

通信协议与集群组网

利用无人机实现中继通信的功能，其通信平台具有部署方便、移动性强、性能较佳、控制灵活等特点，且通信设备也容易升级换代。相对于移动通信，其部署周期要小很多，成本也相应低很多；而相比于卫星通信系统，其通信延迟较小，造价也低。因此无人机的中继通信具有一定的天然优势。

根据无人机结构和承担的任务可知，无人机所能承载的通信设备有限；在通信过程中，无人机上的通信设备一直在随其运动，它和地面通信设备的距离也一直在变化，无人机需要应对由于其移动所带来的环境变化和信号变化的不利影响，保障通信的实时性和稳定性；由于无人机多采用电池供电，而且传输的距离又远，所以要求通信设备的功耗要低、信号传输距离要远，数据链的接收灵敏度要高、接收性能良好等；在某些复杂的应用环境中，无人机需要实现高速图像数据的传输通信，由于无人机飞行的低空性，就需要考虑多路径、杂波等恶劣信道对传输性能的影响，尽量减少信息传输的误码率；在无人机多机编队和多机联合作战时，无人机需要多点通信，要兼容处理各种信号，采取抗电磁干扰措施，实现指令的有效传输和多机的协同通信。

移动网络各种场景中无人机通信的关键技术及功能

无人机功能	关键技术	无人机数目(个)
通信基站	空地海一体化通信	1
采集传感器信息并发送至基站	动态频谱共享	多
采集传感器信息并发送至基站	有限长信道编码	多
与地面用户或基站通信	超大阵列天线	1
收发信息的用户	太赫兹	4
接收信息进行自学习的用户	人工智能	多
运送药品的用户	区块链、人工智能	1

无人机常用应用层协议 (DDS、MAVlink)

DDS (Data Distribution Service for Real-Time Systems) 是一种面向数据的中间件，它为实时系统提供了一种标准的方式来发布、发现和订阅分布式数据。

DDS 的核心概念是数据发布/订阅模型，其中发布者向主题 (Topic) 发布数据，而订阅者可以从这些主题接收数据。DDS 使用一种称为“服务发现协议” (SMP 或 SDP) 的技术来发现网络上的其他 DDS 实体 (如发布者、订阅者等)。这意味着当一个 DDS 节点启动时，它可以自动找到网络上可用的主题和服务。DDS 可以利用不同的传输层协议进行通信，包括 TCP/IP、UDP、共享内存等。这使得 DDS 能够适应各种网络环境和性能需求。DDS 提供了多种服务质量 (QoS) 策略，比如可靠性、持久性、顺序保证等。这些策略可以根据应用的具体需求调整，以优化数据传输。DDS 支持定义数据类型，并确保这些类型在不同平台上保持一致。这使得 DDS 能够处理复杂的异构系统。DDS 的设计目标之一就是提供低延迟、高吞吐量的实时通信。

DDS 作为无人机应用层协议有以下优点：
实时性能：DDS 是专为实时系统设计的，能够满足无人机在飞行控制和其他关键任务中的低延迟要求。
松耦合：DDS 允许发布者和订阅者之间松散地耦合，这意味着系统中的组件可以独立开发和更新，而不会影响其他部分。
灵活性：DDS 支持动态网络拓扑，因此可以在飞行过程中添加或删除节点。这对于需要灵活调整任务配置的无人机来说非常重要。
可扩展性：由于 DDS 是基于主题的数据发布/订阅模型，所以随着系统的复杂性和规模的增长，它可以轻松地处理更多的数据流和参与者。服

服务质量 (QoS) 策略: DDS 允许用户根据具体需求选择不同的 QoS 策略, 比如可靠性、持久性、顺序保证等, 以优化数据传输。**标准化:** DDS 是由 OMG 组织制定的标准, 这有助于确保不同供应商之间的互操作性和兼容性。**ROS 集成:** DDS 与机器人操作系统 (ROS) 有很好的整合, 特别是在 ROS 2 中, DDS 作为默认的通信框架被广泛采用。**国防和工业应用:** 鉴于 DDS 最初是在军事领域开发的, 并已成功应用于多个行业, 它的可靠性和安全性对无人机应用来说是一个重要的优势。

