

# Application Layer ARQ and Network Coding for QoS Improving in UAV-assisted networks

Danil Vasiliev, Andrei Chunaev, Albert Abilov, Irina Kaysina, Daniil Meitis  
Izhevsk State Technical Universtity  
Izhevsk, Russia  
danil.s.vasilyev@istu.ru

**Abstract**—In this paper, we assess the efficiency of developed application layer automatic repeat-request (AL-ARQ) algorithm in UAV-assisted ad hoc network. We emulate HD video streaming using Ubuntu Mate 16 virtual machines, B.A.T.M.A.N. routing protocol, and NS-3 simulation tool. We cope with congestions at the relay node using COPE-like network coding implemented in the routing protocol. The relay node could be stationary like a copter or moving in circle like a fixed-wing drone. AL-ARQ demonstrated packet loss ratio  $PLR_{ave} < 0.01$  for both scenarios for distances between nodes up to 400 meters. In both cases higher  $PLR_{ave}$  caused more retransmissions at the application layer and gave more opportunities for network coding to encode packets at the relay node.

## I. INTRODUCTION

The capabilities of Unmanned Aerial Vehicles (UAVs or drones) which can fly autonomously without an onboard human operator have rapidly evolved. In UAV systems drones should be connected for the data delivery (pictures or live video from on-board camera, control data, GPS coordinates, etc.). A set of drones can be connected with each other and/or a base station using different wireless technologies like Wi-Fi, LTE, mmWave and it can be considered as a set of nodes in a Flying Ad Hoc Network (FANET) [1], [2].

A group of small UAVs can be applied in many tasks. Due to cheap and fast deployment it promises new possibilities for search operations, pollution monitoring, communication in disaster area, etc. High mobility of flying nodes affects the quality of service (QoS) metrics in FANET. Thus FANET has special conditions for data delivery, routing, and applications [3], [4].

FANETs are characterized by highly mobile nodes and wireless communication links with high packet loss ratio ( $PLR$ ). They have frequent link outages according to the positions of UAVs and ground stations. In such networks, links can go down frequent. Hence, it is important to use strong error protection for video traffic in order to achieve an acceptable quality.

In this context one of the most important problem for the real-time video streaming in FANET is the error resilience on the channels and routes between flying nodes. Researchers implement algorithms for QoS improving at all layers of OSI model (e.g., FEC, ARQ, Network Coding). Application layer protocols are very flexible. Many algorithms could be implemented at the application layer without any modernization of lower layers.

There are two well-known approaches to improve QoS metrics: Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). Pure FEC schemes doesn't need any feedback and usually used for real-time video streaming. This coding scheme divides all data into blocks and uses bit or packet redundancy to correct bit or packet errors. Wireless connection causes burst errors and it is challenging to correct these errors by FEC.

ARQ on other hand can handle burst losses much better because it uses a reliable error-correct procedures and can be applied at any layer of OSI model. This method provides feedback-based recovery by retransmission of lost packets indicating them with help of acknowledgement (ACK) or negative acknowledgment (NACK) messages. In ad hoc network we could use a relay to further improve QoS between source and sink nodes.

We consider the FANET of three nodes. Nodes could play three different roles: a source, a sink, and a relay. Two nodes try to exchange data through the relay. Source-node A provides HD video stream to sink-node C through the relay-node B by means of WiFi 802.11n connection. The node C sends the video back to the node A through the same relay-node B. We considered stationary and mobile relay with different distances between nodes and emulation parameters, because UAVs quickly change the position relative to each other that leads to rapid change of the factors influencing the QoS parameters, e.g. packet loss ratio ( $PLR$ ) and burst length distribution.

Ad hoc networks with low reliable transmission channels, e.g. Flying Ad Hoc Networks (FANETs), need new techniques for data transmission. In this article we increase quality of service (QoS) metrics in network coded relaying scenarios using application layer automatic repeat-request algorithm (AL-ARQ).

The main motivation of this paper is to emulate the relay node as a bottleneck for video streams from node A to C and from node C to A. We consider the COPE-like network coding as an approach to cope with congestions at the relay node.

The remainder of this paper is organized as follows: Section II, overview of network coding principles and applications, Section III, application layer ARQ; Section IV, emulated scenarios; Section V, results; Section VI, conclusion.

## II. NETWORK CODING: PRINCIPLES AND APPLICATIONS

### A. Principles

Network coding (NC) is a spectral efficient aid to multicasting [5]. It provides means for the sender to encode information in a way useful to several receivers at once and for each receiver to decode original data. The primary fact upon which NC is based is that transmitted packet could contain repair information for several previously sent packets. A node receives repair packet and has original packets saved in node buffer. A node uses packets from its buffer as a basis to produce new information with help of received repair packets. Information in repair packets is encoded so that it benefits multiple nodes. The main idea of NC is that transmitting time of such repair packet is smaller than a time of independent transmission of several packets. A node needs to transmit lesser data and as result could benefit from better spectral efficiency. In general, for any network the maximal flow value from node A to node B is equal to the minimal cut capacity over cuts separating A and B [6], [7].

Good coordination of state-of-art technologies with NC may permit gains that are not possible in any other way. To date, it can be stated only that NC is a new concept that has many niche applications.

### B. Applications

In order to introduce fundamental concepts of considered approach in a simple manner and to show applications that will be used in emerging networks we classify existing NC types [8]. Let's consider a flow of logically grouped packets. In general a flow is generated by one source, and packets from a flow could be encoded to carry repair information. If encoded packets belong to the same flow, we call it intra-flow NC (or single-source NC). If a node encodes packets from different sources (different flows), we call it inter-flow NC (or multi-source NC). The number of paths between a source and a destination also influences NC; it could be single-path or multi-path NC, correspondingly [9].

