With varying packets size and time interval, the linear increase in the size of packets, enables the receiver to determine which packets have been received and which one was lost on the way [4]. Therefore, the receiver can locate any packet in the probing train when knowing the size of that packet. The relationship between the packets *order* and *size* within the probing train of 30 packets is almost incrementally linearly proportional, taking a packet size difference of $\Delta P = 50$ bytes.

– *Variable time interval:* Here the time interval between the probing train packets is also not fixed, but decreases linearly. In other words, if the time interval at packet i is T_i, and this interval at the following packet $i + 1$ is $T_{i+1} = T_i - \Delta T$ according to the following:

$$T_1 > T_2 > T_3 \ldots > T_{N-2} > T_{N-1} \quad (1)$$

The relationship of time interval between probing packets and the order of these packets is decreasingly ascendingly proportional, taking a time interval $\Delta T = 0{,}5$ ms.

The receiver should have a prior knowledge about the time intervals between the probing train packets when they were sent, and should register arrival time of each packet. The receiver can locate any received packet through its size, so the duty of the receiver is to compare time intervals between probing packets when sending them (already available), and when received. While sending the probing packets at a rate equal or less than the available bandwidth value $Rsend \le Abw$, the time intervals between consecutive received probing packets are equal to those between consecutive sent probing packets. If we neglect the delay experienced by probing packets while passing through the network (delay suffered by all packets and it is the time they need to cross the network path), i. e.:

$$T_{1,snd}=T_{1,rcv};\ T_{2,snd}=T_{2,rcv};\ \ldots;\ T_{N\text{-}1,snd}=T_{N\text{-}1,rcv} \quad (2)$$

When the probing packets *transmission rate* exceeds available bandwidth, the relationship (2) becomes not correct and the probing packets have to stand in the waiting queue, leading to an increase in time intervals between the consecutive probing packets when received. Based on the above, the receiver determines the available bandwidth at the packet, among consecutive probing packets, where the time interval starts to increase in comparison with what it had from previous values. The above can be summarized by the following equations:

$$\begin{gathered} T_{i,snd} = T_{i,rcv}\ ;\ R_i \le AB_w \\ T_{i,snd} < T_{i,rcv}\ ;\ R_i \le AB_w \end{gathered} \quad (3)$$

We can calculate the instantaneous transmission rate at packet i as follows:

$$R_i = \frac{P_i}{T_i}$$

While the instantaneous probing rate of the next packet is:

$$R_{i+1} = \frac{P_{i+1}}{T_{i-1}} = \frac{P_i + \Delta P}{T_i - \Delta T}$$

At package i the instantaneous probing rate, relative to primitive values, is as follows:

$$R_i = \frac{P_1+(i+1)\Delta P}{T_1-(i-1)\Delta T} \quad (4)$$

We note that the instantaneous probing rate is a function of packet order in the probing train. Where the values of each of the following parameters: $P1$, $T1$, ΔP and ΔT are known and specified in advance. Fig. 5 shows the change in the probing rate.

From the above it is clear that the instantaneous probing rate changes with time within what we call the possible or *available probing range*.

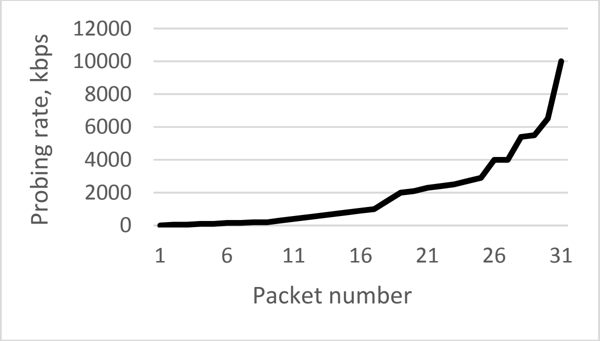

Fig. 5. Relationship between instantaneous probing rate and probing packets order, $N = 30$ packets, $\Delta T = 0.5$ ms, $\Delta P = 50$ bytes.

