# Simulation Study of Application Layer Relaying Algorithms with Data-link ARQ in Flying Ad hoc Networks

Danil S. Vasiliev, Kira Vladykina, Albert Abilov
Izhevsk State Technical University
Izhevsk, Russia
danil.s.vasilyev@istu.ru

***Abstract*—In this paper, we propose relaying algorithms and investigate their efficiency at the application layer of flying ad hoc networks (FANETs). We consider two scenarios with a network of two nodes (the source and the destination) and two scenarios with a network of twelve nodes (the source, the destination, and ten nodes that form "the swarm"). We use 802.11n standard at the data-link layer and optimized link-state routing protocol (OLSR) at the network layer of OSI model. We propose chunk-by-chunk, fifty-fifty, and ratio-based relaying algorithms. We compare efficiencies of these algorithms by packet delivery metric (PDR).**

## I. Introduction

Wireless ad hoc networks have dynamic and unstable topology. Nodes in such networks are mobile and each node could route packets to its neighbors. Special routing protocols find and maintain routes between mobile nodes in ad hoc network, e.g., ad hoc on-demand distance vector routing protocol (AODV), optimized link-state routing protocol (OLSR), hybrid wireless mesh protocol (HWMP).

Flying ad hoc network (FANET) is one of many types of mobile ad hoc networks (MANET) [1]. In this network each node is unmanned aerial vehicle (UAV). FANETs are used for monitoring and video surveillance in civil and military missions. On-board camera at the source transmits video data to the destination using wireless channels between flying nodes. A group of flying nodes could work together to complete one task. This group is called "a swarm".

Unpredictable nature of wireless medium and unstable network topology cause low quality of service (QoS) metrics in FANETs. Many approaches and algorithms are proposed to improve QoS metrics in FANETs [2], [3], [4].

Many researchers present approaches to use UAV as a relay node in MANET [5, 6]. We propose new algorithms at the application layer that use flying relay node: fifty-fifty, ratio-based, and chunk-by-chunk. In previous articles we presented analytical description of these algorithms [7] and three approaches to select the relay [8]. In this article we study how proposed algorithms work together with data-link ARQ algorithm and 802.11n standard. We simulate algorithms in NS-3 simulation tool to estimate QoS metric improvement in scenarios with highly mobile nodes (with speeds up to 50 meters per second).

The remainder of this paper is organized as follows: Section 2, routing and error resilience in FANETs; Section 3, relaying algorithms; Section 4, simulated scenarios; Section 5, results; Section 6, conclusion.

## II. Routing and Error Resilience in FANETs

### *A. Routing protocols*

Routing protocols in ad hoc network could be divided in three groups: reactive, proactive, and hybrid. Proactive protocol (e.g., OLSR) uses control messages to update actual information about network topology. Each node maintains routes to all other nodes in ad hoc network proactively. A node that uses reactive routing protocol (e.g., AODV) finds a route to a destination in ad hoc manner. A node broadcasts control messages to neighbor nodes. Neighbor nodes could repeat this broadcast message to other nodes in a network. If this message is received by the destination, the destination will answer the source with special acknowledgment. The source will transmit data to the destination using this newly constructed route. Hybrid protocols (e.g., HWMP) combine both approaches (proactive and reactive) to improve QoS metrics in ad hoc network.

In the article we study and simulate OLSR protocol because it has demonstrated better packet delivery metric than AODV and less overhead than HWMP in our previous research [9]. We simulate OLSR with hop count metric which maintains routes with minimal hop count between nodes in FANET. We will consider ETX (Expected Transmission Count) and ETT (Expected Transmission Time) routing metrics in our future research.

### *B. Error resilience*

We use 802.11n standard at the data-link layer of ad hoc network. 802.11n standard implements automatic repeat-request (ARQ) algorithm for error resilience. This algorithm copes with frame loss caused by collisions in wireless medium. The source transmits frames to its neighbors. If one of neighbor nodes is the destination, it must send positive acknowledgment (ACK) to the source. If the source has received no positive ACK, it retransmits the frame again to the destination. There are three basic ARQ algorithms: stop-and-wait, go-back-N, and selective-repeat. Selective-repeat algorithm demonstrates the best throughput efficiency, but stop-and-wait algorithm is typically used in real devices (e.g., Wi-Fi access points and dongles).

In the article we simulate data-link layer stop-and-wait ARQ with different maximal count of frame retransmissions (*slrc*). We study efficiencies of proposed relaying algorithms that are implemented at the application layer with different *slrc* value at the data-link layer.

## III. Relaying algorithms

To improve QoS in FANETs we propose three algorithms: fifty-fifty, ratio-based, and chunk-by-chunk. In all algorithms only three nodes are used: the source, the destination, and the relay; they are nodes of overlay network. Algorithms are implemented at the application layer of OSI model. The relay node retransmits data from the source to the destination. The destination playbacks received real-time video. Routes between nodes are found with help of OLSR routing protocol at the network layer of OSI model.

