

DDS 发布订阅通信节点自动发现机制与改进

李大超¹, 孙帷胜², 张志涛²

(1. 海装驻上海地区第十军事代表室, 上海 200241;

2. 中国航空无线电电子研究所, 上海 200233)

[摘要] DDS 作为一种新兴的非确定性网络中的高可用性分布式数据分发规范, 提出了全局数据空间的概念, 通过发布订阅机制进行通信, 然而当前航电网络环境下, 对于应用来说, DDS 封装严密, 很多过程对于应用来说属于黑盒过程, 难以准确把握其流程, 一旦出现异常和故障也难以定位和排故。本文以航电以太网为例, 详细研究 DDS 发布订阅过程与通信行为, 并以一个故障研究与排除实例, 说明 DDS 发布订阅具体流程和逻辑, 并对 DDS 在以太网协议层和应用层的通信行为进行研究。

[关键词] 数据分发服务; 发布订阅机制; 通信行为

[中图分类号] TP311.52

[文献标识码] A

[文章编号] 1006-141X(2023)01-0020-09

Research on DDS Publish-Subscribe Communication Behavior and Improvement

LI Da-chao¹, SUN Wei-sheng², ZHANG Zhi-tao²

(1. NO.10 Military Representatives Office of Navy in Shanghai Area, Shanghai 200241, China;

2. China Aeronautical Radio Electronics Research Institute, Shanghai 200233, China)

Abstract: As a new high-availability data distribution specification in non-deterministic networks, DDS proposes the concept of global data space and communicates through a publish-subscribe mechanism. It is said that DDS is tightly packaged, and many processes considered as black box process to the applications, and it is difficult to accurately grasp the process, and it is difficult to locate and troubleshoot once abnormalities and faults occur. Avionics Ethernet is taken as an example to study the DDS publish-subscribe process and communication behavior in detail, and an example of fault research and troubleshooting is used to illustrate the specific process and logic of DDS publish-subscribe mechanism. And the communication behavior of DDS at Ethernet protocol layer and application layer is studied.

Key words: data distribution service; publish-subscribe mechanism; communication behavior

实时数据分发服务 (DDS: Data Distribution Service) 技术目前已经广泛应用于各个机载项目, 由于其通过全局数据空间^[1]的概念, 基于发布订阅关系而不是基于传统的连接关系进行应用数据传递

和分发, 使应用层和底层通信链路解耦, 从而显著提高了开发效率, 降低了开发风险。同时, 支持跨平台、跨总线通信, 并可实现异构平台通信; 大大降低集成新组件和修改原有组件通信路径时的工作

收稿日期: 2022-11-11

引用格式: 李大超, 孙帷胜, 张志涛. 航空电子领域虚拟化技术应用研究 [J]. 航空电子技术, 2023, 54(1): 20-28.

量和技术风险, 也使系统具有可持续集成和长期可扩展性。因而广受机载应用开发人员的欢迎。

实时数据分发服务(DDS: Data Distribution Service)技术目前已经广泛应用于各个机载项目, 由于其通过全局数据空间^[1]的概念, 基于发布订阅关系而不是基于传统的连接关系进行应用数据传递和分发, 使应用层和底层通信链路解耦, 从而显著提高了开发效率, 降低了开发风险。同时, 支持跨平台、跨总线通信, 并可实现异构平台通信; 大大降低集成新组件和修改原有组件通信路径时的工作量和技术风险, 也使系统具有可持续集成和长期可扩展性。因而广受机载应用开发人员的欢迎。

然而, DDS 的通信虚拟化封装和黑盒特性也导致了应用对于低层通信行为难以确切获悉和控制, 从而导致黑盒效应, 即由于 DDS 协议的高封装性, 导致应用层对于 DDS 协议栈的具体行为和问题产生机理不能掌握, 从而对于某些异常和故障, 无法当场立即推断或定位故障位置和原因, 因而对以太网协议层和应用层的 DDS 通信行为进行研究对于深入理解 DDS 原理, 推进对 DDS 原理、开发和应用的研 究, 并支持相关排故工作来说是非常有必要的。

1 DDS 以太网通信行为研究

DDS 建立通信需要进行多个步骤: 首先是进行域参与者自动发现, 之后进行发布者/订阅者自动发现, 最后正式进行信息传输, 分布式数据分发服务结束之后进行底层连接的中断和通信实体的析构。

DDS 中间件自动将数据从发布者推送到订阅者需要多种类技术支撑, 包括内部数据结构的限定、发布/订阅关系匹配算法和逻辑和以及远程自动发现/自组网技术。

这种协议封装切实降低了应用在通信方面的负担, 使应用只需要关注数据主题索引即可实现高效的数据传输; 而另一方面对于应用来说, 也带来了黑盒程度过高的问题, 因而对 DDS 发布订阅通信节点自动发现机制进行详细研究并基于实际应用效果进行改进, 就成为亟待完成的一项工作。

