

DOI: 10.20535/kpissn.2020.4.226993

UDC 004.415.2

L.M. Oleshchenko^{1*}, K.O. Movchan², V.I. Petrus¹

¹Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

²Ukrainian Scientific and Research Institute of Special Equipment and Forensic Expertise of the Security Service of Ukraine, Kyiv, Ukraine

*corresponding author: oleshchenkoliubov@gmail.com

DATA TRANSMISSION METHOD BASED ON MULTIPLEXING AND TAKING INTO ACCOUNT CHANGE OF NODES RATIO IN MANET NETWORKS

Background. The task of building a network with a continuous mode of data transmission for mobile nodes is to build a wireless network and maintain a state of loss of the minimum number of packets during the exchange of information. One of the main problems of MANET (Mobile Ad hoc Networks) network management is their dynamics and the fact that mobile devices are limited in their computing capabilities. There is a problem of traffic routing within MANET networks and the internal hierarchy of transmitters involved in the system. These networks also have problems with bandwidth optimization and power management. Lack of fixed topology introduces new scientific problems, such as network configuration, device search and topology support, as well as special addressing and self-destruction of the network. This task is relevant for data transmission between mobile devices in terms of reconnaissance operations.

Objective. The goal of the article is the research of reducing packet delays in nodes during dynamic routing, reducing service traffic and software optimization of data transmission in MANET relative to existing methods.

Methods. To achieve this goal, we use computer simulation, statistical and empirical methods. Based on the optimization of dynamic routing using the Ad hoc On-Demand Distance Vector algorithm, four main indicators are identified: speed of path establishment, message delay in one node, occurrence of loops, the amount of service traffic. To test the proposed method, a graph of twelve vertices simulating a wireless peer-to-peer MANET network is simulated. Testing is performed in more than a hundred iterations. At each iteration, the vertices of the network graph are randomly selected and the data transfer between each vertex of the graph is modeled.

Results. The use of the created software provides the analysis of the indicators of the quality of continuous data transmission and network routing by the proposed method. The proposed method shows reducing the amount of office traffic by 15 % and the best results by 5–10 % reduction of delay time within the node and when processing messages.

Conclusions. A method is proposed to reduce the delay time within a node that changes its position when processing a message, to reduce the amount of service traffic when transmitting data between nodes in the MANET network. Computer simulation is used to study the proposed method.

Keywords: wireless networks; dynamic routing; peer-to-peer; ad-hoc networks; MANET; Ad hoc On-Demand Distance Vector algorithm.

Introduction

Today we can see an increase in the amount of information generated by humans and the number of mobile devices being developed, including quadcopters, mobile computers with a wireless network interface and, consequently, the information that needs to be transmitted between these devices, generated in real time, and needs processing. MANET (Mobile Ad-hoc network) does not contain any infrastructure and every device is connected wirelessly. Ad hoc networking would allow the military to maintain an information network between the soldiers, vehicles and military information headquarters. At the time of disaster, there is need of wireless network. The places where wired network may be affected by the disasters, MANET can be implemented [1, 2]. In some cases, mobile computers

equipped with wireless interfaces are used to build this type of network, namely, combining several mobile devices with wireless network adapters into one data network. This type of mobile devices may belong to different persons, i. e. there is no single centralized administrative management. In addition, the nodes of such networks must interact correctly with each other at the organization level as a whole. Such characteristics are possessed by networks for fast deployment, namely self-organized networks. Also, such networks quickly adapt to the conditions of the variable characteristic of network bandwidth.

Problem statement

One of the fundamental problems of MANET network management is their dynamics and the fact that mobile devices are limited in number and in

their computing power [3]. The goal of the article is the research of reducing packet delays in nodes during dynamic routing, reducing service traffic and software optimization of data transmission in MANET relative to existing methods.

Analysis of existing methods

All devices in MANET are self-configured. Every node in this network are mobile and can change to other devices frequently (Fig. 1). Ad-hoc networks are different from other networks because they have flat infrastructure, work on shared medium (for example, radio), have dynamic topology. In this network every computer device acts like a router as well as end hosts.