Algorithms depend on peer-to-peer interaction between nodes in FANET. The source is a flying node that transmits video data from the on-board camera; the destination is a ground node that receives the video data; the relay is chosen from neighbor nodes in FANET. Proposed algorithms cope with packet loss caused by high node mobility and unstable topology of the network. Each algorithm uses different criteria to choose between direct transmission from the source to the destination and the transmission using the relay. Each algorithm is delay-tolerant and use serial numbers in packet headers at the application layer to identify packet loss. We present all algorithms in pseudo code.

### A. Fifty-fifty algorithm

In fifty-fifty algorithm (algorithm 1) the source transmits the half of all packets using the relay.

**Algorithm 1** Fifty-fifty algorithm at the source

```
initialize IP addresses of the destination and the relay and variable S
while data buffer is not empty do
    pick data from the buffer
    if S=0 then
        send data to the destination
        set S=1
    else
        send data to the relay
        set S=0
    end if
end while
```

If the source transmitted previous packet directly to the destination, then it transmits current packet using the relay. If the source transmitted previous packet using the relay, then it transmits current packet directly to the destination.

### B. Ratio-based algorithm

In ratio-based algorithm (algorithm 2) packet delivery ratio (*PDR*) is calculated. *PDR* is calculated as the number of received positive acknowledgments (ACKs) divided by the number of transmitted packets.

The source receives two ACKs for each transmitted packet. One ACK is received from the destination directly. The variable $N$ is the number of ACKs received from the destination by the source. The destination transmits second ACK to the relay and the relay retransmits this ACK to the source. The variable $M$ is the number of ACKs received from the relay by the source.

**Algorithm 2** Ratio-based algorithm at the source

```
initialize IP addresses of the destination and the relay and variables S, N, M, K
while data buffer is not empty do
    receive ack
    pick ackIPaddress
    if ackIPaddress is the destination then
        N = N + 1
    end if
    if ackIPaddress is the relay then
        M = M + 1
    end if
    if K = 1000 then
        if S=0 then
            if N < M then
                S = 1
            end if
        else
            if N > M then
                S = 0
            end if
        end if
        M = 0
        N = 0
        K = 0
    end if
    pick data from the buffer
    if S=0 then
        send data to the destination
    else
        send data to the relay
    end if
    K = K + 1
end while
```

This algorithm compares two paths every $K$ transmitted packets. If the source transmitted previous 1000 packets directly to the destination and $N < M$, then it transmits next 1000 packets using the relay. If the source transmitted previous 1000 packets using the relay and $N < M$, then it transmits next 1000 packets directly to the destination.

### B. Chunk-by-chunk algorithm

Chunk-by-chunk algorithm compares two paths for each packet. If the source didn't receive ACK for previous packet, it changes the algorithm state. The source receives two ACKs for each transmitted packet. One ACK is received from the destination directly. The destination transmits second ACK to the relay and the relay retransmits this ACK to the source.

When ACK was received from the destination directly by the source, variable *D* is true. When ACK was received from the destination using the relay, variable *R* is true. This algorithm makes decision for every transmitted packet.

**Algorithm 3** Chunk-by-chunk algorithm at the source

```
initialize IP addresses of the destination and the relay and variables S, R, D
while data buffer is not empty do
    receive ack
    pick ackIPaddress
    if ackIPaddress is the destination then
        D = true
    end if
    if ackIPaddress is the relay then
        R = true
    end if
    if S=0 then
        if D = false then
            S = 1
        end if
    else
        if R = false then
            S = 0
        end if
    end if
    R = false
    D = false
    pick data from the buffer
    if S=0 then
        send data to the destination
    else
        send data to the relay
    end if
end while
```

If the source transmitted previous packet directly to the destination and variable *D* is false, then it transmits current packet using the relay. If the source transmitted previous packet using the relay and variable *R* is false, then it transmits current packet directly to the destination.

## IV. SIMULATED SCENARIOS

We study efficiencies of proposed algorithms (fifty-fifty, ratio-based, chunk-by-chunk) using NS-3 simulation tool. We simulate nodes that are connected by wireless standard 802.11n. In this standard data-link ARQ algorithm is used to cope with packet loss caused by collisions in wireless medium. We tested data-link ARQ algorithm with different maximal retransmission count values (*slrc*) in four scenarios: “quadrocopter”, “fixed-wing drone”, “swarm 1”, and “swarm 2”.

### A. Scenario “quadrocopter”

In “quadrocopter” scenario only two stationary nodes were simulated. Node 1 is the source, and node 0 is the destination (Fig. 1).