作为关键技术的远程自动发现机制, 是通过双层实体的自动自主通讯伙伴发现, 建立虚拟的“全局数据空间”, 使不同分布式节点能够相互发现域参与者、数据写者和数据读者, 从而自动建立发布/

订阅关系。

自动发现技术在后台为每个域参与者构建了全局的发布/订阅关系拓扑图, 域参与者能够与远程通信的其他域参与者构建虚拟连接, 准确地将上层传递的相关主题数据通过发布订阅关系由发布者“投递”到订阅者。

基于以上分析的 DDS 内部核心实体角色的功能定位和逻辑关系, 将自动发现技术划分为两个逻辑层: 域参与者层、发布者/订阅者层。域参与者层主要以网络节点为单位, 完成本地域内参与者与远程域参与者之间的自动发现, 为域参与者构建与其具有通信关系的其他域参与者之间的, 以网络内参与通信的节点为单位的拓扑结构; 发布/订阅端点层是在构建的域参与者拓扑关系的基础上, 具体构建发布端点和远程订阅端点之间的发布/订阅关系, 建立分布式节点之间的虚拟连接^[2]。域参与者主要面向应用和参与通信的主机节点, 而发布者/订阅者层主要以数据和数据主题为中心。两层之间具有严格的先后顺序关系, 只有域参与者自动发现完成后, 才能进入发布/订阅端点自动发现。

完成了自动发现后, DDS 将开始调用底层的通信接口进行实际的数据负载传输, 这个过程同样需要解决一系列问题, 首先是对数据传输的质量控制(QoS)的实现, 诸如传输可靠性(reliability)、端点断言活跃度(liveness)等策略的实现方式, 以及本机端口调度等策略的实现等。综上, 总体来讲, DDS 的自动发现机制由以下几个协议和算法组合而成:

(1) 域参与者和发布者/订阅者自动发现机制, 该机制主要由以数据为中心的“发布-订阅”(DCPS: Data-Centric Publish-Subscribe)协议实现;

(2) 数据结构一致性判定机制, 该机制主要由数据本地重构层(DLRL: Data Local Reconstruction Layer)协议实现;

(3) QoS 机制;

(4) 端口防碰撞算法。

2 自动发现机制

2.1 域参与者自动发现机制

以数据为中心的“发布-订阅”协议, 即 DCPS 是 DDS 自动发现机制的核心和基础。在该层协议

中,域参与者层自动发现通过分布式节点互相交换“域参与者描述信息” Participant-data, RTPS 协议中表述为 Data (p), 并实时动态维护该信息组成的数据库来实现。Participant-data 内部包含了参与者的全局唯一标识符 (Key)、配置的 QoS 策略等信息,通过内建的专门用于订阅和发布此主题数据的内建数据读者 builtin-DataReader 和内建数据写者 builtin-DataWriter 完成交互;这两个实体对用户是不可见的,由 DDS 内部来操作和管理。

以 A 和 B 两个分布式节点为例(下文称为 NodeA 和 NodeB),如图 6 所示。NodeA 创建本

节点的 Participant, 并通过内建数据写者 builtin-DataWriter 向外周期广播 NodeA 的 Participant-data, 当 NodeB 创建本节点的域参与者 Participant 后,也会通过内置的 builtin-DataWriter 向外随机周期广播 NodeB 的 Participant-data。当 NodeA 通过内置的 builtin-DataReader 发现 NodeB 的 Participant-data 后,更新本地的关系数据库,添加新发现节点 NodeB 的 Participant-DATA, 同理 NodeB 对 NodeA 也是如此。同时当 NodeA 再次收到 NodeB 的 Participant-DATA, 发现已经存在本地信息库中时,就不再做任何处理,如图 1 所示。

