ICS 43.020
CCS T 40

团　　体　　标　　准

T/CSAE xx－20xx

智能网联汽车用数据分发服务（DDS）测试方法

Test methods of data distribution service (DDS) for intelligent and connected vehicles

（报批稿）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

20xx-xx-xx 发布　　20xx-xx-xx 实施

中国汽车工程学会 发布

目　　次

前　　言 IV
引　　言 V
1　范围 1
2　规范性引用文件 1
3　术语和定义 1
4　缩略语 2
5　RTPS 协议一致性测试 3
5.1　RTPS 消息格式测试 3
5.1.1　RTPS 消息头部 3
5.1.2　HeaderExtension 子消息 3
5.1.3　AckNack 子消息 4
5.1.4　Data 子消息 5
5.1.5　DataFrag 子消息 6
5.1.6　Gap 子消息 8
5.1.7　Heartbeat 子消息 9
5.1.8　HeartbeatFrag 子消息 10
5.1.9　InfoDestination 子消息 11
5.1.10　InfoReply 子消息 12
5.1.11　InfoSource 子消息 12
5.1.12　InfoTimestamp 子消息 13
5.1.13　NackFrag 子消息 14
5.1.14　Pad 子消息 15
5.2　RTPS 读写操作测试 16
5.2.1　尽力无状态写操作 16
5.2.2　可靠无状态写操作 17
5.2.3　尽力有状态写操作 22
5.2.4　可靠有状态写操作 23
5.2.5　尽力无状态读操作 28
5.2.6　尽力有状态读操作 29
5.2.7　可靠有状态读操作 29
5.3　发现协议建立连接操作测试 32
5.3.1　通知周期 32
5.3.2　租期 33

5.3.3 组播地址列表 34
5.3.4 单播地址列表 35
5.3.5 作为 Publisher 建立连接 36
5.3.6 作为 Subscriber 建立连接 38
6 协议功能测试 40
6.1 RTPS 读写功能测试 40
6.1.1 尽力无状态写操作 40
6.1.2 尽力有状态写操作 42
6.1.3 可靠无状态写操作 43
6.1.4 可靠有状态写操作 45
6.1.5 尽力无状态读操作 47
6.1.6 尽力有状态读操作 50
6.1.7 可靠有状态读操作 53
6.2 发现协议建立连接操作能测试 55
6.2.1 作为 Publisher 先进入网络 55
6.2.2 作为 Publisher 后进入网络 56
6.2.3 作为 Subscriber 先进入网络 57
6.2.4 作为 Subscriber 后进入网络 57
6.2.5 作为 Publisher 离开网络 58
6.2.6 作为 Subscriber 离开网络 59
6.3 QoS 功能测试 60
6.3.1 QoS DURABILITY 60
6.3.2 QoS HISTORY 63
6.3.3 QoS RESOURCE_LIMITS 65
6.3.4 QoS LIVELINESS 67
6.3.5 QoS PARTITION 70
6.3.6 QoS DEADLINE 72
7 安全功能测试 74
7.1 认证功能测试 74
7.1.1 作为发布者的认证功能 74
7.1.2 作为订阅者的认证功能 75
7.2 访问控制功能测试 76
7.2.1 合法访问者 76
7.2.2 非法访问者 77
7.3 加密功能测试 78
7.3.1 作为发送者的加密功能 78
7.3.2 作为接收者的解密功能 78
7.4 日志功能测试 79

7.4.1 测试目标 79
7.4.2 测试概要 79
7.4.3 适用条件 79
7.4.4 前置条件 79
7.4.5 测试环境 79
7.4.6 测试步骤 80
7.4.7 通过条件 80
7.5 数据标签功能测试 80
7.5.1 测试目标 80
7.5.2 测试概要 80
7.5.3 适用条件 80
7.5.4 前置条件 80
7.5.5 测试环境 80
7.5.6 测试步骤 80
7.5.7 通过条件 80
附录 A （规范性） 测试环境和测试前置条件 81
A.1 测试环境 81
A.2 测试前置条件 81
附录 B （规范性） RTPS 协议一致性测试参数 82

前　言

本文件按照GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国智能网联汽车产业创新联盟提出。

补充归口单位信息

本文件起草单位：中国信息通信研究院、长城汽车股份有限公司、国汽（北京）智能网联汽车研究院有限公司、吉利汽车研究院（宁波）有限公司、中汽创智科技有限公司、中国第一汽车股份有限公司研发总院、广州汽车集团股份有限公司汽车工程研究院、合众新能源汽车股份有限公司、北京汽车研究总院有限公司、北京车和家信息技术有限公司、北京万集科技股份有限公司、中兴通讯股份有限公司、北京华玉通软科技有限公司、上海映驰科技有限公司、惠州市德赛西威汽车电子股份有限公司、东风汽车集团有限公司研发总院。

本文件主要起草人：何巍、李巍、司远、李万里、常伟、尹鸿苇、檀庭跃、周云安、夏晶晶、薛信钊、李长龙、孔祥明、汤利顺、刘晓东、何烈焰、黄光健、张少宇、苏冲、王彬、李彬、栗相楠、彭华元、傅建雄、马冰、陈晓、李玉鹏、毕晓鹏、孙大力、赵建洪、王志杰、叶政、陈巧燕、吴小龙、谢瑛。

引　言

数据分发服务（Data Distribution Service），英文简称 DDS，是一种分布式实时通信中间件技术规范。DDS 采用发布/订阅体系架构，以数据为中心，提供丰富的服务质量策略和信息安全特性，能保障数据实时、高效、灵活和安全地分发，可满足各种分布式实时通信应用的需求。

DDS 最早应用在国防工业，目前已经广泛应用于国防、民航、工业控制等许多领域。在汽车领域，随着车辆对智能化和网联化需求的日益增长，对车载电子设备之间及车辆与外部设备之间进行实时数据传输的需求日益迫切，DDS 的技术特点使其成为开发先进车载应用的一种有效的中间件技术方案。

由于 DDS 协议的复杂性，对采用了 DDS 技术的车载设备和系统的验收与评价是一件困难和复杂的工作。本文件依据 DDS 的技术要求制定对应的 DDS 测试方法，为相关测试工作提供依据。

智能网联汽车用数据分发服务（DDS）测试方法

1 范围

本文件规定了智能网联汽车采用 DDS 中间件的设备或系统的测试方法。

本文件适用于对采用了 DDS 中间件的车载电子控制单元与相关系统的测试。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

OMG DDS v1.4	Data Distribution Service (DDS)
OMG DDSI-RTPS v2.5	The Real-time Publish-Subscribe Protocol DDS Interoperability Wire Protocol (DDSI-RTPS) Specification
OMG DDS-Security v1.1	DDS Security

3 术语和定义

下列术语和定义适用于本文件。

3.1

发布者 publisher

DDS 标准中定义的发送数据的对象。

3.2

订阅者 subscriber

DDS 标准中定义的接收数据的对象。

3.3

节点 entity

RTPS 协议中定义的基本功能单元。

3.4

参与者 participant

RTPS 协议中定义的属于同一个域（Domain）的节点。

3.5

端点 endpoint

RTPS 协议中定义的发送数据或接收数据的节点。

3.6

写入者 writer

RTPS 协议中定义的发送数据的端点。

3.7

读取者 reader

RTPS 协议中定义的接收数据的端点。

3.8

被测设备 device under test

被测试对象的物理设备。

3.9

链路伙伴 link partner

实现了 DDS 功能，在测试中与被测设备建立数据传输链路，配合测试仪表进行测试的物理设备。

4 缩略语

下列缩略语适用于本文件。

API: 应用编程接口（Application Programming Interface）
DCPS: 以数据为中心的发布订阅（Data Centric Publish-Subscribe）
DDS: 数据分发服务（Data Distribution Service）
DUT: 被测设备（Device Under Test）
ECU: 电子控制单元（Electronic Control Unit）
LP: 链路伙伴（Link Partner）
OMG: 对象管理组（Object Management Group）
QoS: 服务质量（Quality of Service）
RTPS: 实时发布订阅协议（real time publish subscribe protocol）
SEDP: 简单端点发现协议（Simple Endpoint Discovery Protocol）
SPDP: 简单参与者发现协议（Simple Participant Discovery Protocol）

5 RTPS 协议一致性测试

5.1 RTPS 消息格式测试

5.1.1 RTPS 消息头部

5.1.1.1 测试目标

验证 RTPS 消息头部字段的格式。

5.1.1.2 测试概要

验证 RTPS 消息头部各字段的格式是否符合标准要求。

5.1.1.3 适用条件

所有支持 DDS 协议的设备。

5.1.1.4 前置条件

符合附录 A.2 测试前置条件。

5.1.1.5 测试环境

符合附录 A.1 测试环境 1。

5.1.1.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；

b) 设置测试仪表开始捕捉 DUT 发送的消息；

c) 启动 DUT 和测试仪表，待其建立连接；

d) 测试仪表从收到的消息序列中提取 RTPS 消息头部包含的字段：protocol，version，vendorId，guidPrefix。

5.1.1.7 通过条件

按 5.1.1.6 中 d)：应满足如下字段值要求：

protocol = <protocol>
version = <version>
vendorId = <vendorId>
guidPrefix = <guidPrefix>