Available probing range is defined by two limit values: the lower probing rate $R_{\min}$, and the higher probing rate $R_{\max}$, which can be calculated by the following equations:

$$R_i = \frac{P_1}{T_1};\ R_{max} = \frac{P_N}{T_N} = \frac{P_1+(N-1)\Delta P}{T_1-(N-1)\Delta T} \quad (5)$$

where N is the number of packets forming the probing train. The length of the probing train is calculated by combining the time intervals between all the packets as follows:

$$T_{total} = \sum_{i=1}^{N} T_i = T_1 + (T_1 - \Delta T) + (T_1 - 2\Delta T) + (T_1 - (N-1)\Delta T) \quad (6)$$

Combining accuracy and speed in the estimation process is an essential requirement for approving the tool. For this, we tried to look for a method that can combine between both requirements accuracy and speed, with the fulfillment of the following constraints:

a. *Excluding synchronization*: between sender and receiver, that is, all the information needed by the receiver can be found in received probing packets, without the need to reconnect with the sender.

b. *Eliminating repetition*: since it has a bad effect on the estimation-process continuation time, and the tool should have an acceptable transparency for the network.

c. *Accuracy of estimation*: accuracy is relative, and we think that accuracy of the tool should be high at low available bandwidth values rather than at high values (problems of packets delay and loss increase when available bandwidth is small).

A measurement tool called *FindPath* is used which gives importance for low values of available bandwidth, which distinguishes it in comparison with other tools. The design of this tool relies on the combination of the two principles mentioned above, in which both of probing packets size and time intervals between these packets are changed. The tool uses a single probing train (i.e., repetition canceling), whose length is defined based on the desired precision. The FindPath tool using UDP protocol sends a single probing train consisting of N packets differ in their size from each other linearly, and time intervals between the probing train packets is not constant but decreases linearly as well. All what we have to do is determining the appropriate values of $R_{\min}$ and $R_{\max}$ for the required estimation process. Initially, identifying the values of $R_{\min}$ and $R_{\max}$ formed a challenge and was time consuming.

FindPath can estimate the available bandwidth within [0, 8] Mbps, with the focus on low values of the available bandwidth. As we mentioned the size of packets in a probing train built by FindPath, increases in a linear way. The largest packet size is 1500 Bytes (taking into account UDP protocol overhead added to the original packet size), while the smallest packet size being sent is determined based on the number of sent packets N, and the amount of size change ΔP. Number of packets used in the probing train, determines the number of levels of available bandwidth which can be estimated, and so this determines the level of required precision. If the number of packets is $N = 30$ packets (additional packet is sent after a specified time interval when sending probe packets stops. In order to obtain the last value which can be estimated), and packet size difference between two consequent packets is $\Delta P = 50$ bytes, the values of packets sizes which form the probing train are the following: 50, 100, 150, …, 1500 bytes, i. e. P1 = 50 bytes.

In order to determine the time interval between successive packets, as we said earlier, the time interval between the probing packets decreases linearly with the successive sending of packets. The time interval of the first packet is $T_1 = 16$ ms and the time interval of the last packet is $T_{N-1} = 1.5$ ms. The number of packets *is* $N = 30$. The amount of decrease in time is $\Delta T = 0.5$ ms. From all what preceded and basing on equations (1, 2, 3, 4), we get the following parameters: $P_1 = 50$ bytes, $T_1 = 16$ ms, $\Delta P = 50$ bytes, $\Delta T = 0.5$ ms. Substituting in equation (5) we calculate the minimum and maximum transmission rates:

$$R_{min} = \frac{P_1}{T_1} = \frac{50.8}{16\,ms} = 25\ Kbps$$

$$R_{max} = \frac{P_N}{T_{N-1}} = \frac{1500.8}{1.5ms} = 8\ Mbps$$

Substituting in equation (6) we get $T_{\text{total}} = 262.5$ ms.

To carry out the task of available bandwidth estimation, we designed an experimental network within a controllable environment using six COEX Clover drones as shown in Fig. 6, where four UAVs were used in order to generate, send and receive each of the cross-traffic and probing packets, as well as two routing nodes. One of the conducted experiments adopted the following scenario: the transmitters were hovering at a height of 23 meters, and the receiving nodes at 8 meters, while the relay nodes at a height of 13 meters and 18 meters, respectively.

