# PROTOCOL ANALYSIS FOR SHORT-RANGE TRANSCEIVERS

**Antouan Anguelov, Roumen Trifonov, Ognian Nakov**

***Abstract**: Local communication protocols play an important role in Multi-Robot Systems (MRS). Establishing "stack" of communication protocols is the key to potential problems in communication connections. In addition, an application layer protocol should be planned to support the efficient transmission of control messages as event driven communication between the multiple robots. This paper aims to analyze main networking protocols useful for a hybrid short-range wireless transceiver based on the light and sound mediums presented in the paper "Emerging and secured mobile ad-hoc wireless network (MANET) for swarm applications" [1]. Specific issues on the networking protocol level will be explored. We will review and evaluate the effects of different key futures such as topology, quality of service (QoS), reliability and architecture patterns.*

***Key words:** point-to-point, MESH, SWARM, Distributed Robotic Systems, Autonomous Robots, Fog-based Multi-Robot Systems (MRS), mobile robot systems, swarm application layer protocols (SALP), event-driven communication, fog computing*

## 1. INTRODUCTION

Communication between mobile robots is necessary and, in many cases, critical in SWARM applications. Swarm is a system of peer-to-peer networked agents – in our case mobile robots that create a decentralized and distributed group of less powerful computes, storages synchronized over communication service. The communication infrastructure between the robots can be damaged or disrupted at some point, which requires the mobile robots to form an ad hoc network, using each other as redirecting nodes to create a new communication channel. Thus, routing messages between mobile robots or from robots to a central server requires specific routing protocols that can operate without central control (server) following dynamic changes in topology due to the mobility of the environment in which a large number of robots operate. The messages between MRS also need to apply to certain criteria such as: to be short, to have ability to be recovered or resent again in case of packet loses, etc. Messages in the various applications of mobile robots [1] can be categorized into the two main categories: (1) Unicast messages are sent from one mobile robot to another when the data is related to a specific purpose - such as images, requests for help; (2) Multicast messages sent from a central server or a mobile robot to a group of robots, usually for coordination and additional control information. So, we are dividing the mobile robot communication to two types: Coordination (short massages) or data oriented (long massages). In this paper will be focusing on short messages routing protocols need as important and emergency telecommunication between the mobile robots.

## 2. HISTORY

Communication is an important consideration as especially for the teams of robots operating in dynamic and potentially changing environments. These types of environments require agents to be capable of coordinated sensing, processing, and communicating with each other [2]. One of the main issues we need to consider during network design is how the MRS agents act and manage their interaction. Mobile robots often move dynamically, they can join and leave the network after the deployment due to mobility or battery depletion and cloud-based networking would not be an appropriate solution in a swarm. Here is fog-based multi-robot networking come. Fog computing is moving the control from the cloud to MRS.

So, there are several known topologies we need to mention regarding the fog computing:

1. Centralized. Where a single point of control manages the behavior of all the robots in the team [3]. Such architecture suffers from a single point of failure problem, and we really would like to avoid it because it reduces its reliability. Also, scalability is diminished.

2. Decentralized/distributed. In this case, robots take actions based on their own local view following certain strategic guidelines and goals for the team given as task. This model offers robustness and ability to adjust to failures, since no centralized control is used. It allows dynamic network configurations.

3. Hierarchical/hybrid with multi-hop. Specifically, in this strategy, a robot controls a group of other robots. Each of those robots in turn controls a group of other robots. It combines a decentralized control and provides robustness with hierarchical control to achieve global synchronization and coordination.

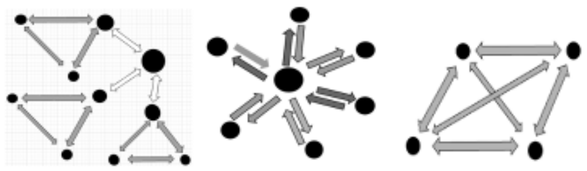

***Figure 1.** Hybrid, Centralized, and distributed networking topology*

If we give an example with birds in the real world, we defiantly should represent each swarm robot as an autonomously moving agent with the ability to freely communicate and interact directly with others. Unfortunately some agents will not be reaching others because of their position and environment conditions and multi-hop function would be needed. So, the distributed hybrid topology of the communicational system, which is characterized by no center and no fixed infrastructure is shown in Figure 1, which will satisfy communication topology in swarm R2R. Moreover such as a peer-to-peer system that would take advantage of resources distributed across the robots would be possible to provide secure and emergency applications too. Each agent in such an ad-hoc R2R network (cluster) should support routing function together with a token-based protocol (algorithm) for mutual exclusion. Several algorithms achieve mutual exclusion with free from deadlocks and starvations. Examples of permission-based distributed mutual exclusion algorithms are Lamport's algorithm (3(N-1) messages), Ricart-Agrawala (RA) algorithm (2(N-1) messages) and Maekawa's algorithm. Susuki-Kasami's algorithm (N messages) and Raymond's tree-based algorithm (log(N) messages) are examples of token-based mutual exclusion algorithms [4].

