Wireless Netw (2018) 24:2509–2519
https://doi.org/10.1007/s11276-017-1485-2

AODVCS, a new bio-inspired routing protocol based on cuckoo search algorithm for mobile ad hoc networks

Akram Kout[1] · **Said Labed**[1] · **Salim Chikhi**[1] · **El Bay Bourennane**[2]

Published online: 11 March 2017

Abstract Mobile ad hoc networks (MANETs) are becoming an emerging technology that offer several advantages to users in terms of cost and ease of use. A MANET is a collection of mobile nodes connected by wireless links that form a temporary network topology that operates without a base station and centralized administration. Routing is a method through which information is forwarded from a transmitter to a specific recipient. Routing is a strategy that guarantees, at any time, the connection between any two nodes in a network. In this work, we propose a novel routing protocol inspired by the cuckoo search method. Our routing protocol is implemented using Network simulator 2. We chose Random WayPoint model as our mobility model. To validate our work, we opted for the comparison with the routing protocol ad hoc on-demand distance vector, destination sequence distance vector and the bio-inspired routing protocol AntHocNet in terms of the quality of service parameters: packet delivery ratio and end-to-end delay (E2ED).

✉ Akram Kout
akram.kout@univ-constantine2.dz

Said Labed
said.labed@univ-constantine2.dz

Salim Chikhi
salim.chikhi@univ-constantine2.dz

El Bay Bourennane
ebourenn@u-bourgogne.fr

[1] SCAL Team, MISC Laboratory, Abdelhamid Mehri Constantine 2 University, 25000 Ali Mendjli, Algeria

[2] LE2I Laboratory UMR-CNRS, University of Burgundy, 21000 Dijon, France

Keywords Mobile ad-hoc networks · Bio-inspired routing · Cuckoo search algorithm · Random WayPoint · QoS · Network simulator 2

1 Introduction

Mobile ad hoc network (MANET) is a collection of wireless mobile nodes that dynamically forms a temporary network without using any centralized administration. The challenge in MANET is to find a path between communicating nodes. This type of network is characterized by the absence of centralized infrastructure, dynamic topology, constraint of energy, heterogeneity of nodes, multi-hop, and limited bandwidth.

Routing in MANET means finding a path between the source and destination where packets can be forwarded. This Process is extremely challenging because of the dynamic features, limited bandwidth, frequent topology changes caused by node mobility and energy consumption of MANET. Routing protocol should not only find the shortest path between the source and destination, but should also be adaptive. Routing protocols can be classified into three main categories, namely, proactive, reactive, and hybrid protocols. The third category combines the mechanisms of the two cited protocols, hence its name.

- *Proactive routing protocols* Control packets are constantly broadcasted in the network to maintain the state of the link between each pair of nodes. A table is constructed at each node, where each entry indicates the next hop to a certain destination. This method provides readily available information during data transfer. However, one disadvantage of this algorithm is the need to track all topology changes. This

requirement is difficult to fulfil when several nodes are present or when nodes are mobile [1]. The most important routing protocols in this class are destination sequence distance vector (DSDV [2]) and optimized link state routing (OLSR [3]).

- *Reactive routing protocols* (or on demand), paths are created and maintained as needed. When a node requires routing, a global discovery procedure of paths is launched to obtain a valid path to the destination. This approach is efficient, but it can lead increased delays because routing information is not immediately available when needed. The most important routing protocols in this class are: ad hoc on demand distance vector (AODV [4]) and dynamic source routing (DSR [5]).
- *Hybrid protocols* Hybrid protocols are a combination of the two cited approaches. These protocols use a proactive approach to examine the close neighborhood (on two or three hops), and they have the paths in the immediate neighborhood. The hybrid protocol uses the techniques of reactive protocols beyond the predefined area to identify routes between non-neighboring nodes (more than two or three hops). The most important routing protocol in this class is Zone Routing Protocol (ZRP [6]). Following Di Caro et al. [7] proposal, we employ AntHocNet, which is a hybrid and multipath routing algorithm for MANET. Routes in AntHocNet are reactively established and proactively maintained during communication sessions. When a demand for a route exists, the algorithm initially checks the routing table of the source node to obtain information about the destination. If routing information of the destination is not available, the algorithm will broadcast a forward ant. Each neighbor then receives a copy of the ant and checks its routing table. The ant will be broadcasted again if routing information of the destination is not available; otherwise, the ant will be sent to the next hop using stochastic decision. Each forward ant maintains a list of visited nodes. When the forward ant reaches the destination, it uses the list to take the same path back to the source and update the pheromone values deposited in the links. To avoid generating a huge number of ants of the same generation, limit a on the length of a path is defined. When the length of the path traveled by an ant exceeds a, the ant is discarded. Whenever an intermediate node receives two ants of the same generation, such a node checks the length of the paths traveled by both ants. The ant that traveled a longer path is discarded.