The goal of developing network coding was increased throughput and minimized latency. It will be convenient to classify NC architectures according to the OSI model: physical (analog NC, physical NC), data-link layer (COPE, Straightforward NC, BEND), transport (QUIC), and application (R2) layers.

Networks with less rigid protocols (e.g. ad hoc wireless and sensor networks) are prime candidates to implement NC schemes for data collection in sensor networks, data delivery from mobile objects with sensors (like a drone), P2P video streaming, etc. [9]

Although it is possible, by a choice of special routing protocols, to provide a connection between two mobile objects for a considerable amount of the time, multi-hop wireless communication is greatly hampered by low throughput and high interference, and there a lot of situations when it may be impossible. It is clear that NC approach could find its application in wireless networks to benefit from their physical nature.

A technique using the idea of inter-flow NC had already been developed for data-link layer, but a more efficient system

for use at physical layer was devised. If a pair of NC transmitters A and C were to emit signals simultaneously, an intermediate node B would receive them simultaneously, but eventually signals in similar frequency band would collide and transmitted information would be damaged and useless. The idea of NC approach is that whenever such signals collide, the relay B broadcasts received combination of signals. Receivers would decode a payload using data from their buffers [10]. Although scrambled signals caused by multi-source transmission is not very useful they could be made much more so with physical NC. Physical NC developments may make it possible to utilise the packet collisions more directly [11].

Data-link implementation of inter-flow NC is COPE architecture [12]. This architecture is suitable for ad hoc networks and improves its overall throughput. In such network all nodes use broadcast transmission, each node stores overheard packets in a buffer and uses a bitmap in a header to inform neighbours. A node XORs multiple packets into a single packet. Destination node decodes the packet using XOR with the packets from its buffer.

Inter-flow NC (e.g. PNC and COPE) is an important approach in the most ad hoc systems. For example, it could be used for data delivery using backbone of cars in VANET [13]. COPE is easier to implement using existent technologies, but PNC grants higher coding gain. Novel data-link NC algorithms, e.g. BEND, give additional coding opportunities using multiple relay nodes instead of one.

On the other hand, when the intra-flow NC came into service it found an application as P2P video streaming scheme. To deal with multi-sink transmission, and to simplify data dissemination process, packets of one flow are encoded between themselves. While in common P2P streaming system video is downloading in small fragments from a group of peers and each video segment is acknowledged by sinks independently, it was announced that the protocol based on random linear network coding (RLNC) were considering larger segments, or generations. Generations contain linear combinations of packets that belong to the same flow and each generation is served by multiple seeds [14]. This type of NC allows to simplify data delivery process: a source pushes linear-combined repair packets to sinks, and sinks send acknowledgement for each generation of packets (not for each segment like in TCP). Hence intra-flow NC may be suitable for high data-rate D2D applications [15], [16].

The advent of NC approach has brought its new applications. Ad hoc network of flying nodes might be optimized as any other network with help of both inter- and intra-flow NC to achieve higher throughput.

### C. Network Topologies

We consider simple network topology of three nodes. Two users send HD video to each other through flying relay, e.g. a copter or a fixed-wing drone. Such ad hoc scenario is useful when more robust network infrastructure is not accessible (e.g. in disaster area). Actually, inter-flow NC produces coding gain only in networks with specific topologies.

Suppose the relay B, located between nodes A and C, retransmits encoded data to both neighbors, and packets are travelling in both directions: from A to C and from C to A.

Special NC algorithm at a relay B, which combines received (or overheard) information into repair packets, seeks coding opportunities. Nodes A and C are both sources and sinks. By a complete departure from the conventional scheme, inter-flow NC (both PNC and COPE) effectively reduces the number of mandatory time slots during the transmission in this network: the relay needs only one time slot to transmit repair packet instead of two time slots for two regular packets. To decode new data definitely, the receiver must also have the previously sent packet in its buffer. This pair of packets (previous packet in the buffer and received repair packet) is used to decode and obtain new information. Every repair packet transmitted by a relay represents a pair of XORed packets (one from each neighbor). Conversely, pairs of sources and sinks are required to provide the relay with a coding opportunity. So arranged, the nodes make up a triplet (two-way-relay channel, TWRC) [17], [18].

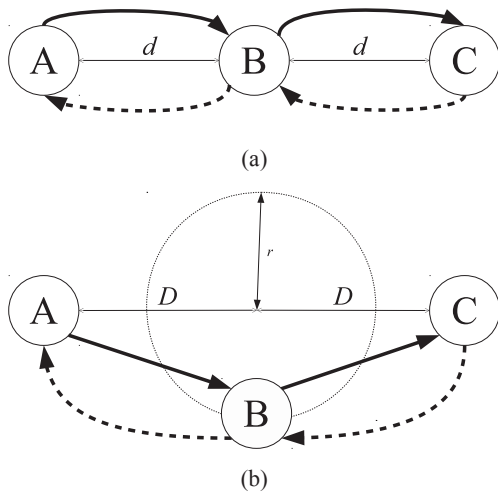


Fig. 1. Emulated network topologies with stationary (a) and mobile relay (b).

