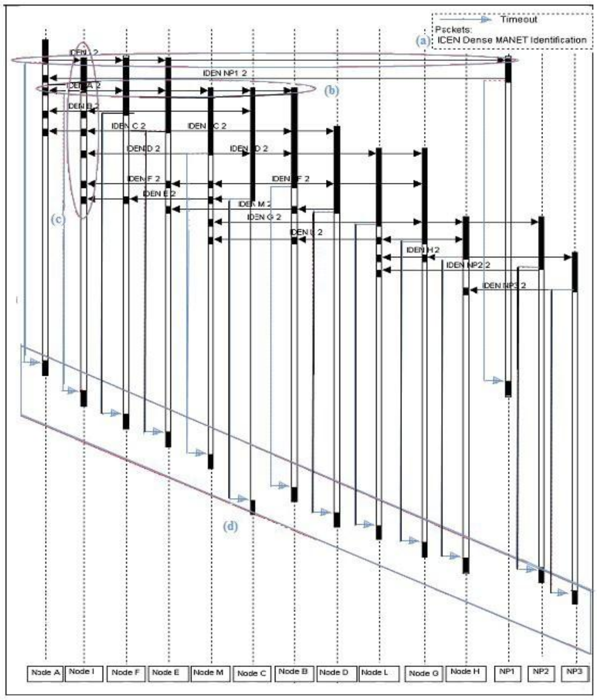

Figure 5.2: Cyclical diagram of the dense MANET recognition protocol.

The current PDM implementation assumes that each participant node adopts a zero PATTERN solution, such as [96, 91]; guaranteeing IP address uniqueness is out of the scope of our research. After a specified time interval, any node autonomously checks whether its list contains more than n nodes, and autonomously decides whether it belongs or not to the dense MANET. Let us observe that discovery broadcasts could provoke packet collisions; the problem is well-known in the literature and usually identified as the "broadcast storm" issue [192]. In order to avoid the problem, any Replication Strategy node defers node broadcasts of a random and limited time interval. Techniques for collision and overhead reduction such as the ones presented in [92, 78] do not fit dense MANET recognition, since local density evaluation is based on the assumption that each node receives messages from all its neighbors. The dense MANET recognition protocol has demonstrated to scale well also in large deployment scenarios, with a completion time linearly dependent on the dense MANET diameter, i.e., on the maximum hop distance among any couple of nodes belonging to the dense region, and an overall number of exchanged messages equal to the number of dense MANET participants. Given that dense MANET nodes can move during and after the recognition process, the proposed algorithm achieves an approximated solution and requires including a lightweight lazily-consistent maintenance phase. Nodes periodically exchange bye-bye packets with their neighbors; each node receiving a bye-bye message records its source in a table entry, with an associated timeout; next bye-bye packets received from the same source restart a new timeout. Dense MANET nodes periodically check whether their table entries are still valid; if an entry has expired, the node removes it from the table, and verifies whether the condition for dense MANET belonging still holds. Let us rapidly observe that dense MANETs are virtual and dynamic group organizations of MANET nodes. Therefore, there is also the possibility to have more than one coexisting dense MANET, each one with a different NoN, involving the same MANET peers, e.g., in the case that several starters trigger the dense MANET recognition process with different NoN values. However, since the addressed deployment scenarios generally exhibit high density gradients close to boundaries, several different NoN values usually determine the same DM (NoN) sets. Given the limited applicability and the additional overhead of maintaining more than one dense MANET over the same set of physical nodes, the current Replication Strategy implementation does not support multiple overlapping dense MANETs with different NoN: the NoN value for a dense area cannot be changed for a long time interval after the first recognition procedure.

5. 2 HEAD SELECTION OF REPLICA

The Replication Strategy middleware works to assign the Head role to a node located in a topologically central position. The protocol explores a Head candidate as a subset of nodes in the dense MANET, called Searched Nodes (ENs). Figure 5.3 shows a practical example of application of that guideline. The first step of the protocol considers node H: its farthest node is I, located at 4-hop distance; so, H is tagged with the value of that distance (H4 in the figure). Then, the Replication Strategy Head selection protocol considers A, because it is the first node along the path from H to I: A's farthest node is I, at 3-hop distance (A3). At the next iteration, the protocol explores node D, which is chosen as replica Head by respecting the termination criteria described in the following of the section. Node D can reach any other node in the depicted dense MANET with a maximum path of two hops. Let us observe that Replication Strategy provides a simple way to react also to Head exits from the dense MANET. If the Head realizes it is going to exit, e.g., because its battery power is lower than a specified threshold, it delegates its role to the first neighbor node found with suitable battery charge and local memory. In the case the Head abruptly fails, any resource delegate that senses the Head unavailability can trigger a new selection procedure.