Several optimization problems have no general algorithm that should be solved in polynomial time. This problem limits the use of exact methods for solving problems of small sizes. The majority of meta-heuristics for solving optimization problems that were presented in recent literature are biologically inspired. Examples of these meta-heuristics are genetic algorithms, simulated annealing, neural networks, particle swarm optimization and ant colony optimization and cuckoo search.

These algorithms are inspired by the principles of natural biological evolution and distributed collective behavior of social colonies have shown excellence in dealing with complex optimization problems and are becoming increasingly popular given the limited transmission range of such networks. This decentralized operation relies on the cooperative participation of all nodes. In this study, we propose a new routing protocol inspired by the cuckoo search method.

The paper is organized as follows: Sect. 2 describes the cuckoo search based routing in MANET. The proposed approach is described in Sect. 3. Simulation results are discussed in Sect. 4. Finally, Sect. 5 offers conclusions and directions for future works.

2 Cuckoo search-based routing in MANET

2.1 Cuckoos in nature

Cuckoos are fascinating birds, because of the beautiful sounds they make and their aggressive reproduction strategy. Some cuckoo species lay their eggs in communal nests, but some of them remove the eggs of other cuckoos to increase the hatching probability of their own eggs [8]. Numerous species engage in obligate brood parasitism by laying eggs in the nests of other host birds and species. Brood parasitism has three basic types, namely, intraspecific brood parasitism, cooperative breeding, and nest takeover. Some host birds can engage in direct conflict with intruding cuckoos. If a host bird discovers that the eggs are not their own, they will either throw these alien eggs away or simply abandon the nest and build a new one elsewhere. Some cuckoo species, such as the New World brood-parasitic *Tapera* have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colour and pattern of the eggs of a few chosen host species [8]. Mimicry reduces the probability of their eggs being abandoned thereby increasing their reproductivity.

2.1.1 Lévy flight

For simplicity in describing our approach, we used three ideal rules established by Yang and Deb [8]:

1. Each cuckoo lays one egg at a time t and dumps its egg in a randomly chosen nest.

2. The best nests with high quality of eggs will be carried over to the next generations.
3. The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with probability $P_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest and build a completely new nest. For simplicity, this last assumption can be approximated by the fraction pa of the n nests are replaced by new nests (with new random solutions).

2.2 Cuckoo search based routing in MANET

Chandra and Sivarama [9] proposed a secure routing protocol, Trust Predicated Routing (TPR), based on cuckoo search. TPR guarantees the complete security and reliability of the proposed routing protocol. Cuckoo search reduced time consumption and facilitated efficient routing. The simulation showed that the proposed protocol achieved a high packet delivery rate and issued a weak end-to-end delay (E2ED) and low energy consumption.

Algorithm 1 Pseudo code of the Cuckoo Search (CS)

```
Objective function f(x), x = (x_1, ..., x_d)^T
Generate initial population of n host nests x_i (i = 1, 2, ..., n)
while (t <stop criterion) do
    Get a cuckoo (say, i) randomly by Lévy Flight
    evaluate its quality/fitness Fi
    Choose a nest among n (say, j) randomly
    if (F_i > F_j) then
        replace j by the new solution;
    end if
    abandon A fraction P_a of worse with new ones at new location via levy flights;
    Keep the best solutions (or nests with quality solutions);
    Rank the solutions and find the current best
end while
return the best solution
```

A Lévy flight is performed when generating new solution $x^{(t+1)}$ for a cuckoo i:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Lévy(\lambda) \quad (1)$$

where $\alpha > 0$ is the step size related to the scales of the problem of interests. In most cases, we can use $\alpha = 1$. Equation 1 is essentially the stochastic equation for random walk. A random walk is generally a Markov chain, wherein the next status/location depends on the current location (the first term in Eq. 1) and transition probability (the second term). Product $\oplus$ means entry-wise multiplications. This entry-wise product is similar to those used in PSO, but random walk via Lévy flight is more efficient in exploring the search space because its step length is extended in the long run.