DDS 的使用限制主要有以下方面。 DDS 具有许多高级功能和配置选项, 这使得它相对较为复杂。与一些简单的消息传递系统相比, DDS 可能需要更多的计算和内存资源来运行。为了支持实时性能, DDS 引入了一些额外的开销, 比如缓存管理、网络协议处理等。这些开销可能会影响系统的整体效率。

MAVLink 协议是一种轻量级的通信协议, 特别为小型无人航空系统 (UAS) 设计。它在无人机应用层中的作用是提供一种标准化的方式来交换飞行控制信息、传感器数据和其他与任务相关的数据。

MAVlink 作为无人机应用层协议有以下优点: 紧凑的消息结构: 消息体通常很小, 适合于低带宽和高延迟的无线链路。跨平台支持: 可以在各种硬件平台上运行, 包括嵌入式设备和 Windows 系统。易于实现: 有大量的开源库和工具可供使用, 简化了开发过程。

MAVlink 主要有以下限制。 缺乏服务质量保证: 它不提供 QoS 机制来确保关键数据的可靠传输。缺乏实时性能: 虽然可以用于实时系统, 但不是专门设计为满足严格的实时要求。没有内置安全功能: 需要外部机制来保护数据的安全性和完整性。

相比较而言, 两种协议具有相似的功能, 而具有不同的适用范围。主要区别为 DDS 是一种更为复杂也更加强大的实时数据分发协议。DDS 设计方向是为了解决大规模的分布式系统中的数据共享问题, 因此更适用于需要强大数据管理和分发能力的大型、复杂的分布式系统。并且允许用户根据自身需求配置不同的 QoS 策略, 定制数据传输。MAVlink 相比而言更适用于简单的无人机应用, 尤其在资源受限的环境中。

无人机常用路由协议 (AODV、OLSR、DSR、DSDV)

AODV: Ad-hoc On-Demand Distance Vector Routing (AODV) 是一种用于移动自组网 (MANET) 的路由协议。在这样的网络中, 没有固定的基础设施或中心控制节点, 每个节点都必须能够作为路由器转发数据包, 以实现整个网络的通信。由于这种网络环境中的节点可以随意移动, 因此网络拓扑可能会频繁变化。

其工作原理如下: 当源节点需要发送数据到目的节点, 并且在本地路由表中没有可用路由时, 它会广播一个“路由请求” (RREQ) 消息。RREQ 消息沿着网络传递, 每个收到它的节点都会记录下该请求的来源并转发给其邻居。目的节点接收到 RREQ 后, 回复一个“路由应答” (RREP) 消息, 其中包括到达源节点的最短路径信息。路由应答沿原路径返回, 每个节点在收到应答后更新其路由表, 以便将来可以通过该路径直接发送数据。

在无人机自组网中, AODV 协议的应用非常关键。由于无人机通常具有高度移动性, 它们之间的通信网络必须能够快速适应不断变化的拓扑。此外, 由于无人机可能执行远程任务或者处于偏远地区, 网络带宽和能量效率都是重要的考虑因素。

AODV 的优势在于其按需特性, 这意味着它仅在实际需要时才创建和维护路由, 这有助于减少不必要的网络开销。然而, 对于高动态性的场景, 如多无人机协同飞行, AODV 可能需要进行优化以提高路由稳定性、降低延迟和改善整体性能。

OLSR: Optimized Link State Routing Protocol (OLSR) 是一种用于移动自组网 (MANET) 的链路状态路由协议。与 AODV 这样的距离向量协议不同, OLSR 使用了全局网络拓扑信息来进行路由决策。

