International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-3, February 2019

A Novel Cache Discovery Framework Based on Clusters and Highest Node Density

Y.J.Sudha Rani, M.Seetha

Abstract— With the increasing popularity of MANETs, due to increase in mobility for communication, the application developers are building more and more complex applications for MANET. The recent advancements in mobility for communication has also discovered newer communication methods and network architectures such as Internet of Things. Nevertheless, due to the higher deployment and component costs of IoT, MANET is still a preferable architecture for many fields of utilizations. Also, in IoT, the security issues are higher compared with the MANETs as the incorporation of the smart devices can introduce the higher attention of the intruders into the networks. Henceforth, the preferable lower cost and higher utilization factors of MANETs are increasing the application developers focus on building complex applications, which demands higher specifications in the MANET devices. Agreeable Caching is a methodology that has been utilized broadly inside MANETs to address different execution and asset challenges. Connected appropriately, Helpful Caching can essentially lessen in general transfer speed use. Nonetheless, making the MANET devices with higher capabilities with more sophisticated hardware is always the bottleneck for building cost effective solutions. Hence, the MANET architects are focusing on building more capable architectures. The cooperative caching strategies can significantly increase the architectural capabilities. Several parallel research works have demonstrated cooperative caching mechanisms in the recent times. Regardless to mention, the parallel research outcomes are also highly criticized due to the fact higher complexity during cache discovery. The outcomes from these parallel researches have failed to justify the effect of dynamic clusters and the effect of node density during the discovery process. The proposed framework for cache discovery and cooperative caching replies on identification of the cluster heads with priority identification for determining the possible cooperativeness in the cache. Also, the proposed framework determines the best possible cooperative caching regions based on another novel method called density identification. Finally, the proposed method demonstrates 100% cache cooperation possibilities with 50% less time complexity.

Keywords— Cooperative Cache, MANET, Clustering, Density Estimation, ZRP, DSDV, DSR, AODV, TORA, BGP, EIGRP.

I. INTRODUCTION

A versatile specially appointed system (MANET), otherwise called remote impromptu system or impromptu remote system, is a persistently self-designing, framework less system of cell phones associated remotely. Every gadget in a MANET can move autonomously toward any path and will thusly change its connects to different gadgets often. Each must forward traffic disconnected to its very own utilization, and along these lines be a switch. The essential test in building a MANET is preparing every gadget to constantly keep up the data required to legitimately course traffic. Such systems may work without anyone else's input or might be associated with the bigger Internet. They may contain one or numerous and diverse handsets between hubs. These outcomes in a unique, self-ruling topology. MANETs are a sort of remote specially appointed system that normally has a routable systems administration condition over a Link Layer impromptu system. MANETs comprise of a shared, self-framing, self-mending system. The conspicuous intrigue of MANETs is that the system is decentralized, and hubs/gadgets are portable, in other words, there is no settled framework which gives the likelihood to various applications in various regions, for example, ecological observing, calamity help, and military correspondences. Since the mid-2000s enthusiasm for MANETs has significantly expanded which, partially, is because of the reality portability can enhance organize limit, appeared by Grossglauser et al. [1] alongside the presentation of new advances. The applicability of the MANET is extended into multiple other architecture as demonstrated by P. Ranjan et al. [2]. Due to the extensibility of the architecture, the applicability of MANETs are also increasing. The complex applications running on the MANETs demands more hardware capabilities. The major capability demand is to enable the complete network with more temporary storages in order to support the increasing demand of processing capabilities. The most prominent method of increase the temporary processing storage in MANETs is to include cooperative caching technique for the MANETs. During the course of study, this work identifies the following motivations to involve in the cooperative caching research:

a) Firstly, additional shared storage can be made available to the MANET devices and to the complete network. In order to perform computation on the large amount of the data available, the cache sizes are to be increased. However, the increasing the cache size is not a feasible solution due to the cost to performance bottleneck.

b) Secondly, during the data processing on the MANET architecture, it is often observed that the application demands accessing shared data, which can easily be achieved by cooperative caching methods.

c) Finally, due to the mobility nature of the applications, it is also possible that the devices move out of the reach of the main data source, still demands a strong accessibility to the data. In situation can also be managed by deploying the cache cooperativeness.

Manuscript published on 28 February 2019.
* Correspondence Author (s)
Y.J.Sudha Rani*, Research Scholar, JNTUH- Hyderabad, Telangana, India
Dr. M. Seetha, HOD, Professor, G Narayanamma Institute of Technology and Science Hyderabad, Telangana, India

As similarly suggested by Steve Glass et al. [3], in order to resolve the above problems, it is must to design a framework to fulfil the generic process as cache discovery, cache management and cache consistency.

Realizing the demand for the research, in the work, a novel cache discovery technique is demonstrated. The proposed framework replies on strong analysis of the cluster-based cache discovery and cache cooperation based on the density analysis. In order to achieve the above two objectives, this work deploys machine learning methods, which is elaborated in the further sections of the work.

Furthermore, this work is organized such that, in Section – II the fundamentals of the MANETs are identified and discussed to realize the improvement points in the architecture, in the Section – III, the outcomes from the parallel researches are discussed on cache discovery techniques to realize the most recent improvements and the existing drawbacks on the technique, in the Section – IV, the problem is formulated using mathematical modelling, in Section – V, the novel cluster building algorithm is proposed and discussed, in the Section – VI, the novel algorithm for density based cache cooperation is highlighted, in the Section – VII the complete framework working pattern is elaborated, in the Section – VIII, the results obtained from different phases of the work is furnished, discussed and explained, in the Section – IX, this work presents the final conclusion.

II. MANET ARCHITECTURE

In this section of the work, the typical architectures of MANET implementations are identified in order to realize the feasibilities of cache discovery and more so, the cache cooperation possibilities.

The decentralized idea of remote impromptu systems makes them reasonable for an assortment of utilizations where focal hubs can't be depended on and may enhance the adaptability of systems contrasted with remote oversaw systems, however hypothetical and functional breaking points to the general limit of such systems have been recognized. Negligible setup and speedy arrangement make impromptu systems appropriate for crisis circumstances like cataclysmic events or military clashes. The nearness of dynamic and versatile directing conventions empowers impromptu systems to be shaped rapidly.

A. Vehicular ad hoc networks

VANETs are utilized for correspondence among vehicles and roadside hardware. Insightful vehicular specially appointed systems are a sort of man-made consciousness that causes vehicles to carry on in shrewd habits amid vehicle-to-vehicle impacts, mishaps. Vehicles are utilizing radio waves to speak with one another, making correspondence organizes in a split second on-the-fly while vehicles are proceeding onward the streets [Fig – 1].