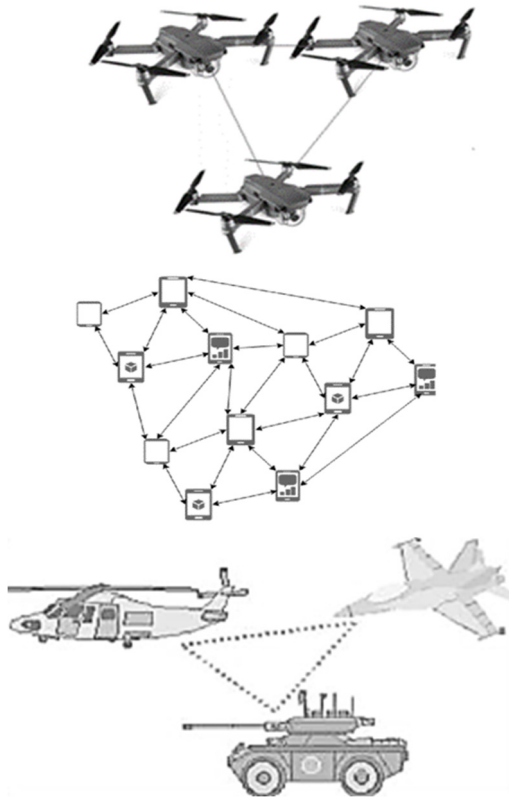


Fig. 1. Mobile Ad-hoc networks [1]

Hardware constraints are a rather influential lever that various tools try to circumvent, but in general the solution requires newer and more optimized algorithms and protocols designed to work with the network. There is also the problem of routing traffic within networks and the internal hierarchy of transmitters involved in the system. The system itself, in turn, with the growing number of nodes may have a problem routing system traffic and balancing the

network load. These networks also have problems with bandwidth optimization and power management [4]. In addition, the lack of a fixed infrastructure introduces new scientific problems, such as network configuration, device search and topology support, as well as special network addressing and self-destruct.

Proactive protocols determine routes independent of traffic pattern. Traditional link-state and distance-vector routing protocols are proactive. Examples: DSDV (Dynamic sequenced distance-vector), OLSR (Optimized Link State Routing). Reactive protocols maintain routes only if needed. Examples: DSR (Dynamic source routing), AODV (on-demand distance vector). Hybrid protocols. Example: Zone Routing Protocol (intra-zone: proactive; inter-zone: on-demand) (Fig. 2).

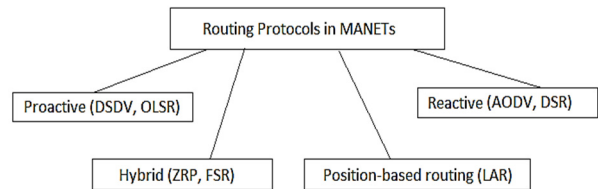


Fig. 2. Routing Protocols in MANETs

Position-based routing methods use information on the exact locations of the nodes. This information is obtained for example via a GPS receiver. Based on the exact location the best path between source and destination nodes can be determined. Example: Location-Aided Routing in mobile ad hoc networks (LAR) [3].

AODV is an on-demand, single path, loop-free distance vector protocol. It combines the on-demand route discovery mechanism with the concept of destination sequence numbers. AODV takes a hop-by-hop routing approach. Proactive protocols may have lower latency since routes are maintained at all times. Reactive protocols may have higher latency because a route from X to Y may be found only when X attempts to send to Y . Reactive protocols may have lower overhead since routes are determined only if needed. Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating. Which approach achieves a better trade-off depends on the traffic and mobility patterns.

When one node needs to send a message to another node for which it does not already have a route it broadcasts a Route Request (RREQ) message. By sending RREQ the node is asking the network how to get to the destination (Fig. 3).

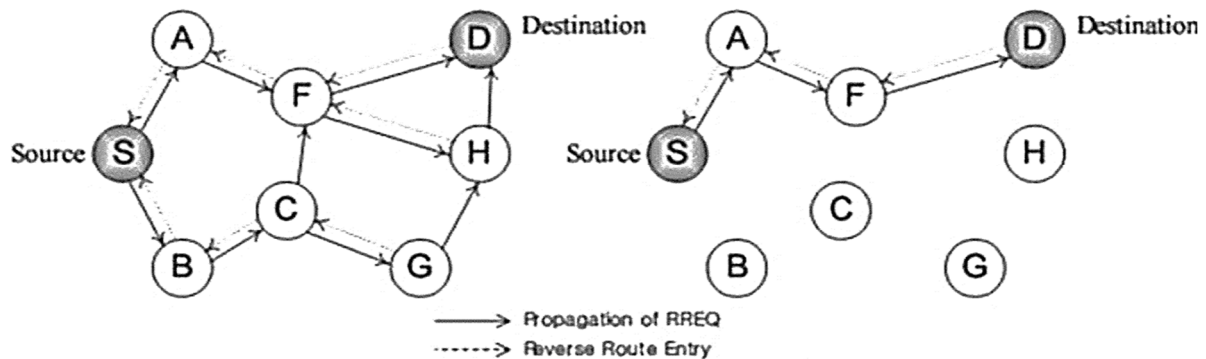


Fig. 3. Route request packet (RREQ) [5]

AODV protocol is designed for dynamic, rapidly changing routing of network agents for mobile ad-hoc wireless self-organizing networks. Support as unicast, unidirectional data transmission to a single destination, or packet routing in a data network that does not support multicast packet messaging is a partial case of multicast, that is the recipient of a message. This exchange is supported at both the channel and application levels. The implementation of this algorithm has several characteristics:

- distribution;
- iteration;
- asynchrony.