其工作原理如下：每个节点周期性地发送 Hello 消息给其邻居，以检测和更新邻居列表。基于邻居列表，每个节点选择一组 MPR 节点，并在下一次 Hello 消息中宣布这个选择。当收到一个 Hello 消息时，节点会更新其邻居和 MPR 集合。节点根据邻居和 MPR 信息生成一个 TC 消息，其中包含所有可达节点的信息。TC 消息通过 MPR 节点进行转发，从而在整个网络中传播拓扑信息。每个节点根据接收到的 TC 消息更新其路由表，以便为数据包找到最佳路径。

在无人机通信中，OLSR 可能比 AODV 更适合一些场景，因为它提供了更好的网络可视性和更快的收敛速度。然而，OLSR 的主要缺点是它需要更多的带宽和计算资源来维护和传播拓扑信息。

DSR: Dynamic Source Routing (DSR) 是一种用于移动自组网 (MANET) 的源路由协议。与 AODV 和 OLSR 这样的传统路由协议不同，DSR 不需要每个节点都维护整个网络的拓扑信息，而是让数据包携带完整的路径信息。

DSR 的工作原理如下：当源节点需要发送数据到目的节点时，它会在其路由缓存中查找是否有到达目的节点的有效路由。如果找到有效路由，则直接使用该路由发送数据包。如果没有找到有效路由，则源节点会广播一个“路由请求” (Route Request, RREQ) 消息。RREQ 消息沿着网络传递，每个收到它的节点都会记录下该请求的来源并转发给其邻居。目的节点接收到 RREQ 后，回复一个“路由应答” (Route Reply, RREP) 消息，其中包括到达源节点的最短路径信息。路由应答沿原路径返回，每个节点在收到应答后更新其路由缓存，以便将来可以通过该路径直接发送数据。

在无人机通信中，DSR 可能比其他路由协议更适合一些场景，因为它提供了更低控制开销和更高的灵活性。然而，DSR 的主要缺点是它对网络中的环路非常敏感，需要采取额外的措施来避免环路。此外，由于每个数据包都需要携带完整的路径信息，这可能会导致数据包头部过大，从而影响传输效率。

DSDV: Destination-Sequenced Distance Vector (DSDV) 是一种用于移动自组网 (MANET) 的距离向量路由协议。与 AODV 和 OLSR 这样的按需路由协议不同，DSDV 使用周期性更新机制来维护网络中的路由表。