## 3. PROTOCOL ANALYSIS FOR THE SHORT-RANGE TRANSCEIVERS

The basic OSI model (Figure 2) consists of seven different logical layers, providing everything they need for two or more hosts to communicate.

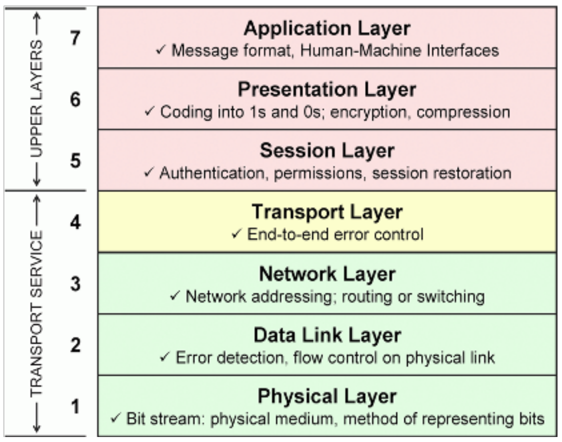
UPPER LAYERS
TRANSPORT SERVICE
7
Application Layer
✓ Message format, Human-Machine Interfaces
6
Presentation Layer
✓ Coding into 1s and 0s; encryption, compression
5
Session Layer
✓ Authentication, permissions, session restoration
4
Transport Layer
✓ End-to-end error control
3
Network Layer
✓ Network addressing; routing or switching
2
Data Link Layer
✓ Error detection, flow control on physical link
1
Physical Layer
✓ Bit stream: physical medium, method of representing bits

***Figure 2.** Basic reference of the Open System Interconnection (OSI) model.*

### 3.1 PHYSICAL LAYER

The physical layer defines characteristics such as timing and voltage. This layer is responsible for physically sensing the carrier and detecting the collision and performs the auto-negotiation process. Implementation of encoding like Barker binary codes with Linear Feedback Shift Registers [LFSR], different modulation schemas could be programmed by FPGA / DSP [1]. In addition, modulation used in [1] as baker codes spatial modulation can be implemented.

### 3.2 PHYSICAL LAYER

The data link layer can be subdivided into two other layers: Media Access Control (MAC) layer, and the Logical Link Control (LLC) layer. The MAC layer establishes the computer’s identity on the network, via its MAC address. MAC address is responsible for using that information to decide when to transmit a packet and retransmitting the packet if a collision happens. That is the way broadcast method related to the carrier sense multiple access collision detect (CSMA) technology [7] could help. In the contention communication model (peer-to-peer) a contention method such as carrier sense with multiple access/collision detection or collision preventing (CSMA /CD or CA) can be used for communications control. There is no controlling master, and individual agents have to contend (complete) for access to the transmission medium.

In an arrangement such as our ad-hoc hybrid distributed network by using optical and sound propagation collisions are unavoidable and robot communication controller has to deal with them. This variety of light and sound signals can severely affect the transmission reliability, despite pointing the devices to each other may seem enough for reliable communication. That is the way the “Fast listen” algorithm while talk is used to check if the channel is busy during, we are transmitting zeros. In fact, CSMA/CD-HA protocol that operates in the Data Link Layer (OSI Layer 2) could solve the problem [8] by collisions handling together with Listen-Before-Transmit (LBT) technique to avoid communication conflicts [9].

The LLC layer controls frame synchronization and provides a degree of error checking. Here we have protocol such as PPP (Point-to-Point Protocol) which uses the HDLC Frame Check Sequence for error detection. PPP is based on a variant of HDLC. PPP on serial links like ours is usually encapsulated in a framing similar to HDLC, described by IETF RFC 1662

## 3.3 NETWORK, TRANSPORT AND SESSION LAYER

The Network Layer is responsible for determining how the data will reach the recipient. This layer handles things like addressing, routing, and logical protocols. The network layer is also responsible for its error handling, and packet sequencing and congestion control.