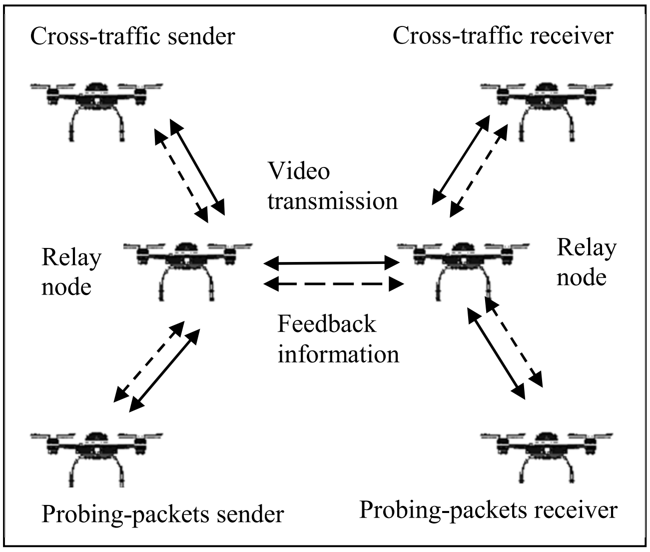

Fig. 6. Test network composed of six COEX Clover drones, cross-traffic sender and receiver, probing-packets sender and receiver, and two routing nodes.

In the process of estimation, we measure the available bandwidth for the link with a capacity of 10 Mbps, located between the two relay UAVs. The *cross-traffic* generator generates cross-traffic packets at a steady rate. Estimation process is repeated several times. The objective of repetition is to get accurate values by calculating the average of the values that have been achieved.

Fig. 7 shows the relationship between the available bandwidth and cross-traffic for FindPath. Fig. 8 shows the relationship between the relative error and the available bandwidth for FindPath.

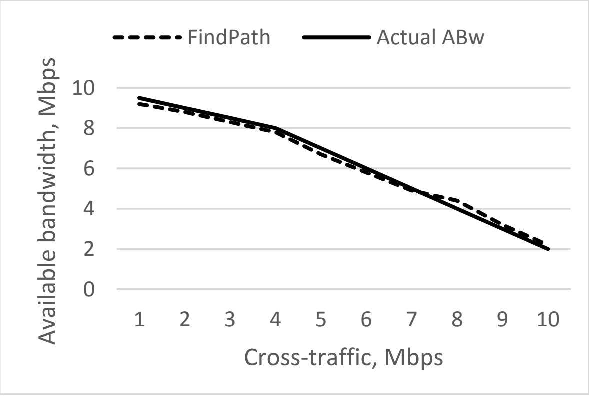

Fig. 7. Available bandwidth estimation results

Values of cross-traffic that have been adopted are: 2 Mbps, 3 Mbps, 4 Mbps, 5 Mbps, 6 Mbps, 7 Mbps, 8 Mbps, 8.5 Mbps, 9 Mbps, 9.5 Mbps. It was noticed that FindPath gives acceptable accuracy at low values of available bandwidth. We can increase the accuracy or change the range of measurement by increasing the number of sent packets and by changing the time interval between those packets.

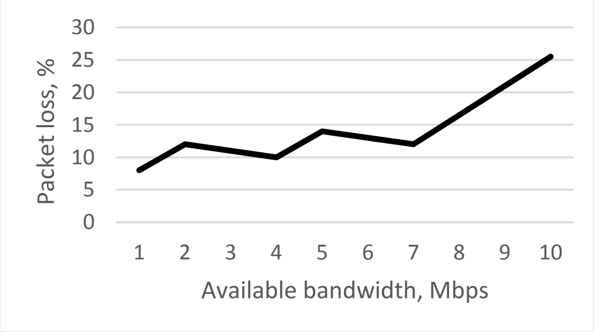

Fig. 8. Relationship between available bandwidth and relative error for *FindPath*.

Video streaming in multi-drone networks requires high bandwidth and frequently suffers from packet loss due to the high mobility of the nodes and to congestion. To overcome congestion and distribute the load in a balanced way, the FindPath tool was used for estimating the available bandwidth.