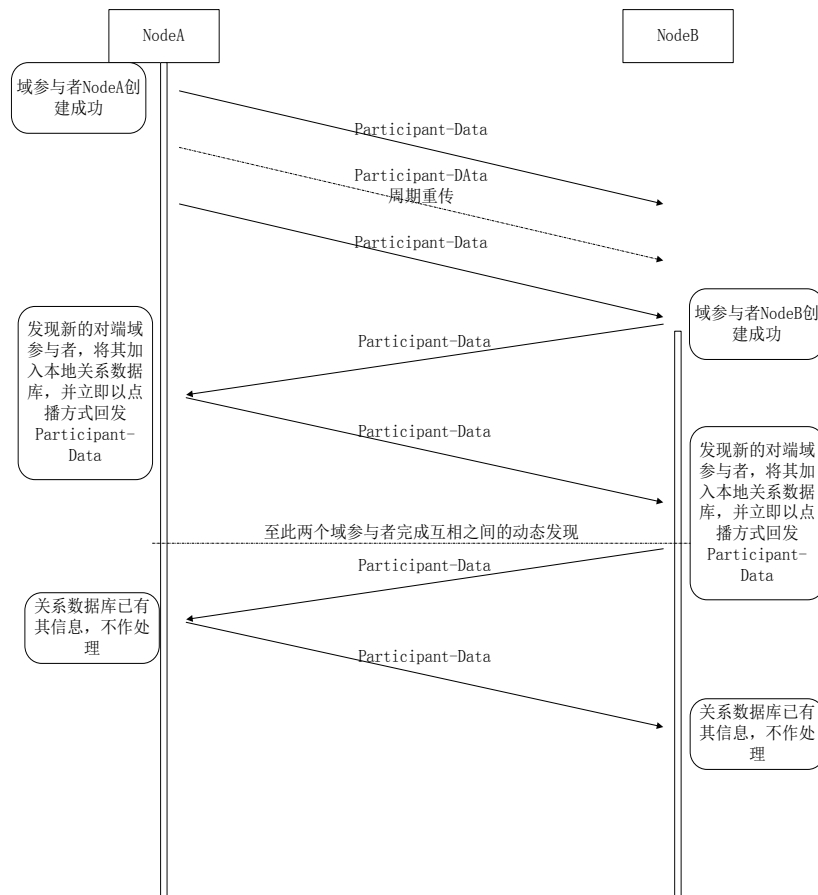


图 1 域参与者自动发现机制

2.2 发布者 / 订阅者自动发现机制

发布 / 订阅端点自动发现技术目的在于构建“全局数据空间”的虚拟连接,通过域参与者之间交换其所属的发布端点描述信息 (publicationDATA) 和订阅端点描述信息 (subscribeDATA) 来实现。publicationDATA 和 subscribeDATA 的交换通过数据分发服务中间件内置的两对数据写者和数据读者发布和

订阅完成,分别是 pb-DataReader、pb-DataWriter 和 sb-DataReader、sb-DataWriter。

当 DDS 相关实体创建完成后,使用“内建”(built-in) DDS 数据读者 (DataReaders) 和内建数据写者 (DataWriters) 进行发现操作,利用域参与者发现协议 (PDP: Participant Discovery Protocol) 和终端发现协议 (EDP: Endpoint Discovery Protocol),

每个域参与者将会创建一个内建的 DDS 数据读者 (DataReaders) 和内建数据写者 (DataWriters)，其中，数据写者将会周期性地发送数据实例到一组预设的地址中以声明域参与者的存在^[3]，并与其他域参与者的内建数据读者和数据写者的通信来交换以下内容，如表 1 所示。

表 1 RTPS 部分协议信息

变量名	名称	解释
ProtocolVersion	协议版本	用于标识该 RTPS 数据包所符合的协议版本
guidPrefix	GUID 标签	全局 ID 的标签
VendorId	供应商 ID	供应商的 ID
expectsInlineQos	预期内置 Qos	对可与该数据交互的主体所要采用的 QOS 策略的一个声明

其中 guidPrefix 标签由三部分内容组成：Host_id、App_id、Instance_id，每部分均为 4 字节长度，如表 2 所示。

表 2 guidPrefix 信息格式

GUID		
Host_id	App_id	Instance_id
4 字节	4 字节	4 字节

域参与者层自动发现结束后，会触发域参与者之间的匹配事件，即域参与者与远程域参与者之间的 pb-DataReader/sb-DataReader 和 pb-DataWriter/sb-DataWriter 进行匹配，匹配成功以后，通过 reliable 方式进行发布和订阅。在创建 DataReader/ DataWriter 时，域参与者基于已构建的域参与者通信关系信息库，向与其通信的其他域参与者广播新创建的 Reader /Writer 的描述信息 subscribeDATA 和 publicationDATA，以声明该域参与者创建了新的发布或订阅端点，其他远程域参与者收到后，将这些信息与本地订阅端点或发布端点匹配，若成功，则建立新的发布 / 订阅关系，并将这些信息添加到本地发布 / 订阅端点信息库。另外，当域参与者所属的 DataReader/DataWriter 的 QoS 策略发生变化或被删除时，也会触发 publicationDATA/subscribeDATA 的发布事件^[2]。以 NodeA 内部创建数据作者 DataWriter C 为例，发布 / 订阅端点自动发现流程如图 2 所示。