The major function of transport layer protocols includes flow control, reliability, multiplexing, connection-oriented communications, and congestion avoidance. This layer is also responsible for providing error checking and performing data recovery when necessary. The complex connection mechanism is mandatory, even if the entire network only consists of two devices trying to communicate with each other.

The session layer is the first layer where efficient transmissions over the network are not considered. This layer is a part of the middleware.

Here we would provoke all readers to pay attention to Ref. [10] implementing an old but very flexible solution - Scalable Node Address Protocol (SNAP). SNAP covers all 2-5 layers. It can replace known data link layer like protocols like CSMA, PPP or network layers such as IP, and UDP/TCP transport layer protocol. SNAP can be used as a very simple protocol without any flags or error detection, or the programmer can use up to 24 flags and any of the defined error detection methods, depending on the current need (or personal skill). Since S.N.A.P is scalable, both simple (read as cheap) and sophisticated nodes can communicate with each other in the network. The SNAP protocol can theoretically support up to 16 million nodes in a single network. Since these are mesh networks, there is no single point of failure: any node can talk directly to any other node that is in range, and any node can talk indirectly to any other node via intermediate nodes. Furthermore, SNAP based networks are self-healing – if a node fails for any reason, other nodes will automatically route signals around the failed node.

## 3.4 APPLICATION LAYER

Most of the applications in robots involve real-time communication. For example, a remote-controlled robot requires a real-time response and distributed fault-tolerant network so it would operate properly in critical applications. A delay in communication may result in an undesired outcome and can be dangerous if for example it involves heavy or fast-moving robots. An example of a controlled robot that depends on real-time response is “da Vinci”, a remote-controlled surgery robot [11]. To achieve fast response time, a lightweight messaging communication protocol must be involved within the distributed redundancy application network. That is why we propose that collaborative robots should communicate with each other over a cluster-based distributed fault-tolerant network where everyone is publishing or subscribing to a specific topic name. The topic is used to identify the content of the message. Hence, a robot that requires a certain kind of data, subscribes to the appropriate topic. In swarm scenarios there are multiple concurrent publishers and subscribers for a single topic, and a single robot may publish and/or subscribe to multiple topics. Each robot could play a server role (broker) to establish a high availability cluster via the so-called Load Balancer (LB). It distributes MQTT connections and traffic from devices across the clusters and enhances the High Availability (HA) of the cluster, balances the loads among the cluster nodes and makes the dynamic expansion possible.

Such topology follows a modern hybrid publish-subscribe and point-to-point design pattern at the application layer with a few messages protocols for the specific of our paperwork:

(1) “MAVLink” - lightweight messaging protocol for communicating [12] between drones and between onboard drone components via the serial interface. This paper will discuss XRCE-DDS [13] and MQ Telemetry Transport, which claim to be lightweight message protocols used in IoT and open-source robotic operating system (ROS)[5].

(2) Real-Time Publish-Subscribe (RTPS) [15]. RTPS is the underlying protocol of the Object Management Group's (OMG) Data Distribution Service (DDS) [14] standard. It aims to enable scalable, real-time, dependable, high-performance and interoperable data communication using the publish/subscribe pattern. Fast RTPS is a very lightweight cross-platform implementation of the latest version of the RTPS protocol and a minimum DDS API.

(3) MQ Telemetry Transport for Sensor Nodes (MQTT-SN) [22] which are designed for IoT devices. MQTT is a lightweight messaging protocol for machine-to-machine communication, typically used for IoT messaging but we see potential in implementing it for the needs of the collaborative SWARM applications.

(4) The XRCE protocol is a client-server protocol between resource-constrained devices (clients) and an XRCE Agent (server). The protocol allows resource collaborative robots with sleep/wake cycles to have access to the DDS Global Data Space over limited-bandwidth networks.