5.1.2 HeaderExtension 子消息

5.1.2.1 测试目标

验证 HeaderExtension 类型子消息格式。

5.1.2.2 测试概要

验证 DUT 发送的 HeaderExtension 类型子消息的格式是否正确。

5.1.2.3 适用条件

DUT 支持 RTPS v2.5 及以上版本的设备。

5.1.2.4 前置条件

符合附录 A.2 测试前置条件。

5.1.2.5 测试环境

符合附录 A.1 测试环境 1。

5.1.2.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 HeaderExtension 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 HeaderExtension 子消息；
f) 从 HeaderExtension 子消息中提取子消息头部；
g) 从 HeaderExtension 子消息中提取子消息内容。

5.1.2.7 通过条件

按 5.1.2.6 中 e)：成功提取 HeaderExtension 子消息。

按 5.1.2.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x00

flags：

ParamatersFlag = <ParamatersFlag>
ChecksumFlag_C1 = <ChecksumFlag_C1>
ChecksumFlag_C2 = <ChecksumFlag_C2>
Wextension8Flag = <Wextension8Flag>
UExtension4Flag = <UExtension4Flag>
TimestampFlag = <TimestampFlag>
LengthFlag = <LengthFlag>
EndiannessFlag = <EndiannessFlag>

submessageLength = <submessageLength>

按 5.1.2.6 中 g)：子消息内容字段值应满足要求：

messageLength = <messageLength>，（仅当 LengthFlag =1）
rtpsSendTimestamp = <rtpsSendTimestamp>，（仅当 TimestampFlag =1）
uExtension4 = <uExtension4>，（仅当 UExtension4Flag =1）
wExtension8 = <wExtension8>，（仅当 Wextension8Flag =1）
messageChecksum = <messageChecksum>,（仅当<ChecksumFlag_C1,ChecksumFlag_C2> !=00）
parameters = <parameters>，（仅当 ParamatersFlag !=0）

5.1.3 AckNack 子消息

5.1.3.1 测试目标

验证 AckNack 类型子消息格式。

5.1.3.2 测试概要

验证 DUT 发送的 AckNack 类型子消息的格式是否正确。

5.1.3.3 适用条件

所有支持 DDS 协议的设备。

5.1.3.4 前置条件

符合附录 A.2 测试前置条件。

5.1.3.5 测试环境

符合附录 A.1 测试环境 1。

5.1.3.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) 测试仪表从收到的消息序列中提取 AckNack 子消息；
e) 从 AckNack 子消息中提取子消息头部；
f) 从 AckNack 子消息中提取子消息内容。

5.1.3.7 通过条件

按 5.1.3.6 中 d)：成功提取 AckNack 子消息。

按 5.1.3.6 中 e)：子消息头部字段值应满足要求：

submessageId = 0x06
flags：
        FinalFlag = <FinalFlag>
        EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.3.6 中 f)：子消息内容字段值应满足要求：

readerId = <readerId>
writerId = <writerId>
readerSNState = <readerSNState>
count = <count>

5.1.4 Data 子消息

5.1.4.1 测试目标

验证 Data 类型子消息格式。

5.1.4.2 测试概要

验证 DUT 发送的 Data 类型子消息的格式是否正确。

5.1.4.3 适用条件

所有支持 DDS 协议的设备。

5.1.4.4 前置条件

符合附录 A.2 测试前置条件。

5.1.4.5 测试环境

符合附录 A.1 测试环境 1。

5.1.4.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 Data 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 Data 子消息；
f) 从 Data 子消息中提取子消息头部；
g) 从 Data 子消息中提取子消息内容。

5.1.4.7 通过条件

按 5.1.4.6 中 e)：成功提取 Data 子消息。

按 5.1.4.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x15
flags：
NonStandardPayloadFlag = <NonStandardPayloadFlag>
KeyFlag = <KeyFlag>
DataFlag = <DataFlag>
InlineQosFlag = <InlineQosFlag>
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.4.6 中 g)：子消息内容字段值应满足要求：

extraFlags = 0x0000
octetsToInlineQos = <octetsToInlineQos>
readerId = <readerId>
writerId = <writerId>
writerSN = <writerSN>
inlineQos = <inlineQos>,（仅当 InlineQosFlag=1）
serializedPayload = <serializedPayload>,（仅当 KeyFlag=1 或 DataFlag=1）

5.1.5 DataFrag 子消息

5.1.5.1 测试目标

验证 DataFrag 类型子消息格式。

5.1.5.2 测试概要

验证 DUT 发送的 DataFrag 类型子消息的格式是否正确。

5.1.5.3 适用条件

所有支持 DDS 协议的设备。

5.1.5.4 前置条件

符合附录 A.2 测试前置条件。

5.1.5.5 测试环境

符合附录 A.1 测试环境 1。

5.1.5.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 DataFrag 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 DataFrag 子消息；
f) 从 DataFrag 子消息中提取子消息头部；
g) 从 DataFrag 子消息中提取子消息内容。

5.1.5.7 通过条件

按 5.1.5.6 中 e)：成功提取 DataFrag 子消息。

按 5.1.5.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x16
flags：
NonStandardPayloadFlag = <NonStandardPayloadFlag>
KeyFlag = <KeyFlag>
InlineQosFlag = <InlineQosFlag>
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.5.6 中 g)：子消息内容字段值应满足要求：

extraFlags = 0x0000;
octetsToInlineQos = <octetsToInlineQos>
readerId = <readerId>
writerId = <writerId>
writerSN = <writerSN>
fragmentStartingNum = <fragmentStartingNum>
fragmentsInSubmessage = <fragmentsInSubmessage>
fragmentSize = <fragmentSize>
dataSize = <dataSize>
inlineQos = <inlineQos>
serializedPayload = <serializedPayload>,（仅当 InlineQosFlag=1）

5.1.6 Gap 子消息

5.1.6.1 测试目标

验证 Gap 类型子消息格式。

5.1.6.2 测试概要

验证 DUT 发送的 Gap 类型子消息的格式是否正确。

5.1.6.3 适用条件

所有支持 DDS 协议的设备。

5.1.6.4 前置条件

符合附录 A.2 测试前置条件。

5.1.6.5 测试环境

符合附录 A.1 测试环境 1。

5.1.6.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，设置测试仪表开始捕捉消息；
b) DUT 用如下 QoS 策略创建 DataWriter：
   reliability.kind = RELIABLE
   durability.kind = TRANSIENT_LOCAL
   history.kind = KEEP_ALL
c) 启动 DUT，发送 5 个 DATA 子消息；
d) 测试仪表用如下 QoS 策略创建与 DUT 相同 Topic 的 DataReader；
   reliability.kind = RELIABLE
   durability.kind = VOLATILE
   history.kind = KEEP_ALL
e) 待测试仪表与 DUT 建立连接；
f) 测试仪表从收到的消息序列中提取 DUT 发送的 Gap 子消息；
g) 从 Gap 子消息中提取子消息头部；
h) 从 Gap 子消息中提取子消息内容。

5.1.6.7 通过条件

按 5.1.6.6 中 f)：成功提取 Gap 子消息。
按 5.1.6.6 中 g)：子消息头部字段值应满足要求：
   submessageId = 0x08
   flags：
      FilteredCountFlag = <FilteredCountFlag>
      GroupInfoFlag = <GroupInfoFlag>
      EndiannessFlag = <EndiannessFlag>
   submessageLength = <submessageLength>
按 5.1.6.6 中 h)：子消息内容字段值应满足要求：

readerId = <readerId>
writerId = <writerId>
gapStart = 1
gapList：
gapList.bitmapBase = 6
gapList.numBits = 0
gapStartGSN = <gapStartGSN>，（仅当 GroupInfoFlag=1）
gapEndGSN = <gapEndGSN>，（仅当 GroupInfoFlag=1）
filteredCount = <filteredCount>，（仅当 FilteredCountFlag=1）

5.1.7 Heartbeat 子消息

5.1.7.1 测试目标

验证 Heartbeat 类型子消息格式。

5.1.7.2 测试概要

验证 DUT 发送的 Heartbeat 类型子消息的格式是否正确。

5.1.7.3 适用条件

所有支持 DDS 协议的设备。

5.1.7.4 前置条件

符合附录 A.2 测试前置条件。

5.1.7.5 测试环境

符合附录 A.1 测试环境 1。

5.1.7.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) 测试仪表从收到的消息序列中提取 Heartbeat 子消息；
e) 从 Heartbeat 子消息中提取子消息头部；
f) 从 Heartbeat 子消息中提取子消息内容。