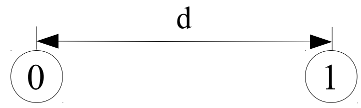



Fig. 1. Scenario “quadrocopter”

In each simulation the distance between nodes is changing from 400 to 560 meters with interval of 10 meters. When the distance is 400 meters, average packet delivery ratio (*PDR*) is 1. When the distance is higher than 560 meters, average packet delivery ratio is 0.

### B. Scenario “fixed-wing drone”

In “fixed-wing drone” scenario simulated network consists of one stationary node (ground station) and one mobile node that imitates a drone with fixed wings. Node 1 is the source, and node 0 is the destination (Fig. 2).

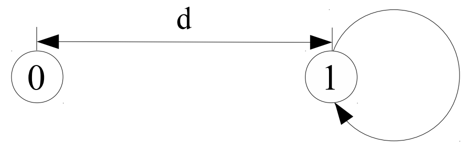



Fig. 2. Scenario “fixed-wing drone”

The source is moving in a circle of radius 50 meters with constant velocity of 50 meters per second. During one simulation of this scenario the distance between the source and the destination is changing from 450 to 550 meters. The wireless channel between the source and the destination is changing during simulation according to this distance. Both nodes in “quadrocopter” and “fixed-wing drone” scenarios use same parameters (Table I). In “quadrocopter” scenario only 1000 packets are transmitted during the simulation time. In “fixed-wing drone” scenario total number of transmitted packets is increased to 10 000 packets. Simulation time in the “fixed-wing drone” scenario is also higher (120 seconds). *PDR* metric is measured in both scenarios.

TABLE I. PARAMETERS OF “QUADROCOPTER” AND “FIXED-WING DRONE” SCENARIOS

| Parameters | quadrocopter | fixed-wing drone |
|---|---|---|
| Simulation time, seconds | 20 | 120 |
| Velocity, meters per second | – | 50 |
| Distance, meters | 400…560 | 450...550 |
| Wireless standard | 802.11n, 5 GHz, MCS1, $P_{Tx}$ = 17.5dBm | |
| Propagation model | Friis | |
| Protocol stack | UDP/IP | |
| Payload rate, mbit per second | 1 | |
| Packet size, byte | 1250 | |

We simulate 5 GHz wireless band because it is recommended for 802.11n wireless standard. This standard is used in modern wireless devices.

### C. Scenario “swarm 1”

To increase quality of service in FANET we propose simple peer-to-peer network. The network consists of three nodes: the source, the destination, and the relay. The relay is used to retransmit packets from the source to the destination.

To study the efficiency of this approach we simulated the ad hoc network of 12 nodes. All nodes are located in the square with side A. The network topology of “swarm 1” scenario is presented in fig. 3 and simulation parameters are presented in table II.

In this scenario the source S could transmit data to the destination D directly (SD path), or the source S could use the relay R to retransmit information to the destination D (SRD path).We could use three different algorithms to choose current path in overlay network from the set {SD, SRD}:

*1) ratio-based:* Calculated packet delivery ratio is used to choose one path over another. The source calculates *PDR* for each group of 1000 packets based on received ACKs, and make a decision about quality of service in overlay network. The path with the best *PDR* metric is selected by the source for the next interval of time. This interval depends on the transmission speed.

*2) fifty-fifty:* All packets with odd packet numbers are transmitted directly and packets with even packet numbers are transmitted through the relay R.

*3) chunk-by-chunk:* The decision to transmit a packet is made based on previous ACK. If no ACK was received for previous transmitted packet, the source changes the path for the current packet.

We simulated three different algorithms: ratio-based, fifty-fifty, chunk-by-chunk. We also simulated transmission without the relay. In Fig. 3 two stationary nodes are located at the corners of the square with the side A. The source S is located at the top-right corner, the destination D is located at bottom-left corner. Ten nodes represent drones: they are moving between the source and the destination following Gauss-Markov mobility model.

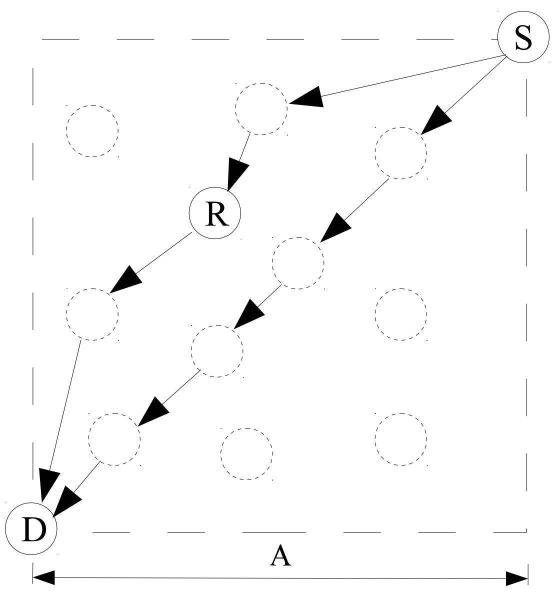



Fig. 3. Scenario “swarm 1”

Coordinates of mobile nodes are bounded by the square and they reflect from its borders without any speed reduction during the simulation run.