Four or more nodes may similarly form three or more triplets, or even a chain. There are many different NC atoms: the triangle, the wedge, the cross, and the star [17]. Network topology could be decomposed based on these atoms to benefit from opportunistic COPE, BEND or PNC approach. We consider COPE-like network coding, because it is easier to implement in scheme with only one relay and 802.11n standard.

#### D. Network Coding and Quality of Service

In [19] authors consider novel ARQ scheme for cooperative wireless networks. The scheme adopts network coding techniques in order to enhance the total bandwidth of the network by minimizing the total number of transmissions. The performance of the proposed approach is evaluated by means of computer simulations and compared to other cooperative schemes, while an analytical solution is provided to validate the results.

In [20] the performance of multipath routing with and without NC for various motion and packet loss scenarios is

compared via simulation. Path redundancy and/or NC are activated when packet loss is severe.

In this article we analyze efficiency of COPE-like network coding and developed application layer ARQ (AL-ARQ) algorithm in UAV-assisted networks.

### III. APPLICATION LAYER ARQ

Wireless medium causes many challenges for packet delivery. To improve the data transmission quality, one can apply such methods for packet loss recovery as Automatic Repeat Request (ARQ) and Forward Error Correction (FEC) at various OSI layers. In 802.11n standard FEC and ARQ are used at data link layer. TCP implements ARQ approach at transport layer [21],[22],[23],[24]. In the article, we consider implementation of ARQ at the application layer and analyze relationships between selective-repeat ARQ and COPE-like network coding in UAV-assisted networks.

Proposed application layer ARQ (AL-ARQ) algorithm uses the selective-repeat approach. In this approach, the destination finds lost packets by their sequence numbers (SN) and requests them from the source repeatedly. Thus, the destination forms negative acknowledgement (NACK) that includes SNs of lost packets as shown in fig. 2. If requested packet is not received, then, if the retransmission timeout (RTO) expires, the destination sends NACK again to request lost packet until that packet is not relevant for the application.

We use two buffers: one is implemented on the source to store packets for retransmission; the other is implemented on the destination and used to maintain packet order that can be affected by channel jitter and lost packets recovery delay.

AL-ARQ algorithm calculates RTO in four steps based on approximation of Round Trip Time (RTT) statistics:

1) Current Smoothed Round Trip Time ( $SRTT_{cur}$ ) is calculated with priority of last values:

$$SRTT_{cur} = 7/8 \times SRTT_{prev} + 1/8 \times RTT_{cur}, \quad (1)$$

where  $SRTT_{prev}$  – previous SRTT,  $RTT_{cur}$  – current RTT.

2) Current deviation  $DEV_{cur}$  is determined:

$$DEV_{cur} = |RTT_{cur} - SRTT_{prev}|. \quad (2)$$

3) Then current smoothed deviation  $SDEV_{cur}$  is obtained by the following equation:

$$SDEV_{cur} = 3/4 \times SDEV_{prev} + 1/4 \times DEV_{cur}. \quad (3)$$

4) Finally, RTO is calculated using the following formula:

$$RTO = SRTT_{cur} + 4 \times SDEV_{cur}. \quad (4)$$

Fig. 3 shows the message formats that are used in the proposed algorithm. All fields are related to the application layer. Burst loss is typical in wireless networks, so AL-ARQ algorithm combines few burst losses in one NACK message. For application layer tree and mesh network structures AL-ARQ algorithm needs global  $SN_G$  field, at the same time  $SN$  field is used for loss detection between neighbor nodes.

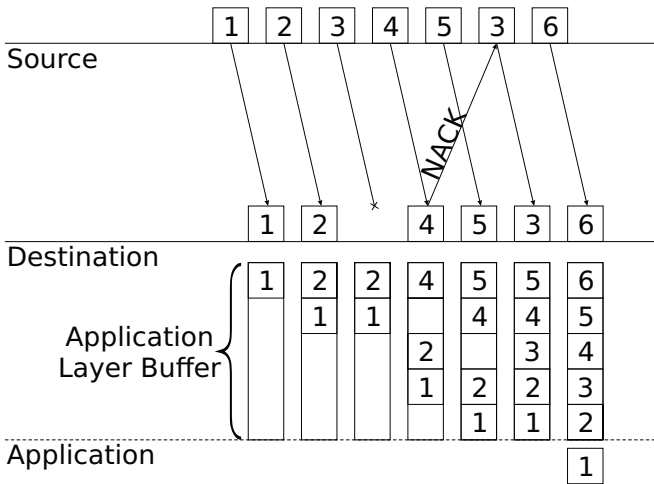


Fig. 2. AL-ARQ recovery principles

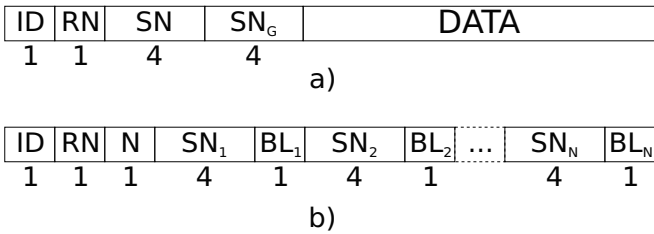


Fig. 3. AL-ARQ message formats a) for data transferring and b) for NACK and cancellation message: RN – retransmission number, SN – sequence number,  $SN_G$  – global SN that can be used packet loss detection in tree and mesh topologies; N – burst loss number, BL – burst loss length. Numbers under the fields indicate the size of the corresponding field in bytes.