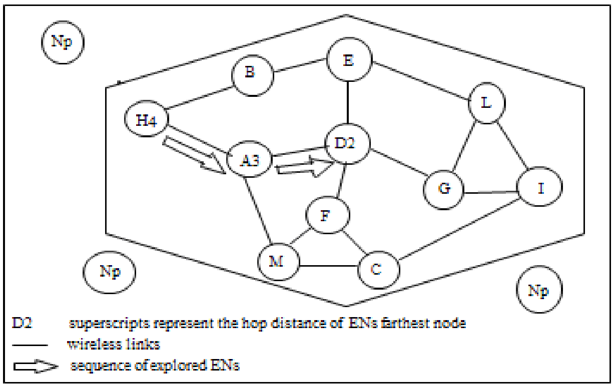

Figure 5.3. Replication Strategy exploring the sequence of ENs from H to A and A to D.

After having informally introduced the main guidelines of the protocol, let us know precisely specify how the Head selection works. Each EN executes three operations: i) it determines the number of hops of the shortest paths connecting it to any farthest node in the dense MANET (the maximum of those hop numbers is called *EN_value*); ii) it identifies its neighbors located in the direction of its farthest nodes (*forwarding_neighbors*); and iii) it autonomously chooses the next EN among all the nonsearched forwarding neighbors of already searched ENs with lowest associated values. To take possible device heterogeneity into account, Replication Strategy promotes the exploration only of ENs suitable to play the role of replica Head once selected. For instance, if a potential EN device has insufficient memory and too low battery life (if compared with configurable Replication Strategy thresholds), it is excluded from the Head selection protocol. The protocol ends when either the Replication Strategy heuristic criterion determines there are no more promising nodes, or the *current EN_value = Min_Int((undesired searched EN_value)/2)*, where *Min_Int(x)* returns the least integer greater than *x*. Since Replication Strategy considers bi-directional links among MANET nodes, when the above equation is verified, it is easy to demonstrate that Replication Strategy has reached the optimal solution for the Head selection. In particular, Replication Strategy combines two different strategies against Head assignment degradation, one proactive and one reactive, which operate, respectively, over large and medium time periods (*Tp* and *Tr* with *Tp>>Tr*). The proactive maintenance strategy establishes that the current Head always triggers a new Head selection after *Tp* seconds. In addition to probabilistically improving the centrality of the Head position, the periodical re-execution of the selection process contributes to distribute the burden of the role among different nodes, thus avoiding depleting the energy resources of a single participant. Moreover, let us rapidly observe that only the nodes located in the proximity of the dense MANET topology center have high probability to assume the Head role. Therefore, a *target_EN_value* can be easily determined equal to the *EN_value* of the current Head, thus speeding up Head selection and reducing the protocol overhead. In addition to the above proactive degradation counteraction, Replication Strategy exploits a reactive strategy that consists in repeating the farthest node determination at regular *Tr* periods, with the goal of understanding the current Head distance from the optimal placement. If the distance of Head farthest nodes, i.e., its *new_EN_value*, has increased if compared with the distance estimated at the moment of its selection, i.e., its *EN_value*, Replication Strategy realizes that the Head has moved from the topologic center in a significant way; in that case, the Head itself triggers a new selection process. The following subsections detail the adopted heuristics to limit the number of searched ENs and the exploited solution to determine, given a node, its farthest nodes in the dense MANET.

REFERENCES

A. K. Saha and D. B. Johnson (2004). Modeling Mobility for Vehicular Ad Hoc Networks. In *ACM VANET'04*.

I. Chlamtac, M. Conti, and J.-N. Liu (2003). Mobile ad hoc networking: Imperatives and challenges. *Elsevier Journal on Ad Hoc Networks*, 1(3).

L. Buttyan and J.-P. Hubaux (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5).

L. Dlagnekov and S. Belongie (2005). Recognizing Cars. Technical Report CS605-081, UCSD CSE.

M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. (2005). Iftode. VITP: an information transfer protocol for vehicular computing . In *ACM VANET*.

V. Naumov, R. Baumann, and T. Gross (2006). An Evaluation of Inter-Vehicle Ad Hoc Networks Based on Realistic Vehicular Traces. In *ACM MOBIHOC*.

Z. Fu, P. Zerfos, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla (2003). The impact of multihop wireless channel on tcp throughput and loss. In *IEEE INFOCOM*.

Corresponding Author

Navneet Garg*

Research Scholar, OPJS University, Churu, Rajasthan

E-Mail – ashokkumarpksd@gmail.com