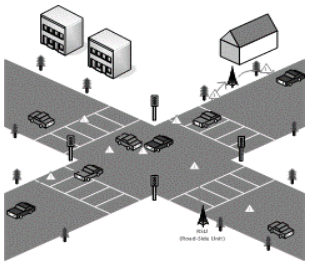

Fig. 1 Schematic of the VANET

B. Smart phone ad hoc networks

A SPAN uses existing equipment and programming in industrially accessible cell phones to make distributed systems without depending on cell bearer systems, remote passages, or customary system framework. Most as of late, Apple's iPhone with adaptation 8.4 iOS and higher have been empowered with multi-peer specially appointed work organizing capacity, in iPhones, permitting a large number of cell phones to make impromptu systems without depending on cell correspondences [Fig – 2].

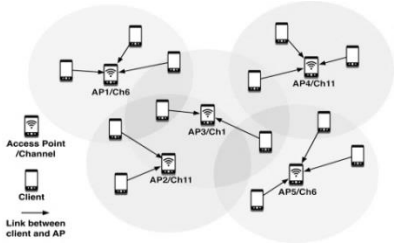

Fig. 2 Schematic of the SPNET

C. Wireless mesh networks

Work systems take their name from the topology of the resultant system. In a completely associated work, every hub is associated with each other hub, shaping a "work". A halfway work, on the other hand, has a topology in which a few hubs are not associated with others, in spite of the fact that this term is only from time to time being used. Remote specially appointed systems can appear as a work systems or others. A remote impromptu system does not have settled topology, and its availability among hubs is absolutely reliant on the conduct of the gadgets, their versatility designs, separate with one another, and so forth. Consequently, remote work systems are a specific sort of remote impromptu systems, with uncommon accentuation on the resultant system topology. While some remote work systems (especially those inside a home) have generally rare versatility and consequently inconsistent connection breaks, other progressively portable work systems require visit steering changes in accordance with record for lost connections. Google Home, Google Wi-Fi, and Google OnHub all help Wi-Fi work (i.e., Wi-Fi specially appointed) organizing. Apple's AirPort permits the arrangement of remote work organizes at home, interfacing different Wi-Fi gadgets together and giving great remote inclusion and availability at home [Fig – 3].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

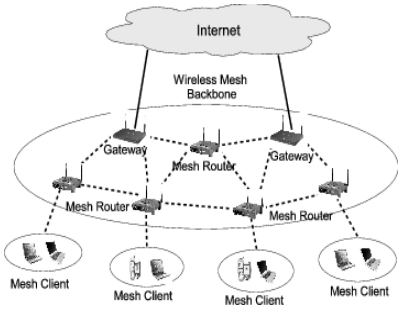

Fig. 3 Schematic of the WMNET

D. Army tactical MANETs

Armed force has need "moving" interchanges for quite a while. Specially appointed versatile interchanges come in well to satisfy this need, particularly its infrastructure less nature, quick arrangement and task. Military MANETs are utilized by military units with accentuation on fast organization, infrastructure less, every single remote system (no settled radio towers), heartiness (connect breaks are no issue), security, range, and moment task. MANETs can be utilized in armed force "bouncing" mines, in companies where officers impart in remote territories, giving them prevalence in the front line. Strategic MANETs can be shaped naturally amid the mission and the system "vanishes" when the mission is finished or decommissioned. It is in some cases called "on-the-fly" remote strategic system [Fig – 4].

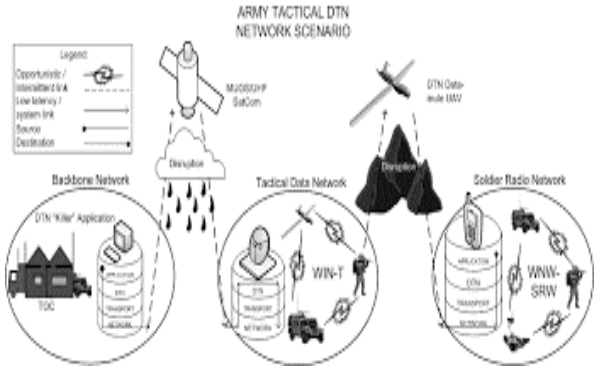

Fig. 4 Schematic of the AT-MANET

E. Air Force UAV Ad hoc networks

UAVs have likewise been utilized by US Air Force for information gathering and circumstance detecting, without taking a chance with the pilot in an outside threatening condition. With remote impromptu system innovation implanted into the UAVs, various UAVs can speak with one another and fill in as a group, cooperatively to finish an undertaking and mission. In the event that a UAV is obliterated by a foe, its information can be immediately offloaded remotely to other neighbouring UAVs. The UAV impromptu correspondence organize is likewise some of the time alluded to UAV moment sky arrange [Fig – 5].

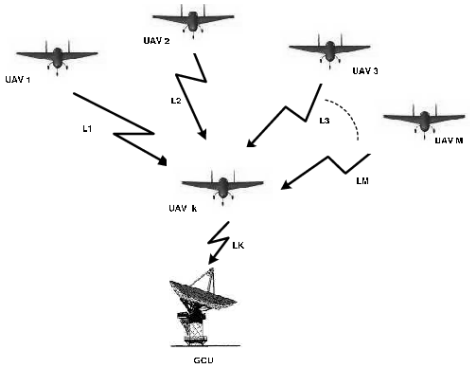

Fig. 5 Schematic of the UAVNET

F. Navy ad hoc networks

Naval force sends customarily utilize satellite correspondences and other oceanic radios to speak with one another or with ground station back ashore. Be that as it may, such interchanges are confined by deferrals and restricted transmission capacity. Remote specially appointed systems empower dispatch territory systems to be shaped while adrift, empowering rapid remote interchanges among boats, upgrading their sharing of imaging and sight and sound information, and better co-appointment in combat zone activities. Some resistance organizations, (for example, Rockwell Collins and Rohde and Schwartz) have created items that upgrade deliver to-ship and ship-to-shore correspondences.

G. Ad hoc networks of robots

Robots are mechanical frameworks that drive computerization and perform errands that would appear to be troublesome for man. Endeavor have been made to co-ordinate and control a gathering of robots to embrace cooperative work to finish an undertaking. Brought together control is frequently founded on a "star" approach, where robots alternate to converse with the controller station. In any case, with remote specially appointed systems, robots can shape a correspondence arrange on-the-fly, i.e., robots can now "talk" to one another and team up in a conveyed manner. With a system of robots, the robots can discuss among themselves, share neighbourhood data, and distributive choose how to determine an undertaking in the best and proficient way [Fig – 6].