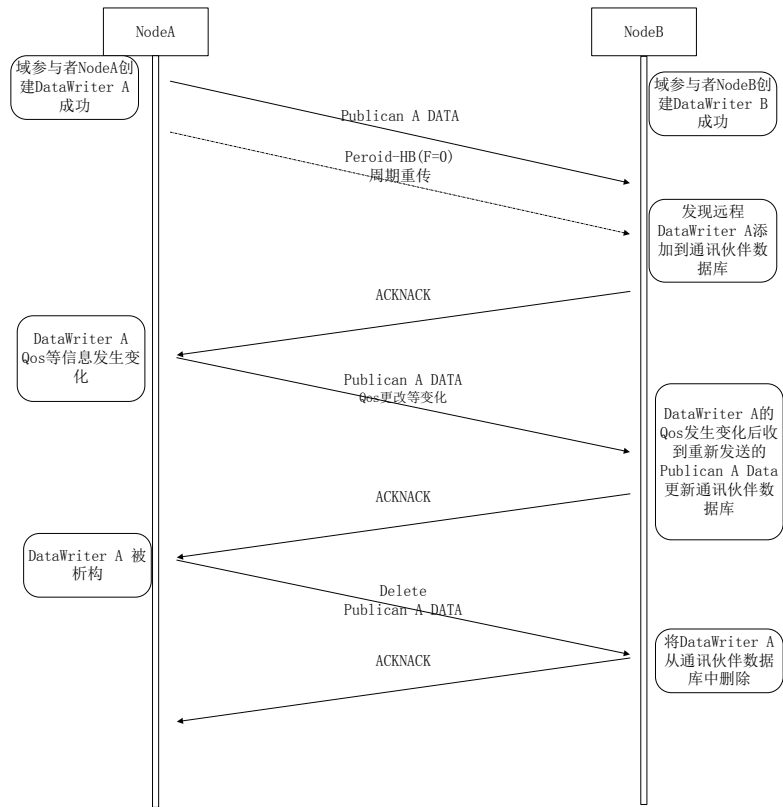


图 2 发布 / 订阅端点自动发现流程

但进行了自动发现后并不意味着可以直接进行数据传输，DDS 还具有其他机制和规约来确保数据一致性和连接可靠性，目的是确保连接的稳定和数据的可靠传输。

3 通信配置机制

完成发布 / 订阅的握手关系后，即进入正式的数据传输环节，这时，收发两端的数据读者和数据写者均获得了对方的 IP 地址、进程 ID 等信息，并得知对方发布或者订阅的主题 Topic 是什么，当一个数据读者订阅的数据主题与一个数据写者发布的数据主题相同时，则有结成发布订阅关系的潜在可能，之所以说是潜在可能是因为除了发布订阅数据主题一致，还需要其他多种条件。为满足这些条件，需要对应的机制和算法来对通信方式进行配置，这些条件限制是为了确保通信的可行性。以保证已经互相发现的两端之间的通信能够切实进行。

3.1 数据结构一致性

另外，DDS 在数据本地重构层（DLRL: Data Local Reconstruction Layer）协议层已经通过将数据序列化（serized）实现了通信负载与数据结构的强耦合，所以发布订阅两端的数据主题的数据类型名 type_name 和数据类型结构 type_object（工程应用中有时记为 type_code）同样必须一致。该设计的一个明显优势是使业务数据与链路层的数据表征方式解耦，如大小端，字节序等数据一致性问题将被 DLRL 确保。因而无需应用层处理。

其中 type_name 为应用通过数据结构注册接口对数据结构定义的名称，type_object 则表述具体的数据结构定义，以 IDL 语言的形式^[4]。

3.2 Qos 双端兼容性

OMG 组织所定义的数据分发服务 Qos 部分配置因为其定义方式互相之间会有影响，或者其实现方法存在互斥部分，故而存在兼容性限制^[5]，其中主要项目如表 3 所示。

表 3 DDS 双端兼容性限制简表

策略名称	释义	作用	规则
Deadline	截止日期，指对数据样本更新频率的限制。	如果规定时间内不能更新一次数据样本，则不予进行发布订阅	收端 Deadline 要求必须比发端宽松，或者保持一致
Reliability	可靠性，指对数据送达可靠性的保证	Reliability 设定为 reliable，则 DDS 通过一次握手的方式保证传输数据到达，如果设定为 best effort 则为尽力传输	在收端为 reliable 时，则发端必须也是 reliable，或者需要保持一致。
Latency budget	时延预计，指对时延规定的最大差值。	指对时延规定的最大差值。超过该差值的将被丢弃。	收端 Latency budget 要求必须比发端宽松，或者保持一致
Durablity	持续性，确定是否要让 DDS 中间件向新接入网域的数据发布之前发送过的数据	确定是否要让 DDS 中间件向新接入网域的数据发布之前发送过的数据，以及配置其具体发布方案	收发两端必须一致
Ownership	独占权，配置 DDS 多个发布者一个订阅者时，是否允许订阅者得到相同 keyID 的数据实体的多个数据样本	当为“shared”时，可以接受多个写者的数据，当为“exclusive”时，只能接受 Ownership strength 最高的数据	收发两端必须一致
Liveliness	活跃度，用于决定该 DDS 实体存活性和活跃程度（通信功能可用度）各个判定指标和阈值的策略	包括 Liveliness Durablity 和 Liveliness kind 两项用于决定该 DDS 实体存活性和活跃程度（通信功能可用度）各个判定指标和阈值	收发两端必须一致
Destination order kind	收发顺序，用于决定收端以什么样的顺序收取数据	具有以发时间戳顺序收取、收时间戳顺序收取等选项	收发两端必须一致
Partition	分区	在域的区分内进一步划分通信分区，默认不分，但划分了的话则只有同一分区内的 Topic 可以互相通信	收发两端必须一致

