Journal of Cleaner Production 217 (2019) 584–593

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities

S.K. Lakshmanaprabu [a, *], K. Shankar [b], S. Sheeba Rani [c], Enas Abdulhay [d], N. Arunkumar [e], Gustavo Ramirez [f], J. Uthayakumar [g]

[a] *B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India*
[b] *Kalasalingam Academy of Research and Education, Krishnankoil, India*
[c] *Sri Krishna College of Engineering, Coimbatore, India*
[d] *Biomedical Engineering Department, Jordan University of Science and Technology, Irbid, Jordan*
[e] *Department of Electronics and Instrumentation Engineering, SASTRA University, Tanjavur, India*
[f] *Department of Telematics, University of Cauca, Colombia*
[g] *Department of Computer Science, Pondicherry University, Puducherry, India*

ARTICLE INFO

Article history:
Received 19 October 2018
Received in revised form
5 December 2018
Accepted 11 January 2019
Available online 12 January 2019

Keywords:
VANET
Big data
Hadoop
Ant colony optimization
Smart cities

ABSTRACT

Rapid growth in urban population creates various kinds of issues like long hours traffic-jams, pollution which makes the city life insecure and non-liveable. The concept of a smart city is introduced to improve the quality of city life. Smart cities are being developed to satisfy the need for the safety of its users' and secure journeys over in the urban scenario by proposing the smart mobility concept. At the same time, Vehicular adhoc network (VANET) comes under the type of mobile adhoc network (MANET), wherever the vehicles are treated as nodes in a network. The application of Big Data technologies to VANET gains useful insight from the massive quantity of operational data to enhance traffic management process like planning, engineering as well as operation. During the real-time processes, the VANET generates large data, and the VANET characteristics are mapped to Big Data attributes. Moreover, ant colony optimization (ACO) algorithm is employed for routing in vehicular networks over Hadoop Map Reduce standalone distributed framework and over multi-node cluster with 2, 3, 4 and 5 nodes. The simulation outcomes ensure that the processing time of the algorithm is significant decreases with a rise in the node count of the Hadoop framework.

1. Introduction

A smart city is a development vision to combine information communication with an Internet of things (IoT) to handle a city's assets in a secure manner (Harteinstein and Labertaux, 2010). Presently, in smart cities, people look ahead for more than just vehicle quality and reliability. The next future for automotive revolution is expected by equipping automobiles with wireless communication capabilities. In the smart city environment, road safety had become a significant challenge for the government as well as the automobile industry in the last two decades. The recent growth in wireless technologies has attracted industries, academicians and professional to concentrate their work on enhancing road safety. In the past few years, the rise of wireless technologies enabled the researchers to design communication systems where vehicles contribute in the communication networks. Vehicular Ad hoc Network (VANET) comes under the subgroup of conventional Mobile Ad hoc Network (MANET). In VANET, the vehicles are treated as mobile nodes, which are integrated to "on-board" devices, traversing on constrained routes (i.e., roads as well as lanes), in addition to communication with every other node for information exchange through Vehicle-to-Vehicle (V2V) communication protocols, among vehicles and preset road-side units (RSU), against Vehicle-to-Infrastructure (V2I) communication (Harteinstein and Labertaux, 2010). An architecture of VANET in a smart city is shown in Fig. 1. In the coming generation, networked vehicles demonstrate the upcoming integration of computers,

* Corresponding author.
E-mail addresses: prabusk.leo@gmail.com (S.K. Lakshmanaprabu), shankarcrypto@gmail.com (K. Shankar), sheebaranis@skcet.ac.in (S. Sheeba Rani), ewabdulhay@just.edu.jo (E. Abdulhay), arun.nura@gmail.com (N. Arunkumar), gramirez@unicauca.edu.co (G. Ramirez), uthayresearchscholar@gmail.com (J. Uthayakumar).

https://doi.org/10.1016/j.jclepro.2019.01.115

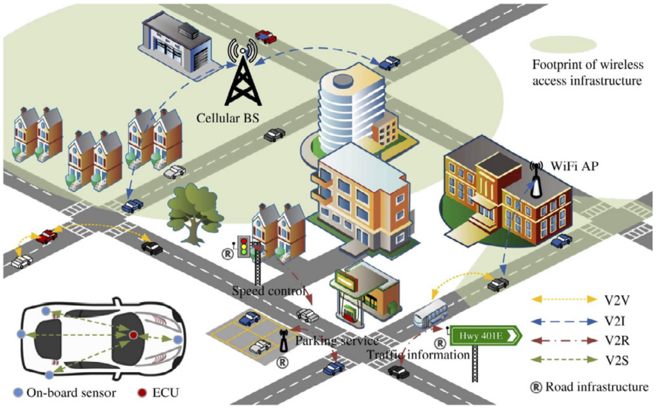


Fig. 1. The Architecture of VANET in smart city.

communication infrastructures, as well as vehicles (Lind et al., 1999). Vehicular communication is assumed as a controller for fully automated cars in the forthcoming days. Currently, there is a requirement to facilitate vehicular communication for applications like secure messaging, traffic as well as congestion monitoring as well as Internet access for various purposes. VANET is a kind of spontaneous adhoc network developed using vehicles moves on the roads. VANET is assumed as one of the well-known technology for enhancing the effectiveness as well as the safety of recent transportation systems. For example, vehicles exchange information with their neighboring vehicles to mitigate traffic jam close to the affected area.

The VANET application facilitates vehicle to join to the World Wide Web to get recent news, traffic, as well as climatic conditions. Applications like secure messaging are the near-space application, wherever vehicles closer to one other (specifically a sort of some meters), status information will be transmitted to raise safety alerts. The goal is to improve security by increasing the alert in crises. Applications for VANETs are mainly used for safety problems (For example, traffic services, alarm as well as warning messaging, audio/video streaming as well as comprehensive infotainment, to enhance the superiority of transportation by time-critical security as well as traffic management applications (Anna et al. Cusani). Simultaneously, amusement applications are also trending (For example, streaming of video-on-demand, Internet access by the passenger to take pleasure in the journey). Applications of alarm communication have strong delay limitations in the order of milliseconds, and also extremely high-reliability necessities. Contrastingly, applications like traffic as well as congestion monitoring require data gathering from vehicles presents around some kilometers. These applications are delay-tolerant, i.e., the delay constraints on data delivery are comparatively relaxed.

Internet access needs connectivity to the backbone network through RSU. The non-critical applications can generate many commercial opportunities through growing market dispersion of the technology as well as building it further cost effectual. However, comfort, as well as infotainment applications, intend to offer desirable information to road traveler as well as leisure to make the trip more enjoyable. Various kinds of application exist using conventional IP (For example, media streaming, voice over IP, browsing data, and so on) to particular applications to the vehicular environments (For example, advertisements, map downloading, halting payment, tollgate service, etc.). Additionally, intra, as well as inter-vehicle data communication, can be distinguished. The intra vehicle communication involves data transmission within a vehicle, and the inter-vehicle communication indicates the data transmission among vehicles, or vehicles and sensors, deployed in different places like roadways, signboards, halting areas, and so forth. Inter-vehicle communication is found to be difficult since vehicle communication needs to be supported even the vehicles are static as well as mobile. For example, the usage of prepaid or automatic billing system during the slow movement of vehicles in tollgates rather than stopping at tollgate utilizing a small electronic transmitter. Besides, cameras, as well as velocity sensors, are combined to compute the vehicle velocity. The quality of service offered in VANET is highly influenced by the vehicle movement and rapid topology changes. Various types of vehicles move around in VANET, based on traffic constraints, speed constraints in specific roadsides (i.e., national highways, street roads, city neighborhoods), and vehicle types (i.e., heavy-duty vehicles, four-wheeler, and two-wheeler). On comparing with the conventional mobile nodes in MANET (Gupta et al., 2018), vehicles in VANET roam around at a faster rate in the range of 0–40 m/s. Every unique characteristic of VANET allows fitting in the group of opportunistic networks, which indicates the change like the network and connection availability is unsatisfied. For instance, for maintaining the network connectivity in VANET, it is a generic method to link all vehicles moving on the roads in contrary directions using opportunistic connectivity links. (Agarwal et al., 2010). Besides, link breakages disrupt the stable and effective durable V2V communication, and the connection will be lost. At the same time, the constrained infrastructure coverage, due to the limited number of fixed access points leads to short and irregular V2I connectivity. The techniques based on both horizontal and vertical handover process are analyzed and present in the literature (Vegni et al., 2011a), (Vegni et al., 2011b), (Inzerilli et al., 2010). In summary, V2V communication offers various benefits: (i) allows short and medium range communication, (ii) low installation cost, (iii) short messages delivery, and (iv) reduces delay. However, V2V communication has the following limitations which are resolved with the combination of V2I, like (i) varying topology changes because of high mobility, (ii) issues in elongated range communications and (iii) conventional routing algorithms.