V. OVERHEAD REDUCTION

Sensor nodes send their data to the base station using paths that have minimum number of relay nodes, i.e., the number of hops traversed by the packets are kept to minimum. Available bandwidth values on the links are used as weights on a graph and the best path is chosen with respect to these dynamically changing weights. Link is not used if alternate lower loaded paths are available. The algorithm starts where there is only one stage to reach destination, finds optimal solution, then gradually finds current optimal solution from preceding one, until the whole task is solved without checking the calculated optimal paths in every iteration. This results in reduction in calculations. S is a source node, and D is a destination node (refer to Fig. 1). Decision variables x_n are immediate destination on stage n, if n= 4, the selected route is $S \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 = D$, where $f_n\ (s_t, x_n)$ is total cost of best overall policy for remaining stages, state s_t, stage n, and selects x_n as immediate destination. x_n* value of x_n that minimizes $f_n\ (s_t, x_n)$, $f_n{}^*(s_t)$ minimum value of $f_n\ (s_t, x_n)$,

$$f_n{}^*(x_n) = \min f_n\,(st, x_n) = f_n\,(st, x_n{}^*)$$
$$f_n\,(st, x_n) \ = c_{ij} + f_{n+1}{}^*(x_n)$$

where c_{ij} is the immediate cost (stage n) and $f_{n+1}{}^*(x_n)$ is the minimum future cost (stages $n + 1$), i is the current state s_t and j is the immediate destination x_n. Destination (state D) is reached at the end of stage 4, $f_5{}^*(\mathrm{D}) = 0$.

The objective is to find $f_1{}^*(\mathrm{S})$ and corresponding route by successively finding $f_4{}^*(s_t)$, $f_3{}^*(s_t)$, $f_2{}^*(s_t)$, for each possible states s_t and then using $f_2{}^*(s_t)$ to solve for $f_1{}^*(\mathrm{S})$.

To determine the routes for nodes in FANETs efficiently, scalable and distributed routing scheme can be applied. Possible paths between ingress/egress node pairs and the amount of bandwidth available on each of them should be known, regardless of the underlying topology. Each path is annotated by the allocated bandwidth. The number of hops can be used to guide the selection of one path if multiple suitable paths exist. *Bandwidth metric* is used along with the *number of hops*. The amount of available bandwidth along each path between the nodes changes dynamically. To handle such dynamic situations, each node maintains the amount of *available bandwidth* on all the paths to other nodes, and a list of *logical paths* to other nodes. A logical path between the pair of nodes consists of a sequence of nodes. To limit the amount of information to be maintained, some lengthy paths should be eliminated. The *end-to-end delay* may become too long and the service becomes unsatisfactory even though the available bandwidth meets the basic request. A pre-define path length L_r should be defined. If the path length between two nodes L_i is higher than L_r then this logical path is eliminated.

The source node forwards *probes* along all the existing paths to the destination node. If multiple suitable paths are available, then one can be selected either randomly, or based on other criteria such as the number of physical hops along the path, or the actual bottleneck capacity of the path. To choose the possible paths, a *delay threshold* d_{th} is used. For a possible logical path, the number of logical hops H should not exceed the specified delay threshold. In this manner, the overhead incurred in forwarding the probe on paths that may not satisfy the requirements is reduced. If no path between the chosen ingress/egress node pair has sufficient bandwidth to satisfy the requirement, the probe is *pruned* and not forwarded further. Otherwise, it is forwarded to the next node along the selected path. This continues until the probe reaches the destination node. Logical paths from source node to destination node may share a common portion at the beginning. *The probe is forwarded only once along the shared path***,** a technique called *probe aggregation*, which aids in the reduction of the overhead associated with forwarding the probes. The constraints of the flow are expressed in terms of the required bandwidth B_W. The bandwidth of the path is compared with B_W, and if this bandwidth is higher than B_W, the probe is forwarded to next node. This process of forwarding the probe is repeated. If the bandwidth of the entire path *is* greater than B_W the probe is forwarded to next node. If the paths share a common egress node, a single probe is forwarded to this node. The probe is pruned when the bandwidth of the paths between *certain nodes* is not sufficient. The number of probe messages has an upper bound of U_p.

The control message overhead of the routing scheme increases slowly with the delay threshold d_{th} than that of the traditional routing scheme. The probe overhead can be reduced by employing *probe aggregation* and *probe pruning* techniques. We use the delay threshold d_{th} to compute the logical paths: a logical path from a source proxy to a destination proxy should be no more than d_{th} logical hops longer than the shortest logical path between them. Fig. 9 illustrates the amount of overhead relative to the delay threshold measured by logical hops. The *overhead*, which is the total number of logical hops that probes traverse, is used as a performance metric to evaluate the scalability. The lower the overhead, the less the consumed bandwidth, and hence the more efficient the scheme. Routing effectively reduces the control overhead by sending only one probe between every involved ingress/egress pair. As shown in Fig. 9 the overhead of ABODV routing grows less than traditional routing, e.g., when $d_{th} = 4$ the overhead decreases to around 98% in the case of pruning, and to 78% in the case of probe aggregation. The scalability of routing via probe flooding can be improved by precomputing only the optimal paths.