In our previous article [8] we estimated QoS for “swarm” scenarios with different relay (or peer) selection algorithms. Predictive algorithm demonstrated the best results; in this algorithm the source used information about node velocities to predict geographical locations of neighbor nodes.

In “swarm 1” and “swarm 2” scenarios the relay R is randomly chosen from the group of mobile nodes (“the swarm”), e.g., IP address of this flying node is manually selected by the user as the relay. This approach has its benefits because the source doesn’t need any additional information about other flying nodes. We study how each relaying algorithm could cope with the mobility of the relay node in simulated scenarios.

In the start of simulation we define random coordinates for all mobile node limited by the size of the square with help of RandomBoxPositionAllocator class in NS-3. Mobility model is realized with help of GaussMarkovMobilityModel class.

We use IP addresses from 10.1.1.0 network (255.255.255.0 mask). Stationary nodes have first two addresses: the source is 10.1.1.2 and the destination is 10.1.1.1. All other addresses are used by mobile nodes. OLSR is used as a routing protocol for all nodes in simulated ad hoc network.

### *D. Fourth scenario*

In “swarm 2” scenario the source S is located in the middle of the square (Fig. 4). The destination D is at bottom-left corner, and the relay is randomly chosen from ten mobile nodes in the start of simulation. Mobile nodes use the same mobility model as in scenario “swarm 1”.

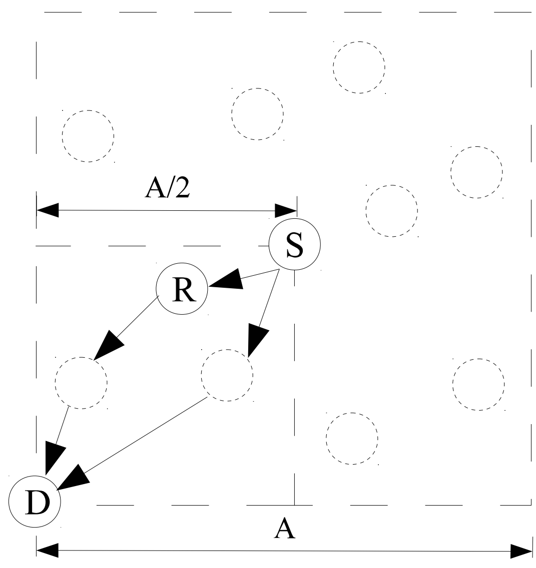



Fig. 4. Scenario “swarm 2”

Simulation parameters are similar for scenarios “swarm 1” and “swarm 2” (Table II). Simulated payload rate is too high to use MCS 0 (BPSK); that’s why we use fixed MCS 1 (QPSK) without link adaptation algorithm to provide identical data-link layer parameters for all nodes during the simulation.

TABLE II. PARAMETERS OF “SWARM 1” AND “SWARM 2” SCENARIOS

| Parameters | Value |
|---|---|
| Simulation time, seconds | 120 |
| Velocity, meters per second | 40-50 |
| Side A, meters | 400…1000 |
| Wireless standard | 802.11n, 5 GHz, MCS1, $P_{Tx}$ = 17.5dBm |
| Propagation model | Friis |
| Protocol stack | UDP/IP |
| Payload rate, mbit per second | 1 |
| Packet size, byte | 1250 |

In scenarios “swarm 1” and “swarm 2” we simulated application layer algorithm and estimated quality of service in simulated networks. We assessed packet delivery ratio (*PDR*) as a quality of service metric. We simulated four different scenarios: “qudracopter”, “fixed-wing drone”, “swarm 1”, and “swarm 2”. We used ratio-based, fifty-fifty, and chunk-by-chunk algorithms to select a path in overlay network for scenarios “swarm 1” and “swarm 2”. We used Gauss-Markov mobility model to simulate highly mobile nodes.

### E. Quality of service metric

We measured packet delivery ratio (*PDR*) to estimate quality of service in simulated networks. The *PDR* metric was calculated as follows:

$$PDR = \frac{Rx}{Tx} \quad (1)$$

where Rx – received packet count, Tx — sent packet count.

We simulated each scenario *N* times with different random seeds. We calculated average packet delivery ratio ($PDR_{ave}$) as follows:

$$PDR_{avre} = \frac{\sum_{i=1}^{N} PDR}{N} \quad (2)$$

where *PDR* – packet delivery ratio for current simulation, *N* – total number of simulations.

## V. RESULTS

### A. Scenario “quadrocopter”

In the “quadrocopter” scenario the source streams data to the destination. Both nodes are stationary. We measured average packet delivery ratio ($PDR_{ave}$) for each simulation run with different values of the distance *d* in the interval from 400 to 560 meters and values of maximal retransmission count *slrc*={1,3,7} (Fig. 5).