3.3 基于以太网 UDP 的端口防碰撞算法

在基于以太网 UDP 的 DDS 的实际应用中, 一个很大的问题是如何保证 DDS 所占用的 UDP 端口在自己内部存在多个 Topic 和多个域参与者时不会因为互相抢占端口造成通信问题。为解决这个问题, DDS 在实际应用于 UDP 网络时往往需要一套 UDP 端口防碰撞算法。

DDS 通信过程中可用端口号由以下 2 个公式确定^[6]:

当使用 UDP 组播时适用:

$$7400 + (250 \times D) + 0 \quad (1)$$

当使用 UDP 单播时适用:

$$7400 + (250 \times D) + (2 \times p) + 10 \quad (2)$$

其中的 D 指的是初始化域时所输入的域 ID, 而 p 指的是域的域参与者序号。

一个进程里面同一个域的域参与者序号相同, 不同进程的同一个域的域参与者序号是从 0 开始依次加 1 的正整数。每个进程的同一个域的域参与者序号取决于进程调用初始化域操作的先后顺序, 这就是说最先调用的进程域参与者序号为 0, 其它进程的域参与者序号根据初始化域顺序依次加 1。当 DDS 使用组播通信时 (包括内建 Topic 进行发现操作等) 时适用公式 1, 当使用 UDP 的点播时使用公式 2, 这 2 个公式分别表示发现组播占用端口和发现单播占用端口。其中, 每一个域参与者的域参与者序号 p 可以通过 DDS 的域参与者自动发现机制来进行排序——由于域参与者可以通过域参与者自动发现机制得知本机其他域参与者是否存在, 从而可确定自身在本机的序号。

DDS 的域号取值为 0 到 232, 域参与者序号为 0 到 65535, 由公式 1 可以得出发现组播端口号占用范围为 7400 到 65400, 由公式 2 可以得出发现单播端口号占用范围为 7410 到 196480。并且一个 DDS 实体创建之后在其生命周期内, 占用的 DDS 端口号便不再变动。

该设计同样可以拓展移植到其他总线和信道上。

4 分析实例

如前文所述, DDS 建立通信关系本身分为两个步骤, 首先需要进行域参与者的创建和彼此发现,

之后基于域参与者创建数据读者和写者, 进行切实的数据发布和订阅。这个过程同样有可能会发生问题。而根据对自动发现机制的研究可以知道故障的定位和排除。现以两个故障实例来说明。

4.1 自动发现问题

某开发项目出现过疑难故障: 调用了 DDS 中间件接口进行 DDS 通信的节点 A 反复重启后, 发现对端节点 B 无法正常接收该节点发送的 DDS 信息, 通过监控工具监控发现, DDS 数据正常发出, 但对端节点 B 没有正常接收到。

根据排故观察结果, 利用相关工具进行排故和现象观察, 本故障发生时, DDS 数据包与 RTPS 协议相关数据包 (主要包括心跳 heartbeat 包和持续性 liveness 两种) 在节点 A 发送数据后, 应用发出了数据, 但对端接收失败, 且一般需要重启计算机, 或者重启网卡 (禁用后启用) 才能恢复, 但如果再次重新启动 DDS 中间件和应用, 则有概率再次出现。

DDS 本身域参与者进行互相发现的机制, 以 A 和 B 两个分布式节点为例, 根据前文所述域参与者自动发现机制: 当域参与者完成互相之间的发现后, 两端域参与者下属的数据读者和数据写者通过同样的过程互相发现, 即交换 DataReader-data(data(r)) 和 DataWriter-data(data(w)), 同时完成 Topic 发布订阅关系的匹配, 若完成, 则可以通过以太网进行实际的业务数据通信, 如图 3。

根据前文所述域参与者自动发现机制: DDS 通过全局唯一 ID (GUID) 机制来标示不同的域参与者实体, 其中, Host_id 由于与 IP 地址对标, 出错可能性极小, 即便 IP 地址有重复的, 也会在以太网层面暴露出来, Instance_id 由于本身是在一个应用程序实例内部通过一个全局变量来进行标示, 本身具有防碰撞算法和相关的辅助可靠性算法来保证其不会重复, 因而出错可能性同样很小。只有 App_id, 由于默认是与进程 ID 相同, 据实际环境分析, 在频繁启动含有域参与者的创建的应用时, 会出现这种情况: 原有创建了用于通信的域参与者 A 的节点 A 进程 1 已关闭, A 的 GUID1 如下:

表 4 节点 A GUID1 情况

GUID1		
Host_id	App_id	Instance_id
【本机 IP 地址】	【进程 1 process ID】	0

重新启动的节点 A 进程 2，操作系统因为原进程 1 已终止，1 的 process ID 已释放，有一定概率（事实上为了保证剩余的 process ID 不会过快消耗，Windows 的算法倾向于尽可能复用释放了的 process ID）使用进程 1 的 process ID，导致进程 2 的 process ID= 进程 1 的 process ID，而 DDS 本身的持续性防碰撞算法又无法跨进程提示新的域参与者其 GUID 与旧的已被析构的域参与者一致，从而导致了新的域参与者与旧的域参与者 GUID 相同：

表 5 节点 A GUID2 情况

GUID2		
Host_id	App_id	Instance_id
【本机 IP 地址】	【进程 2 process ID】	0

由于进程 2 的 process ID= 进程 1 process ID，导

致 GUID1= GUID2。

又由于 DDS 对传输服务质量可靠性的配置，DDS 两个域参与者完成了握手，建立了发布/订阅关系后，必须能够容忍短时间的（默认阈值为 30 秒）DDS 互相之间心跳 heartbeat 数据包的丢失，从而避免因偶发的网络掉线或网络环境不良导致的 DDS 域参与者互相之间不能正确保持发布/订阅，而 DDS 的关系数据库中的通讯伙伴列表（initial peers tables）中区分对端不同的域参与者的唯一依据是 GUID，相同的 GUID 将导致对端认为这是“原来的”域参与者实体，因此节点 B 域参与者依据 DDS 发现算法认为已经与“A”完成握手并建立发布/订阅关系，无需回应后来的来自“A”的 data(p) 握手包，导致一个发布/订阅关系的异常状态：节点 B 域参与者与节点 A 新建的域参与者 B 状态不对称——一个企图通信，一个企图握手，从而导致通信失败，登录数据无法被正确发送到对端上，如图 4 所示。