The Lévy flight essentially provides a random walk, whereas random step length is drawn from a Lévy distribution

$$Lévy \sim \mu = t^{-\lambda}, (1 < \lambda \leq 3) \quad (2)$$

Some new solutions should be generated by Lévy walk around the best solution to speed up local search.

Kaur and Kaur [10] proposed two new protocols based on the cuckoo optimization algorithm (COA); they also used the principle of operation of AODV and dynamic manet on demand (DYMO proposed by Gupta et al. [11]) routing protocols. COA finds the shortest path between two nodes and reduces congestion in the AODV and DYMO routing protocols. COA also improves energy efficiency, lifetime and quality of service (QoS) of DYMO and AODV routing protocols.

3 Proposed approach

Our study is into two phases. The first phase is the implementation of a new routing protocol, and the second phase is its simulation in order to evaluate its performance. We aim to improve the AODV routing protocol using the recent meta-heuristics cuckoo search Algorithm (CSA) which is utilized to identify the best path between two nodes.

We implemented a new protocol, which we called AODVCS. This protocol is divided into two major parts.

Part 1: Detection of topology and construction of the initial population of solutions

The network is browsed from the source node until destination is reached to determine a portion of topology that contains the source and destination. A population of initial solutions is built, which contains different possible solutions.

Part 2: Routing management

Cuckoo search is applied to obtain the shortest path between the source and destination in a number of valid paths.

The following algorithm represents the main algorithm of the proposed approach for reactive routing protocol

Algorithm 2 AODVCS

```
Begin
Detection of the topology and construction of the initial population of solutions
Routing Management by applying Cuckoo search()
End
```

3.1 Detection of topology and construction of the initial population of solutions

This phase starts with the source node in cases that require a route to a certain destination, but such a route is not available. Route is obtained by broadcasting a Route REQuest (RREQ). To build the initial population used by cuckoo search, we modified the structure of the packet RREQ (Table 1) by adding a new field named *rq_route* (Table 2). This field is a table that contains the *id*s of intermediate nodes between the source and destination. The nodes are arranged in such a way that a node adds its identifier in the *rq_route* field each time RREQ is received. This method ensures that a solution is built.

Solution representation The solution is represented by a table that contains the *id*s of participating nodes in the route between the source and destination. Nodes that received a RREQ are associated with the source to reach the destination. The first column of the table contains the source node, the last contains the destination node, and the other columns contain the intermediate nodes (Table 3).

When the destination node receives a packet-type RREQ, it stores the header of this packet in a vector of the RREQs associated with the same source and destination to create a population of solutions.

A MANET of communicating stations may be represented in the form of graph G = (V, E). Set V represents the stations and E represents the links of communication [12]. When a MANET is represented as a graph, we can apply the concepts of the theory of graphs to solve its problems.

After constructing the population in the destination node, we used the information on the communication links between the intermediate nodes from the source to the destination that exist in the *rq_route* field to create an adjacency matrix that represents the links between the nodes of the network that received the RREQ. The following algorithm shows the process of creating the adjacency matrix.

Algorithm 3 Creation of the adjacency matrix

```
Varaiables
s : solution    /* Individual */
n : node
Inputs
p : initial population
dest : node
Begin
Initialize Adjacency_matrix to 0;
for each s in p do
    for (each n in s) and (n ≠ dest) do
        Adjacency_matrix[n][successor(n)]=1;
    end for
end for
Return Adjacency_matrix;
```

Table 1 General format of a packet RREQ

@Src	Num.seq.src	Broadcast_id	@dest	Num.seq.dest	Nbr_hops

Table 2 Format of a package RREQ after modification

@Src	Num.seq.src	Broadcast_id	@dest	Num.seq.dest	Nbr_hops	**rq_route**

Table 3 Example of a solution

0	3	5	4	11	15

3.2 Routing management

This phase involves the application of CSA on the population created in the first phase. CSA is used to optimize the search for a route between two nodes.

The fitness that we will optimize using the cuckoo search is the distance from the source to the destination. The value of fitness is obtained from the field of the number of hops (*rq_hop_count*) in the RREQ packet for initial solutions. This value is then calculated for the new solutions obtained by the Lévy flight. The CSA used in our approach is summarized in the following pseudo code.

3.2.1 The Lévy flight

The Lévy flight is used to generate new solutions. Changes are made on the intermediate nodes to maintain the source and destination nodes. The result obtained by the Lévy flight is of real type (Tables 4, 5).