5.1.7.7 通过条件

按 5.1.7.6 中 d)：成功提取 Heartbeat 子消息。
按 5.1.7.6 中 e)：子消息头部字段值应满足要求：

submessageId = 0x07
flags：
GroupInfoFlag = <GroupInfoFlag>
LivelinessFlag = <LivelinessFlag>
FinalFlag = <FinalFlag>
EndiannessFlag = <EndiannessFlag>

submessageLength = <submessageLength>

按 5.1.7.6 中 f)：子消息内容字段值应满足要求：

readerId = <readerId>

writerId = <writerId>

firstSN = <firstSN>

lastSN = <lastSN>

count = <count>

currentGSN = <currentGSN>，（仅当 GroupInfoFlag=1）

firstGSN = <firstGSN>，（仅当 GroupInfoFlag=1）

lastGSN = <lastGSN>，（仅当 GroupInfoFlag=1）

writerSet = <writerSet>，（仅当 GroupInfoFlag=1）

secureWriterSet = <secureWriterSet>，（仅当 GroupInfoFlag=1）

5.1.8 HeartbeatFrag 子消息

5.1.8.1 测试目标

验证 HeartbeatFrag 类型子消息格式。

5.1.8.2 测试概要

验证 DUT 发送的 HeartbeatFrag 类型子消息的格式是否正确。

5.1.8.3 适用条件

所有支持 DDS 协议的设备。

5.1.8.4 前置条件

符合附录 A.2 测试前置条件。

5.1.8.5测试环境

符合附录 A.1 测试环境 1。

5.1.8.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；

b) 设置测试仪表开始捕捉 DUT 发送的消息；

c) 启动 DUT 和测试仪表，待其建立连接；

d) DUT 发送包含 HeartbeatFrag 子消息的 RTPS 消息；

e) 测试仪表从收到的消息序列中提取 HeartbeatFrag 子消息；

f) 从 HeartbeatFrag 子消息中提取子消息头部；

g) 从 HeartbeatFrag 子消息中提取子消息内容。

5.1.8.7 通过条件

按 5.1.8.6 中 e)：成功提取 HeartbeatFrag 子消息。

按 5.1.8.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x13

flags：

EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.8.6 中 g)：子消息内容字段值应满足要求：

readerId = <readerId>
writerId = <writerId>
writerSN = <writerSN>
lastFragmentNum = <lastFragmentNum>
count = <count>

5.1.9 InfoDestination 子消息

5.1.9.1 测试目标

验证 InfoDestination 类型子消息格式。

5.1.9.2 测试概要

验证 DUT 发送的 InfoDestination 类型子消息的格式是否正确。

5.1.9.3 适用条件

所有支持 DDS 协议的设备。

5.1.9.4 前置条件

符合附录 A.2 测试前置条件。

5.1.9.5 测试环境

符合附录 A.1 测试环境 1。

5.1.9.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 InfoDestination 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 InfoDestination 子消息；
f) 从 InfoDestination 子消息中提取子消息头部；
g) 从 InfoDestination 子消息中提取子消息内容。

5.1.9.7 通过条件

按 5.1.9.6 中 e)：成功提取 InfoDestination 子消息。

按 5.1.9.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x0e
flags：
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.9.6 中 g)：子消息内容字段值应满足要求：

guidPrefix = <GUIDPREFIX>

5.1.10 InfoReply 子消息

5.1.10.1 测试目标

验证 InfoReply 类型子消息格式。

5.1.10.2 测试概要

验证 DUT 发送的 InfoReply 类型子消息的格式是否正确。

5.1.10.3 适用条件

所有支持 DDS 协议的设备。

5.1.10.4 前置条件

符合附录 A.2 测试前置条件。

5.1.10.5 测试环境

符合附录 A.1 测试环境 1。

5.1.10.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 InfoReply 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 InfoReply 子消息；
f) 从 InfoReply 子消息中提取子消息头部；
g) 从 InfoReply 子消息中提取子消息内容。

5.1.10.7 通过条件

按 5.1.10.6 中 e)：成功提取 InfoReply 子消息。
按 5.1.10.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x0f
flags：
MulticastFlag = <MulticastFlag>
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.10.6 中 g)：子消息内容字段值应满足要求：

unicastLocatorList = <unicastLocatorList>
multicastLocatorList = <multicastLocatorList>，（仅当 MulticastFlag=1）

5.1.11 InfoSource 子消息

5.1.11.1 测试目标

验证 InfoSource 类型子消息格式。

5.1.11.2 测试概要

验证 DUT 发送的 InfoSource 类型子消息的格式是否正确。

5.1.11.3 适用条件

所有支持 DDS 协议的设备。

5.1.11.4 前置条件

符合附录 A.2 测试前置条件。

5.1.11.5 测试环境

符合附录 A.1 测试环境 1。

5.1.11.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 InfoSource 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 InfoSource 子消息；
f) 从 InfoSource 子消息中提取子消息头部；
g) 从 InfoSource 子消息中提取子消息内容。

5.1.11.7 通过条件

按 5.1.11.6 中 e)：成功提取 InfoSource 子消息。

按 5.1.11.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x0c
flags：
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.11.6 中 g)：子消息内容字段值应满足要求：

protocolVersion = <PROTOCOLVERSION>
vendorId = <VENDORID>
guidPrefix = <GUIDPREFIX>

5.1.12 InfoTimestamp 子消息

5.1.12.1 测试目标

验证 InfoTimestamp 类型子消息格式。

5.1.12.2 测试概要

验证 DUT 发送的 InfoTimestamp 类型子消息的格式是否正确。

5.1.12.3 适用条件

所有支持 DDS 协议的设备。

5.1.12.4 前置条件

符合附录 A.2 测试前置条件。

5.1.12.5 测试环境

符合附录 A.1 测试环境 1。

5.1.12.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 InfoTimestamp 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 InfoTimestamp 子消息；
f) 从 InfoTimestamp 子消息中提取子消息头部；
g) 从 InfoTimestamp 子消息中提取子消息内容。

5.1.12.7 通过条件

按 5.1.12.6 中 e)：成功提取 InfoTimestamp 子消息。
按 5.1.12.6 中 f)：子消息头部字段值应满足要求：
submessageId = 0x09
flags：
InvalidateFlag = <InvalidateFlag>
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>
按 5.1.12.6 中 g)：子消息内容字段值应满足要求：
timestamp = <timestamp>，（仅当 InvalidateFlag=0）

5.1.13 NackFrag 子消息

5.1.13.1 测试目标

验证 NackFrag 类型子消息格式。

5.1.13.2 测试概要

验证 DUT 发送的 NackFrag 类型子消息的格式是否正确。

5.1.13.3 适用条件

所有支持 DDS 协议的设备。

5.1.13.4 前置条件

符合附录 A.2 测试前置条件。

5.1.13.5 测试环境

符合附录 A.1 测试环境 1。

5.1.13.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 NackFrag 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 NackFrag 子消息；
f) 从 NackFrag 子消息中提取子消息头部；
g) 从 NackFrag 子消息中提取子消息内容。

5.1.13.7 通过条件

按 5.1.13.6 中 e)：成功提取 NackFrag 子消息。

按 5.1.13.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x12
flags：
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

按 5.1.13.6 中 g)：子消息内容字段值应满足要求：

readerId = <readerId>
writerId = <writerId>
writerSN = <writerSN>
fragmentNumberState = <fragmentNumberState>
count = <count>

5.1.14 Pad 子消息

5.1.14.1 测试目标

验证 Pad 类型子消息格式。

5.1.14.2 测试概要

验证 DUT 发送的 Pad 类型子消息的格式是否正确。

5.1.14.3 适用条件

所有支持 DDS 协议的设备。

5.1.14.4 前置条件

符合附录 A.2 测试前置条件。

5.1.14.5 测试环境

符合附录 A.1 测试环境 1。

5.1.14.6 测试步骤

该测试步骤如下：

a) DUT 创建一个 RTPS Writer，测试仪表创建与 DUT 相同 Topic 的 RTPS Reader；

b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT 和测试仪表，待其建立连接；
d) DUT 发送包含 Pad 子消息的 RTPS 消息；
e) 测试仪表从收到的消息序列中提取 Pad 子消息；
f) 从 Pad 子消息中提取子消息头部；
g) 从 Pad 子消息中提取子消息内容。

5.1.14.7 通过条件

按 5.1.14.6 中 e)：成功提取 Pad 子消息。
按 5.1.14.6 中 f)：子消息头部字段值应满足要求：

submessageId = 0x01
flags：
EndiannessFlag = <EndiannessFlag>
submessageLength = <submessageLength>