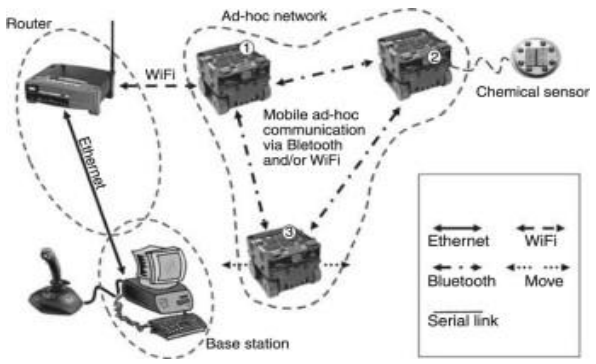

Fig. 6 Schematic of the ROBONET

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

H. Disaster rescue ad hoc network

Another regular citizen utilization of remote impromptu system is open security. On occasion of fiascos (surges, storms, quakes, fires, and so forth.), a speedy and moment remote correspondence organize is important. Particularly on occasion of seismic tremors when radio towers had crumpled or were crushed, remote impromptu systems can be shaped freely. Fire fighters and protect specialists can utilize impromptu systems to convey and save those harmed. Business radios with such ability are accessible available.

Further, it is natural to realize that, in all applicable situation of MANET, the use of cooperative caching will significantly improve the performances.

One method that has been utilized, as it were, in tending to these issues is Cooperative Caching. Storing itself centresaround the spatial and worldly attributes of information. For the most part, if an information thing has been gotten to by one hub, quite possibly the information thing may be

required by a similar hub (spatial) sooner rather than later (fleeting). Agreeable Caching is a type of "in-arrange" reserving in which at least two hubs cooperate to help an aggregate store. By cooperating, the hubs can give upgraded storing administrations that can address the impediments and difficulties to a lot more prominent degree than can be given by a solitary hub.

The generic process for establishing cooperative caching strategies are classified in three phases as discovery, management and sustain. Further, in this section of the work the process phases are realized.

A. Cache Discovery

The first phase of the process is discovery phase. In this phase, the software agent, during the operation stage, tries to identify the availability of the data in the local cache. If the data item is found locally, then the phase ends and returns the data item. Other, if the data item is not found in the local cache, then the discovery software agent sends signal to all other nodes searching for the data. Once the node with the desired data item is found, then the node is identified.

B. Cache Management

The second phase of the process is management phase. In this phase, the software agent determines the location of the identified shared data to be stored. The identified location, called the cooperative cache, must comply with the lower bandwidth utilization, such that while sharing the data from cooperative cache, bandwidth overhead should be less.

C. Cache Consistency

The final phase of the process is consistency phase. The software agent deployed in the network checks periodically and decides whether to keep the cache data or to discard the cache data during the lifetime of the network. In general, this strategy is managed using any lifetime notifier property of the data.

As demonstrated in the work of S. Lim et al. [4], the dismissal of the data from the cache in the MANET or other form of MANET architectures is highly recommended in order to reduce the chances of cache miss. Also, the work of N. Kumar et al. [5] have signified that the presence of stale data in the cooperative cache during MANET of other forms of MANET communication can create higher delay in the network and finally results into non-responsive network or nodes. Yet another problem, as highlighted by R. Shokri et al. [6], the longer use of similar cache location with the stale data, can also intricate the higher changes of attacks on the network due to over exposing of the cached node locations.

Henceforth, with the understanding of the fundamental process of cooperative caching, in the next section of this work, the recent advancements are highlighted and analysed.

III. OUTCOMES OF THE PARALLEL RESEARCHES

In this section of the work, the outcomes from parallel research approaches are analysed for different strategies of cooperative caching.

MANETs are remote systems in which hubs are prepared to do physically moving starting with one area then onto the next while giving different administrations to their clients. The development of the hubs results in unique system topology and prompts a few impediments and specialized difficulties. A few confinements are because of the physical sizes of the gadgets themselves. Different difficulties are the roundabout aftereffect of the versatile, impromptu nature of the system itself.

When utilizing broadcasting to find information things inside the aggregate store, a hub uses the communicate bolster given by the basic information conventions to send the information demand to all hubs inside its transmission extend. The work of Gast et al. [7] and the recommendation by the Internet Protocol standards [8] confines to the same theory.

Also, considering on the outcomes of Majd et al. [9] the fundamental type of broadcasting is utilized in the Split Cache approach. At the point when a demand is communicated, it incorporates a jump check parameter that constrains the scope of the demand. In this plan, the reserve is apportioned into two sections dependent on fame.

Another methodology that depends after communicating. In this plan, hubs are assembled into bunches and the aggregate reserve is parcelled into a common territory and a private region as demonstrated by Caetano et al. [10]. In the same direction, yet another work by Meisel et al. [11] and further enhanced by Pappas et al. [12, 13], CCN based methodology that uses broadcasting for reserve disclosure. In the fundamental CCN approach, a hub with substance can report that substance to the system.

Further, with the detailed understanding of the parallel and recent research outcomes, in the next section of the work, the problem is identified and formulated.

IV. PROBLEM FORMULATION

In this section of the work, the problems associated with cooperative caching are identified and formulated using mathematical model. The advantage of analysing the problem using mathematical modelling is to find attributes or parameters, which can be enhanced further associated with the existing algorithm.

In order to formulate the problem in this case, the work deploys two lemmas for reducing the overhead of cache discovery and cache management respectively.

Lemma – 1: During the discovery phase, use of cluster-based approach can reduce the time complexity of the algorithm.

Where, N &n_i, denotes the total number of nodes and every individual node in the network respectively t, denotes the time to communicate to one node during the discovery phase CH, denotes the total possible clusters in the network NCH_i, denotes the nodes in i^{th} clusters

Proof: Firstly, let's assume that all the nodes in the network are part of the N set of nodes and can be denoted as,

$$N = \sum_{i=0}^{k} n_i \quad \text{(Eq.1)}$$

Secondly, the unit communication time for each node to node is considered as t. Hence, during the discovery phase, the total time (ϑ) taken can be formulated as,

$$\vartheta = N * t \quad \text{(Eq.2)}$$

This also can be realized as,

$$\vartheta = \sum_{i=0}^{k} n_i * t \quad \text{(Eq.3)}$$

Further, as proposed, the complete network is divided into CH number of possible clusters. Thus, the number of nodes ($\Im$) in each cluster can be formulated as,

$$\Im = \sum_{i=0}^{k} n_i / CH \quad \text{(Eq.4)}$$

This can also be rewritten as,

$$\Im = N / CH \quad \text{(Eq.5)}$$

Hence, it is natural to understand that the number of total nodes will be much greater than the number of nodes in each cluster and can be denoted as,

$$N \gg \Im \quad \text{(Eq.6)}$$

Further, the communication time (τ) between each node in the single cluster can be formulated as,

$$\tau = \Im * t \quad \text{(Eq.7)}$$

In the light of the Eq. 6, as the time t is same for both the situations, it is natural to realize that, the communication time for the complete network is much higher than the communication time in a single cluster and can be denoted as,

$$\vartheta \gg \tau \quad \text{(Eq.8)}$$

Finally, in general situation, it is understandable that the availability of the cache data is most likely to be found in the same cluster as the nodes in the same clusters are mostly deployed for the same task and demands similar data.