To keep the message size as small as possible, stateful endpoints are used which violates the idempotence principle. One example is the numeric topic id, which replaces the long topic string an MQTT client should send with each message. As a prerequisite, the client should negotiate the topic id with the broker. The protocol does not define how the broker or client should react on a restart of its counterpart when all subscriptions and topic registrations are gone. The TCP-less communication protocol is MQTT-SN. As its postfix “SN” implies, the

MQTT spin-off is designed for constrained sensor networks. It adds some improvements to the protocol, like an error status. MQTT-SN allows us to build up a network of constrained devices with a central broker connected to many clients. Message distribution is controlled by a message bus like pub-sub mechanism and topic registration. MQTT also includes support for detecting failed connections and introduces keep-alive signaling between clients and servers. MQTT [16] provides three, so-called, QoS-levels for delivery. These levels guarantee that a message is delivered (i) at most once (QoS 0), (ii) at least once (QoS 1), (iii) or exactly once (QoS 2) to the receiver. If at most once is specified, no retransmissions are used. This is the default QoS level and is used if occasional message loss can be tolerated. If at least once is specified, a message identifier is specified in the message header and the receiver can acknowledge the packet using this message identifier. A sender that does not receive an expected acknowledgment, it sets a duplicate bit in the message header and retransmits the message. This QoS level is used when guaranteed message delivery is required and duplicates can be tolerated. If exactly once is specified, at least two message exchanges are necessary, After the sender has received the acknowledgment from the receiver, it tells the receiver that it can complete processing the message and waits for an acknowledgment for this signaling. Consequently, this is the most reliable QoS level, but introduces some overhead and is the slowest variant.

Pluggable transport layer. Micro XRCE-DDS is not built over a specific transport as serial or UDP. It is agnostic about the transport used and give the user the possibility of implementing easily his tailored transport. By default, UPD, TCP, and serial transports are provided. By abstracting out low-level networking and communication details and providing a flexible integration framework, DDS Micro minimizes the amount of device or application-specific code that needs to be created.

Usually, SWARM is determined by hybrid point-to-point and point-to-multi point connections. Each agent plays specific role: central – as mission operator or a client (drones, mobile platforms). Communication between the agents can be internal as onboard communication between the internal running applications/services or offboard where different applications/services from one node exchange data with applications/services to another node/s. Specific futures listed in Table 1.

**Table 1**. Short-range protocol key futures

| **Key futures** | **MAVLink** | **Ros serial** | **DDS XRCE** |
|---|---|---|---|
| **Infra-structure** | PtP | PtP/PtM | PtP/PtM |
| **Sensors/ comm.** | Onboard/ offboard | On-board/ offboard | Onboard/ offboard |
| **Network Role** | Central/ GCS | client | Agent/ client |
| **Platform** | multi | multi | multi |
| **Type** | UDP/ TCP/ serial | serial/ TCP | UDP/TCP/ serial or SHM |
| **Long Range/ Wireless** | OK | N/C | N/C |
| **QoS** | OK | N/C | OK |
| **Domain** | Drone/ GCS | ROS | multi |
| **Protocol Dialects** | OK | N/C | N/C |
| **Program-ming Languages** | multi | multi | multi |
| **Sequrity** | OK | N/C | OK |
| **Max. number systems** | 255 | N/C | N/C |

The so called SALP (swarm application protocols) Ref. in Table 1 contain hybrid usage of: (1) MAVLink being a very lightweight messaging protocol for communicating with drones (and between onboard drone components) could be use in wireless environments for communication between the GCS and all the nodes in PtP applications;
(2) DDS-XRCE, would be one of the upcoming protocol standards for ROS 2 communication with focused on microcontroller applications which require a publisher/subscriber architecture. Some examples of this kind of applications are found in a sensor network, IoT and robotics. This protocol is very fast by using shared memory communication (SHM) among entities running on the same host. DDS
Often XRCE is used for inter-process communication (IPC) - a true zero-copy, traffic-shm (Shared Memory) that allows to share data from publishers to subscribers without a single copy - for example the Eclipse project “AceOrix” [6]. It makes the XRCE protocol powerful where large amounts of data have to be transferred between onboard different processes when it comes to driver assistance or automated driving systems;

(3) Rosserial is wrapping standard ROS serialized messages and multiplexing multiple topics and services over a character device such as a serial port or network socket. It provides a ROS communication protocol that works over your embedded Linux system's serial UART, or its WI-FI or network connection. It allows embedded Linux system to run

Linux processes that are full-fledged ROS nodes that can directly publish and subscribe to ROS topics;
Expect that the network node/MRS would be handling many links of communication between different applications both onboard and offboard in real-time handling sensor, positioning or other type of traffic.