5.2 RTPS 读写操作测试

5.2.1 尽力无状态写操作

5.2.1.1 发送 DATA 子消息

5.2.1.1.1 测试目标

验证 DUT 尽力无状态写操作发送 DATA 子消息的操作是否正确。

5.2.1.1.2 测试概要

验证 DUT 尽力无状态写操作发送 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.1.1.3 适用条件

DUT 支持尽力无状态写操作。

5.2.1.1.4 前置条件

符合附录 B 前置条件 1。

5.2.1.1.5 测试环境

符合附录 A.1 测试环境 1。

5.2.1.1.6 测试步骤

该测试步骤如下：
a) DUT 创建尽力无状态 RTPS Writer；
b) 测试仪表创建尽力无状态 RTPS Reader，Topic 与 DUT 相同；
c) 设置测试仪表开始捕捉 DUT 发送的消息；
d) 启动 DUT 和测试仪表，待其建立连接；
e) DUT 发送至少 10 个 DATA 子消息；
f) 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.1.1.7 通过条件

按 5.2.1.1.6 中 f)：成功收到 DUT 发送的 DATA 子消息。

按 5.2.1.1.6 中 f)：测试仪表收到的 DATA 子消息与步骤 e）DUT 发送的 DATA 子消息的数量和内容一致。

5.2.1.2 发送 GAP 子消息

5.2.1.2.1 测试目标

验证 DUT 尽力无状态写操作发送 GAP 子消息的操作是否正确。

5.2.1.2.2 测试概要

验证 DUT 尽力无状态写操作发送 GAP 子消息的过程是否符合标准中定义的状态机。

5.2.1.2.3 适用条件

DUT 支持尽力无状态写操作。

5.2.1.2.4 前置条件

符合附录 A.2 测试前置条件。

5.2.1.2.5 测试环境

符合附录 A.1 测试环境 1。

5.2.1.2.6 测试步骤

该测试步骤如下：

a) DUT 创建尽力无状态 RTPS Writer；
b) 测试仪表用 DUT 的 Topic 创建一个 Content-filtered Topics，实现两个 filter：
   filter A：过滤掉序列号 3 至 5 的消息；
   filter B：只保留序列号 3 至 5 的消息；
c) 测试仪表用以上 filter 分别创建一个尽力无状态 RTPS Reader；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动测试仪表；
f) 启动 DUT，待其与两个 RTPS Reader 均建立连接；
g) DUT 连续发送 10 个 DATA 子消息；
h) 测试仪表从收到的消息序列中提取 Gap 子消息。

5.2.1.2.7 通过条件

按 5.2.1.2.6 中 h)：收到 GAP 子消息。

5.2.2 可靠无状态写操作

5.2.2.1 发送 DATA 子消息

5.2.2.1.1 测试目标

验证 DUT 可靠无状态写操作发送 DATA 子消息的操作是否正确。

5.2.2.1.2 测试概要

验证 DUT 可靠无状态写操作发送 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.2.1.3 适用条件

DUT 支持可靠无状态写操作。

5.2.2.1.4 前置条件

符合附录 A.2 测试前置条件。

5.2.2.1.5 测试环境

符合附录 A.1 测试环境 1。

5.2.2.1.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠无状态 RTPS Writer；
b) 测试仪表创建可靠无状态 RTPS Reader，Topic 与 DUT 相同；
c) 设置测试仪表开始捕捉 DUT 发送的消息；
d) 启动 DUT 和测试仪表，待其建立连接；
e) DUT 发送至少 10 个 DATA 子消息；
f) 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.2.1.7 通过条件

按 5.2.2.1.6 中 f)：收到 DUT 发送的 HEARTBEAT 子消息。

按 5.2.2.1.6 中 f)：收到 DUT 发送的 DATA 子消息。

按 5.2.2.1.6 中 f)：测试仪表收到的 DATA 子消息与步骤 e) DUT 发送的 DATA 子消息的数量和内容一致。

5.2.2.2 发送 GAP 子消息

5.2.2.2.1 测试目标

验证 DUT 可靠无状态写操作发送 GAP 子消息的操作是否正确。

5.2.2.2.2 测试概要

验证 DUT 可靠无状态写操作发送 GAP 子消息的过程是否符合标准中定义的状态机。

5.2.2.2.3 适用条件

DUT 支持可靠无状态写操作。

5.2.2.2.4 前置条件

符合附录 A.2 测试前置条件。

5.2.2.2.5 测试环境

符合附录 A.1 测试环境 1。

5.2.2.2.6 测试步骤

该测试步骤如下：

a) 设置测试仪表开始捕捉消息；

b) DUT 用如下 QoS 策略创建 DataWriter：

reliability.kind = RELIABLE

durability.kind = TRANSIENT_LOCAL

history.kind = KEEP_ALL

c) 启动 DUT，发送 5 个 DATA 子消息；

d) 测试仪表用如下 QoS 策略创建与 DUT 相同 Topic 的 DataReader：

reliability.kind = RELIABLE

durability.kind = VOLATILE

history.kind = KEEP_ALL

e) 待测试仪表与 DUT 建立连接；

f) 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.2.2.7 通过条件

按 5.2.2.2.6 中 f)：测试仪表收到 GAP 子消息。

按 5.2.2.2.6 中 f)：GAP 子消息字段值应满足要求：

gapStart = 1

gapList：

gapList.bitmapBase = 6

gapList.numBits = 0

5.2.2.3 HEARTBEAT 子消息

5.2.2.3.1 测试目标

验证 DUT 可靠无状态写操作发送 HEARTBEAT 子消息的操作是否正确。

5.2.2.3.2 测试概要

验证 DUT 可靠无状态写操作发送 HEARTBEAT 子消息的过程是否符合标准中定义的状态机。

5.2.2.3.3 适用条件

DUT 支持可靠无状态写操作。

5.2.2.3.4 前置条件

符合附录 A.2 测试前置条件。

5.2.2.3.5 测试环境

符合附录 A.1 测试环境 1。

5.2.2.3.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠无状态 RTPS Writer，确认 Writer.pushMode=false;

b) 设置 DUT 的 Writer.heartbeatPeriod = 3s；

c) 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动 DUT 和测试仪表，待其建立连接；
f) 等待至少 30s；
g) 测试仪表记录并验证从 DUT 发送的消息；
h) 设置 DUT 的 Writer.heartbeatPeriod = 5s；
i) 设置测试仪表开始捕捉 DUT 发送的消息；
j) 重启 DUT 和测试仪表，待其建立连接；
k) 等待至少 30s；
l) 测试仪表记录并验证从 DUT 发送的消息。

5.2.2.3.7 通过条件

按 5.2.2.3.6 中 m)：测试仪表收到 DUT 发送的连续的 HEARTBEAT 子消息，发送周期 3s。
按 5.2.2.3.6 中 m):测试仪表收到 DUT 发送的 HEARTBEAT 子消息,HEARTBEAT.FinalFlag = 1(SET)。
按 5.2.2.3.6 中 l)：测试仪表收到 DUT 发送的连续的 HEARTBEAT 子消息，发送周期 5s。
按 5.2.2.3.6 中 l):测试仪表收到 DUT 发送的 HEARTBEAT 子消息,HEARTBEAT.FinalFlag = 1(SET)。

5.2.2.4 nackResponseDelay 的值

5.2.2.4.1 测试目标

验证 DUT 在可靠无状态写操作过程中 nackResponseDelay 的值。

5.2.2.4.2 测试概要

验证 DUT 在可靠无状态写操作过程中是否按 nackResponseDelay 设置的时间回复 DATA。

5.2.2.4.3 适用条件

DUT 支持可靠无状态写操作。

5.2.2.4.4 前置条件

符合附录 A.2 测试前置条件。

5.2.2.4.5 测试环境

符合附录 A.1 测试环境 1。

5.2.2.4.6 测试步骤

该测试步骤如下：
a) DUT 创建可靠无状态 RTPS Writer；
b) 设置 DUT 的 nackRsponseDelay = 3s；
c) 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动 DUT 和测试仪表，待其建立连接；
f) DUT 发送 10 个 DATA 子消息；
g) 测试仪表发送 ACKNACK 消息请求重发最后 1 个 DATA 子消息，并开始计时；
h) 测试仪表捕捉 DATA 子消息；
i) 测试仪表验证收到的 DATA 子消息的内容；

j） 测试仪表计算从步骤 g）到步骤 h）经过的时间 t1；
k） 设置 DUT 的 nackRsponseDelay = 5s；
l） 设置测试仪表开始捕捉 DUT 发送的消息；
m） 重新启动 DUT 和测试仪表，待其建立连接；
n） DUT 发送 10 个 DATA 子消息；
o） 测试仪表发送 ACKNACK 消息请求重发最后 1 个 DATA 子消息，并开始计时；
p） 测试仪表捕捉 DATA 子消息；
q） 测试仪表验证收到的 DATA 子消息的内容；
r） 测试仪表计算从步骤 o）到步骤 p）经过的时间 t2。