Hence, it is proven that during the discovery phase of cooperative caching, clustering the nodes into multiple clusters can reduce the discovery time.

Lemma – 2: Considering the possible cache location in the higher node density locations can reduce the overhead of network communications.

Where, D and d_i, denotes the total distance vector and the distance from identified node respectively T and t_i, denotes the communication time vector and the communication time from the identified node respectively

Proof: Firstly, let's assume that each distance component and communication time is the part of distance vector and communication time vector and can be denoted as,

$$D = [d_1, d_2, d_3, d_4 ..... d_n] \quad \text{(Eq.9)}$$

And,

$$T = [t_1, t_2, t_3, t_4 ..... t_n] \quad \text{(Eq.10)}$$

Here, it is natural to realize that, the elements of the distance vector (D) is proportional to the communication time vector (T).

Further, assuming K number of nodes are close together and making the region a higher density region and L number of nodes are placed far from the selected node.

Hence, the maximum distance in the K nodes can be denoted as d_K and the in the L nodes can be denoted as d_L. Similarly, the communication time can be denoted as, t_K and t_L.

It is natural to realize that, if the nodes are closely situated, the maximum distance will be less than the distance of far situated node. This can be formulated as,

$$d_K \ll d_L \quad \text{(Eq.11)}$$

By the same principle, in case of communication time also, the communication time will be less for the closely situated nodes and can be represented as,

$$t_K \ll t_L \quad \text{(Eq.12)}$$

Finally, in fundamental situations, the utilization of the bandwidth depends on the time needed for the communication and the distance that the data travels. Assuming the bandwidth requirement ($\hbar$) for closely situated nodes and for the far situated nodes (λ) can be formulated as,

$$\hbar = d_K * t_K \quad \text{(Eq.13)}$$

And

$$\lambda = d_L * t_L \quad \text{(Eq.14)}$$

Considering Eq. 11 and Eq. 12, it is natural to release that, the bandwidth requirements will be less in case of closely situated nodes and can be formulated as,g

$$\hbar \ll \lambda \quad \text{(Eq.15)}$$

Hence, it is proven that, if the cooperative cache is configured in the highly dense node situations, then the communication and bandwidth overhead will reduce.

Further, in the light of the problem formulation, in the upcoming sections of this work, the proposed algorithms are demonstrated.

V. CLUSTER BUILDING ALGORITHM

As clearly defined in the problem formulation section of this work, building the cluster can significantly reduce the problem of communication overhead. Thus, building the best possible cluster is one of the prime objective of this work.

In this section of the work, the cluster building algorithm for MANET is elaborated. The assumption of this algorithm is to have a pre-group of the nodes which are connected to a single application. During demonstration of the results, it is assumed that all the nodes are part of three disruptive application processing.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The proposed algorithm deploys machine learning approaches in order to reduce the time complexity further and the use of fitness function makes this algorithm suitable under the machine learning algorithm category.

This first algorithm results into suitable clusters in the group. The algorithm achieves its sophistication with the help of application grouping to reduce unnecessary cluster members and achieve higher accuracy.

Once the cluster is built, the next step is to identify the node with the desired data. Here, the search space for the data is reduced in single clusters and as proven in the Lemma – 1, the search time will reduce significantly.

Hence, this algorithm reduces the time complexity of cache discovery process. Regardless to mention, the generic algorithms for identifying the cache are evaluated based on the cache hit and cache miss ratio, whereas this algorithm accuracy must be measured in terms of time to discover cache for MANET routing protocols.

Algorithm - 1: Fitness Function Based Dependency Clustering (FFDC)

Step - 1. Accept the list of nodes in the network
Step - 2. For each node
- a. Access the Coordinate
- b. Access the Cache Size (CZ)
- c. Access the Distance from the Base Station
- d. Access the Energy Disruption Rate (EDR)
- e. Access the Application group

Step - 3. Separate the nodes of the network based on application groups
Step - 4. For each application group
- a. For each node
  - i. Calculate the distance from other nodes using coordinates as $SQRT\{(X1-X2)^2+(Y1-Y2)^2\}$
  - ii. Build the distance vector for each node
- b. Select the first random node based on the least distance vector sum
- c. Repeat Step – 4.b until least distance vector is achieved
- d. Build the sub groups of nodes

Step - 5. For each sub group under the application groups
- a. Build the fitness function as (High CZ, Low EDR)
- b. If FitnessValue (Group[i]) >FitnessValue (Group[j])
  - i. Reject Group[j]
  - ii. Store Group[i] under cluster group

Step - 6. Generate final cluster groups
Step - 7. Apply Discovery phase to identify the cache location

After the cluster building, in the further section of this work, the suitable node or node region is identified in order to build the cooperative cache.

VI. NODE DENSITY IDENTIFICATION ALGORITHM

In this section of the work, the proposed node density identification algorithm is proposed in order to realize the most effective location for the cache, so that the communication over-head can be reduced.

This proposed algorithm applies the kernel density estimation function in order to find the highest population area.

Algorithm - 2: Kernel Density Estimation Based Dense Region Detector (KDE-DRD)

Step - 1. Accept the cluster list from FFDC algorithm
Step - 2. For each node
- a. Access the Coordinate (CC)
- b. Access the Cache Size (CZ)

Step - 3. For each node
- a. Accept the first random node in the sample
- b. Calculate the likelihood using KDE using CC, CZ
- c. For Other nodes - Except SelectedNode
  - i. If Node[i].LikelyHood> 0.5
  - ii. Place in the node in DensitySet

Step - 4. Calculate the population of the DensitySet
Step - 5. Report the highest population DensitySet
Step - 6. Build the cooperative cache in any selected node under HighestDensitySet

This algorithm uses two popular concepts from the machine learning algorithm space as Likelihood and Population. The KDE algorithm ensures to evaluate the similar characteristics of two given nodes in the network and reports the similarities in terms of fraction called the likelihood. The standard value of the likelihood is expected to be higher than 0.5 in order to denote the likeliness. The population is the size of the set or can be considered as number of elements in a set or in this case the number of nodes in the cluster under application group under similar likelihood.