Each node maintains an internal routing table with information data for each router or network agent. In AODV by using sequence numbers when updating the route, the possibility of the problem of “infinity accounts”, inherent in other routing protocols, is excluded. Each node polls each subsequent node and asks each subsequent node to see if there is a node next to them.

The communication protocol is based on RREQ message packets consisting of the following fields:

- type (8 bits) “1”;
- the “J” flag is set if RREQ is broadcast;
- the “R” flag is set when restoring the route;
- the flag “U” is set if the sequence number of the receiving node is unknown;
- 11 bits reserved;
- destination number of the receiving node (Destination Sequence Number);
- Originator IP address;
- sequence number of the initiator node (Originator Sequence Number);
- the flag “G” is set if it is necessary to set not only the route to the receiving node, but also the return route from the receiving node to the initiating node;

- Hop Count 8 bits;
- RREQ ID (RREQ message identification number);
- Destination IP Address;
- The “D” flag is set if the RREQ message needs to be delivered directly to the receiving node.

In the *Open Shortest Path First (OSPF)* protocol, the main idea is based on the capabilities of dynamic networks, namely channel tracking, the so-called link-state process, which encapsulates the common solution of finding the shortest path based on the Dijkstra algorithm, from one vertices to all other vertices. The main approach is to disseminate information between nodes about routes between nodes of one dynamic ad-hoc network through auxiliary intermediate messaging. The disadvantage is that the initialization stage takes a relatively long time and is a very expensive operation both in terms of the amount of memory used and the number of operations. Sync tables take up a lot of memory, and depending on the size of the graph can grow enormously. Any changes in the addition or subtraction of a node require a restructuring of the overall network map and initialization of the entire procedure. Also, the algorithm does not handle situations of changing the physical position of the node and is more focused on fixed wireless networks.

The *Routing Information Protocol (RIP)* has a fairly simple algorithmic implementation, which makes it quite attractive for implementation problems in small computer networks. Like other analogues, the protocol implements a set of functionalities for computer networks with dynamic updating of route information. The source of data for each agent of the network are neighboring agents, which transmit on demand a certain general information characteristics of the network to the node, and the feature of the network is the direction of information

transmission and the distance in the “jumps”. This kind of characteristic is needed to indicate the distance between the nodes. This protocol does not take into account the interference that may exist during data transfer within a single file or message transaction, which creates a number of problems for implementation in a fast network deployment. It is necessary to enter the designation of the edge with a certain physical and information characteristics between the two agents of the network graph, which is the transmission of packets. The basic logic of supporting continuous data transmission is to minimize the transactional moments of data transmission, taking into account the complexity of the path, which is directly proportional to the distance of the path. The disadvantage of the algorithm is the maximum number of jumps within the network is 15 edges due to the internal matrix implementation. Each node once every 30 seconds sends packets with information about the network map to the nodes. This type of operation is extremely stressful for the network, as it requires the transfer of large amounts of traffic. Because network synchronization packets are sent to all agents that have a connection to a common graph, the messages must be divided into node-specific and non-graph-specific, or other stand-alone systems.

The *Enhanced Interior Gateway Routing Protocol (EIGRP)* is designed to operate within an autonomous system. It differs in that it has a wide range of network configuration options and the ability to select different components for calculations.

The distributed algorithm has a number of advantages, such as self-organization and ease of implementation, but there are problems with network scaling that may have solutions. The object of work can also be problems with disappearance because the destruction of one of the nodes of the network and the calculation to infinity, when the packet goes on the network until it bypasses each node, with possible looping within the network. When implementing continuous node synchronization, this algorithm receives an extremely large increase in performance when adding nodes in real time. Accordingly, since AODV meets all these requirements, it can be predicted that by adding a number of improvements to the algorithm itself, we can get significant performance improvements.

The proposed method

Basic calculation algorithm. In this research, AODV is chosen as the basic protocol, which contains a remote-vector routing algorithm, which

makes it easy to extend. An algorithm of this kind is distributed, because the calculations are not stored in a specific place, but distributed in clusters of certain groups of vertices of the graph, message transmission may not be widespread due to the asynchrony of operations in the whole network structure [6]. Architecturally, it is specified that each node has a certain entity, which in turn may contain information about the tabular values of routing data with one record for each. The data structure is a vector of structures that store several components (selected line and distance) [7].

