Contents

Communication Protocol and Cluster Networking 2

Common application layer protocols for UAV (DDS, MAVlink) 3

Common routing protocols for UAVs (AODV, OLSR, DSR, DSDV) 4

Coarse-grained networking communication simulation 6

MQTT Networking Communication Simulation 6

Redis networking communication simulation 7

NS-3 route simulation 8

Communication Protocol and Cluster Networking

Using UAV to realize the function of relay communication, its communication platform has the characteristics of convenient deployment, strong mobility, good performance, flexible control, and easy upgrading of communication equipment. Compared with mobile communication, its deployment cycle is much shorter and the cost is much lower. Compared with satellite communication system, its communication delay is smaller and the cost is lower. Therefore, the relay communication of UAV has certain natural advantages.

According to the structure and tasks of the UAV, the communication equipment that the UAV can carry is limited; in the process of communication, the communication equipment on the UAV has been moving with it, and the distance between it and the ground communication equipment has been changing. The UAV needs to deal with the adverse effects of environmental changes and signal changes caused by its movement, so as to ensure the real-time and stability of communication; Because most UAVs are powered by batteries and the transmission distance is long, the power consumption of communication equipment should be low, the signal transmission distance should be long, the receiving sensitivity of data link should be high, and the receiving performance should be good; In some complex application environments, the unmanned aerial vehicle (UAV) needs to realize the transmission and communication of high-speed image data. Because of the low altitude of UAV flight, it is necessary to consider the impact of multipath, clutter and other harsh channels on the transmission performance, so as to minimize the bit error rate of information transmission; When UAVs are in multi-aircraft formation and multi-aircraft joint operations, UAVs need multi-point communication, compatible processing of various signals, and anti-electromagnetic interference measures to achieve effective transmission of instructions and multi-aircraft cooperative communication.

Key Technologies and Functions of UAV Communication in Various Mobile Network Scenarios

UAV function	**Key technologies**	**Number of UAVs (unit)**
Communication Base Station	Air, space, land and sea integrated communication	1
Collecting sensor information and sending the sensor information to a base station	Dynamic spectrum sharing	many
Collecting sensor information and sending the sensor information to a base station	Finite-length channel coding	many
Communicate with terrestrial users or base station	Very large array antenna	1
Users who send and receive messages	Terahertz	4
User receiving information for self-learning	Artificial intelligence	many
The user who shipped the drug	Blockchain, artificial intelligence	1

Common application layer protocols for UAV (DDS, MAVlink)

DDS (Data Distribution Service for Real) is a data-oriented middleware, which provides a standard way for real-time systems to publish, discover, and subscribe to distributed data.

The core concept of DDS is the data publish/subscribe model, where publishers publish data to topics and subscribers receive data from these topics. DDS uses a technique called the Service Discovery Protocol (SMP or SDP) to discover other DDS entities (such as publishers, subscribers, and so on) on the network. This means that when a DDS node is started, it can automatically find topics and services that are available on the network. DDS can communicate using different transport layer protocols, including TCP/IP, UDP, shared memory, and so on. This allows DDS to adapt to a variety of network environments and performance requirements. DDS provides a variety of quality of service (QoS) policies such as reliability, persistence, and order guarantees. These policies can be tailored to the specific needs of the application to optimize data transfer. DDS supports the definition of data types and ensures that these types are consistent across platforms. This allows DDS to handle complex heterogeneous systems. One of the design goals of DDS is to provide real-time communication with low latency and high throughput.

As an application layer protocol for UAVs, DDS has the following advantages: Real-time performance: DDS is designed for real-time systems and can meet the low latency requirements of UAVs in flight control and other critical tasks. Loose coupling: DDS allows publishers and subscribers to be loosely coupled, which means that components in the system can be developed and updated independently without affecting other parts. Flexibility: DDS supports dynamic network topologies, so nodes can be added or removed during flight. This is very important for UAVs that need to flexibly adjust their mission configuration. Scalability: Because DDS is a topic-based data publish/subscribe model, it can easily handle more data streams and participants as the system grows in complexity and size. Quality of Service (QoS) policies: DDS allows the user to select different QoS policies, such as reliability, persistence, and order guarantees, to optimize data transmission according to specific requirements. Standardization: DDS is a standard developed by the OMG organization, which helps ensure interoperability and compatibility between different vendors. ROS integration: DDS is well integrated with the Robot Operating System (ROS), especially in ROS 2, where DDS is widely adopted as the default communication framework. Defense and Industrial Applications: Given that DDS was originally developed in the military and has been successfully used in several industries, its reliability and safety are an important advantage for UAV applications.