The worst results was for *slrc*=1. This parameter regulates the number of retransmissions in the case of frame loss at data-link layer. There are no retransmission of lost frames for *slrc*=1.

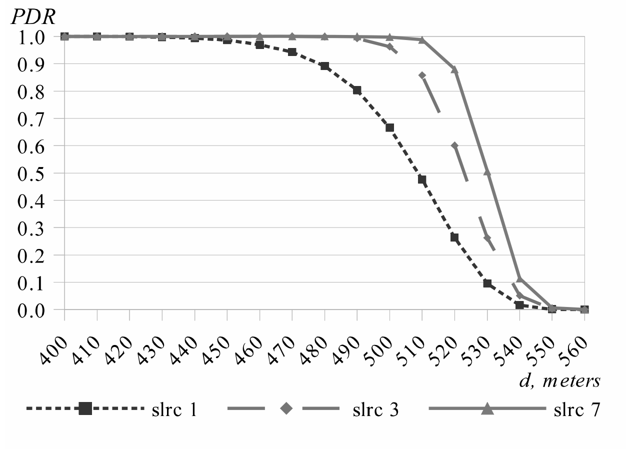



Fig. 5. Measurement results for “quadrocopter” scenario. Average packet delivery ratio ($PDR_{ave}$) for each *slrc* value

The highest parameter value (*slrc*=7) granted the best results for $PDR_{ave}$ metric. This value of parameter is default for many Wi-Fi devices (access points, dongles). Mediocre results was measured for *slrc*=3.

### B. Scenario “fixed-wing drone”

In the “fixed-wing drone” scenario the source was moving in a circle and the destination was stationary. $PDR_{ave}$ metric was calculated for different values of maximal retransmission count *slrc*={1,3,7} (Fig. 6).

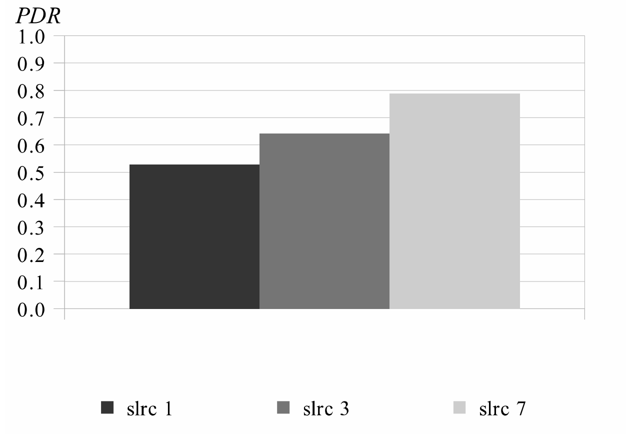
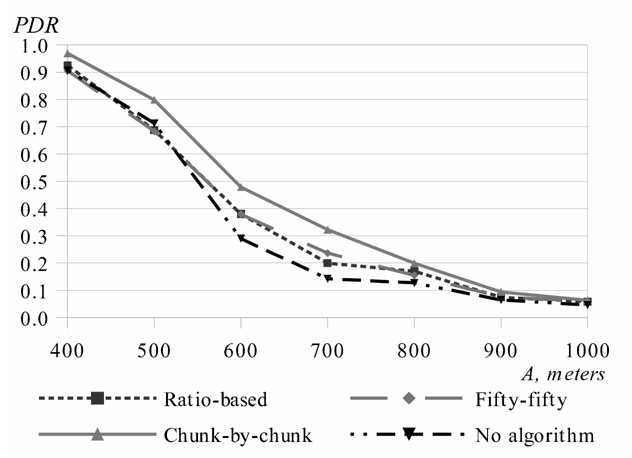



Fig. 6. Measurement results for “fixed-wing drone” scenario. Average packet delivery ratio ($PDR_{ave}$) for each *slrc* value

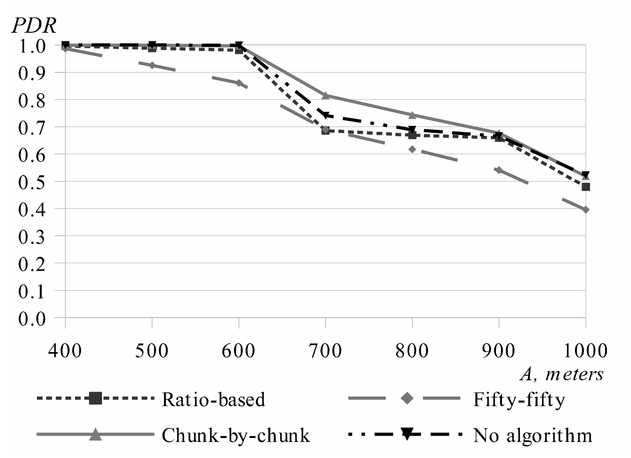
$PDR_{ave}$ values were 0.53 for *slrc*=1, 0.64 for slrc=3, 0.78 for slrc=7. Simulation results demonstrate that we could cope the packet loss caused by node mobility using higher number of frame retransmissions.