During the intermediate calculations of the node there are problems of estimating the distance or other characteristics, such as:

- distances;
- number of jumps;
- delay time;
- the length of messages waiting to be sent inside the column.

The hardware support of this algorithm is implemented by sets of drivers for network equipment and provides an increase in the bandwidth of channel traffic and makes it impossible to account for traffic in networks of this kind due to asynchrony. Also a known problem is the low speed in cases of looping inside the node due to the limited implementation of data structures to save the routing table [8–10].

To improve the protocol architecture for continuous data transmission support in peer-to-peer networks, a number of optimizations is proposed, where each node is assigned a network ID that is automatically calculated when connected to it. For each node, the costs and path to the neighboring node are calculated, and this data is stored in the routing table. In part, each node can prioritize the most accessible nodes that are closest and have the least load and synchronize routing tables, excluding duplicates.

When caching the result, the packet delay time before sending to the recipient will be reduced. This mechanism allows quickly restoring routes in case of disruption of one of the internal connections. When using the HTTP protocol, it is necessary to multiplex the streams using a binary layer of frames, which allows transmitting packets in both directions without delays between streams. This has several advantages: parallel multiplexed packet data requests do not block each other; when storing resources on multiple nodes, the speed increases.

Compression of data headers by HPACK reduces service traffic. For “hot” nodes, a request-pipe is applied at the message queue processing

level. The node monitors the queue and ignores duplicate messages by pre-identifying each binary value consisting of a timestamp (4 bytes), a node ID (3 bytes) and a counter (3 bytes). This model allows creating up to 220 unique messages per second. After segmenting a packet and an identifier, we can compare packets and identifiers for duplication.

Features of algorithm implementation

The target algorithm for continuous data transmission for research and improvement is AODV, and this modification must be added for the condition of use in conditions of rapid network deployment. The modification is as follows.

- Fix a problem with missing nodes and message loops within the network.
- Modify the message transmission on the edges of trusted network agents and rebuild the current table.
 - Replace the default data structures with those that match the dynamic change of characteristics.
 - Reduce message delay time.
 - Reduce the number of lost packets to support the continuous exchange of information.
 - Reduce the amount of traffic generated by the network during operation.

The above problems are critical and are solved by the following algorithmic implementations.

The problem of disappearance of nodes

The problem of node disappearance is one of the most important for optimization in this work and the development of software for routing algorithms. Solving this problem can greatly improve the overall time of each node under certain loads. When constructing a path, the network decides to send a group of packets over the tested channel, but because the nature of dynamic channels is unpredictable, sometimes nodes of this kind may disappear from the field of view of networks, and sent packets begin to search for their destination “wander” on the network, which leads to loops and loss due to a number of jumps entered in the message headers as inactive, after which initiates a new route and retransmission of messages, which reduces speed and leads to loss of time and overall network performance.

The solution to this problem will be the introduction of an intermediate additional routing table with the states of the neighbors, where the depth $n < 3$, the neighbor knows more about the neighbors of the node initiating the route. It is worth noting

that due to the small depth of awareness in the network topology, updating such tables will be much faster, and as part of the general remote-vector algorithm, less calls will be generated much more often and less traffic for partial node synchronization.

In the event of the disappearance of node X in the intermediate maps, a check will be made for the presence of the node in the neighbors:

```
(this.target (source.node) .contains (x=>
  x.super(x.contains (x.mappedNodeUniqueId)).
  getMonade as expression)
.Invoke (this.localGraphMap => x.reduce
(node, target) =>
  (x.getAnyClosest (oldWay) .id =
  = x.nextNodeId)))
```

This improvement allows finding the way back to the sender in dynamic mode due to the asynchrony of the algorithm without loss of speed, giving a response to the call to send a new packet, to modify the transmission of messages on the edges of network agents that are trusted and rebuild the current table.

The following modification must be added during transmission to eliminate packet looping in the middle of the graph.

- When sending the message width, index each packet according to the combinations of machine number, node, message number, creation time for the uniqueness of the message.
- When receiving a message with the top to perform “coloring” of the node relative to the message, marking it as already received, saving the hash of the message body, to eliminate re-looping. Only hash is stored, takes up little space. Hash comparisons are a quick operation in Redis implementation.

This improvement, along with the hashing of the HPACK algorithm headers, is the most productive due to the avoidance of loops. Header compression also reduces the amount of service traffic, which has a positive effect on the overall trend of the algorithm (Fig. 4).