Henceforth, this algorithm is building the cooperative cache in the higher node density regions in order to reduce the communication over-head as proven in Lemma – 2.

Further, in the next section of the work, the complete working model of the framework is elaborated.

VII. PROPOSED FRAMEWORK

In this section of the work, the proposed framework as combination of the proposed and existing algorithms are discussed [Fig – 7].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The proposed framework is not a replacement of existing cache cooperative frameworks, rather can be seen as improvements of the performances.

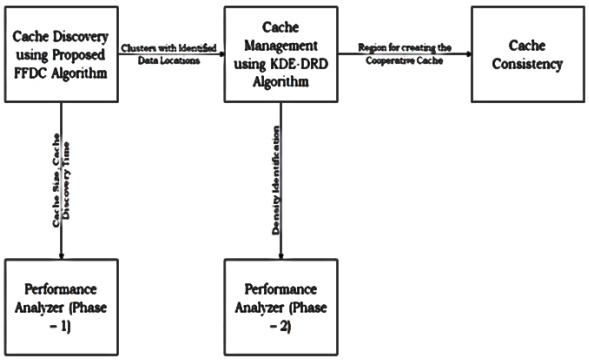

Fig. 7 Schematic of the Proposed Framework

The Ad-Hoc organize clients manage a gathering of information and projects that are explicit to the application and must be shared by every one of the hubs. This information must be conveyed among the cell phones. At the point when a versatile hub needs a duplicate of it, the information ought to be transmitted over radio connections that are pricey in power utilization, consequently information developments must be restricted and completely considered.

During the execution of the proposed framework, firstly, the parametric inputs such as initial node coordinates, size of the available cache for each node, distance from the base station and under running condition the rate of energy disruption is provided to the first part of the framework as discovery phase.

In the discovery phase, due to the proposed FFDC algorithm, the time for the cache discovery is significantly reducing.

In remote Ad-hoc arranges, the information reserving procedure turns into an issue. Reserving information at hubs assists a remote system framework with running quicker, lessen the expense of data transfer capacity and dodge over-burden at hubs with even more effectively. After it has been chosen to store, the following issue is the thing that to reserve and where.

Further, after the building of the clusters, the highest density node regions are identified, and the cooperative cache is built.

Furthermore, the proposed framework allows the network architects to build and incorporate the existing algorithms for cache consistency verification using dirty bit or similar existing algorithms.

In the next section of the work, the obtained results from the proposed algorithms are discussed.

VIII. RESULTS AND DISCUSSION

The obtained results from the proposed algorithms are highly satisfactory and are discussed here. The results are discussed in multiple phases as Experimental Setup, Clustering Result, Density Based Region Building Results and Cache Discovery Results.

A. Experimental Setup

The experimental setup used in this work is furnished here [Table – 1].

TABLE IEXPERIMENTAL SETUP

Number of Nodes	45
Dimension of the MANET area	100 X 100 cm square
Initial Battery Level	100%
Initial Cache Size	1 MB
Routing Algorithms Tested	ZRP, DSDV, DSR, AODV, TORA, BGP, EIGRP
Node Placement	Random
Number of Applications (Running using the MANET)	3

The setup is initialized with 45 nodes and the assumed space of the MANET is 100 X 100 square cm. All the nodes in the MANET network is initialized with 100 or full battery level and the cache size is kept minimum in order to test the complete capability of the proposed algorithms.

The proposed algorithms are tested under all possible proactive and reactive and hybrid algorithms in order to justify the advantages achieved under this framework.

The nodes are placed randomly and considering that a total of 3 applications are running, which makes all the nodes falling under any of the 3 application groups.

Few random placements of the nodes are showcased here [Fig – 8] [Fig – 9] [Fig – 10].

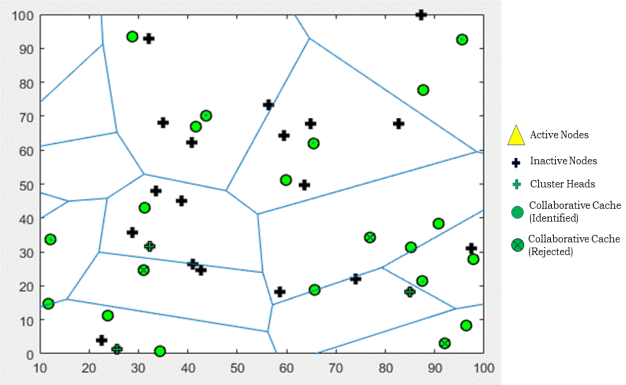

Fig. 8 Sample Node Placements at time t1

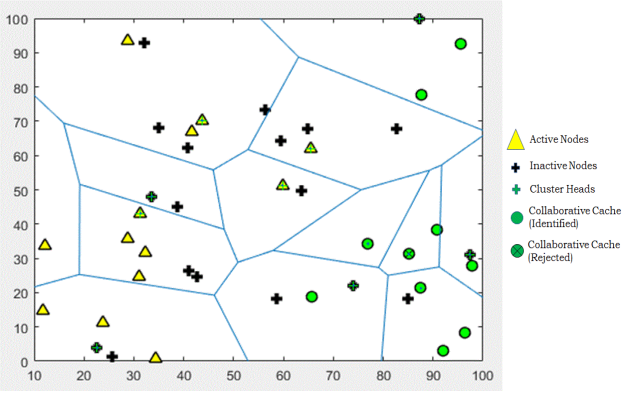

Fig. 9 Sample Node Placements at time t2

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

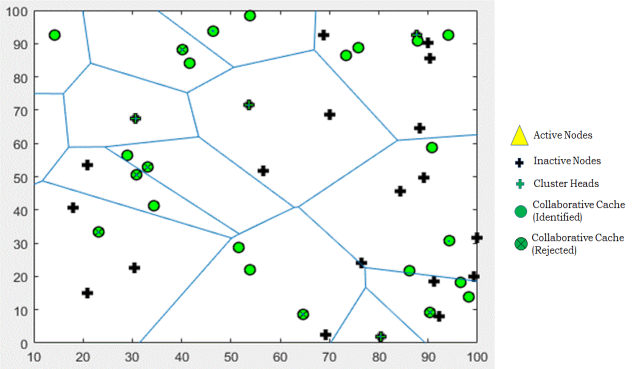

Fig. 10 Sample Node Placements at time t3

The time instances are random and not mentioned in any specific order.

B. Clustering Result

The proposed clustering algorithm analyses the node set in three phases. Firstly, based on distance vector, secondly, based on the cache size and finally based on the energy disruption the node groups are categorised. The initial network setup is elaborated here [Table – 2].