The Lévy flight method used in our approach is summarized in the following pseudo code.

Algorithm 4 Cuckoo search used by AODVCS

```
Inputs
Initial population
Adjacency_matrix
Output
The best route
Begin
for each solution in the population s do
    Obtain a new route g by Lévy flight;
    Repair the route obtained by Lévy flight using the Adjacency_matrix;
    Replace g by s if the fitness of g is better than the fitness of s;
    for a fraction P_a of bad solution s do
        Create a new solution g by the Lévy flight;
        Evaluate the fitness of g;
        Replace g by s if the fitness of g is best than the fitness of s;
    end for
    Keep the best solutions;
    Classify the solutions and find the best current;
end for
End
```

Algorithm 5 Lévy flight

```
Inputs
s : current solution
n, src, dest : node
Begin
Initialize the settings of Lévy flight;
for (each n in s) and (n ≠ src) and (n ≠ dest) do
    Apply the equation of Lévy flight;   /* The Equation 1 is provided in section 2.1.1 */
    Transform the real value to integer value;
end for
Return s;
End
```

Table 4 Example of a solution before the application of the Lévy flight

1	2	8	4	11

Table 5 Example of solution after the application of the Levy flight

1	2.25	8.25	4.25	11

Table 6 Example of solution after the transformation

1	2	9	5	11

Table 5 shows that the Lévy flight provides a real type solution, whereas the solutions in our protocol are positive integers. Thus, we used a random approach to transform the real values into integers (Table 6).

- The result of the Lévy flight is not necessarily valid. We proposed a route repair algorithm (RRA) to repair invalid solutions.
- This algorithm has three parts. The first part determines whether two successive nodes have a link between them. The nodes are kept if a link exists. If a link doesnt exist, we turn to the second part, which involves checking whether the first node has a link with the other successive nodes. If a node exists with a link to the first node, the intermediate nodes are removed. If such a node does not exsit, we turn to the last part, which involves checking whether a node entered the neighbors of the first node that has a link with the second node between the two nodes.

RRA is summarized as follows.

Algorithm 6 Route Repair Algorithm

```
Variable
isRepaired: a Boolean which shows if that route is repaired or not;
Inputs
Adjacency_matrix;
The route obtained by Lévy flight;
Output
Repaired route;
Begin
for each node i in the route do
    if (Adjacency_matrix[i][i+1] != 1) then
        IsRepaired = FALSE;
        for every other node x successor of the node n do
            if Adjacency_matrix[i][x] == 1 then
                Remove the intermediate nodes between i and x ;
                IsRepaired = TRUE;
            end if
        end for
        if IsRepaired == FALSE then
            for each neighbor v of i  do
                if Adjacency_matrix[i + 1][v] == 1 then
                    Insert the neighbor between i and (i+1);
                    break;
                end if
            end for
        end if
    end if
end for
if the route is valid then
    isRepaired= TRUE;
else
    isRepaired= FALSE;
end if
Return the repaired route;
```

4 Performance analysis

- Simulation environment

Network simulator 2 is a free-to-use network simulator in the public domain that uses discrete event simulation [13]. NS-2

runs in windows or Unix and works in combination with C++ and Object Tool Command Language (OTcl). The first programming language is used to develop and implement low-level operations, whereas the second one is used for simulations of code scripts scenarios. This combination of languages can quickly generate large-scale scenarios.

NS-2 is a central discrete scheduler of events, such as packet and timer expirations. The simulator can be easily extended with additional protocols. In the present study, AntHocNet was appended and AODVCS was implemented. AODV and DSDV were already available. Simulation in NS-2 is conducted at the packet level thereby facilitating a comprehensive analysis of simulation results.

Network ANimator (NAM) and XGraph tools are provided in NS-2 to graphically represent the progress of simulation and plot the results of simulation in the form of curves. The overall structure of NS-2 is shown in Fig. 1.

For the traffic/mobility model, we chose traffic sources at a constant rate, namely, constant bit rate (CBR) associated with the UDP protocol and varying numbers of CBR sources. Mobile nodes use the mobility model generated with a module of NS-2. To estimate the impact of density on the performance of ad hoc routing protocols, we varied the number of nodes from 40, 50, 60 and 100 nodes. The speed of nodes is uniformly distributed between 5 and 20 m/s. The physical characteristics of mobile nodes are summarized in Table 7.