Reduce message delay time

As an improvement, we change the delay time of the message during the processing of the message packet within the node by optimizing the database.

At the moment, the distance-vector algorithm is based on the vector of values of the routing table, which is a two-link list (Fig. 5). The search is performed by scanning the entire collection of such records in the table, which on a large number of messages and nodes gives complexity n (vector length).

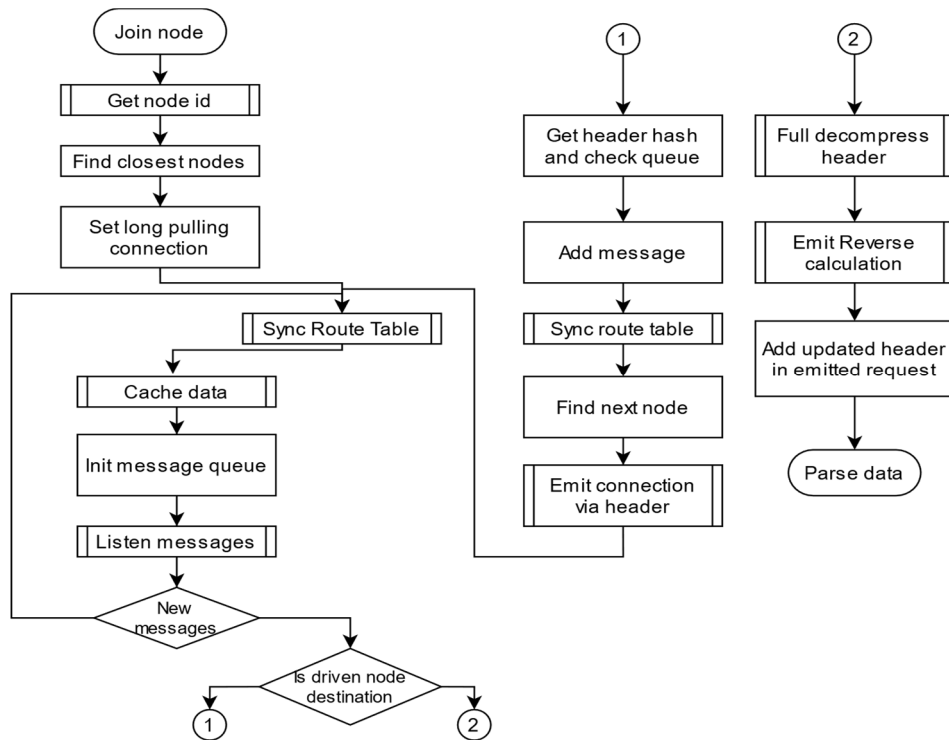


Fig. 4. Algorithm of the proposed method

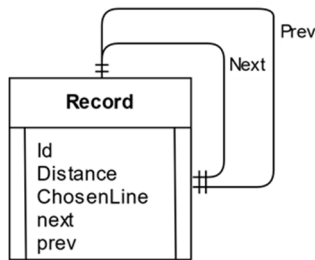


Fig. 5. The current scheme of saving the routing table

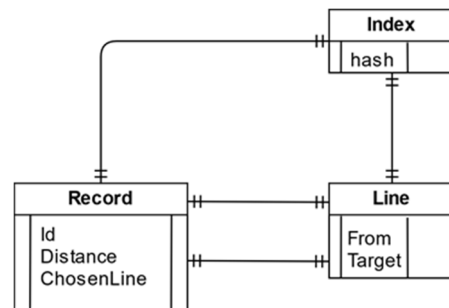


Fig. 6. Routing table saving scheme

Applying a graph-oriented approach in combination with the implementation of Redis hash tables and collections and taking it into RAM, we obtain an increase in $\log(n) - \sqrt{n} \log(n)$ times by optimizing the tables with the scheme shown in Fig. 6.

By reducing the entities, we reduce the frequency of table updates, respectively, creating opportunities to create an index without losing updates as part of Redis. By providing connected access to each other, we cannot worry about performance. The increment to the logarithmic value is provided by indexing, where the Line table is one key with unique values.

Evaluation of the effectiveness of the proposed method

To evaluate the method of continuous data transmission, it is necessary to determine the key parameters that are important for the operation of the algorithm in both the minimum mode and the maximum.

After a detailed study of the issue, we enter the following indicators:

- number of nodes involved in routing;
- each node has a route to any other node in the network;
- the possibility of loops;

- the level of knowledge of each node of the topology of the entire network;
- the number of resources to find a backup communication channel;
- use of office traffic.

We also enter the following quantitative characteristics:

- number of nodes;
- time to overcome the path;
- delay in the node.