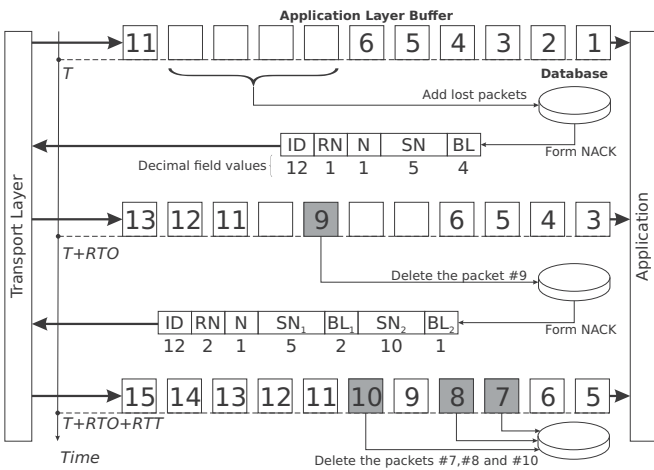


Fig. 4. AL-ARQ algorithm behavior on the destination in the case of burst loss. Recently recovered packets are shown in gray color

Lost packets, affected by this burst, form the group of lost packets at the destination. AL-ARQ divides the group of lost packets into several bursts of length  $BL_1, BL_2 \dots BL_N$ . Sometimes source buffer doesn't contain packets needed by the destination. In this case, the source sends cancellation message that contains SNs of non-existent packets as a response to incoming NACK. The destination deletes these SNs from lost packets database.

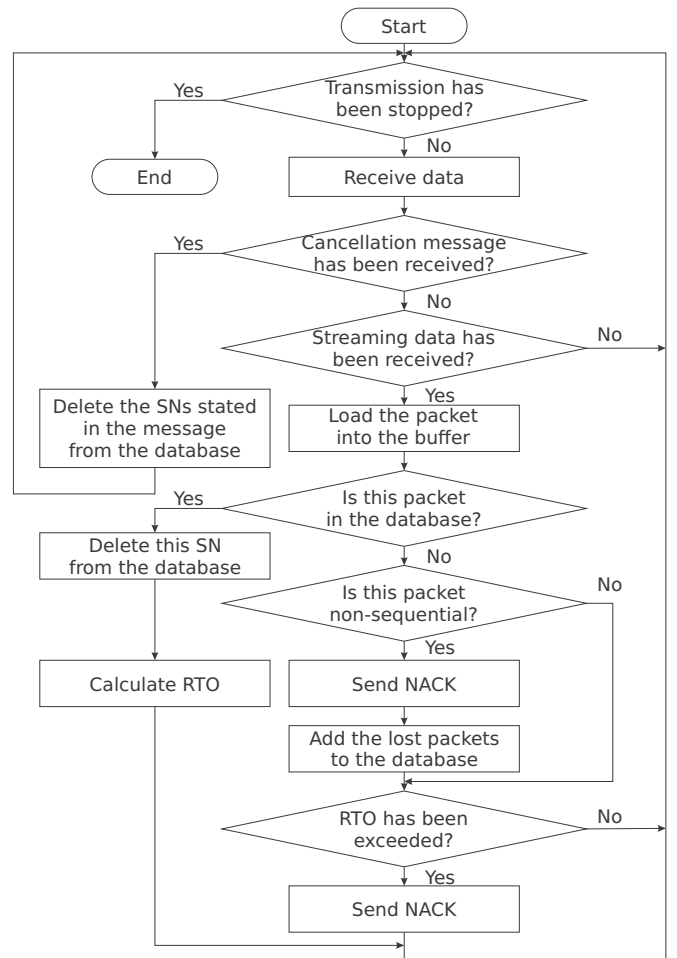


Fig. 5. AL-ARQ algorithm flowchart

Fig. 4 illustrates AL-ARQ algorithm behavior in the case of burst loss. The destination adds SNs of lost packets, burst loss number, and retransmission number to the database and forms NACK based on these values. Later, one of requested packets arrives and the destination removes its SN from the database. Initially the destination uses default RTT value (50 ms), but later it calculates this timeout more precisely. If the RTO expires, the destination forms new NACK that contains two or more  $SN_i$  and  $BL_i$  fields and sends it again. The destination repeats NACK steadily until those packets are not relevant for the application or until it gets cancellation message from the source. Fig.5 shows the flowchart of the algorithm.

#### IV. EMULATED SCENARIOS

We developed the testbed to estimate the effectiveness of a combination AL-ARQ with COPE-like network coding. The testbed includes emulated nodes and wireless transmission medium. We have implemented emulated nodes as virtual machines with the Ubuntu Mate 16.04 operation system.

##### A. Protocol Stack

The virtual machines run B.A.T.M.A.N. routing protocol, which implements the network coding algorithm (Table I). We enabled COPE-like network coding in B.A.T.M.A.N. using the option "CONFIG\_BATMAN\_ADV\_NC=y".



TABLE I. EMULATION PARAMETERS

Parameter Name	Value
Operating System	Ubuntu Mate 16
Application Layer	Selective-Repeat ARQ
Video Coding	H.264
Transport Layer	UDP
Routing Protocol	B.A.T.M.A.N.
Network Coding	COPE
Wireless Standard	802.11n

We emulate the topology of three nodes (fig.1) to test the efficiency of video streaming from embedded laptop HD camera using *gststreamer* software based on H.264/UDP/BATMAN&COPE/802.11n protocol stack. 802.11n standard and the transmission medium are emulated in NS-3 with help of virtual TAP interface in VMs. The first UDP-stream is from node A to node C, and the second UDP-stream is from node C to node A. There is no direct connection between nodes A and C, thus they should use the relay node B to transmit video data. We performed emulation runs of AL-ARQ algorithm using the relay in emulated environment. The transmission results were measured for different protocol and channel parameters. In first scenario the relay node was stationary (fig.1a). In the second scenario the relay node was moving in circle with speed of 50 meters per second and radius  $r=50$  meters (fig.1b).