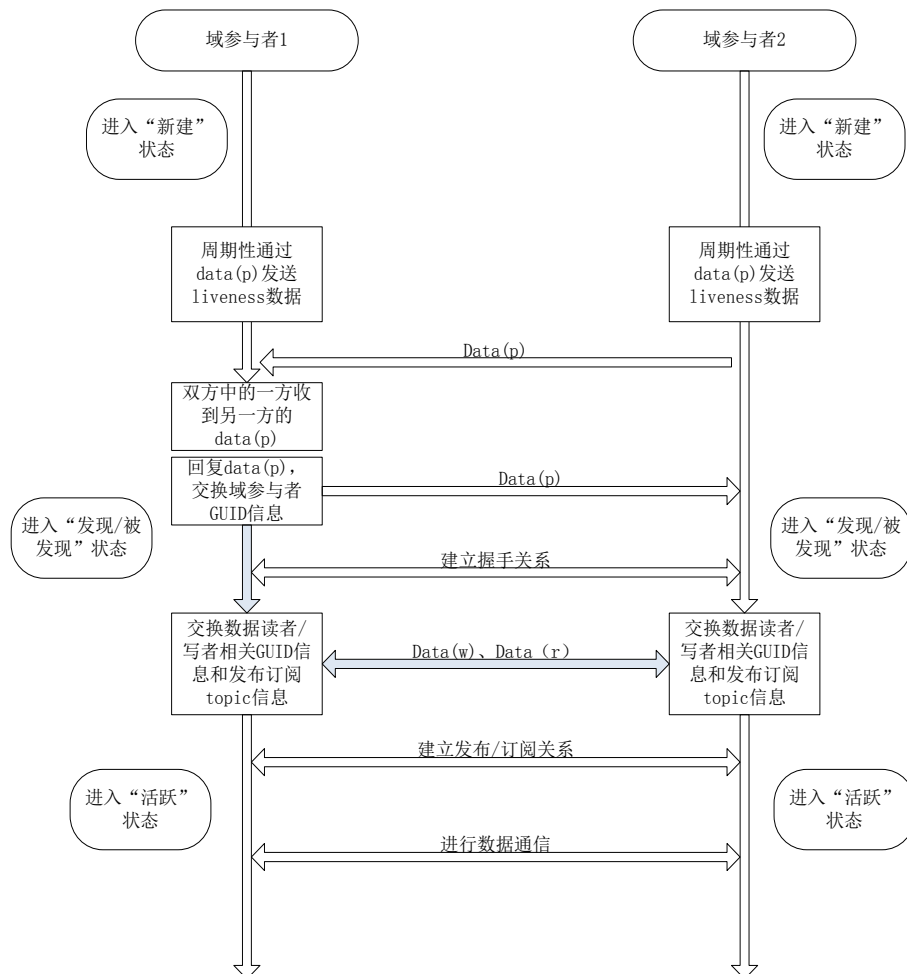


图 3 正常情况下 DDS 域参与者发现机制

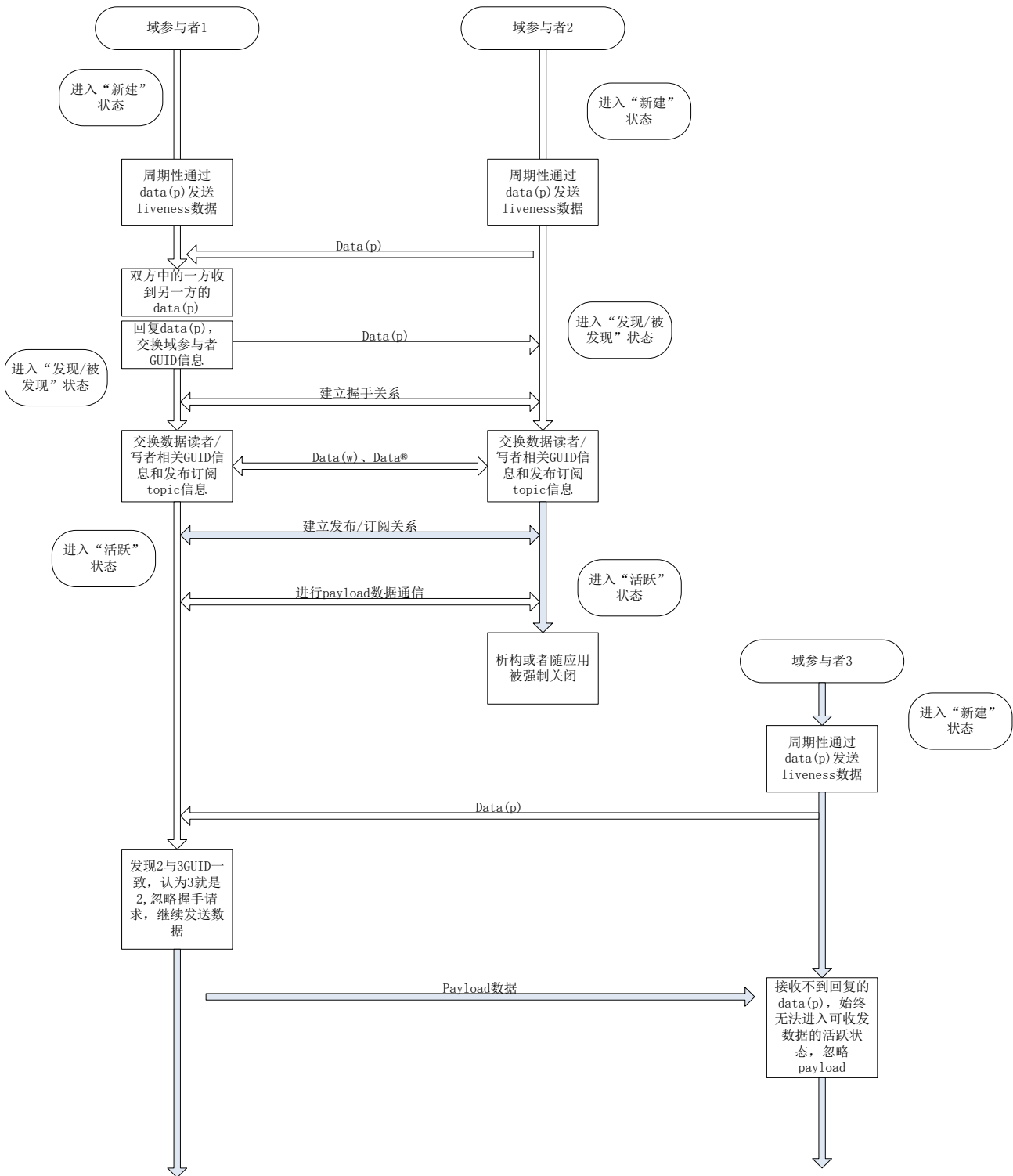


图 4 故障状态下的 DDS 域参与者发现机制

根据上述分析, 依据 DDS 自动发现机制原理, 通过技术手段修改了中间件, 通过在每次域参与者创建时自主写入 `app_id`, 并保证每次 `app_id` 与前值不同, 从而确保每一个新创建的域参与者的 GUID 均与前值不同。