These characteristics fully assess the improvements in the framework of continuous data transmission in decentralized networks with variable topology.

Testing of the proposed method is performed using a computer model that simulated the system. In Fig. 7 the graph has 12 vertices and approximately the same length of edges. The edges are generated according to the position of the vertices, which, in turn, are placed randomly.

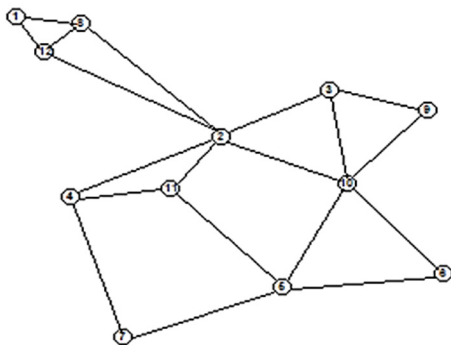


Fig. 7. The investigated graph of the MANET network

Table 1 shows the test results and the results of the comparison of methods.

Table 1. Comparison of general characteristics of algorithms

Method with modification	Method without modification	Characteristic
7.31	6.15	The number of nodes involved in routing
+	+	Each node has a route to any other node in the network
0.2	0.4	The possibility of loops
+	+	The level of knowledge of each node of the topology of the entire network
352b	438b	Use of office traffic

The overall result is the average of all attempts to send a packet from one random node to another in the opposite location with the appropriate initialization.

The results shown in Table 2 can give a clear view that in the range from 40 to 100 kb/s network load, the proposed algorithm also has a number of improvements, namely the reduction of message processing delay in the node measured as the average statistical data testing.

Table 2. Time delay of processing in each node

Load kb/s	Modified (delay (s))	Not Modified (delay (s))
20	0.0093	0.0081
40	0.00103	0.00119
50	0.0142	0.0157
70	0.0162	0.0171
100	0.0271	0.0276
120	0.0302	0.0294

This improvement allows us to see that:

- delay within the node decreased by 7–15 %, depending on the topology of the graph vertex and the interaction of the network agent with other agents;
- as part of the network operation with the number of nodes that will increase, this improvement may increase network performance, in particular, its bandwidth;
- the path time of the message, which depends on the amount of passage and processing of the message by all nodes-transmitters of the network, has also improved (depending on the position of the graph in space and edges).

Developed software architecture

Given the possible areas of application of this algorithm, it is decided that the system architecture should be built on a client-server model of the modular type. The modular architecture allows implementing the principle of Dependency Injection, which from the point of view of embedding the architecture can in the development of compilation programming languages with strict typing to provide the resulting software flexibility and support for the implementation of polymorphic hooks.

Client part - Web App. Resources are application resources that describe buttons, tab names, background names, text constants, extensions, internal images that cannot be permanently unloaded

into the phone memory, used and initialized by type scoped with dispose elements after the need for the latter.

Middleware. Middleware is internal logic of the application, combined with modules, extensions, databases, internal states of the system (StateMachine).

Redis is used as internal database of the application, which is characterized by extremely high performance in storing and processing data. Document-oriented key storage system is a key-based, open-source dictionary display that can be freely picked up and implemented in the application itself. It is used to optimize headers and as a new data structure instead of a vector representation of standard data structures in the AODV algorithm, to change the two-link list of vertex structures with a full scan – search for it.

RabbitMQ is used as internal software component focused on processing the message queue. Used for packet numbering, queue processing, internal coloring of the graph as part of the interaction-transmission of the message.

Inner Queue is used as hashed table of message data that stores the general collection of messages that come to the node and are there until the packet is sent.

Network Graph Map is used as generated map of the network graph, vertices and edges with their characteristics.

Network Part Map is used as partial map of the network of adjacent vertices connected to the current vertex, used to optimize intermediate table updates.

Connection Manage control is a software application module that is responsible for managing connections to this node.

Balancing and route algorithm is implementation of the algorithm implemented in the application.

Mobile OS API. Mobile OS API provides a mobile application for working with embedded systems and the ability to obtain data from them for optimal system operation and a better understanding of the state in which the system itself.

Services is created within the operation of the Android operating system, data interfaces and communication.

Sensors is a set of built-in APIs for working with sensors of the device based on Android OS, the set of the latter depends on the device itself.

Wireless is a set of drivers that allows controlling a wireless connection to other devices.

Input is a set of drivers that read all user interactions with the system and the device and provide access to the keyboard and read data from the monitor, microphone, gyroscope, etc. Intertwined with sensors.

Hardware API. It is a set of calls for interaction with the hardware of the system, to create additional daemons inside the Linux machine.

This implementation contains the following benefits and improvements.