This paper tries to map the features of the VANET to the characteristics of big data to ensure that the VANET issues can be assumed as a Big Data problem, which can be handled by big data techniques and tools. In this paper, different features of VANET are mapped to the five attributes of big data namely Volume, Variety, Velocity, Value, and Veracity. Routing has forever been the main dispute in VANET because of the mobility as well as common modifications in the network topology. Moreover, the ant colony optimization (ACO) algorithm is also used for routing process in VANET.

The following part of the paper is arranged as follows: In section 2, the VANET is mapped as the big data problem. The proposed model is discussed in section 3, and the ACO routing algorithm is explained in section 4. In section 5, the outcomes of the simulation are discussed, and the paper is ended in section 6.

2. VANET as a big data problem

Big data is a fascinating model with the importance of converting the ranges of products in addition to services in industrial and business sectors. It is considered as a significant driving factor for the second economy (an idea presented by financial analyst W.B. Arthur that alludes to the financial exercises on processors, connectors, sensors, as well as executors) (Arthur, 2011). Big data is described as (gartner) “Big data is referred as high volume, high velocity, and high variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.” A detailed definition of big data include five *V*'s: (1) massive data quantity, (2) various kinds of data, (3) high velocity of generating and updating data, (4) veracity (uncertainty in addition to noise) of gathered information, as well as (5) significant value (Huang et al., 2015). The initial four *V*'s involved in information gathering, communication as well as storing purposes. The last *V* concentrates on the extraction of rates out of data utilizing statistical as well as analytical approaches (For example, machine learning algorithms, complex network theory). Big data methods are concerned to solve system level issues which can't be resolved through traditional approaches.

Fig. 2 illustrates the streamline of big data analysis (Jagadish et al., 2014). The first stage includes the acquisition and selection of the information needed to resolve the issue. In the next step, preprocessing of data is carried out to remove unwanted and retain essential data. This is especially vital for sensor gathered information which is noisy in nature, and it is needed to expel vulnerabilities from the sensor data. In the next stage, the process of integrating, aggregating and representing the data will be carried out. The fourth step explores new findings out of the processed information using statistical as well as analytical approaches. In the final phase, information will be illustrated in a diagrammatic way for the understanding of humans and to take decisions.

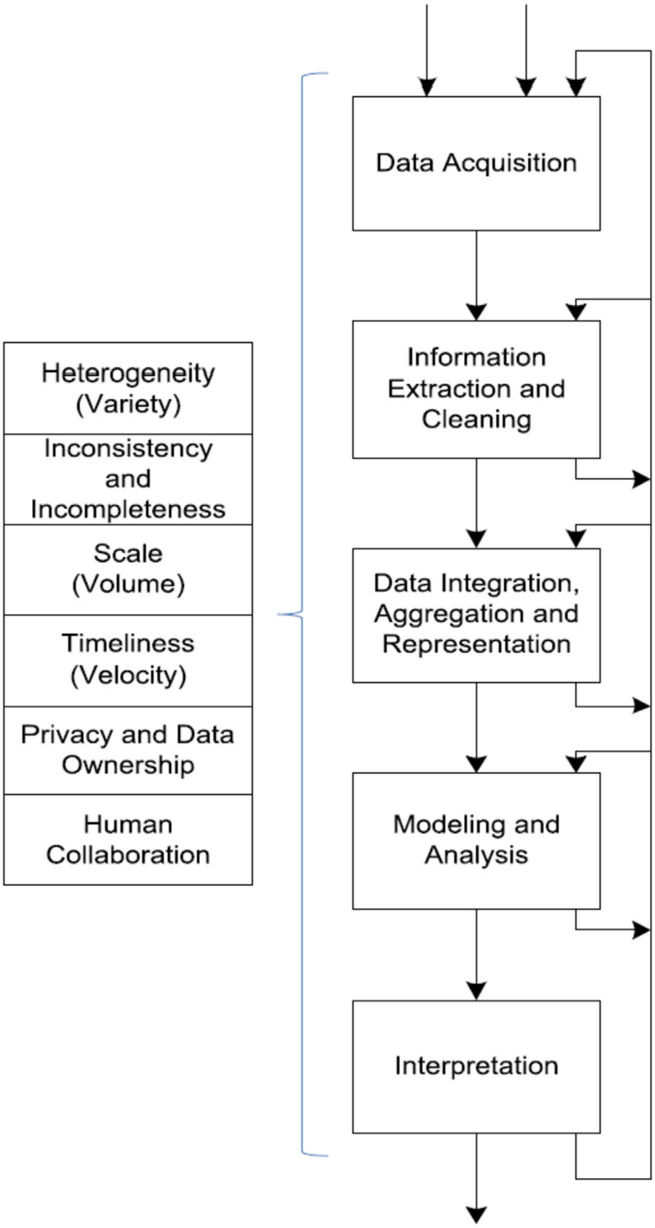

Heterogeneity (Variety)
Inconsistency and Incompleteness
Scale (Volume)
Timeliness (Velocity)
Privacy and Data Ownership
Human Collaboration
Data Acquisition
Information Extraction and Cleaning
Data Integration, Aggregation and Representation
Modeling and Analysis
Interpretation

Fig. 2. The pipeline of big data.

Several industrial sectors generate new data and digitizing previous data, and therefore they become the proper sources for Big Data. In recent days, every industry has begun to adopt a digital representation of data, leads to the massive quantity of data. Some of the different application areas of big data are the network, engineering, healthcare, transport, business, government, entertainment, tracking and monitoring services, etc.

Big Data assist organizations to facilitate their important clients by providing useful services based on the past information. It has resolved different problems and integrated issues such as storage, scalability, processing, timeliness, privacy as well as security. Recently, big data is being employed in the transportation area, i.e. VANET. The important issues in VANET exist in MANET also. To some degree of indiscriminate way, vehicles inclined to progress fixedly. The links can be portrayed clearly between RSU. The vehicles are constrained in the extent of movement and restricted to pursue a paved path. In this paper, we concentrate on VANETs which accesses the massive quantity of data in real-time to plan and manage safety as well as effective traffic flow in transportation systems. The state of art methods are available in (Al-Sultan et al., 2014), (Karim, 2008), (Mohapatra and Krishnamurthy, 2005), (Offor, 2012), (Qian and Moayeri, 2008), includes different features of VANET. This paper tries to map the features of the VANET to the attributes of big data to ensure which the VANET issues can be assumed as Big Data problem, which can be handled by big data techniques and tools. Different features of VANET are mapped to the five attributes of big data namely Volume, Variety, Velocity, Value, and Veracity are explained here.