完成修改后, 进行多次测试, 发现故障消失, 双端可正常发现彼此并进行通信, 问题解决。

4.2 通信配置问题

某开发项目的相关 DDS 中间件出现以下问题: 同一网络中, 两个端点互相之间无法发现对方, 但可以进行以太网通信, 即网络信道完好。

对问题进行分析, 首先进行了本端点自发自收测试, 发现受测 DDS 中间件进行自发自收自循环测试时运行正常, 即应用和中间件基本功能完好, 之

后使用相关工具进行 QoS 双端兼容性测试,发现两端的 QoS 配置符合兼容性要求。

考虑是否因通信配置导致自动发现无法完成,通过修改配置的方式,让 DDS 中间件改变运行模式,直接尝试向固定网络地址单播发送 Participant-data、publicationDATA 和 subscribeDATA,发现这种情况下双端可以完成互相发现,又通过抓包工具发现,在正常模式下,即进行 Participant-data 等数据的广播发送时,双端无法收到对端的广播数据。

因此推定网络对于 DDS 中间件发送广播发送数据存在问题,通过进一步排查发现,DDS 中间件发送广播包的生存时间值(Time To Live TTL)为 1,而交换机会让 TTL 减 1,因而导致广播数据到达对端时 TTL 为 0,因而被网卡丢弃,从而导致双端无法互相发现。

通过扩展 QoS 对 TTL 进行配置,将数值增大,完成修改后,发现故障消失,双端可正常发现彼此并进行通信,问题解决。

5 结论

DDS 本身作为一种基于全局数据空间和发布订阅机制的新型数据分发服务协议,具有很多常规数据传输协议所无法比拟的优势,但是其协议的高度封装性和黑盒特性使得在航电网络环境下其因某些特殊原因出现某些严重故障的时候难以进行精确的定位和排除,因而对 DDS 在航电网络中通信行为进行详细研究就成了必要的工作。

根据以上对 DDS 发布订阅通信节点自动发现机制研究,在工程实践过程中发现和排除了故障,另外,根据相关研究,提出了以下关于 DDS 应用程序和中间件的设计开发准则:

(1) 根据 DDS 自动发现机制的机理,要注意 GUID 的管理,尤其是其唯一性需要确保,由于发布订阅的发现机制需要确保每一个实体在的 GUID 在全局数据空间中的唯一性,而该唯一性由 GUID 表征,因而 GUID 的唯一性需要在 DDS 应用或者中间件开发过程中考虑到,除了空间上的唯一性,也要保证时间唯一性,即任何时刻,一个全局数据空间里不得有两个不同实体具有相同的 GUID,另外,为了便于查找和归纳,GUID 可以根据节点特性制定,

除了 IP 地址外,主机名和配置项编号也可以被用作 GUID 的一部分;

(2) 使用 DDS 时需要注意满足 DDS 发布订阅节点自动发现机制的其他特性,确保自动发现机制可部署,例如使用组播方式进行自动发现,则需要注意网络环境是否具备组播能力,以及对端节点是否能够正确接收组播消息;

(3) 在通信配置方面,需要注意确保通信双端兼容性,这包括数据结构一致性、Qos 双端兼容性和端口等通信资源的可用性,双端兼容并不强求通信两端的各种配置完全一样,但必须符合 DDS 协议规则。另外,DDS 所需要占用的一些有限的排他性的资源,如网络端口、内存、CPU 时间片等,需要结合工程实际为 DDS 相关应用和中间件留出所需余量,并进行统一管理。

当能够掌握航电网络中 DDS 的自动发现机制、发布订阅、双端状态一致性和数据传输等的具体原理后,即可利用原理对实际出现的一些深层故障进行分析并将之排除。而更进一步地,通过对 DDS 发布订阅通信节点自动发现机制的研究,不仅仅能够发现并排除故障,更重要的是能够总结并提出设计准则,依据准则指导改进 DDS 的相关应用和中间件设计,从而避免此类问题和故障的出现,切实提高相关工程应用的开发效率和质量。

参考文献

- [1]Data distribution service(DDS)[EB/OL][2-10-04-10].<http://www.omg.org/spec/DDS/1.4,2015.04.10>.
- [2] 邹歌,刘云飞.实时数据分发服务的自动发现技术[J].计算机技术与发展,2017,27(1):25-29.
- [3]The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol specification[EB/OL][2-008-06-13].<http://www.omg.org/spec/DDS/2.1/PDF,2008.06.13>.
- [4] 张琚,尹逊和.基于 DDS 的数据分发中间件的升级设计[J].北京交通大学学报,2011,35(5):31-37.
- [5]OMG organization. DDS Qos and Features[EB/OL][2017-09-22].<http://www.OMG.com,2009/2017.9.22>.
- [6]RTI core libraries and utilities users manual[EB/OL].<http://support.rti.com,2007>.