TABLE IIINITIAL NETWORK SETUP

Application Group	Node Number	Coordinate - X	Coordinate - Y	Cache Size	Energy Disp Rate
App-2	Node - 01	20.664531	31.888399	0.14864	0.1995
App-2	Node - 02	94.499638	20.419879	0.14727	0.1995
App-3	Node - 03	77.037164	93.971716	0.14596	0.1995
App-3	Node - 04	53.252631	67.051655	0.14452	0.1995
App-1	Node - 05	90.849277	39.779308	0.14312	0.1989
App-2	Node - 06	30.395177	16.946628	0.14178	0.0973
App-2	Node - 07	47.805868	20.311743	0.14044	0.0994
App-3	Node - 08	73.480698	6.187273	0.13910	0.0991
App-1	Node - 09	95.760129	55.334276	0.13776	0.0995
App-1	Node - 10	24.493251	52.1831	0.13638	0.0996
App-3	Node - 11	3.34984	91.462536	0.13499	0.0995
App-3	Node - 12	48.089315	82.988504	0.13362	0.0996
App-2	Node - 13	60.891704	0.001875	0.13226	0.1994
App-1	Node - 14	36.430716	69.337715	0.13091	0.1994
App-3	Node - 15	78.821971	24.501411	0.12957	0.1991
App-1	Node - 16	61.124482	18.193996	0.12826	0.0982
App-2	Node - 17	72.364259	46.208793	0.12689	0.0994
App-1	Node - 18	83.922094	7.475875	0.12554	0.0986
App-1	Node - 19	38.357786	58.104835	0.12419	0.0971
App-2	Node - 20	75.299951	74.980479	0.12282	0.1990
App-3	Node - 21	68.756038	96.659404	0.12149	0.1977
App-2	Node - 22	12.722171	36.154001	0.12024	0.1976
App-2	Node - 23	99.603831	69.304409	0.11896	0.1989
App-3	Node - 24	70.288487	15.04682	0.11759	0.1991
App-2	Node - 25	8.519967	96.071189	0.11628	0.1992
App-1	Node - 26	38.25332	56.779527	0.11506	0.1990
App-1	Node - 27	43.770182	5.017008	0.11381	0.1972
App-3	Node - 28	76.078311	62.713586	0.11252	0.0973
App-2	Node - 29	62.310664	60.183176	0.11121	0.0990
App-1	Node - 30	60.891704	1.001875	0.10994	0.1971
App-3	Node - 31	49.147517	16.593373	0.10866	0.1989
App-2	Node - 32	14.428134	74.194237	0.10747	0.1975
App-1	Node - 33	91.282739	88.024079	0.10616	0.0980
App-2	Node - 34	35.354088	67.861447	0.10494	0.0989
App-1	Node - 35	99.57073	26.21558	0.10375	0.0990
App-1	Node - 36	10.74737	5.845918	0.10257	0.0980
App-2	Node - 37	47.098513	33.289519	0.10151	0.0979
App-2	Node - 38	89.864595	88.227072	0.10037	0.0988
App-3	Node - 39	85.399404	93.654858	0.09918	0.0989
App-3	Node - 40	3.098375	27.72663	0.09804	0.1983

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

App-3	Node - 41	81.640109	10.979579	0.09690	0.1986
App-1	Node - 42	15.624787	63.706146	0.09575	0.1987
App-2	Node - 43	26.68181	47.153596	0.09460	0.1981
App-1	Node - 44	11.321026	77.820895	0.09361	0.1987
App-3	Node - 45	71.223372	77.654445	0.09254	0.1980

Hence, after the first iteration of the algorithm, the nodes are classified into multiple clusters. The result is presented here [Table – 3].

TABLE IIIAFTER FIRST ITERATION OF THE FFDC

Node Number	Selected Cluster
Node - 01	CH-2
Node - 02	CH-2
Node - 03	CH-3
Node - 04	CH-3
Node - 05	CH-1
Node - 06	CH-2
Node - 07	CH-2
Node - 08	CH-3
Node - 09	CH-1
Node - 10	CH-1
Node - 11	CH-3
Node - 12	CH-3
Node - 13	CH-2
Node - 14	CH-1
Node - 15	CH-3
Node - 16	CH-1
Node - 17	CH-2
Node - 18	CH-1
Node - 19	CH-1
Node - 20	CH-2
Node - 21	CH-3
Node - 22	CH-2
Node - 23	CH-2
Node - 24	CH-3
Node - 25	CH-2
Node - 26	CH-1
Node - 27	CH-1
Node - 28	CH-3
Node - 29	CH-2
Node - 30	CH-1
Node - 31	CH-3
Node - 32	CH-2
Node - 33	CH-1
Node - 34	CH-2
Node - 35	CH-1
Node - 36	CH-1
Node - 37	CH-2
Node - 38	CH-2
Node - 39	CH-3
Node - 40	CH-3
Node - 41	CH-3
Node - 42	CH-1
Node - 43	CH-2
Node - 44	CH-1
Node - 45	CH-3

Hence, after the second iteration of the algorithm, the nodes are classified into multiple clusters. The result is presented here [Table – 4].

TABLE IVAFTER SECOND ITERATION OF THE FFDC

Node Number	Selected Cluster
Node - 01	CH-3
Node - 02	CH-3
Node - 03	CH-3
Node - 04	CH-3
Node - 05	CH-3
Node - 06	CH-3
Node - 07	CH-3
Node - 08	CH-3
Node - 09	CH-3
Node - 10	CH-3
Node - 11	CH-3
Node - 12	CH-3
Node - 13	CH-3
Node - 14	CH-3
Node - 15	CH-2
Node - 16	CH-2
Node - 17	CH-2
Node - 18	CH-2
Node - 19	CH-2
Node - 20	CH-2
Node - 21	CH-2
Node - 22	CH-2
Node - 23	CH-2
Node - 24	CH-2
Node - 25	CH-2
Node - 26	CH-2
Node - 27	CH-2
Node - 28	CH-2
Node - 29	CH-2
Node - 30	CH-1
Node - 31	CH-1
Node - 32	CH-1
Node - 33	CH-1
Node - 34	CH-1
Node - 35	CH-1
Node - 36	CH-1
Node - 37	CH-1
Node - 38	CH-1
Node - 39	CH-1
Node - 40	CH-1
Node - 41	CH-1
Node - 42	CH-1
Node - 43	CH-1
Node - 44	CH-1
Node - 45	CH-1

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Hence, after the third iteration of the algorithm, the nodes are classified into multiple clusters. The result is presented here [Table – 5].