- ***Real-time data:*** In VANET, the information will be generated in real time and gets updated after a predetermined time duration, saved in the database required for routing purposes (Karim, 2008). It is mapped to the Volume part of Big Data, indicating that the quantity of data created through various data sources contributed to the Big Data at present life.
- ***Variable vehicle count:*** It relies upon the exceptionally final variable number of vehicles. This vehicle is fitted by means of different GPS empowered sensor devices, which generates various forms of data. It is mapped to different portions of big data that indicates the data generated out of different sources, which varies from the type of data being delivered by them.
- ***Dynamic topology and Mobility modeling:*** In VANET, a node indicates a vehicle, which is highly mobile as well as dynamic in nature (Qian and Moayeri, 2008). It leads to quick topology modifications because of the outcome of rapid speed which is resulted from the frequent fragmentation based on effective diameter as well as network density (Al-Sultan et al., 2014), (Offor, 2012). This characteristic of VANET legitimizes the Velocity in Big Data, indicating that the data has been produced at a rapid rate.
- ***Vast Scale network and high computation capability:*** Several GPS authenticated devices add computation complexity of nodes by large-scale networks (Mohapatra and Krishnamurthy, 2005). It requires the correct rate of data to predict the routing decisions. Consequently, it is proportionate to *Value* division of Big Data that is significant to analyze as well as predict the nature of the nodes to formalize the policy or take decisions for the optimization of the output.
- ***Anonymous recipient and significant support from Infrastructure:*** Detection of vehicles in a specific territory out of the existing infrastructure is essential for most of the applications in VANET. In this manner, there is a prerequisite from the nearby

nodes to be straightforward in their estimation (Karim, 2008). It is mapped to the *Veracity* division of Big Data that identifies the data certainty and reliability based on data gathering, processing techniques, trusted infrastructure as well as the data source. This guarantees that data utilized is shielded out off unapproved access in addition to alterations all through the lifespan. Therefore, VANET issue can be considered as a case study for Big Data analytics.

In (Wu et al., 2018), a map-based relaying (MBR) algorithm in the MAC layer is developed to improve the results of the routing protocols in urban VANETs. For increased reliability of data transmission at intersections, MBR utilizes the digital map and the Geohash coding system to relay the packets (frames) in the MAC layer and selects the optimal relay node based on the relative position of the nodes and the intersection. Comparison of routing protocols to be used between vehicles are compared, and it is discussed which routing protocol can be used in different situations (YILDIRIM et al., 2018). In (ADARAMOLA, 2018), an investigation study is made to transmit data traffic efficiently in cities in the presence of various challenges. The selected metrics are the packet delivery ratio (PDR), end-to-end delay and routing overhead.

3. Proposed model

The examination in VANETs can be employed to enhance the transportation framework, provide chain management, in addition to logistics through estimating numerous parameters about the roadways, vehicles, as well as drivers. The collection of data needs sensors in the roadways as well as vehicles which learns the traffic level, climatic conditions, working of the vehicle, and reaction of drivers to the dynamic conditions. In a million miles of streets, billions of vehicles, as well as driver's data are, gathered in several years, the sheer count of data points is astonishing. It is just at this point, using modern Big Data technologies, i.e., a large amount of data can be investigated in shorter time duration for it to be valuable. Out of this massive selection of data, it is needed to construct adaptive methods to assist transportation organizations to decide on the selection of optimal routes with the best time to delivery, safety, cost, and fuel utilization. When calculating traffic conditions during poor climatic conditions, logistics organization can provide an optimal path which allows best routes for proper time delivery. There is a requirement of creating models that may foresee regardless of whether accidents are probably going to happen due to the nature of street as well as driver reaction, in the long run prompting more secure streets. This research has uncovered various advantages such as high safety, low transportation cost, user-friendly, and predictable material deliverance for producing, that is particularly critical for modern just-in-time manufacturing. An individual application can be employed to create or generate different kinds of data which is established quickly and has to process rapidly. The conventional relational databases cannot handle this massive quantity of data, and there is a requirement of efficient methods to manage it.

Hadoop is an open-source environment which processes the high volume of information (Barlow, 2013). At the point, when an organization requires saving of large measure of data, they have two choices: use a large machine by additional CPU, RAM, disk space and so forth otherwise discuss with database sellers for better solutions. But, each of the two choices has their issues, and there is a tradeoff over securing a big device. Besides, scaling is a significant process. The second choice infers scaling level. The result given by this choice is costly and needs more investment with specialized skills. Presently, “the data is continuously created by customers, different applications and/or devices but also machine generated and such data are exponentially leading the modification in the Big Data space. The process of dealing with the big dataset in the order of terabytes or even petabytes is difficult” (Barské, 2013). In Modern computing platform, a distributed data processing engine utilized for Big Data is Hadoop, execution of the Map Reduce framework. Hadoop provides reliability, sharing and distributed storage using the Hadoop Distributed File System (HDFS) in addition to distributed computing abilities using Map Reduce, a programming replica. The architecture of HDFS and Map reduce are given in Fig. 3 and Fig. 4.

Hadoop is an open-source framework which processes massive quantity of data to develop and execute different distributed applications. Hadoop has a distributed storage called HDFS in addition to distributed computing by a programming replica known as MapReduce (Warden, 2012), (Zikopoulos et al., 2012). Hadoop isn't an alternate option for databases or data warehouses, and it gives an outline for Big Data processing. In relational databases, the processing of organized data is substantially simple. This work is introduced by Apache Software Foundation in Java to help distributed large-scale applications. Hadoop gives distributed computing and distributed storage. Hadoop empowers the applications to work with the massive number of nodes as well as an enormous amount of data. It is utilized broadly at numerous business platforms like Yahoo search as its main benefactors. The two primary segments of Hadoop are HDFS as well as MapReduce.

3.1. Hadoop Distributed File System (HDFS)

An HDFS cluster operates in a master-worker pattern employing two kinds of nodes: a NameNode (master) and DataNodes

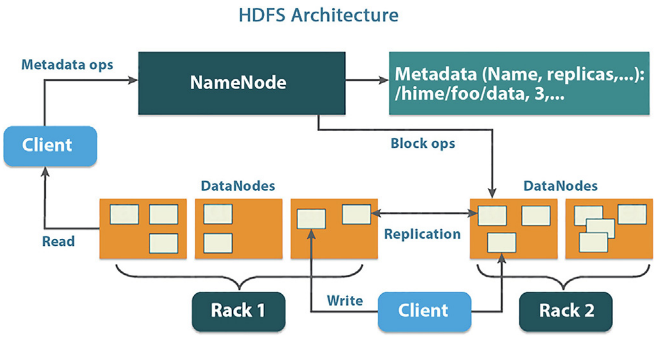


Fig. 3. Architecture of HDFS

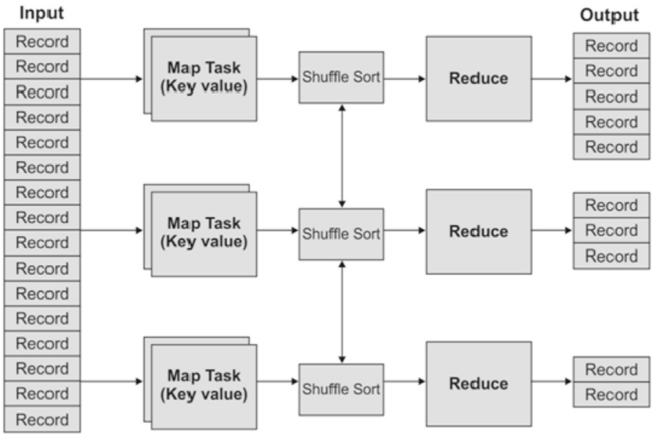


Fig. 4. The architecture of Map reduce.