- *Performance analysis* We used the following metrics, which facilitate the performance analysis of our routing protocol.
  - *Packet delivery ratio (PDR)* is obtained by dividing the number of packets received by the destination over the number of packets sent by the source. PDR specifies packet loss rate, which limits maximum network throughput. A high PDR indicates that the network is more reliable.
  - *End-to-end delay (E2ED)* The Mean E2ED is the average time required by a data packet to reach the destination. This measure is calculated by subtracting the time at which the first packet was transmitted by the source from the time at which the first packet has reached its destination. This calculation includes all possible delays caused by buffering during the latency of route discovery, queuing at the interface, retransmission delays at the MAC layer, propagation time and transfer. This process is important to estimate the delay introduced by path discovery. A small value of E2ED indicates best network performance.

We calculated both metric E2ED and PDR, the following table (Table 8) shows simulation results per protocol category:

To better observe data flow, NS-2 offered tool called XGraph (Fig. 2) which shows the evolution of sent packets according to the simulation time. Our protocol ensures good flow, which is attributed to optimal routes relative to the other protocols used in this synthesis. This finding justifies the reduced E2ED.

4.1 Discussion

(a) *Impact on packet drop and packet delivery ratio* Figures 3 and 4 show the impact of the proposed scheme on Packets Drop and Packet Delivery Ratio

Table 7 Physical characteristics of mobile nodes

Parameter	Value
Simulator version	NS-2.35
Routing protocols	AODV, DSDV, AntHocNet, AODVCS
Topology	1000 * 1000
Channel type	Channel/WirelessChannel
Mobility model	Random WayPoint (RWP)
Radio propagation model	Propagation/TwoRayGround
Packet length	512
Antenna type	Omni Antenna
Simulation type	300
Number of nodes	40, 50, 80, 100
Transport agent	UDP
Speed (min-max)	5–20 m/s

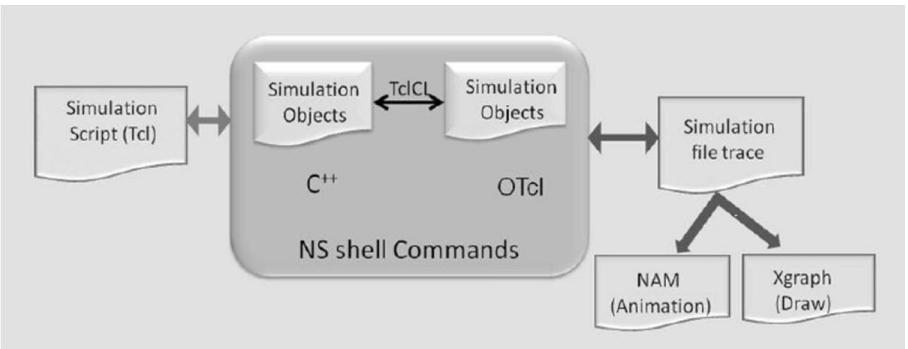

Fig. 1 Structure and components of NS-2

Table 8 Obtained results

Node density	Protocol	PDR(%)	E2ED(ms)
40	AODV	90.84	88.96
	DSDV	25.75	240.94
	AntHocNet	49.55	171.24
	AODVCS	90.24	66.58
50	AODV	91.42	116.88
	DSDV	27.60	506.19
	AntHocNet	60.96	156.89
	AODVCS	91.74	71.91
80	AODV	90.12	113.42
	DSDV	20.90	145.12
	AntHocNet	36.55	201.9
	AODVCS	89.68	98.47
100	AODV	89	129.54
	DSDV	12.62	110.27
	AntHocNet	18.23	329.61
	AODVCS	88.52	83.90

(PDR). The results obtained show that the proposed AODVCS is better than AODV, DSDV and AntHocNet algorithms. As the proposed algorithm uses reactive approach, so less number of packets were dropped than the other algorithms because of the its structure. Consequently PDR is better in AODVCS when compared with AODV, DSDV and AntHocNet in all scenarios: 40, 50, 80 and 100 nodes configuration.

(b) *Impact on received data packets*

Figure 5 shows the impact of proposed scheme on received Data Packets. The results obtained show that the proposed AODVCS receives more numbers of data packets than the algorithms like AODV, DSDV and AntHocNet. The reason for receiving higher packet rate is same as reason given in Section above and in all scenarios.