TABLE VAFTER SECOND ITERATION OF THE FFDC

Node Number	Selected Cluster
Node - 01	CH-2
Node - 02	CH-2
Node - 03	CH-2
Node - 04	CH-2
Node - 05	CH-2
Node - 06	CH-1
Node - 07	CH-1
Node - 08	CH-1
Node - 09	CH-1
Node - 10	CH-1
Node - 11	CH-1
Node - 12	CH-1
Node - 13	CH-2
Node - 14	CH-2
Node - 15	CH-2
Node - 16	CH-1
Node - 17	CH-1
Node - 18	CH-1
Node - 19	CH-1
Node - 20	CH-2
Node - 21	CH-2
Node - 22	CH-2
Node - 23	CH-2
Node - 24	CH-2
Node - 25	CH-2
Node - 26	CH-2
Node - 27	CH-2
Node - 28	CH-1
Node - 29	CH-1
Node - 30	CH-2
Node - 31	CH-2
Node - 32	CH-2
Node - 33	CH-1
Node - 34	CH-1
Node - 35	CH-1
Node - 36	CH-1
Node - 37	CH-1
Node - 38	CH-1
Node - 39	CH-1
Node - 40	CH-2
Node - 41	CH-2
Node - 42	CH-2
Node - 43	CH-2
Node - 44	CH-2
Node - 45	CH-2

Furthermore, the analysis for final clustering is considered based on the node clustering similarity for each iteration. The result is documented here [Table – 6].

TABLE VIAFTER FINAL ITERATION OF THE FFDC

Node Number	Iteration – 1	Iteration – 2	Iteration – 3
Node - 33	CH-1	CH-1	CH-1
Node - 35	CH-1	CH-1	CH-1
Node - 36	CH-1	CH-1	CH-1
Node - 20	CH-2	CH-2	CH-2
Node - 22	CH-2	CH-2	CH-2
Node - 23	CH-2	CH-2	CH-2
Node - 25	CH-2	CH-2	CH-2

Finally, the accuracy of the clusters can be determined by the application group associations. If all the nodes under a single application group falls under a single cluster, then the clustering algorithm accuracy will be considered 100 percent. The result of the final accuracy is analysed here [Table – 7].

TABLE VIIACCURACY ANALYSIS

Application Group	Node Number	Final Cluster	Selection Status
App – 1	Node - 33	CH-1	Correct
App – 1	Node - 35	CH-1	Correct
App – 1	Node - 36	CH-1	Correct
App – 2	Node - 20	CH-2	Correct
App – 2	Node - 22	CH-2	Correct
App – 2	Node - 23	CH-2	Correct
App – 2	Node - 25	CH-2	Correct

Thus, all the nodes under this simulated setup is clustered correctly showing 100% accuracy.

C. Density Based Region Building Results

The density-based region building is done using the proposed KDE-DRD algorithm. The proposed algorithm is motived by the KDE machine learning algorithm and the results are elaborated here.

Firstly, during the first iteration of the KDE-DRD algorithm the obtained result is documented here [Table – 8].

TABLE VIIIAFTER FIRST ITERATION OF THE KDE-DRD

Node Number	Selected Cluster
Node - 01	R1
Node - 02	R1
Node - 03	R1
Node - 04	R1
Node - 05	R1
Node - 06	R1
Node - 07	R1
Node - 08	R1
Node - 09	R1
Node - 10	R1
Node - 11	R1
Node - 12	R1
Node - 13	R1
Node - 14	R1
Node - 15	R1
Node - 16	R1
Node - 17	R1
Node - 18	R1

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Node - 19	R1
Node - 20	R1
Node - 21	R1
Node - 22	R1
Node - 23	R1
Node - 24	R1
Node - 25	R1

Secondly, during the second iteration of the KDE-DRD algorithm the obtained result is documented here [Table – 9].

TABLE IXAFTER SECOND ITERATION OF THE KDE-DRD

Node Number	Selected Cluster
Node - 26	R2
Node - 27	R2
Node - 28	R2
Node - 29	R2
Node - 30	R2
Node - 31	R2
Node - 32	R2
Node - 33	R2
Node - 34	R2
Node - 35	R3
Node - 36	R3
Node - 37	R3
Node - 38	R3
Node - 39	R3
Node - 40	R3
Node - 41	R3
Node - 42	R3
Node - 43	R3
Node - 44	R3
Node - 45	R3

Thirdly, during the second iteration of the KDE-DRD algorithm the obtained result is documented here [Table – 10].

TABLE XAFTER THIRD ITERATION OF THE KDE-DRD

Node Number	Selected Cluster
Node - 35	R3
Node - 36	R3
Node - 37	R3
Node - 38	R3
Node - 39	R3
Node - 40	R3
Node - 41	R3
Node - 42	R3
Node - 43	R3
Node - 44	R3
Node - 45	R3

Finally, the density analysis is carried out and the results are furnished here [Table – 11].

TABLE XIAFTER SECOND ITERATION OF THE KDE-DRD

Region Number	Number of Node (Density)
R1	25
R2	9
R3	11

The result is also analysed graphically here [Fig – 11].

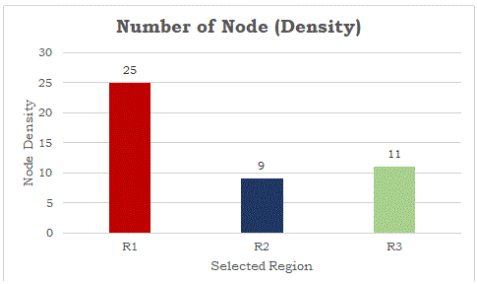

Fig. 11. Sample Node Placements at time t3

It is natural to understand that the region R1 is having the highest density of the nodes and selected by the algorithm to establish the node cooperative cache.

During the simulation, the following network map is generated [Fig – 12].

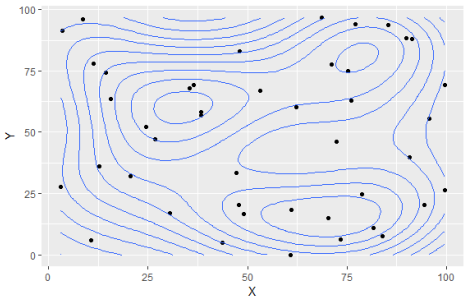

Fig. 12. Density Based Cache Region Identification

D. Cache Discovery Results

The proposed framework is expected to reduce the cooperative cache discovery and building time. Hence, the performance of the algorithm must be tested against the time complexity rather than the cache hit or the cache miss ratio.

Although, most of the parallel research outcomes have failed to demonstrate the applicability of cooperative caching strategies for all routing algorithms. Nevertheless, this work demonstrates a significant improvement of time complexity for all types of routing algorithms. The results are documented here [Table – 12].