5.2.2.4.7 通过条件

按 5.2.2.4.6 中 h）：测试仪表收到 1 个 DATA 子消息。

按 5.2.2.4.6 中 i）：测试仪表收到的 DATA 子消息与步骤 f）DUT 发送的最后一个 DATA 子消息的内容一致。

按 5.2.2.4.6 中 j）：3s < t1 < 4s。

按 5.2.2.4.6 中 p）：测试仪表收到 1 个 DATA 子消息。

按 5.2.2.4.6 中 q）：测试仪表收到的 DATA 子消息与步骤 n）DUT 发送的最后一个 DATA 子消息的内容一致。

按 5.2.2.4.6 中 r）：5s < t2 < 6s。

5.2.2.5 数据重发操作

5.2.2.5.1 测试目标

验证 DUT 可靠无状态写操作的数据重发操作是否正确。

5.2.2.5.2 测试概要

验证 DUT 在可靠无状态写操作过程中是否可以重新发送 RTPS Reader 没有正确接收的 DATA 子消息。

5.2.2.5.3 适用条件

DUT 支持可靠无状态写操作。

5.2.2.5.4 前置条件

符合附录 A.2 测试前置条件。

5.2.2.5.5 测试环境

符合附录 A.1 测试环境 1。

5.2.2.5.6 测试步骤

该测试步骤如下：

a） DUT 创建可靠无状态 RTPS Writer；
b） 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
c） 设置测试仪表开始捕捉 DUT 发送的消息；
d） 启动 DUT 和测试仪表，待其建立连接；
e） DUT 发送 10 个 DATA 子消息；

f） 测试仪表发送 ACKNACK 消息请求重发前 5 个 DATA 子消息；
g） 测试仪表捕捉 DATA 子消息；
h） 测试仪表验证收到的 DATA 子消息的内容；
i） DUT 重新发送 10 个 DATA 子消息；
j） 测试仪表发送 ACKNACK 消息请求重发后 5 个 DATA 子消息；
k） 测试仪表捕捉 DATA 子消息；
l） 测试仪表验证收到的 DATA 子消息的内容。

5.2.2.5.7 通过条件

按 5.2.2.5.6 中 g）：测试仪表收到 5 个 DATA 子消息。

按 5.2.2.5.6 中 h）：测试仪表收到的 DATA 子消息与步骤 e）DUT 发送的前 5 个 DATA 子消息的内容一致。

按 5.2.2.5.6 中 k）：测试仪表收到 5 个 DATA 子消息。

按 5.2.2.5.6 中 l）：测试仪表收到的 DATA 子消息与步骤 i）DUT 发送的后 5 个 DATA 子消息的内容一致。

5.2.3 尽力有状态写操作

5.2.3.1 发送 DATA 子消息

5.2.3.1.1 测试目标

验证 DUT 尽力有状态写操作发送 DATA 子消息的操作是否正确。

5.2.3.1.2 测试概要

验证 DUT 尽力有状态写操作发送 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.3.1.3 适用条件

DUT 支持尽力有状态写操作。

5.2.3.1.4 前置条件

符合附录 A.2 测试前置条件。

5.2.3.1.5 测试环境

符合附录 A.1 测试环境 1。

5.2.3.1.6 测试步骤

该测试步骤如下：

a） DUT 创建尽力有状态 RTPS Writer；
b） 测试仪表创建尽力 RTPS Reader，Topic 与 DUT 相同；
c） 设置测试仪表开始捕捉 DUT 发送的消息；
d） 启动 DUT 和测试仪表，待其建立连接；
e） DUT 发送至少 10 个 DATA 子消息；
f） 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.3.1.7 通过条件

按 5.2.3.1.6 中 f）：成功收到 DUT 发送的 DATA 子消息。

按 5.2.3.1.6 中 f)：测试仪表收到的 DATA 子消息与步骤 e）DUT 发送的 DATA 子消息的数量和内容一致。

5.2.3.2 发送 GAP 子消息

5.2.3.2.1 测试目标

验证 DUT 尽力有状态写操作发送 GAP 子消息的操作是否正确。

5.2.3.2.2 测试概要

验证 DUT 尽力有状态写操作发送 GAP 子消息的过程是否符合标准中定义的状态机。

5.2.3.2.3 适用条件

DUT 支持尽力有状态写操作。

5.2.3.2.4 前置条件

符合附录 A.2 测试前置条件。

5.2.3.2.5 测试环境

符合附录 A.1 测试环境 1。

5.2.3.2.6 测试步骤

该测试步骤如下：

a) DUT 创建尽力有状态 RTPS Writer；
b) 测试仪表用 DUT 的 Topic 创建一个 Content-filtered Topics，实现两个 filter：
   filter A：过滤掉序列号 3 至 5 的消息；
   filter B：只保留序列号 3 至 5 的消息；
c) 测试仪表用以上 filter 分别创建一个尽力有状态 RTPS Reader；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动测试仪表；
f) 启动 DUT，待其与两个 RTPS Reader 均建立连接；
g) DUT 连续发送 10 个 DATA 子消息；
h) 测试仪表从收到的消息序列中提取 Gap 子消息。

5.2.3.2.7 通过条件

按 5.2.3.2.6 中 h)：收到 GAP 子消息。

5.2.4 可靠有状态写操作

5.2.4.1 发送 DATA 子消息

5.2.4.1.1 测试目标

验证 DUT 可靠有状态写操作发送 DATA 子消息的操作是否正确。

5.2.4.1.2 测试概要

验证 DUT 可靠有状态写操作发送 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.4.1.3 适用条件

DUT 支持可靠有状态写操作。

5.2.4.1.4 前置条件

符合附录 A.2 测试前置条件。

5.2.4.1.5 测试环境

符合附录 A.1 测试环境 1。

5.2.4.1.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Writer，确认 Writer.pushMode=false；
b) 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
c) 设置测试仪表开始捕捉 DUT 发送的消息；
d) 启动 DUT 和测试仪表，待其建立连接；
e) DUT 发送至少 10 个 DATA 子消息；
f) 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.4.1.7 通过条件

按 5.2.4.1.6 中 f)：收到 DUT 发送的 HEARTBEAT 子消息。

按 5.2.4.1.6 中 f)：成功收到 DUT 发送的 DATA 子消息。

按 5.2.4.1.6 中 f)：测试仪表收到的 DATA 子消息与步骤 e）DUT 发送的 DATA 子消息的数量和内容一致。

5.2.4.2 发送 GAP 子消息

5.2.4.2.1 测试目标

验证 DUT 可靠有状态写操作发送 GAP 子消息的操作是否正确。

5.2.4.2.2 测试概要

验证 DUT 可靠有状态写操作发送 GAP 子消息的过程是否符合标准中定义的状态机。

5.2.4.2.3 适用条件

DUT 支持可靠有状态写操作。

5.2.4.2.4 前置条件

符合附录 A.2 测试前置条件。

5.2.4.2.5 测试环境

符合附录 A.1 测试环境 1。

5.2.4.2.6 测试步骤

该测试步骤如下：

a) 设置测试仪表开始捕捉消息；
b) DUT 用如下 QoS 策略创建 DataWriter：

reliability.kind = RELIABLE
durability.kind = TRANSIENT_LOCAL
history.kind = KEEP_ALL

c) 启动 DUT，发送 5 个 DATA 子消息；
d) 测试仪表用如下 QoS 策略创建与 DUT 相同 Topic 的 DataReader：
reliability.kind = RELIABLE
durability.kind = VOLATILE
history.kind = KEEP_ALL
e) 待测试仪表与 DUT 建立连接；
f) 测试仪表记录并验证从 DUT 收到的消息序列。

5.2.4.2.7 通过条件

按 5.2.4.2.6 中 f)：测试仪表收到 GAP 子消息。
按 5.2.4.2.6 中 f)：GAP 子消息字段值应满足要求：
gapStart = 1
gapList：
gapList.bitmapBase = 6
gapList.numBits = 0