### B. Packet Loss Evaluation

Different metrics can be used to evaluate quality of service (QoS) between the source and the destination, e.g. packet loss ratio, frame loss ratio, delay, etc. We measured packet loss ratio (*PLR*) that shows the relation between lost and transmitted packets.

The sink calculates *PLR* with help of application layer sequence numbers (SN) described in Section III. To calculate *PLR* between the source and the sink we use SN field. SN field defines current SN and previous SN. In addition, the sink defines measurement interval value on its own (1000 packets by default). The source delivers original packets strictly ordered by their SNs. Original packets are marked with zero-value in RN field (fig.3). The sink counts incoming original packets by their SNs, so it can detect lost packets. If the source repeats a packet as a response to incoming NACK, it must use non-zero RN field to indicate retransmission number. The sink stores newly arrived packets ordered by SN in its buffer, arranges lost packets in groups in the database and uses these groups to form NACKs to the source.

*PLR* is calculated for the packets that sent from sink buffer to the *gststreamer* (Fig. 4). Thus, the destination can measure *PLR* after packet recovery and give an insight on the efficiency of AL-ARQ. Difference between current and previous SNs determines burst loss. Out-of-sequence packet is considered as packet loss.

In all emulation runs *PLR* value were measured for every 1000 transmitted packets based on sequence numbers as described above. Average  $PLR_{ave}$  values were calculated to find dependencies between packet loss and distances between nodes in considered network topology.

### C. End-to-End Delay and Throughput

End-to-end delay of video transmission depends on parameters of AL-ARQ algorithm. In AL-ARQ buffer size *buf* defines the maximal application layer delay for video transmission. We considered *buf* values in range from 0.5 to 3 seconds for buffers at the source and the sink. COPE-like network coding also negatively affects end-to-end delay due to time-consuming encoding process at the relay. AL-ARQ calculates current RTT between the source and the destination during the transmission (default value is 50 ms) and corrects current retransmission timeout value.

Throughput of considered system is limited by physical layer technology. We considered 802.11n standard with modulating-coding schemes (*MCS*) 1 and 2. *MCS1* and *MCS2* use QPSK modulation and give throughput of 13 Mbps and 19.5 Mbps data rates (without MIMO), respectively.

### D. Network Coding Metrics

COPE-like network coding opportunistically encodes packets from two video streams at the relay node. During the emulation the relay has opportunities to broadcast one encoded packet instead of two unicast packets. We measure the number  $N$  of encoded packets at the relay node B and the fraction  $F(\%)$  of encoded data to all packets at the relay node during the emulation run. These metrics demonstrate the dependency of network coding opportunities on the distance between nodes  $d$  and buffer size *buf* of AL-ARQ. Each encoded packet directly affect spectral efficiency, because the relay needs only one transmission of encoded packet instead of two transmissions of unicast packets.

## V. RESULTS

We tested an ad hoc network composed of three virtual machines (VM) with Ubuntu Mate 16 operating system. Wireless channels between VMs were emulated with help of NS-3 simulation tool as 802.11n standard. We streamed a H.264 compressed video from node A to node C with HD quality (i.e., resolution: 1280x720). Node C streamed received video back to the node A. Node B was the relay between nodes A and C. Node B encoded received packets using COPE-like network coding. We used AL-ARQ approach at nodes A and C and carried out scenarios with different parameters of modulating-coding scheme (*MCS*), AL-ARQ buffer size (*buf*), video quality, and distance  $d$  between nodes.

Emulation results are presented in figures 6 to 12. We calculated three metrics: average packet loss ratio  $PLR_{ave}$ , the number  $N$  of encoded packets at the relay node B, and the fraction  $F(\%)$  of encoded data to all packets at the relay node during the emulation run. Figure 6 shows the relationship between  $PLR_{ave}$  metric and the distance  $d$  between nodes in network coded relaying scenario with stationary relay. Distances of 300, 400, and 500 meters were considered. HD-video was transmitted at *MCS2* during 1 hour emulation.  $PLR_{ave}=0.075$  is the highest for distance  $d$  of 500 meters. With help of AL-ARQ application with the buffer size  $buf=0.5s$  the metric dropped down to near zero in cases of 300 and 400 meters. For distance of 500 meters  $PLR_{ave}=0.007$ .

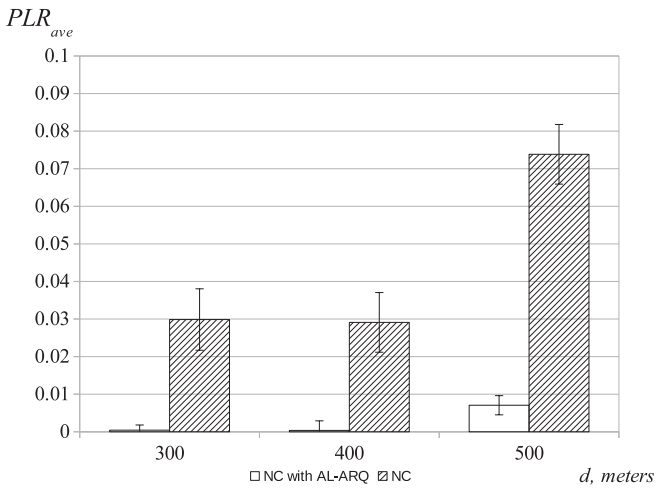


Fig.6. Average packet loss ratio  $PLR_{ave}$  in network coded relaying scenario with stationary relay and AL-ARQ algorithm for various distances between nodes (720p, MCS2,  $buf = 0.5s$ )