TABLE XIICACHE DISCOVERY TIME ANALYSIS

Test Simulation	Cache Discovery Time with Proposed Framework (Sec)							Cache Discovery Time without Proposed Framework (Sec)						
	ZRP	DSDV	DSR	AODV	TORA	BGP	EIGRP	ZRP	DSDV	DSR	AODV	TORA	BGP	EIGRP

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Simulation – 1	0.01	0.009	0.009	0.009	0.009	0.01	0.008	0.03	0.05	0.02	0.03	0.01	0.01	0.05
Simulation – 2	0.011	0.01	0.011	0.01	0.01	0.009	0.009	0.05	0.02	0.04	0.01	0.01	0.03	0.01
Simulation – 3	0.01	0.011	0.009	0.008	0.009	0.007	0.009	0.02	0.01	0.05	0.03	0.03	0.01	0.04
Simulation – 4	0.01	0.01	0.01	0.01	0.008	0.009	0.008	0.04	0.01	0.04	0.05	0.03	0.05	0.02

The result is analysed graphically here [Fig – 13].

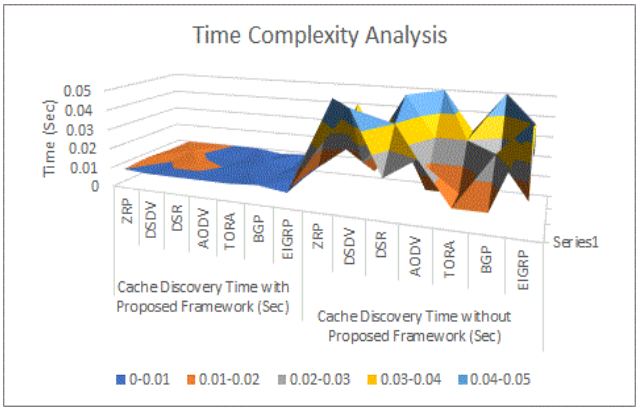

Fig. 13. Time Complexity Analysis for Routing Algorithms

Further the improvements over the time complexity is analysed here in terms of percentage of improvements [Table – 13].

Test Simulation	Cache Discovery Time Improvements (%)						
	ZRP	DSDV	DSR	AODV	TORA	BGP	EIGRP
Simulation – 1	66.67	82.00	55.00	70.00	10.00	0.00	84.00
Simulation – 2	78.00	50.00	72.50	0.00	0.00	70.00	10.00
Simulation – 3	50.00	-10.00	82.00	73.33	70.00	30.00	77.50
Simulation – 4	75.00	0.00	75.00	80.00	73.33	82.00	60.00

Hence, it is natural to understand that except in the case of DSDV algorithm during the simulation trial – 3, the performance of the proposed framework has helped the routing algorithms to improve the time complexity nearly 50%.Henceforth, after the analyse of the results obtained from the proposed framework, the work presents its conclusion in the next section.

IX. CONCLUSION

The cooperative caching method is highly popular among the MANET architects for building higher capacity networks with the increased possibilities to handle sophisticated applications. Often it is observed that the cache capabilities cannot be increased physically due to the cost factors associated with such improvements. Competing with significantly large number of parallel research outcomes, this work proposes a newer dimension of the research. Instead of proposing a cache building strategy, this work proposes a novel framework to improve the time complexity and reduce communication over-head. This work deploys a Fitness Function Based Dependency Clustering mechanism for reducing the search space for the data during discovery phase. Due to the reduction of the discovery phase search space reduction, the time complexity reduces significantly. Due to this enhancement, the work demonstrates 100% accuracy in the discovery phase. Also, in the cache building or management phase, this work deploys another novel Kernel Density Estimation Based Dense Region detection algorithm for selecting the cooperative caching region in order to reduce the time complexity for standard routing algorithms. The work demonstrates more than 50% improvement over time complexity for all types of standard routing algorithms with the aim to improve the architecture of MANET cache cooperative even better place to build higher capacity applications.

REFERENCES

1. Grossglauser et al. Mobility increases the capacity of ad-hoc wireless networks. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. 3. IEEE Proceedings. pp. 1360–1369. 2001.
2. P. Ranjan, K. K. Ahirwar, "Comparative study of VANET and MANET routing protocols", Proc. Int. Conf. Adv. Comput. Commun. Technol. (ACCT), pp. 517-523, 2011.
3. Steve Glass, Leveraging MANET based Cooperative Cache Discovery Techniques in VANETs: A Survey and Analysis, IEEE Communications Surveys & Tutorials (Volume: 19 , Issue: 4 , Fourthquarter 2017)
4. S. Lim, C. Yu, C. R. Das, "Cooperative cache invalidation strategies for Internet-based vehicular ad hoc networks", Proc. 18th Int. Conf. Comput. Commun. Netw. (ICCCN), pp. 1-6, 2009.
5. N. Kumar, J.-H. Lee, "Peer-to-peer cooperative caching for data dissemination in urban vehicular communications", IEEE Syst. J., vol. 8, no. 4, pp. 1136-1144, Dec. 2014.
6. R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, J.-P. Hubaux, "Hiding in the mobile crowd: Location privacy through collaboration", IEEE Trans. Depend. Secure Comput., vol. 11, no. 3, pp. 266-279, May/Jun. 2014.
7. M. S. Gast, 802.11 Wireless Networks: The Definitive Guide, Beijing, China:O'Reilly, Apr. 2002.
8. Internet Protocol, Sep. 1981, [online] Available: https://tools.ietf.org/html/rfc791.
9. N. E. Majd, S. Misra, R. Tourani, "Split-cache: A holistic caching framework for improved network performance in wireless ad hoc networks", Proc. IEEE Glob. Commun. Conf. (GLOBECOM), pp. 137-142, Dec. 2014.
10. M. F. Caetano, J. L. Bordim, "A cluster based collaborative cache approach for MANETs", Proc. 1st Int. Conf. Netw. Comput. (ICNC), pp. 104-111, Nov. 2010.
11. M. Meisel, V. Pappas, L. Zhang, "Ad hoc networking via named data", Proc. 5th ACM Int. Workshop Mobility Evol. Internet Archit., pp. 3-8, 2010.
12. M. Meisel, V. Pappas, L. Zhang, "Listen first broadcast later: Topology-agnostic forwarding under high dynamics", Proc. Annu. Conf. Int. Technol. Alliance Netw. Inf. Sci., pp. 1-8, 2010.
13. Narayana M V, Narsimha G, Sarma SSVN: Genetic - ZHLS Routing Protocol for Fault Tolerance and Load Balancing: Journal of Theoretical & Applied Information Technology; 2016.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)