The limitations on the use of DDS are as follows. DDS has many advanced features and configuration options that make it relatively complex. DDS may require more computational and memory resources to run than some simple message-passing systems. To support real-time performance, DDS introduces some additional overhead, such as cache management, network protocol processing, and so on. These overheads may affect the overall efficiency of the system.

The MAVLink protocol is a lightweight communication protocol specifically designed for small unmanned aerial systems (UAS). Its role in the UAV application layer is to provide a standardized way to exchange flight control information, sensor data, and other mission-related data.

MAVlink has the following advantages as a UAV application layer protocol: Compact message structure: The message body is usually very small, which is suitable for wireless links with low bandwidth and high latency. Cross-platform support: Can run on a variety of hardware platforms, including embedded devices and Windows systems. Ease of implementation: There are a number of open source libraries and tools available that simplify the development process.

MAVlink has the following main limitations. Lack of quality of service guarantees: it does not provide QoS mechanisms to ensure reliable delivery of critical data. Lack of real-time performance: Although it can be used in real-time systems, it is not specifically designed to meet strict real-time requirements. There are no built-in security features: external mechanisms are required to protect the security and integrity of data.

Comparatively speaking, the two protocols have similar functions but different scopes of application. The main difference is that DDS is a more complex and powerful real-time data distribution protocol. DDS is designed to solve the problem of data sharing in large-scale distributed systems, so it is more suitable for large-scale and complex distributed systems that need powerful data management and distribution capabilities. It also allows users to configure different QoS policies and customize data transmission according to their own needs. MAVlink is more suitable for simple UAV applications, especially in resource-constrained environments.

Common routing protocols for UAVs (AODV, OLSR, DSR, DSDV)

AODV: Ad-hoc On-Demand Distance Vector Routing (AODV) is a routing protocol for Mobile Ad Hoc Networks (MANET). In such a network, there is no fixed infrastructure or central control node, and each node must be able to forward packets as a router to enable communication throughout the network. Since the nodes in such a network environment can move around at will, the network topology may change frequently.

It works as follows: When a source node needs to send data to a destination node and no route is available in the local routing table, it broadcasts a Route Request (RREQ) message. The RREQ message is passed along the network, and each node that receives it notes the source of the request and forwards it to its neighbors. After receiving the RREQ, the destination node replies with a "route reply" (RREP) message that includes information about the shortest path to the source node. The route reply is returned along the original path, and each node updates its routing table after receiving the reply so that data can be sent directly over the path in the future.

The application of AODV protocol is very important in UAV Ad Hoc network. As UAVs are often highly mobile, the communication network between them must be able to quickly adapt to changing topologies. In addition, because UAVs may perform remote missions or be in remote areas, network bandwidth and energy efficiency are important considerations.

The strength of AODV is its on-demand nature, which means that it creates and maintains routes only when they are actually needed, which helps reduce unnecessary network overhead. However, for highly dynamic scenarios, such as multi-UAV cooperative flight, AODV may need to be optimized to improve routing stability, reduce latency, and improve overall performance.

OLSR: Optimized Link State Routing Protocol (OLSR) is a link-state routing protocol for mobile ad-hoc networks (MANET). Unlike distance vector protocols such as AODV, OLSR uses global network topology information to make routing decisions.

It works as follows: Each node periodically sends Hello messages to its neighbors to detect and update the neighbor list. Based on the neighbor list, each node selects a set of MPR nodes and announces this selection in the next Hello message. When a Hello message is received, the node updates its neighbor and MPR sets. The node generates a TC message according to the neighbor and MPR information, which contains the information of all reachable nodes. The TC message is forwarded through the MPR node to propagate the topology information throughout the network. Each node updates its routing table based on the received TC message to find the best path for the packet.

In UAV communication, OLSR may be more suitable for some scenarios than AODV because it provides better network visibility and faster convergence. However, the main disadvantage of OLSR is that it requires more bandwidth and computational resources to maintain and disseminate topology information.

DSR: Dynamic Source Routing (DSR) is a source routing protocol for Mobile Ad Hoc Networks (MANET). Unlike traditional routing protocols such as AODV and OLSR, DSR does not require each node to maintain the topology information of the entire network, but allows packets to carry complete path information.