### C. Scenario “swarm 1”

Results for “swarm 1” and “swarm 2” scenarios are presented at figures 7, 8, and 9. “No algorithm” line presents PDR metric for data delivery from the source to the destination with help of 802.11n standard and OLSR protocol but without relaying algorithm at the application layer.

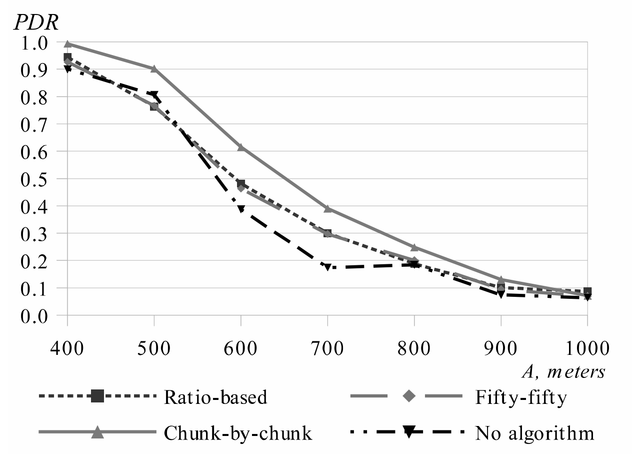



(a) $PDR_{ave}$ for different criteria in "swarm 1" scenario

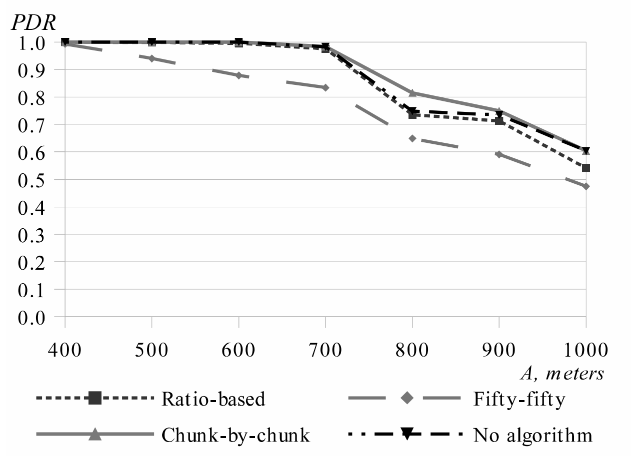



(b) $PDR_{ave}$ for different criteria in "swarm 2" scenario

Fig. 7. Measurement results for "swarm 1" (a) and "swarm 2" (b) scenarios for *slrc*=1

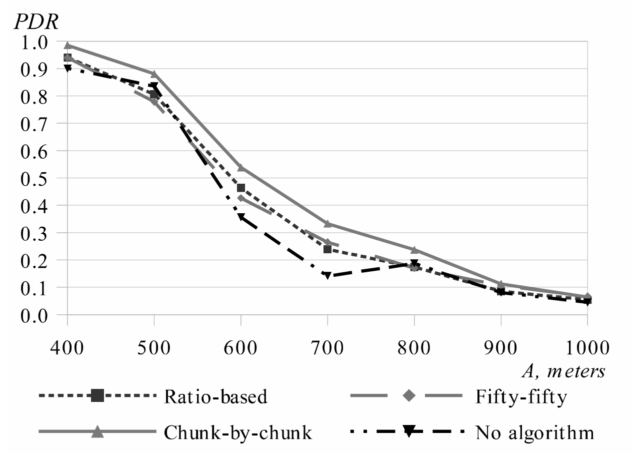



(a) $PDR_{ave}$ for different criteria in "swarm 1" scenario

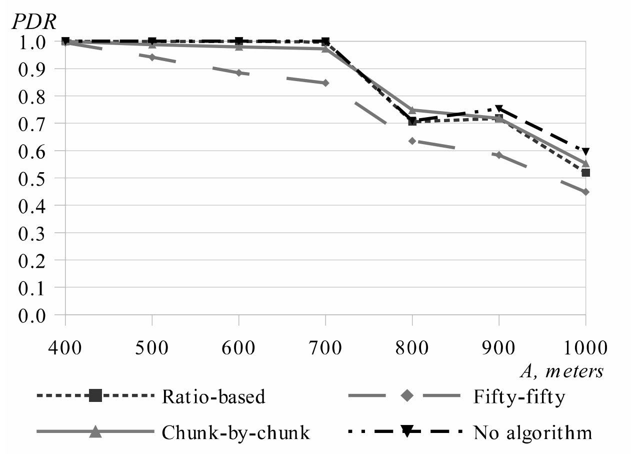



(b) $PDR_{ave}$ for different criteria in "swarm 2" scenario

Fig. 8. Measurement results for "swarm 1" (a) and "swarm 2" (b) scenarios for *slrc*=3