5.2.4.3 HEARTBEAT 子消息

5.2.4.3.1 测试目标

验证 DUT 在可靠有状态写操作发送 HEARTBEAT 子消息的操作是否正确。

5.2.4.3.2 测试概要

验证 DUT 在可靠有状态写操作发送 HEARTBEAT 子消息的过程是否符合标准中定义的状态机。

5.2.4.3.3 适用条件

DUT 支持可靠有状态写操作。

5.2.4.3.4 前置条件

符合附录 A.2 测试前置条件。

5.2.4.3.5 测试环境

符合附录 A.1 测试环境 1。

5.2.4.3.6 测试步骤

该测试步骤如下：
a) DUT 创建可靠有状态 RTPS Writer，确认 Writer.pushMode=false;
b) 设置 DUT 的 Writer.heartbeatPeriod = 3s；
c) 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动 DUT 和测试仪表，待其建立连接；
f) 等待至少 30s；
g) 测试仪表记录并验证从 DUT 发送的消息；

h) 设置 DUT 的 Writer.heartbeatPeriod = 5s；
i) 设置测试仪表开始捕捉 DUT 发送的消息；
j) 重启 DUT 和测试仪表，待其建立连接；
k) 等待至少 30s；
l) 测试仪表记录并验证从 DUT 发送的消息。

5.2.4.3.7 通过条件

按 5.2.4.3.6 中 g)：测试仪表收到 DUT 发送的连续的 HEARTBEAT 子消息，发送周期 3s。

按 5.2.4.3.6 中 g)：测试仪表收到 DUT 发送的 HEARTBEAT 子消息，HEARTBEAT.FinalFlag = 0（NOT_SET）。

按 5.2.4.3.6 中 l)：测试仪表收到 DUT 发送的连续的 HEARTBEAT 子消息，发送周期 5s。

按 5.2.4.3.6 中 l)：测试仪表收到 DUT 发送的 HEARTBEAT 子消息，HEARTBEAT.FinalFlag = 0（NOT_SET）。

5.2.4.4 nackResponseDelay 的值

5.2.4.4.1 测试目标

验证 DUT 在可靠有状态写操作过程中 nackResponseDelay 的值。

5.2.4.4.2 测试概要

验证 DUT 在可靠有状态写操作过程中是否按 nackResponseDelay 设置的时间回复 DATA。

5.2.4.4.3 适用条件

DUT 支持可靠有状态写操作。

5.2.4.4.4 前置条件

符合附录 A.2 测试前置条件。

5.2.4.4.5 测试环境

符合附录 A.1 测试环境 1。

5.2.4.4.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Writer；
b) 设置 DUT 的 nackRsponseDelay = 3s；
c) 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
d) 设置测试仪表开始捕捉 DUT 发送的消息；
e) 启动 DUT 和测试仪表，待其建立连接；
f) DUT 发送 10 个 DATA 子消息；
g) 测试仪表发送 ACKNACK 消息请求重发最后 1 个 DATA 子消息，并开始计时；
h) 测试仪表捕捉 DATA 子消息；
i) 测试仪表验证收到的 DATA 子消息的内容；
j) 测试仪表计算从步骤 7）到步骤 8）经过的时间 t1；
k) 设置 DUT 的 nackRsponseDelay = 5s；
l) 设置测试仪表开始捕捉 DUT 发送的消息；

m） 重新启动 DUT 和测试仪表，待其建立连接；
n） DUT 发送 10 个 DATA 子消息；
o） 测试仪表发送 ACKNACK 消息请求重发最后 1 个 DATA 子消息，并开始计时；
p） 测试仪表捕捉 DATA 子消息；
q） 测试仪表验证收到的 DATA 子消息的内容；
r） 测试仪表计算从步骤 7）到步骤 8）经过的时间 t2。

5.2.4.4.7 通过条件

按 5.2.4.4.6 中 h）：测试仪表收到 1 个 DATA 子消息。

按 5.2.4.4.6 中 i）：测试仪表收到的 DATA 子消息与步骤 f）DUT 发送的最后一个 DATA 子消息的内容一致。

按 5.2.4.4.6 中 j）：3s < t1 < 4s。

按 5.2.4.4.6 中 p）：测试仪表收到 1 个 DATA 子消息。

按 5.2.4.4.6 中 q）：测试仪表收到的 DATA 子消息与步骤 n）DUT 发送的最后一个 DATA 子消息的内容一致。

按 5.2.4.4.6 中 r）：5s < t2 < 6s。

5.2.4.5 数据重发操作

5.2.4.5.1 测试目标

验证 DUT 可靠有状态写操作的数据重发操作是否正确。

5.2.4.5.2 测试概要

验证 DUT 在可靠有状态写操作过程中是否可以重新发送 RTPS Reader 没有正确接收的 DATA 子消息。

5.2.4.5.3 适用条件

DUT 支持可靠有状态写操作。

5.2.4.5.4 前置条件

符合附录 A.2 测试前置条件。

5.2.4.5.5 测试环境

符合附录 A.1 测试环境 1。

5.2.4.5.6 测试步骤

该测试步骤如下：

a） DUT 创建可靠有状态 RTPS Writer；
b） 测试仪表创建可靠 RTPS Reader，Topic 与 DUT 相同；
c） 设置测试仪表开始捕捉 DUT 发送的消息；
d） 启动 DUT 和测试仪表，待其建立连接；
e） DUT 发送 10 个 DATA 子消息；
f） 测试仪表发送 ACKNACK 消息请求重发前 5 个 DATA 子消息；
g） 测试仪表捕捉 DATA 子消息；
h） 测试仪表验证收到的 DATA 子消息的内容；

i） DUT 重新发送 10 个 DATA 子消息；
j） 测试仪表发送 ACKNACK 消息请求重发后 5 个 DATA 子消息；
k） 测试仪表捕捉 DATA 子消息；
l） 测试仪表验证收到的 DATA 子消息的内容。

5.2.4.5.7 通过条件

按 5.2.4.5.6 中 g）：测试仪表收到 5 个 DATA 子消息。

按 5.2.4.5.6 中 h）：测试仪表收到的 DATA 子消息与步骤 e）DUT 发送的前 5 个 DATA 子消息的内容一致。

按 5.2.4.5.6 中 k）：测试仪表收到 5 个 DATA 子消息。

按 5.2.4.5.6 中 l）：测试仪表收到的 DATA 子消息与步骤 i）DUT 发送的后 5 个 DATA 子消息的内容一致。

5.2.5 尽力无状态读操作

5.2.5.1 测试目标

验证 DUT 尽力无状态读操作接收 DATA 子消息的操作是否正确。

5.2.5.2 测试概要

验证 DUT 尽力无状态读操作接收 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.5.3 适用条件

DUT 支持尽力无状态读操作。

5.2.5.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.5.5 测试环境

符合附录 A.1 测试环境 1。

5.2.5.6 测试步骤

该测试步骤如下：
a） 测试仪表创建尽力无状态 RTPS Writer；
b） DUT 创建尽力无状态 RTPS Reader，Topic 与 DUT 相同；
c） 启动 DUT 和测试仪表，待其建立连接；
d） 测试仪表发送至少 10 个 DATA 子消息；
e） 测试仪表查验 DUT，记录并验证 DUT 收到的消息。

5.2.5.7 通过条件

按 5.2.5.6 中 e）：DUT 收到测试仪表发送的 DATA 子消息。

按 5.2.5.6 中 e）：DUT 收到的 DATA 子消息与步骤 d）测试仪表发送的 DATA 子消息的数量和内容一致。

5.2.6 尽力有状态读操作

5.2.6.1 测试目标

验证 DUT 尽力有状态读操作接收 DATA 子消息的操作是否正确。

5.2.6.2 测试概要

验证 DUT 尽力有状态读操作接收 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.6.3 适用条件

DUT 支持尽力有状态读操作。

5.2.6.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.6.5 测试环境

符合附录 A.1 测试环境 1。

5.2.6.6 测试步骤

该测试步骤如下：

a) DUT 创建尽力有状态 RTPS Reader；
b) 测试仪表创建尽力有状态 RTPS Writer，Topic 与 DUT 相同；
c) 启动 DUT 和测试仪表，待其建立连接；
d) 测试仪表发送至少 10 个 DATA 子消息；
e) 测试仪表查验 DUT，记录并验证 DUT 收到的消息。

5.2.6.7 通过条件

按 5.2.6.6 中 e)：DUT 收到测试仪表发送的 DATA 子消息。

按 5.2.6.6 中 e)：DUT 收到的 DATA 子消息与步骤 d）测试仪表发送的 DATA 子消息的数量和内容一致。

5.2.7 可靠有状态读操作

5.2.7.1 接收 DATA 子消息

5.2.7.1.1 测试目标

验证 DUT 可靠有状态读操作接收 DATA 子消息的操作是否正确。

5.2.7.1.2 测试概要

验证 DUT 可靠有状态读操作接收 DATA 子消息的过程是否符合标准中定义的状态机。

5.2.7.1.3 适用条件

DUT 支持可靠有状态读操作。

5.2.7.1.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.7.1.5 测试环境

符合附录 A.1 测试环境 1。

5.2.7.1.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Reader；
b) 测试仪表创建可靠有状态 RTPS Writer，Topic 与 DUT 相同；
c) 启动 DUT 和测试仪表，待其建立连接；
d) 测试仪表发送至少 10 个 DATA 子消息；
e) 测试仪表查验 DUT，记录并验证 DUT 收到的消息。