- The system is not tied to a certain algorithm, the algorithm through the Dependency Injection mechanism can be changed to another, the implementation of a new algorithm is much easier.
- Detachment allows transferring and implement the algorithm itself, other than the mobile application, to other systems.
- The data representation layer is a user interface and is responsible for presenting data to the user and receiving control commands and data.
- The application level implements the basic logic of the application, which carries out the necessary information processing.
- The data access layer provides data storage and access.

The developed software application contains a modified algorithm with the ability to interact with the hardware API for Android and supports the exchange of information on MANET network.

Conclusions

This research presents the quality analysis of traffic routing method in the middle of the network, which is reduced to routing optimization based on the Ad hoc On-Demand Distance Vector algorithm. In this research four main indicators are considered: the speed of establishing the path, the delay of the message in one node, the occurrence of loops, the amount of service traffic.

To test the proposed method a graph simulating a wireless peer-to-peer MANET network of twelve graph vertices is modeled. Testing is performed in more than a hundred iterations. At each iteration vertices are randomly selected and data transfer between each of the graph vertex is initiated.

Due to the use of the created software, the analysis of quality indicators of continuous data transmission and network routing is performed by the proposed method. The proposed method shows the best results by 5–10 % reduction of the delay time inside the node and when processing the message, while the amount of service traffic is reduced by 15 %.

References

- [1] P. Khanpara, "Security issues in MANETs", in *Int. J. Adv. Res. Sci. Eng.*, vol. 6, no. 6, pp. 220–225, 2017.
- [2] P. Khanpara and B. Trivedi, "Security in mobile ad hoc networks", in *Proc. Int. Conf. Commun. Netw.*, vol. 508, pp. 501–511, 2017. doi:10.1007/978-981-10-2750-5_52
- [3] *Wireless ad hoc network. Routing*. [Online]. Available: https://en.wikipedia.org/wiki/Wireless_ad_hoc_network#Routing
- [4] T. Hara, "Data management issues in mobile ad hoc networks", in *Proc. Jpn Acad. Ser. B*, vol. 93, no. 5, pp. 270–296, 2017. doi: 10.2183/pjab.93.018
- [5] L. Odedra *et al.*, "Detection and prevention of selfish attack in MANET using dynamic learning", *IOSR J. Comput. Eng.*, vol. 18, no. 3, pp. 54–61, 2016.
- [6] S. Basagni *et al.*, "Mobility-adaptive protocols for managing large ad hoc networks", in *ICC 2001. IEEE Int. Conf. Commun.*, vol. 5, pp. 1539–1543, 2001. doi: 10.1109/ICC.2001.937178
- [7] I.F. Akyildiz *et al.*, "Wireless mesh networks: A survey" in *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, 2005. doi: 10.1016/j.comnet.2004.12.001
- [8] C.E. Perkins *et al.*, "Performance comparison of two on-demand routing protocols for ad hoc networks", *IEEE Pers. Commun.*, vol. 8, no. 1, pp. 16–28, 2001. doi: 10.1109/98.904895
- [9] C.E. Perkins *et al.* (2003). *Ad hoc on-demand distance vector (AODV) routing* [Online]. Available: <https://tools.ietf.org/html/rfc3561>
- [10] I.D. Chakeres and L. Klein-Berndt, "AODVjr, AODV simplified", *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 100–101, 2002. doi: 10.1145/581291.581309

Л.М. Олещенко, К.О. Мовчан, В.І. Петрусь

МЕТОД ПЕРЕДАЧІ ДАНИХ НА ОСНОВІ МУЛЬТИПЛЕКСУВАННЯ ТА ВРАХУВАННЯ ЗМІНИ ВІДНОШЕННЯ ВУЗЛІВ У МЕРЕЖАХ MANET

Проблематика. Задача побудови мережі з безперервним режимом передачі даних для рухомих вузлів зводиться до побудови безпроводової мережі та підтримання стану втрати мінімальної кількості пакетів під час обміну інформацією. Однією з головних проблем управління мережами MANET (Mobile Ad hoc Networks) є їх динаміка та те, що мобільні пристрої обмежені у своїх обчислювальних можливостях. Існує проблема маршрутизації трафіка всередині мереж MANET і внутрішньої ієрархії передавачів, залучених до системи. До того ж ці мережі мають проблеми оптимізації пропускнуої спроможності та управління потужністю. Брак фіксованої топології вводить нові наукові проблеми, зокрема мережеву конфігурацію, пошук пристроїв і підтримки такої топології, а також спеціальну адресацію та саморуйнування мережі. Ця задача є актуальною для передачі даних між рухомими пристроями в умовах розвідувальних операцій.