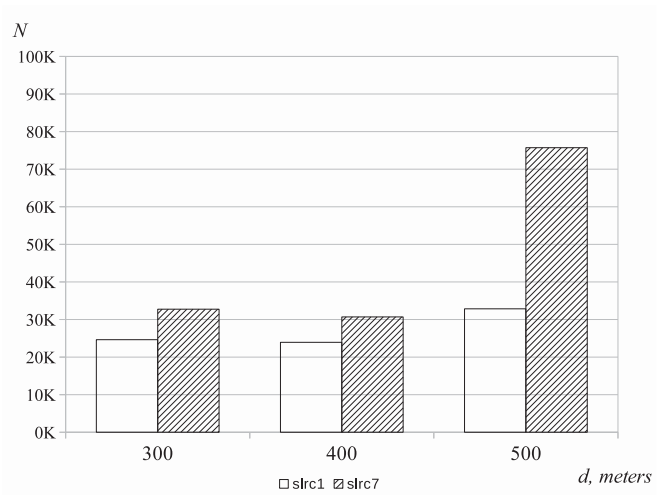


Fig. 9. Total number  $N$  of encoded packets at the stationary relay node in network coded scheme with AL-ARQ in scenarios with enabled (slrc7) and disabled (slrc1) data-link layer ARQ (720p, MCS2,  $buf=2s$ )

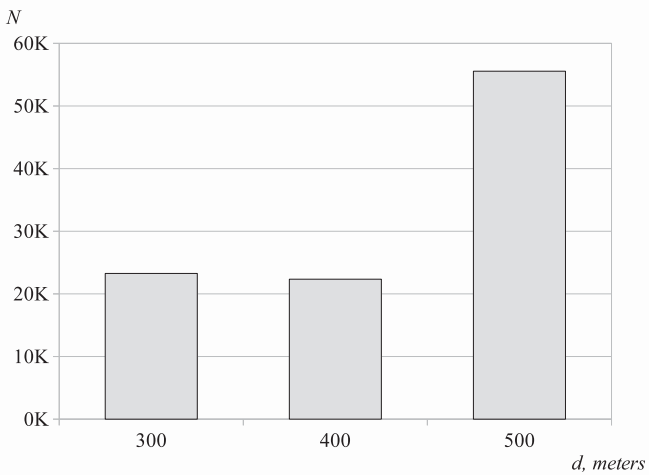


Fig.7. Total number  $N$  of encoded packets at the relay node in network coded relaying scenario with stationary relay and AL-ARQ algorithm for various distances between nodes (720p, MCS2,  $buf = 0.5s$ )

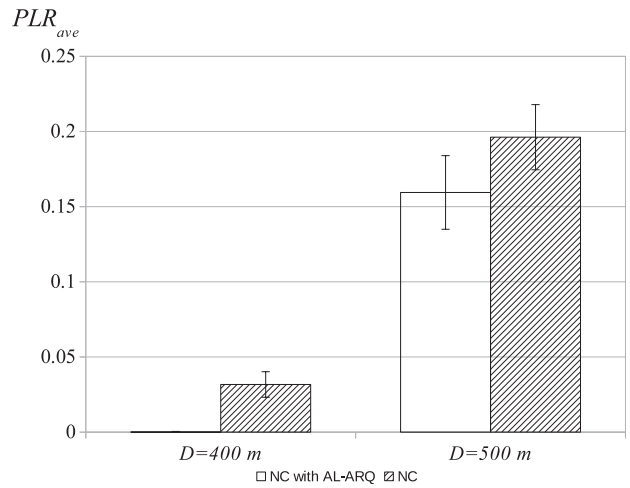


Fig. 10. Average packet loss ratio  $PLR_{ave}$  in network coded relaying scenario with mobile relay and AL-ARQ algorithm nodes (720p, MCS2,  $buf = 1s$ )

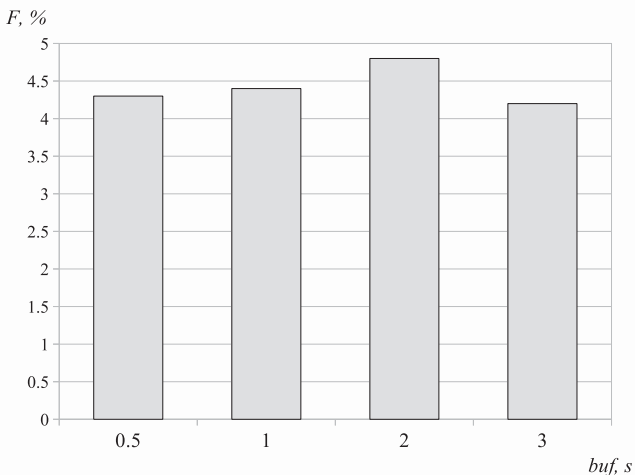


Fig. 8. The fraction  $F$  (%) of encoded packets at the relay node in network coded relaying scenario with stationary relay and AL-ARQ algorithm for distance between nodes  $d=500$  meters (720p, MCS2)