5.2.7.1.7 通过条件

按 5.2.7.1.6 中 e)：DUT 收到测试仪表发送的 DATA 子消息。

按 5.2.7.1.6 中 e)：DUT 收到的 DATA 子消息与步骤 d）测试仪表发送的 DATA 子消息的数量和内容一致。

5.2.7.2 必须回复 ack 操作

5.2.7.2.1 测试目标

验证 DUT 在可靠有状态读操作过程中必须回复 ACKNACK 子消息的操作是否正确。

5.2.7.2.2 测试概要

验证 DUT 在可靠有状态读操作过程中，当收到的子消息 HEARTBEAT 中的 FinalFlag=NOT_SET 时 DUT 必须回复 ACKNACK 子消息的过程是否符合标准中定义的状态机。

5.2.7.2.3 适用条件

DUT 支持可靠有状态读操作。

5.2.7.2.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.7.2.5 测试环境

符合附录 A.1 测试环境 1。

5.2.7.2.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Reader；
b) 测试仪表创建可靠有状态 RTPS Writer，Topic 与 DUT 相同；
c) 设置测试仪表 HEARTBEAT.FinalFlag = NOT_SET；
d) 启动 DUT 和测试仪表，待其建立连接；
e) 测试仪表发送至少 10 个 DATA 子消息；
f) 测试仪表记录并验证与 DUT 交互的消息；
g) 测试仪表查验 DUT，记录并验证 DUT 收到的消息。

5.2.7.2.7 通过条件

按 5.2.7.2.6 中 f)：DUT 在收到 HEARTBEAT 消息后经过 heartbeatResponseDelay 发送 ACKNACK 子消息。

按 5.2.7.2.6 中 g)：DUT 收到的 DATA 子消息与步骤 e）测试仪表发送的 DATA 子消息内容一致。

5.2.7.3 也许回复 ack 操作

5.2.7.3.1 测试目标

验证 DUT 在可靠有状态读操作过程中也许回复 ACKNACK 子消息的操作是否正确。

5.2.7.3.2 测试概要

验证 DUT 在可靠有状态读操作过程中，当收到的子消息 HEARTBEAT 中的 FinalFlag=SET 时 DUT 也许回复 ACKNACK 子消息的过程是否符合标准中定义的状态机。

5.2.7.3.3 适用条件

DUT 支持可靠有状态读操作。

5.2.7.3.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.7.3.5 测试环境

符合附录 A.1 测试环境 1。

5.2.7.3.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Reader；
b) 测试仪表创建可靠有状态 RTPS Writer，Topic 与 DUT 相同；
c) 设置测试仪表 HEARTBEAT.FinalFlag = SET；
d) 启动 DUT 和测试仪表，待其建立连接；
e) 测试仪表发送至少 10 个 DATA 子消息；
f) 测试仪表记录并验证与 DUT 交互的消息；
g) 测试仪表查验 DUT，记录并验证 DUT 收到的消息。

5.2.7.3.7 通过条件

按 5.2.7.3.6 中 f):DUT 可能在收到 HEARTBEAT 消息后经过 heartbeatResponseDelay 发送 ACKNACK 子消息。

按 5.2.7.3.6 中 g)：DUT 收到的 DATA 子消息与步骤 e）测试仪表发送的 DATA 子消息内容一致。

5.2.7.4 请求重发数据

5.2.7.4.1 测试目标

验证 DUT 在可靠有状态写操作过程中请求重新发送数据的操作是否正确。

5.2.7.4.2 测试概要

验证 DUT 在可靠有状态写操作过程中如遇到数据丢失，要求写入者重新发送数据的操作是否符合

标准中定义的状态机。

5.2.7.4.3 适用条件

DUT 支持可靠有状态读操作。

5.2.7.4.4 前置条件

DUT 给测试仪表提供接收到的 DATA 子消息。

5.2.7.4.5 测试环境

符合附录 A.1 测试环境 1。

5.2.7.4.6 测试步骤

该测试步骤如下：

a) DUT 创建可靠有状态 RTPS Reader；
b) 测试仪表创建可靠有状态 RTPS Writer，Topic 与 DUT 相同；
c) 设置测试仪表 HEARTBEAT.FinalFlag = NOT_SET；
d) 启动 DUT 和测试仪表，待其建立连接；
e) 设置测试仪表开始捕捉与 DUT 交互的消息；
f) 测试仪表发送 2 个 DATA 子消息（seq_num = 1..2）；
g) 等待 3s；
h) 测试仪表发送 2 个 DATA 子消息（seq_num = 4..5）；
i) 测试仪表记录并验证与 DUT 交互的消息；
j) 测试仪表查验 DUT，记录并验证 DUT 收到的消息；
k) 测试仪表创建一个 DATA 子消息（seq_num = 3）并发送；
l) 测试仪表记录并验证与 DUT 交互的消息；
m) 测试仪表检查 DUT 收到的消息。

5.2.7.4.7 通过条件

按 5.2.7.4.6 中 i)：测试仪表收到 DUT 发送的 ACKNACK 子消息，请求重发 seq_num = 3 的 DATA 子消息。

按 5.2.7.4.6 中 j)：DUT 收到 4 个 DATA 子消息（seq_num = 1..2 和 seq_num = 4..5）。

按 5.2.7.4.6 中 l)：测试仪表收到 DUT 发送的 ACKNACK 子消息，确认收到 seq_num = 3 的 DATA 子消息。

按 5.2.7.4.6 中 m)：DUT 收到 1 个 DATA 子消息（seq_num = 3）。

5.3 发现协议建立连接操作测试

5.3.1 通知周期

5.3.1.1 测试目标

验证 DUT 启动时发现协议发送 SPDPdiscoveredParticipantData 的周期参数是否有效。

5.3.1.2 测试概要

验证 DUT 启动时发现协议发送 SPDPdiscoveredParticipantData 的周期参数是否有效。

5.3.1.3 适用条件

所有支持 DDS 协议的设备。

5.3.1.4 前置条件

符合附录 A.2 测试前置条件；
DUT 提供发送 SPDPdiscoveredParticipantData 的周期 AnnouncementPeriod。

5.3.1.5 测试环境

符合附录 A.1 测试环境 1。

5.3.1.6 测试步骤

该测试步骤如下：

a) DUT 创建 Publisher；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT；
d) 测试仪表记录并验证从 DUT 收到的消息，记录时间不小于 AnnouncementPeriod x 20。

5.3.1.7 通过条件

按 5.3.1.6 中 d）：DUT 发送连续的 SPDPdiscoveredParticipantData 子消息，发送周期为 announcementperiod。

5.3.2 租期

5.3.2.1 测试目标

验证 DUT 启动时发现协议发送 SPDPdiscoveredParticipantData 的租期参数是否有效。

5.3.2.2 测试概要

验证 DUT 启动时发现协议发送的 SPDPdiscoveredParticipantData 的租期参数是否有效。

5.3.2.3 适用条件

所有支持 DDS 协议的设备。

5.3.2.4 前置条件

符合附录 A.2 测试前置条件；
DUT 提供租期 leaseDuration。

5.3.2.5 测试环境

符合附录 A.1 测试环境 1。

5.3.2.6 测试步骤

该测试步骤如下：

a) DUT 创建 Publisher；
b) 配置 DUT Participant 的通知周期 announcementperiod < leaseDuration；
c) 设置测试仪表开始捕捉 DUT 发送的消息；
d) 启动 DUT；

e) 测试仪表记录并验证从 DUT 收到的消息，记录时间不小于 AnnouncementPeriod x 20 或 leaseDuration x 2 中的最大者；
f) 停止 DUT；
g) 配置 DUT Participant 的通知周期 announcementperiod > leaseDuration；
h) 设置测试仪表开始捕捉 DUT 发送的消息；
i) 启动 DUT；
j) 测试仪表记录并验证从 DUT 收到的消息，记录时间不小于 AnnouncementPeriod x 20 或 leaseDuration x 2 中的最大者。

5.3.2.7 通过条件

按 5.3.2.6 中 e)：捕捉到 SPDPdiscoveredParticipantData，周期为 announcementperiod。
按 5.3.2.6 中 j)：不应捕捉到 SPDPdiscoveredParticipantData。

5.3.3 组播地址列表

5.3.3.1 测试目标

验证 DUT 启动时发现协议的 defaultMulticastLocatorList 的值是否有效。

5.3.3.2 测试概要

验证 DUT 启动时发现协议是否按照 defaultMulticastLocatorList 中的默认组播地址列表发送 SPDPdiscoveredParticipantData。