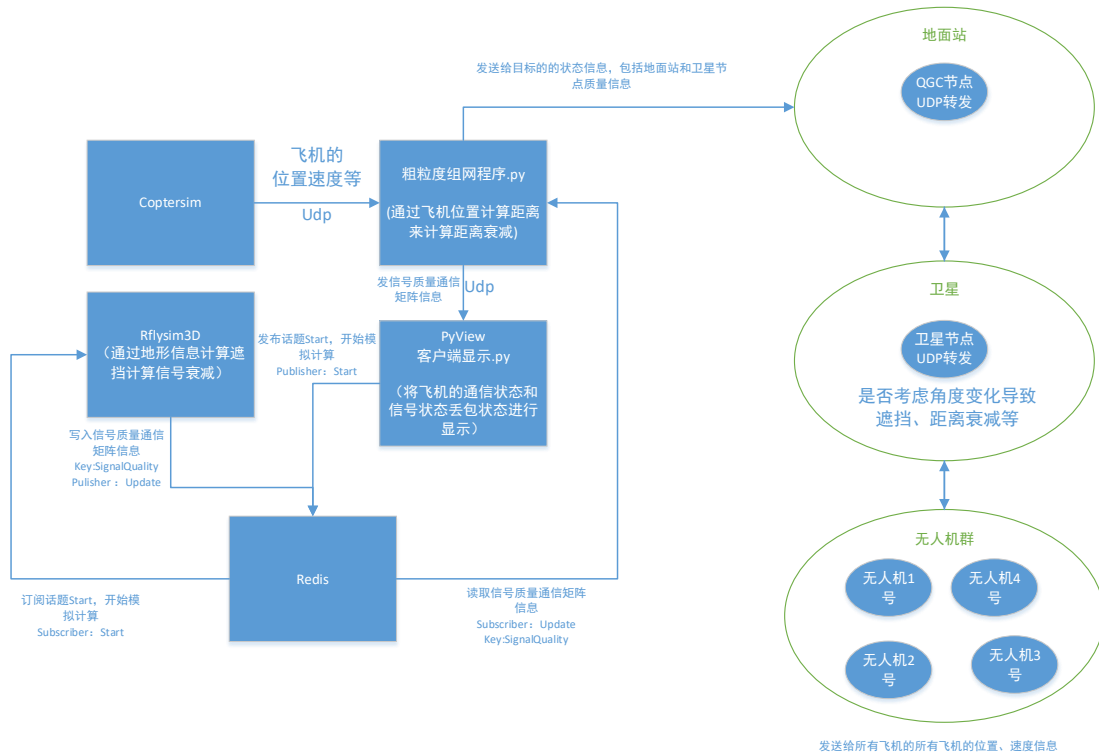
DSDV 的工作原理如下：每个节点定期发送“路由更新” (Route Update, RU) 消息给其邻居，该消息包含了到其他所有已知节点的路由信息。每个收到 RU 消息的节点会更新其路由表，并将这些更新转发给它的邻居。路由表中的每条路由都有一个序列号，新接收到的路由如果序列号比当前存储的路由更大，则会被认为是更好的路由，从而被更新到路由表中。当源节点需要发送数据到目的节点时，它会在本地路由表中查找到达目的节点的最佳路径，并使用这个路径发送数据包。

在无人机通信中，DSDV 可能比其他路由协议更适合一些场景，因为它提供了更强的健壮性和更低的延迟。然而，DSDV 的主要缺点是它需要大量的带宽和计算资源来维护和传播全网范围的路由信息。此外，由于 DSDV 基于距离向量算法，它可能对网络中的链路波动和错误非常敏感。

粗粒度组网通信仿真

粗粒度组网仿真是基于 Python 程序和粗粒度组网接口编写的一套模拟无人机之间集群通信的工具。其架构如下图所示，其中粗粒度组网程序是粗粒度组网仿真的核心，其负责完成接收无人机位置信息、网络通信链路状态计算、网络路由算法实现与路由功能、完成丢包和延迟的模拟计算、计算通信质量矩阵并同步给可视化模块；CopterSim 负责模拟无人机除无人机之间通信外的其他功能，给粗粒度组网仿真提供无人机位置、速度信息等；PyView 客户端显示程序是粗粒度组网仿真程序提供的网络状态可视化程序，其不仅提供了对网络节点

位置、运动的描述，还负责网络节点间的网络拓扑刻画，节点间链路状态可视化等。



MQTT 组网通信仿真

MQTT (Message Queuing Telemetry Transport) 通信组网仿真是指在模拟环境中建立和测试 MQTT 协议网络的过程，用于验证消息发布/订阅机制、网络拓扑结构以及设备间的交互行为。在仿真中，可以使用软件工具或编程语言库来模拟实际物联网 (IoT) 中的 MQTT 组件，并观察它们如何协同工作。

一个典型的 MQTT 通信组网仿真环境包括以下组成部分：**MQTT Broker**(代理服务器)，**Publisher** (发布者)，**Subscriber** (订阅者)，网络配置和消息流。