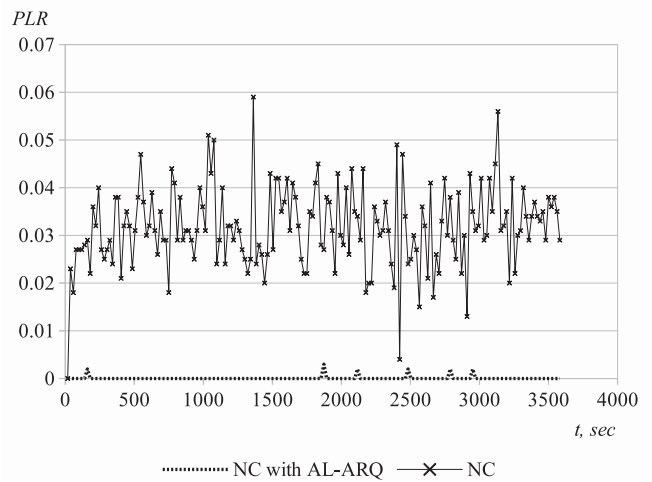


Fig. 11. Instant packet loss ratio values  $PLR$  at the node A in network coded relaying scenario with mobile relay and AL-ARQ algorithm for  $D=400$  meters (720p, MCS2,  $buf = 1s$ )

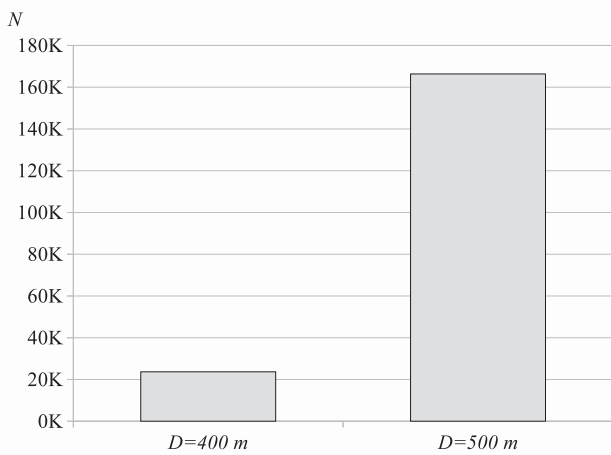


Fig. 12. Total number  $N$  of encoded packets at the relay node in network coded relaying scenario with mobile relay and AL-ARQ algorithm for various distances between nodes (720p, MCS2,  $buf=1s$ )

Figure 7 shows the total number  $N$  of encoded packets at the relay node in network coded relaying scenario with stationary relay for the same scenario. The highest  $N=55000$  was measured for  $d=500$  meters.

To recover more lost packet we could make AL-ARQ buffer larger at the source and the sink, e.g.  $buf=2s$ . For larger buffer in the same scenario and same distance  $PLR_{ave}=0.003$ . The total number of encoded packets  $N$  grows up from 55548 to 75715 due to higher number of retransmissions during the emulation run. Figure 8 shows that if we enlarge the size of AL-ARQ buffer further and set the distance  $d=500$  meters, MCS2, and  $buf=3s$  the fraction  $F$  of encoded packets would go down from 4.8% to 4.2% (54295 packets). When we set  $buf=2s$  and disabled data link layer ARQ (fig.9), we measured the number of encoded packets twice lower ( $N=32843$ ) than with enabled ARQ for distance  $d=500$  meters. With disabled data-link layer ARQ the metric  $N$  was lower for all measured points.

The scenario with mobile relay was emulated for distance  $D=\{400, 500\}$  meters. Node B was moving in circle, and  $D$  was the distance between the center of that circle and other nodes.

Figure 10 demonstrates that for  $D=400$  meters all lost packets were recovered, but for  $D=500$  meters the QoS metric was too low for video streaming. Figure 11 shows instant values of  $PLR$  metric for each thousand packets during the emulation run for  $D=400$  meters. Average  $PLR_{ave}$  dropped from 0.032 to 0.000 for this scenario due to AL-ARQ. The metric  $PLR_{ave}=0.196$  in the case of  $D=500$  meters was too high to correct it by AL-ARQ algorithm.

Figure 12 shows that the metric  $N$  was much higher for  $D=500$  meters than for  $D=400$  meters, because more retransmissions caused higher throughput demand and higher opportunities for encoding of packets at the relay node. The fraction  $F$  of encoded packets was 2.4% and 4.5% for  $D=400$  and  $D=500$ , respectively.

During the emulation run the number  $N$  and the fraction  $F$  (%) were affected by the video quality of the source. We tested 320p, 640p, and 720p video sources. For example, for video quality 320p  $N$  was lower than 5000 packets and  $F<1\%$ . Highest number  $N=96716$  ( $F=6.1\%$ ) of encoded packets at the  $d=700$  meters and MCS1 for 720p video. Emulations demonstrated efficiency of developed application layer ARQ algorithm for network coded relaying scenarios with 802.11n standard.

## VI. CONCLUSION

We analyzed quality of service (QoS) metrics UAV-assisted ad hoc network of three nodes. The relay node was a copter (stationary) or a fixed-wing drone (was moving in circle) and used COPE-like networks coding, while other two nodes used application layer selective-repeat ARQ (AL-ARQ) to improve QoS metrics. Two nodes communicated through the relay with help of 802.11n standard. The relay node opportunistically encoded received packets to effectively use available channel.