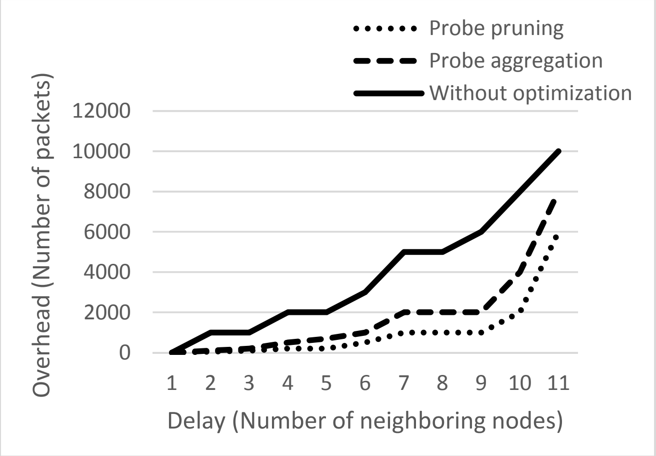

Fig. 9. Probe aggregation and probe pruning

VI. CONCLUSION

The characteristics of ad hoc networks and their routing protocols were investigated. Possible metrics to measure the performance of FANETs routing protocols were studied. ABODV protocol was introduced and its core architecture was described. The basic actions related to the route discovery process were studied. A special simulator showed that ABODV protocol will perform better in the networks with dynamic traffic, and it's more scalable than AODV. This enhancement can be added to any reactive routing protocol. Moreover, we found that FindPath gives good accuracy at low values of available bandwidth, better than at high values of available bandwidth, in comparison with other existing tools for available bandwidth estimation. Therefore, if the nodes in a FANET have low available bandwidth, we can use FindPath to optimally determinate the value of available bandwidth that we have to use for sending our data, which offers us the ability to overcome congestion and to achieve load balance over the network paths.

ACKNOWLEDGMENT

The reported study was funded by RFBR, project number 19-29-06076.

REFERENCES

[1] E. Goldoni and M. Schivi, “End-to-End Available Bandwidth Estimation, Tools, an Experimental Comparison”, University of Pavia, 2010.

[2] D.S. Vasiliev, D.S. Meitis, A.V. Abilov, “Simulation-based comparison of AODV, OLSR and HWMP protocols for flying Ad Hoc networks”, Lecture Notes in Computer Science, vol. 8638, 2014, pp. 245-252.

[3] A. Koucheryavy, A. Vladyko, and R. Kirichek, “State of the art and research challenges for public flying ubiquitous sensor networks”, Lecture Notes in Computer Science, vol. 9247, 2015, pp. 299-308.

[4] R. Kirichek, A. Paramonov, A.B. Koucheryavy, “Flying ubiquitous sensor networks as a queueing system”, in Proc. *17th International conference on advanced communications technology* (ICACT), 2015, pp. 127-132.

[5] N. Dao, A. Koucheryavy, and A. Paramonov, “Analysis of routes in the network based on a swarm of UAVs”, Lecture Notes in Electrical Engineering, vol. 376, 2016, pp. 1261-1271.

[6] I. Nurilloev, A. Paramonov, and A. Koucheryavy, “Connectivity estimation in wireless sensor networks”, Lecture Notes in Computer Science, vol. 9870, 2016, pp. 269-277.

[7] D.S. Vasiliev, K. Vladykina, A.V. Abilov, “Simulation study of application layer relaying algorithms with data-link ARQ in flying ad hoc networks”, in proc. *19th conference of open innovations association FRUCT*, 2016, pp. 256-263.

[8] J. Martinez, I. Sanudo, and M. Bertogna, “Analytical characterization of end-to-end communication delays with logical execution time”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, №. 11, 2018, pp. 2244-2254.

[9] D.S. Vasiliev, A.V. Abilov, I.A. Kaisina, D.S. Meitis, “Simulation-based assessment of quality of service in UAV-assisted MMWAVE system in crowded area”, *Conference of Open Innovations Association FRUCT*, vol. 23, 2018, pp. 391-397.