## 4. CONCLUSION

With this research in the paper would like to announce several ideas, to provoke further experiments winch can show practical implementations of the designed real-time secure SWARM communication infrastructure.
However, several limitations merit comment. R2R communication is in the developing stage. All big manufacturers are focused on cloud and cloud-based multi-robot systems. Fog computing and fog-based communication are just rising. This causes R2R communication to be still out of the standardization for example by IEEE.
Protocol domain in for short-range data transmissions are not under current research focus.
Today most networking protocols operate efficiently only when the number of nodes in the SWARM is small. Such protocols might not be practical for large networks. This can be the result of a high number of control message exchanges that are not localized to the event area for example. In some points this approach is the loss of effective bandwidth caused by the additional idle time between frames when nodes wait for their priority slot to come up.
We mentioned also several known protocols that are used at different OSI layers for VLC, IoT and TCP but they still need to be adopted by practical experience in production. On the other hand, the new fog communication models need to be implemented in the middleware level in a well-known open-source robot operational system (ROS) [5].
Contrariwise, we will keep this discussion open for experts in robot protocols such and we focus on messaging protocols used in robot operational system (ROS) as mini-XRCE-DDS or MQTT-SN. MQTT-SN [16] was the best candidate for messaging protocol in our research. However, the lack of platforms supporting the protocol has to be considered.
Nevertheless, despite these limitations above, we believe this study opens new paths of investigation because new insights into it will likely generate novel methods. There is, therefore, a great need for further research in this area.

## 5. ACKNOWLEDGMENTS

This paper was endorsed by the Faculty of Computer Systems and Technologies at the Technical University of Sofia. We would like to acknowledge our colleagues from the Faculty for the valuable insights, comments and contribution towards the proposed protocols.

## REFERENCES:

[1] Antouan Anguelov, Roumen Trifonov and Ognian Nakov. 2019. Emerging and secured mobile ad-hoc wireless network (MANET) for swarm applications. In Proceedings of 9th Balkan Conference on Informatics (BCI'19). September 26–28, 2019, Sofia, Bulgaria, 4 pages, https://doi.org/10.1145/3351556.3351557

[2] Alan Wagner, Ronald Arkin, "Multi-Robot Communication-Sensitive Reconnaissance"

[3] Imad Jawhar, Nader Mohamed, Jie Wu and Jameela Al-Jaroodi, "Networking of Multi-Robot Systems: Architectures and Requirements", 30 November 2018, https://www.mdpi.com/2224-2708/7/4/52/pdf-vor.

[4] Kayhan Erciyes, Orhan Dagdeviren, "A distributed mutual exclusion algorithm for mobile ad hoc networks", International Journal of Computer Networks & Communications (IJCNC) Vol.4, No.2, March 2012

[5] Robot operating system (ROS), https://micro.ros.org/docs/concepts/middleware/rosserial/

[6] Inter-process-communication (IPC) middleware for various operating systems, Eclipse.org, https://github.com/eclipse-iceoryx/iceoryx

[7] Pooya Karimian, "Audio communication for multi-robot systems", B.Sc., Sharif University of Technology, 2003, https://pooyak.com/work/pubs/pooya-karimian-msc-thesis.pdf.

[8] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, Yuval Elovici Ben-Gurion University of the Negev, Israel, "MOSQUITO: Covert Ultrasonic Transmissions between Two Air-Gapped Computers using Speaker-to-Speaker Communication", Cyber-Security Research Center, March, 2018, https://arxiv.org/pdf/1803.03422.pdf.

[9] Han Zhang, Wei Wang, Yang Zhou, Chen Wang, Ruifeng Fan, Guangming Xie, "CSMA/CA-based Electrocommunication System Design for Underwater Robot Groups", 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) September 24–28, 2017, Vancouver, BC, Canada

[10] Scalable Node Address Protocol, https://web.archive.org/web/20110718041306/

[11] Wikipedia, "da Vinci Surgical System", https://en.wikipedia.org/wiki/Da_Vinci_Surgical_System

[12] Mavlink, The MAVLink command protocol, https://mavlink.io/en/services/command.html

[13] MicroXRCE-DDS Documentation, Release 1.0.0, eProsima, https://media.readthedocs.org/pdf/micro-xrce-dds/latest/micro-xrce-dds.pdf

[14] "Data distribution service (DDS) specification, version 1.4,",https://www.omg.org/spec/DDS/, 2015.

[15] "The real-time publish-subscribe protocol (rtps) DDS interoperability wire protocol, version 2.2," https://www.omg.org/spec/ DDSI-RTPS/2.2/, 2014.

[16] Andy Stanford-Clark and Hong Linh Truong, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2" http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf, November 14, 2013.