Emulation results demonstrated higher number of encoded packets, e.g.,  $N>30000$ , in scenarios with higher packet loss rate  $PLR_{ave}>0.05$ . In our scenario data-link layer implementation of COPE-like network coding is used with application layer selective-repeat ARQ. We also used UDP protocol that lacks any congestion control in contrary to TCP. In considered scheme the relay node coped with demand in higher throughput using network coding.

Results show that applying ARQ retransmission at the application layer can improve significantly the QoS for the video streaming. COPE-like network coding could be used to retransmit packets in more effective way. Further investigation of video streaming with help of network coding and ARQ methods are needed to improve QoS in ad hoc networks with dynamic topologies.

## ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research project № 18-37-00218 mol\_a.

## REFERENCES

- [1] P. Bok Y. Tuchelmann, "Context-Aware QoS Control for Wireless Mesh Networks of UAVs", Computer Communications and Networks (ICCCN), in Proceedings of 20th International Conference, 2011
- [2] R. Beard, T. McLain, D. Kingston, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs", Proc. IEEE – Special Issue Multi-Robot Syst., vol. 94, no. 7, pp.1306-1324, Jul. 2006
- [3] I. Bekmezci, O. Sahingoz and S. Temel, "Flying Ad-Hoc Networks (FANETs): A survey", in Ad Hoc Networks, no. 11, 2013.
- [4] O. Sahingoz, "Mobile Networking with UAVs: Opportunities and Challenges", in Proceedings IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 933-941, May. 2013
- [5] Ahlswede, R., Cai, N., Li, S.-Y., Yeung, R.: Network information flow. IEEE Trans. Inform. Theory 46, 2000, pp. 1204–1216.
- [6] D. R. Ford and D. R. Fulkerson, Flows in Networks. Princeton, NJ, USA: Princeton University Press, 2010.
- [7] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," IEEE Transactions on Information Theory, vol. 25, pp. 572–584, September 1979.

- [8] T. Matsuda, T. Noguchi, and T. Takine, "Survey of network coding and its applications," *IEICE Transactions on Communications*, vol. E94.B, no. 3, pp. 698–717, 2011.
- [9] B. Adamson, C. Adjih, J. Bilbao, V. Firoiu, F. Fitzek, E. Lochin, A. Masucci, M.-J. Montpetit, M. V. Pedersen, G. Peralta, V. Roca, P. Saxena, and S. Sivakumar, "Network Coding Taxonomy," Internet-Draft draft-irtf-nwcrng-networkcoding-taxonomy-05, Internet Engineering Task Force, July 2017. Work in Progress.
- [10] J. Goseling, M. Gastpar, and J. H. Weber, "Random access with physical layer network coding," *IEEE Transactions on Information Theory*, vol. 61, pp. 3670–3681, July 2015.
- [11] S. Zhang, S. C. Liew, and P. P. Lam, "Physical layer network coding," *CoRR*, vol. abs/0704.2475, 2007.
- [12] Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., Crowcroft, J.: XORs in the air: practical wireless network coding. *IEEE/ACM Trans. Netw.* 16, 2008, 497–510.
- [13] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, "How to utilize interflow network coding in vanets: A backbone-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2223–2237, Aug 2016.
- [14] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 1655–1666, December 2007.
- [15] G. Araniti, M. Condoluci, P. Scopelliti, A. Molinaro, and A. Iera, "Multicasting over emerging 5g networks: Challenges and perspectives," *IEEE Network*, vol. 31, pp. 80–89, March 2017.
- [16] A. Pyattaev, O. Galinina, S. Andreev, M. Katz, and Y. Koucheryavy, "Understanding practical limitations of network coding for assisted proximate communication," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 156–170, Feb 2015.
- [17] J. He and S. C. Liew, "Building blocks of physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 2711–2728, May 2015.
- [18] D.S. Vasiliev, I.A. Kaysina, A. Abilov, "Performance evaluation of COPE-like Network Coding in Flying Ad Hoc Networks: Simulation-based Study", *Proceedings of 17th International Conference NEW2AN 2017 10th Conference ruSMART 2017 Third Workshop NsCC 2017*, pp. 577-586, August 28–30, 2017.
- [19] A. Antonopoulos and C. Verikoukis, "Network Coding-Based Cooperative ARQ Scheme," *2011 IEEE International Conference on Communications (ICC)*, Kyoto, 2011, pp. 1-5.
- [20] S. Y. Oh, M. Gerla and A. Tiwari, "Robust MANET routing using adaptive path redundancy and coding," *2009 First International Communication Systems and Networks and Workshops*, Bangalore, 2009, pp. 1-10.
- [21] Sundararajan, J.K., Shah, D., Medard, M., Jakubczak, S., Mitzenmacher, M., Barros, J.: Network coding meets TCP: theory and implementation. *Proc. IEEE* 99(3), 2011, pp. 490–512.
- [22] Huang, Y., Ghaderi, M., Towsley, D., Gong, W.: TCP performance in coded wireless mesh networks. In: 2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, San Francisco, CA, 2008, pp. 179–187
- [23] Khaled Alferaidi, Robert Piechocki: TCP-MAC Cross Layer Integration for Xor Network Coding. Springer Nature Switzerland AG 2019, pp. 860–875.
- [24] F. Gabriel, S. Wunderlich, S. Pandi, F. H. P. Fitzek and M. Reisslein, "Caterpillar RLNC With Feedback (CRLNC-FB): Reducing Delay in Selective Repeat ARQ Through Coding," in *IEEE Access*, vol. 6, pp. 44787-44802, 2018. doi: 10.1109/ACCESS.2018.2865137