[10] T.D. Dinh, V.D. Pham, R. Kirichek, and A. Koucheryavy, “Flying network for emergencies”, *Communications in Computer and Information Science*, vol. 919, 2018, pp. 58-70.

[11] R. Pirmagomedov, R. Kirichek, M. Blinnikov, and A. Koucheryavy, “UAV-based gateways for wireless nanosensor networks deployed over large areas”, *Computer Communications*, vol. 146, 2019, pp. 55-62.

[12] I. Bekmezci, O. K. Sahingoz, and S. Temel, "Flying ad-hoc networks (FANETs): A survey", Ad Hoc Networks, vol. 11, 2013, pp. 1254-1270.

[13] O.K. Sahingoz, "Networking models in flying ad-hoc networks (FANETs): Concepts and challenges", *Journal of Intelligent & Robotic Systems*, vol. 74, 2014, pp. 513-527.

[14] P. K. Singh, G. Sharma, “An efficient prevention of black hole problem in AODV routing protocol in MANET”, *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, pp. 902-906.

[15] J. Souza, J. Jailton, T. Carvalho, J. Araújo, and R. Francês, “A Proposal for Routing Protocol for FANET: A Fuzzy System Approach with QoE/QoS Guarantee”, *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[16] Y. Peng and Z. Yan, “Available bandwidth estimating method in IEEE802.11e based mobile ad hoc network”, In proc. *9th International Conference on Fuzzy Systems and Knowledge Discovery* (FSKD), 2012, pp. 2138–2142.

[17] S.S. Chaudhari and R.C. Biradar, “Survey of bandwidth estimation techniques in communication networks”, *Wireless Personal Communications*, vol. 83, 2015, pp. 1–52.

[18] A. Farshad, M. Lee, M.K. Marina, F. Garcia, “On the impact of 802.11n frame aggregation on end-to-end available bandwidth estimation”, in proc. *11th Annual IEEE International Conference on Sensing, Communication, and Networking* (SECON), 2014, pp. 108–116.

[19] H. Liu and L. Cheng, “Available bandwidth estimation strategy based on the network allocation vector”, *Journal of Networking*, vol. 7, 2012, pp. 2089–2095.

[20] E.P. da Silva Mineiro and D.C. Muchaluat-Saade, “CAC-OLSR: extending OLSR to provide admission control in wireless mesh networks”, *International Journal of Wireless Information Networks*, vol. 21, 2014, pp. 223–237.

[21] S.S. Chaudhari, and R. C. Biradar. “Collision probability based available bandwidth estimation in mobile ad hoc networks”. In proc. *5th International Conference on the Applications of Digital Information and Web Technologies* (ICADIWT), 2014, pp. 244–249.

[22] T. Yang, Y. Jin, Y. Chen, and Y. Jin. “RT-WABest: A novel end-to-end bandwidth estimation tool in IEEE 802.11 wireless network*”, International Journal of Distributed Sensor Networks*, vol. 13, 2017.

[23] T. Hayajna, M. Kadoch, and B. Rong, “Ensuring reliable and stable

[24] communications in mobile ad-hoc networks", *Vehicular Technology Conference (VTC-Fall), IEEE 86th*, 2017.

[25] R. Yiftach and I. Bergel. "Optimal and suboptimal routing based on partial CSI in wireless ad-hoc networks", *IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC),* 2015, pp. 191-195

[26] L.X. Cai, H.Y. Hwang, X. Shen, J.W. Mark, and L. Cai, "Optimizing geographic routing for millimeter directional antenna", *6th International Conference on Broadband Communications, Networks, and Systems*, 2009.

[27] A. Thornburg, B. Tianyang, and Robert W. Heath Jr. "Performance Analysis of mmWave Ad Hoc Networks", *IEEE Trans. Signal Processing*, vol. 64, 2016, pp. 4065-4079.

[28] K. Day, A.Touzene, B. Arafeh, and N. Alzeidi, "Parallel routing in mobile ad-hoc Networks", *International Journal of Computer Networks and Communications*, vol. 3, 2011, pp. 77–94.

[29] M.S. Islam, A. Riaz, and M. Tarique, "Performance analysis of the Routing protocols for video Streaming over mobile ad hoc Networks", *International Journal of Computer Networks &Communications*, vol. 4, 2012, pp. 133-150.