(c) *Impact on end-to-end delay*

Figure 6 shows the impact of the proposed scheme on End-to-End delay. The obtained results

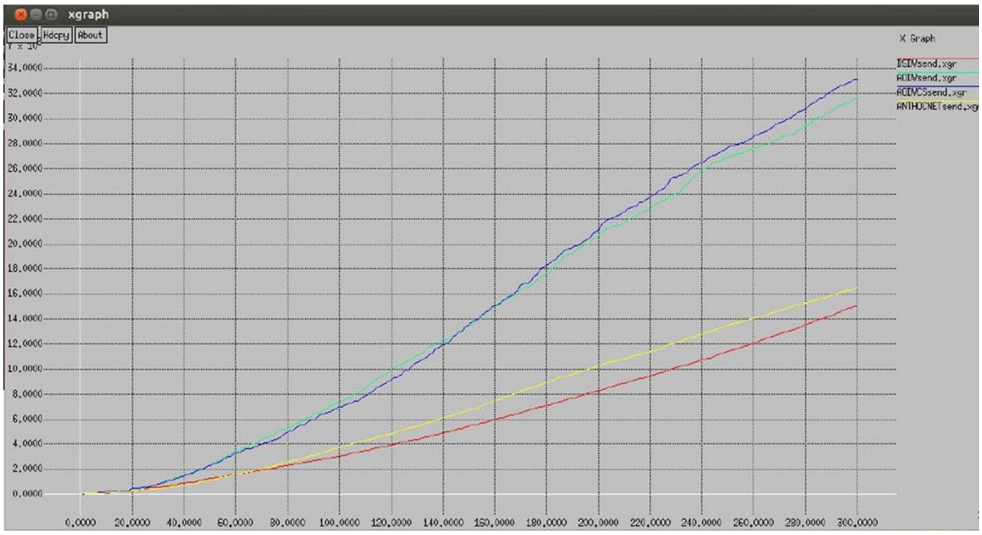

Fig. 2 Sent packets for AODV, DSDV, AntHocNet and AODVCS

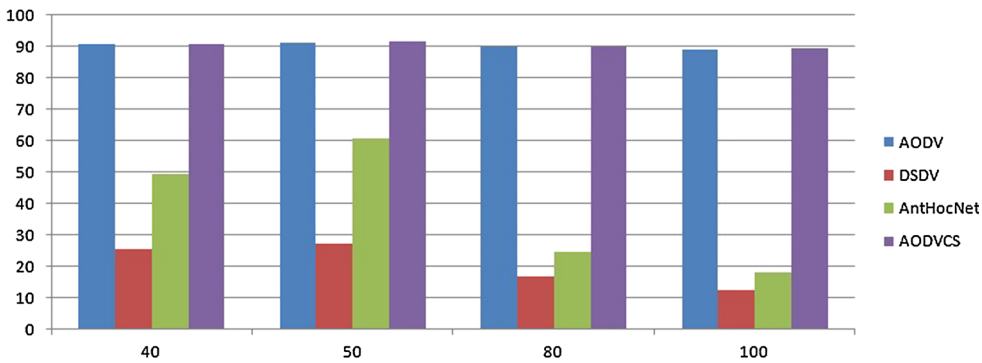

Fig. 3 Packet delivery rate results

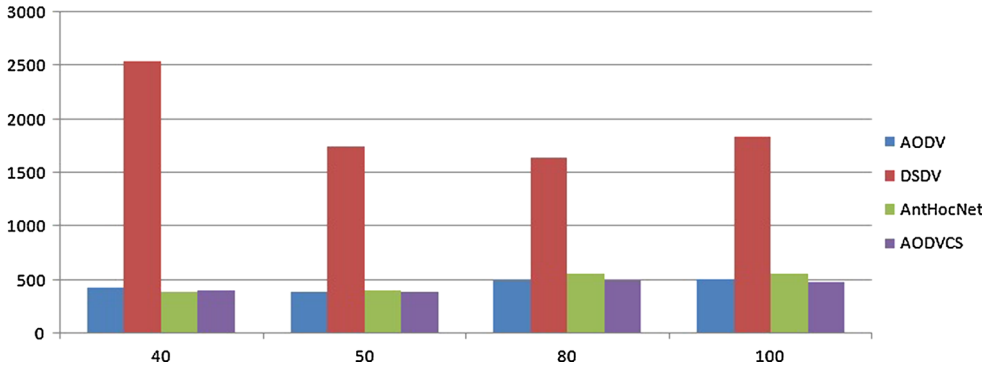

Fig. 4 Dropped packets results

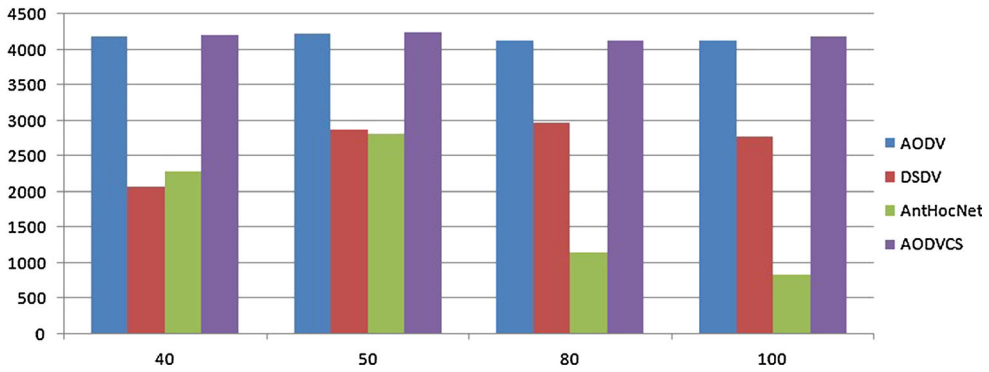

Fig. 5 Received packets results

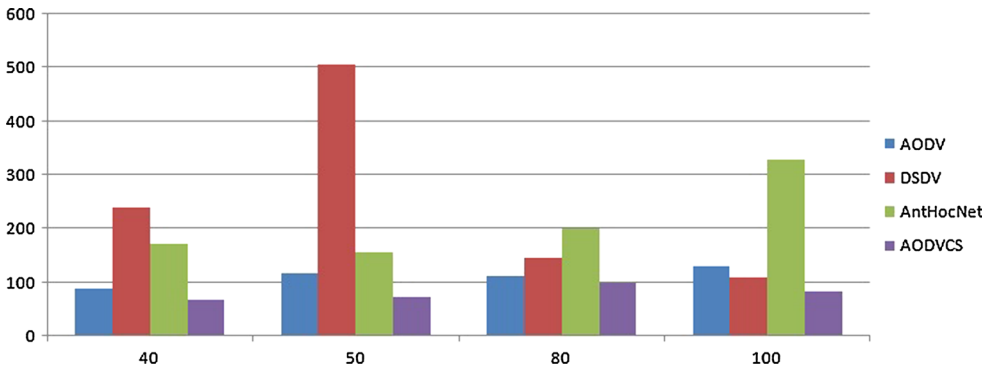

Fig. 6 End-to-end delay results

shows that the proposed scheme is better than the existing schemes with respect to end-to-end delay because AODVCS uses cuckoo search, which is a fast meta-heuristic that provides the best route between two nodes within a reasonable time.

To test and prove the performance of our protocol, we used the NS-2 for implementation. The results of tests and comparisons are very encouraging. The effectiveness of CSA for solving the routing problem in MANET was proven. AODVCS, which is a reactive protocol, is more efficient than AODV in terms of E2ED. AODVCS uses cuckoo search, which is a fast meta-heuristic that provides the best route between two nodes within a reasonable time. AODVCS is also better than DSDV, which is a proactive protocol that updates routing tables thereby slowing it down. Compared with the bio-inspired routing protocol, AntHocNet, that uses many agents to establish a path (forward ant, backward ant, notification and error ant), the results are quite remarkable. The PDR results in AODVCS and AODV we similar, but the result of AODVCS was better than that of DSDV and AntHocNet. The results for different numbers of node show that AODVCS is a scalable

protocol, because it continues to provide good results as the number of nodes increases. The performance of our AODVCS protocol is satisfactory and better than that of AODV, DSDV and AntHocNet in terms of PDR and E2ED. This finding is attributed to the use of cuckoo search which is a fast meta-heuristic that is simple to be implemented. Besides the size of population, P_a is the only parameter to be selected. Such a selection facilitates the flexibility and ease of use of the algorithm compared with other bio-inspired algorithms.