5.3.3.3 适用条件

所有支持 DDS 协议的设备。

5.3.3.4 前置条件

符合附录 A.2 测试前置条件。

5.3.3.5 测试环境

符合附录 A.1 测试环境 1。

5.3.3.6 测试步骤

该测试步骤如下：
a) DUT 创建 Publisher；
b) 设置测试仪表开始捕捉 DUT 发送的消息；
c) 启动 DUT；
d) 测试仪表记录所有从 DUT 收到的消息，记录时间不少于 30s；
e) 测试仪表提取并验证从 DUT 收到的 SPDPdiscoveredParticipantData 子消息。

5.3.3.7 通过条件

按 5.3.3.6 中 e)：收到 DUT 周期发送的 SPDPdiscoveredParticipantData 子消息。
按 5.3.3.6 中 e)：DUT 向如下组播地址发送 SPDPdiscoveredParticipantData。
IP 地址：239.255.0.1
端口号：PB + DG * domainId + d2
其中，

DG = DomainId Gain = 250
PG = ParticipantId Gain = 2
PB = Port Base number = 7400
d0, d1, d2, d3 = additional offsets
d0 = 0
d1 = 10
d2 = 1
d3 = 11

5.3.4 单播地址列表

5.3.4.1 测试目标

验证DUT启动时发现协议的defaultUnicastLocatorList的值是否有效。

5.3.4.2 测试概要

验证 DUT 启动时发现协议是否按照 defaultUnicastLocatorList 中的默认单播地址列表发送SPDPdiscoveredParticipantData。

5.3.4.3 适用条件

所有支持DDS协议的设备。

5.3.4.4 前置条件

符合附录A.2测试前置条件；
厂商提供defaultUnicastLocatorList的值。

5.3.4.5 测试环境

符合附录A.1测试环境1。

5.3.4.6 测试步骤

该测试步骤如下：
a) DUT创建Publisher；
b) 设置测试仪表开始捕捉DUT发送的消息；
c) 启动DUT；
d) 测试仪表记录所有从DUT收到的消息，记录时间不少于30s；
e) 测试仪表提取并验证从DUT收到的SPDPdiscoveredParticipantData子消息。

5.3.4.7 通过条件

按5.3.4.6中d)：收到DUT周期发送的SPDPdiscoveredParticipantData子消息。
按5.3.4.6中d)：DUT向如下组播地址发送SPDPdiscoveredParticipantData子消息。
IP地址：厂商提供的defaultUnicastLocatorList包含的所有单播IP地址
端口号：PB + DG * domainId + d3 + PG * participantId
其中，
DG = DomainId Gain = 250
PG = ParticipantId Gain = 2
PB = Port Base number = 7400

d0, d1, d2, d3 = additional offsets
d0 = 0
d1 = 10
d2 = 1
d3 = 11

按 5.3.4.6 中 e)：SPDPdiscoveredParticipantData 子消息包含的 defaultUnicastLocatorList 字段里的 UnicastLocator 信息与厂商提供的信息相同。

5.3.5 作为 Publisher 建立连接

5.3.5.1 测试目标

验证 DUT 启动时作为 Publisher 发现协议的过程是否正确。

5.3.5.2 测试概要

验证 DUT 启动时作为 Publisher 发现协议的过程是否正确。

5.3.5.3 适用条件

所有支持 DDS 协议的设备。

5.3.5.4 前置条件

符合附录 A.2 测试前置条件；

5.3.5.5 测试环境

符合附录 A.1 测试环境 2。

5.3.5.6 测试步骤

该测试步骤如下：

a) DUT 创建 Publisher；
b) 设置测试仪表开始捕捉消息；
c) 启动 DUT；
d) 测试仪表记录时间不少于 10s；
e) 测试仪表提取并验证从 DUT 收到的 SPDPdiscoveredParticipantData 子消息；
f) LP 创建 subscriber,Topic 与 DUT 相同；
g) 启动 LP；
h) 等待不少于 10s；
i) DUT 发送至少 10 个 DATA 子消息；
j) 测试仪表验证 DUT 与 LP 交互的消息;
k) 记录并验证从 LP 收到的 DATA 子消息。

5.3.5.7 通过条件

按 5.3.5.6 中 e)：DUT 向组播地址 239.255.0.1 周期发送的 SPDPdiscoveredParticipantData 子消息。

按 5.3.5.6 中 e)：SPDPdiscoveredParticipantData 子消息包含如下信息：

readerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_DETECTOR (0x000100c7)
writerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_ANNOUNCER (0x000100c2)

entityId = ENTITYID_PARTICIPANT (0x000001c1)

按 5.3.5.6 中 j)：收到如下消息：

LP 向组播地址 239.255.0.1 周期发送的 SPDPdiscoveredParticipantData 子消息：

readerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_DETECTOR (0x000100c7)

writerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_ANNOUNCER (0x000100c2)

entityId = ENTITYID_PARTICIPANT (0x000001c1)

DUT 向 LP 发送的单播 SPDPdiscoveredParticipantData 子消息：

readerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_DETECTOR (0x000100c7)

writerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_ANNOUNCER (0x000100c2)

entityId = ENTITYID_PARTICIPANT (0x000001c1)

LP 向 DUT 发送的单播 SPDPdiscoveredParticipantData 子消息：

readerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_DETECTOR (0x000100c7)

writerId = ENTITYID_SPDP_BUILTIN_PARTICIPANT_ANNOUNCER (0x000100c2)

entityId = ENTITYID_PARTICIPANT (0x000001c1)

DUT 向 LP 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_DETECTOR (0x000003c7)

writerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_ANNOUNCER (0x000003c2)

DUT 向 LP 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_DETECTOR (0x000004c7)

writerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_ANNOUNCER (0x000004c2)

DUT 向 LP 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_P2P_BUILTIN_PARTICIPANT_MESSAGE_READER (0x000200c7)

writerId = ENTITYID_P2P_BUILTIN_PARTICIPANT_MESSAGE_WRITER (0x000200c2)

LP 向 DUT 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_DETECTOR (0x000003c7)

writerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_ANNOUNCER (0x000003c2)

LP 向 DUT 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_DETECTOR (0x000004c7)

writerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_ANNOUNCER (0x000004c2)

LP 向 DUT 发送的单播 Heartbeat 子消息：

readerId = ENTITYID_P2P_BUILTIN_PARTICIPANT_MESSAGE_READER (0x000200c7)

writerId = ENTITYID_P2P_BUILTIN_PARTICIPANT_MESSAGE_WRITER (0x000200c2)

LP 向 DUT 发送的单播 DATA 子消息：

readerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_DETECTOR (0x000004c7)

writerId = ENTITYID_SEDP_BUILTIN_SUBSCRIPTIONS_ANNOUNCER (0x000004c2)

PID_PARTICIPANT_GUID

entityKind = 0xc1

PID_TOPIC_NAME

topicName = <topicName>

PID_TYPE_NAME

typeName = <typeName>

PID_ENDPOINT_GUID

entityKind =

0x04 (no Key)
0x07 (with Key)
DUT 向 LP 发送的单播 DATA 子消息：
readerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_DETECTOR (0x000003c7)
writerId = ENTITYID_SEDP_BUILTIN_PUBLICATIONS_ANNOUNCER (0x000003c2)
PID_PARTICIPANT_GUID
entityKind = 0xc1
PID_TOPIC_NAME
topicName = <topicName>
PID_TYPE_NAME
typeName = <typeName>
PID_ENDPOINT_GUID
entityKind =
0x03 (no Key)
0x02 (with Key)

按 5.3.5.6 中 k)：LP 收到的 DATA 子消息与步骤 i）DUT 发送的 DATA 子消息的数量和内容一致。

5.3.6 作为 Subscriber 建立连接

5.3.6.1 测试目标

验证 DUT 启动时作为 Subscriber 发现协议的过程是否正确。

5.3.6.2 测试概要

验证 DUT 启动时作为 Subscriber 发现协议的过程是否正确。

5.3.6.3 适用条件

所有支持 DDS 协议的设备。

5.3.6.4 前置条件

符合附录 A.2 测试前置条件；

5.3.6.5 测试环境

符合附录 A.1 测试环境 2。

5.3.6.6 测试步骤

该测试步骤如下：

a) LP 创建 Publisher,Topic 与 DUT 相同；
b) 设置测试仪表开始捕捉消息；
c) 启动 LP；
d) 测试仪表记录时间不少于 10s；
e) 测试仪表提取并验证从 LP 收到的 SPDPdiscoveredParticipantData 子消息；
f) DUT 创建 subscriber；
g) 启动 DUT；
h) 等待不少于 10s；
i) LP 发送至少 10 个 DATA 子消息；