Мета дослідження. Дослідити зменшення затримок пакетів у вузлах під час динамічної маршрутизації, зменшення службового трафіка та програмну оптимізацію передачі даних у мережах MANET відносно наявних методів.

Методика реалізації. Для досягнення цієї мети використовуються методи комп'ютерного моделювання, статистичні та емпіричні методи. На основі оптимізації динамічної маршрутизації з використанням Ad hoc On-Demand Distance Vector алгоритму виділено чотири основні показники: швидкість встановлення шляху, затримка повідомлення в одному вузлі, виникнення зациклень, обсяг службового трафіка. Для тестування запропонованого методу моделюється граф із дванадцяти вершин, що імітує безпроводову однорангову мережу MANET. Тестування проводиться в кількості понад сто ітерацій. За кожної ітерації випадковим чином вибираються вершини графа мережі та моделюється передача даних між кожною вершиною графа.

Результати дослідження. За використання створеного програмного забезпечення було проведено аналіз показників якості безперервної передачі даних і маршрутизації мережі запропонованим методом. Запропонований метод показав зменшення обсягу службового трафіка на 15 % і на 5–10 % кращі результати зменшення часу затримки в межах вузла та при обробці повідомлення.

Висновки. Запропонований метод дає змогу зменшити час затримки в межах вузла, який змінює своє положення при обробці повідомлення, а також зменшити об'єм службового трафіка при передачі даних між вузлами в мережі MANET. Для дослідження запропонованого методу використано комп'ютерне моделювання.

Ключові слова: безпроводові мережі; динамічна маршрутизація; peer-to-peer; Ad-hoc мережі; MANET; алгоритм Ad hoc On-Demand Distance Vector.

Л.М. Олещенко, К.А. Мовчан, В.І. Петрусь

МЕТОД ПЕРЕДАЧІ ДАНИХ НА ОСНОВЕ МУЛЬТИПЛЕКСИРОВАНИЯ И УЧЕТА ИЗМЕНЕНИЯ ОТНОШЕНИЯ УЗЛОВ В СЕТЯХ MANET

Проблематика. Задача построения сети с непрерывным режимом передачи данных для подвижных узлов сводится к построению беспроводной сети и поддержания состояния потери минимального количества пакетов при обмене информацией. Одной из главных проблем управления сетями MANET (Mobile Ad hoc Networks) является их динамика и то, что мобильные устройства ограничены в своих вычислительных возможностях. Существует проблема маршрутизации трафика внутри сетей MANET и внутренней иерархии передатчиков, вовлеченных в систему. Также эти сети имеют проблемы оптимизации пропускной

способности и управления мощностью. Отсутствие фиксированной топологии вводит новые научные проблемы, такие, как сетевая конфигурация, поиск устройств и поддержка такой топологии, а также специальная адресация и саморазрушения сети. Эта задача актуальна для передачи данных между подвижными устройствами в условиях разведывательных операций.

Цель исследования. Исследовать уменьшение задержек пакетов в узлах при динамической маршрутизации, а также уменьшение служебного трафика и программную оптимизацию передачи данных в сетях MANET относительно существующих методов.

Методика реализации. Для достижения этой цели используются методы компьютерного моделирования, статистические и эмпирические методы. На основе оптимизации динамической маршрутизации с использованием Ad hoc On-Demand Distance Vector алгоритма выделено четыре основных показателя: скорость установления пути, задержка сообщения в одном узле, возникновение заикленности, объем служебного трафика. Для тестирования предложенного метода моделируется граф из двенадцати вершин, имитирующий беспроводную одноранговую сеть MANET. Тестирование проводится в количестве более ста итераций. При каждой итерации случайным образом выбираются вершины графа сети и моделируется передача данных между каждой вершиной графа.

Результаты исследования. Благодаря использованию созданного программного обеспечения проведен анализ показателей качества непрерывной передачи данных и маршрутизации сети предложенным методом. Предложенный метод показал уменьшение объема служебного трафика на 15 % и лучшие на 5–10 % результаты уменьшения времени задержки в рамках узла и при обработке сообщения.

Выводы. Предложенный метод позволяет уменьшить время задержки в рамках узла, который меняет свое положение при обработке сообщения, а также уменьшить объем служебного трафика при передаче данных между узлами в сети MANET. Для исследования предложенного метода использовалось компьютерное моделирование.

Ключевые слова: беспроводные сети; динамическая маршрутизация; peer-to-peer; Ad-hoc сети; MANET; алгоритм Ad hoc On-Demand Distance Vector.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
01 жовтня 2020 року

Прийнята до публікації
10 грудня 2020 року