DSR works as follows: When a source node needs to send data to a destination node, it checks its route cache to see if there is a valid route to the destination node. If a valid route is found, the packet is sent directly using that route. If no valid route is found, the source node broadcasts a Route Request (RREQ) message. The RREQ message travels along the network, and each node that receives it notes the source of the request and forwards it to its neighbors. After receiving the RREQ, the destination node replies with a Route Reply (RREP) message that includes information about the shortest path to the source node. The route reply is returned along the original path, and each node updates its route cache after receiving the reply so that data can be sent directly over the path in the future.

In UAV communication, DSR may be more suitable for some scenarios than other routing protocols because it provides lower control overhead and higher flexibility. However, the main disadvantage of DSR is that it is very sensitive to loops in the network and requires additional measures to avoid loops. In addition, since each packet needs to carry the complete path information, it may cause the packet header to be too large, thus affecting the transmission efficiency.

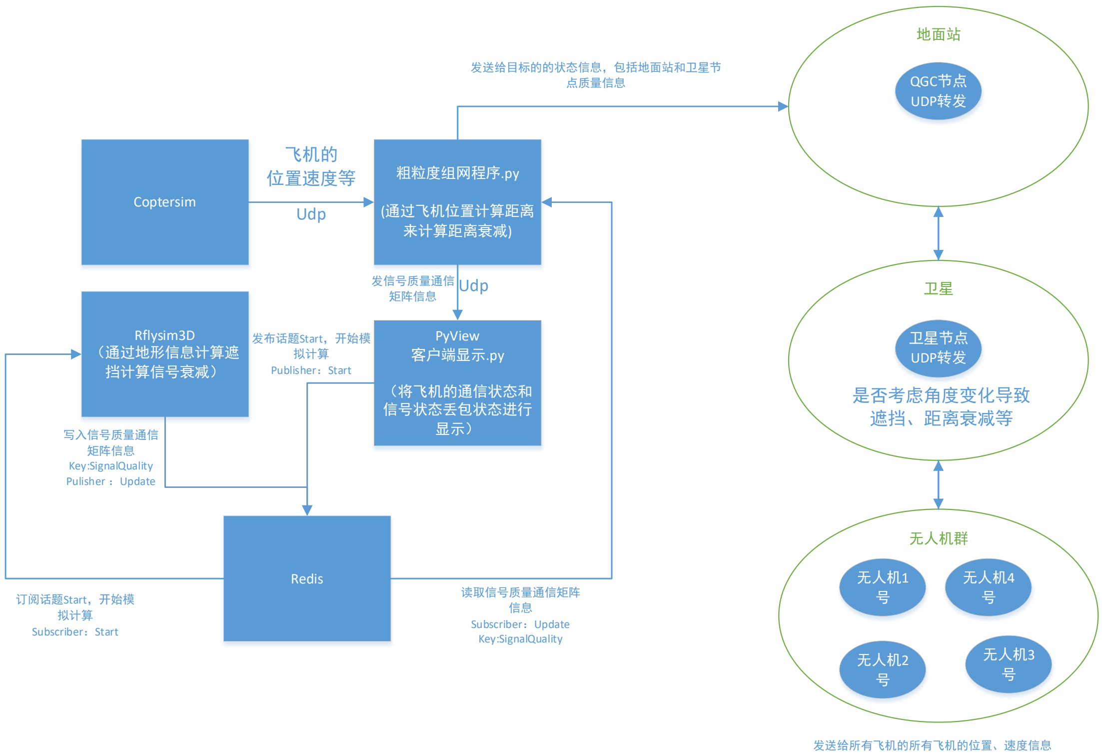
DSDV: Destination-Sequenced Distance Vector (DSDV) is a distance vector routing protocol for Mobile Ad Hoc Networks (MANET). Unlike on-demand routing protocols such as AODV and OLSR, DSDV uses a periodic update mechanism to maintain routing tables in the network.

DSDV works as follows: Each node periodically sends a Route Update (RU) message to its neighbors, which contains routing information to all other known nodes. Each node that receives the RU message updates its routing table and forwards these updates to its neighbors. Each route in the routing table has a sequence number, and a newly received route with a larger sequence number than the currently stored route is considered a better route and is updated in the routing table. When a source node needs to send data to a destination node, it finds the best path to the destination node in the local routing table and uses this path to send the packet.

In UAV communication, DSDV may be more suitable for some scenarios than other routing protocols because it provides stronger robustness and lower latency. However, the main disadvantage of DSDV is that it requires a large amount of bandwidth and computing resources to maintain and disseminate network-wide routing information. Moreover, since DSDV is based on the distance vector algorithm, it may be very sensitive to link fluctuations and errors in the network.