(workers). File system namespace is overseen through the NameNode. It additionally states the file system tree and data about the considerable number of files as well as indexes in the tree. The DataNodes gather and also recover blocks while they are passed on to (Through customers otherwise the NameNode), as well as they pass it on the reverse to the NameNode occasionally with arrangements of blocks. NameNode chooses the replica of data blocks. The standard block size is 64 MB in a conventional HDFS through a replica factor of 3 (neighborhood rack has a second duplicate as well as remote rack has the third duplicate). The replica factor can be modified based on the client's need. HDFS has standard replication scheme in which every DataNode comprises at most one duplicate of any block as well as every rack comprise at most two duplications of a similar block, as there are adequate racks over the cluster. Every block contains sufficient count of repetitions and is guaranteed through the NameNode. It will ceaselessly ensure whether the block is over-replicated otherwise under-replicated. On the off chance that it is over-replicated, NameNode would choose a duplicate to leave. If it is under-replicated, NameNode will make other models in some other node or a similar node. The addition or deletion of duplicates takes place using the replication strategy. To pursue a file out of the HDFS, the customer initially link the NameNode as well as get the information, For instance, wherever the blocks of the necessary data and copies of each node residing. At that point, the customer's peruse blocks out off a client that is nearer to the customer. With a specific end goal to create a file, the customer requirements are linked to the NameNode to get the authorization for creation. For writing a file, the customer requires to make contact with the NameNode and access authorization for writing purposes. Then, the customer writes the data into whichever node as well as transfers it to block content to another node otherwise similar node to make replication. At the end of every operation, the DataNode transmits a heartbeat message to the NameNode representing which the DataNode is still active, as well as the block replicas, are accessible. When the NameNode doesn't acquire some message out of a DataNode in a specific period, subsequently the DataNode is assumed to be out of service besides replicas reside over the DataNode be copied into another node.

3.2. Map reduce

MapReduce is a linearly scalable programming model developed utilizing Google, which processes the massive amount of data using many computers simultaneously. It operates by two functions, map function (MF) as well as reduce function (RF). The former one takes a position of key/value input as well it maps into zero otherwise more set of key/value. The latter one takes every particular key as well as the cluster is integrated into a distinct key/value set (gartner). Map Reduce provide an easy way of use, scalability, as well as failure property.

MF: The MF is stated as (Dean and Ghemawat, 2008), “Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key (I) and passes them to the RF”. It is stated that,

$$\text{Map (k1,v1)} \rightarrow \text{list(k2,v2)} \tag{1}$$

In MF, the input keys and values undergo comparison with the intermittent keys to identify the similar intermediate keys and passed it to the RF.

RF: It is stated by the client (Lämmel, 2008), “The RF, also written by the user, gets an intermittent key (I) as well as a set of values for that key. It is merged to create a smaller set of ranges. The intermittent values are provided to the user's RF through an iterator. It allows us to manage the list of values which is huge to fit in storage space.” It can be represented as,

$$\text{Reduce(k2, list (v2))} \rightarrow \text{(k2,v3)} \tag{2}$$

The intermittent keys, as well as values, comprise the similar areas as the output keys as well as ranges. However, the actual input keys, as well as the ranges, comprise the variety of keys and ranges when contrasting it to the final output keys as well as ranges.

3.3. Execution overview of Hadoop

The Map Reduce library initially split the input files of Client program into M bits of ordinarily 16 MB–64 MB for each part. Presently, several duplicates of the program over a cluster of devices are taking place. Every function comes under the control of the master. The remaining nodes which are allocated job via the master are workers. Here, M map task, as well as R, reduce task to be assigned are taken. The idle workers are selected through the master, besides it allocates a task to everyone either a map or reduce task. The content of the relating input file is perused through a worker which is assigned a map task. It parses key/value matches from the input data, in addition, it sends every pair to the client characterized MF. The MF produces the mediatory key/esteem pairs and buffers it into storage. These buffered sets are well-known to neighborhood disks. The partitioning function will divide the area into R regions. The result of the Map is put away in the adjacent disc temporarily that is expelled when the RF is finished. These impermanent memories of Map help in the fault tolerant as well as recover process. The outcome of the Map is afterward conceded back to the master on the neighboring disc. The master is subjected to divert the areas to reduce workers.

After reaching a warning on these areas out of the master, the reduce worker read the buffered information out of the neighboring disks of the map workers employing remote procedure call. The reduce worker peruses the intermittent data for partitioning as well as it sorts it via the intermittent keys, so every event of a similar key is assembled. The arrangement is required as of commonly a wide range of keys are mapped to the same reduce task. On the off chance, that the measure of intermittent data is very substantial to fit in storage, an exterior range is required. The reduce worker repeats on the arranged intermittent data. At that point, the client's reduces work is known the key as well as the respective set of intermittent ranges. When there is an occurrence of a system malfunction, finished reducing tasks are not required to be repeated as its outcome is stored in a global system. In the case of reducing partition, the outcome of the reduce work is included to the last outcome file. The master empowers the client program behind finishing the entire map in addition to reduce tasks. The common Map Reduce framework is depicted in Fig. 4.

4. ACO routing

VANET aims to determine an optimal route among nodes, i.e., regulate the capability of the vehicles with the changeable topology of moving vehicles. Therefore, routing has forever been the main dispute in VANETs as of the mobility as well as common modifications in the network topology. To attain this, a routing mechanism is required which can be shared among two communication entities as well as it incorporates the process of a route organization, forwarding decision information as well as the action in managing the route and/or recover when the routing process fails.

Though several routing protocols exist, ACO algorithm offers several benefits such as Inherent parallelism, Positive Feedback accounts for rapid discovery of good solutions, Efficient for Traveling Salesman Problem and similar problems and it can be used in dynamic applications (adapts to changes such as new distances, etc.). So, ACO algorithm is preferred over other algorithms. ACO algorithm (Medina and Cortes, 2010a) follow the fundamental standard of procuring routing data by employing the path sampling process by sending control packets, known as ants. The ants are produced concurrently and autonomously at the nodes, with the method of testing a route from one node to the other node. The ant gathers data regarding the path eminence like latency, hop count, and so on, utilizes this information for updating the routing information at the mediatory nodes. Generally, the ant samples the entire path; so that the routing data gets modernized in a genuine in a pure Monte Carlo way with no rely on the bootstrapping data out of one node to the subsequent one. The routing table holds a vector of real value fields of every destination, every one for the well-known close by the node. These fields are used for the evaluation of goodness of traversing to the nearby node on the way to a particular destination node. The set of fields are pheromone variables, which is continuously gets updated based on the path quality ranges computed via the ants. The repetitive, as well as simultaneous creation of path-sampling ants, leads to the accessibility at every node on a set of paths. Every node has a set of paths with a computed assessment of quality.

Consequently, the ants utilize the routing tables which represent the path to its destination they sample: every node chooses the subsequent hop in a heuristic way, providing higher likelihood to links with more pheromone ranges. Hereafter, the routing tables will be known as pheromone tables. This procedure is almost same as the pheromone deposition and follows the nature of real ant colonies. The artificial ants are practically independent agents; besides, by the updation and stochastic pursuing of pheromone tables, it contributes in a stigmergic communication process. The outcome is a collective learning behavior, in that single ant has very low complication as well as less significant, whereas the entire swarm gathers and maintains up-to-date routing data.

The pheromone data is employed for the routing of data packets, probably in the similar means as routing ants packets stochastically, providing a high possibility of links with more pheromone ranges. Similarly, information for the same destination is broadcasted in many routers leads to proper load distribution. For data packets, some processes are employed to eliminate low-quality routes, since ants have great exploration, hence less good routes are sporadically sampled as well as managed. When adequate ants are passed to various destinations, nodes get updated information about the optimal paths and adapts with the distribution of load in an automatic manner. The route selection by the ACO algorithm is shown in Fig. 5.