5 Conclusion

We implemented a new reactive routing protocol based on a bio-inspired method called cuckoo search. This method is more powerful than other methods. This study is divided into two phases. The first phase is the implementation of our approach, and the second phase is the creation of a simulation scenario to test and evaluate our routing protocol and compare it with AODV, DSDV and AntHocNet routing protocols. Our protocol, namely, AODVCS is more efficient than other protocols in terms of two performance metrics, E2ED and PDR. We propose conducting additional complex simulations using other mobility models. This protocol can be generalized for other ad hoc networks, such as vehicular ad hoc network (VANET) and flying ad hoc network (FANET).

References

1. Dash, M., & Balabantaray, M. (2014). Routing problem: MANET and ant colony algorithm. *International Journal of Research in Computer and Communication Technology*, *3*(9), 954–960.
2. Perkins, C., & Watson, T. (1994). Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers. In *ACM SIGCOMM conference on communications architectures*.
3. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *IEEE international multi topic conference (INMIC)* (pp. 62–68).
4. Perkins, C., & Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proceeding of WMCSA* (pp. 90–100).
5. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski (Ed.), *The Kluwer International Series in Engineering and Computer Science, Series Vol. 353 of Mobile Computing* (pp. 153–181), Springer.
6. Hass, Z., & Pearlman, R. (2002). *Zone routing protocol for ad-hoc networks*. IETF MANET, draft-ietf-manet-zone-zrp-04.txt.
7. Di Caro, G., Ducatelle, F., & Gambardella, L. M. (2004). AntHocNet: an ant-based hybrid routing algorithm for mobile adhoc networks. In *Proceedings of parallel problem solving from nature (PPSNVIII), Series Vol. 3242 of LNCS* (pp. 461–470) Springer.
8. Yang, X., & Deb, S. (2009). Cuckoo search via Lévy flights. In *Nature & biologically inspired computing* NaBIC'2009 (pp. 210–214), Coimbatore.
9. Chandra, S. J., & Sivarama, P. R. (2015). Trust predicated routing framework with optimized cluster head selection using cuckoo search algorithm for MANET. *IEIE Transactions on Smart Processing and Computing*, *4*(2), 115-125.
10. Kaur, J., & Kaur, G. R. (2014). Performance analysis of AODV and DYMO routing protocols in MANETs using cuckoo search optimization. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(8), 236–247.
11. Gupta, A. K., Sadawarti, H., & Verma, A. K. (2013). Implementation of DYMO routing protocol. *International Journal of Information Technology, Modeling and Computing (IJITMC)*, *1*(2), 49–57.
12. Kout, A., Labed, S., & Chikhi, S. (2015). Netlogo, agent-based tool for modeling and simulation of routing problem in ad-hoc networks. In *The second international conference on advances in information processing and communication technology—IPCT 2015* (pp. 154–160), Roma, Italy.
13. http://www.isi.edu/nsnam//. consulted on June 21st, 2016.

Akram Kout was born in 1989. He got his Master degree from the University of Constantine, in 2012. He is currently a Ph.D. Student in SCAL team (Soft Computing and Artificial Life) of MISC Laboratory (Modeling and Implementation of Complex Systems) at the Constantine 2 University, Constantine, Algeria. His research areas include routing algorithms for mobile ad-hoc networks, graph theory and artificial life.

Said Labed received his Ph.D. in Computer Science from the University of Constantine, in 2013. He is member of the SCAL team (Soft Computing and Artificial Life) of MISC Laboratory (Modeling and Implementation of Complex Systems). His research areas include soft computing, artificial life, distributed systems, routing algorithms for mobile ad-hoc networks, graph theory and parallel computing.

Salim Chikhi received his M.Sc. degree in computer systems from University Mentouri of Constantine, Algeria, in collaboration with Glasgow University, UK, in 1993. He received his Ph.D. degree in computer science from University Mentouri of Constantine, Algeria in 2005. Currently, he is a full professor at the University Constantine2, Algeria. He is the head of MISC laboratory (Modeling and Implementation of Complex Systems) and the leader of the SCAL team (Soft Computing and Artificial Life). His research areas include soft computing and artificial life techniques and their applications to real life problems, namely routing in logistics, biometry, and natural language processing.

El-Bay Bourennane is full Professor of Electronics at the University of Burgundy, Dijon. He is researcher at the LE2I laboratory (Laboratory of Electronics, Computer Science and Image), Dijon, France. His research interest includes Dynamic reconfigurable systems, Image processing, embedded system, methodologies and tools for co-modelling and co-design of intensive signal processing specific embedded systems, FPGA design and Real-time systems.