无人机 MQTT 通信组网提供了一种灵活、高效的解决方案，使得分布式无人机系统能够在复杂环境下高效、安全地完成数据交互与远程控制任务。

平台提供使用 `mqtt` 完成无人机集群控制解决方案，实例如下：

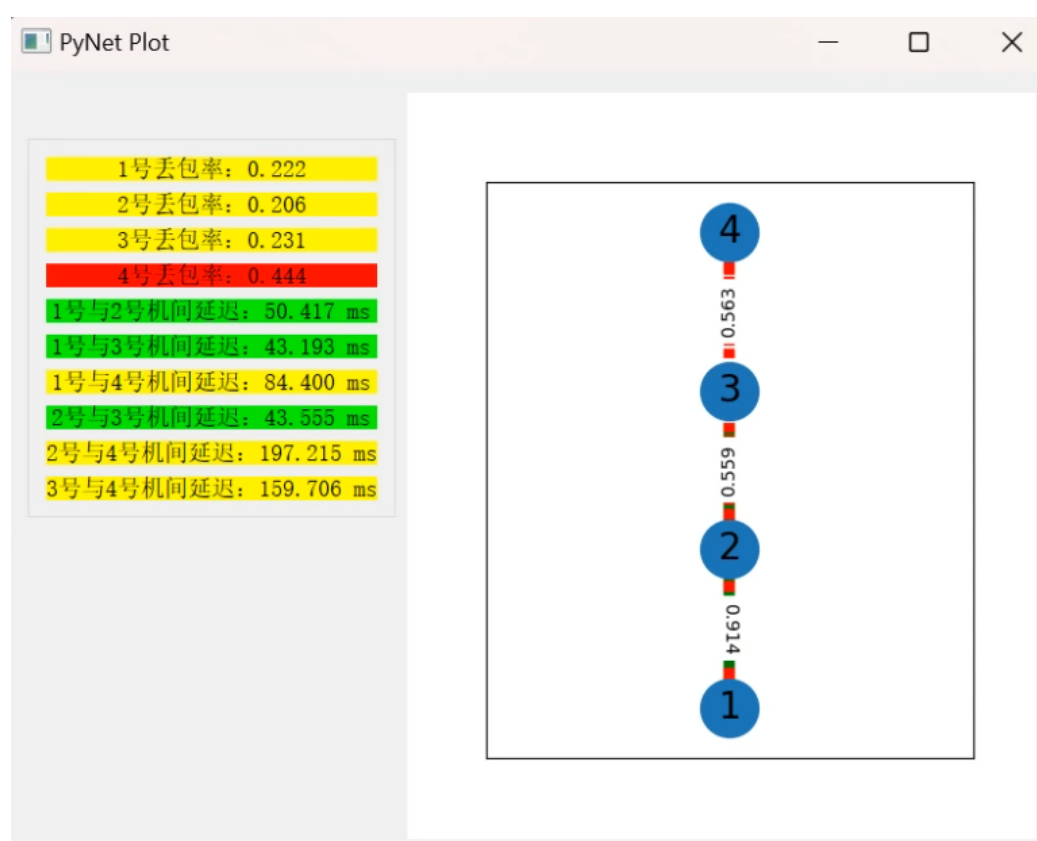
```

收到其他三个飞机的数据
修改视角到跟随飞机4
1 2 3号飞机的仿真时间(单位s)、通信延迟(单位ms)、全局坐标(x,y,z 单位m)
1号飞机, 仿真时间: 333.775 通信延迟: 0.0 全局坐标xyz: [0.030672127659528402, -0.03486671651646889, -7.9450585858657852]
2号飞机, 仿真时间: 330.745 通信延迟: 0.0 全局坐标xyz: [0.054557514816683916, 1.9714877312055594, -7.731608299692358]
3号飞机, 仿真时间: 327.68 通信延迟: 0.0 全局坐标xyz: [1.9442549353087522, -0.04008493359361443, -8.130830028232882]
休眠一秒
1 2 3号飞机的仿真时间(单位s)、通信延迟(单位ms)、全局坐标(x,y,z 单位m)
1号飞机, 仿真时间: 334.775 通信延迟: 0.0 全局坐标xyz: [0.020448785544782133, -0.03874208834188453, -7.94921596970655]
2号飞机, 仿真时间: 331.745 通信延迟: 0.0 全局坐标xyz: [0.04970700058357158, 1.9675023392758137, -7.7423737888831745]
3号飞机, 仿真时间: 328.7 通信延迟: 0.0 全局坐标xyz: [1.9666724927168917, -0.06086303399318827, -8.165138767752024]
请检查数据是否变化
尝试重新解锁飞机
起飞到十米高
PX4: Armed!
4号飞机到达设定高度, 且3号飞机已经启动
开始追踪3号飞机
    
```