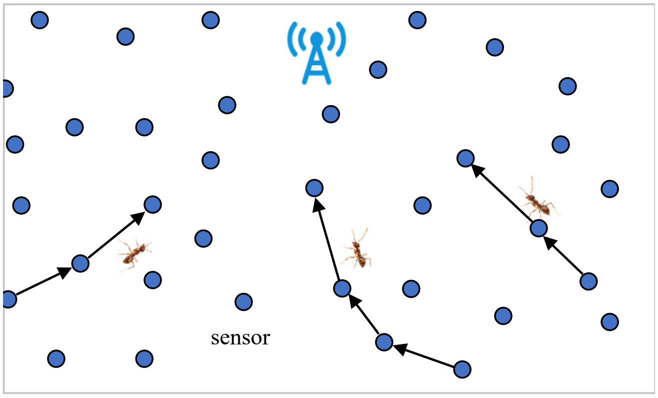


Fig. 5. Route selection by ACO algorithm.

ACO algorithm uses ant-like control packets to explore the routes between pair of nodes as well as to improve the available routing information. These ants act as a query packet which tries to set up all possible routes from the source towards the destination node. The algorithm considers that the VANET contains one destination and make use of two different types of ants, forward ants traverses out off origin towards destination, discovering novel routes as well as gather information, in addition to backward ants travels backside to the origin and destination to get the data is updated in every sensor as they move (Cheng et al., 2011), (Cobo et al., 2010), (Camilo et al., 2006), (Medina and Cortes, 2010b). This process will be carried out to lay pheromone to the traditional path. The pheromone values compute how ant starts at the source node, as well as bound for a destination node moves out of the one node to the subsequent node using multihop. Every forward ant has to choose next-hop node out off neighbor candidate list to set up its paths. The neighbor candidate list for every node contains a set of nodes deployed in wireless communication coverage of the node. The possibility of an ant moves out of a node i to other one j in the ACO routing algorithm is given as follows (Wang et al., 2008):

$$P_{ij}^{k}(t) = \frac{\left[\psi_{ij}(t)\right]^{\alpha} \times \left[\varepsilon_{ij}(t)\right]^{\beta}}{\sum_{S_l \in C(s_i)} \left[\psi_{il}(t)\right]^{\alpha} \times \left[\varepsilon_{il}(t)\right]^{\beta}} \tag{3}$$

where P_{ij}^{k} be the relocate packet likelihood of node i to other one j for ant k in time t, $\psi_{ij}(t)$ is the amount of pheromone laid over the route between i as well as j by ants in time t, $\varepsilon_{ij}(t)$ be the data of probing for which identified route, α and β are the two constant exponents incorporated with the algorithm. The location function, $\varepsilon_{ij}(t)$ presented through the conventional routing algorithm is represented as (Fadlullah et al., 2010):

$$\varepsilon_{ij} = \frac{1}{d_{ij}} \tag{4}$$

where d_{ij} is the Euclidean distance between node ito node j. When the ant identifies the destination, a path will be established from source to destination. Next, the destination node creates a response packet (backward ant) and it takes a reverse path to the source node and releases certain amount of pheromone. The pheromone $\psi_{ij}(t)$, gets updated at the terminated of every searching period:

$$\psi_{ij}(t+1) = (1-\rho) \times \psi_{ij}(t) + \Delta\psi_{ij}(t) \tag{5}$$

where ρ is the pheromone evaporation factor, as well as $\rho \in (0,1)$ with $\Delta\psi_{ij}(t)$, is the pheromone rise over the path from node i to node j, i.e., the amount of released pheromone by ant k in wireless association among node i in addition to node j:

$$\Delta\psi_{ij} = \sum_{k=1}^{n} \Delta\psi_{ij}^{k} \tag{6}$$

Where $\Delta\psi_{ij}^{k}$ in the classical algorithm, when the ant k select (i, j) is represented as below

$$\Delta\psi_{ij}^{k} = \frac{A}{L_k} \tag{7}$$

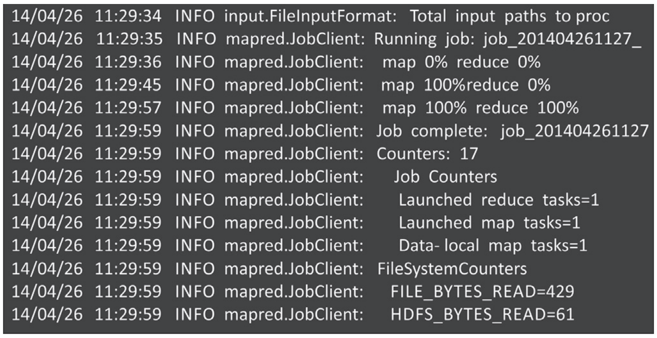
where L_k is the extent of path found through ant k and A is a constant. This algorithm will be executed until the algorithm exceeds a particular count of iterations for a particular number of ants.

5. Performance evaluation

The ACO algorithm is simulated in JAVA for individual impartial distributed Hadoop map-reduce framework. The parameter values used in the experiment are Number of ants: 50, evaporation rate: 0.1, pheromone deposition rate (Q): 02, pheromone factor: 1 and heuristics factor: 2. It is validated on the given count of nodes, and then also tested on multi-nodes which are entirely distributed Hadoop cluster environment utilizing 2, 3, 4 and 5 nodes correspondingly. In Map Reduce programming, the task is carried out through partitioning the whole program into smaller programs as well as independently executes them. Several data investigation techniques are concerned in the parallelization of MapReduce technique for easy utility and efficiency. A main feature of Hadoop is the process of dividing the data; besides, it performs computation among 1000 nodes as well as the implementation of application computations in parallel closer to its data. A Hadoop cluster scales computation capacity, storage capacity as well as I/O bandwidth through the inclusion of commodity servers (Lämmel, 2008). It can toutilize the available commodity of hardware, consequently, the computation cost to construct the Hadoop cluster be low. To execute a Map Reduce program, a small Hadoop cluster is required which might incorporate a master node as well as many worker nodes. The master node contains a JobTracker, TaskTracker NameNode as well as DataNode. A slave otherwise worker node behaves as a DataNode as well as TaskTracker, although it is probable to comprise data-only worker nodes as well as compute-only worker nodes" (Lind et al., 1999). MapReduce programming model involves a series of steps which are listed below (Lämmel, 2008):

- Iteration on the input
- Calculation of key/value pairs out off every piece of input
- Clustering entire intermittent ranges via key
- Iteration on the resultant group
- Reducing every class

The estimation involves a collection of input key/value pairs, as well as it produces a collection of output key/value pairs. Two functions, map in addition to reduce, are employed to describe the estimation. For implementing this, it is assumed that the starting node as key, distance as the range in Mapper part, existing node as key as well as lowest of every probable distance to this point as the rate in Reducer part for key/value pair. The time taken for every job in diverse cases with variant node count is tabulated in Table 1. It is reported that in NetBeans IDE 7.4 with 100 nodes, the total time of 1.05 s is needed to finish the job whereas the Hadoop separate node consumes 1.2 s to complete a similar task. Furthermore, Hadoop distributed system with 2, 3, 4, 5 nodes take1.15, 1, 0.95 and 0.75 s respectively for the execution of the job. Fig. 6 shows the snapshot on the output of Hadoop map reduce framework.



```
14/04/26  11:29:34  INFO  input.FileInputFormat:  Total  input  paths  to proc
14/04/26  11:29:35  INFO  mapred.JobClient:  Running  job:  job_201404261127_
14/04/26  11:29:36  INFO  mapred.JobClient:    map  0%  reduce  0%
14/04/26  11:29:45  INFO  mapred.JobClient:    map  100%reduce  0%
14/04/26  11:29:57  INFO  mapred.JobClient:    map  100%  reduce  100%
14/04/26  11:29:59  INFO  mapred.JobClient:  Job  complete:   job_201404261127
14/04/26  11:29:59  INFO  mapred.JobClient:  Counters:  17
14/04/26  11:29:59  INFO  mapred.JobClient:     Job  Counters
14/04/26  11:29:59  INFO  mapred.JobClient:      Launched  reduce  tasks=1
14/04/26  11:29:59  INFO  mapred.JobClient:      Launched  map  tasks=1
14/04/26  11:29:59  INFO  mapred.JobClient:      Data- local  map  tasks=1
14/04/26  11:29:59  INFO  mapred.JobClient:  FileSystemCounters
14/04/26  11:29:59  INFO  mapred.JobClient:     FILE_BYTES_READ=429
14/04/26  11:29:59  INFO  mapred.JobClient:     HDFS_BYTES_READ=61
```

Fig. 6. Sample output of Hadoop map-reduce framework.