(a) $PDR_{ave}$ for different criteria in "swarm 1" scenario




(b) $PDR_{ave}$ for different criteria in "swarm 2" scenario

Fig. 9. Measurement results for "swarm 1" (a) and "swarm 2" (b) scenarios for *slrc*=7

In “swarm 1” scenario the source (S) and the destination (D) are stationary and located in opposite corners of the square with side A from 400 to 1000 meters; ten nodes move freely between them. As shown in previous scenarios (“quadrocopter” and “fixed-wing drone”) $PDR_{ave}$ depends on the value of maximal retransmission count (*slrc*). In all scenarios we analyzed *slrc*={1,3,7}.

One of ten mobile nodes is randomly chosen by the source as the relay. The source could use ratio-based, fifty-fifty, and chunk-by-chunk algorithms to improve QoS in the network.

Chunk-by-chunk algorithm demonstrated the best results for slrc={1,3,7} and A={400,..,800}. For square side bigger than 800 all algorithms provide $PDR_{ave}$<0.1.

Ratio-based algorithm demonstrated the best $PDR_{ave}$=0.94 for slrc=3 and A=400. The best result for fifty-fifty algorithm was 0.9 (slrc=1, A=400). Chunk-by-chunk algorithm demonstrated $PDR_{ave}$ higher than 0.5 for A=600 meters and *slrc*={1,3,7}. This algorithm is also demonstrates the best $PDR_{ave}$=0.2 for A=800 meters and *slrc*=1.

Normally higher maximal number of frame retransmissions cause higher packet delivery ratio. But big *slrc* value does not always lead to the best result. For example, $PDR_{ave}$ could be 0.6 for slrc=3 (chunk-by-chunk algorithm, A=600 meters), but go down to 0.55 for slrc=7. When square side A is 800 meters $PDR_{ave}$=0.25 for slrc=3, and $PDR_{ave}$ =0.24 for slrc=7.

The same tendency is shown for A=1000 meters. More retransmissions needs higher throughput. But throughput is bounded by the unstable wireless channel between mobile nodes in ad hoc network. That is why higher *slrc* parameter could cause higher packet loss in simulated scenario. For A>500 meters quality of service are low for all algorithms and all *slrc* values. When A is 1000 meters, $PDR_{ave}$ is drastically low for chunk-by-chunk, ratio-based, and fifty-fifty algorithms combined with data-link layer retransmissions up to *slrc*=7, e.g. 0.09 for chunk-by-chunk algorithm (*slrc*=3) and 0.07 for fifty-fifty algorithm (*slrc*=7).

Larger square side causes low quality of service in simulated scenario. Packet loss in this scenario cannot be coped with help of retransmission mechanism at the data-link layer. All proposed application layer relaying algorithms improve QoS in this scenario.

### *D. Scenario “swarm 2”*

“Swarm 1” and “swarm 2” scenarios are very similar, but the source is located in the center of the square in “swarm 2” scenario.

When square side A is 400 meters, no packet loss was detected in simulated scenario for all *slrc* values. When square side A is 600 meters, transmission without the relay demonstrated $PDR_{ave}$=1 for all slrc values. Fifty-fifty algorithm demonstrated the worst results: $PDR_{ave}$=0.86 for *slrc*=1, $PDR_{ave}$=0.89 for *slrc*=7. Ratio-based algorithm demonstrated $PDR_{ave}$=1 for *slrc*=7 only. Chunk-by-chunk algorithm showed $PDR_{ave}$=1 for *slrc*={1,3} and $PDR_{ave}$ =0.98 for *slrc*=7. When square side A is 800 meters, chunk-by-chunk algorithm demonstrated the best $PDR_{ave}$=0.81. Other algorithms showed lower results. The worst $PDR_{ave}$ =0.62 was measured for fifty-fifty algorithm. When square side A is 1000 meters, all algorithms demonstrated results lower than $PDR_{ave}\leq0.6$. Transmission without the relay demonstrated $PDR_{ave}$ =0.6 for *slrc*={3,7}. Fifty-fifty algorithm demonstrated the worst $PDR_{ave}$=0.39 for *slrc*=1.Ratio-based and chunk-by-chunk algorithms showed mediocre results.

In this scenario the distance between the relay and the destination could be greater than the distance between the source and the destination during simulation run. That is why proposed relaying algorithms use direct path to the destination more frequently. This scenario demonstrated that these algorithms could improve quality of service up to $PDR_{ave}$=1 for ad hoc networks bounded by square with side A={500,600,700}.