Redis 组网通信仿真

无人机 Redis 通信组网这一概念融合了无人机技术与 Redis 数据存储系统的特性，通常在设计分布式、实时数据处理和高速缓存的无人机网络系统中可能会涉及。然而，标准的 Redis 本身并不直接提供用于远程设备间直接通信的功能，它主要是一个内存数据库和缓存系统，适合于在服务器端进行高性能的数据读写操作。

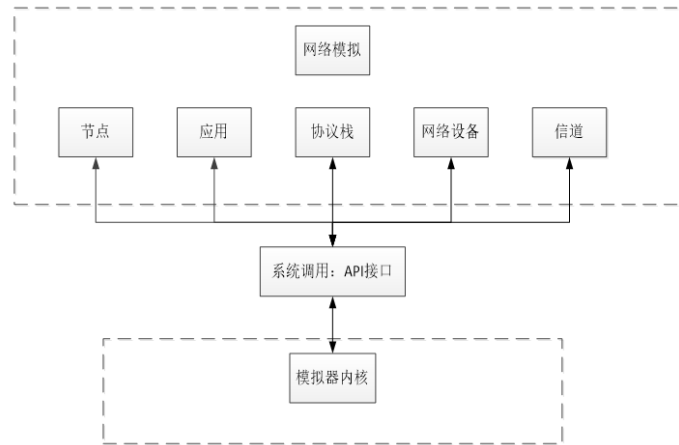
无人机通过无线通信链路（如 4G/5G、卫星或自组网）将采集到的传感器数据传输到中心服务器。中心服务器上部署 Redis 服务，用于快速存储、检索及分发这些实时数据，由于 Redis 的内存存储和高效性能，使得大量实时数据能够迅速被处理和分析。平台提供有使用 Redis 完成无人机通信组网的示例，下图是实例效果：



NS-3 路由仿真

NS-3 是一个开源的、面向对象的网络仿真平台，可用于模拟各种类型的网络，包括有线和无线网络。它提供了一个灵活的仿真框架，使用户能够自定义网络拓扑、协议栈和应用程序行为，并能够详细跟踪网络事件和性能指标。NS-3 具有高度的可扩展性和可定制性，可以应用于学术研究、教育和工业应用等各种领域。NS-3 还提供了丰富的文档和示例，以帮助用户快速入门并利用其功能。因此，我们采用 NS-3 作为细粒度组网的仿真平台，它是采用 C++ 或者 Python 的开源平台。

在 NS-3 中，将抽象的网络概念定义为了具体的网络模型，分为节点 (Node)、应用 (Application)、协议栈 (Stack)、网络设备 (NetDevice)、信道 (Channel)，如图所示。



各网络模型通过系统调 (API) 来建立其联系。节点相当于现实中的计算机或终端设备，应用安装在节点上以提供服务，协议栈提供网络各层协议，网络设备是抽象的物理硬件和软件的总和，信道建立不同节点之间的通信通道。NS-3 功能强大，是一个开源软件，基于 TCP/IP 协议栈，研究人员能够独立的设计各层网络协议、进行网络性能仿真。NS-3 具有丰富的接口，仿真结果可以通过一些绘图工具直观看到，比如 Netanim、PyViz 和 Gunplot。NS-3 也能产生 pcap 和 trace 文件，能用 WireShark 工具抓包分析。