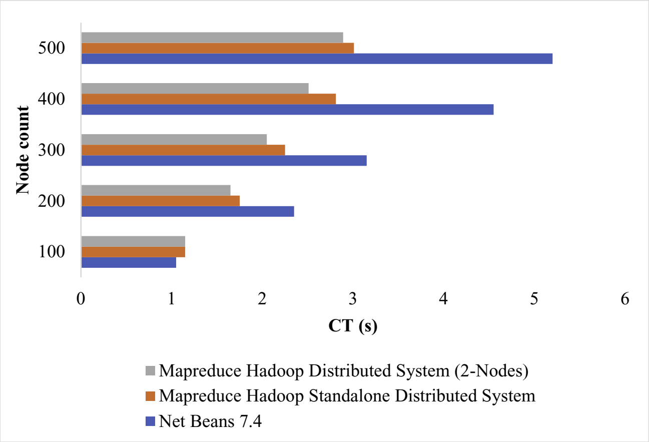


Fig. 7. Comparison of CT (in s) by ACO algorithm under 2 nodes.

Table 1
Comparison of CT (in s) by ACO algorithm under 2 nodes.

Node count	Net Beans 7.4	MapReduce Hadoop Standalone Distributed System	MapReduce Hadoop Distributed System (2-Nodes)
100	1.05	1.15	1.15
200	2.35	1.75	1.65
300	3.15	2.25	2.05
400	4.55	2.81	2.51
500	5.2	3.01	2.89

The time taken for every job in diverse cases with variant node count under Map reduces Hadoop distributed system of 2 nodes is tabulated in Table 1 and shown in Fig. 7. It is reported that in Net Beans IDE 7.4 with 100 nodes, the total time of 1.05 s is needed to finish the job whereas the Hadoop separate node consumes 1.15 s to complete a similar task. Furthermore, Hadoop distributed system with 2 nodes needed 1.15 s for the execution of the job. Likewise, for 200 nodes, the Net Beans 7.4 takes a total of 2.35 to complete the task whereas the Map Reduce Standalone distributed system takes only 1.75s to execute the same task. But, the MapReduce Hadoop Distributed System needs only a minimum of 1.65s which is lower than the other ones. In the same way, for the increasing node count, the CT taken to complete the task is gradually reduced. For the node count of 500, the CT taken by the MapReduce Hadoop Distributed System with 2 nodes 2.89s which is much lower than the CT needed by the Net Beans 7.4 and Map Reduce Hadoop Standalone Distributed System.

Fig. 8 and Table 2 shows the comparative results of CT under MapReduce Hadoop Distributed System of 3 nodes. It is found that Beans IDE 7.4 with 100 nodes needs the total time of 1.05 s is needed to finish the job whereas the Hadoop separate node consumes 1.15 s to complete a similar task. In addition, Hadoop distributed system with 3 nodes needed 1.00 s for the execution of the job. Likewise, for 200 nodes, the MapReduce Hadoop Distributed System needs only a minimum of 1.55s which is lower than the other ones. At the same time, under the node count of 500, the CT taken by the MapReduce Hadoop Distributed System with 3 nodes 2.56s which is much lower than the CT needed by the Net Beans 7.4 and Map Reduce Hadoop Standalone Distributed System.

The time taken for every job in diverse cases with variant node count under Map reduces Hadoop distributed system of 4 nodes is tabulated in Table 3 and Fig. 9. It is reported that the Hadoop distributed system with 4 nodes needed 0.95 s for the execution of

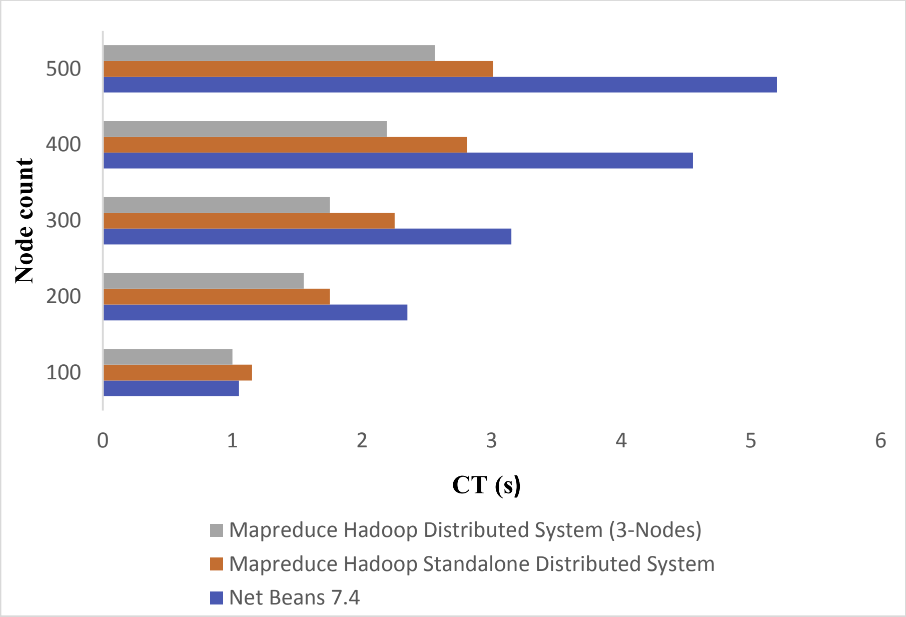


Fig. 8. Comparison of CT (in s) by ACO algorithm under 3 nodes.

Table 2
Comparison of CT (in s) by ACO algorithm under 2 nodes.

Node count	Net Beans 7.4	MapReduce Hadoop Standalone Distributed System	MapReduce Hadoop Distributed System (3-Nodes)
100	1.05	1.15	1.00
200	2.35	1.75	1.55
300	3.15	2.25	1.75
400	4.55	2.81	2.19
500	5.2	3.01	2.56

Table 3
Comparison of CT (in s) by ACO algorithm under 4 nodes.

Node count	Net Beans 7.4	MapReduce Hadoop Standalone Distributed System	MapReduce Hadoop Distributed System (4-Nodes)
100	1.05	1.15	0.95
200	2.35	1.75	1.35
300	3.15	2.25	1.65
400	4.55	2.81	2.07
500	5.2	3.01	2.45

the job. Likewise, for 200 and 300 nodes, the MapReduce Hadoop Distributed System needs only a minimum of 1.35s and 1.65s which is lower than the other ones. In the same way, for the node count of 400 and 500 nodes, the CT taken by the MapReduce Hadoop Distributed System with 4 nodes is 2.07 and 2.45s respectively.

Fig. 10 and Table 4 provides the results attained by CT under MapReduce Hadoop Distributed System of 5 nodes. It is found that Hadoop distributed system with 5 nodes needed 0.75 s for the execution of the job. Likewise, for 200 nodes, the MapReduce Hadoop Distributed System needs only a minimum of 1.05s which is lower than the other ones. At the same time, under the node count of 300, 400 and 500, the CT taken by the MapReduce Hadoop Distributed System are 1.45s, 1.80s and 2.14s respectively.