## VI. CONCLUSION

We proposed chunk-by-chunk, ratio-based, and fifty-fifty algorithms to retransmit data in flying ad hoc network and tested them in simulated scenarios. In this article we demonstrated results for four scenarios: “quadrocopter”, “fixed-wing drone”, “swarm 1”, “swarm 2”. We studied data-link automatic repeat request (ARQ) algorithm in two-node scenarios: “quadrocopter” and “fixed-wing drone”. In these scenarios we increased $PDR_{ave}$ up to 25% increasing the number of packet retransmissions (*slrc*) from 1 to 7. ARQ method effectively improved quality of service in this scenario and could cope with packet loss caused by node mobility.

We study proposed approach to retransmit data in ad hoc network in twelve-node scenarios: “swarm 1” and “swarm 2”. In these scenarios we used the relay to retransmit data stream from the source to the destination (ratio-based, chunk-by-chunk, fifty-fifty algorithms). The best result was showed by chunk-by-chunk algorithm: it increased $PDR_{ave}$ up to 9% in comparison with direct transmission from the source to the destination. This algorithm could be combined with data-link ARQ algorithm to get higher quality of service metric in ad hoc network. But higher *slrc* could lead to higher packet loss and proposed relay technique is vital to get higher $PDR_{ave}$.

The simulation results demonstrated that data-link ARQ algorithm combined with the retransmission technique using the relay could not overcome all packet loss caused by high node mobility in flying ad hoc network.

## ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research project No. 16-07-01201 a.

## REFERENCES

[1] Bekmezci I., Sahingoz O.K., Temel S. Flying Ad-Hoc Networks (FANETs): A survey. / Ad Hoc Networks, vol. 11, no. 3 / – 2013 – pp. 1254–1270.

[2] Koucheryavy, A., Vladyko, A., Kirichek, R. State of the art and research challenges for public flying ubiquitous sensor networks / // *Proceedings of NEW2AN/ruSMART* 2015. LNCS 9247. – Switzerland, 2015. – P. 299–308 / – ISBN 978-3-319-23126-6.

[3] Li, J., Gong, E., Sun, Z., Liu, W., Xie, H. AeroMTP: A fountain code-based multipath transport protocol for airborne networks / Jie

Li, Erling Gong, Zhiqiang Sun, Wei Liu, Hongwei Xie // Chinese Journal of Aeronautics. Volume 28, Issue 4, August 2015 – P. 1147–1162.

[4] Pimentel, L., Rosário, D., Cerqueira, E., Braun, T., Abelém, A., Gerla, M. Context-aware adaptation mechanism for video dissemination over Flying Ad-Hoc Networks / Larissa Pimentel, Denis Rosário, Eduardo Cerqueira, Torsten Braun, Antônio Abelém, Mário Gerla *// 2014 IFIP Wireless Days (WD)* – P. 1–3.

[5] Huo, J.; Xu, Z.; Zhang, Y.; Shan, X. A UAV mobile strategy in mobile ad hoc networks / Jinhai Huo ; Zanxin Xu ; Yaodong Zhang ; Xiuming Shan // *Proceedings of 2011 International Conference on Electronics, Communications and Control (ICECC)*. September 9-11, Ningbo, China, 2011 / – ISBN 978-1-4577-0320-1.

[6] Charlesworth, P.B. Simulating Missions of a UAV with a Communications Payload / Philip B. Charlesworth // Proccedings of 2013 *UKSim 15th International Conference on Computer Modelling and Simulation (UKSim)*. April 10-12, Cambridge, United Kingdom, 2013 –P. 650-655 / – ISBN 978-1-4673-6421-8.

[7] Vasiliev, D. S.; Abilov, A.V. Relaying Algorithms with ARQ in Flying Ad hoc Networks / D. S. Vasiliev, A. V. Abilov // *Proceedings of EEE International Siberian Conference on Control and Communications (SIBCON-2015)*. May 21-23, Omsk, Russia, 2015. – Omsk, 2015.

[8] Vasiliev, D. S.; Abilov, A., Khvorenkov V.V Peer Selection Algorithm In Flying Ad hoc Networks / D.S Vasiliev, A. Abilov, V.V. Khvorenkov // *Proceedings of EEE International Siberian Conference on Control and Communications (SIBCON-2016).* May 12-14, Moscow, Russia, 2016. – Moscow, 2016.

[9] Vasiliev, D. S.; Meitis, D. S; Abilov, A. V. Simulation-based Comparison of AODV, OLSR and HWMP Protocols for Flying Ad Hoc Networks / D. S. Vasiliev, D. S. Meitis, A. V. Abilov // *Proceedings of NEW2AN/ruSMART 2014. LNCS* 8638. – Switzerland, 2014. – P. 245–252 / – ISBN 978-3-319-10352-5.