Coarse-grained networking communication simulation

Coarse-grained networking simulation is a set of tools to simulate cluster communication between UAVs based on Python program and coarse-grained networking interface. The architecture is shown in the figure below, where the coarse-grained networking program is the core of the coarse-grained networking simulation, and it is responsible for receiving the UAV location information, calculating the network communication link status, implementing the network routing algorithm and routing functions, completing the simulation calculation of packet loss and delay, calculating the communication quality matrix and synchronizing it to the visualization module; CopterSim is responsible for simulating other functions of UAV except communication between UAVs, and providing UAV position and speed information for coarse-grained networking simulation; The PyView client display program is a network state visualization program provided by the coarse-grained networking simulation program, which not only provides the description of the location and movement of network nodes, but also is responsible for the network topology characterization between network nodes and the visualization of the link state between nodes.



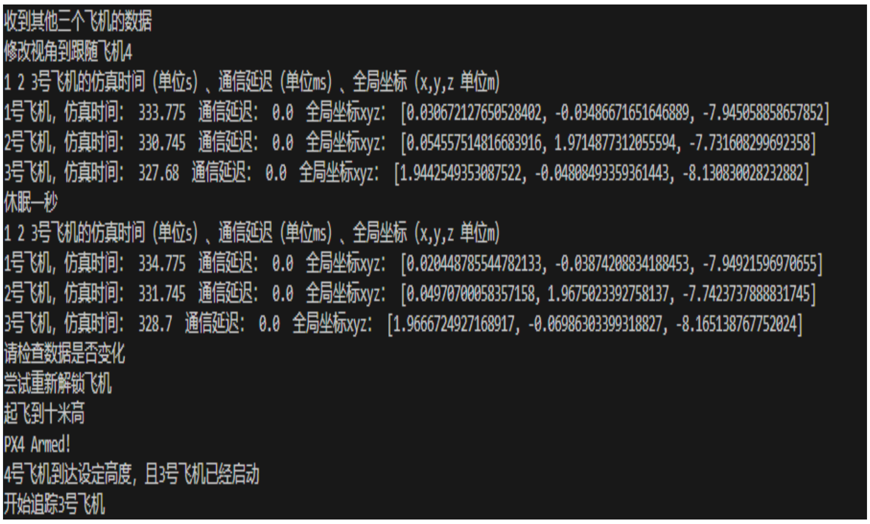
MQTT Networking Communication Simulation

MQTT (Message Queuing Telemetry Transport) communication networking simulation refers to the process of establishing and testing MQTT protocol network in a simulated environment, which is used to verify the message publish/subscribe mechanism, network topology and interaction behavior between devices. In simulation, software tools or programming language libraries can be used to simulate MQTT components in the real Internet of Things (IoT) and observe how they work together.

A typical MQTT communication network simulation environment includes the following components: MQTT Broker, Publisher, Subscriber, network configuration and message flow.

UAV MQTT communication network provides a flexible and efficient solution, which enables distributed UAV system to complete data interaction and remote control tasks efficiently and safely in complex environments.

The platform provides a solution for completing UAV cluster control by using mqtt. Examples are as follows:



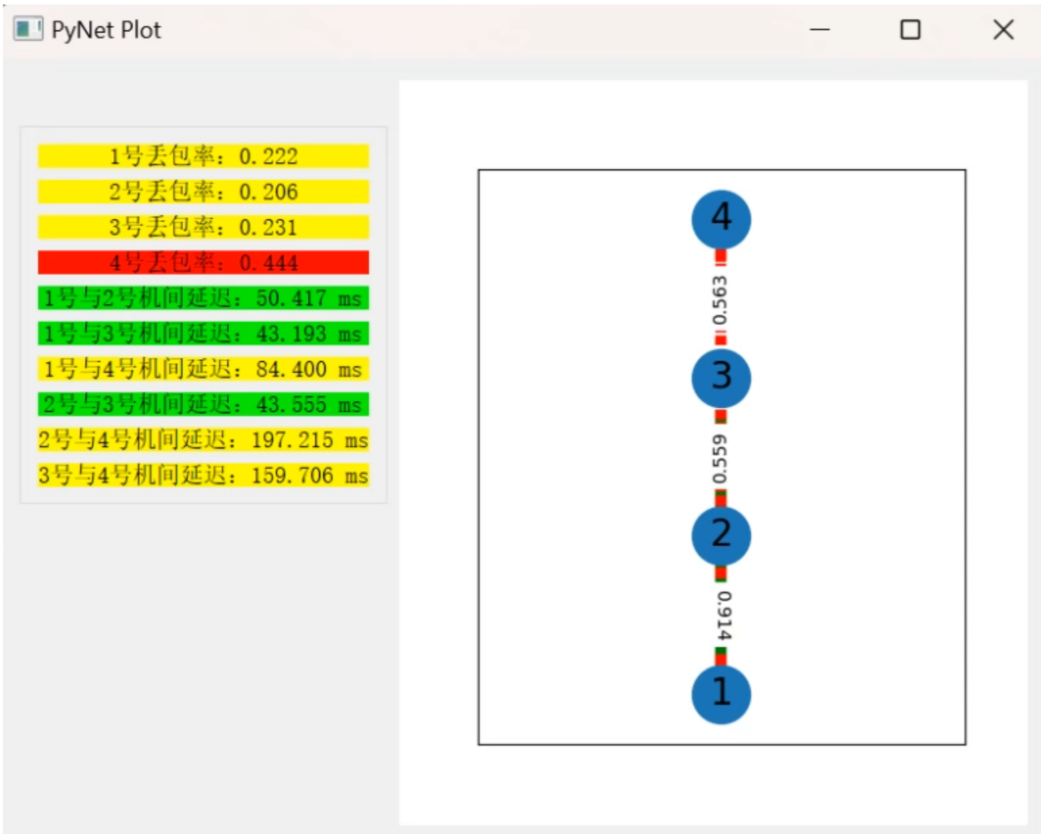
Redis networking communication simulation

The concept of UAV Redis communication network combines the characteristics of UAV technology and Redis data storage system, which may be involved in the design of distributed, real-time data processing and caching UAV network system. However, the standard Redis itself does not directly provide the function of direct communication between remote devices. It is mainly an in-memory database and cache system, which is suitable for high-performance data reading and writing operations on the server side.

The UAV transmits the collected sensor data to the central server through a wireless communication link (such as 4G/5G, satellite or ad hoc network). The Redis service is deployed on the central server to quickly store, retrieve and distribute these real-time data. Due to the memory storage and efficient performance of Redis, a large amount of real-time data can be quickly processed and analyzed.

The platform provides an example of using Redis to complete UAV communication networking.

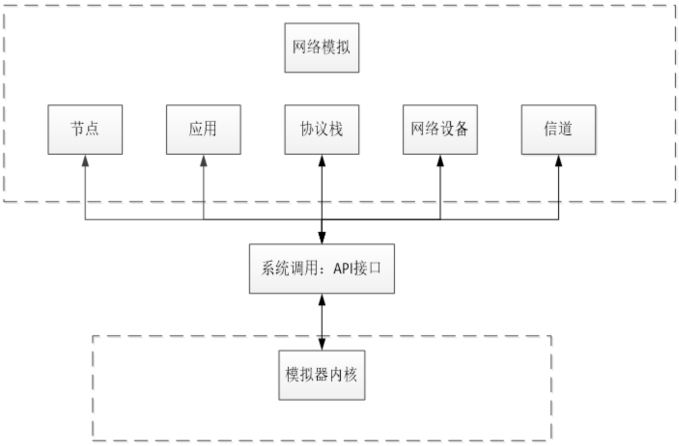
The following figure shows the example effect:



NS-3 route simulation

NS-3 is an open-source, object-oriented network simulation platform that can be used to simulate various types of networks, including wired and wireless networks. It provides a flexible emulation framework that enables users to customize network topology, protocol stack, and application behavior, and to track network events and performance metrics in detail. NS-3 is highly scalable and customizable, and can be used in various fields such as academic research, education, and industrial applications. The NS-3 also provides extensive documentation and examples to help users get started quickly and take advantage of its capabilities. Therefore, we use NS-3 as the simulation platform for fine-grained networking, which is an open source platform using C + + or Python.

In NS-3, an abstract network concept is defined as a specific network model, which is divided into Node, Application, Stack, Net Device and Channel, as shown in the figure.



The network models are linked through a system interface (API). Nodes are equivalent to computers or terminal devices in reality. Applications are installed on nodes to provide services. Protocol stacks provide protocols at all levels of the network. Network devices are the sum of abstract physical hardware and software. Channels establish communication channels between different nodes. NS-3 is a powerful open source software, based on TCP/IP protocol stack, researchers can independently design each layer of the network protocol, network performance simulation. NS-3 has rich interfaces, and the simulation results can be visualized through some drawing tools, such as Netanim, PyViz and Gunplot. NS-3 can also generate pcap and trace files, which can be captured and analyzed by WireShark tool.