An overall comparison of CT with varying number of nodes with the node count of 100, 200, 300, 400 and 500 are shown in Table 5 and Fig. 11. From the Fig., it is clearly shown that the CT is significantly reduced with the increased number of nodes. For instance, under the node count of 100, the attained CT is 1.15 and 0.75s respectively. These values ensure that the reduction in CT with increased number of nodes.

Next, the throughput analysis of the proposed model is made and is shown in Fig. 12. The throughput refers to the total number of packets reached at the destination to the total number of packets transmitted in the network. The higher value of throughput indicates the better performance. From the Fig., it is clearly shown that the throughput tends to increase with increasing number of rounds.

From the above figures and tables, it is revealed that the increase in the node count of the graph decreases the processing period of the task in Hadoop platform because of the distributed processing of the information on various nodes as well as the disparity in processing time rises by means of conventional Java circumstances as the count of nodes in Hadoop platform increases.

6. Conclusion

Developing particular technologies for Smart cities serves as a platform for addressing and discussing theoretical and practical cleaner production, encompassing environmental, and sustainability issues. In recent days, various applications have been developed to incorporate it into the smart city environment to take the sophistication and safety in a city to the next level. At the same time, big data has grasped many researches in several domains since a massive amount of information is being created through the different day to day applications. The attributes linked to big data are Volume, Variety, Velocity, Veracity as well as Value. This paper has discussed and maps some of the VANET distinctiveness to the attributes of big data attributes to solve the VANET problem by considering it as a big data problem. This paper uses the ACO algorithm on Big Data framework by the use of independent distributed Hadoop Map Reduce platform, also, to Hadoop distributed platform with 2, 3, 4 and 5nodes is also validated. The experimental values obtained by the proposed method undergone comparison with the traditional NetBeans IDE environment

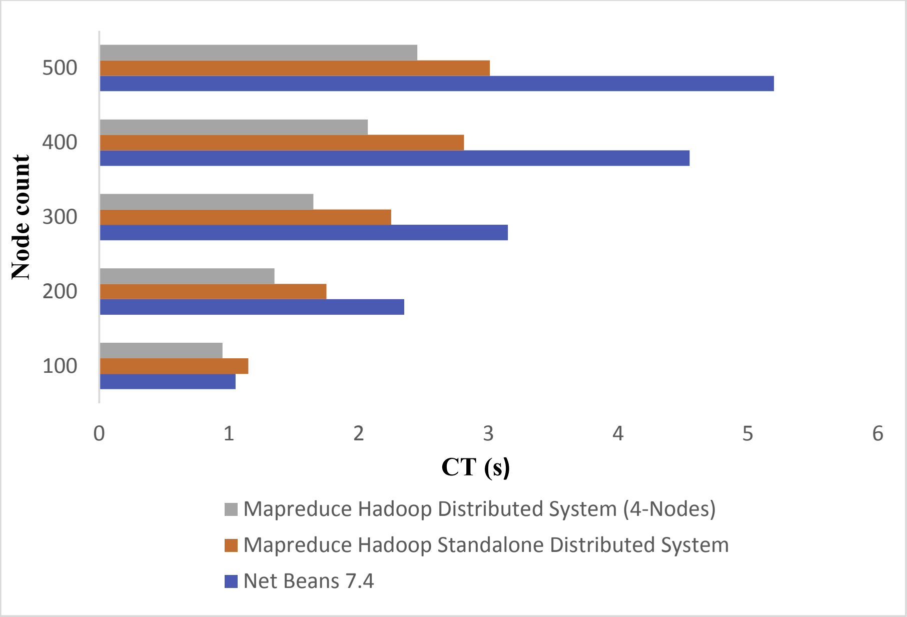


Fig. 9. Comparison of CT (in s) by ACO algorithm under 4 nodes.

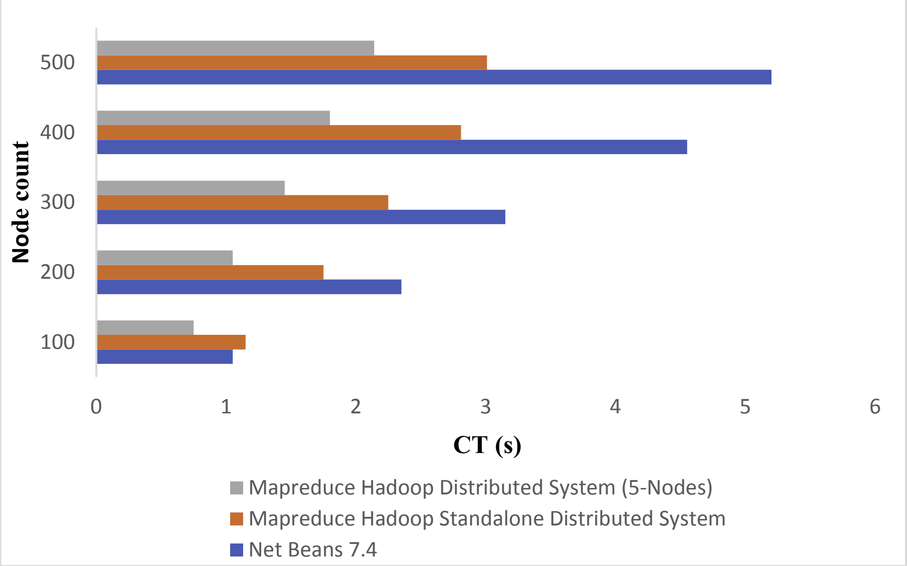


Fig. 10. Comparison of CT (in s) by ACO algorithm under 5 nodes.

Table 4
Comparison of CT (in s) by ACO algorithm under 5 nodes.

Node count	Net Beans 7.4	MapReduce Hadoop Standalone Distributed System	MapReduce Hadoop Distributed System (5 Nodes)
100	1.05	1.15	0.75
200	2.35	1.75	1.05
300	3.15	2.25	1.45
400	4.55	2.81	1.80
500	5.2	3.01	2.14

Table 5
Comparison of CT (in s) by ACO algorithm under varying number of nodes.

Node count	2-Nodes	3-Nodes	4-Nodes	5-Nodes
100	1.15	1	0.95	0.75
200	1.65	1.55	1.35	1.05
300	2.05	1.75	1.65	1.45
400	2.51	2.19	2.07	1.8
500	2.89	2.56	2.45	2.14

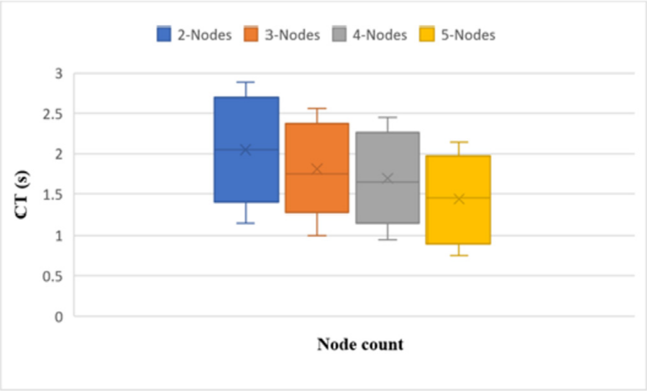


Fig. 11. Overall comparison of CT in seconds under varying number of nodes.

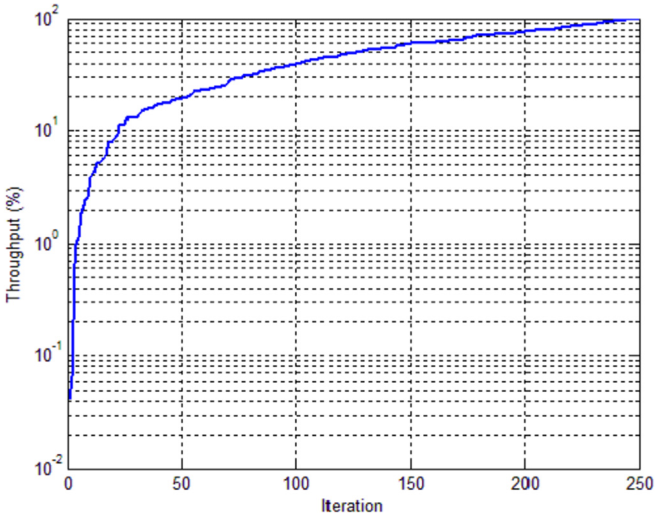


Fig. 12. Throughput analysis.

concerning increased count of nodes in the system. From the results, it is shown that the NetBeans IDE 7.4 with 100 nodes, the total time of 1.05 s is needed to finish the job whereas the Hadoop separate node consumes 1.2 s to complete the similar task. Furthermore, Hadoop distributed system with 2, 3, 4, 5 nodes take1.15, 1, 0.95 and 0.75 s respectively for the execution of the job. From the results, it is revealed which the increase in the count of cluster nodes utilizing Hadoop framework significantly minimizes the processing time of the routing algorithm. In the future, the proposed method can be enhanced by the hybridization of the ACO algorithm with any other metaheuristic algorithms.

References

Adaramola, O.J., 2018. Network parameters evaluation in vehicular Ad-hoc network (VANET) routing protocols for efficient message delivery in city environment. J. Adv. Computer Eng. Tech. 4 (3), 41–50.

Agarwal, A., Little, T.D.C., 2010. Opportunistic networking in delay tolerant vehicular Ad hoc networks. In: Watfa, M. (Ed.), Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, pp. 282–300.

Al-Sultan, S., Al-Doori, M.M., Al-Bayatti, A.H., Zedan, H., 2014. A comprehensive survey on vehicular Ad Hoc network. J. Netw. Comput. Appl. 37, 380–392.

Anna Maria, Mauro Biagi, Roberto Cusani. "Chapter 1 Smart Vehicles, Technologies and Main Applications in Vehicular Ad Hoc.

Arthur, W.B., Oct. 2011. The Second Economy. McKinsey Q.

Barlow, M., 2013. Real-Time Big Data Analytics: EmergingArchitecture. O'Reilly, Sebastopol.

Barské, A., 2013. Big Data Whitepaper,. Erdogan: ArzuBarské - Erdogan 2013 ©, version 2.0.

Camilo, T., Carreto, C., Silva, J.S., Boavida, F., 2006. An energy-efficient ant-based routing algorithm for wireless sensor networks. In: ANTS 2006, LNCS-4150. Springer, Heidelberg, pp. 49–59.

Cheng, D., Xun, Y., Zhou, T., Li, W., 2011. An energy aware ant colony routing algorithms for the routing of wireless sensor networks. In: ICICIS 2011, Part I, CCIS-134. Springer, Heidelberg, pp. 395–401.

Cobo, L., Quintero, A., Pierre, S., 2010. Ant-based routing for wireless multimedia sensor networks using multiple QoS. Comput. Network. 54, 2991–3010.

Dean, J., Ghemawat, S., 2008. MapReduce : simplified data processing on large clusters. Commun. ACM 51 (1), 1–13.

Fadlullah, Z.M., Taleb, T., Vasilakos, A.V., Guizani, M., Kato, N., 2010. DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis. IEEE/ACM Trans. Netw. 18 (4), 1234–1247.

Gupta, Deepak, Khanna, Ashish, Lakshmanaprabu, S.K., Shankar, K., Furtado, Vasco, Joel, J., Rodrigues, P.C., November 2018. Efficient artificial fish swarm based clustering Approach on mobility aware energy-efficient for MANET. Transactions on Emerging Telecommunications Technologies e3524. https://doi.org/10.1002/ett.3524.

Harteinstein, H., Labertaux, K.P. (Eds.), 2010. VANET Vehicular Applications and Inter-networking Technologies. John Wiley & Sons, Ltd.. Mar.

http://www.gartner.com/it-glossary/big-data.

Huang, T., Lan, L., Fang, X., An, P., Min, J., Wang, F., 2015. Promises and challenges of big data computing in health sciences. Big Data Res 2 (1), 2–11.

Inzerilli, T., Vegni, A.M., Neri, A., Cusani, R., Cross-layer, A., 2010. Location-based Approach for mobile-controlled connectivity. Int. Journal of Digital Multimedia Broadcasting. Hindawi Publ. Corp., Article ID 597105, 13 pages, doi:(2010).

Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., Shahabi, C., 2014. Big data and its technical challenges, Commun. ACM 57 (7), 86–94.

Karim, R., 2008. VANET: Superior System for Content Distribution in Vehicular Network Applications. Rutgers University, Department of Computer Science. Tech. Rep.

Lämmel, R., 2008. Google's MapReduce programming model—Revisited. Sci. Comput. Program. 70 (1), 1–30.

Lind, R., et al., 1999. The Network Vehicle-a glimpse into the future of mobile multimedia. IEEE Aerospace and Electronic Systems Magazine 14 (9), 27–32. Sept.

Medina, C.D., Cortes, N.C., 2010. Routing algorithms for wireless sensor networks using ant colony optimization. In: MICAI 2010, Part II, LNAL-6438. Springer, Heidelberg, pp. 337–348.

Medina, C.D., Cortes, N.C., 2010. Routing algorithms for wireless sensor networks using ant colony optimization. In: MICAI 2010, Part II, LNAL-6438. Springer, Heidelberg, pp. 337–348.

Mohapatra, P., Krishnamurthy, S.V., 2005. AD HOC NETWORKS : Technologies and Protocols. Springer, Boston, USA.

Offor, P.I., 2012. Vehicle Ad hoc network (VANET): safety benefits and security challenges. Social Sci. Res. Network. https://doi.org/10.2139/ssrn.2206077.

Qian, Y., Moayeri, N., 2008. In: Design of Secure and Application-Oriented VANETs. Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, Singapore, pp. 2794–2799.

Vegni, A.M., Cusani, R., 2011. Connectivity support in heterogeneous wireless networks. In: Lin, Jia-Chin (Ed.), Recent Advances in Wireless Communications and Networks. INTECH Pub, August, 978-9-53307-274-6.

Vegni, A.M., Inzerilli, T., Cusani, R., 2011. Seamless connectivity techniques in vehicular Ad-hoc networks. In: Almeida, Miguel (Ed.), Advances in Vehicular Networking Technologies. INTECH Pub, April, 978-979-53307-241-8.

Wang, X., Li, Q., Xiong, N., Pan, Y., 2008. Ant colony optimization-based location-aware routing for wireless sensor networks. In: Li, Y., Huynh, D.T., Das, S.K., Du, D.Z. (Eds.), WASA 2008, vol. 5258. LNCS Heidelberg: Springer, pp. 109–120.

Warden, P., 2012. Big Data Glossary. Sebastopol: O'Reilly Media, Inc.

Wu, J., Lu, H., Xiang, Y., Wu, R., Wang, F., 2018. MBR: a map-based relaying algorithm for reliable data transmission through intersection in VANETs. IEEE Trans. Intell. Transport. Syst. https://doi.org/10.1109/TITS.2018.2877993.

Yildirim, A.K., Boyaci, A., Qaraqe, M., 2018. A comparision of vanet routing protocols performance for emergency situations. J. Techn. Appl. Sci. 1 (1), 1–10.

Zikopoulos, P.C., Eaton, C., deRoos, D., Deutsch, T., Lapis, G., 2012. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill, United States of America.